

Mule

1.0.337

Generated by Doxygen 1.8.8

Tue Oct 7 2014 09:39:47

Contents

1 Multiscale	1
1.1 Brief description	1
1.2 Contact	1
2 Namespace Index	3
2.1 Namespace List	3
3 Hierarchical Index	5
3.1 Class Hierarchy	5
4 Class Index	11
4.1 Class List	11
5 File Index	21
5.1 File List	21
6 Namespace Documentation	35
6.1 multiscale Namespace Reference	35
6.1.1 Enumeration Type Documentation	37
6.1.1.1 ColourCode	37
6.1.1.2 NumberIteratorType	37
6.1.1.3 UnixColourCode	38
6.1.1.4 WindowsColourCode	38
6.1.2 Variable Documentation	38
6.1.2.1 ERR_INDEX_OUT_OF_BOUNDS_BEGIN	38
6.1.2.2 ERR_INDEX_OUT_OF_BOUNDS_END	38
6.1.2.3 ERR_MSG	39
6.1.2.4 ERR_UNDEFINED_ENUM_VALUE	39
6.1.2.5 ERR_UNIMPLEMENTED_METHOD	39
6.1.2.6 EXEC_ERR_CODE	39
6.1.2.7 EXEC_SUCCESS_CODE	39
6.2 multiscale::analysis Namespace Reference	39
6.2.1 Typedef Documentation	40

6.2.1.1	Polygon	40
6.2.2	Enumeration Type Documentation	41
6.2.2.1	Shape2D	41
6.2.2.2	SpatialEntityPseudo3DType	41
6.3	multiscale::verification Namespace Reference	41
6.3.1	Typedef Documentation	53
6.3.1.1	ConstraintAttributeType	53
6.3.1.2	FilterNumericMeasureAttributeType	53
6.3.1.3	LogicPropertyAttributeType	53
6.3.1.4	NumericMeasureCollectionType	53
6.3.1.5	NumericMeasureType	53
6.3.1.6	NumericSpatialMeasureType	53
6.3.1.7	NumericStatisticalMeasureType	54
6.3.1.8	PrimaryConstraintAttributeType	54
6.3.1.9	PrimaryLogicPropertyAttributeType	54
6.3.1.10	PrimaryNumericMeasureAttributeType	54
6.3.1.11	SubsetAttributeType	54
6.3.1.12	TemporalNumericCollectionType	54
6.3.1.13	TemporalNumericMeasureType	55
6.3.1.14	TimeseriesComponentType	55
6.3.2	Enumeration Type Documentation	55
6.3.2.1	ApproximateBayesianModelCheckingResult	55
6.3.2.2	BayesianModelCheckingResult	55
6.3.2.3	BinaryNumericMeasureType	55
6.3.2.4	BinaryStatisticalMeasureType	56
6.3.2.5	BinaryStatisticalQuantileMeasureType	56
6.3.2.6	ChangeMeasureType	56
6.3.2.7	ComparatorType	56
6.3.2.8	HeterogeneousTimeseriesComponentType	56
6.3.2.9	HomogeneousTimeseriesComponentType	57
6.3.2.10	HomogeneousTimeseriesMeasureType	57
6.3.2.11	SimilarityMeasureType	57
6.3.2.12	SpatialMeasureType	57
6.3.2.13	StatisticalModelCheckingResult	58
6.3.2.14	SubsetOperationType	58
6.3.2.15	SubsetSpecificType	58
6.3.2.16	TimeseriesMeasureType	58
6.3.2.17	UnaryNumericMeasureType	59
6.3.2.18	UnaryStatisticalMeasureType	59
6.3.3	Function Documentation	59

6.3.3.1	operator<<	59
6.3.3.2	operator<<	60
6.3.3.3	operator<<	60
6.3.3.4	operator<<	60
6.3.3.5	operator<<	60
6.3.3.6	operator<<	61
6.3.3.7	operator<<	61
6.3.3.8	operator<<	61
6.3.3.9	operator<<	61
6.3.3.10	operator<<	62
6.3.3.11	operator<<	62
6.3.3.12	operator<<	62
6.3.3.13	operator<<	62
6.3.3.14	operator<<	63
6.3.3.15	operator<<	63
6.3.4	Variable Documentation	63
6.3.4.1	handleProbabilityError	63
6.3.4.2	handleUnexpectedTokenError	63
6.3.4.3	NR_SPATIAL_MEASURE_TYPES	64
6.3.4.4	NR_SUBSET_SPECIFIC_TYPES	64
6.3.4.5	WRN_LOGIC_PROPERTY_EVAL_FALSE	64
6.3.4.6	WRN_OUTPUT_SEPARATOR	64
6.4	multiscale::verification::spatialmeasure Namespace Reference	64
6.4.1	Function Documentation	65
6.4.1.1	computeSpatialMeasureType	65
6.4.1.2	computeSpatialMeasureTypeIndex	65
6.4.1.3	getMaxValidSpatialMeasureValue	65
6.4.1.4	getMinValidSpatialMeasureValue	66
6.4.1.5	validateSpatialMeasureType	66
6.4.1.6	validateSpatialMeasureTypeIndex	66
6.5	multiscale::verification::subsetsspecific Namespace Reference	67
6.5.1	Function Documentation	67
6.5.1.1	computeSubsetSpecificType	67
6.5.1.2	computeSubsetSpecificTypeIndex	67
6.5.1.3	validateSubsetSpecificType	68
6.5.1.4	validateSubsetSpecificTypeIndex	68
6.6	multiscale::video Namespace Reference	68
6.7	multiscaletest Namespace Reference	69
6.8	multiscaletest::verification Namespace Reference	69
6.8.1	Function Documentation	69

6.8.1.1	parseInputString	69
7	Class Documentation	71
7.1	multiscale::verification::AbstractSyntaxTree Class Reference	71
7.1.1	Detailed Description	73
7.1.2	Constructor & Destructor Documentation	73
7.1.2.1	AbstractSyntaxTree	73
7.1.2.2	\sim AbstractSyntaxTree	73
7.1.3	Member Function Documentation	73
7.1.3.1	evaluate	73
7.1.3.2	getComparator	73
7.1.3.3	getProbability	74
7.1.3.4	initialiseTree	74
7.1.4	Member Data Documentation	74
7.1.4.1	ERR_ABSTRACT_SYNTAX_TREE_NOT_INITIALISED	74
7.1.4.2	isInitialised	74
7.1.4.3	probabilisticLogicProperty	74
7.2	multiscale::AdditionOperation Class Reference	75
7.2.1	Detailed Description	75
7.2.2	Member Function Documentation	75
7.2.2.1	operator()	75
7.3	multiscale::AlgorithmException Class Reference	76
7.3.1	Detailed Description	78
7.3.2	Constructor & Destructor Documentation	78
7.3.2.1	AlgorithmException	78
7.3.2.2	AlgorithmException	78
7.3.2.3	AlgorithmException	78
7.4	multiscale::verification::AndConstraintAttribute Class Reference	78
7.4.1	Detailed Description	79
7.4.2	Member Data Documentation	79
7.4.2.1	constraint	79
7.5	multiscale::verification::AndLogicPropertyAttribute Class Reference	79
7.5.1	Detailed Description	79
7.5.2	Member Data Documentation	79
7.5.2.1	logicProperty	80
7.6	multiscale::video::AnnularSector Class Reference	80
7.6.1	Detailed Description	81
7.6.2	Constructor & Destructor Documentation	81
7.6.2.1	AnnularSector	81
7.6.2.2	\sim AnnularSector	81

7.6.3	Member Function Documentation	81
7.6.3.1	getConcentration	81
7.6.3.2	getEndingAngle	81
7.6.3.3	getEndingRadius	82
7.6.3.4	getStartingAngle	82
7.6.3.5	getStartingRadius	82
7.6.3.6	initialise	82
7.6.3.7	toString	82
7.6.4	Member Data Documentation	82
7.6.4.1	concentration	82
7.6.4.2	endingAngle	82
7.6.4.3	endingRadius	83
7.6.4.4	startingAngle	83
7.6.4.5	startingRadius	83
7.7	multiscale::verification::ApproximateBayesianModelChecker Class Reference	83
7.7.1	Detailed Description	87
7.7.2	Constructor & Destructor Documentation	88
7.7.2.1	ApproximateBayesianModelChecker	88
7.7.2.2	~ApproximateBayesianModelChecker	88
7.7.3	Member Function Documentation	88
7.7.3.1	acceptsMoreTraces	88
7.7.3.2	doesPropertyHold	88
7.7.3.3	doesPropertyHoldConsideringResult	88
7.7.3.4	getDetailedResults	88
7.7.3.5	getDetailedUpdatedResults	89
7.7.3.6	initialise	89
7.7.3.7	isModelCheckingResultTrueConsideringComparator	89
7.7.3.8	isValidShapeParameter	89
7.7.3.9	requiresMoreTraces	89
7.7.3.10	updateDerivedModelCheckerForFalseEvaluation	90
7.7.3.11	updateDerivedModelCheckerForTrueEvaluation	90
7.7.3.12	updateMean	90
7.7.3.13	updateMeanAndVariance	90
7.7.3.14	updateModelCheckingResult	90
7.7.3.15	updateModelCheckingResult	90
7.7.3.16	updateModelCheckingResultEnoughTraces	91
7.7.3.17	updateModelCheckingResultNotEnoughTraces	91
7.7.3.18	updateVariance	91
7.7.3.19	validateInput	91
7.7.3.20	validateShapeParameters	92

7.7.3.21	validateVarianceThreshold	92
7.7.4	Member Data Documentation	92
7.7.4.1	alpha	92
7.7.4.2	beta	92
7.7.4.3	ERR_SHAPE_PARAMETERS_BEGIN	92
7.7.4.4	ERR_SHAPE_PARAMETERS_END	93
7.7.4.5	ERR_SHAPE_PARAMETERS_MIDDLE	93
7.7.4.6	ERR_UNEXPECTED_MODEL_CHECKING_RESULT	93
7.7.4.7	ERR_VARIANCE_THRESHOLD_BEGIN	93
7.7.4.8	ERR_VARIANCE_THRESHOLD_END	93
7.7.4.9	mean	93
7.7.4.10	modelCheckingResult	93
7.7.4.11	MSG_OUTPUT_MORE_TRACES_REQUIRED	93
7.7.4.12	MSG_OUTPUT_RESULT_BEGIN	94
7.7.4.13	MSG_OUTPUT_RESULT_END	94
7.7.4.14	MSG_OUTPUT_RESULT_MIDDLE1	94
7.7.4.15	MSG_OUTPUT_RESULT_MIDDLE2	94
7.7.4.16	MSG_OUTPUT_SEPARATOR	94
7.7.4.17	probability	94
7.7.4.18	variance	94
7.7.4.19	varianceThreshold	94
7.8	multiscale::verification::ApproximateBayesianModelCheckerFactory Class Reference	95
7.8.1	Detailed Description	96
7.8.2	Constructor & Destructor Documentation	97
7.8.2.1	ApproximateBayesianModelCheckerFactory	97
7.8.2.2	~ApproximateBayesianModelCheckerFactory	97
7.8.3	Member Function Documentation	97
7.8.3.1	createInstance	97
7.8.4	Member Data Documentation	97
7.8.4.1	alpha	97
7.8.4.2	beta	97
7.8.4.3	varianceThreshold	97
7.9	multiscaletest::ApproximateBayesianModelCheckerTest Class Reference	98
7.9.1	Detailed Description	101
7.9.2	Constructor & Destructor Documentation	101
7.9.2.1	ApproximateBayesianModelCheckerTest	101
7.9.3	Member Function Documentation	101
7.9.3.1	InitialiseModelChecker	101
7.9.3.2	SetAlphaParamForBetaPrior	101
7.9.3.3	SetBetaParamForBetaPrior	101

7.9.3.4	SetVarianceThreshold	102
7.9.4	Member Data Documentation	102
7.9.4.1	alphaParamForBetaPrior	102
7.9.4.2	betaParamForBetaPrior	102
7.9.4.3	varianceThreshold	102
7.10	multiscale::verification::ApproximateProbabilisticModelChecker Class Reference	102
7.10.1	Detailed Description	106
7.10.2	Constructor & Destructor Documentation	106
7.10.2.1	ApproximateProbabilisticModelChecker	106
7.10.2.2	\sim ApproximateProbabilisticModelChecker	106
7.10.3	Member Function Documentation	106
7.10.3.1	acceptsMoreTraces	106
7.10.3.2	doesPropertyHold	106
7.10.3.3	doesPropertyHoldConsideringProbabilityComparator	106
7.10.3.4	getDetailedResults	107
7.10.3.5	initialise	107
7.10.3.6	initialiseNumberOfRequiredTraces	107
7.10.3.7	isBetweenZeroAndOne	107
7.10.3.8	requiresMoreTraces	107
7.10.3.9	updateDerivedModelCheckerForFalseEvaluation	108
7.10.3.10	updateDerivedModelCheckerForTrueEvaluation	108
7.10.3.11	validateInput	108
7.10.4	Member Data Documentation	108
7.10.4.1	delta	108
7.10.4.2	epsilon	108
7.10.4.3	ERR_INVALID_INPUT_BEGIN	109
7.10.4.4	ERR_INVALID_INPUT_END	109
7.10.4.5	ERR_INVALID_INPUT_MIDDLE	109
7.10.4.6	MSG_OUTPUT_MORE_TRACES_REQUIRED	109
7.10.4.7	MSG_OUTPUT_RESULT_BEGIN	109
7.10.4.8	MSG_OUTPUT_RESULT_END	109
7.10.4.9	MSG_OUTPUT_RESULT_MIDDLE1	109
7.10.4.10	MSG_OUTPUT_RESULT_MIDDLE2	110
7.10.4.11	MSG_OUTPUT_SEPARATOR	110
7.10.4.12	nrOfRequiredTraces	110
7.10.4.13	probability	110
7.11	multiscale::verification::ApproximateProbabilisticModelCheckerFactory Class Reference	110
7.11.1	Detailed Description	112
7.11.2	Constructor & Destructor Documentation	113
7.11.2.1	ApproximateProbabilisticModelCheckerFactory	113

7.11.2.2	<code>~ApproximateProbabilisticModelCheckerFactory</code>	113
7.11.3	Member Function Documentation	113
7.11.3.1	<code>createInstance</code>	113
7.11.4	Member Data Documentation	113
7.11.4.1	<code>delta</code>	113
7.11.4.2	<code>epsilon</code>	113
7.12	<code>multiscaletest::ApproximateProbabilisticModelCheckerTest</code> Class Reference	114
7.12.1	Detailed Description	116
7.12.2	Constructor & Destructor Documentation	116
7.12.2.1	<code>ApproximateProbabilisticModelCheckerTest</code>	116
7.12.3	Member Function Documentation	116
7.12.3.1	<code>InitialiseModelChecker</code>	116
7.12.3.2	<code>SetDelta</code>	116
7.12.3.3	<code>SetEpsilon</code>	116
7.12.4	Member Data Documentation	117
7.12.4.1	<code>delta</code>	117
7.12.4.2	<code>epsilon</code>	117
7.13	<code>multiscale::verification::BayesianModelChecker</code> Class Reference	117
7.13.1	Detailed Description	121
7.13.2	Constructor & Destructor Documentation	122
7.13.2.1	<code>BayesianModelChecker</code>	122
7.13.2.2	<code>~BayesianModelChecker</code>	122
7.13.3	Member Function Documentation	122
7.13.3.1	<code>acceptsMoreTraces</code>	122
7.13.3.2	<code>computeBayesFactorValue</code>	122
7.13.3.3	<code>computeBinomialPDF</code>	122
7.13.3.4	<code>computeMaximumBinomialPDF</code>	123
7.13.3.5	<code>doesPropertyHold</code>	123
7.13.3.6	<code>doesPropertyHoldConsideringProbabilityComparator</code>	123
7.13.3.7	<code>doesPropertyHoldConsideringResult</code>	123
7.13.3.8	<code>getDetailedResults</code>	124
7.13.3.9	<code>getDetailedUpdatedResults</code>	124
7.13.3.10	<code>indicatorFunction</code>	124
7.13.3.11	<code>initialise</code>	124
7.13.3.12	<code>isValidShapeParameter</code>	124
7.13.3.13	<code>requiresMoreTraces</code>	126
7.13.3.14	<code>updateDerivedModelCheckerForFalseEvaluation</code>	126
7.13.3.15	<code>updateDerivedModelCheckerForTrueEvaluation</code>	126
7.13.3.16	<code>updateModelCheckingResult</code>	126
7.13.3.17	<code>updateModelCheckingResult</code>	126

7.13.3.18 updateModelCheckingResultEnoughTraces	127
7.13.3.19 updateModelCheckingResultNotEnoughTraces	127
7.13.3.20 updateTypeIErrorUpperBound	127
7.13.3.21 validateBayesFactorThreshold	127
7.13.3.22 validateInput	128
7.13.3.23 validateShapeParameters	128
7.13.4 Member Data Documentation	128
7.13.4.1 alpha	128
7.13.4.2 bayesFactorThreshold	128
7.13.4.3 bayesFactorThresholdInverse	128
7.13.4.4 beta	129
7.13.4.5 ERR_BAYES_FACTOR_THRESHOLD_BEGIN	129
7.13.4.6 ERR_BAYES_FACTOR_THRESHOLD_END	129
7.13.4.7 ERR_SHAPE_PARAMETERS_BEGIN	129
7.13.4.8 ERR_SHAPE_PARAMETERS_END	129
7.13.4.9 ERR_SHAPE_PARAMETERS_MIDDLE	129
7.13.4.10 ERR_UNEXPECTED_MODEL_CHECKING_RESULT	129
7.13.4.11 modelCheckingResult	129
7.13.4.12 MSG_OUTPUT_MORE_TRACES_REQUIRED	130
7.13.4.13 MSG_OUTPUT_RESULT_BEGIN	130
7.13.4.14 MSG_OUTPUT_RESULT_END	130
7.13.4.15 MSG_OUTPUT_RESULT_MIDDLE1	130
7.13.4.16 MSG_OUTPUT_RESULT_MIDDLE2	130
7.13.4.17 MSG_OUTPUT_RESULT_MIDDLE3	130
7.13.4.18 MSG_OUTPUT_SEPARATOR	130
7.13.4.19 probability	130
7.13.4.20 typeIErrorUpperBound	131
7.14 multiscale::verification::BayesianModelCheckerFactory Class Reference	131
7.14.1 Detailed Description	132
7.14.2 Constructor & Destructor Documentation	133
7.14.2.1 BayesianModelCheckerFactory	133
7.14.2.2 ~BayesianModelCheckerFactory	133
7.14.3 Member Function Documentation	133
7.14.3.1 createInstance	133
7.14.4 Member Data Documentation	133
7.14.4.1 alpha	133
7.14.4.2 bayesFactorThreshold	133
7.14.4.3 beta	133
7.15 multiscaletest::BayesianModelCheckerTest Class Reference	134
7.15.1 Detailed Description	137

7.15.2	Constructor & Destructor Documentation	137
7.15.2.1	BayesianModelCheckerTest	137
7.15.3	Member Function Documentation	137
7.15.3.1	InitialiseModelChecker	137
7.15.3.2	SetAlphaParamForBetaPrior	137
7.15.3.3	SetBayesFactorThreshold	137
7.15.3.4	SetBetaParamForBetaPrior	138
7.15.4	Member Data Documentation	138
7.15.4.1	alphaParamForBetaPrior	138
7.15.4.2	bayesFactorThreshold	138
7.15.4.3	betaParamForBetaPrior	138
7.16	multiscale::BetaDistribution Class Reference	138
7.16.1	Detailed Description	141
7.16.2	Member Function Documentation	141
7.16.2.1	cdf	141
7.16.2.2	computeCdf	141
7.16.2.3	isValidShapeParameter	141
7.16.2.4	validateShapeParameters	143
7.16.3	Member Data Documentation	143
7.16.3.1	ERR_SHAPE_PARAMETERS_BEGIN	143
7.16.3.2	ERR_SHAPE_PARAMETERS_END	143
7.16.3.3	ERR_SHAPE_PARAMETERS_MIDDLE	143
7.17	multiscale::verification::BinaryNumericFilterAttribute Class Reference	143
7.17.1	Detailed Description	144
7.17.2	Member Data Documentation	144
7.17.2.1	binaryNumericMeasure	144
7.17.2.2	firstFilterNumericMeasure	144
7.17.2.3	secondFilterNumericMeasure	145
7.18	multiscale::verification::BinaryNumericMeasureAttribute Class Reference	145
7.18.1	Detailed Description	145
7.18.2	Member Data Documentation	145
7.18.2.1	binaryNumericMeasureType	145
7.19	multiscale::verification::BinaryNumericMeasureGrammar< Iterator > Class Template Reference	146
7.19.1	Detailed Description	149
7.19.2	Constructor & Destructor Documentation	149
7.19.2.1	BinaryNumericMeasureGrammar	149
7.19.3	Member Function Documentation	149
7.19.3.1	assignNamesToRules	149
7.19.3.2	initialise	149
7.19.3.3	initialiseDebugSupport	149

7.19.3.4	initialiseGrammar	149
7.19.3.5	initialiseRulesDebugging	150
7.19.4	Member Data Documentation	150
7.19.4.1	binaryNumericMeasureRule	150
7.19.4.2	binaryNumericMeasureTypeParser	150
7.20	multiscale::verification::BinaryNumericMeasureTypeParser Struct Reference	150
7.20.1	Detailed Description	152
7.20.2	Constructor & Destructor Documentation	152
7.20.2.1	BinaryNumericMeasureTypeParser	152
7.21	multiscale::verification::BinaryNumericNumericAttribute Class Reference	153
7.21.1	Detailed Description	153
7.21.2	Member Data Documentation	153
7.21.2.1	binaryNumericMeasure	153
7.21.2.2	firstNumericMeasure	154
7.21.2.3	secondNumericMeasure	154
7.22	multiscale::verification::BinaryNumericTemporalAttribute Class Reference	154
7.22.1	Detailed Description	155
7.22.2	Member Data Documentation	155
7.22.2.1	binaryNumericMeasure	155
7.22.2.2	firstTemporalNumericMeasure	155
7.22.2.3	secondTemporalNumericMeasure	155
7.23	multiscale::verification::BinaryStatisticalMeasureAttribute Class Reference	155
7.23.1	Detailed Description	156
7.23.2	Member Data Documentation	156
7.23.2.1	binaryStatisticalMeasureType	156
7.24	multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator > Class Template Reference	156
7.24.1	Detailed Description	159
7.24.2	Constructor & Destructor Documentation	159
7.24.2.1	BinaryStatisticalMeasureGrammar	159
7.24.3	Member Function Documentation	159
7.24.3.1	assignNamesToRules	159
7.24.3.2	initialise	159
7.24.3.3	initialiseDebugSupport	159
7.24.3.4	initialiseGrammar	160
7.24.3.5	initialiseRulesDebugging	160
7.24.4	Member Data Documentation	160
7.24.4.1	binaryStatisticalMeasureRule	160
7.24.4.2	binaryStatisticalMeasureTypeParser	160
7.25	multiscale::verification::BinaryStatisticalMeasureTypeParser Struct Reference	160
7.25.1	Detailed Description	162

7.25.2 Constructor & Destructor Documentation	162
7.25.2.1 BinaryStatisticalMeasureTypeParser	162
7.26 multiscale::verification::BinaryStatisticalNumericAttribute Class Reference	163
7.26.1 Detailed Description	163
7.26.2 Member Data Documentation	163
7.26.2.1 binaryStatisticalMeasure	163
7.26.2.2 firstNumericMeasureCollection	163
7.26.2.3 secondNumericMeasureCollection	164
7.27 multiscale::verification::BinaryStatisticalQuantileMeasureAttribute Class Reference	164
7.27.1 Detailed Description	164
7.27.2 Member Data Documentation	164
7.27.2.1 binaryStatisticalQuantileMeasureType	165
7.28 multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator > Class Template Reference	165
7.28.1 Detailed Description	168
7.28.2 Constructor & Destructor Documentation	168
7.28.2.1 BinaryStatisticalQuantileMeasureGrammar	168
7.28.3 Member Function Documentation	168
7.28.3.1 assignNamesToRules	168
7.28.3.2 initialise	168
7.28.3.3 initialiseDebugSupport	168
7.28.3.4 initialiseGrammar	169
7.28.3.5 initialiseRulesDebugging	169
7.28.4 Member Data Documentation	169
7.28.4.1 binaryStatisticalQuantileMeasureRule	169
7.28.4.2 binaryStatisticalQuantileMeasureTypeParser	169
7.29 multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser Struct Reference	169
7.29.1 Detailed Description	171
7.29.2 Constructor & Destructor Documentation	171
7.29.2.1 BinaryStatisticalQuantileMeasureTypeParser	171
7.30 multiscale::verification::BinaryStatisticalQuantileNumericAttribute Class Reference	172
7.30.1 Detailed Description	172
7.30.2 Member Data Documentation	172
7.30.2.1 binaryStatisticalQuantileMeasure	172
7.30.2.2 numericMeasureCollection	173
7.30.2.3 parameter	173
7.31 multiscale::verification::BinaryStatisticalQuantileSpatialAttribute Class Reference	173
7.31.1 Detailed Description	174
7.31.2 Member Data Documentation	174
7.31.2.1 binaryStatisticalQuantileMeasure	174

7.31.2.2	parameter	174
7.31.2.3	spatialMeasureCollection	174
7.32	multiscale::verification::BinaryStatisticalSpatialAttribute Class Reference	174
7.32.1	Detailed Description	175
7.32.2	Member Data Documentation	175
7.32.2.1	binaryStatisticalMeasure	175
7.32.2.2	firstSpatialMeasureCollection	175
7.32.2.3	secondSpatialMeasureCollection	176
7.33	multiscale::BinomialDistribution Class Reference	176
7.33.1	Detailed Description	178
7.33.2	Member Function Documentation	178
7.33.2.1	cdf	178
7.33.2.2	computeCdf	178
7.33.2.3	computePdf	179
7.33.2.4	pdf	179
7.33.2.5	validateInput	179
7.33.2.6	validateNrOfSuccesses	180
7.33.3	Member Data Documentation	181
7.33.3.1	ERR_NR_OF_SUCCESSES_BEGIN	181
7.33.3.2	ERR_NR_OF_SUCCESSES_END	181
7.33.3.3	ERR_NR_OF_SUCCESSES_MIDDLE	181
7.34	multiscale::video::CartesianToConcentrationsConverter Class Reference	181
7.34.1	Detailed Description	183
7.34.2	Constructor & Destructor Documentation	183
7.34.2.1	CartesianToConcentrationsConverter	183
7.34.2.2	~CartesianToConcentrationsConverter	183
7.34.3	Member Function Documentation	184
7.34.3.1	convert	184
7.34.3.2	outputResults	184
7.34.3.3	readConcentrations	184
7.34.3.4	readHeaderLine	184
7.34.3.5	readInputData	184
7.34.4	Member Data Documentation	185
7.34.4.1	concentrations	185
7.34.4.2	ERR_CONC	185
7.34.4.3	ERR_IN_EXTRA_DATA	185
7.34.4.4	ERR_INPUT_OPEN	185
7.34.4.5	ERR_NEG_SIM_TIME	185
7.34.4.6	ERR_NONPOS_DIMENSION	185
7.34.4.7	height	185

7.34.4.8 <inputfilepath></inputfilepath>	185
7.34.4.9 OUTPUT_FILE_EXTENSION	186
7.34.4.10 outputFilepath	186
7.34.4.11 RADIUS_MAX	186
7.34.4.12 RADIUS_MIN	186
7.34.4.13 simulationTime	186
7.34.4.14 width	186
7.35 multiscale::video::CartesianToPolarConverter Class Reference	186
7.35.1 Detailed Description	188
7.35.2 Constructor & Destructor Documentation	188
7.35.2.1 CartesianToPolarConverter	188
7.35.2.2 ~CartesianToPolarConverter	189
7.35.3 Member Function Documentation	189
7.35.3.1 convert	189
7.35.3.2 outputResultsAsFile	189
7.35.3.3 outputResultsAsScript	189
7.35.3.4 readConcentrations	189
7.35.3.5 readHeaderLine	189
7.35.3.6 readInputData	190
7.35.3.7 transformToAnnularSectors	190
7.35.4 Member Data Documentation	190
7.35.4.1 annularSectors	190
7.35.4.2 concentrations	190
7.35.4.3 ERR_CONC	190
7.35.4.4 ERR_IN_EXTRA_DATA	191
7.35.4.5 ERR_INPUT_OPEN	191
7.35.4.6 ERR_NEG_SIM_TIME	191
7.35.4.7 ERR_NONPOS_DIMENSION	191
7.35.4.8 inputFilepath	191
7.35.4.9 nrOfConcentricCircles	191
7.35.4.10 nrOfSectors	191
7.35.4.11 OUTPUT_FILE_EXTENSION	191
7.35.4.12 outputFilepath	192
7.35.4.13 RADIUS_MAX	192
7.35.4.14 RADIUS_MIN	192
7.35.4.15 simulationTime	192
7.36 multiscale::verification::ChangeMeasureAttribute Class Reference	192
7.36.1 Detailed Description	193
7.36.2 Member Data Documentation	193
7.36.2.1 changeMeasureType	193

7.37 multiscale::verification::ChangeMeasureEvaluator Class Reference	193
7.37.1 Detailed Description	194
7.37.2 Member Function Documentation	194
7.37.2.1 computeNumericMeasureValueChange	194
7.37.2.2 computeTimeValueDifference	194
7.37.2.3 evaluate	195
7.37.2.4 evaluate	195
7.38 multiscale::verification::ChangeMeasureGrammar< Iterator > Class Template Reference	196
7.38.1 Detailed Description	199
7.38.2 Constructor & Destructor Documentation	199
7.38.2.1 ChangeMeasureGrammar	199
7.38.3 Member Function Documentation	199
7.38.3.1 assignNamesToRules	199
7.38.3.2 initialise	199
7.38.3.3 initialiseDebugSupport	199
7.38.3.4 initialiseGrammar	199
7.38.3.5 initialiseRulesDebugging	200
7.38.4 Member Data Documentation	200
7.38.4.1 changeMeasureRule	200
7.38.4.2 changeMeasureTypeParser	200
7.39 multiscale::verification::ChangeMeasureTypeParser Struct Reference	200
7.39.1 Detailed Description	202
7.39.2 Constructor & Destructor Documentation	202
7.39.2.1 ChangeMeasureTypeParser	202
7.40 multiscale::verification::ChangeTemporalNumericCollectionAttribute Class Reference	202
7.40.1 Detailed Description	203
7.40.2 Member Data Documentation	203
7.40.2.1 changeMeasure	203
7.40.2.2 temporalNumericCollection	203
7.41 multiscale::verification::ChangeTemporalNumericMeasureAttribute Class Reference	203
7.41.1 Detailed Description	204
7.41.2 Member Data Documentation	204
7.41.2.1 changeMeasure	204
7.41.2.2 comparator	204
7.41.2.3 lhsTemporalNumericMeasure	205
7.41.2.4 rhsTemporalNumericMeasure	205
7.42 multiscale::analysis::CircularityMeasure Class Reference	205
7.42.1 Detailed Description	206
7.42.2 Member Function Documentation	206
7.42.2.1 compute	206

7.42.2.2	compute	206
7.43	multiscale::analysis::CircularMatFactory Class Reference	206
7.43.1	Detailed Description	209
7.43.2	Constructor & Destructor Documentation	209
7.43.2.1	CircularMatFactory	209
7.43.2.2	\sim CircularMatFactory	209
7.43.3	Member Function Documentation	209
7.43.3.1	createCircularMask	209
7.43.3.2	createFromViewerImage	210
7.43.3.3	isValidViewerImage	210
7.43.3.4	maxColourBarIntensityFromViewerImage	210
7.43.3.5	processConcentrations	211
7.43.4	Member Data Documentation	211
7.43.4.1	COLOURBAR_MAX_X	211
7.43.4.2	COLOURBAR_MAX_Y	211
7.43.4.3	ERR_UNIMPLEMENTED_METHOD	211
7.43.4.4	INPUT_IMG_HEIGHT	211
7.43.4.5	INPUT_IMG_WIDTH	211
7.43.4.6	INTENSITY_MAX	212
7.43.4.7	ROI_RADIUS	212
7.43.4.8	ROI_START_X	212
7.43.4.9	ROI_START_Y	212
7.44	multiscale::verification::Cluster Class Reference	212
7.44.1	Detailed Description	214
7.45	multiscale::analysis::Cluster Class Reference	215
7.45.1	Detailed Description	219
7.45.2	Constructor & Destructor Documentation	219
7.45.2.1	Cluster	219
7.45.2.2	\sim Cluster	219
7.45.3	Member Function Documentation	219
7.45.3.1	addEntity	219
7.45.3.2	areValidOriginDependentValues	219
7.45.3.3	getEntities	220
7.45.3.4	getEntitiesCentrePoints	220
7.45.3.5	getEntitiesContourPoints	220
7.45.3.6	getEntitiesConvexHull	220
7.45.3.7	getMinAreaEnclosingCircleCentre	220
7.45.3.8	getMinAreaEnclosingCircleRadius	220
7.45.3.9	getMinAreaEnclosingRect	221
7.45.3.10	getMinAreaEnclosingTriangle	221

7.45.3.11 initialise	221
7.45.3.12 isCircularMeasure	221
7.45.3.13 isRectangularMeasure	221
7.45.3.14 isTriangularMeasure	221
7.45.3.15 setOriginDependentMembers	222
7.45.3.16 type	222
7.45.3.17 updateArea	222
7.45.3.18 updateCentrePoint	222
7.45.3.19 updateClusterednessDegree	222
7.45.3.20 updateDensity	223
7.45.3.21 updatePerimeter	223
7.45.3.22 validateOriginDependentValues	223
7.45.4 Member Data Documentation	223
7.45.4.1 entities	223
7.45.4.2 ERR_ORIGIN_DEPENDENT_VALUES	223
7.45.4.3 ERR_UNDEFINED_SHAPE	223
7.45.4.4 minAreaEnclosingCircleCentre	224
7.45.4.5 minAreaEnclosingCircleRadius	224
7.45.4.6 minAreaEnclosingRect	224
7.45.4.7 minAreaEnclosingTriangle	224
7.46 multiscale::analysis::ClusterDetector Class Reference	224
7.46.1 Detailed Description	228
7.46.2 Constructor & Destructor Documentation	228
7.46.2.1 ClusterDetector	228
7.46.2.2 ~ClusterDetector	229
7.46.3 Member Function Documentation	229
7.46.3.1 addEntitiesToClusters	229
7.46.3.2 analyseClusters	229
7.46.3.3 analyseClustersOriginDependentValues	229
7.46.3.4 clearPreviousDetectionResults	230
7.46.3.5 computeAveragePileUpDegree	230
7.46.3.6 computeClusterednessIndex	230
7.46.3.7 convertEntities	230
7.46.3.8 convertEpsValue	230
7.46.3.9 convertNonPiledUpEntities	231
7.46.3.10 convertPiledUpEntities	231
7.46.3.11 createDetectorSpecificTrackbars	231
7.46.3.12 detectAndAnalyseClusters	231
7.46.3.13 detectClusters	232
7.46.3.14 detectEntitiesInImage	232

7.46.3.15 getClusterConvexHull	232
7.46.3.16 getClusters	232
7.46.3.17 getCollectionOfSpatialEntityPseudo3D	232
7.46.3.18 getDetectorTypeAsString	233
7.46.3.19 getEps	233
7.46.3.20 getMinPoints	233
7.46.3.21 getValidMinPointsValue	233
7.46.3.22 initialiseDetectorSpecificFields	233
7.46.3.23 processImageAndDetect	233
7.46.3.24 setEps	234
7.46.3.25 setMinPoints	234
7.46.3.26 updateClusterOriginDependentValues	234
7.46.4 Member Data Documentation	234
7.46.4.1 clusters	234
7.46.4.2 DETECTOR_TYPE	235
7.46.4.3 entityPileupDegree	235
7.46.4.4 eps	235
7.46.4.5 EPS_MAX	235
7.46.4.6 EPS_MIN	235
7.46.4.7 EPS_REAL_MAX	235
7.46.4.8 EPS_REAL_MIN	235
7.46.4.9 MIN_POINTS_MAX	235
7.46.4.10 MIN_POINTS_MIN	235
7.46.4.11 minPoints	236
7.46.4.12 TRACKBAR_EPS	236
7.46.4.13 TRACKBAR_MINPOINTS	236
7.47 multiscale::verification::CommandLineModelChecking Class Reference	236
7.47.1 Detailed Description	243
7.47.2 Constructor & Destructor Documentation	243
7.47.2.1 CommandLineModelChecking	243
7.47.2.2 ~CommandLineModelChecking	243
7.47.3 Member Function Documentation	243
7.47.3.1 areApproximateBayesianModelCheckingArgumentsPresent	243
7.47.3.2 areApproximateProbabilisticModelCheckingArgumentsPresent	243
7.47.3.3 areBayesianModelCheckingArgumentsPresent	244
7.47.3.4 areInvalidExecutionArguments	245
7.47.3.5 areInvalidModelCheckingArguments	245
7.47.3.6 areInvalidModelCheckingArgumentsPresent	245
7.47.3.7 areInvalidModelCheckingTypeSpecificArguments	245
7.47.3.8 areModelCheckingTypeSpecificArgumentsPresent	246

7.47.3.9	areStatisticalModelCheckingArgumentsPresent	246
7.47.3.10	areUnrecognizedArgumentsPresent	246
7.47.3.11	isValidArguments	247
7.47.3.12	isValidArgumentsConsideringConfiguration	248
7.47.3.13	execute	248
7.47.3.14	handleHelpRequest	248
7.47.3.15	initialise	248
7.47.3.16	initialiseAllowedArgumentsConfiguration	249
7.47.3.17	initialiseApproximateBayesianModelChecker	249
7.47.3.18	initialiseApproximateBayesianModelCheckerArgumentsConfiguration	249
7.47.3.19	initialiseApproximateProbabilisticModelChecker	249
7.47.3.20	initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration	249
7.47.3.21	initialiseBayesianModelChecker	250
7.47.3.22	initialiseBayesianModelCheckerArgumentsConfiguration	250
7.47.3.23	initialiseClassMembers	250
7.47.3.24	initialiseModelChecker	250
7.47.3.25	initialiseModelCheckerTypeDependentClassMembers	251
7.47.3.26	initialiseModelCheckerTypeSpecificArgumentsConfiguration	251
7.47.3.27	initialiseModelCheckingManager	251
7.47.3.28	initialiseOptionalArgumentsConfiguration	251
7.47.3.29	initialiseOptionalArgumentsDependentClassMembers	251
7.47.3.30	initialiseProbabilisticBlackBoxModelChecker	251
7.47.3.31	initialiseRequiredArgumentsConfiguration	252
7.47.3.32	initialiseRequiredArgumentsDependentClassMembers	252
7.47.3.33	initialiseStatisticalModelChecker	252
7.47.3.34	initialiseStatisticalModelCheckerArgumentsConfiguration	252
7.47.3.35	isHelpArgumentPresent	253
7.47.3.36	parseAndStoreArgumentsValues	253
7.47.3.37	printHelpClosingMessage	253
7.47.3.38	printHelpContentsMessage	253
7.47.3.39	printHelpIntroMessage	253
7.47.3.40	printHelpMessage	253
7.47.3.41	printModelCheckingInitialisationMessage	254
7.47.3.42	removeApproximateBayesianModelCheckingArguments	254
7.47.3.43	removeApproximateProbabilisticModelCheckingArguments	254
7.47.3.44	removeBayesianModelCheckingArguments	254
7.47.3.45	removeModelCheckingTypeSpecificArguments	255
7.47.3.46	removeOptionalArguments	256
7.47.3.47	removeRequiredArguments	256
7.47.3.48	removeStatisticalModelCheckingArguments	256

7.47.4 Member Data Documentation	257
7.47.4.1 allowedArguments	257
7.47.4.2 ARG_APPROXIMATE_BAYESIAN_ALPHA_DESCRIPTION	257
7.47.4.3 ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG	257
7.47.4.4 ARG_APPROXIMATE_BAYESIAN_BETA_DESCRIPTION	257
7.47.4.5 ARG_APPROXIMATE_BAYESIAN_BETA_NAME_LONG	257
7.47.4.6 ARG_BAYES_FACTOR_THRESHOLD_DESCRIPTION	257
7.47.4.7 ARG_BAYES_FACTOR_THRESHOLD_NAME_LONG	257
7.47.4.8 ARG_BAYESIAN_ALPHA_DESCRIPTION	258
7.47.4.9 ARG_BAYESIAN_ALPHA_NAME_LONG	258
7.47.4.10 ARG_BAYESIAN_BETA_DESCRIPTION	258
7.47.4.11 ARG_BAYESIAN_BETA_NAME_LONG	258
7.47.4.12 ARG_DELTA_DESCRIPTION	258
7.47.4.13 ARG_DELTA_NAME_LONG	258
7.47.4.14 ARG_EPSILON_DESCRIPTION	258
7.47.4.15 ARG_EPSILON_NAME_LONG	259
7.47.4.16 ARG_EXTRA_EVALUATION_PROGRAM_DESCRIPTION	259
7.47.4.17 ARG_EXTRA_EVALUATION_PROGRAM_NAME_BOTH	259
7.47.4.18 ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG	259
7.47.4.19 ARG_EXTRA_EVALUATION_TIME_DESCRIPTION	259
7.47.4.20 ARG_EXTRA_EVALUATION_TIME_NAME_BOTH	259
7.47.4.21 ARG_EXTRA_EVALUATION_TIME_NAME_LONG	259
7.47.4.22 ARG_HELP_DESCRIPTION	259
7.47.4.23 ARG_HELP_NAME_BOTH	260
7.47.4.24 ARG_HELP_NAME_LONG	260
7.47.4.25 ARG_LOGIC_QUERIES_DESCRIPTION	260
7.47.4.26 ARG_LOGIC_QUERIES_NAME_BOTH	260
7.47.4.27 ARG_LOGIC_QUERIES_NAME_LONG	260
7.47.4.28 ARG_MODEL_CHECKER_TYPE_DESCRIPTION	260
7.47.4.29 ARG_MODEL_CHECKER_TYPE_NAME_BOTH	260
7.47.4.30 ARG_MODEL_CHECKER_TYPE_NAME_LONG	260
7.47.4.31 ARG_SPATIAL_TEMPORAL_TRACES_DESCRIPTION	261
7.47.4.32 ARG_SPATIAL_TEMPORAL_TRACES_NAME_BOTH	261
7.47.4.33 ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG	261
7.47.4.34 ARG_TYPE_I_ERROR_DESCRIPTION	261
7.47.4.35 ARG_TYPE_I_ERROR_NAME_LONG	261
7.47.4.36 ARG_TYPE_II_ERROR_DESCRIPTION	261
7.47.4.37 ARG_TYPE_II_ERROR_NAME_LONG	261
7.47.4.38 ARG_TYPE_SEMATICS_TABLE_DESCRIPTION	262
7.47.4.39 ARG_TYPE_SEMATICS_TABLE_NAME_BOTH	262

7.47.4.40 ARG_TYPE_SEMANTICS_TABLE_NAME_LONG	262
7.47.4.41 ARG_VARIANCE_THRESHOLD_DESCRIPTION	262
7.47.4.42 ARG_VARIANCE_THRESHOLD_NAME_LONG	262
7.47.4.43 ARG_VERBOSE_DESCRIPTION	262
7.47.4.44 ARG_VERBOSE_NAME_BOTH	262
7.47.4.45 ARG_VERBOSE_NAME_LONG	262
7.47.4.46 CONFIG_CAPTION_ALLOWED_ARGUMENTS	263
7.47.4.47 CONFIG_CAPTION_APPROXIMATE_BAYESIAN_MODEL_CHECKER_ARGUMENTS	263
7.47.4.48 CONFIG_CAPTION_APPROXIMATE_PROBABILISTIC_MODEL_CHECKER_ARGUMENTS	263
7.47.4.49 CONFIG_CAPTION_BAYESIAN_MODEL_CHECKER_ARGUMENTS	263
7.47.4.50 CONFIG_CAPTION_MODEL_CHECKER_TYPE_SPECIFIC_ARGUMENTS	263
7.47.4.51 CONFIG_CAPTION_OPTIONAL_ARGUMENTS	263
7.47.4.52 CONFIG_CAPTION_PROBABILISTIC_BLACK_BOX_MODEL_CHECKER_ARGUMENTS	263
7.47.4.53 CONFIG_CAPTION_REQUIRED_ARGUMENTS	263
7.47.4.54 CONFIG_CAPTION_STATISTICAL_MODEL_CHECKER_ARGUMENTS	264
7.47.4.55 ERR_INVALID_COMMAND_LINE_ARGUMENTS	264
7.47.4.56 ERR_INVALID_MODEL_CHECKING_ARGUMENTS	264
7.47.4.57 ERR_INVALID_MODEL_CHECKING_TYPE	264
7.47.4.58 extraEvaluationProgramPath	264
7.47.4.59 extraEvaluationTime	264
7.47.4.60 HELP_AUTHOR_LABEL	264
7.47.4.61 HELP_AUTHOR_MSG	265
7.47.4.62 HELP_COPYRIGHT_LABEL	265
7.47.4.63 HELP_COPYRIGHT_MSG	265
7.47.4.64 HELP_DESCRIPTION_LABEL	265
7.47.4.65 HELP_DESCRIPTION_MSG	265
7.47.4.66 HELP_NAME_LABEL	265
7.47.4.67 HELP_NAME_MSG	265
7.47.4.68 HELP_REPORTING_BUGS_LABEL	265
7.47.4.69 HELP_REPORTING_BUGS_MSG	266
7.47.4.70 HELP_USAGE_LABEL	266
7.47.4.71 HELP_USAGE_MSG	266
7.47.4.72 logicQueriesFilepath	266
7.47.4.73 MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME	266
7.47.4.74 MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_BEGIN	266
7.47.4.75 MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_END	266
7.47.4.76 MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_MIDDLE1	266
7.47.4.77 MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_MIDDLE2	267

7.47.4.78 MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME	267
7.47.4.79 MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_BEGIN	267
7.47.4.80 MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_END	267
7.47.4.81 MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_MIDDLE	267
7.47.4.82 MODEL_CHECKER_BAYESIAN_NAME	267
7.47.4.83 MODEL_CHECKER_BAYESIAN_PARAMETERS_BEGIN	267
7.47.4.84 MODEL_CHECKER_BAYESIAN_PARAMETERS_END	268
7.47.4.85 MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE1	268
7.47.4.86 MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE2	268
7.47.4.87 MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME	268
7.47.4.88 MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_PARAMETERS	268
7.47.4.89 MODEL_CHECKER_STATISTICAL_NAME	268
7.47.4.90 MODEL_CHECKER_STATISTICAL_PARAMETERS_BEGIN	268
7.47.4.91 MODEL_CHECKER_STATISTICAL_PARAMETERS_END	268
7.47.4.92 MODEL_CHECKER_STATISTICAL_PARAMETERS_MIDDLE	269
7.47.4.93 MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN	269
7.47.4.94 MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC	269
7.47.4.95 MODEL_CHECKER_TYPE_BAYESIAN	269
7.47.4.96 MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX	269
7.47.4.97 MODEL_CHECKER_TYPE_STATISTICAL	269
7.47.4.98 modelCheckerFactory	269
7.47.4.99 modelCheckerParameters	270
7.47.4.100modelCheckerType	270
7.47.4.101modelCheckerTypeName	270
7.47.4.102modelCheckerTypeSpecificArguments	270
7.47.4.103modelCheckingManager	270
7.47.4.104MSG_MODEL_CHECKING_HELP_REQUESTED	270
7.47.4.105optionalArguments	270
7.47.4.106requiredArguments	271
7.47.4.107shouldVerboseDetailedResults	271
7.47.4.108racesFolderPath	271
7.47.4.109typeSemanticsTableFilepath	271
7.47.4.110variablesMap	271
7.48 multiscale::verification::ComparatorAttribute Class Reference	272
7.48.1 Detailed Description	272
7.48.2 Member Data Documentation	272
7.48.2.1 comparatorType	272
7.49 multiscale::verification::ComparatorEvaluator Class Reference	272

7.49.1	Detailed Description	273
7.49.2	Member Function Documentation	273
7.49.2.1	evaluate	273
7.50	multiscale::verification::ComparatorGrammar< Iterator > Class Template Reference	274
7.50.1	Detailed Description	276
7.50.2	Constructor & Destructor Documentation	276
7.50.2.1	ComparatorGrammar	276
7.50.3	Member Function Documentation	276
7.50.3.1	assignNamesToRules	276
7.50.3.2	initialise	276
7.50.3.3	initialiseDebugSupport	276
7.50.3.4	initialiseGrammar	276
7.50.3.5	initialiseRulesDebugging	277
7.50.4	Member Data Documentation	277
7.50.4.1	comparatorRule	277
7.50.4.2	comparatorTypeParser	277
7.51	multiscale::verification::ComparatorNonEqualTypeParser Struct Reference	277
7.51.1	Detailed Description	279
7.51.2	Constructor & Destructor Documentation	279
7.51.2.1	ComparatorNonEqualTypeParser	279
7.52	multiscale::verification::ComparatorTypeParser Struct Reference	279
7.52.1	Detailed Description	280
7.52.2	Constructor & Destructor Documentation	280
7.52.2.1	ComparatorTypeParser	280
7.53	multiscaletest::CompleteTraceTest Class Reference	280
7.53.1	Detailed Description	283
7.53.2	Member Function Documentation	283
7.53.2.1	InitialiseTrace	283
7.53.3	Member Data Documentation	283
7.53.3.1	clustersClusterednessMaxValue	283
7.53.3.2	clustersClusterednessMinValue	283
7.54	multiscale::ConsolePrinter Class Reference	283
7.54.1	Detailed Description	285
7.54.2	Member Function Documentation	286
7.54.2.1	getUnixColourCode	286
7.54.2.2	isStdOutTerminalWhichSupportsColour	287
7.54.2.3	printColouredMessage	287
7.54.2.4	printColouredMessageWithColouredTag	287
7.54.2.5	printEmptyLine	288
7.54.2.6	printMessage	288

7.54.2.7	printMessageUsingColour	288
7.54.2.8	printMessageWithColouredTag	289
7.54.2.9	printNewLine	290
7.54.2.10	printNonColouredMessage	290
7.54.2.11	printWarningMessage	290
7.54.2.12	terminalSupportsColour	291
7.54.2.13	terminalSupportsColour	291
7.54.2.14	unixColourCodeToString	291
7.54.3	Member Data Documentation	291
7.54.3.1	CSI_COLOUR_CODE_END_TAG	291
7.54.3.2	CSI_COLOUR_START_VALUE	291
7.54.3.3	CSI_RESET_CODE	292
7.54.3.4	CSI_SEPARATOR	292
7.54.3.5	CSI_START_TAG	292
7.54.3.6	ERR_INVALID_COLOUR_CODE	292
7.54.3.7	SEPARATOR	292
7.54.3.8	TERM_ENV_VARIABLE	292
7.54.3.9	WARNING_TAG	292
7.55	multiscale::verification::ConstraintAttribute Class Reference	292
7.55.1	Detailed Description	293
7.55.2	Member Data Documentation	294
7.55.2.1	firstConstraint	294
7.55.2.2	nextConstraints	294
7.56	multiscale::verification::ConstraintVisitor Class Reference	294
7.56.1	Detailed Description	298
7.56.2	Constructor & Destructor Documentation	298
7.56.2.1	ConstraintVisitor	298
7.56.3	Member Function Documentation	298
7.56.3.1	evaluate	298
7.56.3.2	evaluate	298
7.56.3.3	evaluateFilterNumericMeasure	299
7.56.3.4	evaluateNextConstraints	299
7.56.3.5	evaluateNumericMeasure	299
7.56.3.6	evaluateSpatialMeasureConstraint	299
7.56.3.7	evaluateTypeConstraint	301
7.56.3.8	evaluateUnarySpatialConstraint	301
7.56.3.9	evaluateUnaryTypeConstraint	302
7.56.3.10	filterSpatialEntitiesWrtSpatialMeasure	302
7.56.3.11	filterSpatialEntitiesWrtType	302
7.56.3.12	filterSpatialEntitiesWrtTypeConsideringEqualComparator	303

7.56.3.13 filterSpatialEntitiesWrtTypeConsideringNonEqualComparator	303
7.56.3.14 operator()	303
7.56.3.15 operator()	304
7.56.3.16 operator()	305
7.56.3.17 operator()	305
7.56.3.18 operator()	305
7.56.3.19 operator()	305
7.56.3.20 operator()	306
7.56.3.21 operator()	306
7.56.3.22 operator()	306
7.56.3.23 operator()	306
7.56.3.24 translateSemanticTypeToAbstractNaturalNumber	307
7.56.4 Member Data Documentation	307
7.56.4.1 constraintTimePoint	307
7.56.4.2 initialTimePoint	307
7.56.4.3 typeSemanticsTable	307
7.57 multiscale::analysis::DataPoint Class Reference	307
7.57.1 Detailed Description	309
7.57.2 Constructor & Destructor Documentation	309
7.57.2.1 ~DataPoint	309
7.57.3 Member Function Documentation	309
7.57.3.1 distanceTo	309
7.58 multiscale::analysis::DBSCAN Class Reference	310
7.58.1 Detailed Description	311
7.58.2 Constructor & Destructor Documentation	312
7.58.2.1 DBSCAN	312
7.58.2.2 ~DBSCAN	312
7.58.3 Member Function Documentation	312
7.58.3.1 addUnclassifiedNodesToSeedsList	312
7.58.3.2 allocateDistanceMatrix	312
7.58.3.3 assignBorderNodesToClusters	312
7.58.3.4 constructDistanceMatrix	312
7.58.3.5 expandCoreCluster	313
7.58.3.6 findClosestCoreDataPoint	313
7.58.3.7 labelUnclassifiedAndNoiseAsBorder	313
7.58.3.8 retrieveNeighbours	314
7.58.3.9 run	315
7.58.3.10 runAlgorithm	315
7.58.4 Member Data Documentation	315
7.58.4.1 CLUSTERING_BORDER	315

7.58.4.2	CLUSTERING_NOISE	316
7.58.4.3	CLUSTERING_UNCLASSIFIED	316
7.58.4.4	distanceMatrix	316
7.58.4.5	eps	316
7.58.4.6	minPoints	316
7.58.4.7	nrOfDataPoints	316
7.59	multiscale::analysis::Detector Class Reference	316
7.59.1	Detailed Description	321
7.59.2	Constructor & Destructor Documentation	321
7.59.2.1	Detector	321
7.59.2.2	~Detector	322
7.59.3	Member Function Documentation	322
7.59.3.1	addAverageMeasuresToPropertyTree	322
7.59.3.2	addNumericStateVariableToPropertyTree	322
7.59.3.3	addSpatialEntitiesToPropertyTree	322
7.59.3.4	addSpatialEntityPropertiesToTree	322
7.59.3.5	addSpatialEntityTypeToPropertyTree	323
7.59.3.6	clearPreviousDetectionResults	323
7.59.3.7	constructSpatialEntityPropertyTree	323
7.59.3.8	createDetectorSpecificTrackbars	324
7.59.3.9	createTrackbars	324
7.59.3.10	createTrackbarsWindow	324
7.59.3.11	detect	324
7.59.3.12	detect	324
7.59.3.13	detectInDebugMode	324
7.59.3.14	detectInReleaseMode	325
7.59.3.15	displayImage	325
7.59.3.16	displayResultsInWindow	325
7.59.3.17	findGoodIntersectionPoints	325
7.59.3.18	findGoodPointsForAngle	325
7.59.3.19	getCollectionOfSpatialEntityPseudo3D	326
7.59.3.20	getDetectorTypeAsString	326
7.59.3.21	initialise	326
7.59.3.22	initialiseDetectorSpecificFields	326
7.59.3.23	initialiseDetectorSpecificFieldsIfNotSet	326
7.59.3.24	initialiseDetectorSpecificImageDependentFields	327
7.59.3.25	initialiseImageDependentFields	327
7.59.3.26	initialiseImageOrigin	327
7.59.3.27	isValidInputImage	327
7.59.3.28	minAreaRectCentre	327

7.59.3.29 outputAveragedMeasuresToCsvFile	327
7.59.3.30 outputResults	328
7.59.3.31 outputResultsToCsvFile	328
7.59.3.32 outputResultsToCsvFile	328
7.59.3.33 outputResultsToFile	328
7.59.3.34 outputResultsToImage	328
7.59.3.35 outputResultsToXMLFile	329
7.59.3.36 outputResultsToXMLFile	329
7.59.3.37 outputSpatialEntitiesToCsvFile	329
7.59.3.38 polygonAngle	329
7.59.3.39 polygonAngle	330
7.59.3.40 printOutputErrorMessage	330
7.59.3.41 processImageAndDetect	330
7.59.3.42 processPressedKeyRequest	330
7.59.3.43 setDetectorSpecificFieldsInitialisationFlag	330
7.59.3.44 storeOutputImageOnDisk	331
7.59.4 Member Data Documentation	331
7.59.4.1 avgClusterednessDegree	331
7.59.4.2 avgDensity	331
7.59.4.3 CSV_EXTENSION	331
7.59.4.4 debugMode	331
7.59.4.5 detectMethodCalled	331
7.59.4.6 detectorSpecificFieldsInitialised	332
7.59.4.7 ERR_INVALID_IMAGE	332
7.59.4.8 ERR_OUTPUT_FILE	332
7.59.4.9 ERR_OUTPUT_WITHOUT_DETECT	332
7.59.4.10 image	332
7.59.4.11 IMG_EXTENSION	332
7.59.4.12 KEY_ESC	332
7.59.4.13 KEY_SAVE	333
7.59.4.14 LABEL_ATTRIBUTE	333
7.59.4.15 LABEL_AVG_CLUSTEREDNESS	333
7.59.4.16 LABEL_AVG_DENSITY	333
7.59.4.17 LABEL_COMMENT	333
7.59.4.18 LABEL_COMMENT_CONTENTS	333
7.59.4.19 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE	333
7.59.4.20 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_NAME	333
7.59.4.21 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_VALUE	333
7.59.4.22 LABEL_EXPERIMENT_TIMEPOINT_SPATIAL_ENTITY	334
7.59.4.23 LABEL_SPATIAL_ENTITY_ANGLE	334

7.59.4.24 LABEL_SPATIAL_ENTITY_AREA	334
7.59.4.25 LABEL_SPATIAL_ENTITY_CENTROID_X	334
7.59.4.26 LABEL_SPATIAL_ENTITY_CENTROID_Y	334
7.59.4.27 LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE	334
7.59.4.28 LABEL_SPATIAL_ENTITY_CLUSTEREDNESS	334
7.59.4.29 LABEL_SPATIAL_ENTITY_DENSITY	334
7.59.4.30 LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN	334
7.59.4.31 LABEL_SPATIAL_ENTITY_PERIMETER	335
7.59.4.32 LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE	335
7.59.4.33 LABEL_SPATIAL_ENTITY_SPATIAL_TYPE	335
7.59.4.34 LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE	335
7.59.4.35 origin	335
7.59.4.36 OUTPUT_CLUSTEREDNESS	335
7.59.4.37 OUTPUT_DENSITY	335
7.59.4.38 outputFilepath	335
7.59.4.39 outputImage	336
7.59.4.40 WIN_OUTPUT_IMAGE	336
7.59.4.41 XML_EXTENSION	336
7.60 multiscale::Distribution Class Reference	336
7.60.1 Detailed Description	338
7.60.2 Member Function Documentation	338
7.60.2.1 validateProbability	339
7.60.3 Member Data Documentation	340
7.60.3.1 ERR_PROBABILITY_VALUE_BEGIN	340
7.60.3.2 ERR_PROBABILITY_VALUE_END	340
7.61 multiscale::DivisionOperation Class Reference	340
7.61.1 Detailed Description	341
7.61.2 Member Function Documentation	341
7.61.2.1 operator()	341
7.62 multiscaletest::EmptyTraceTest Class Reference	341
7.62.1 Detailed Description	344
7.62.2 Member Function Documentation	344
7.62.2.1 InitialiseTrace	344
7.63 multiscale::analysis::Entity Class Reference	344
7.63.1 Detailed Description	347
7.63.2 Constructor & Destructor Documentation	347
7.63.2.1 Entity	347
7.63.2.2 Entity	348
7.63.2.3 ~Entity	348
7.63.3 Member Function Documentation	348

7.63.3.1	isValid	348
7.63.3.2	distanceTo	348
7.63.3.3	distanceTo	348
7.63.3.4	getArea	348
7.63.3.5	getCentre	348
7.63.3.6	getContourPoints	349
7.63.3.7	getPerimeter	349
7.63.3.8	getPileUpDegree	349
7.63.3.9	toString	349
7.63.3.10	validateInputValues	349
7.63.4	Member Data Documentation	349
7.63.4.1	area	349
7.63.4.2	centre	350
7.63.4.3	contourPoints	350
7.63.4.4	ERR_DISTANCE	350
7.63.4.5	ERR_INPUT	350
7.63.4.6	OUTPUT_SEPARATOR	350
7.63.4.7	perimeter	350
7.63.4.8	pileUpDegree	350
7.64	multiscale::verification::EquivalenceConstraintAttribute Class Reference	351
7.64.1	Detailed Description	351
7.64.2	Member Data Documentation	351
7.64.2.1	constraint	351
7.65	multiscale::verification::EquivalenceLogicPropertyAttribute Class Reference	351
7.65.1	Detailed Description	352
7.65.2	Member Data Documentation	352
7.65.2.1	logicProperty	352
7.66	EuclideanDataPoint Class Reference	353
7.66.1	Detailed Description	354
7.66.2	Constructor & Destructor Documentation	354
7.66.2.1	EuclideanDataPoint	354
7.66.2.2	EuclideanDataPoint	355
7.66.2.3	~EuclideanDataPoint	355
7.66.3	Member Function Documentation	355
7.66.3.1	distanceTo	355
7.66.4	Member Data Documentation	355
7.66.4.1	x	355
7.66.4.2	y	355
7.67	multiscale::ExceptionHandler Class Reference	355
7.67.1	Detailed Description	356

7.67.2 Member Function Documentation	356
7.67.2.1 printDetailedErrorMessage	356
7.67.2.2 printRawErrorMessage	356
7.68 multiscale::FileOpenException Class Reference	357
7.68.1 Detailed Description	360
7.68.2 Constructor & Destructor Documentation	360
7.68.2.1 FileOpenException	360
7.68.2.2 FileOpenException	360
7.68.2.3 FileOpenException	360
7.69 multiscale::Filesystem Class Reference	360
7.69.1 Detailed Description	362
7.69.2 Member Function Documentation	362
7.69.2.1 getFilesInFolder	362
7.69.2.2 isValidFilePath	362
7.69.2.3 isValidFolderPath	362
7.69.2.4 isValidFilePath	363
7.69.2.5 isValidFolderPath	363
7.69.2.6 nativeFormatFilePath	363
7.69.3 Member Data Documentation	363
7.69.3.1 ERR_INVALID_PATH_BEGIN	363
7.69.3.2 ERR_INVALID_PATH_END	363
7.70 multiscale::verification::FilterNumericMeasureAttribute Class Reference	364
7.70.1 Detailed Description	364
7.70.2 Member Data Documentation	364
7.70.2.1 filterNumericMeasure	364
7.71 multiscale::verification::FilterNumericVisitor Class Reference	365
7.71.1 Detailed Description	367
7.71.2 Constructor & Destructor Documentation	367
7.71.2.1 FilterNumericVisitor	367
7.71.3 Member Function Documentation	367
7.71.3.1 evaluate	367
7.71.3.2 evaluate	367
7.71.3.3 operator()	368
7.71.3.4 operator()	368
7.71.3.5 operator()	368
7.71.3.6 operator()	368
7.71.3.7 operator()	369
7.71.4 Member Data Documentation	369
7.71.4.1 spatialEntity	369
7.71.4.2 timePoint	369

7.71.4.3	typeSemanticsTable	369
7.72	multiscale::verification::FilterSubsetAttribute Class Reference	369
7.72.1	Detailed Description	370
7.72.2	Member Data Documentation	370
7.72.2.1	constraint	370
7.72.2.2	subsetSpecific	370
7.73	multiscale::verification::FutureLogicPropertyAttribute Class Reference	371
7.73.1	Detailed Description	371
7.73.2	Member Data Documentation	371
7.73.2.1	endTimepoint	371
7.73.2.2	logicProperty	371
7.73.2.3	startTimepoint	372
7.74	multiscale::Geometry2D Class Reference	372
7.74.1	Detailed Description	375
7.74.2	Member Function Documentation	375
7.74.2.1	angleBtwPoints	375
7.74.2.2	angleOfLineWrtOxAxis	376
7.74.2.3	areaOfTriangle	376
7.74.2.4	areCollinear	376
7.74.2.5	areEqualPoints	377
7.74.2.6	areIdenticalLines	378
7.74.2.7	areIdenticalLines	378
7.74.2.8	areOnTheSameSideOfLine	378
7.74.2.9	distanceBtwPoints	379
7.74.2.10	distanceBtwPoints	379
7.74.2.11	distanceFromPointToLine	379
7.74.2.12	findPointsOnEdge	380
7.74.2.13	inverseTranslate	380
7.74.2.14	isAngleBetween	380
7.74.2.15	isAngleBetweenNonReflex	380
7.74.2.16	isBetweenCoordinates	381
7.74.2.17	isOppositeAngleBetween	381
7.74.2.18	isOppositeAngleBetweenNonReflex	381
7.74.2.19	isPointOnEdge	382
7.74.2.20	isPointOnLineSegment	382
7.74.2.21	lineCircleIntersection	382
7.74.2.22	lineCircleOneIntersectionPoint	383
7.74.2.23	lineCircleTwoIntersectionPoints	383
7.74.2.24	lineEquationDeterminedByPoints	383
7.74.2.25	lineIntersection	384

7.74.2.26 lineIntersection	384
7.74.2.27 lineIntersection	385
7.74.2.28 lineSegmentCircleIntersection	385
7.74.2.29 lineSegmentIntersection	386
7.74.2.30 middlePoint	386
7.74.2.31 minimumDistancePointIndex	386
7.74.2.32 oppositeAngle	387
7.74.2.33 orthogonalLineToAnotherLineEdgePoints	387
7.74.2.34 slopeOfLine	387
7.74.2.35 translate	387
7.74.3 Member Data Documentation	388
7.74.3.1 MATRIX_START_INDEX	388
7.74.3.2 PI	388
7.75 multiscale::verification::GlobalLogicPropertyAttribute Class Reference	388
7.75.1 Detailed Description	388
7.75.2 Member Data Documentation	389
7.75.2.1 endTimepoint	389
7.75.2.2 logicProperty	389
7.75.2.3 startTimepoint	389
7.76 multiscale::verification::HeterogeneousTimeseriesComponentAttribute Class Reference	389
7.76.1 Detailed Description	390
7.76.2 Member Data Documentation	390
7.76.2.1 heterogeneousTimeseriesComponent	390
7.77 multiscale::verification::HeterogeneousTimeseriesComponentTypeParser Struct Reference	390
7.77.1 Detailed Description	392
7.77.2 Constructor & Destructor Documentation	392
7.77.2.1 HeterogeneousTimeseriesComponentTypeParser	392
7.78 multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation > Class Template Reference	393
7.78.1 Detailed Description	393
7.78.2 Member Function Documentation	393
7.78.2.1 computeEndIndex	394
7.78.2.2 hasValidSuccessor	394
7.78.2.3 isstartIndex	394
7.79 multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute Class Reference	395
7.79.1 Detailed Description	395
7.79.2 Member Data Documentation	395
7.79.2.1 homogeneousTimeseriesComponent	396
7.79.2.2 homogeneousTimeseriesMeasure	396
7.79.2.3 temporalNumericMeasureCollection	396

7.80 multiscale::verification::HomogeneousTimeseriesComponentAttribute Class Reference	396
7.80.1 Detailed Description	397
7.80.2 Member Data Documentation	397
7.80.2.1 homogeneousTimeseriesComponent	397
7.81 multiscale::verification::HomogeneousTimeseriesComponentTypeParser Struct Reference	397
7.81.1 Detailed Description	399
7.81.2 Constructor & Destructor Documentation	399
7.81.2.1 HomogeneousTimeseriesComponentTypeParser	399
7.82 multiscale::verification::HomogeneousTimeseriesMeasureAttribute Class Reference	400
7.82.1 Detailed Description	400
7.82.2 Member Data Documentation	400
7.82.2.1 homogeneousTimeseriesMeasure	400
7.83 multiscale::verification::HomogeneousTimeseriesMeasureTypeParser Struct Reference	400
7.83.1 Detailed Description	402
7.83.2 Constructor & Destructor Documentation	402
7.83.2.1 HomogeneousTimeseriesMeasureTypeParser	402
7.84 multiscale::verification::ImplicationConstraintAttribute Class Reference	403
7.84.1 Detailed Description	403
7.84.2 Member Data Documentation	403
7.84.2.1 constraint	403
7.85 multiscale::verification::ImplicationLogicPropertyAttribute Class Reference	403
7.85.1 Detailed Description	404
7.85.2 Member Data Documentation	404
7.85.2.1 logicProperty	404
7.86 multiscale::IndexOutOfBoundsException Class Reference	404
7.86.1 Detailed Description	407
7.86.2 Constructor & Destructor Documentation	407
7.86.2.1 IndexOutOfBoundsException	407
7.86.2.2 IndexOutOfBoundsException	407
7.86.2.3 IndexOutOfBoundsException	407
7.87 multiscale::InvalidInputException Class Reference	407
7.87.1 Detailed Description	410
7.87.2 Constructor & Destructor Documentation	410
7.87.2.1 InvalidInputException	410
7.87.2.2 InvalidInputException	410
7.87.2.3 InvalidInputException	410
7.88 multiscale::IOException Class Reference	410
7.88.1 Detailed Description	413
7.88.2 Constructor & Destructor Documentation	413
7.88.2.1 IOException	413

7.88.2.2	IOException	413
7.88.2.3	IOException	413
7.89	multiscale::LexicographicNumberIterator Class Reference	413
7.89.1	Detailed Description	416
7.89.2	Constructor & Destructor Documentation	416
7.89.2.1	LexicographicNumberIterator	416
7.89.2.2	~LexicographicNumberIterator	416
7.89.3	Member Function Documentation	416
7.89.3.1	digitsToNumber	416
7.89.3.2	hasNextInitialised	417
7.89.3.3	initialise	417
7.89.3.4	isLargerThanUpperBound	417
7.89.3.5	number	417
7.89.3.6	numberToDigits	418
7.89.3.7	padWithZeros	419
7.89.3.8	resetCurrentNumber	419
7.89.3.9	reverseDigits	419
7.89.4	Member Data Documentation	419
7.89.4.1	currentNumberDigits	419
7.89.4.2	upperBoundDigits	420
7.90	multiscale::verification::LogicPropertyAttribute Class Reference	420
7.90.1	Detailed Description	422
7.90.2	Constructor & Destructor Documentation	422
7.90.2.1	LogicPropertyAttribute	422
7.90.2.2	LogicPropertyAttribute	422
7.90.3	Member Data Documentation	422
7.90.3.1	firstLogicProperty	422
7.90.3.2	nextLogicProperties	422
7.91	multiscale::verification::LogicPropertyDataReader Class Reference	422
7.91.1	Detailed Description	424
7.91.2	Member Function Documentation	424
7.91.2.1	appendLineUsingStringBuilder	424
7.91.2.2	createNewLogicProperty	424
7.91.2.3	processLineFromInputFile	425
7.91.2.4	readLogicPropertiesFromFile	425
7.91.2.5	readLogicPropertiesFromOpenStream	425
7.91.2.6	readLogicPropertiesFromValidFilepath	426
7.91.2.7	removeStringBuilderContents	426
7.91.3	Member Data Documentation	426
7.91.3.1	CHAR_START_COMMENT	426

7.91.3.2	CHAR_START_LOGIC_PROPERTY	426
7.91.3.3	ERR_INVALID_INPUT_PATH	426
7.91.3.4	ERR_OPEN_INPUT_FILE	426
7.91.3.5	stringBuilder	426
7.92	multiscale::verification::LogicPropertyGrammar< Iterator > Class Template Reference	427
7.92.1	Detailed Description	432
7.92.2	Constructor & Destructor Documentation	432
7.92.2.1	LogicPropertyGrammar	432
7.92.3	Member Function Documentation	432
7.92.3.1	assignNamesToComparatorRules	432
7.92.3.2	assignNamesToComposedLogicPropertyRules	433
7.92.3.3	assignNamesToLogicPropertiesRules	433
7.92.3.4	assignNamesToLogicPropertyRules	433
7.92.3.5	assignNamesToPrimaryLogicPropertyRules	433
7.92.3.6	assignNamesToProbabilisticLogicPropertyRules	433
7.92.3.7	assignNamesToRules	433
7.92.3.8	assignNamesToSimilarityMeasureRules	433
7.92.3.9	initialise	433
7.92.3.10	initialiseComparatorRuleDebugging	434
7.92.3.11	initialiseComparatorRules	434
7.92.3.12	initialiseComposedLogicPropertyErrorHandlingSupport	434
7.92.3.13	initialiseComposedLogicPropertyRule	434
7.92.3.14	initialiseComposedLogicPropertyRuleDebugging	434
7.92.3.15	initialiseDebugSupport	434
7.92.3.16	initialiseErrorHandlingSupport	434
7.92.3.17	initialiseGrammar	435
7.92.3.18	initialiseLogicPropertiesErrorHandlingSupport	435
7.92.3.19	initialiseLogicPropertiesRules	435
7.92.3.20	initialiseLogicPropertiesRulesDebugging	435
7.92.3.21	initialiseLogicPropertyRule	435
7.92.3.22	initialiseLogicPropertyRuleDebugging	435
7.92.3.23	initialisePrimaryLogicPropertyErrorHandlingSupport	435
7.92.3.24	initialisePrimaryLogicPropertyRule	435
7.92.3.25	initialisePrimaryLogicPropertyRuleDebugging	436
7.92.3.26	initialiseProbabilisticLogicPropertyErrorHandlingSupport	436
7.92.3.27	initialiseProbabilisticLogicPropertyRule	436
7.92.3.28	initialiseProbabilisticLogicPropertyRuleDebugging	436
7.92.3.29	initialiseRulesDebugging	436
7.92.3.30	initialiseSimilarityMeasureRuleDebugging	436
7.92.3.31	initialiseSimilarityMeasureRules	436

7.92.4 Member Data Documentation	436
7.92.4.1 andLogicPropertyRule	437
7.92.4.2 binaryNumericNumericRule	437
7.92.4.3 changeMeasureRule	437
7.92.4.4 changeTemporalNumericMeasureRule	437
7.92.4.5 comparatorNonEqualTypeParser	437
7.92.4.6 comparatorRule	437
7.92.4.7 equivalenceLogicPropertyRule	437
7.92.4.8 futureLogicPropertyRule	437
7.92.4.9 globalLogicPropertyRule	438
7.92.4.10 implicationLogicPropertyRule	438
7.92.4.11 logicPropertyRule	438
7.92.4.12 nextKLogicPropertyRule	438
7.92.4.13 nextLogicPropertyRule	438
7.92.4.14 notLogicPropertyRule	438
7.92.4.15 orLogicPropertyRule	438
7.92.4.16 primaryLogicPropertyRule	438
7.92.4.17 primaryNumericMeasureRule	439
7.92.4.18 probabilisticLogicPropertyComparatorRule	439
7.92.4.19 probabilisticLogicPropertyRule	439
7.92.4.20 probabilityRule	439
7.92.4.21 similarityMeasureRule	439
7.92.4.22 similarityMeasureTypeParser	439
7.92.4.23 similarityTemporalNumericCollectionRule	439
7.92.4.24 temporalNumericCollectionRule	440
7.92.4.25 temporalNumericComparisonRule	440
7.92.4.26 temporalNumericMeasureRule	440
7.92.4.27 unaryNumericNumericRule	440
7.92.4.28 untilLogicPropertyRule	440
7.93 multiscale::verification::LogicPropertyVisitor Class Reference	440
7.93.1 Detailed Description	445
7.93.2 Constructor & Destructor Documentation	445
7.93.2.1 LogicPropertyVisitor	445
7.93.3 Member Function Documentation	445
7.93.3.1 areSimilarValues	445
7.93.3.2 constructEvaluationLogicProperty	446
7.93.3.3 evaluate	446
7.93.3.4 evaluate	446
7.93.3.5 evaluateChangeLhsTemporalNumericMeasure	446
7.93.3.6 evaluateChangeTemporalNumericMeasure	447

7.93.3.7	evaluateFutureLogicProperty	447
7.93.3.8	evaluateGlobalLogicProperty	447
7.93.3.9	evaluateNextKLogicProperty	448
7.93.3.10	evaluateNextKLogicProperty	448
7.93.3.11	evaluateNextLogicProperties	448
7.93.3.12	evaluateNextLogicProperty	449
7.93.3.13	evaluatePrecedingLogicProperties	449
7.93.3.14	evaluateSimilarityTemporalNumericCollection	449
7.93.3.15	evaluateTemporalNumericComparison	450
7.93.3.16	evaluateTemporalNumericMeasure	450
7.93.3.17	evaluateUntilLogicProperty	451
7.93.3.18	isLhsSimilarToRhs	452
7.93.3.19	operator()	452
7.93.3.20	operator()	452
7.93.3.21	operator()	453
7.93.3.22	operator()	453
7.93.3.23	operator()	453
7.93.3.24	operator()	453
7.93.3.25	operator()	454
7.93.3.26	operator()	454
7.93.3.27	operator()	454
7.93.3.28	operator()	454
7.93.3.29	operator()	455
7.93.3.30	operator()	455
7.93.3.31	operator()	455
7.93.3.32	operator()	456
7.93.3.33	operator()	456
7.93.3.34	operator()	456
7.93.3.35	printExceptionMessage	456
7.93.4	Member Data Documentation	457
7.93.4.1	evaluationLogicProperty	457
7.93.4.2	precedingTruthValue	457
7.93.4.3	trace	457
7.93.4.4	typeSemanticsTable	457
7.94	multiscale::analysis::MatFactory Class Reference	457
7.94.1	Detailed Description	460
7.94.2	Constructor & Destructor Documentation	460
7.94.2.1	MatFactory	460
7.94.2.2	~MatFactory	460
7.94.3	Member Function Documentation	460

7.94.3.1	convertToIntensity	460
7.94.3.2	create	461
7.94.3.3	createFromViewerImage	461
7.94.3.4	initInputFile	461
7.94.3.5	isValidViewerImage	461
7.94.3.6	maxColourBarIntensityFromViewerImage	462
7.94.3.7	processConcentrations	462
7.94.4	Member Data Documentation	462
7.94.4.1	cols	462
7.94.4.2	ERR_IMG_RESOLUTION	462
7.94.4.3	ERR_IN_EXTRA_DATA	462
7.94.4.4	ERR_INPUT_OPEN	462
7.94.4.5	rows	463
7.94.4.6	simulationTime	463
7.95	multiscale::MinEnclosingTriangleFinder Class Reference	463
7.95.1	Detailed Description	467
7.95.2	Constructor & Destructor Documentation	467
7.95.2.1	MinEnclosingTriangleFinder	467
7.95.2.2	\sim MinEnclosingTriangleFinder	467
7.95.3	Member Function Documentation	467
7.95.3.1	advance	467
7.95.3.2	advanceBToRightChain	468
7.95.3.3	areIdenticalLines	468
7.95.3.4	areIntersectingLines	468
7.95.3.5	find	469
7.95.3.6	findGammaIntersectionPoints	469
7.95.3.7	findMinEnclosingTriangle	469
7.95.3.8	findMinEnclosingTriangle	470
7.95.3.9	findMinTriangle	471
7.95.3.10	findVertexCOnSideB	471
7.95.3.11	gamma	471
7.95.3.12	height	472
7.95.3.13	height	472
7.95.3.14	initialise	472
7.95.3.15	initialiseAlgorithmVariables	473
7.95.3.16	initialiseConvexPolygon	473
7.95.3.17	intersects	473
7.95.3.18	intersectsAbove	473
7.95.3.19	intersectsAboveOrBelow	474
7.95.3.20	intersectsBelow	474

7.95.3.21 isFlushAngleBetweenPredecessorAndSuccessor	474
7.95.3.22 isGammaAngleBetween	474
7.95.3.23 isGammaAngleEqualTo	475
7.95.3.24 isLocalMinimalTriangle	475
7.95.3.25 isNotBTangency	475
7.95.3.26 isValidMinimalTriangle	475
7.95.3.27 lineEquationParameters	476
7.95.3.28 middlePointOfSideB	476
7.95.3.29 moveAIfLowAndBIfHigh	476
7.95.3.30 predecessor	476
7.95.3.31 returnMinEnclosingTriangle	476
7.95.3.32 searchForBTangency	477
7.95.3.33 successor	477
7.95.3.34 updateMinEnclosingTriangle	477
7.95.3.35 updateSideB	477
7.95.3.36 updateSidesBA	478
7.95.3.37 updateSidesCA	478
7.95.4 Member Data Documentation	478
7.95.4.1 a	478
7.95.4.2 area	478
7.95.4.3 b	478
7.95.4.4 c	479
7.95.4.5 CONVEX_HULL_CLOCKWISE	479
7.95.4.6 ERR_MIDPOINT_SIDE_B	479
7.95.4.7 ERR_NR_POINTS	479
7.95.4.8 ERR_SIDE_B_GAMMA	479
7.95.4.9 ERR_TRIANGLE_VERTICES	479
7.95.4.10 ERR_VERTEX_C_ON_SIDE_B	479
7.95.4.11 INTERSECTS_ABOVE	479
7.95.4.12 INTERSECTS_BELOW	480
7.95.4.13 INTERSECTS_CRITICAL	480
7.95.4.14 INTERSECTS_LIMIT	480
7.95.4.15 nrOfPoints	480
7.95.4.16 polygon	480
7.95.4.17 sideAEndVertex	480
7.95.4.18 sideAStartVertex	480
7.95.4.19 sideBEndVertex	480
7.95.4.20 sideBStartVertex	481
7.95.4.21 sideCEndVertex	481
7.95.4.22 sideCStartVertex	481

7.95.4.23	VALIDATION_SIDE_A_TANGENT	481
7.95.4.24	VALIDATION_SIDE_B_TANGENT	481
7.95.4.25	VALIDATION_SIDES_FLUSH	481
7.95.4.26	validationFlag	481
7.95.4.27	vertexA	482
7.95.4.28	vertexB	482
7.95.4.29	vertexC	482
7.96	multiscaletest::MinEnclosingTriangleFinderTest Class Reference	482
7.96.1	Detailed Description	486
7.96.2	Constructor & Destructor Documentation	486
7.96.2.1	MinEnclosingTriangleFinderTest	486
7.96.2.2	~MinEnclosingTriangleFinderTest	486
7.96.3	Member Function Documentation	486
7.96.3.1	ArePointsEnclosed	486
7.96.3.2	GetRandomNrOfExecutions	486
7.96.3.3	GetRandomNrOfPoints	486
7.96.3.4	IsOneEdgeFlush	486
7.96.3.5	IsTriangleTouchingPolygon	486
7.96.3.6	RunTest	487
7.96.3.7	TestMorePoints	487
7.96.3.8	TestMorePointsAndNonEmptyTriangle	487
7.96.3.9	TestNoPoints	487
7.96.3.10	TestOnePoint	487
7.96.3.11	TestPointsWithNegativeCoordinates	487
7.96.3.12	TestPointsWithNegativeXCoordinate	487
7.96.3.13	TestPointsWithNegativeYCoordinate	487
7.96.3.14	TestRandomPoints	487
7.96.3.15	TestThreePoints	488
7.96.3.16	TestTwoPoints	488
7.96.3.17	ValidateTestResults	488
7.96.4	Member Data Documentation	488
7.96.4.1	area	488
7.96.4.2	convexHull	488
7.96.4.3	MAX_NR_EXECUTIONS	488
7.96.4.4	MAX_NR_POINTS	488
7.96.4.5	MIN_NR_EXECUTIONS	488
7.96.4.6	MIN_NR_POINTS	488
7.96.4.7	POINT_IN_TRIANGLE_THRESH	489
7.96.4.8	points	489
7.96.4.9	triangle	489

7.97 multiscale::verification::ModelChecker Class Reference	489
7.97.1 Detailed Description	493
7.97.2 Constructor & Destructor Documentation	493
7.97.2.1 ModelChecker	493
7.97.2.2 ~ModelChecker	493
7.97.3 Member Function Documentation	493
7.97.3.1 acceptsMoreTraces	493
7.97.3.2 doesPropertyHold	494
7.97.3.3 doesPropertyHoldUsingPValues	494
7.97.3.4 evaluate	494
7.97.3.5 getDetailedResults	494
7.97.3.6 getDetailedResultsUsingPValues	494
7.97.3.7 isGreaterThanOrEqualToComparator	495
7.97.3.8 requiresMoreTraces	495
7.97.3.9 updateAlternativeHypothesisPValue	495
7.97.3.10 updateDerivedModelCheckerForFalseEvaluation	495
7.97.3.11 updateDerivedModelCheckerForTrueEvaluation	495
7.97.3.12 updateHypothesesPValues	496
7.97.3.13 updateHypothesesPValuesConsideringComparator	496
7.97.3.14 updateHypothesesPValuesForGreaterThan	496
7.97.3.15 updateHypothesesPValuesForLessThan	496
7.97.3.16 updateModelChecker	496
7.97.3.17 updateModelCheckerForEvaluationResult	497
7.97.3.18 updateModelCheckerForFalseEvaluation	497
7.97.3.19 updateModelCheckerForTrueEvaluation	497
7.97.3.20 updateNullAndAlternativeHypothesesPValues	497
7.97.3.21 updateNullHypothesisPValue	498
7.97.4 Member Data Documentation	499
7.97.4.1 abstractSyntaxTree	499
7.97.4.2 alternativeHypothesisPValue	499
7.97.4.3 arePValuesUpdatedFlag	499
7.97.4.4 MSG_OUTPUT_P_VALUE_BEGIN	499
7.97.4.5 MSG_OUTPUT_P_VALUE_END	499
7.97.4.6 MSG_OUTPUT_P_VALUE_MIDDLE1	499
7.97.4.7 MSG_OUTPUT_P_VALUE_MIDDLE2	500
7.97.4.8 nullHypothesisPValue	500
7.97.4.9 totalNumberOfEvaluations	500
7.97.4.10 totalNumberOfTrueEvaluations	500
7.97.4.11 typeSemanticsTable	500
7.98 multiscale::verification::ModelCheckerFactory Class Reference	501

7.98.1 Detailed Description	501
7.98.2 Constructor & Destructor Documentation	502
7.98.2.1 ModelCheckerFactory	502
7.98.2.2 ~ModelCheckerFactory	502
7.98.3 Member Function Documentation	502
7.98.3.1 createInstance	502
7.99 multiscaletest::ModelCheckerTest Class Reference	502
7.99.1 Detailed Description	505
7.99.2 Member Function Documentation	505
7.99.2.1 Initialise	505
7.99.2.2 InitialiseAbstractSyntaxTree	506
7.99.2.3 InitialiseModelChecker	506
7.99.2.4 InitialiseSpatioTemporalTraces	506
7.99.2.5 InitialiseSpatioTemporalTraceWithClusterednessValues	506
7.99.2.6 InitialiseTypeSemanticsTable	506
7.99.2.7 RunModelCheckingTest	506
7.99.2.8 RunTest	507
7.99.2.9 ValidateTestResults	507
7.99.3 Member Data Documentation	507
7.99.3.1 abstractSyntaxTree	507
7.99.3.2 evaluationResult	507
7.99.3.3 modelChecker	507
7.99.3.4 traces	507
7.99.3.5 typeSemanticsTable	508
7.100 multiscale::verification::ModelCheckingException Class Reference	508
7.100.1 Detailed Description	511
7.100.2 Constructor & Destructor Documentation	511
7.100.2.1 ModelCheckingException	511
7.100.2.2 ModelCheckingException	511
7.101 multiscale::verification::ModelCheckingHelpRequestException Class Reference	511
7.101.1 Detailed Description	514
7.101.2 Constructor & Destructor Documentation	514
7.101.2.1 ModelCheckingHelpRequestException	514
7.101.2.2 ModelCheckingHelpRequestException	514
7.102 multiscale::verification::ModelCheckingManager Class Reference	514
7.102.1 Detailed Description	517
7.102.2 Constructor & Destructor Documentation	517
7.102.2.1 ModelCheckingManager	517
7.102.2.2 ~ModelCheckingManager	517
7.102.3 Member Function Documentation	518

7.102.3.1 areUnfinishedModelCheckingTasks	518
7.102.3.2 createModelCheckers	518
7.102.3.3 createNewEvaluationResults	518
7.102.3.4 executeExtraEvaluationProgram	518
7.102.3.5 executeExtraEvaluationProgramAndPrintMessage	518
7.102.3.6 getNextSpatialTemporalTrace	519
7.102.3.7 initialise	519
7.102.3.8 initialiseExtraEvaluationTimeCounters	519
7.102.3.9 initialiseLogicProperties	519
7.102.3.10 initialiseTypeSemanticsTable	520
7.102.3.11 isEvaluationTimeRemaining	520
7.102.3.12 isValidLogicProperty	520
7.102.3.13 outputDetailedEvaluationResults	520
7.102.3.14 outputModelCheckerResults	521
7.102.3.15 outputModelCheckersResults	522
7.102.3.16 outputModelCheckersResultsAndPrintMessage	522
7.102.3.17 parseLogicProperties	522
7.102.3.18 parseLogicPropertiesAndPrintMessage	522
7.102.3.19 parseLogicProperty	522
7.102.3.20 parseLogicPropertyAndPrintMessages	523
7.102.3.21 printParsingMessage	523
7.102.3.22 runModelCheckerForTrace	523
7.102.3.23 runModelCheckers	523
7.102.3.24 runModelCheckersAndPrintMessage	524
7.102.3.25 runModelCheckersAndRequestAdditionalTraces	524
7.102.3.26 runModelCheckersForCurrentlyExistingTraces	524
7.102.3.27 runModelCheckersForTrace	524
7.102.3.28 runModelCheckingAndOutputResults	524
7.102.3.29 runModelCheckingTasks	525
7.102.3.30 setExtraEvaluationProgramPath	525
7.102.3.31 setShouldPrintDetailedEvaluation	525
7.102.3.32 storeNewSpatialTemporalTracePath	525
7.102.3.33 updateEvaluationResults	526
7.102.3.34 updateExtraEvaluationStartTime	526
7.102.3.35 updateTraceReader	526
7.102.3.36 waitBeforeRetry	526
7.102.4 Member Data Documentation	526
7.102.4.1 abstractSyntaxTrees	527
7.102.4.2 evaluationResults	527
7.102.4.3 extraEvaluationElapsedTIme	527

7.102.4.4 extraEvaluationProgramPath	527
7.102.4.5 extraEvaluationStartTime	527
7.102.4.6 extraEvaluationTime	527
7.102.4.7 logicProperties	527
7.102.4.8 logicPropertyReader	528
7.102.4.9 modelCheckers	528
7.102.4.10parser	528
7.102.4.11PARSER_EMPTY_LOGIC_PROPERTY	528
7.102.4.12shouldPrintDetailedEvaluation	528
7.102.4.13TRACE_INPUT_REFRESH_TIMEOUT	528
7.102.4.14traceReader	528
7.102.4.15tracesPaths	529
7.102.4.16typeSemanticsTable	529
7.103multiscale::verification::ModelCheckingOutputWriter Class Reference	529
7.103.1 Detailed Description	533
7.103.2 Member Function Documentation	533
7.103.2.1 isTraceEvaluatedForLogicProperty	533
7.103.2.2 isTraceEvaluatedTrueForLogicProperty	533
7.103.2.3 printDetailedEvaluationResults	534
7.103.2.4 printDetailedEvaluationResults	534
7.103.2.5 printDetailedEvaluationResultsForLogicProperties	534
7.103.2.6 printDetailedEvaluationResultsIntroductionMessage	535
7.103.2.7 printDetailedTraceEvaluationResult	535
7.103.2.8 printEvaluationResultsSummary	535
7.103.2.9 printEvaluationResultsSummary	536
7.103.2.10printExecuteExtraEvaluationProgramMessage	536
7.103.2.11printFailedMessage	536
7.103.2.12printInitialisationMessage	537
7.103.2.13printIntroductionMessage	538
7.103.2.14printLogicPropertyDetailedEvaluationResults	538
7.103.2.15printLogicPropertyForResult	538
7.103.2.16printLogicPropertyWithTag	539
7.103.2.17printModelCheckingDetailedResult	539
7.103.2.18printModelCheckingResult	539
7.103.2.19printModelCheckingResultMessage	539
7.103.2.20printModelCheckingResultsIntroductionMessage	540
7.103.2.21printParsingLogicPropertiesBeginMessage	540
7.103.2.22printParsingLogicPropertiesEndMessage	540
7.103.2.23printParsingLogicPropertyMessage	540
7.103.2.24printResultTag	541

7.103.2.25printSeparatorTag	541
7.103.2.26printStartModelCheckingExecutionMessage	541
7.103.2.27printStartTraceEvaluationMessage	541
7.103.2.28printSuccessMessage	541
7.103.2.29printTimeoutMessage	542
7.103.2.30printTraceEvaluationResult	542
7.103.2.31printTruthValueDependentMessage	542
7.103.2.32updateSummaryEvaluationResults	542
7.103.3 Member Data Documentation	543
7.103.3.1 MSG_EVALUATION_RESULTS_INTRODUCTION	543
7.103.3.2 MSG_EVALUATION_SUMMARY_BEGIN	543
7.103.3.3 MSG_EVALUATION_SUMMARY_END	543
7.103.3.4 MSG_EXECUTION_TIMEOUT_BEGIN	543
7.103.3.5 MSG_EXECUTION_TIMEOUT_END	543
7.103.3.6 MSG_INIT_EXECUTION_PARAMETERS	544
7.103.3.7 MSG_INIT_EXTRA_EVALUATION_TIME	544
7.103.3.8 MSG_INIT_LOGIC_PROPERTIES_PATH	544
7.103.3.9 MSG_INIT_TRACES_FOLDER_PATH	544
7.103.3.10MSG_INTRO_CONTACT	544
7.103.3.11MSG_INTRO_COPYRIGHT	544
7.103.3.12MSG_INTRO_MODEL_CHECKING_PARAMETERS	544
7.103.3.13MSG_INTRO_MODEL_CHECKING_TYPE	544
7.103.3.14MSG_INTRO_NAME	545
7.103.3.15MSG_LOGIC_PROPERTY HOLDS	545
7.103.3.16MSG_LOGIC_PROPERTY_HOLDS_FALSE	545
7.103.3.17MSG_LOGIC_PROPERTY_HOLDS_TRUE	545
7.103.3.18MSG_PARSING_INTRODUCTION	545
7.103.3.19MSG_RESULTS_INTRODUCTION	545
7.103.3.20MSG_START_EXTRA_EVALUATION_PROGRAM_EXECUTION	545
7.103.3.21MSG_START_MODEL_CHECKING_EXECUTION	545
7.103.3.22MSG_START_TRACE_EVALUATION	546
7.103.3.23TAG_DETAILS	546
7.103.3.24TAG_EXECUTE	546
7.103.3.25TAG_FAILED	546
7.103.3.26TAG_FALSE	546
7.103.3.27TAG_INIT	546
7.103.3.28TAG_INTRO	546
7.103.3.29TAG_PARSING	546
7.103.3.30TAG_RESULT	546
7.103.3.31TAG_SEPARATOR	547

7.103.3.32TAG_SUCCESS	547
7.103.3.33TAG_TIMEOUT	547
7.103.3.34TAG_TRUE	547
7.104 multiscale::MultiplicationOperation Class Reference	547
7.104.1 Detailed Description	548
7.104.2 Member Function Documentation	548
7.104.2.1 operator()	548
7.105 multiscale::MultiscaleException Class Reference	548
7.105.1 Detailed Description	550
7.105.2 Constructor & Destructor Documentation	550
7.105.2.1 MultiscaleException	550
7.105.2.2 MultiscaleException	550
7.105.2.3 MultiscaleException	550
7.105.3 Member Function Documentation	550
7.105.3.1 constructExplanatoryString	550
7.105.3.2 rawMessage	550
7.105.3.3 what	551
7.105.4 Member Data Documentation	551
7.105.4.1 explanatoryString	551
7.105.4.2 message	551
7.106 multiscaletest::MultiscaleTest Class Reference	551
7.106.1 Detailed Description	552
7.106.2 Constructor & Destructor Documentation	552
7.106.2.1 ~MultiscaleTest	553
7.106.3 Member Function Documentation	553
7.106.3.1 RunTest	553
7.106.3.2 SetUp	553
7.106.3.3 TearDown	553
7.106.3.4 ValidateTestResults	553
7.106.4 Member Data Documentation	553
7.106.4.1 validationFlag	553
7.107 multiscale::verification::NextKLogicPropertyAttribute Class Reference	553
7.107.1 Detailed Description	554
7.107.2 Member Data Documentation	554
7.107.2.1 logicProperty	554
7.107.2.2 nrOfTimepointsAhead	554
7.108 multiscale::verification::NextLogicPropertyAttribute Class Reference	554
7.108.1 Detailed Description	555
7.108.2 Member Data Documentation	555
7.108.2.1 logicProperty	555

7.109 multiscale::verification::Nil Class Reference	555
7.109.1 Detailed Description	556
7.110 multiscale::verification::NotConstraintAttribute Class Reference	556
7.110.1 Detailed Description	557
7.110.2 Member Data Documentation	557
7.110.2.1 constraint	557
7.111 multiscale::verification::NotLogicPropertyAttribute Class Reference	557
7.111.1 Detailed Description	557
7.111.2 Member Data Documentation	557
7.111.2.1 logicProperty	558
7.112 multiscale::NumberIterator Class Reference	558
7.112.1 Detailed Description	561
7.112.2 Constructor & Destructor Documentation	561
7.112.2.1 NumberIterator	561
7.112.2.2 ~NumberIterator	561
7.112.3 Member Function Documentation	561
7.112.3.1 hasNext	561
7.112.3.2 hasNextInitialised	561
7.112.3.3 hasNextNotInitialised	561
7.112.3.4 init	561
7.112.3.5 initialise	562
7.112.3.6 number	562
7.112.3.7 reset	562
7.112.3.8 resetCurrentNumber	562
7.112.4 Member Data Documentation	562
7.112.4.1 isInitialised	562
7.112.4.2 upperBound	562
7.113 multiscale::Numeric Class Reference	562
7.113.1 Detailed Description	567
7.113.2 Member Function Documentation	567
7.113.2.1 almostEqual	567
7.113.2.2 applyOperation	567
7.113.2.3 areOverflowUnderflowFlagsSet	568
7.113.2.4 average	568
7.113.2.5 average	568
7.113.2.6 combinations	568
7.113.2.7 computeCombinations	569
7.113.2.8 computeKurtosisFirstTerm	569
7.113.2.9 computeKurtosisLastTerm	569
7.113.2.10 computeKurtosisMiddleTerm	569

7.113.2.11computeMode	570
7.113.2.12computeQuartileValue	570
7.113.2.13computeSkewFirstTerm	570
7.113.2.14computeSkewLastTerm	570
7.113.2.15covariance	570
7.113.2.16covariance	571
7.113.2.17factorial	571
7.113.2.18geometricMean	571
7.113.2.19geometricMean	571
7.113.2.20greaterOrEqual	572
7.113.2.21harmonicMean	572
7.113.2.22harmonicMean	572
7.113.2.23isPositive	573
7.113.2.24kurtosis	574
7.113.2.25kurtosis	574
7.113.2.26lessOrEqual	574
7.113.2.27log	575
7.113.2.28maximum	575
7.113.2.29maximum	575
7.113.2.30maximum	575
7.113.2.31median	576
7.113.2.32median	577
7.113.2.33minimum	577
7.113.2.34minimum	577
7.113.2.35mode	577
7.113.2.36mode	578
7.113.2.37numberInverse	579
7.113.2.38percentile	579
7.113.2.39percentile	579
7.113.2.40printNoValuesWarningMessage	579
7.113.2.41product	580
7.113.2.42product	580
7.113.2.43quartile	580
7.113.2.44quartile	580
7.113.2.45resetOverflowUnderflowFlags	581
7.113.2.46sign	581
7.113.2.47skew	581
7.113.2.48skew	581
7.113.2.49standardDeviation	582
7.113.2.50standardDeviation	583

7.113.2.51sum	583
7.113.2.52sum	583
7.113.2.53validateLogBase	583
7.113.2.54validateLogNumber	584
7.113.2.55validateLogNumberAndBase	584
7.113.2.56validatePercentile	584
7.113.2.57validateQuartile	584
7.113.2.58variance	585
7.113.2.59variance	585
7.113.3 Member Data Documentation	585
7.113.3.1 epsilon	585
7.113.3.2 ERR_COMBINATIONS_END	585
7.113.3.3 ERR_COMBINATIONS_MIDDLE	585
7.113.3.4 ERR_COMBINATIONS_START	585
7.113.3.5 ERR_LOG_BASE_END	586
7.113.3.6 ERR_LOG_BASE_START	586
7.113.3.7 ERR_LOG_NUMBER_END	586
7.113.3.8 ERR_LOG_NUMBER_START	586
7.113.3.9 ERR_OVERFLOW_UNDERFLOW	586
7.113.3.10ERR_PERCENTILE_VALUE_END	586
7.113.3.11ERR_PERCENTILE_VALUE_START	586
7.113.3.12ERR_QUARTILE_VALUE_END	586
7.113.3.13ERR_QUARTILE_VALUE_START	587
7.113.3.14WRN_AVERAGE_FUNCTION_NAME	587
7.113.3.15WRN_COVARIANCE_FUNCTION_NAME	587
7.113.3.16WRN_GEOMETRIC_MEAN_FUNCTION_NAME	587
7.113.3.17WRN_HARMONIC_MEAN_FUNCTION_NAME	587
7.113.3.18WRN_KURTOSIS_FUNCTION_NAME	587
7.113.3.19WRN_MAXIMUM_FUNCTION_NAME	587
7.113.3.20WRN_MEDIAN_FUNCTION_NAME	587
7.113.3.21WRN_MINIMUM_FUNCTION_NAME	587
7.113.3.22WRN_MODE_FUNCTION_NAME	588
7.113.3.23WRN_NOT_ENOUGH_VALUES_END	588
7.113.3.24WRN_NOT_ENOUGH_VALUES_START	588
7.113.3.25WRN_NUMBER_INVERSE	588
7.113.3.26WRN_PERCENTILE_FUNCTION_NAME	588
7.113.3.27WRN_PRODUCT_FUNCTION_NAME	588
7.113.3.28WRN_QUARTILE_FUNCTION_NAME	588
7.113.3.29WRN_SKEW_FUNCTION_NAME	588
7.113.3.30WRN_STANDARD_DEVIATION_FUNCTION_NAME	588

7.113.3.31WRN_SUM_FUNCTION_NAME	589
7.113.3.32WRN_VARIANCE_FUNCTION_NAME	589
7.114multiscale::verification::NumericEvaluator Class Reference	589
7.114.1 Detailed Description	590
7.114.2 Member Function Documentation	590
7.114.2.1 evaluate	590
7.114.2.2 evaluate	590
7.114.2.3 evaluate	590
7.114.2.4 evaluate	591
7.114.2.5 evaluate	591
7.115multiscale::NumericException Class Reference	592
7.115.1 Detailed Description	594
7.115.2 Constructor & Destructor Documentation	594
7.115.2.1 NumericException	594
7.115.2.2 NumericException	594
7.115.2.3 NumericException	594
7.116multiscale::verification::NumericMeasureAttribute Class Reference	594
7.116.1 Detailed Description	595
7.116.2 Member Data Documentation	595
7.116.2.1 numericMeasure	595
7.117multiscale::verification::NumericMeasureCollectionAttribute Class Reference	595
7.117.1 Detailed Description	595
7.117.2 Member Data Documentation	595
7.117.2.1 numericMeasureCollection	596
7.118multiscale::verification::NumericMeasureCollectionEvaluator Class Reference	596
7.118.1 Detailed Description	596
7.118.2 Member Function Documentation	596
7.118.2.1 evaluateSpatialMeasureCollection	597
7.118.2.2 evaluateTemporalNumericCollection	598
7.119multiscale::verification::NumericMeasureCollectionVisitor Class Reference	598
7.119.1 Detailed Description	602
7.119.2 Constructor & Destructor Documentation	602
7.119.2.1 NumericMeasureCollectionVisitor	602
7.119.3 Member Function Documentation	602
7.119.3.1 computeHomogeneousComponentTimeSpans	602
7.119.3.2 computeHomogeneousComponentValues	602
7.119.3.3 evaluateChangeTemporalNumericCollection	603
7.119.3.4 evaluateHomogeneousHomogeneousTimeseries	604
7.119.3.5 evaluateTemporalNumericMeasureCollection	604
7.119.3.6 evaluateTemporalNumericMeasureCollectionTimepoints	605

7.119.3.7 evaluateTimeseriesComponent	606
7.119.3.8 evaluateTimeseriesTimeseriesComponent	606
7.119.3.9 indicesSubCollection	606
7.119.3.10operator()	607
7.119.3.11operator()	607
7.119.3.12operator()	607
7.119.3.13operator()	608
7.119.3.14operator()	609
7.119.3.15operator()	609
7.119.3.16subCollection	610
7.119.4 Member Data Documentation	610
7.119.4.1 trace	610
7.119.4.2 typeSemanticsTable	610
7.120multiscale::NumericRangeManipulator Class Reference	610
7.120.1 Detailed Description	611
7.120.2 Member Function Documentation	611
7.120.2.1 convertFromRange	611
7.121multiscale::verification::NumericSpatialMeasureAttribute Class Reference	611
7.121.1 Detailed Description	612
7.121.2 Member Data Documentation	612
7.121.2.1 numericSpatialMeasure	612
7.122multiscale::verification::NumericStateVariableAttribute Class Reference	612
7.122.1 Detailed Description	613
7.122.2 Member Data Documentation	614
7.122.2.1 semanticType	614
7.122.2.2 stateVariable	614
7.123multiscale::verification::NumericStateVariableGrammar< Iterator > Class Template Reference	614
7.123.1 Detailed Description	617
7.123.2 Constructor & Destructor Documentation	617
7.123.2.1 NumericStateVariableGrammar	617
7.123.3 Member Function Documentation	617
7.123.3.1 assignNamesToRules	617
7.123.3.2 initialise	617
7.123.3.3 initialiseDebugSupport	618
7.123.3.4 initialiseErrorHandlingSupport	618
7.123.3.5 initialiseGrammar	618
7.123.3.6 initialiseRulesDebugging	618
7.123.4 Member Data Documentation	618
7.123.4.1 numericStateVariableRule	618
7.123.4.2 semanticTypeRule	618

7.123.4.3 stateVariableNameRule	618
7.123.4.4 stateVariableRule	619
7.123.4.5 stateVariableTypeRule	619
7.124 multiscale::verification::NumericStateVariableId Class Reference	619
7.124.1 Detailed Description	621
7.124.2 Constructor & Destructor Documentation	621
7.124.2.1 NumericStateVariableId	621
7.124.2.2 ~NumericStateVariableId	621
7.124.3 Member Function Documentation	621
7.124.3.1 getName	621
7.124.3.2 getSemanticType	621
7.124.3.3 operator<	622
7.124.3.4 setName	622
7.124.3.5 setSemanticType	622
7.124.3.6 toString	622
7.124.4 Member Data Documentation	622
7.124.4.1 name	622
7.124.4.2 OUTPUT_STRING_BEGIN	622
7.124.4.3 OUTPUT_STRING_END	623
7.124.4.4 OUTPUT_STRING_SEPARATOR	623
7.124.4.5 semanticType	623
7.125 multiscaletest::NumericStateVariableTraceTest Class Reference	623
7.125.1 Detailed Description	626
7.125.2 Member Function Documentation	626
7.125.2.1 InitialiseTrace	626
7.125.3 Member Data Documentation	626
7.125.3.1 clustersClusterednessMinValue	626
7.126 multiscale::verification::NumericStatisticalMeasureAttribute Class Reference	626
7.126.1 Detailed Description	627
7.126.2 Member Data Documentation	627
7.126.2.1 numericStatisticalMeasure	627
7.127 multiscale::verification::NumericVisitor Class Reference	627
7.127.1 Detailed Description	630
7.127.2 Constructor & Destructor Documentation	630
7.127.2.1 NumericVisitor	630
7.127.3 Member Function Documentation	631
7.127.3.1 evaluate	631
7.127.3.2 evaluateNumericSpatialMeasure	632
7.127.3.3 evaluatePrimaryNumericMeasure	632
7.127.3.4 operator()	632

7.127.3.5 operator()	632
7.127.3.6 operator()	633
7.127.3.7 operator()	633
7.127.3.8 operator()	633
7.127.3.9 operator()	633
7.127.3.10operator()	634
7.127.3.11operator()	634
7.127.3.12operator()	634
7.127.3.13operator()	635
7.127.4 Member Data Documentation	635
7.127.4.1 timePoint	635
7.127.4.2 typeSemanticsTable	635
7.128multiscale::OperatingSystem Class Reference	635
7.128.1 Detailed Description	637
7.128.2 Member Function Documentation	637
7.128.2.1 executeProgram	637
7.128.2.2 executeProgramAndVerifyPath	637
7.128.2.3 executeProgramOSSpecific	637
7.128.2.4 getEnvironmentVariable	638
7.128.3 Member Data Documentation	639
7.128.3.1 ERR_EXECUTE_PROGRAM	639
7.128.3.2 ERR_INVALID_PROGRAM_PATH	639
7.128.3.3 TIMEOUT_MAX_NR_SECONDS	639
7.128.3.4 TIMEOUT_NR_SECONDS	639
7.129multiscale::verification::OrConstraintAttribute Class Reference	639
7.129.1 Detailed Description	640
7.129.2 Member Data Documentation	640
7.129.2.1 constraint	640
7.130multiscale::verification::OrLogicPropertyAttribute Class Reference	640
7.130.1 Detailed Description	640
7.130.2 Member Data Documentation	641
7.130.2.1 logicProperty	641
7.131multiscale::verification::Parser Class Reference	641
7.131.1 Detailed Description	642
7.131.2 Constructor & Destructor Documentation	642
7.131.2.1 Parser	642
7.131.2.2 ~Parser	642
7.131.3 Member Function Documentation	642
7.131.3.1 checkIfErrorCase	642
7.131.3.2 initialise	643

7.131.3.3 isStringParsedCompletely	643
7.131.3.4 parse	643
7.131.3.5 parseLogicalQuery	643
7.131.3.6 parseLogicalQuery	644
7.131.3.7 setLogicalQuery	644
7.131.4 Member Data Documentation	644
7.131.4.1 grammar	644
7.131.4.2 logicalQuery	644
7.131.4.3 logicalQueryEnd	644
7.131.4.4 logicalQueryIterator	644
7.132 multiscale::verification::ParserGrammarExceptionHandler Class Reference	645
7.132.1 Detailed Description	646
7.132.2 Member Function Documentation	646
7.132.2.1 getIntroductoryErrorMessage	646
7.132.2.2 handleExpectedTokenAtEndOfString	646
7.132.2.3 handleExtraInputException	646
7.132.2.4 handleProbabilityException	647
7.132.2.5 handleUnexpectedTokenException	647
7.132.2.6 handleUnexpectedTokenInString	647
7.132.2.7 handleUnparseableInputException	647
7.132.2.8 trimRight	648
7.133 multiscale::verification::ParserGrammarExtraInputException Class Reference	648
7.133.1 Detailed Description	649
7.133.2 Constructor & Destructor Documentation	649
7.133.2.1 ParserGrammarExtraInputException	650
7.133.3 Member Function Documentation	650
7.133.3.1 getErrorString	650
7.133.4 Member Data Documentation	650
7.133.4.1 errorString	650
7.134 multiscale::verification::ParserGrammarProbabilityException Class Reference	650
7.134.1 Detailed Description	652
7.134.2 Constructor & Destructor Documentation	652
7.134.2.1 ParserGrammarProbabilityException	652
7.134.3 Member Function Documentation	652
7.134.3.1 getErrorString	652
7.134.3.2 getExpectedToken	652
7.134.4 Member Data Documentation	652
7.134.4.1 errorString	652
7.134.4.2 expectedToken	652
7.135 multiscale::verification::ParserGrammarUnexpectedTokenException Class Reference	653

7.135.1 Detailed Description	654
7.135.2 Constructor & Destructor Documentation	654
7.135.2.1 ParserGrammarUnexpectedTokenException	654
7.135.3 Member Function Documentation	654
7.135.3.1 getErrorString	654
7.135.3.2 getExpectedToken	654
7.135.4 Member Data Documentation	654
7.135.4.1 errorString	654
7.135.4.2 expectedToken	654
7.136 multiscale::verification::ParserGrammarUnparseableInputException Class Reference	655
7.136.1 Detailed Description	656
7.136.2 Constructor & Destructor Documentation	656
7.136.2.1 ParserGrammarUnparseableInputException	656
7.136.3 Member Function Documentation	656
7.136.3.1 getErrorString	656
7.136.4 Member Data Documentation	656
7.136.4.1 errorString	656
7.137 multiscale::video::PolarCsvToInputFilesConverter Class Reference	656
7.137.1 Detailed Description	659
7.137.2 Constructor & Destructor Documentation	659
7.137.2.1 PolarCsvToInputFilesConverter	659
7.137.2.2 ~PolarCsvToInputFilesConverter	659
7.137.3 Member Function Documentation	660
7.137.3.1 computeConcentration	660
7.137.3.2 computeConcentrationWrtArea	660
7.137.3.3 computeNextPositionConcentration	660
7.137.3.4 computeNonScaledConcentration	660
7.137.3.5 computeNormalisedConcentration	660
7.137.3.6 computeScaledConcentration	661
7.137.3.7 computeSimulationTime	661
7.137.3.8 convert	661
7.137.3.9 initInputModule	661
7.137.3.10 initIterators	661
7.137.3.11 initMaximumConcentration	662
7.137.3.12 initOutputFile	662
7.137.3.13 processInputModule	662
7.137.3.14 processLine	662
7.137.3.15 splitFirstPartInConcentrations	663
7.137.3.16 splitLineInConcentrations	664
7.137.3.17 splitOtherPartsInConcentrations	664

7.137.3.18updateMaximumConcentration	664
7.137.3.19validateInput	664
7.137.3.20validateInputLine	665
7.137.3.21validateSelectedConcentrationIndex	666
7.137.4 Member Data Documentation	666
7.137.4.1 circlesIterator	666
7.137.4.2 concentrationsIndex	666
7.137.4.3 ERR_INPUT_OPEN	666
7.137.4.4 ERR_INVALID_VALUE_LINE	666
7.137.4.5 ERR_INVALID_VALUE_TOKEN	666
7.137.4.6 ERR_NEG_CONCENTRATION	666
7.137.4.7 ERR_NEG_SIM_TIME	666
7.137.4.8 ERR_NR_CONCENTRATIONS	667
7.137.4.9 ERR_SELECTED_CONCENTRATION_INDEX	667
7.137.4.10INPUT_FILE_SEPARATOR	667
7.137.4.11filepath	667
7.137.4.12maximumConcentration	667
7.137.4.13nrOfConcentrationsForPosition	667
7.137.4.14nrOfConcentricCircles	667
7.137.4.15nrOfSectors	667
7.137.4.16OUTPUT_EXTENSION	667
7.137.4.17OUTPUT_FILE_SEPARATOR	667
7.137.4.18OUTPUT_SEPARATOR	668
7.137.4.19filepath	668
7.137.4.20RADIUS_MIN	668
7.137.4.21sectorsIterator	668
7.137.4.22selectedConcentrationIndex	668
7.137.4.23useLogScaling	668
7.138multiscale::video::PolarGnuplotScriptGenerator Class Reference	668
7.138.1 Detailed Description	670
7.138.2 Member Function Documentation	670
7.138.2.1 generateBody	670
7.138.2.2 generateFooter	671
7.138.2.3 generateHeader	672
7.138.2.4 generateScript	672
7.138.2.5 outputContent	672
7.138.2.6 outputFooter	673
7.138.2.7 outputHeader	674
7.138.2.8 readContentTemplate	674
7.138.3 Member Data Documentation	674

7.138.3.1 CONTENT_IN	674
7.138.3.2 FOOTER_IN	674
7.138.3.3 GNUPLOT_EXTENSION	674
7.138.3.4 HEADER_IN	675
7.138.3.5 REPLACE_CONTENT_CONCENTRATION	675
7.138.3.6 REPLACE_CONTENT_END_ANGLE	675
7.138.3.7 REPLACE_CONTENT_INDEX	675
7.138.3.8 REPLACE_CONTENT_RADIUS	675
7.138.3.9 REPLACE_CONTENT_START_ANGLE	675
7.138.3.10 REPLACE_HEADER_FILENAME	675
7.138.3.11 REPLACE_HEADER_SIM_TIME	675
7.139 multiscale::verification::PrimaryConstraintAttribute Class Reference	676
7.139.1 Detailed Description	676
7.139.2 Member Data Documentation	676
7.139.2.1 primaryConstraint	676
7.140 multiscale::verification::PrimaryLogicPropertyAttribute Class Reference	676
7.140.1 Detailed Description	677
7.140.2 Member Data Documentation	677
7.140.2.1 primaryLogicProperty	677
7.141 multiscale::verification::PrimaryNumericMeasureAttribute Class Reference	677
7.141.1 Detailed Description	678
7.141.2 Member Data Documentation	678
7.141.2.1 primaryNumericMeasure	678
7.142 multiscale::verification::PrimaryNumericMeasureGrammar< Iterator > Singleton Reference	678
7.142.1 Detailed Description	682
7.142.2 Constructor & Destructor Documentation	682
7.142.2.1 PrimaryNumericMeasureGrammar	682
7.142.3 Member Function Documentation	682
7.142.3.1 assignNamesToNumericSpatialMeasureRules	682
7.142.3.2 assignNamesToPrimaryNumericMeasureRules	682
7.142.3.3 assignNamesToRules	682
7.142.3.4 initialise	683
7.142.3.5 initialiseDebugSupport	683
7.142.3.6 initialiseErrorHandlingSupport	683
7.142.3.7 initialiseGrammar	683
7.142.3.8 initialiseNumericSpatialMeasureErrorHandlingSupport	683
7.142.3.9 initialiseNumericSpatialMeasureRule	683
7.142.3.10 initialiseNumericSpatialMeasureRuleDebugging	683
7.142.3.11 initialisePrimaryNumericMeasureRule	683
7.142.3.12 initialisePrimaryNumericMeasureRuleDebugging	684

7.142.3.13initialiseRulesDebugging	684
7.142.4 Member Data Documentation	684
7.142.4.1 binaryNumericMeasureRule	684
7.142.4.2 binaryNumericNumericRule	684
7.142.4.3 binaryStatisticalMeasureRule	684
7.142.4.4 binaryStatisticalQuantileMeasureRule	684
7.142.4.5 binaryStatisticalQuantileSpatialRule	684
7.142.4.6 binaryStatisticalSpatialRule	685
7.142.4.7 numericSpatialMeasureRule	685
7.142.4.8 numericStateVariableRule	685
7.142.4.9 primaryNumericMeasureRule	685
7.142.4.10spatialMeasureCollectionRule	685
7.142.4.11unaryNumericMeasureRule	685
7.142.4.12unaryNumericNumericRule	685
7.142.4.13unaryStatisticalMeasureRule	686
7.142.4.14unaryStatisticalSpatialRule	686
7.143multiscale::verification::ProbabilisticBlackBoxModelChecker Class Reference	686
7.143.1 Detailed Description	689
7.143.2 Constructor & Destructor Documentation	689
7.143.2.1 ProbabilisticBlackBoxModelChecker	689
7.143.2.2 ~ProbabilisticBlackBoxModelChecker	689
7.143.3 Member Function Documentation	689
7.143.3.1 acceptsMoreTraces	689
7.143.3.2 doesPropertyHold	689
7.143.3.3 getDetailedResults	690
7.143.3.4 requiresMoreTraces	690
7.143.3.5 updateDerivedModelCheckerForFalseEvaluation	690
7.143.3.6 updateDerivedModelCheckerForTrueEvaluation	690
7.144multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory Class Reference	690
7.144.1 Detailed Description	692
7.144.2 Constructor & Destructor Documentation	692
7.144.2.1 ProbabilisticBlackBoxModelCheckerFactory	692
7.144.2.2 ~ProbabilisticBlackBoxModelCheckerFactory	693
7.144.3 Member Function Documentation	693
7.144.3.1 createInstance	693
7.145multiscaletest::ProbabilisticBlackBoxModelCheckerTest Class Reference	693
7.145.1 Detailed Description	696
7.145.2 Member Function Documentation	696
7.145.2.1 InitialiseModelChecker	696
7.146multiscale::verification::ProbabilisticLogicPropertyAttribute Class Reference	696

7.146.1 Detailed Description	698
7.146.2 Member Function Documentation	698
7.146.2.1 evaluate	698
7.146.2.2 getComparator	698
7.146.2.3 getProbability	698
7.146.3 Member Data Documentation	698
7.146.3.1 comparator	698
7.146.3.2 ERR_TRACE_LENGTH_ZERO	699
7.146.3.3 evaluationLogicProperty	699
7.146.3.4 logicProperty	699
7.146.3.5 probability	699
7.147 multiscale::verification::ProbabilityErrorHandler Struct Reference	699
7.147.1 Detailed Description	700
7.147.2 Member Function Documentation	700
7.147.2.1 getExpectedTokenAsString	700
7.147.2.2 operator()	701
7.148 multiscale::video::RectangularCsvToInputFilesConverter Class Reference	701
7.148.1 Detailed Description	704
7.148.2 Constructor & Destructor Documentation	704
7.148.2.1 RectangularCsvToInputFilesConverter	704
7.148.2.2 ~RectangularCsvToInputFilesConverter	704
7.148.3 Member Function Documentation	704
7.148.3.1 computeConcentration	704
7.148.3.2 computeNextPositionConcentration	704
7.148.3.3 computeNonScaledConcentration	705
7.148.3.4 computeNormalisedConcentration	705
7.148.3.5 computeScaledConcentration	705
7.148.3.6 computeSimulationTime	705
7.148.3.7 convert	705
7.148.3.8 initInputFile	706
7.148.3.9 initIterators	706
7.148.3.10 initMaximumConcentration	706
7.148.3.11 initOutputFile	706
7.148.3.12 processInputFile	706
7.148.3.13 processLine	707
7.148.3.14 splitLineInConcentrations	707
7.148.3.15 splitLineInConcentrations	707
7.148.3.16 updateMaximumConcentration	707
7.148.3.17 validateInput	708
7.148.3.18 validateInputLine	709

7.148.3.19validateSelectedConcentrationIndex	709
7.148.4 Member Data Documentation	709
7.148.4.1 columnsIterator	709
7.148.4.2 concentrationsIndex	709
7.148.4.3 ERR_INPUT_OPEN	709
7.148.4.4 ERR_INVALID_VALUE_LINE	709
7.148.4.5 ERR_INVALID_VALUE_TOKEN	709
7.148.4.6 ERR_NEG_CONCENTRATION	710
7.148.4.7 ERR_NEG_SIM_TIME	710
7.148.4.8 ERR_NR_CONCENTRATIONS	710
7.148.4.9 ERR_SELECTED_CONCENTRATION_INDEX	710
7.148.4.10height	710
7.148.4.11INPUT_FILE_SEPARATOR	710
7.148.4.12filepath	710
7.148.4.13maximumConcentration	710
7.148.4.14nrOfConcentrationsForPosition	710
7.148.4.15OUTPUT_EXTENSION	711
7.148.4.16OUTPUT_FILE_SEPARATOR	711
7.148.4.17OUTPUT_SEPARATOR	711
7.148.4.18filepath	711
7.148.4.19rowsIterator	711
7.148.4.20selectedConcentrationIndex	711
7.148.4.21useLogScaling	711
7.148.4.22width	711
7.149multiscale::video::RectangularEntityCsvToInputFilesConverter Class Reference	712
7.149.1 Detailed Description	714
7.149.2 Constructor & Destructor Documentation	714
7.149.2.1 RectangularEntityCsvToInputFilesConverter	714
7.149.2.2 ~RectangularEntityCsvToInputFilesConverter	714
7.149.3 Member Function Documentation	714
7.149.3.1 computeCoordinate	714
7.149.3.2 computeSimulationTime	715
7.149.3.3 convert	716
7.149.3.4 initInputFile	716
7.149.3.5 initIterators	716
7.149.3.6 initOutputFile	716
7.149.3.7 processInputFile	716
7.149.3.8 processLine	717
7.149.3.9 splitLineInCoordinates	717
7.149.3.10validateCoordinate	717

7.149.3.1 validateEntitiesGrid	717
7.149.3.12 validateInput	718
7.149.3.13 validateInputLine	719
7.149.3.14 validateMaxNrOfEntitiesPerPosition	719
7.149.3.15 validateSimulationTime	719
7.149.4 Member Data Documentation	719
7.149.4.1 entitiesIterator	719
7.149.4.2 ERR_INPUT_OPEN	719
7.149.4.3 ERR_INVALID_NR_ENTITIES	720
7.149.4.4 ERR_INVALID_OX_COORDINATE	720
7.149.4.5 ERR_INVALID_OY_COORDINATE	720
7.149.4.6 ERR_INVALID_VALUE_LINE	720
7.149.4.7 ERR_INVALID_VALUE_TOKEN	720
7.149.4.8 ERR_MAX_NR_ENTITIES	720
7.149.4.9 ERR_NEG_SIM_TIME	720
7.149.4.10 ERR_NR_COORDINATES	720
7.149.4.11 height	720
7.149.4.12 INPUT_FILE_SEPARATOR	720
7.149.4.13 outputPath	721
7.149.4.14 maxNrOfEntitiesPerPosition	721
7.149.4.15 nrOfEntities	721
7.149.4.16 OUTPUT_EXTENSION	721
7.149.4.17 OUTPUT_FILE_SEPARATOR	721
7.149.4.18 OUTPUT_SEPARATOR	721
7.149.4.19 outputFilepath	721
7.149.4.20 width	721
7.150 multiscale::video::RectangularGnuplotScriptGenerator Class Reference	722
7.150.1 Detailed Description	723
7.150.2 Member Function Documentation	723
7.150.2.1 generateBody	723
7.150.2.2 generateFooter	724
7.150.2.3 generateHeader	724
7.150.2.4 generateScript	724
7.150.2.5 outputContent	725
7.150.2.6 outputFooter	725
7.150.2.7 outputHeader	725
7.150.3 Member Data Documentation	725
7.150.3.1 CONTENT_IN	725
7.150.3.2 FOOTER_IN	726
7.150.3.3 GNUPLOT_EXTENSION	726

7.150.3.4 HEADER_IN	726
7.150.3.5 OUTPUT_SEPARATOR	726
7.150.3.6 REPLACE_DIMENSION_EXTRA	726
7.150.3.7 REPLACE_HEADER_FILENAME	726
7.150.3.8 REPLACE_HEADER_HEIGHT	726
7.150.3.9 REPLACE_HEADER_SIM_TIME	726
7.150.3.10 REPLACE_HEADER_WIDTH	727
7.151 multiscale::analysis::RectangularMatFactory Class Reference	727
7.151.1 Detailed Description	730
7.151.2 Constructor & Destructor Documentation	730
7.151.2.1 RectangularMatFactory	730
7.151.2.2 ~RectangularMatFactory	730
7.151.3 Member Function Documentation	730
7.151.3.1 createFromViewerImage	730
7.151.3.2 isValidViewerImage	731
7.151.3.3 maxColourBarIntensityFromViewerImage	731
7.151.3.4 processConcentrations	731
7.151.4 Member Data Documentation	732
7.151.4.1 COLOURBAR_MAX_X	732
7.151.4.2 COLOURBAR_MAX_Y	732
7.151.4.3 ERR_CONC	732
7.151.4.4 INPUT_IMG_HEIGHT	732
7.151.4.5 INPUT_IMG_WIDTH	732
7.151.4.6 ROI_HEIGHT	732
7.151.4.7 ROI_START_X	732
7.151.4.8 ROI_START_Y	732
7.151.4.9 ROI_WIDTH	732
7.152 multiscale::verification::Region Class Reference	733
7.152.1 Detailed Description	734
7.153 multiscale::analysis::Region Class Reference	735
7.153.1 Detailed Description	739
7.153.2 Constructor & Destructor Documentation	739
7.153.2.1 Region	739
7.153.2.2 ~Region	739
7.153.3 Member Function Documentation	739
7.153.3.1 areValidInputPolygons	739
7.153.3.2 areValidInputPolygons	739
7.153.3.3 areValidInputValues	740
7.153.3.4 computeArealfOuterBoderDefined	740
7.153.3.5 computeClusterednessDegreelfOuterBorderDefined	740

7.153.3.6 getInnerBorderPolygons	740
7.153.3.7 getOuterBorderPolygon	741
7.153.3.8 isCircularMeasure	741
7.153.3.9 isRectangularMeasure	741
7.153.3.10sTriangularMeasure	741
7.153.3.11isValidInputPolygon	741
7.153.3.12type	741
7.153.3.13updateArea	742
7.153.3.14updateCentrePoint	742
7.153.3.15updateClusterednessDegree	742
7.153.3.16updateDensity	742
7.153.3.17updatePerimeter	742
7.153.3.18validateInputValues	742
7.153.4 Member Data Documentation	743
7.153.4.1 CONTOUR_CLOSED	743
7.153.4.2 CONTOUR_ORIENTED	743
7.153.4.3 innerBorderPolygons	743
7.153.4.4 outerBorderPolygon	743
7.154 multiscale::analysis::RegionDetector Class Reference	744
7.154.1 Detailed Description	749
7.154.2 Constructor & Destructor Documentation	749
7.154.2.1 RegionDetector	749
7.154.2.2 ~RegionDetector	749
7.154.3 Member Function Documentation	749
7.154.3.1 approximatePolygonOuterBorder	749
7.154.3.2 changeContrastAndBrightness	749
7.154.3.3 clearPreviousDetectionResults	750
7.154.3.4 computeAverageClusterednessDegree	750
7.154.3.5 computeAverageDensity	750
7.154.3.6 computeAverageMeasures	750
7.154.3.7 convertAlpha	750
7.154.3.8 convertBeta	751
7.154.3.9 createDetectorSpecificTrackbars	751
7.154.3.10createPolygon	751
7.154.3.11createPolygons	751
7.154.3.12createPolygonsFromContours	752
7.154.3.13createRegionFromPolygon	753
7.154.3.14existContours	753
7.154.3.15findPolygonsInImage	753
7.154.3.16findRegions	753

7.154.3.17getAlpha	754
7.154.3.18getBeta	754
7.154.3.19getBlurKernelSize	754
7.154.3.20getCollectionOfSpatialEntityPseudo3D	754
7.154.3.21getDetectorTypeAsString	754
7.154.3.22getEpsilon	755
7.154.3.23getMorphologicalCloselterations	755
7.154.3.24getOriginXCoordinate	755
7.154.3.25getOriginYCoordinate	755
7.154.3.26getRegionAreaThresh	755
7.154.3.27getRegions	755
7.154.3.28getThresholdValue	755
7.154.3.29initialiseDetectorSpecificFields	756
7.154.3.30initialiseDetectorSpecificImageDependentFields	756
7.154.3.31isValidContour	756
7.154.3.32isValidHole	756
7.154.3.33morphologicalClose	756
7.154.3.34outputRegionInnerBordersToImage	757
7.154.3.35outputRegionOuterBorderToImage	757
7.154.3.36outputRegionToImage	757
7.154.3.37outputResultsToImage	757
7.154.3.38processImageAndDetect	757
7.154.3.39regionDensity	758
7.154.3.40setAlpha	758
7.154.3.41setBeta	758
7.154.3.42setBlurKernelSize	758
7.154.3.43setEpsilon	759
7.154.3.44setMorphologicalCloselterations	759
7.154.3.45setOriginXCoordinate	759
7.154.3.46setOriginYCoordinate	759
7.154.3.47setPolygonInnerContours	760
7.154.3.48setPolygonOuterContour	760
7.154.3.49setRegionAreaThresh	760
7.154.3.50setThresholdValue	760
7.154.3.51smoothImage	761
7.154.3.52sumOfAverageCentroidDistances	761
7.154.3.53thresholdImage	761
7.154.4 Member Data Documentation	761
7.154.4.1 alpha	761
7.154.4.2 ALPHA_MAX	762

7.154.4.3 ALPHA_REAL_MAX	762
7.154.4.4 ALPHA_REAL_MIN	762
7.154.4.5 beta	762
7.154.4.6 BETA_MAX	762
7.154.4.7 BETA_REAL_MAX	762
7.154.4.8 BETA_REAL_MIN	762
7.154.4.9 blurKernelSize	762
7.154.4.10 CANNY_THRESH_MAX	763
7.154.4.11 CONTOUR_AREA_ORIENTED	763
7.154.4.12 DETECTOR_TYPE	763
7.154.4.13 DISPLAY_LINE_THICKNESS	763
7.154.4.14 epsilon	763
7.154.4.15 EPSILON_MAX	763
7.154.4.16 HIERARCHY_FIRST_CHILD_INDEX	763
7.154.4.17 HIERARCHY_NEXT_INDEX	763
7.154.4.18 HIERARCHY_PARENT_INDEX	763
7.154.4.19 HIERARCHY_PREV_INDEX	763
7.154.4.20 INTENSITY_MAX	764
7.154.4.21 KERNEL_MAX	764
7.154.4.22 MORPH_ITER_MAX	764
7.154.4.23 morphologicalCloseIterations	764
7.154.4.24 POLYGON_CLOSED	764
7.154.4.25 REGION_AREA_THRESH_MAX	764
7.154.4.26 regionAreaThresh	764
7.154.4.27 regions	764
7.154.4.28 THRESHOLD_CLUSTEREDNESS	765
7.154.4.29 THRESHOLD_HOLE_AREA	765
7.154.4.30 THRESHOLD_MAX	765
7.154.4.31 thresholdValue	765
7.154.4.32 TRACKBAR_ALPHA	765
7.154.4.33 TRACKBAR_BETA	765
7.154.4.34 TRACKBAR_CANNY	765
7.154.4.35 TRACKBAR_EPSILON	765
7.154.4.36 TRACKBAR_KERNEL	765
7.154.4.37 TRACKBAR_MORPH	766
7.154.4.38 TRACKBAR_REGION_AREA_THRESH	766
7.154.4.39 TRACKBAR_THRESHOLD	766
7.154.4.40 USE_CANNY_L2	766
7.155 multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename > Struct Template Reference	766

7.155.1 Detailed Description	767
7.155.2 Member Typedef Documentation	767
7.155.2.1 type	767
7.156 multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename > Struct Template Reference	767
7.156.1 Detailed Description	768
7.156.2 Member Typedef Documentation	768
7.156.2.1 type	768
7.157 multiscale::RGBColourGenerator Class Reference	768
7.157.1 Detailed Description	769
7.157.2 Member Function Documentation	769
7.157.2.1 computeRGBValues	769
7.157.2.2 convertHSVToRGB	770
7.157.2.3 convertRGBToString	771
7.157.2.4 generate	771
7.157.2.5 generate	771
7.157.3 Member Data Documentation	771
7.157.3.1 blue	771
7.157.3.2 green	772
7.157.3.3 HUE_MAX	772
7.157.3.4 HUE_MIN	772
7.157.3.5 red	772
7.157.3.6 SATURATION	772
7.157.3.7 VALUE	772
7.158 multiscale::RuntimeException Class Reference	772
7.158.1 Detailed Description	775
7.158.2 Constructor & Destructor Documentation	775
7.158.2.1 RuntimeException	775
7.158.2.2 RuntimeException	775
7.158.2.3 RuntimeException	775
7.159 multiscale::verification::SemanticType Class Reference	775
7.159.1 Detailed Description	776
7.159.2 Member Data Documentation	776
7.159.2.1 DEFAULT_VALUE	776
7.160 multiscale::verification::SemanticTypeAttribute Class Reference	777
7.160.1 Detailed Description	778
7.160.2 Constructor & Destructor Documentation	778
7.160.2.1 SemanticTypeAttribute	778
7.160.3 Member Data Documentation	778
7.160.3.1 semanticType	778

7.161 multiscale::verification::SemanticTypeGrammar< Iterator > Class Template Reference	778
7.161.1 Detailed Description	781
7.161.2 Constructor & Destructor Documentation	781
7.161.2.1 SemanticTypeGrammar	781
7.161.3 Member Function Documentation	781
7.161.3.1 assignNamesToRules	781
7.161.3.2 initialise	781
7.161.3.3 initialiseDebugSupport	781
7.161.3.4 initialiseErrorHandlingSupport	781
7.161.3.5 initialiseGrammar	782
7.161.3.6 initialiseRulesDebugging	782
7.161.4 Member Data Documentation	782
7.161.4.1 semanticTypeRule	782
7.161.4.2 semanticTypeStringRule	782
7.162 multiscale::verification::SemanticTypeStringGrammar< Iterator > Class Template Reference	782
7.162.1 Detailed Description	785
7.162.2 Constructor & Destructor Documentation	785
7.162.2.1 SemanticTypeStringGrammar	785
7.162.3 Member Function Documentation	785
7.162.3.1 assignNamesToRules	785
7.162.3.2 initialise	785
7.162.3.3 initialiseDebugSupport	786
7.162.3.4 initialiseErrorHandlingSupport	786
7.162.3.5 initialiseGrammar	786
7.162.3.6 initialiseRulesDebugging	786
7.162.4 Member Data Documentation	786
7.162.4.1 SEMANTIC_CRITERION_STRING_PATTERN	786
7.162.4.2 semanticTypeStringRule	786
7.163 multiscale::analysis::Silhouette Class Reference	786
7.163.1 Detailed Description	788
7.163.2 Member Function Documentation	788
7.163.2.1 computeAverageDissimilarityBtwEntityAndCluster	788
7.163.2.2 computeAverageDissimilarityToOtherClusters	788
7.163.2.3 computeAverageDissimilarityWithinCluster	788
7.163.2.4 computeAverageMeasure	789
7.163.2.5 computeMeasure	789
7.163.2.6 computeOverallAverageMeasure	789
7.163.2.7 validateClusterIndex	789
7.163.2.8 validateElementIndex	791
7.163.2.9 validateEntityIndex	791

7.164multiscale::verification::SimilarityMeasureAttribute Class Reference	791
7.164.1 Detailed Description	792
7.164.2 Member Data Documentation	792
7.164.2.1 similarityMeasure	792
7.165multiscale::verification::SimilarityMeasureTypeParser Struct Reference	792
7.165.1 Detailed Description	794
7.165.2 Constructor & Destructor Documentation	794
7.165.2.1 SimilarityMeasureTypeParser	794
7.166multiscale::verification::SimilarityTemporalNumericCollectionAttribute Class Reference	794
7.166.1 Detailed Description	795
7.166.2 Member Data Documentation	795
7.166.2.1 lhsTemporalNumericCollection	795
7.166.2.2 rhsTemporalNumericCollection	795
7.166.2.3 similarityMeasure	795
7.166.2.4 toleratedSimilarityDifference	795
7.167multiscale::analysis::SimulationClusterDetector Class Reference	795
7.167.1 Detailed Description	798
7.167.2 Constructor & Destructor Documentation	799
7.167.2.1 SimulationClusterDetector	799
7.167.2.2 ~SimulationClusterDetector	799
7.167.3 Member Function Documentation	799
7.167.3.1 computePileUpDegreeAtPosition	799
7.167.3.2 detectEntitiesInImage	799
7.167.3.3 getEntityCentrePoint	800
7.167.3.4 getEntityContourPoints	801
7.167.3.5 initialiseDetectorSpecificImageDependentFields	801
7.167.3.6 initialiseThresholdedImage	801
7.167.3.7 isEntityAtPosition	801
7.167.3.8 outputClusterCircularShape	802
7.167.3.9 outputClusterRectangularShape	803
7.167.3.10outputClusterShape	803
7.167.3.11outputClusterTolImage	803
7.167.3.12outputClusterTriangularShape	804
7.167.3.13outputResultsTolImage	804
7.167.4 Member Data Documentation	804
7.167.4.1 DATAPOINT_THICKNESS	804
7.167.4.2 DATAPOINT_WIDTH	804
7.167.4.3 ENTITY_THRESH	804
7.167.4.4 entityHeight	805
7.167.4.5 entityWidth	805

7.167.4.6 height	805
7.167.4.7 THRESHOLD	805
7.167.4.8 THRESHOLD_MAX	805
7.167.4.9 thresholdedImage	805
7.167.4.10width	805
7.168multiscaletest::SpatialEntitiesTraceTest Class Reference	806
7.168.1 Detailed Description	809
7.168.2 Member Function Documentation	809
7.168.2.1 InitialiseTrace	809
7.168.3 Member Data Documentation	809
7.168.3.1 clustersClusterednessMaxValue	809
7.168.3.2 clustersClusterednessMinValue	809
7.169multiscale::verification::SpatialEntity Class Reference	809
7.169.1 Detailed Description	812
7.169.2 Constructor & Destructor Documentation	812
7.169.2.1 SpatialEntity	812
7.169.2.2 ~SpatialEntity	812
7.169.3 Member Function Documentation	813
7.169.3.1 getSemanticType	813
7.169.3.2 getSpatialMeasureValue	813
7.169.3.3 operator<	813
7.169.3.4 setSemanticType	813
7.169.3.5 setSpatialMeasureValue	813
7.169.3.6 toString	814
7.169.3.7 validateSpatialMeasureValue	814
7.169.4 Member Data Documentation	814
7.169.4.1 ERR_INVALID_SPATIAL_MEASURE_BEGIN	814
7.169.4.2 ERR_INVALID_SPATIAL_MEASURE_END	814
7.169.4.3 ERR_INVALID_SPATIAL_MEASURE_MIDDLE	814
7.169.4.4 OUTPUT_SPATIAL_MEASURE_VALUE_SEPARATOR	814
7.169.4.5 semanticType	815
7.169.4.6 spatialMeasureValues	815
7.170multiscale::analysis::SpatialEntityPseudo3D Class Reference	815
7.170.1 Detailed Description	820
7.170.2 Constructor & Destructor Documentation	820
7.170.2.1 SpatialEntityPseudo3D	820
7.170.2.2 ~SpatialEntityPseudo3D	820
7.170.3 Member Function Documentation	820
7.170.3.1 convertPoints	820
7.170.3.2 fieldNamesToString	820

7.170.3.3 fieldValuesToString	820
7.170.3.4 getAngle	820
7.170.3.5 getArea	821
7.170.3.6 getCentre	821
7.170.3.7 getCircularMeasure	821
7.170.3.8 getClusterednessDegree	821
7.170.3.9 getDensity	821
7.170.3.10 getDistanceFromOrigin	821
7.170.3.11 getPerimeter	822
7.170.3.12 getRectangularMeasure	822
7.170.3.13 getShape	822
7.170.3.14 getShapeAsString	822
7.170.3.15 getTriangularMeasure	822
7.170.3.16 initialise	822
7.170.3.17 isCircularMeasure	822
7.170.3.18 isRectangularMeasure	823
7.170.3.19 isTriangularMeasure	823
7.170.3.20 normalisedShapeMeasure	823
7.170.3.21 shapeAsString	823
7.170.3.22 toString	823
7.170.3.23 type	824
7.170.3.24 typeAsString	824
7.170.3.25 updateArea	824
7.170.3.26 updateCentrePoint	824
7.170.3.27 updateClusterednessDegree	824
7.170.3.28 updateDensity	824
7.170.3.29 updateMeasures	824
7.170.3.30 updateMeasuresIfRequired	825
7.170.3.31 updatePerimeter	825
7.170.3.32 updateShape	825
7.170.4 Member Data Documentation	825
7.170.4.1 angle	825
7.170.4.2 area	825
7.170.4.3 centre	826
7.170.4.4 circularMeasure	826
7.170.4.5 clusterednessDegree	826
7.170.4.6 CONVEX_HULL_CLOCKWISE	826
7.170.4.7 density	826
7.170.4.8 distanceFromOrigin	826
7.170.4.9 ERR_INPUT	826

7.170.4.10ERR_UNDEFINED_TYPE	827
7.170.4.11OUTPUT_SEPARATOR	827
7.170.4.12perimeter	827
7.170.4.13rectangularMeasure	827
7.170.4.14shape	827
7.170.4.15STR_CIRCLE	827
7.170.4.16STR_CLUSTER	827
7.170.4.17STR_RECTANGLE	827
7.170.4.18STR_REGION	828
7.170.4.19STR_TRIANGLE	828
7.170.4.20STR_UNDEFINED	828
7.170.4.21triangularMeasure	828
7.170.4.22updateFlag	828
7.171 multiscale::verification::SpatialMeasureAttribute Class Reference	828
7.171.1 Detailed Description	829
7.171.2 Member Data Documentation	829
7.171.2.1 spatialMeasureType	829
7.172 multiscale::verification::SpatialMeasureCollectionAttribute Class Reference	829
7.172.1 Detailed Description	830
7.172.2 Member Data Documentation	830
7.172.2.1 spatialMeasure	830
7.172.2.2 subset	830
7.173 multiscale::verification::SpatialMeasureCollectionGrammar< Iterator > Class Template Reference .	831
7.173.1 Detailed Description	836
7.173.2 Constructor & Destructor Documentation	836
7.173.2.1 SpatialMeasureCollectionGrammar	836
7.173.3 Member Function Documentation	837
7.173.3.1 assignNamesToComposedConstraintRules	837
7.173.3.2 assignNamesToConstraintRules	837
7.173.3.3 assignNamesToConstraintsRules	837
7.173.3.4 assignNamesToFilterNumericMeasureRules	837
7.173.3.5 assignNamesToNumericSpatialMeasureRules	837
7.173.3.6 assignNamesToPrimaryConstraintRules	837
7.173.3.7 assignNamesToRules	837
7.173.3.8 assignNamesToSpatialMeasureCollectionRules	837
7.173.3.9 assignNamesToSpatialMeasureRules	838
7.173.3.10assignNamesToSubsetRules	838
7.173.3.11initialise	838
7.173.3.12initialiseComposedConstraintErrorHandlingSupport	838
7.173.3.13initialiseComposedConstraintRule	838

7.173.3.14initialiseComposedConstraintRuleDebugging	838
7.173.3.15initialiseConstraintRule	838
7.173.3.16initialiseConstraintRuleDebugging	838
7.173.3.17initialiseConstraintsErrorHandlingSupport	839
7.173.3.18initialiseConstraintsRules	839
7.173.3.19initialiseConstraintsRulesDebugging	839
7.173.3.20initialiseDebugSupport	839
7.173.3.21initialiseErrorHandlingSupport	839
7.173.3.22initialiseFilterNumericMeasureErrorHandlingSupport	839
7.173.3.23initialiseFilterNumericMeasureRule	839
7.173.3.24initialiseFilterNumericMeasureRuleDebugging	839
7.173.3.25initialiseGrammar	840
7.173.3.26initialisePrimaryConstraintErrorHandlingSupport	840
7.173.3.27initialisePrimaryConstraintRule	840
7.173.3.28initialisePrimaryConstraintRuleDebugging	840
7.173.3.29initialiseRulesDebugging	840
7.173.3.30initialiseSpatialMeasureCollectionErrorHandlingSupport	840
7.173.3.31initialiseSpatialMeasureCollectionRule	840
7.173.3.32initialiseSpatialMeasureCollectionRuleDebugging	841
7.173.3.33initialiseSpatialMeasureRule	841
7.173.3.34initialiseSpatialMeasureRuleDebugging	841
7.173.3.35initialiseSubsetErrorHandlingSupport	841
7.173.3.36initialiseSubsetRule	841
7.173.3.37initialiseSubsetRuleDebugging	841
7.173.4 Member Data Documentation	841
7.173.4.1 andConstraintRule	841
7.173.4.2 binaryNumericFilterRule	842
7.173.4.3 binaryNumericMeasureRule	842
7.173.4.4 comparatorRule	842
7.173.4.5 constraintRule	842
7.173.4.6 equivalenceConstraintRule	842
7.173.4.7 filterNumericMeasureRule	842
7.173.4.8 filterSubsetRule	842
7.173.4.9 implicationConstraintRule	843
7.173.4.10notConstraintRule	843
7.173.4.11orConstraintRule	843
7.173.4.12primaryConstraintRule	843
7.173.4.13primaryNumericMeasureRule	843
7.173.4.14semanticTypeRule	843
7.173.4.15spatialMeasureCollectionRule	843

7.173.4.16spatialMeasureRule	844
7.173.4.17spatialMeasureTypeParser	844
7.173.4.18subsetOperationTypeParser	844
7.173.4.19subsetRule	844
7.173.4.20subsetSpecificRule	844
7.173.4.21subsetSpecificTypeParser	844
7.173.4.22subsetSubsetOperationRule	844
7.173.4.23unaryNumericFilterRule	845
7.173.4.24unaryNumericMeasureRule	845
7.173.4.25unarySpatialConstraintRule	845
7.173.4.26unaryTypeConstraintRule	845
7.174multiscale::verification::SpatialMeasureEvaluator Class Reference	845
7.174.1 Detailed Description	846
7.174.2 Member Function Documentation	846
7.174.2.1 evaluate	846
7.175multiscale::verification::SpatialMeasureTypeParser Struct Reference	846
7.175.1 Detailed Description	848
7.175.2 Constructor & Destructor Documentation	848
7.175.2.1 SpatialMeasureTypeParser	848
7.176multiscale::verification::SpatialNumericComparisonAttribute Class Reference	848
7.176.1 Detailed Description	849
7.176.2 Member Data Documentation	849
7.176.2.1 comparator	849
7.176.2.2 numericMeasure	849
7.176.2.3 spatialMeasure	849
7.177multiscale::verification::SpatialTemporalDataReader Class Reference	849
7.177.1 Detailed Description	852
7.177.2 Constructor & Destructor Documentation	853
7.177.2.1 SpatialTemporalDataReader	853
7.177.2.2 ~SpatialTemporalDataReader	853
7.177.3 Member Function Documentation	853
7.177.3.1 addEntitiesToTimePoint	853
7.177.3.2 addNumericStateVariableToTimePoint	853
7.177.3.3 addSpatialEntityToTimePoint	853
7.177.3.4 addTimePointToTrace	854
7.177.3.5 clearInputFilesSets	854
7.177.3.6 constructSpatialTemporalTrace	854
7.177.3.7 constructSpatialTemporalTrace	854
7.177.3.8 convertTimePointPropertyTreeToTrace	855
7.177.3.9 createDerivedSpatialEntity	856

7.177.3.10generateSpatialTemporalTrace	856
7.177.3.11generateSpatialTemporalTrace	856
7.177.3.12getFilesInFolder	856
7.177.3.13getNextSpatialTemporalTrace	857
7.177.3.14getNextSpatialTemporalTrace	857
7.177.3.15getRandomUnprocessedInputFile	857
7.177.3.16getRandomValidUnprocessedInputFilepath	857
7.177.3.17hasNext	857
7.177.3.18isValidNext	858
7.177.3.19initialise	858
7.177.3.20initialise	858
7.177.3.21isValidInputFile	858
7.177.3.22processInvalidInputFile	858
7.177.3.23processValidInputFile	859
7.177.3.24refresh	859
7.177.3.25setSpatialEntityMeasureValues	859
7.177.3.26setSpatialEntitySemanticTypeValue	859
7.177.3.27setTimePointValue	860
7.177.3.28timePointHasValue	860
7.177.3.29updateInputFilesSets	860
7.177.3.30validateFolderPath	860
7.177.4 Member Data Documentation	861
7.177.4.1 ERR_INVALID_FOLDER_PATH	861
7.177.4.2 ERR_NO_VALID_INPUT_FILES_REMAINING	861
7.177.4.3 ERR_UNDEFINED_SPATIAL_ENTITY_TYPE	861
7.177.4.4 folderPath	861
7.177.4.5 INPUT_FILES_EXTENSION	861
7.177.4.6 INPUT_FILES_SCHEMA_PATH	861
7.177.4.7 LABEL_EXPERIMENT	861
7.177.4.8 LABEL_NUMERIC_STATE_VARIABLE	862
7.177.4.9 LABEL_NUMERIC_STATE_VARIABLE_NAME	862
7.177.4.10LABEL_NUMERIC_STATE_VARIABLE_SEMANTIC_TYPE	862
7.177.4.11LABEL_NUMERIC_STATE_VARIABLE_VALUE	862
7.177.4.12LABEL_SPATIAL_ENTITY	862
7.177.4.13LABEL_SPATIAL_ENTITY_SEMANTIC_TYPE	862
7.177.4.14LABEL_SPATIAL_ENTITY_SPATIAL_TYPE	862
7.177.4.15LABEL_TIMEPOINT_VALUE	862
7.177.4.16processedInputFiles	863
7.177.4.17unprocessedInputFiles	863
7.178 multiscale::verification::SpatialTemporalException Class Reference	863

7.178.1 Detailed Description	866
7.178.2 Constructor & Destructor Documentation	866
7.178.2.1 SpatialTemporalException	866
7.178.2.2 SpatialTemporalException	866
7.179multiscale::verification::SpatialTemporalTrace Class Reference	866
7.179.1 Detailed Description	869
7.179.2 Constructor & Destructor Documentation	869
7.179.2.1 SpatialTemporalTrace	869
7.179.2.2 SpatialTemporalTrace	869
7.179.2.3 ~SpatialTemporalTrace	869
7.179.3 Member Function Documentation	869
7.179.3.1 addTimePoint	869
7.179.3.2 addTimePointsToSubTrace	870
7.179.3.3 clear	870
7.179.3.4 getTimePoint	870
7.179.3.5 indexOfFirstTimePointGreaterOrEqualToValue	870
7.179.3.6 initialise	871
7.179.3.7 length	871
7.179.3.8 nextTimePointValue	871
7.179.3.9 nextTimePointValueForLastTimePoint	871
7.179.3.10resetSubTracestartIndex	871
7.179.3.11setSubTrace	872
7.179.3.12setSubTraceIndex	872
7.179.3.13subTrace	872
7.179.3.14updateLastTimePointValue	872
7.179.3.15validateIndex	873
7.179.3.16validateTimePointValue	873
7.179.3.17validateTimePointValue	873
7.179.3.18validateValue	873
7.179.4 Member Data Documentation	874
7.179.4.1 beginIndex	874
7.179.4.2 ERR_ITERATOR_NEXT	874
7.179.4.3 ERR_NEXT_TIMEPOINT_VALUE_NOT_EXISTS	874
7.179.4.4 ERR_TIMEPOINT_END_END	874
7.179.4.5 ERR_TIMEPOINT_END_MIDDLE	874
7.179.4.6 ERR_TIMEPOINT_END_START	874
7.179.4.7 ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_END	874
7.179.4.8 ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_START	875
7.179.4.9 ERR_TIMEPOINT_VALUE_INVALID_END	875
7.179.4.10ERR_TIMEPOINT_VALUE_INVALID_MIDDLE	875

7.179.4.11ERR_TIMEPOINT_VALUE_INVALID_START	875
7.179.4.12ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_END	875
7.179.4.13ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_START	875
7.179.4.14sLastTimePointValueInitialised	875
7.179.4.15lastTimePointValue	875
7.179.4.16TIMEPOINT_INDEX_NOT_FOUND	876
7.179.4.17timePoints	876
7.180multiscale::StandardNumberIterator Class Reference	876
7.180.1 Detailed Description	879
7.180.2 Constructor & Destructor Documentation	879
7.180.2.1 StandardNumberIterator	879
7.180.2.2 ~StandardNumberIterator	879
7.180.3 Member Function Documentation	879
7.180.3.1 hasNextInitialised	879
7.180.3.2 initialise	879
7.180.3.3 number	879
7.180.3.4 resetCurrentNumber	880
7.180.4 Member Data Documentation	880
7.180.4.1 currentNumber	880
7.181multiscale::verification::StateVariableAttribute Class Reference	880
7.181.1 Detailed Description	881
7.181.2 Member Data Documentation	881
7.181.2.1 name	881
7.182multiscale::verification::StatisticalModelChecker Class Reference	882
7.182.1 Detailed Description	886
7.182.2 Constructor & Destructor Documentation	887
7.182.2.1 StatisticalModelChecker	887
7.182.2.2 ~StatisticalModelChecker	887
7.182.3 Member Function Documentation	887
7.182.3.1 acceptsMoreTraces	887
7.182.3.2 computeFPrimeValue	887
7.182.3.3 computeFPrimeValueFirstTerm	887
7.182.3.4 computeFPrimeValueSecondTerm	888
7.182.3.5 computeFValue	888
7.182.3.6 computeFValueFirstTerm	888
7.182.3.7 computeFValueSecondTerm	888
7.182.3.8 computeIndifferenceIntervalHalf	888
7.182.3.9 doesPropertyHold	889
7.182.3.10doesPropertyHoldConsideringProbabilityComparator	889
7.182.3.11doesPropertyHoldConsideringResult	889

7.182.3.12getDetailedResults	889
7.182.3.13getDetailedUpdatedResults	890
7.182.3.14initialise	890
7.182.3.15sValidTypeError	890
7.182.3.16requiresMoreTraces	890
7.182.3.17updateDerivedModelCheckerForFalseEvaluation	890
7.182.3.18updateDerivedModelCheckerForTrueEvaluation	891
7.182.3.19updateInitialisedModelCheckingResult	891
7.182.3.20updateModelCheckingResult	891
7.182.3.21updateModelCheckingResult	891
7.182.3.22updateModelCheckingResultEnoughTraces	891
7.182.3.23updateModelCheckingResultNotEnoughTraces	892
7.182.3.24validateTypesErrors	892
7.182.4 Member Data Documentation	892
7.182.4.1 a1FromPaper	892
7.182.4.2 a2FromPaper	892
7.182.4.3 b1FromPaper	892
7.182.4.4 b2FromPaper	893
7.182.4.5 ERR_TYPES_ERROR_VALUES_BEGIN	893
7.182.4.6 ERR_TYPES_ERROR_VALUES_END	893
7.182.4.7 ERR_TYPES_ERROR_VALUES_MIDDLE	893
7.182.4.8 ERR_UNEXPECTED_MODEL_CHECKING_RESULT	893
7.182.4.9 INDIFFERENCE_INTERVAL_HALF_K	893
7.182.4.10ndifferenceIntervalHalf	893
7.182.4.11LOGARITHM_ZERO_VALUE	893
7.182.4.12minTypesError	894
7.182.4.13modelCheckingResult	894
7.182.4.14MSG_OUTPUT_MORE_TRACES_REQUIRED	894
7.182.4.15MSG_OUTPUT_RESULT_BEGIN	894
7.182.4.16MSG_OUTPUT_RESULT_END	894
7.182.4.17MSG_OUTPUT_RESULT_MIDDLE	894
7.182.4.18MSG_OUTPUT_SEPARATOR	895
7.182.4.19probability	895
7.182.4.20typeIError	895
7.182.4.21typeIIError	895
7.183multiscale::verification::StatisticalModelCheckerFactory Class Reference	895
7.183.1 Detailed Description	897
7.183.2 Constructor & Destructor Documentation	897
7.183.2.1 StatisticalModelCheckerFactory	898
7.183.2.2 ~StatisticalModelCheckerFactory	898

7.183.3 Member Function Documentation	898
7.183.3.1 createInstance	898
7.183.4 Member Data Documentation	898
7.183.4.1 typeIError	898
7.183.4.2 typeIIError	898
7.184 multiscaletest::StatisticalModelCheckerTest Class Reference	898
7.184.1 Detailed Description	901
7.184.2 Constructor & Destructor Documentation	901
7.184.2.1 StatisticalModelCheckerTest	901
7.184.3 Member Function Documentation	901
7.184.3.1 InitialiseModelChecker	901
7.184.3.2 SetTypeIError	901
7.184.3.3 SetTypeIIError	901
7.184.4 Member Data Documentation	902
7.184.4.1 typeIError	902
7.184.4.2 typeIIError	902
7.185 multiscale::StringManipulator Class Reference	902
7.185.1 Detailed Description	904
7.185.2 Member Function Documentation	904
7.185.2.1 convert	904
7.185.2.2 count	904
7.185.2.3 filenameFromPath	905
7.185.2.4 replace	906
7.185.2.5 split	906
7.185.2.6 toString	906
7.185.2.7 trimRight	907
7.185.2.8 trimRight	907
7.185.3 Member Data Documentation	907
7.185.3.1 DIR_SEPARATOR	907
7.185.3.2 ERR_INVALID_CONVERSION_BEGIN	907
7.185.3.3 ERR_INVALID_CONVERSION_END	907
7.185.3.4 ERR_INVALID_CONVERSION_MIDDLE	908
7.186 multiscale::verification::SubsetAttribute Class Reference	908
7.186.1 Detailed Description	908
7.186.2 Member Data Documentation	908
7.186.2.1 subset	908
7.187 multiscale::verification::SubsetOperationAttribute Class Reference	909
7.187.1 Detailed Description	909
7.187.2 Member Data Documentation	909
7.187.2.1 subsetOperationType	909

7.188 multiscale::verification::SubsetOperationTypeParser Struct Reference	909
7.188.1 Detailed Description	911
7.188.2 Constructor & Destructor Documentation	911
7.188.2.1 SubsetOperationTypeParser	911
7.189 multiscale::verification::SubsetSpecificAttribute Class Reference	911
7.189.1 Detailed Description	911
7.189.2 Member Data Documentation	912
7.189.2.1 subsetSpecificType	912
7.190 multiscale::verification::SubsetSpecificTypeParser Struct Reference	912
7.190.1 Detailed Description	913
7.190.2 Constructor & Destructor Documentation	913
7.190.2.1 SubsetSpecificTypeParser	913
7.191 multiscale::verification::SubsetSubsetOperationAttribute Class Reference	913
7.191.1 Detailed Description	914
7.191.2 Member Data Documentation	914
7.191.2.1 firstSubset	914
7.191.2.2 secondSubset	914
7.191.2.3 subsetOperation	915
7.192 multiscale::verification::SubsetVisitor Class Reference	915
7.192.1 Detailed Description	917
7.192.2 Constructor & Destructor Documentation	917
7.192.2.1 SubsetVisitor	917
7.192.3 Member Function Documentation	917
7.192.3.1 evaluate	917
7.192.3.2 evaluateSubsetOperation	918
7.192.3.3 filterTimePoint	919
7.192.3.4 operator()	919
7.192.3.5 operator()	919
7.192.3.6 operator()	920
7.192.3.7 operator()	921
7.192.3.8 setTimePointConsideredSpatialEntityType	921
7.192.4 Member Data Documentation	921
7.192.4.1 timePoint	921
7.192.4.2 typeSemanticsTable	921
7.193 multiscale::SubtractionOperation Class Reference	922
7.193.1 Detailed Description	922
7.193.2 Member Function Documentation	922
7.193.2.1 operator()	922
7.194 multiscale::verification::TemporalDataReader Class Reference	922
7.194.1 Detailed Description	925

7.194.2 Constructor & Destructor Documentation	925
7.194.2.1 TemporalDataReader	925
7.194.3 Member Function Documentation	925
7.194.3.1 addNumericStateVariablesToTimePoint	925
7.194.3.2 createTimePointFromTokens	925
7.194.3.3 processLineTokens	925
7.194.3.4 read	926
7.194.3.5 readFromFile	926
7.194.3.6 readFromValidOpenedInputFile	926
7.194.3.7 readInputFileContents	926
7.194.3.8 readInputFileHeader	927
7.194.3.9 setTimePointValue	927
7.194.3.10 validateObservableVariables	927
7.194.4 Member Data Documentation	927
7.194.4.1 lineNumber	927
7.194.4.2 ERR_EMPTY_OBSERVABLE_VARIABLE_NAME	928
7.194.4.3 ERR_INVALID_INPUT_FILE_PATH_BEGIN	928
7.194.4.4 ERR_INVALID_INPUT_FILE_PATH_END	928
7.194.4.5 ERR_INVALID_NR_LINE_TOKENS_BEGIN	928
7.194.4.6 ERR_INVALID_NR_LINE_TOKENS_END	928
7.194.4.7 ERR_INVALID_NR_LINE_TOKENS_MIDDLE	928
7.194.4.8 ERR_INVALID_NR_OBSERVABLE_VARIABLES_BEGIN	928
7.194.4.9 ERR_INVALID_NR_OBSERVABLE_VARIABLES_END	929
7.194.4.10 ERR_OPEN_INPUT_FILE_BEGIN	929
7.194.4.11 ERR_OPEN_INPUT_FILE_END	929
7.194.4.12 filePath	929
7.194.4.13 INPUT_FILE_DELIMITER	929
7.194.4.14 INPUT_FILE_EXTENSION	929
7.194.4.15 observableVariables	929
7.195 multiscale::verification::TemporalNumericCollectionAttribute Class Reference	930
7.195.1 Detailed Description	930
7.195.2 Member Data Documentation	930
7.195.2.1 temporalNumericCollection	930
7.196 multiscale::verification::TemporalNumericCollectionGrammar< Iterator > Class Template Reference	930
7.196.1 Detailed Description	934
7.196.2 Constructor & Destructor Documentation	934
7.196.2.1 TemporalNumericCollectionGrammar	934
7.196.3 Member Function Documentation	935
7.196.3.1 assignNamesToNumericMeasureRules	935
7.196.3.2 assignNamesToRules	935

7.196.3.3 assignNamesToTemporalNumericCollectionRules	935
7.196.3.4 assignNamesToTimeseriesComponentRules	935
7.196.3.5 assignNamesToTimeseriesMeasureRules	935
7.196.3.6 initialise	935
7.196.3.7 initialiseDebugSupport	935
7.196.3.8 initialiseErrorHandlingSupport	936
7.196.3.9 initialiseGrammar	936
7.196.3.10initialiseNumericMeasureErrorHandlingSupport	936
7.196.3.11initialiseNumericMeasureRule	936
7.196.3.12initialiseNumericMeasureRuleDebugging	936
7.196.3.13initialiseRulesDebugging	936
7.196.3.14initialiseTemporalNumericCollectionErrorHandlingSupport	936
7.196.3.15initialiseTemporalNumericCollectionRule	936
7.196.3.16initialiseTemporalNumericCollectionRuleDebugging	937
7.196.3.17initialiseTimeseriesComponentRule	937
7.196.3.18initialiseTimeseriesComponentRuleDebugging	937
7.196.3.19initialiseTimeseriesMeasureRule	937
7.196.3.20initialiseTimeseriesMeasureRuleDebugging	937
7.196.4 Member Data Documentation	937
7.196.4.1 binaryNumericMeasureRule	937
7.196.4.2 binaryNumericNumericRule	937
7.196.4.3 changeMeasureRule	938
7.196.4.4 changeTemporalNumericCollectionRule	938
7.196.4.5 heterogeneousTimeseriesComponentRule	938
7.196.4.6 heterogeneousTimeseriesComponentTypeParser	938
7.196.4.7 homogeneousHomogeneousTimeseriesRule	938
7.196.4.8 homogeneousTimeseriesComponentRule	938
7.196.4.9 homogeneousTimeseriesComponentTypeParser	938
7.196.4.10homogeneousTimeseriesMeasureRule	939
7.196.4.11homogeneousTimeseriesMeasureTypeParser	939
7.196.4.12numericMeasureRule	939
7.196.4.13primaryNumericMeasureRule	939
7.196.4.14spatialMeasureCollectionGrammar	939
7.196.4.15temporalNumericCollectionRule	939
7.196.4.16temporalNumericMeasureCollectionRule	940
7.196.4.17timeseriesComponentRule	940
7.196.4.18timeseriesMeasureRule	940
7.196.4.19timeseriesMeasureTypeParser	940
7.196.4.20timeseriesTimeseriesComponentRule	940
7.196.4.21unaryNumericMeasureRule	940

7.196.4.2 <code>UnaryNumericNumericRule</code>	940
7.197 <code>multiscale::verification::TemporalNumericComparisonAttribute</code> Class Reference	941
7.197.1 Detailed Description	941
7.197.2 Member Data Documentation	941
7.197.2.1 <code>comparator</code>	942
7.197.2.2 <code>lhsTemporalNumericMeasure</code>	942
7.197.2.3 <code>rhsTemporalNumericMeasure</code>	942
7.198 <code>multiscale::verification::TemporalNumericMeasureAttribute</code> Class Reference	942
7.198.1 Detailed Description	943
7.198.2 Member Data Documentation	943
7.198.2.1 <code>temporalNumericMeasure</code>	943
7.199 <code>multiscale::verification::TemporalNumericMeasureCollectionAttribute</code> Class Reference	943
7.199.1 Detailed Description	944
7.199.2 Member Data Documentation	944
7.199.2.1 <code>endTimepoint</code>	944
7.199.2.2 <code>numericMeasure</code>	944
7.199.2.3 <code>startTimepoint</code>	944
7.200 <code>multiscale::verification::TemporalNumericMeasureGrammar< Iterator ></code> Class Template Reference	944
7.200.1 Detailed Description	948
7.200.2 Constructor & Destructor Documentation	948
7.200.2.1 <code>TemporalNumericMeasureGrammar</code>	948
7.200.3 Member Function Documentation	948
7.200.3.1 <code>assignNamesToNumericMeasureCollectionRules</code>	948
7.200.3.2 <code>assignNamesToNumericStatisticalMeasureRules</code>	948
7.200.3.3 <code>assignNamesToRules</code>	949
7.200.3.4 <code>assignNamesToTemporalNumericMeasureRules</code>	949
7.200.3.5 <code>initialise</code>	949
7.200.3.6 <code>initialiseDebugSupport</code>	949
7.200.3.7 <code>initialiseErrorHandlingSupport</code>	949
7.200.3.8 <code>initialiseGrammar</code>	949
7.200.3.9 <code>initialiseNumericMeasureCollectionRule</code>	949
7.200.3.10 <code>initialiseNumericMeasureCollectionRuleDebugging</code>	949
7.200.3.11 <code>initialiseNumericStatisticalMeasureErrorHandlingSupport</code>	950
7.200.3.12 <code>initialiseNumericStatisticalMeasureRule</code>	950
7.200.3.13 <code>initialiseNumericStatisticalMeasureRuleDebugging</code>	950
7.200.3.14 <code>initialiseRulesDebugging</code>	950
7.200.3.15 <code>initialiseTemporalNumericMeasureErrorHandlingSupport</code>	950
7.200.3.16 <code>initialiseTemporalNumericMeasureRule</code>	950
7.200.3.17 <code>initialiseTemporalNumericMeasureRuleDebugging</code>	950
7.200.4 Member Data Documentation	950

7.200.4.1 binaryNumericMeasureRule	951
7.200.4.2 binaryNumericTemporalRule	951
7.200.4.3 binaryStatisticalMeasureRule	951
7.200.4.4 binaryStatisticalNumericRule	951
7.200.4.5 binaryStatisticalQuantileMeasureRule	951
7.200.4.6 binaryStatisticalQuantileNumericRule	951
7.200.4.7 numericMeasureCollectionRule	951
7.200.4.8 numericStateVariableRule	952
7.200.4.9 numericStatisticalMeasureRule	952
7.200.4.10 spatialMeasureCollectionRule	952
7.200.4.11 temporalNumericCollectionRule	952
7.200.4.12 temporalNumericMeasureRule	952
7.200.4.13 unaryNumericMeasureRule	952
7.200.4.14 unaryNumericTemporalRule	952
7.200.4.15 unaryStatisticalMeasureRule	953
7.200.4.16 unaryStatisticalNumericRule	953
7.201 multiscale::verification::TemporalNumericVisitor Class Reference	953
7.201.1 Detailed Description	956
7.201.2 Constructor & Destructor Documentation	956
7.201.2.1 TemporalNumericVisitor	956
7.201.3 Member Function Documentation	957
7.201.3.1 evaluate	957
7.201.3.2 evaluateNumericMeasureCollection	958
7.201.3.3 evaluateNumericStatisticalMeasure	958
7.201.3.4 operator()	958
7.201.3.5 operator()	959
7.201.3.6 operator()	960
7.201.3.7 operator()	960
7.201.3.8 operator()	960
7.201.3.9 operator()	960
7.201.3.10 operator()	961
7.201.3.11 operator()	961
7.201.3.12 operator()	961
7.201.4 Member Data Documentation	962
7.201.4.1 trace	962
7.201.4.2 typeSemanticsTable	962
7.202 multiscale::TestException Class Reference	962
7.202.1 Detailed Description	965
7.202.2 Constructor & Destructor Documentation	965
7.202.2.1 TestException	965

7.202.2.2 TestException	965
7.202.2.3 TestException	965
7.203multiscale::verification::TimePoint Class Reference	965
7.203.1 Detailed Description	968
7.203.2 Constructor & Destructor Documentation	968
7.203.2.1 TimePoint	968
7.203.2.2 TimePoint	969
7.203.2.3 ~TimePoint	969
7.203.3 Member Function Documentation	969
7.203.3.1 addNumericStateVariable	969
7.203.3.2 addSpatialEntity	969
7.203.3.3 existsNumericStateVariable	969
7.203.3.4 getConsideredSpatialEntities	970
7.203.3.5 getConsideredSpatialEntityTypes	970
7.203.3.6 getNumericStateVariable	970
7.203.3.7 getNumericStateVariablesBeginIterator	970
7.203.3.8 getNumericStateVariablesBeginIterator	970
7.203.3.9 getNumericStateVariablesEndIterator	970
7.203.3.10 getNumericStateVariablesEndIterator	971
7.203.3.11 getSpatialEntitiesBeginIterator	971
7.203.3.12 getSpatialEntitiesBeginIterator	971
7.203.3.13 getSpatialEntitiesEndIterator	971
7.203.3.14 getSpatialEntitiesEndIterator	972
7.203.3.15 getValue	972
7.203.3.16 number_of_SpatialEntities	972
7.203.3.17 removeSpatialEntity	972
7.203.3.18 setConsideredSpatialEntityType	973
7.203.3.19 setValue	973
7.203.3.20 spatialEntitiesSetOperation	973
7.203.3.21 timePointDifference	974
7.203.3.22 timePointIntersection	974
7.203.3.23 timePointSetOperation	974
7.203.3.24 timePointUnion	974
7.203.3.25 updateConsideredSpatialEntityTypes	975
7.203.3.26 updateSpatialEntities	975
7.203.4 Member Data Documentation	975
7.203.4.1 consideredSpatialEntityTypes	975
7.203.4.2 ERR_GET_NUMERIC_STATE_VARIABLE_PREFIX	976
7.203.4.3 ERR_GET_NUMERIC_STATE_VARIABLE_SUFFIX	976
7.203.4.4 numericStateVariables	976

7.203.4.5 spatialEntities	976
7.203.4.6 value	976
7.204 multiscale::verification::TimePointEvaluator Class Reference	976
7.204.1 Detailed Description	977
7.204.2 Member Function Documentation	977
7.204.2.1 getSpatialMeasureValues	977
7.204.2.2 getSpatialMeasureValues	978
7.205 multiscale::verification::TimeseriesComponentAttribute Class Reference	979
7.205.1 Detailed Description	979
7.205.2 Member Data Documentation	979
7.205.2.1 timeseriesComponent	979
7.206 multiscale::verification::TimeseriesComponentEvaluator Class Reference	980
7.206.1 Detailed Description	980
7.206.2 Member Function Documentation	981
7.206.2.1 evaluate	981
7.206.2.2 evaluateHomogeneousComponentIndices	982
7.207 multiscale::verification::TimeseriesComponentVisitor Class Reference	982
7.207.1 Detailed Description	985
7.207.2 Constructor & Destructor Documentation	985
7.207.2.1 TimeseriesComponentVisitor	985
7.207.3 Member Function Documentation	985
7.207.3.1 duplicateCollectionElements	985
7.207.3.2 evaluateHeterogeneousComponentsIndices	985
7.207.3.3 operator()	986
7.207.3.4 operator()	986
7.207.4 Member Data Documentation	986
7.207.4.1 values	986
7.208 multiscale::verification::TimeseriesMeasureAttribute Class Reference	986
7.208.1 Detailed Description	987
7.208.2 Member Data Documentation	987
7.208.2.1 timeseriesMeasure	987
7.209 multiscale::verification::TimeseriesMeasureTypeParser Struct Reference	987
7.209.1 Detailed Description	989
7.209.2 Constructor & Destructor Documentation	989
7.209.2.1 TimeseriesMeasureTypeParser	989
7.210 multiscale::verification::TimeseriesTimeseriesComponentAttribute Class Reference	989
7.210.1 Detailed Description	990
7.210.2 Member Data Documentation	990
7.210.2.1 temporalNumericMeasureCollection	990
7.210.2.2 timeseriesComponent	990

7.210.2.3 timeseriesMeasure	990
7.211 multiscaletest::TraceEvaluationTest Class Reference	990
7.211.1 Detailed Description	993
7.211.2 Constructor & Destructor Documentation	993
7.211.2.1 TraceEvaluationTest	993
7.211.3 Member Function Documentation	994
7.211.3.1 InitialiseQuery	994
7.211.3.2 InitialiseTrace	995
7.211.3.3 InitialiseTypeSemanticsTable	995
7.211.3.4 RunEvaluationTest	995
7.211.3.5 RunTest	995
7.211.3.6 ValidateTestResults	995
7.211.4 Member Data Documentation	996
7.211.4.1 a.MaxValue	996
7.211.4.2 a.MinValue	996
7.211.4.3 a.NumericStateVariableId	996
7.211.4.4 aWithTypeNumericStateVariableId	996
7.211.4.5 b.ConstantValue	996
7.211.4.6 b.NumericStateVariableId	996
7.211.4.7 bWithTypeNumericStateVariableId	996
7.211.4.8 c.MaxValue	996
7.211.4.9 c.MinValue	997
7.211.4.10 c.NumericStateVariableId	997
7.211.4.11 evaluationResult	997
7.211.4.12 hrOfTimePoints	997
7.211.4.13 query	997
7.211.4.14 SEMANTIC_TYPE_ORGAN_HEART	997
7.211.4.15 SEMANTIC_TYPE_ORGAN_KIDNEY	997
7.211.4.16 trace	997
7.211.4.17 typeSemanticsTable	998
7.212 multiscale::verification::TypeSemanticsTable Class Reference	998
7.212.1 Detailed Description	1001
7.212.2 Constructor & Destructor Documentation	1001
7.212.2.1 TypeSemanticsTable	1001
7.212.2.2 ~TypeSemanticsTable	1001
7.212.3 Member Function Documentation	1001
7.212.3.1 addSemanticTypesToTable	1001
7.212.3.2 addSemanticTypeToTable	1002
7.212.3.3 addTableEntry	1002
7.212.3.4 areDefaultSemanticCriteriaValues	1002

7.212.3.5 constructTypeSemanticsTable	1002
7.212.3.6 existsEntryInTableForSemanticCriteriaValues	1003
7.212.3.7 getBeginIterator	1003
7.212.3.8 getEndIterator	1003
7.212.3.9 getTypeOfSemanticCriteriaValues	1003
7.212.3.10sNoUndefinedSemanticCriteriaValue	1004
7.212.3.11processSemanticCriteriaDescription	1004
7.212.3.12read	1004
7.212.3.13readFromValidFile	1004
7.212.3.14readFromValidXmlFile	1004
7.212.3.15readTableFromFile	1005
7.212.4 Member Data Documentation	1005
7.212.4.1 DEFAULT_TYPE_VALUE	1005
7.212.4.2 ERR_INVALID_INPUT_FILE_PATH_BEGIN	1005
7.212.4.3 ERR_INVALID_INPUT_FILE_PATH_END	1005
7.212.4.4 ERR_INVALID_INPUT_FILE_RELATIVE_TO_XSD_BEGIN	1005
7.212.4.5 ERR_INVALID_INPUT_FILE_RELATIVE_TO_XSD_END	1005
7.212.4.6 ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_BEGIN	1006
7.212.4.7 ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_END	1006
7.212.4.8 ERR_UNDEFINED_SEMANTIC_CRITERIA_BEGIN	1006
7.212.4.9 ERR_UNDEFINED_SEMANTIC_CRITERIA_END	1006
7.212.4.10ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE1	1006
7.212.4.11ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE2	1006
7.212.4.12nputFilePath	1006
7.212.4.13LABEL_SEMANTIC_CRITERIA_DESCRIPTION	1006
7.212.4.14LABEL_SEMANTIC_TYPE	1007
7.212.4.15LABEL_SEMANTIC_TYPE_SEMATICS	1007
7.212.4.16LABEL_SEMANTIC_TYPE_VALUE	1007
7.212.4.17LABEL_SEMANTIC_TYPES	1007
7.212.4.18LABEL_TYPE_SEMATICS_TABLE	1007
7.212.4.19nrOfDescribedSemanticCriteria	1007
7.212.4.20SEMANTIC_CRITERIA_SEPARATOR	1007
7.212.4.21semanticCriteriaDescription	1007
7.212.4.22table	1007
7.212.4.23TYPE_SEMATICS_TABLE_INPUT_FILE_EXTENSION	1008
7.212.4.24TYPE_SEMATICS_TABLE_XSD_PATH	1008
7.213multiscale::verification::UnaryNumericFilterAttribute Class Reference	1008
7.213.1 Detailed Description	1009
7.213.2 Member Data Documentation	1009
7.213.2.1 filterNumericMeasure	1009

7.213.2.2 unaryNumericMeasure	1009
7.214 multiscale::verification::UnaryNumericMeasureAttribute Class Reference	1009
7.214.1 Detailed Description	1010
7.214.2 Member Data Documentation	1010
7.214.2.1 unaryNumericMeasureType	1010
7.215 multiscale::verification::UnaryNumericMeasureGrammar< Iterator > Class Template Reference	1010
7.215.1 Detailed Description	1013
7.215.2 Constructor & Destructor Documentation	1013
7.215.2.1 UnaryNumericMeasureGrammar	1013
7.215.3 Member Function Documentation	1013
7.215.3.1 assignNamesToRules	1013
7.215.3.2 initialise	1013
7.215.3.3 initialiseDebugSupport	1013
7.215.3.4 initialiseGrammar	1013
7.215.3.5 initialiseRulesDebugging	1014
7.215.4 Member Data Documentation	1014
7.215.4.1 unaryNumericMeasureRule	1014
7.215.4.2 unaryNumericMeasureTypeParser	1014
7.216 multiscale::verification::UnaryNumericMeasureTypeParser Struct Reference	1014
7.216.1 Detailed Description	1016
7.216.2 Constructor & Destructor Documentation	1016
7.216.2.1 UnaryNumericMeasureTypeParser	1016
7.217 multiscale::verification::UnaryNumericNumericAttribute Class Reference	1016
7.217.1 Detailed Description	1017
7.217.2 Member Data Documentation	1017
7.217.2.1 numericMeasure	1017
7.217.2.2 unaryNumericMeasure	1017
7.218 multiscale::verification::UnaryNumericTemporalAttribute Class Reference	1017
7.218.1 Detailed Description	1018
7.218.2 Member Data Documentation	1018
7.218.2.1 temporalNumericMeasure	1018
7.218.2.2 unaryNumericMeasure	1018
7.219 multiscale::verification::UnarySpatialConstraintAttribute Class Reference	1019
7.219.1 Detailed Description	1019
7.219.2 Member Data Documentation	1019
7.219.2.1 comparator	1019
7.219.2.2 filterNumericMeasure	1020
7.219.2.3 spatialMeasure	1020
7.220 multiscale::verification::UnaryStatisticalMeasureAttribute Class Reference	1020
7.220.1 Detailed Description	1020

7.220.2 Member Data Documentation	1021
7.220.2.1 unaryStatisticalMeasureType	1021
7.221 multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator > Class Template Reference	1021
7.221.1 Detailed Description	1024
7.221.2 Constructor & Destructor Documentation	1024
7.221.2.1 UnaryStatisticalMeasureGrammar	1024
7.221.3 Member Function Documentation	1024
7.221.3.1 assignNamesToRules	1024
7.221.3.2 initialise	1024
7.221.3.3 initialiseDebugSupport	1024
7.221.3.4 initialiseGrammar	1024
7.221.3.5 initialiseRulesDebugging	1025
7.221.4 Member Data Documentation	1025
7.221.4.1 unaryStatisticalMeasureRule	1025
7.221.4.2 unaryStatisticalMeasureTypeParser	1025
7.222 multiscale::verification::UnaryStatisticalMeasureTypeParser Struct Reference	1025
7.222.1 Detailed Description	1027
7.222.2 Constructor & Destructor Documentation	1027
7.222.2.1 UnaryStatisticalMeasureTypeParser	1027
7.223 multiscale::verification::UnaryStatisticalNumericAttribute Class Reference	1028
7.223.1 Detailed Description	1028
7.223.2 Member Data Documentation	1028
7.223.2.1 numericMeasureCollection	1028
7.223.2.2 unaryStatisticalMeasure	1028
7.224 multiscale::verification::UnaryStatisticalSpatialAttribute Class Reference	1029
7.224.1 Detailed Description	1029
7.224.2 Member Data Documentation	1029
7.224.2.1 spatialMeasureCollection	1030
7.224.2.2 unaryStatisticalMeasure	1030
7.225 multiscale::verification::UnaryTypeConstraintAttribute Class Reference	1030
7.225.1 Detailed Description	1031
7.225.2 Member Data Documentation	1032
7.225.2.1 comparator	1032
7.225.2.2 semanticType	1032
7.226 multiscale::UnexpectedBehaviourException Class Reference	1032
7.226.1 Detailed Description	1035
7.226.2 Constructor & Destructor Documentation	1035
7.226.2.1 UnexpectedBehaviourException	1035
7.226.2.2 UnexpectedBehaviourException	1035
7.226.2.3 UnexpectedBehaviourException	1035

7.227 multiscale::verification::UnexpectedErrorHandler Struct Reference	1035
7.227.1 Detailed Description	1036
7.227.2 Member Function Documentation	1036
7.227.2.1 getExpectedTokenAsString	1036
7.227.2.2 operator()	1036
7.228 multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponent< Relation > Class Template Reference	1037
7.228.1 Detailed Description	1037
7.228.2 Member Function Documentation	1037
7.228.2.1 computeEndIndex	1038
7.228.2.2 hasValidSuccessors	1038
7.228.2.3 startIndex	1038
7.229 multiscale::UnimplementedMethodException Class Reference	1039
7.229.1 Detailed Description	1042
7.229.2 Constructor & Destructor Documentation	1042
7.229.2.1 UnimplementedMethodException	1042
7.229.2.2 UnimplementedMethodException	1042
7.229.2.3 UnimplementedMethodException	1042
7.230 multiscale::verification::UntilLogicPropertyAttribute Class Reference	1042
7.230.1 Detailed Description	1043
7.230.2 Member Data Documentation	1043
7.230.2.1 endTimepoint	1043
7.230.2.2 logicProperty	1043
7.230.2.3 startTimepoint	1043
7.231 multiscale::UserDefinedTypeName< T > Class Template Reference	1043
7.231.1 Detailed Description	1044
7.231.2 Member Function Documentation	1044
7.231.2.1 name	1044
7.232 multiscale::XmlValidator::XmlValidationErrorHandler Class Reference	1044
7.232.1 Detailed Description	1047
7.232.2 Member Function Documentation	1047
7.232.2.1 constructExceptionMessage	1047
7.232.2.2 error	1047
7.232.2.3 fatalError	1047
7.232.2.4 handleValidationException	1048
7.232.2.5 resetErrors	1048
7.232.2.6 warning	1048
7.232.3 Member Data Documentation	1048
7.232.3.1 ERR_EXCEPTION_BEGIN_MSG	1048
7.232.3.2 ERR_EXCEPTION_COLUMN_MSG	1048

7.232.3.3 ERR_EXCEPTION_END_MSG	1048
7.232.3.4 ERR_EXCEPTION_LINE_MSG	1049
7.232.3.5 ERR_EXCEPTION_MIDDLE_MSG	1049
7.233 multiscale::XmlValidator Class Reference	1049
7.233.1 Detailed Description	1051
7.233.2 Member Function Documentation	1051
7.233.2.1 checkIfValidXmlFile	1051
7.233.2.2 configureParser	1052
7.233.2.3 isValidXmlFile	1052
7.233.2.4 isValidXmlFile	1052
7.233.2.5 isValidXmlPathAndFile	1052
7.233.2.6 loadParserSchema	1054
7.233.2.7 validateXmlFilepath	1054
7.233.2.8 validateXmlSchemaPath	1054
7.233.2.9 verifyIfValidXmlFile	1054
7.233.3 Member Data Documentation	1055
7.233.3.1 ERR_INVALID_SCHEMA_FILEPATH	1055
7.233.3.2 ERR_INVALID_XML_FILEPATH	1055
7.233.3.3 ERR_SCHEMA_CONTENTS	1055
8 File Documentation	1057
8.1 config/mainpage.dox File Reference	1057
8.2 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/Multiscale.hpp File Reference	1057
8.3 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/MultiscaleTest.hpp File Reference	1058
8.4 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/UserDefinedType←Name.hpp File Reference	1058
8.4.1 Macro Definition Documentation	1059
8.4.1.1 DEFINE_TYPE_NAME	1059
8.5 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/AlgorithmException.hpp File Reference	1060
8.6 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ExceptionHandler.hpp File Reference	1060
8.7 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/FileOpenException.hpp File Reference	1061
8.8 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/IndexOutOfBoundsException.hpp File Reference	1062
8.9 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/InvalidInputException.hpp File Reference	1063
8.10 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/IOException.hpp File Reference	1064

8.11 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/MultiscaleException.hpp File Reference	1065
8.11.1 Macro Definition Documentation	1066
8.11.1.1 MS_throw	1066
8.11.1.2 MS_throw_detailed	1068
8.12 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/NumericException.hpp File Reference	1068
8.13 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/RuntimeException.hpp File Reference	1068
8.14 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/TestException.hpp File Reference	1069
8.15 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/UnexpectedBehaviourException.hpp File Reference	1070
8.16 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/UnimplementedMethodException.hpp File Reference	1071
8.17 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/CircularityMeasure.hpp File Reference	1072
8.18 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Cluster.hpp File Reference	1073
8.19 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Cluster.hpp File Reference	1074
8.20 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp File Reference	1075
8.21 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/ClusterDetector.hpp File Reference	1076
8.22 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/DataPoint.hpp File Reference	1077
8.23 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/DBSCAN.hpp File Reference	1078
8.24 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Detector.hpp File Reference	1079
8.25 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Entity.hpp File Reference	1079
8.26 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/CircularMatFactory.hpp File Reference	1080
8.27 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/RectangularMatFactory.hpp File Reference	1081
8.28 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/MatFactory.hpp File Reference	1082
8.29 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Region.hpp File Reference	1083
8.30 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Region.hpp File Reference	1084
8.31 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/RegionDetector.hpp File Reference	1085
8.32 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Shape2D.hpp File Reference	1086

8.33 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Silhouette.hpp File Reference	1087
8.34 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp File Reference	1087
8.35 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/SpatialEntityPseudo3DType.hpp File Reference	1088
8.36 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/CircularMatFactorySample.cpp File Reference	1088
8.36.1 Function Documentation	1089
8.36.1.1 main	1089
8.37 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/LexicographicIteratorSample.cpp File Reference	1089
8.37.1 Function Documentation	1090
8.37.1.1 main	1090
8.38 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/RectangularMatFactorySample.cpp File Reference	1090
8.38.1 Function Documentation	1091
8.38.1.1 main	1091
8.39 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularDetectorRegions.cpp File Reference	1091
8.39.1 Function Documentation	1092
8.39.1.1 areValidParameters	1092
8.39.1.2 initArgumentsConfig	1092
8.39.1.3 loadDetectorParameterValues	1092
8.39.1.4 loadDetectorParameterValues	1093
8.39.1.5 main	1093
8.39.1.6 printHelpInformation	1093
8.39.1.7 printWrongParameters	1093
8.39.1.8 saveDetectorParameterValues	1093
8.39.1.9 saveDetectorParameterValues	1093
8.39.2 Variable Documentation	1093
8.39.2.1 CONFIG_FILE	1093
8.39.2.2 LABEL_ALPHA	1094
8.39.2.3 LABEL_BETA	1094
8.39.2.4 LABEL_BLUR_KERNEL_SIZE	1094
8.39.2.5 LABEL_EPSILON	1094
8.39.2.6 LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS	1094
8.39.2.7 LABEL_REGION_AREA_THRESH	1094
8.39.2.8 LABEL_ROOT_COMMENT	1094
8.39.2.9 LABEL_THRESHOLD_VALUE	1094
8.39.2.10 ROOT_COMMENT	1094

8.40 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularityMeasure.cpp File Reference	1095
8.41 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Cluster.cpp File Reference	1095
8.42 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/SimulationClusterDetector.cpp File Reference	1096
8.43 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/ClusterDetector.cpp File Reference	1096
8.44 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/DBSCAN.cpp File Reference	1097
8.45 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Detector.cpp File Reference	1097
8.46 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Entity.cpp File Reference	1098
8.47 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/CircularMatFactory.cpp File Reference	1098
8.48 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/RectangularMatFactory.cpp File Reference	1099
8.49 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/MatFactory.cpp File Reference	1100
8.50 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.cpp File Reference	1100
8.50.1 Function Documentation	1101
8.50.1.1 areValidParameters	1101
8.50.1.2 initArgumentsConfig	1101
8.50.1.3 loadDetectorParameterValues	1101
8.50.1.4 loadDetectorParameterValues	1102
8.50.1.5 main	1102
8.50.1.6 printHelpInformation	1102
8.50.1.7 printWrongParameters	1102
8.50.1.8 saveDetectorParameterValues	1102
8.50.1.9 saveDetectorParameterValues	1102
8.50.2 Variable Documentation	1102
8.50.2.1 CONFIG_FILE	1102
8.50.2.2 LABEL_ALPHA	1103
8.50.2.3 LABEL_BETA	1103
8.50.2.4 LABEL_BLUR_KERNEL_SIZE	1103
8.50.2.5 LABEL_EPSILON	1103
8.50.2.6 LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS	1103
8.50.2.7 LABEL_REGION_AREA_THRESH	1103
8.50.2.8 LABEL_ROOT_COMMENT	1103
8.50.2.9 LABEL_THRESHOLD_VALUE	1103
8.50.2.10 ROOT_COMMENT	1103

8.51 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Region.cpp File Reference	1104
8.52 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/RegionDetector.cpp File Reference	1104
8.53 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Silhouette.cpp File Reference	1105
8.54 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SimulationDetect→ Clusters.cpp File Reference	1105
8.54.1 Function Documentation	1106
8.54.1.1 areValidParameters	1106
8.54.1.2 initArgumentsConfig	1107
8.54.1.3 loadDetectorParameterValues	1107
8.54.1.4 loadDetectorParameterValues	1107
8.54.1.5 main	1107
8.54.1.6 printHelpInformation	1107
8.54.1.7 printWrongParameters	1107
8.54.1.8 saveDetectorParameterValues	1107
8.54.1.9 saveDetectorParameterValues	1107
8.54.2 Variable Documentation	1108
8.54.2.1 CONFIG_FILE	1108
8.54.2.2 LABEL_EPS	1108
8.54.2.3 LABEL_MINPOINTS	1108
8.54.2.4 LABEL_ROOT_COMMENT	1108
8.54.2.5 ROOT_COMMENT	1108
8.55 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SpatialEntity← Pseudo3D.cpp File Reference	1108
8.56 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/DBSCANTest.cpp File Reference	1109
8.56.1 Function Documentation	1109
8.56.1.1 convertPoints	1110
8.56.1.2 main	1110
8.56.1.3 printResults	1110
8.56.1.4 runTest	1110
8.56.1.5 runTest1	1110
8.56.1.6 runTest2	1110
8.56.1.7 runTest3	1110
8.56.1.8 runTest4	1110
8.56.1.9 runTest5	1111
8.56.1.10 runTests	1111
8.57 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Console← Printer.hpp File Reference	1111

8.58 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Filesystem.hpp File Reference	1112
8.59 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Geometry2D.hpp File Reference	1113
8.60 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/← LexicographicNumberIterator.hpp File Reference	1113
8.61 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/← NumberIteratorType.hpp File Reference	1114
8.62 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/← StandardNumberIterator.hpp File Reference	1114
8.63 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/MinEnclosingTriangleFinder.hpp File Reference	1115
8.64 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumberIterator.hpp File Reference	1116
8.65 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Numeric.hpp File Reference	1116
8.66 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumericRangeManipulator.hpp File Reference	1117
8.67 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/OperatingSystem.hpp File Reference	1118
8.68 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/RGBColourGenerator.hpp File Reference	1118
8.69 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/← BetaDistribution.hpp File Reference	1119
8.70 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/← BinomialDistribution.hpp File Reference	1120
8.71 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/← Distribution.hpp File Reference	1121
8.72 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/StringManipulator.hpp File Reference	1122
8.73 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/XmlValidator.hpp File Reference	1123
8.74 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/ConsolePrinterSample.cpp File Reference	1124
8.74.1 Function Documentation	1125
8.74.1.1 main	1125
8.74.2 Variable Documentation	1125
8.74.2.1 SAMPLE_MSG	1125
8.74.2.2 SAMPLE_TAG	1125
8.75 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/ExecuteProgramSample.cpp File Reference	1125
8.75.1 Function Documentation	1126
8.75.1.1 main	1126
8.76 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/LineCircleIntersectionSample.cpp File Reference	1126
8.76.1 Function Documentation	1127

8.76.1.1	main	1127
8.76.1.2	printPoints	1127
8.77	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/MinEnclosingTriangleFinderSample.cpp File Reference	1127
8.77.1	Function Documentation	1129
8.77.1.1	arePointsEnclosed	1129
8.77.1.2	generateRandomSetOf2DPoints	1129
8.77.1.3	isOneEdgeFlush	1129
8.77.1.4	isTriangleTouchingPolygon	1129
8.77.1.5	isValidTriangle	1129
8.77.1.6	main	1129
8.77.1.7	outputMinEnclosingTriangleFinderResults	1129
8.77.1.8	printPolygon	1130
8.77.1.9	runMinEnclosingTriangleFinder	1130
8.77.1.10	runMinEnclosingTriangleFinder	1130
8.77.1.11	runMinEnclosingTriangleFinderUsingRandomPolygons	1130
8.77.2	Variable Documentation	1130
8.77.2.1	KEY_ESC	1130
8.77.2.2	LINE_THICKNESS	1130
8.77.2.3	MAX_POLYGON_POINTS	1130
8.77.2.4	NR_RAND_POLYGONS	1130
8.77.2.5	POINT_IN_TRIANGLE_THRESH	1131
8.77.2.6	POLYGON_POINT_X_MAX	1131
8.77.2.7	POLYGON_POINT_Y_MAX	1131
8.77.2.8	RADIUS	1131
8.77.2.9	WIN_MIN_AREA_TRIANGLE	1131
8.78	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/RGBColourGeneratorSample.cpp File Reference	1131
8.78.1	Function Documentation	1132
8.78.1.1	main	1132
8.79	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/XmlValidatorSample.cpp File Reference	1132
8.79.1	Function Documentation	1132
8.79.1.1	checkIfValidXmlFile	1132
8.79.1.2	main	1132
8.79.1.3	printValidationResult	1133
8.79.1.4	validateXmlFile	1133
8.80	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ConsolePrinter.cpp File Reference	1133
8.81	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Filesystem.cpp File Reference	1133
8.82	/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Geometry2D.cpp File Reference	1134

8.83 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/LexicographicNumberIterator.cpp File Reference	1135
8.84 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/StandardNumberIterator.cpp File Reference	1135
8.85 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/MinEnclosingTriangleFinder.cpp File Reference	1136
8.86 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/NumberIterator.cpp File Reference	1137
8.87 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Numeric.cpp File Reference	1137
8.88 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/OperatingSystem.cpp File Reference	1138
8.89 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/RGBColourGenerator.cpp File Reference	1138
8.90 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/BetaDistribution.cpp File Reference	1139
8.91 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/BinomialDistribution.cpp File Reference	1139
8.92 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/Distribution.cpp File Reference	1140
8.93 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/StringManipulator.cpp File Reference	1141
8.94 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/XmlValidator.cpp File Reference	1141
8.95 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/Geometry2DTest.cpp File Reference	1142
8.95.1 Function Documentation	1142
8.95.1.1 main	1142
8.95.1.2 TEST	1142
8.95.1.3 TEST	1143
8.95.2 Variable Documentation	1143
8.95.2.1 DOUBLE_COMP_ERROR	1143
8.96 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/MinEnclosingTriangleFinderTest.cpp File Reference	1143
8.96.1 Function Documentation	1144
8.96.1.1 main	1144
8.96.1.2 TEST_F	1144
8.96.1.3 TEST_F	1144
8.96.1.4 TEST_F	1144
8.96.1.5 TEST_F	1144
8.97 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/NumericTest.cpp File Reference	1144
8.97.1 Function Documentation	1146
8.97.1.1 main	1146
8.97.1.2 TEST	1146
8.97.1.3 TEST	1146
8.97.1.4 TEST	1146

8.97.1.5 TEST	1146
8.97.1.6 TEST	1146
8.97.1.7 TEST	1146
8.97.1.8 TEST	1146
8.97.1.9 TEST	1146
8.97.1.10 TEST	1147
8.97.1.11 TEST	1147
8.97.1.12 TEST	1147
8.97.1.13 TEST	1147
8.97.1.14 TEST	1147
8.97.1.15 TEST	1147
8.97.1.16 TEST	1147
8.97.1.17 TEST	1147
8.97.1.18 TEST	1147
8.97.1.19 TEST	1148
8.97.1.20 TEST	1148
8.97.1.21 TEST	1148
8.97.1.22 TEST	1148
8.97.2 Variable Documentation	1148
8.97.2.1 DOUBLE_COMP_ERROR	1148
8.98 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/StatisticsTest.cpp File Reference	148
8.98.1 Function Documentation	1149
8.98.1.1 TEST	1149
8.98.1.2 TEST	1149
8.98.1.3 TEST	1149
8.99 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndConstraintAttribute.hpp File Reference	1149
8.100 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndLogicPropertyAttribute.hpp File Reference	1150
8.101 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericFilterAttribute.hpp File Reference	1151
8.102 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp File Reference	1152
8.103 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericNumericAttribute.hpp File Reference	1154
8.104 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericTemporalAttribute.hpp File Reference	1154
8.105 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp File Reference	1155
8.106 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalNumericAttribute.hpp File Reference	1156
8.107 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp File Reference	1157

8.108/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileNumericAttribute.hpp File Reference	1159
8.109/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileSpatialAttribute.hpp File Reference	1159
8.110/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalSpatialAttribute.hpp File Reference	1160
8.111/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp File Reference	1161
8.112/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericCollectionAttribute.hpp File Reference	1162
8.113/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericMeasureAttribute.hpp File Reference	1163
8.114/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp File Reference	1164
8.115/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp File Reference	1166
8.116/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/EquivalenceConstraintAttribute.hpp File Reference	1167
8.117/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/EquivalenceLogicPropertyAttribute.hpp File Reference	1168
8.118/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FilterNumericMeasureAttribute.hpp File Reference	1169
8.119/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FilterSubsetAttribute.hpp File Reference	1171
8.120/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FutureLogicPropertyAttribute.hpp File Reference	1171
8.121/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/GlobalLogicPropertyAttribute.hpp File Reference	1172
8.122/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HeterogeneousTimeseriesComponentAttribute.hpp File Reference	1173
8.123/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousHomogeneousTimeseriesAttribute.hpp File Reference	1175
8.124/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp File Reference	1175
8.125/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesMeasureAttribute.hpp File Reference	1177
8.126/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationConstraintAttribute.hpp File Reference	1178
8.127/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationLogicPropertyAttribute.hpp File Reference	1179
8.128/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp File Reference	1179
8.129/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextKLogicPropertyAttribute.hpp File Reference	1181
8.130/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextLogicPropertyAttribute.hpp File Reference	1181

8.131/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Nil.hpp File Reference	1182
8.131.1 Function Documentation	1183
8.131.1.1 BOOST_FUSION_ADAPT_STRUCT	1183
8.132/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotConstraintAttribute.hpp File Reference	1183
8.133/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotLogicPropertyAttribute.hpp File Reference	1184
8.134/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp File Reference	1185
8.135/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp File Reference	1187
8.136/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericSpatialMeasureAttribute.hpp File Reference	1187
8.137/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp File Reference	1188
8.138/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStatisticalMeasureAttribute.hpp File Reference	1189
8.139/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrConstraintAttribute.hpp File Reference	1190
8.140/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrLogicPropertyAttribute.hpp File Reference	1191
8.141/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryConstraintAttribute.hpp File Reference	1192
8.142/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryLogicPropertyAttribute.hpp File Reference	1194
8.142.1 Function Documentation	1195
8.142.1.1 BOOST_FUSION_ADAPT_STRUCT	1195
8.143/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryNumericMeasureAttribute.hpp File Reference	1195
8.144/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp File Reference	1196
8.144.1 Function Documentation	1197
8.144.1.1 BOOST_FUSION_ADAPT_STRUCT	1197
8.145/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp File Reference	1197
8.146/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityMeasureAttribute.hpp File Reference	1198
8.147/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityTemporalNumericCollectionAttribute.hpp File Reference	1199
8.148/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp File Reference	1200
8.149/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp File Reference	1201
8.150/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureType.hpp File Reference	1202

8.151/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialNumericComparisonAttribute.hpp File Reference	1202
8.152/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/StateVariableAttribute.hpp File Reference	1203
8.153/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetAttribute.hpp File Reference	1204
8.154/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp File Reference	1205
8.155/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp File Reference	1206
8.156/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSpecificType.hpp File Reference	1208
8.157/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSubsetOperationAttribute.hpp File Reference	1208
8.158/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp File Reference	1209
8.159/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp File Reference	1211
8.160/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericComparisonAttribute.hpp File Reference	1212
8.161/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp File Reference	1213
8.162/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureCollectionAttribute.hpp File Reference	1214
8.163/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesComponentAttribute.hpp File Reference	1214
8.164/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesMeasureAttribute.hpp File Reference	1215
8.165/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesTimeseriesComponentAttribute.hpp File Reference	1216
8.166/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericFilterAttribute.hpp File Reference	1217
8.167/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp File Reference	1218
8.168/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericNumericAttribute.hpp File Reference	1220
8.169/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericTemporalAttribute.hpp File Reference	1220
8.170/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnarySpatialConstraintAttribute.hpp File Reference	1221
8.171/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp File Reference	1222
8.172/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalNumericAttribute.hpp File Reference	1224
8.173/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalSpatialAttribute.hpp File Reference	1224

8.174/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryTypeConstraintAttribute.hpp File Reference	1225
8.175/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UntilLogicPropertyAttribute.hpp File Reference	1226
8.176/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp File Reference	1227
8.177/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelCheckerFactory.hpp File Reference	1228
8.178/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp File Reference	1229
8.179/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelCheckerFactory.hpp File Reference	1229
8.180/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp File Reference	1230
8.181/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/BayesianModelCheckerFactory.hpp File Reference	1230
8.182/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelChecker.hpp File Reference	1231
8.183/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp File Reference	1232
8.184/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingManager.hpp File Reference	1232
8.185/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp File Reference	1233
8.186/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp File Reference	1234
8.187/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelCheckerFactory.hpp File Reference	1234
8.188/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/StatisticalModelChecker.hpp File Reference	1235
8.189/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/StatisticalModelCheckerFactory.hpp File Reference	1236
8.190/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/LogicPropertyDataReader.hpp File Reference	1236
8.191/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp File Reference	1237
8.192/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/TemporalDataReader.hpp File Reference	1238
8.193/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingException.hpp File Reference	1239
8.194/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingHelpRequestException.hpp File Reference	1240
8.195/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarExceptionHandler.hpp File Reference	1241
8.196/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarExtraInputException.hpp File Reference	1242

8.197/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarProbabilityException.hpp File Reference	1243
8.198/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarUnexpectedTokenException.hpp File Reference	1244
8.199/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarUnparseableInputException.hpp File Reference	1245
8.200/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp File Reference	1246
8.201/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/CommandLineModelChecking.hpp File Reference	1247
8.202/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/ProbabilityErrorHandler.hpp File Reference	1248
8.203/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp File Reference	1249
8.204/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp File Reference	1250
8.205/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/NumericStateVariableId.hpp File Reference	1250
8.206/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SemanticType.hpp File Reference	1251
8.207/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialEntity.hpp File Reference	1252
8.208/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp File Reference	1253
8.209/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TimePoint.hpp File Reference	1254
8.210/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp File Reference	1255
8.211/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp File Reference	1256
8.212/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammarDefinition.hpp File Reference	1257
8.213/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammar.hpp File Reference	1258
8.214/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammarDefinition.hpp File Reference	1258
8.215/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammar.hpp File Reference	1259
8.216/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammarDefinition.hpp File Reference	1260
8.217/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammar.hpp File Reference	1260
8.218/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammarDefinition.hpp File Reference	1261
8.219/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.hpp File Reference	1262

8.220/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ComparatorGrammarDefinition.hpp File Reference	1262
8.221/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammar.hpp File Reference	1263
8.222/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammarDefinition.hpp File Reference	1264
8.223/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammar.hpp File Reference	1265
8.224/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammarDefinition.hpp File Reference	1265
8.225/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/Parser.hpp File Reference	1266
8.226/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammar.hpp File Reference	1267
8.227/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammarDefinition.hpp File Reference	1268
8.228/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammar.hpp File Reference	1268
8.229/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammarDefinition.hpp File Reference	1269
8.230/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammar.hpp File Reference	1270
8.231/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammarDefinition.hpp File Reference	1270
8.232/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp File Reference	1271
8.233/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammarDefinition.hpp File Reference	1272
8.234/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp File Reference	1272
8.235/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTablesAutoGenerated.hpp File Reference	1274
8.236/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammar.hpp File Reference	1275
8.237/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammarDefinition.hpp File Reference	1276
8.238/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammar.hpp File Reference	1276
8.239/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammarDefinition.hpp File Reference	1277
8.240/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp File Reference	1278
8.241/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammarDefinition.hpp File Reference	1279
8.242/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammar.hpp File Reference	1279

8.243/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammarDefinition.hpp File Reference	1280
8.244/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ChangeMeasureEvaluator.hpp File Reference	1281
8.245/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp File Reference	1281
8.246/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ConstraintVisitor.hpp File Reference	1282
8.247/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/FilterNumericVisitor.hpp File Reference	1283
8.248/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/LogicPropertyVisitor.hpp File Reference	1284
8.249/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp File Reference	1285
8.250/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionEvaluator.hpp File Reference	1286
8.251/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionVisitor.hpp File Reference	1287
8.252/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp File Reference	1288
8.253/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SpatialMeasureEvaluator.hpp File Reference	1289
8.254/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SubsetVisitor.hpp File Reference	1290
8.255/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TemporalNumericVisitor.hpp File Reference	1291
8.256/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimePointEvaluator.hpp File Reference	1292
8.257/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp File Reference	1293
8.258/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentVisitor.hpp File Reference	1294
8.259/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← LogicPropertyDataReaderSample.cpp File Reference	1295
8.259.1 Function Documentation	1296
8.259.1.1 main	1296
8.259.1.2 printParsingResult	1296
8.259.1.3 printQueries	1296
8.259.1.4 readQueriesFromFile	1296
8.260/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← ParserEvaluationSample.cpp File Reference	1296
8.260.1 Function Documentation	1297
8.260.1.1 initialiseTrace	1297
8.260.1.2 main	1297
8.261/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← ParserSample.cpp File Reference	1297

8.261.1 Function Documentation	1297
8.261.1.1 main	1297
8.262/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← PatternAnalysisSample.cpp File Reference	1298
8.262.1 Function Documentation	1298
8.262.1.1 analysePatterns	1298
8.262.1.2 main	1298
8.263/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← SpatialTemporalDataReaderSample.cpp File Reference	1298
8.263.1 Function Documentation	1299
8.263.1.1 main	1299
8.263.1.2 printSpatialEntities	1299
8.263.1.3 printTimePoint	1300
8.263.1.4 printTrace	1300
8.263.1.5 readValidXmlFiles	1300
8.263.1.6 readValidXmlFilesFromFolder	1300
8.264/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← TemporalDataReaderSample.cpp File Reference	1300
8.264.1 Function Documentation	1301
8.264.1.1 main	1301
8.264.1.2 printSpatialTemporalTrace	1301
8.264.1.3 printTimePoint	1301
8.264.1.4 readAndPrintInputFile	1301
8.265/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/← TypeSemanticsTableDataReaderSample.cpp File Reference	1302
8.265.1 Function Documentation	1302
8.265.1.1 main	1302
8.265.1.2 printTypeSemanticsTableContents	1302
8.265.1.3 readAndPrintTypeSemanticsTable	1303
8.265.1.4 readTypeSemanticsTable	1303
8.266/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← BinaryNumericMeasureAttribute.cpp File Reference	1303
8.267/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← BinaryStatisticalMeasureAttribute.cpp File Reference	1304
8.268/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← BinaryStatisticalQuantileMeasureAttribute.cpp File Reference	1304
8.269/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← ChangeMeasureAttribute.cpp File Reference	1305
8.270/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← ComparatorAttribute.cpp File Reference	1306
8.271/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← HeterogeneousTimeseriesComponentAttribute.cpp File Reference	1307

8.272/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← HomogeneousTimeseriesComponentAttribute.cpp File Reference	1308
8.273/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← HomogeneousTimeseriesMeasureAttribute.cpp File Reference	1309
8.274/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← ProbabilisticLogicPropertyAttribute.cpp File Reference	1310
8.275/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SimilarityMeasureAttribute.cpp File Reference	1310
8.276/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SpatialMeasureAttribute.cpp File Reference	1311
8.277/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SpatialMeasureAttributeAutoGenerated.cpp File Reference	1312
8.278/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SubsetOperationAttribute.cpp File Reference	1313
8.279/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SubsetSpecificAttribute.cpp File Reference	1314
8.280/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← SubsetSpecificAttributeAutoGenerated.cpp File Reference	1315
8.281/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← TimeseriesMeasureAttribute.cpp File Reference	1316
8.282/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← UnaryNumericMeasureAttribute.cpp File Reference	1317
8.283/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/← UnaryStatisticalMeasureAttribute.cpp File Reference	1318
8.284/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ApproximateBayesianModelChecker.cpp File Reference	1319
8.285/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ApproximateBayesianModelCheckerFactory.cpp File Reference	1320
8.286/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ApproximateProbabilisticModelChecker.cpp File Reference	1320
8.287/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ApproximateProbabilisticModelCheckerFactory.cpp File Reference	1320
8.288/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← BayesianModelChecker.cpp File Reference	1321
8.289/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← BayesianModelCheckerFactory.cpp File Reference	1321
8.290/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ModelChecker.cpp File Reference	1322
8.291/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ModelCheckingManager.cpp File Reference	1322
8.292/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ModelCheckingOutputWriter.cpp File Reference	1322
8.293/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ProbabilisticBlackBoxModelChecker.cpp File Reference	1323
8.294/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/← ProbabilisticBlackBoxModelCheckerFactory.cpp File Reference	1323

8.295/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelChecker.cpp File Reference	1324
8.296/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelCheckerFactory.cpp File Reference	1324
8.297/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ LogicPropertyDataReader.cpp File Reference	1325
8.298/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReader.cpp File Reference	1325
8.299/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReaderAutoGenerated.cpp File Reference	1326
8.300/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ TemporalDataReader.cpp File Reference	1327
8.301/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ ParserGrammarExceptionHandler.cpp File Reference	1327
8.302/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/ CommandLineModelChecking.cpp File Reference	1328
8.303/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ AbstractSyntaxTree.cpp File Reference	1329
8.304/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ NumericStateVariableId.cpp File Reference	1329
8.305/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SemanticType.cpp File Reference	1329
8.306/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialEntity.cpp File Reference	1330
8.307/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialTemporalTrace.cpp File Reference	1331
8.308/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ TimePoint.cpp File Reference	1331
8.309/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ TypeSemanticsTable.cpp File Reference	1332
8.310/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/ Mule.cpp File Reference	1332
8.310.1 Function Documentation	1332
8.310.1.1 main	1332
8.310.1.2 runModelCheckingTask	1333
8.311/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryNumericMeasureGrammar.cpp File Reference	1333
8.311.1 Typedef Documentation	1333
8.311.1.1 iteratorType	1333
8.312/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryStatisticalMeasureGrammar.cpp File Reference	1333
8.312.1 Typedef Documentation	1334
8.312.1.1 iteratorType	1334
8.313/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryStatisticalQuantileMeasureGrammar.cpp File Reference	1334
8.313.1 Typedef Documentation	1334

8.313.1.1 iteratorType	1334
8.314/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← ChangeMeasureGrammar.cpp File Reference	1334
8.314.1 Typedef Documentation	1335
8.314.1.1 iteratorType	1335
8.315/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← ComparatorGrammar.cpp File Reference	1335
8.315.1 Typedef Documentation	1335
8.315.1.1 iteratorType	1335
8.316/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← LogicPropertyGrammar.cpp File Reference	1335
8.316.1 Typedef Documentation	1336
8.316.1.1 iteratorType	1336
8.317/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← NumericStateVariableGrammar.cpp File Reference	1336
8.317.1 Typedef Documentation	1336
8.317.1.1 iteratorType	1336
8.318/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← Parser.cpp File Reference	1336
8.319/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← PrimaryNumericMeasureGrammar.cpp File Reference	1337
8.319.1 Typedef Documentation	1337
8.319.1.1 iteratorType	1337
8.320/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← SemanticTypeGrammar.cpp File Reference	1337
8.320.1 Typedef Documentation	1338
8.320.1.1 iteratorType	1338
8.321/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← SemanticTypeStringGrammar.cpp File Reference	1338
8.321.1 Typedef Documentation	1338
8.321.1.1 iteratorType	1338
8.322/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← SpatialMeasureCollectionGrammar.cpp File Reference	1339
8.322.1 Typedef Documentation	1339
8.322.1.1 iteratorType	1339
8.323/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← TemporalNumericCollectionGrammar.cpp File Reference	1339
8.323.1 Typedef Documentation	1339
8.323.1.1 iteratorType	1339
8.324/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/← TemporalNumericMeasureGrammar.cpp File Reference	1340
8.324.1 Typedef Documentation	1340
8.324.1.1 iteratorType	1340

8.325/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/UnaryNumericMeasureGrammar.cpp File Reference	1340
8.325.1 Typedef Documentation	1340
8.325.1.1 iteratorType	1340
8.326/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/UnaryStatisticalMeasureGrammar.cpp File Reference	1341
8.326.1 Typedef Documentation	1341
8.326.1.1 iteratorType	1341
8.327/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateBayesianModelCheckerTest.hpp File Reference	1341
8.327.1 Function Documentation	1342
8.327.1.1 TEST_F	1342
8.327.1.2 TEST_F	1342
8.328/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateProbabilisticModelCheckerTest.hpp File Reference	1342
8.328.1 Function Documentation	1342
8.328.1.1 TEST_F	1342
8.329/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/BayesianModelCheckerTest.hpp File Reference	1343
8.329.1 Function Documentation	1343
8.329.1.1 TEST_F	1343
8.329.1.2 TEST_F	1343
8.330/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckerTest.hpp File Reference	1343
8.330.1 Variable Documentation	1344
8.330.1.1 INPUT_LOGIC_PROPERTY	1344
8.331/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckingTest.cpp File Reference	1344
8.331.1 Function Documentation	1345
8.331.1.1 main	1345
8.332/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ProbabilisticBlackBoxModelCheckerTest.hpp File Reference	1345
8.332.1 Function Documentation	1345
8.332.1.1 TEST_F	1345
8.333/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/StatisticalModelCheckerTest.hpp File Reference	1345
8.333.1 Function Documentation	1346
8.333.1.1 TEST_F	1346
8.333.1.2 TEST_F	1346
8.334/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File Reference	1346
8.334.1 Function Documentation	1350
8.334.1.1 TEST_F	1350

8.334.1.2 TEST_F	1350
8.334.1.3 TEST_F	1350
8.334.1.4 TEST_F	1350
8.334.1.5 TEST_F	1351
8.334.1.6 TEST_F	1351
8.334.1.7 TEST_F	1351
8.334.1.8 TEST_F	1351
8.334.1.9 TEST_F	1351
8.334.1.10TEST_F	1351
8.334.1.11TEST_F	1351
8.334.1.12TEST_F	1351
8.334.1.13TEST_F	1351
8.334.1.14TEST_F	1351
8.334.1.15TEST_F	1351
8.334.1.16TEST_F	1351
8.334.1.17TEST_F	1352
8.334.1.18TEST_F	1352
8.334.1.19TEST_F	1352
8.334.1.20TEST_F	1352
8.334.1.21TEST_F	1352
8.334.1.22TEST_F	1352
8.334.1.23TEST_F	1352
8.334.1.24TEST_F	1352
8.334.1.25TEST_F	1352
8.334.1.26TEST_F	1352
8.334.1.27TEST_F	1352
8.334.1.28TEST_F	1352
8.334.1.29TEST_F	1353
8.334.1.30TEST_F	1353
8.334.1.31TEST_F	1353
8.334.1.32TEST_F	1353
8.334.1.33TEST_F	1353
8.334.1.34TEST_F	1353
8.334.1.35TEST_F	1353
8.334.1.36TEST_F	1353
8.334.1.37TEST_F	1353
8.334.1.38TEST_F	1353
8.334.1.39TEST_F	1353
8.334.1.40TEST_F	1353
8.334.1.41TEST_F	1354

8.334.1.42TEST_F	1354
8.334.1.43TEST_F	1354
8.334.1.44TEST_F	1354
8.334.1.45TEST_F	1354
8.334.1.46TEST_F	1354
8.334.1.47TEST_F	1354
8.334.1.48TEST_F	1354
8.334.1.49TEST_F	1354
8.334.1.50TEST_F	1354
8.334.1.51TEST_F	1354
8.334.1.52TEST_F	1354
8.334.1.53TEST_F	1355
8.334.1.54TEST_F	1355
8.334.1.55TEST_F	1355
8.334.1.56TEST_F	1355
8.334.1.57TEST_F	1355
8.334.1.58TEST_F	1355
8.334.1.59TEST_F	1355
8.334.1.60TEST_F	1355
8.334.1.61TEST_F	1355
8.334.1.62TEST_F	1355
8.334.1.63TEST_F	1355
8.334.1.64TEST_F	1355
8.334.1.65TEST_F	1356
8.334.1.66TEST_F	1356
8.334.1.67TEST_F	1356
8.334.1.68TEST_F	1356
8.334.1.69TEST_F	1356
8.334.1.70TEST_F	1356
8.334.1.71TEST_F	1356
8.334.1.72TEST_F	1356
8.334.1.73TEST_F	1356
8.334.1.74TEST_F	1356
8.334.1.75TEST_F	1356
8.334.1.76TEST_F	1356
8.334.1.77TEST_F	1357
8.334.1.78TEST_F	1357
8.334.1.79TEST_F	1357
8.334.1.80TEST_F	1357
8.334.1.81TEST_F	1357

8.334.1.82TEST_F	1357
8.334.1.83TEST_F	1357
8.334.1.84TEST_F	1357
8.334.1.85TEST_F	1357
8.334.1.86TEST_F	1357
8.334.1.87TEST_F	1357
8.334.1.88TEST_F	1357
8.334.1.89TEST_F	1358
8.334.1.90TEST_F	1358
8.334.1.91TEST_F	1358
8.334.1.92TEST_F	1358
8.334.1.93TEST_F	1358
8.334.1.94TEST_F	1358
8.334.1.95TEST_F	1358
8.334.1.96TEST_F	1358
8.334.1.97TEST_F	1358
8.334.1.98TEST_F	1358
8.334.1.99TEST_F	1358
8.334.1.100TEST_F	1358
8.334.1.101TEST_F	1359
8.334.1.102TEST_F	1359
8.334.1.103TEST_F	1359
8.334.1.104TEST_F	1359
8.334.1.105TEST_F	1359
8.334.1.106TEST_F	1359
8.334.1.107TEST_F	1359
8.334.1.108TEST_F	1359
8.334.1.109TEST_F	1359
8.334.1.110TEST_F	1359
8.334.1.111TEST_F	1359
8.334.1.112TEST_F	1359
8.334.1.113TEST_F	1360
8.334.1.114TEST_F	1360
8.334.1.115TEST_F	1360
8.334.1.116TEST_F	1360
8.334.1.117TEST_F	1360
8.334.1.118TEST_F	1360
8.334.1.119TEST_F	1360
8.334.1.120TEST_F	1360
8.334.1.121TEST_F	1360

8.334.1.12 TEST_F	1360
8.334.1.12 BEST_F	1360
8.334.1.12 TEST_F	1360
8.334.1.12 BEST_F	1361
8.334.1.12 BEST_F	1361
8.334.1.12 TEST_F	1361
8.334.1.12 BEST_F	1361
8.334.1.12 TEST_F	1361
8.334.1.13 BEST_F	1361
8.334.1.13 BEST_F	1361
8.334.1.13 TEST_F	1361
8.334.1.13 BEST_F	1361
8.334.1.13 TEST_F	1361
8.334.1.13 BEST_F	1361
8.334.1.13 TEST_F	1362
8.334.1.13 BEST_F	1362
8.334.1.13 TEST_F	1362
8.334.1.14 TEST_F	1362
8.334.1.14 TEST_F	1362
8.334.1.14 BEST_F	1362
8.334.1.14 BEST_F	1362
8.334.1.14 TEST_F	1362
8.334.1.14 BEST_F	1362
8.334.1.14 TEST_F	1362
8.334.1.14 BEST_F	1362
8.334.1.14 TEST_F	1363
8.334.1.15 TEST_F	1363
8.334.1.15 TEST_F	1363
8.334.1.15 BEST_F	1363
8.334.1.15 BEST_F	1363
8.334.1.15 TEST_F	1363
8.334.1.15 BEST_F	1363
8.334.1.15 TEST_F	1363
8.334.1.15 BEST_F	1363
8.334.1.16 TEST_F	1363
8.334.1.16 TEST_F	1364

8.334.1.16 TEST_F	1364
8.334.1.16BEST_F	1364
8.334.1.16 TEST_F	1364
8.334.1.16BEST_F	1364
8.334.1.16 TEST_F	1364
8.334.1.16TEST_F	1364
8.334.1.16 BEST_F	1364
8.334.1.16 TEST_F	1364
8.334.1.16TEST_F	1364
8.334.1.16 BEST_F	1364
8.334.1.16 TEST_F	1364
8.334.1.17 TEST_F	1364
8.334.1.17TEST_F	1364
8.334.1.17 TEST_F	1364
8.334.1.17BEST_F	1365
8.334.1.17 TEST_F	1365
8.334.1.17BEST_F	1365
8.334.1.17 TEST_F	1365
8.334.1.17TEST_F	1365
8.334.1.17 BEST_F	1365
8.334.1.17 TEST_F	1365
8.334.1.17TEST_F	1365
8.334.1.17 BEST_F	1365
8.334.1.18 TEST_F	1365
8.334.1.18TEST_F	1365
8.334.1.18 TEST_F	1365
8.334.1.18TEST_F	1365
8.334.1.18 BEST_F	1365
8.334.1.18 TEST_F	1366
8.334.1.18TEST_F	1366
8.334.1.18 TEST_F	1366
8.335/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/← EmptyTraceTest.hpp File Reference	1366
8.335.1 Function Documentation	1370
8.335.1.1 TEST_F	1370
8.335.1.2 TEST_F	1370
8.335.1.3 TEST_F	1370
8.335.1.4 TEST_F	1370
8.335.1.5 TEST_F	1370
8.335.1.6 TEST_F	1370
8.335.1.7 TEST_F	1370
8.335.1.8 TEST_F	1370
8.335.1.9 TEST_F	1370
8.335.1.10TEST_F	1370
8.335.1.11TEST_F	1370

8.335.1.12TEST_F	1371
8.335.1.13TEST_F	1371
8.335.1.14TEST_F	1371
8.335.1.15TEST_F	1371
8.335.1.16TEST_F	1371
8.335.1.17TEST_F	1371
8.335.1.18TEST_F	1371
8.335.1.19TEST_F	1371
8.335.1.20TEST_F	1371
8.335.1.21TEST_F	1371
8.335.1.22TEST_F	1371
8.335.1.23TEST_F	1371
8.335.1.24TEST_F	1372
8.335.1.25TEST_F	1372
8.335.1.26TEST_F	1372
8.335.1.27TEST_F	1372
8.335.1.28TEST_F	1372
8.335.1.29TEST_F	1372
8.335.1.30TEST_F	1372
8.335.1.31TEST_F	1372
8.335.1.32TEST_F	1372
8.335.1.33TEST_F	1372
8.335.1.34TEST_F	1372
8.335.1.35TEST_F	1373
8.335.1.36TEST_F	1373
8.335.1.37TEST_F	1373
8.335.1.38TEST_F	1373
8.335.1.39TEST_F	1373
8.335.1.40TEST_F	1373
8.335.1.41TEST_F	1373
8.335.1.42TEST_F	1373
8.335.1.43TEST_F	1373
8.335.1.44TEST_F	1373
8.335.1.45TEST_F	1373
8.335.1.46TEST_F	1373
8.335.1.47TEST_F	1374
8.335.1.48TEST_F	1374
8.335.1.49TEST_F	1374
8.335.1.50TEST_F	1374
8.335.1.51TEST_F	1374

8.335.1.52TEST_F	1374
8.335.1.53TEST_F	1374
8.335.1.54TEST_F	1374
8.335.1.55TEST_F	1374
8.335.1.56TEST_F	1374
8.335.1.57TEST_F	1374
8.335.1.58TEST_F	1374
8.335.1.59TEST_F	1375
8.335.1.60TEST_F	1375
8.335.1.61TEST_F	1375
8.335.1.62TEST_F	1375
8.335.1.63TEST_F	1375
8.335.1.64TEST_F	1375
8.335.1.65TEST_F	1375
8.335.1.66TEST_F	1375
8.335.1.67TEST_F	1375
8.335.1.68TEST_F	1375
8.335.1.69TEST_F	1375
8.335.1.70TEST_F	1375
8.335.1.71TEST_F	1376
8.335.1.72TEST_F	1376
8.335.1.73TEST_F	1376
8.335.1.74TEST_F	1376
8.335.1.75TEST_F	1376
8.335.1.76TEST_F	1376
8.335.1.77TEST_F	1376
8.335.1.78TEST_F	1376
8.335.1.79TEST_F	1376
8.335.1.80TEST_F	1376
8.335.1.81TEST_F	1376
8.335.1.82TEST_F	1376
8.335.1.83TEST_F	1377
8.335.1.84TEST_F	1377
8.335.1.85TEST_F	1377
8.335.1.86TEST_F	1377
8.335.1.87TEST_F	1377
8.335.1.88TEST_F	1377
8.335.1.89TEST_F	1377
8.335.1.90TEST_F	1377
8.335.1.91TEST_F	1377

8.335.1.92TEST_F	1377
8.335.1.93TEST_F	1377
8.335.1.94TEST_F	1377
8.335.1.95TEST_F	1378
8.335.1.96TEST_F	1378
8.335.1.97TEST_F	1378
8.335.1.98TEST_F	1378
8.335.1.99TEST_F	1378
8.335.1.100TEST_F	1378
8.335.1.101TEST_F	1378
8.335.1.102TEST_F	1378
8.335.1.103TEST_F	1378
8.335.1.104TEST_F	1378
8.335.1.105TEST_F	1378
8.335.1.106TEST_F	1378
8.335.1.107TEST_F	1379
8.335.1.108TEST_F	1379
8.335.1.109TEST_F	1379
8.335.1.110TEST_F	1379
8.335.1.111TEST_F	1379
8.335.1.112TEST_F	1379
8.335.1.113TEST_F	1379
8.335.1.114TEST_F	1379
8.335.1.115TEST_F	1379
8.335.1.116TEST_F	1379
8.335.1.117TEST_F	1379
8.335.1.118TEST_F	1379
8.335.1.119TEST_F	1380
8.335.1.120TEST_F	1380
8.335.1.121TEST_F	1380
8.335.1.122TEST_F	1380
8.335.1.123TEST_F	1380
8.335.1.124TEST_F	1380
8.335.1.125TEST_F	1380
8.335.1.126TEST_F	1380
8.335.1.127TEST_F	1380
8.335.1.128TEST_F	1380
8.335.1.129TEST_F	1380
8.335.1.130TEST_F	1380
8.335.1.131TEST_F	1381

8.335.1.13 TEST_F	1381
8.335.1.13 BEST_F	1381
8.335.1.13 TEST_F	1381
8.335.1.13 BEST_F	1381
8.335.1.13 TEST_F	1381
8.335.1.13 BEST_F	1381
8.335.1.13 TEST_F	1381
8.335.1.13 BEST_F	1381
8.335.1.13 TEST_F	1381
8.335.1.14 BEST_F	1382
8.335.1.14 TEST_F	1382
8.335.1.14 BEST_F	1382
8.335.1.14 TEST_F	1382
8.335.1.14 BEST_F	1382
8.335.1.14 TEST_F	1382
8.335.1.14 BEST_F	1382
8.335.1.14 TEST_F	1382
8.335.1.15 TEST_F	1382
8.335.1.15 TEST_F	1382
8.335.1.15 BEST_F	1382
8.335.1.15 TEST_F	1382
8.335.1.15 BEST_F	1383
8.335.1.15 TEST_F	1383
8.335.1.15 TEST_F	1383
8.335.1.15 BEST_F	1383
8.335.1.15 TEST_F	1383
8.335.1.16 TEST_F	1383
8.335.1.16 TEST_F	1383
8.335.1.16 BEST_F	1383
8.335.1.16 TEST_F	1383
8.335.1.16 BEST_F	1383
8.335.1.16 TEST_F	1383
8.335.1.16 BEST_F	1384
8.335.1.16 TEST_F	1384
8.335.1.16 BEST_F	1384
8.335.1.17 TEST_F	1384

8.335.1.17 P EST_F	1384
8.335.1.17BEST_F	1384
8.335.1.17 N EST_F	1384
8.335.1.17 S EST_F	1384
8.335.1.17 T EST_F	1384
8.335.1.17 V EST_F	1384
8.335.1.17 W EST_F	1384
8.335.1.17 Z EST_F	1385
8.335.1.18 D EST_F	1385
8.335.1.18 M EST_F	1385
8.335.1.18 P EST_F	1385
8.335.1.18 R EST_F	1385
8.335.1.18 T EST_F	1385
8.336/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/↔ NumericStateVariableTraceTest.hpp File Reference	1385
8.336.1 Function Documentation	1389
8.336.1.1 TEST_F	1389
8.336.1.2 TEST_F	1389
8.336.1.3 TEST_F	1389
8.336.1.4 TEST_F	1389
8.336.1.5 TEST_F	1389
8.336.1.6 TEST_F	1389
8.336.1.7 TEST_F	1390
8.336.1.8 TEST_F	1390
8.336.1.9 TEST_F	1390
8.336.1.10TEST_F	1390
8.336.1.11TEST_F	1390
8.336.1.12TEST_F	1390
8.336.1.13TEST_F	1390
8.336.1.14TEST_F	1390
8.336.1.15TEST_F	1390
8.336.1.16TEST_F	1390
8.336.1.17TEST_F	1390
8.336.1.18TEST_F	1390
8.336.1.19TEST_F	1391
8.336.1.20TEST_F	1391
8.336.1.21TEST_F	1391
8.336.1.22TEST_F	1391
8.336.1.23TEST_F	1391
8.336.1.24TEST_F	1391

8.336.1.25TEST_F	1391
8.336.1.26TEST_F	1391
8.336.1.27TEST_F	1391
8.336.1.28TEST_F	1391
8.336.1.29TEST_F	1391
8.336.1.30TEST_F	1391
8.336.1.31TEST_F	1392
8.336.1.32TEST_F	1392
8.336.1.33TEST_F	1392
8.336.1.34TEST_F	1392
8.336.1.35TEST_F	1392
8.336.1.36TEST_F	1392
8.336.1.37TEST_F	1392
8.336.1.38TEST_F	1392
8.336.1.39TEST_F	1392
8.336.1.40TEST_F	1392
8.336.1.41TEST_F	1392
8.336.1.42TEST_F	1392
8.336.1.43TEST_F	1393
8.336.1.44TEST_F	1393
8.336.1.45TEST_F	1393
8.336.1.46TEST_F	1393
8.336.1.47TEST_F	1393
8.336.1.48TEST_F	1393
8.336.1.49TEST_F	1393
8.336.1.50TEST_F	1393
8.336.1.51TEST_F	1393
8.336.1.52TEST_F	1393
8.336.1.53TEST_F	1393
8.336.1.54TEST_F	1393
8.336.1.55TEST_F	1394
8.336.1.56TEST_F	1394
8.336.1.57TEST_F	1394
8.336.1.58TEST_F	1394
8.336.1.59TEST_F	1394
8.336.1.60TEST_F	1394
8.336.1.61TEST_F	1394
8.336.1.62TEST_F	1394
8.336.1.63TEST_F	1394
8.336.1.64TEST_F	1394

8.336.1.65TEST_F	1394
8.336.1.66TEST_F	1394
8.336.1.67TEST_F	1395
8.336.1.68TEST_F	1395
8.336.1.69TEST_F	1395
8.336.1.70TEST_F	1395
8.336.1.71TEST_F	1395
8.336.1.72TEST_F	1395
8.336.1.73TEST_F	1395
8.336.1.74TEST_F	1395
8.336.1.75TEST_F	1395
8.336.1.76TEST_F	1395
8.336.1.77TEST_F	1395
8.336.1.78TEST_F	1395
8.336.1.79TEST_F	1396
8.336.1.80TEST_F	1396
8.336.1.81TEST_F	1396
8.336.1.82TEST_F	1396
8.336.1.83TEST_F	1396
8.336.1.84TEST_F	1396
8.336.1.85TEST_F	1396
8.336.1.86TEST_F	1396
8.336.1.87TEST_F	1396
8.336.1.88TEST_F	1396
8.336.1.89TEST_F	1396
8.336.1.90TEST_F	1396
8.336.1.91TEST_F	1397
8.336.1.92TEST_F	1397
8.336.1.93TEST_F	1397
8.336.1.94TEST_F	1397
8.336.1.95TEST_F	1397
8.336.1.96TEST_F	1397
8.336.1.97TEST_F	1397
8.336.1.98TEST_F	1397
8.336.1.99TEST_F	1397
8.336.1.100TEST_F	1397
8.336.1.101TEST_F	1397
8.336.1.102TEST_F	1397
8.336.1.103TEST_F	1398
8.336.1.104TEST_F	1398

8.336.1.10 BEST_F	1398
8.336.1.10 TEST_F	1398
8.336.1.10 TEST_F	1398
8.336.1.10 BEST_F	1398
8.336.1.10 TEST_F	1398
8.336.1.11 TEST_F	1398
8.336.1.11 TEST_F	1398
8.336.1.11 BEST_F	1398
8.336.1.11 TEST_F	1398
8.336.1.11 BEST_F	1398
8.336.1.11 TEST_F	1398
8.336.1.11 BEST_F	1399
8.336.1.11 BEST_F	1399
8.336.1.11 TEST_F	1399
8.336.1.11 BEST_F	1399
8.336.1.11 TEST_F	1399
8.336.1.12 TEST_F	1399
8.336.1.12 TEST_F	1399
8.336.1.12 BEST_F	1399
8.336.1.12 BEST_F	1399
8.336.1.12 TEST_F	1399
8.336.1.12 BEST_F	1399
8.336.1.12 TEST_F	1400
8.336.1.12 BEST_F	1400
8.336.1.12 TEST_F	1400
8.336.1.13 TEST_F	1400
8.336.1.13 BEST_F	1400
8.336.1.13 TEST_F	1400
8.336.1.13 BEST_F	1400
8.336.1.13 TEST_F	1400
8.336.1.13 BEST_F	1400
8.336.1.13 TEST_F	1401
8.336.1.14 TEST_F	1401
8.336.1.14 TEST_F	1401
8.336.1.14 BEST_F	1401
8.336.1.14 TEST_F	1401

8.336.1.14 BEST_F	1401
8.336.1.14 TEST_F	1401
8.336.1.14 TEST_F	1401
8.336.1.14 BEST_F	1401
8.336.1.14 TEST_F	1401
8.336.1.15 TEST_F	1401
8.336.1.15 TEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.15 BEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.15 BEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.15 BEST_F	1402
8.336.1.15 TEST_F	1402
8.336.1.16 BEST_F	1403
8.336.1.16 TEST_F	1403
8.336.1.16 BEST_F	1403
8.336.1.16 TEST_F	1403
8.336.1.16 BEST_F	1403
8.336.1.16 TEST_F	1403
8.336.1.17 TEST_F	1403
8.336.1.17 TEST_F	1403
8.336.1.17 BEST_F	1403
8.336.1.17 TEST_F	1403
8.336.1.17 BEST_F	1404
8.336.1.17 TEST_F	1404
8.336.1.17 BEST_F	1404
8.336.1.17 TEST_F	1404
8.336.1.18 TEST_F	1404
8.336.1.18 TEST_F	1404
8.336.1.18 BEST_F	1404
8.336.1.18 TEST_F	1404

8.336.1.18TEST_F	1404
8.336.1.18TEST_F	1404
8.336.1.18TEST_F	1405
8.337/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.cpp File Reference	1405
8.337.1 Function Documentation	1405
8.337.1.1 main	1405
8.338/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.hpp File Reference	1405
8.339/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File Reference	1405
8.339.1 Function Documentation	1409
8.339.1.1 TEST_F	1409
8.339.1.2 TEST_F	1409
8.339.1.3 TEST_F	1410
8.339.1.4 TEST_F	1410
8.339.1.5 TEST_F	1410
8.339.1.6 TEST_F	1410
8.339.1.7 TEST_F	1410
8.339.1.8 TEST_F	1410
8.339.1.9 TEST_F	1410
8.339.1.10TEST_F	1410
8.339.1.11TEST_F	1410
8.339.1.12TEST_F	1410
8.339.1.13TEST_F	1410
8.339.1.14TEST_F	1410
8.339.1.15TEST_F	1411
8.339.1.16TEST_F	1411
8.339.1.17TEST_F	1411
8.339.1.18TEST_F	1411
8.339.1.19TEST_F	1411
8.339.1.20TEST_F	1411
8.339.1.21TEST_F	1411
8.339.1.22TEST_F	1411
8.339.1.23TEST_F	1411
8.339.1.24TEST_F	1411
8.339.1.25TEST_F	1411
8.339.1.26TEST_F	1411
8.339.1.27TEST_F	1412
8.339.1.28TEST_F	1412
8.339.1.29TEST_F	1412

8.339.1.30TEST_F	1412
8.339.1.31TEST_F	1412
8.339.1.32TEST_F	1412
8.339.1.33TEST_F	1412
8.339.1.34TEST_F	1412
8.339.1.35TEST_F	1412
8.339.1.36TEST_F	1412
8.339.1.37TEST_F	1412
8.339.1.38TEST_F	1412
8.339.1.39TEST_F	1413
8.339.1.40TEST_F	1413
8.339.1.41TEST_F	1413
8.339.1.42TEST_F	1413
8.339.1.43TEST_F	1413
8.339.1.44TEST_F	1413
8.339.1.45TEST_F	1413
8.339.1.46TEST_F	1413
8.339.1.47TEST_F	1413
8.339.1.48TEST_F	1413
8.339.1.49TEST_F	1413
8.339.1.50TEST_F	1413
8.339.1.51TEST_F	1414
8.339.1.52TEST_F	1414
8.339.1.53TEST_F	1414
8.339.1.54TEST_F	1414
8.339.1.55TEST_F	1414
8.339.1.56TEST_F	1414
8.339.1.57TEST_F	1414
8.339.1.58TEST_F	1414
8.339.1.59TEST_F	1414
8.339.1.60TEST_F	1414
8.339.1.61TEST_F	1414
8.339.1.62TEST_F	1414
8.339.1.63TEST_F	1415
8.339.1.64TEST_F	1415
8.339.1.65TEST_F	1415
8.339.1.66TEST_F	1415
8.339.1.67TEST_F	1415
8.339.1.68TEST_F	1415
8.339.1.69TEST_F	1415

8.339.1.70TEST_F	1415
8.339.1.71TEST_F	1415
8.339.1.72TEST_F	1415
8.339.1.73TEST_F	1415
8.339.1.74TEST_F	1415
8.339.1.75TEST_F	1416
8.339.1.76TEST_F	1416
8.339.1.77TEST_F	1416
8.339.1.78TEST_F	1416
8.339.1.79TEST_F	1416
8.339.1.80TEST_F	1416
8.339.1.81TEST_F	1416
8.339.1.82TEST_F	1416
8.339.1.83TEST_F	1416
8.339.1.84TEST_F	1416
8.339.1.85TEST_F	1416
8.339.1.86TEST_F	1416
8.339.1.87TEST_F	1417
8.339.1.88TEST_F	1417
8.339.1.89TEST_F	1417
8.339.1.90TEST_F	1417
8.339.1.91TEST_F	1417
8.339.1.92TEST_F	1417
8.339.1.93TEST_F	1417
8.339.1.94TEST_F	1417
8.339.1.95TEST_F	1417
8.339.1.96TEST_F	1417
8.339.1.97TEST_F	1417
8.339.1.98TEST_F	1417
8.339.1.99TEST_F	1418
8.339.1.100TEST_F	1418
8.339.1.101TEST_F	1418
8.339.1.102TEST_F	1418
8.339.1.103TEST_F	1418
8.339.1.104TEST_F	1418
8.339.1.105TEST_F	1418
8.339.1.106TEST_F	1418
8.339.1.107TEST_F	1418
8.339.1.108TEST_F	1418
8.339.1.109TEST_F	1418

8.341/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/← TraceEvaluationTest.hpp File Reference	1425
8.341.1 Variable Documentation	1426
8.341.1.1 ERR_MSG_TEST	1426
8.342/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/← InputStringParser.hpp File Reference	1426
8.343/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/← ParserTest.cpp File Reference	1427
8.343.1 Function Documentation	1427
8.343.1.1 main	1427
8.344/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/← ParserTest.hpp File Reference	1427
8.344.1 Function Documentation	1438
8.344.1.1 TEST	1438
8.344.1.2 TEST	1438
8.344.1.3 TEST	1438
8.344.1.4 TEST	1438
8.344.1.5 TEST	1438
8.344.1.6 TEST	1438
8.344.1.7 TEST	1439
8.344.1.8 TEST	1439
8.344.1.9 TEST	1439
8.344.1.10TEST	1439
8.344.1.11TEST	1439
8.344.1.12TEST	1439
8.344.1.13TEST	1439
8.344.1.14TEST	1439
8.344.1.15TEST	1439
8.344.1.16TEST	1440
8.344.1.17TEST	1440
8.344.1.18TEST	1440
8.344.1.19TEST	1440
8.344.1.20TEST	1440
8.344.1.21TEST	1440
8.344.1.22TEST	1440
8.344.1.23TEST	1440
8.344.1.24TEST	1440
8.344.1.25TEST	1441
8.344.1.26TEST	1441
8.344.1.27TEST	1441
8.344.1.28TEST	1441

8.344.1.29TEST	1441
8.344.1.30TEST	1441
8.344.1.31TEST	1441
8.344.1.32TEST	1441
8.344.1.33TEST	1441
8.344.1.34TEST	1442
8.344.1.35TEST	1442
8.344.1.36TEST	1442
8.344.1.37TEST	1442
8.344.1.38TEST	1442
8.344.1.39TEST	1442
8.344.1.40TEST	1442
8.344.1.41TEST	1442
8.344.1.42TEST	1442
8.344.1.43TEST	1443
8.344.1.44TEST	1443
8.344.1.45TEST	1443
8.344.1.46TEST	1443
8.344.1.47TEST	1443
8.344.1.48TEST	1443
8.344.1.49TEST	1443
8.344.1.50TEST	1443
8.344.1.51TEST	1443
8.344.1.52TEST	1444
8.344.1.53TEST	1444
8.344.1.54TEST	1444
8.344.1.55TEST	1444
8.344.1.56TEST	1444
8.344.1.57TEST	1444
8.344.1.58TEST	1444
8.344.1.59TEST	1444
8.344.1.60TEST	1444
8.344.1.61TEST	1445
8.344.1.62TEST	1445
8.344.1.63TEST	1445
8.344.1.64TEST	1445
8.344.1.65TEST	1445
8.344.1.66TEST	1445
8.344.1.67TEST	1445
8.344.1.68TEST	1445

8.344.1.69TEST	1445
8.344.1.70TEST	1446
8.344.1.71TEST	1446
8.344.1.72TEST	1446
8.344.1.73TEST	1446
8.344.1.74TEST	1446
8.344.1.75TEST	1446
8.344.1.76TEST	1446
8.344.1.77TEST	1446
8.344.1.78TEST	1446
8.344.1.79TEST	1447
8.344.1.80TEST	1447
8.344.1.81TEST	1447
8.344.1.82TEST	1447
8.344.1.83TEST	1447
8.344.1.84TEST	1447
8.344.1.85TEST	1447
8.344.1.86TEST	1447
8.344.1.87TEST	1447
8.344.1.88TEST	1448
8.344.1.89TEST	1448
8.344.1.90TEST	1448
8.344.1.91TEST	1448
8.344.1.92TEST	1448
8.344.1.93TEST	1448
8.344.1.94TEST	1448
8.344.1.95TEST	1448
8.344.1.96TEST	1448
8.344.1.97TEST	1449
8.344.1.98TEST	1449
8.344.1.99TEST	1449
8.344.1.100TEST	1449
8.344.1.101TEST	1449
8.344.1.102TEST	1449
8.344.1.103TEST	1449
8.344.1.104TEST	1449
8.344.1.105TEST	1450
8.344.1.106TEST	1450
8.344.1.107TEST	1450
8.344.1.108TEST	1450

8.344.1.10 BEST	1450
8.344.1.11 D EST	1450
8.344.1.11 T EST	1450
8.344.1.11 B EST	1450
8.344.1.11 TEST	1450
8.344.1.11 BEST	1451
8.344.1.11 T EST	1451
8.344.1.11 B EST	1451
8.344.1.11 TEST	1451
8.344.1.11 BEST	1451
8.344.1.11 T EST	1451
8.344.1.11 B EST	1451
8.344.1.11 TEST	1451
8.344.1.12 T EST	1451
8.344.1.12 P EST	1451
8.344.1.12 B EST	1451
8.344.1.12 TEST	1451
8.344.1.12 BEST	1452
8.344.1.12 T EST	1452
8.344.1.12 P EST	1452
8.344.1.12 B EST	1452
8.344.1.12 TEST	1452
8.344.1.12 BEST	1452
8.344.1.12 T EST	1452
8.344.1.12 P EST	1452
8.344.1.12 B EST	1452
8.344.1.12 TEST	1452
8.344.1.13 T EST	1452
8.344.1.13 P EST	1452
8.344.1.13 B EST	1452
8.344.1.13 TEST	1452
8.344.1.13 BEST	1453
8.344.1.13 T EST	1453
8.344.1.13 P EST	1453
8.344.1.13 B EST	1453
8.344.1.13 TEST	1453
8.344.1.13 BEST	1453
8.344.1.13 T EST	1453
8.344.1.13 P EST	1453
8.344.1.13 B EST	1453
8.344.1.13 TEST	1453
8.344.1.14 T EST	1453
8.344.1.14 P EST	1454
8.344.1.14 B EST	1454
8.344.1.14 TEST	1454
8.344.1.14 BEST	1454
8.344.1.14 T EST	1454
8.344.1.14 P EST	1454
8.344.1.14 B EST	1454
8.344.1.14 TEST	1454

8.344.1.18 TEST	1459
8.344.1.19 TEST	1460
8.344.1.19 TEST	1461
8.344.1.19 TEST	1462
8.344.1.19 TEST	1463

8.344.1.26 TEST	1468
8.344.1.27 TEST	1469
8.344.1.27 TEST	1469
8.344.1.27 TEST	1469
8.344.1.28 TEST	1470
8.344.1.29 TEST	1471
8.344.1.30 TEST	1472

8.344.1.30 BEST	1472
8.344.1.31 TEST	1472
8.344.1.31 TEST	1472
8.344.1.31 BEST	1473
8.344.1.31 TEST	1473
8.344.1.31 BEST	1473
8.344.1.31 TEST	1473
8.344.1.31 BEST	1473
8.344.1.31 TEST	1473
8.344.1.31 BEST	1473
8.344.1.31 TEST	1473
8.344.1.31 BEST	1473
8.344.1.31 TEST	1473
8.344.1.32 TEST	1473
8.344.1.32 TEST	1474
8.344.1.32 BEST	1474
8.344.1.32 TEST	1474
8.344.1.32 BEST	1474
8.344.1.32 TEST	1474
8.344.1.32 BEST	1474
8.344.1.32 TEST	1474
8.344.1.33 TEST	1475
8.344.1.33 TEST	1475
8.344.1.33 BEST	1475
8.344.1.33 TEST	1475
8.344.1.33 BEST	1475
8.344.1.33 TEST	1475
8.344.1.33 BEST	1475
8.344.1.33 TEST	1475
8.344.1.33 BEST	1475
8.344.1.34 TEST	1476
8.344.1.34 TEST	1476
8.344.1.34 BEST	1476
8.344.1.34 TEST	1476
8.344.1.34 BEST	1476
8.344.1.34 TEST	1476
8.344.1.34 BEST	1476

8.344.1.34 TEST	1477
8.344.1.35 TEST	1478
8.344.1.35 TEST	1478
8.344.1.36 TEST	1479
8.344.1.36 TEST	1479
8.344.1.36 TEST	1479
8.344.1.37 TEST	1480
8.344.1.38 TEST	1481

8.344.1.38 TEST	1481
8.344.1.39 TEST	1482
8.344.1.40 TEST	1483
8.344.1.41 TEST	1484
8.344.1.42 TEST	1485

8.344.1.46 TEST	1490
8.344.1.47 TEST	1491
8.344.1.48 TEST	1492
8.344.1.49 TEST	1492
8.344.1.49 TEST	1492
8.344.1.49 TEST	1493
8.344.1.50 TEST	1493
8.344.1.50 TEST	1494

8.344.1.54 TEST	1499
8.344.1.55 TEST	1500
8.344.1.56 TEST	1501
8.344.1.56 TEST	1501
8.344.2 Variable Documentation	1501
8.344.2.1 CONSTRAINTS_BINARY_OPERATORS	1501
8.344.2.2 LOGIC_PROPERTIES_BINARY_OPERATORS	1501
8.345/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/← AnnularSector.hpp File Reference	1501
8.346/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/← CartesianToPolarConverter.hpp File Reference	1502
8.347/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/← PolarCsvToInputFilesConverter.hpp File Reference	1503
8.348/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/← PolarGnuplotScriptGenerator.hpp File Reference	1503
8.349/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/AnnularSector.cpp File Reference	1504
8.349.1 Variable Documentation	1505
8.349.1.1 SEPARATOR	1505
8.350/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/CartesianToPolar← Converter.cpp File Reference	1505
8.351/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/MapCartesianTo← PolarScript.cpp File Reference	1506
8.351.1 Function Documentation	1506
8.351.1.1 areValidParameters	1507
8.351.1.2 initArgumentsConfig	1507
8.351.1.3 isValidOutputType	1507
8.351.1.4 main	1507

8.351.1.5 printHelpInformation	1507
8.351.1.6 printWrongParameters	1507
8.352/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarCsvToInputFilesConverter.cpp File Reference	1507
8.353/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarGnuplotScriptGenerator.cpp File Reference	1508
8.354/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarMapCsvToInputFiles.cpp File Reference	1509
8.354.1 Function Documentation	1509
8.354.1.1 areValidParameters	1509
8.354.1.2 initArgumentsConfig	1510
8.354.1.3 isValidNrOfConcentrationsForPosition	1510
8.354.1.4 main	1510
8.354.1.5 printHelpInformation	1510
8.354.1.6 printWrongParameters	1510
8.354.1.7 setLogScaling	1510
8.354.1.8 setNumberIteratorType	1510
8.354.1.9 setSelectedConcentrationIndex	1510
8.355/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/CartesianToConcentrationsConverter.hpp File Reference	1511
8.356/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularCsvToInputFilesConverter.hpp File Reference	1511
8.357/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularEntityCsvToInputFilesConverter.hpp File Reference	1512
8.358/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularGnuplotScriptGenerator.hpp File Reference	1513
8.359/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/CartesianToConcentrationsConverter.cpp File Reference	1513
8.360/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/MapCartesianToScript.cpp File Reference	1514
8.360.1 Function Documentation	1515
8.360.1.1 areValidParameters	1515
8.360.1.2 initArgumentsConfig	1515
8.360.1.3 main	1515
8.360.1.4 printHelpInformation	1515
8.360.1.5 printWrongParameters	1515
8.361/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularCsvToInputFilesConverter.cpp File Reference	1515
8.362/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularEntityCsvToInputFilesConverter.cpp File Reference	1516
8.363/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularGnuplotScriptGenerator.cpp File Reference	1517
8.364/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularMapCsvToInputFiles.cpp File Reference	1517

8.364.1 Function Documentation	1518
8.364.1.1 areValidParameters	1518
8.364.1.2 initArgumentsConfig	1518
8.364.1.3 isValidNrOfConcentrationsForPosition	1518
8.364.1.4 main	1519
8.364.1.5 printHelpInformation	1519
8.364.1.6 printWrongParameters	1519
8.364.1.7 setLogScaling	1519
8.364.1.8 setNumberIteratorType	1519
8.364.1.9 setSelectedConcentrationIndex	1519
8.365/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/Rectangular← MapEntityCsvToInputFiles.cpp File Reference	1519
8.365.1 Function Documentation	1520
8.365.1.1 areValidParameters	1520
8.365.1.2 initArgumentsConfig	1520
8.365.1.3 main	1520
8.365.1.4 printHelpInformation	1521
8.365.1.5 printWrongParameters	1521
8.365.1.6 setNumberIteratorType	1521

Chapter 1

Multiscale

1.1 Brief description

The "Multiscale" software is a multiscale model checker implemented during the PhD research project carried out by Ovidiu Parvu, Brunel University, London, United Kingdom, October 2012 - present.

1.2 Contact

For more information, comments, recommendations or suggestions please visit the author's [institutional web page](#), where contact details are provided.

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

multiscale	35
multiscale::analysis	39
multiscale::verification	41
multiscale::verification::spatialmeasure	64
multiscale::verification::subsetsspecific	67
multiscale::video	68
multiscaletest	69
multiscaletest::verification	69

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

multiscale::verification::AbstractSyntaxTree	71
multiscale::AdditionOperation	75
multiscale::verification::AndConstraintAttribute	78
multiscale::verification::AndLogicPropertyAttribute	79
multiscale::video::AnnularSector	80
multiscale::verification::BinaryNumericFilterAttribute	143
multiscale::verification::BinaryNumericMeasureAttribute	145
multiscale::verification::BinaryNumericNumericAttribute	153
multiscale::verification::BinaryNumericTemporalAttribute	154
multiscale::verification::BinaryStatisticalMeasureAttribute	155
multiscale::verification::BinaryStatisticalNumericAttribute	163
multiscale::verification::BinaryStatisticalQuantileMeasureAttribute	164
multiscale::verification::BinaryStatisticalQuantileNumericAttribute	172
multiscale::verification::BinaryStatisticalQuantileSpatialAttribute	173
multiscale::verification::BinaryStatisticalSpatialAttribute	174
multiscale::video::CartesianToConcentrationsConverter	181
multiscale::video::CartesianToPolarConverter	186
multiscale::verification::ChangeMeasureAttribute	192
multiscale::verification::ChangeMeasureEvaluator	193
multiscale::verification::ChangeTemporalNumericCollectionAttribute	202
multiscale::verification::ChangeTemporalNumericMeasureAttribute	203
multiscale::analysis::CircularityMeasure	205
multiscale::verification::CommandLineModelChecking	236
multiscale::verification::ComparatorAttribute	272
multiscale::verification::ComparatorEvaluator	272
multiscale::ConsolePrinter	283
multiscale::verification::ConstraintAttribute	292
multiscale::analysis::DataPoint	307
EuclideanDataPoint	353
multiscale::analysis::Entity	344
multiscale::analysis::DBSCAN	310
multiscale::analysis::Detector	316
multiscale::analysis::ClusterDetector	224
multiscale::analysis::SimulationClusterDetector	795
multiscale::analysis::RegionDetector	744
multiscale::Distribution	336
multiscale::BetaDistribution	138

multiscale::BinomialDistribution	176
multiscale::DivisionOperation	340
multiscale::verification::EquivalenceConstraintAttribute	351
multiscale::verification::EquivalenceLogicPropertyAttribute	351
ErrorHandler	
multiscale::XmlValidator::XmlValidationErrorHandler	1044
std::exception	
std::runtime_error	
multiscale::MultiscaleException	548
multiscale::AlgorithmException	76
multiscale::UnexpectedBehaviourException	1032
multiscale::verification::ModelCheckingException	508
multiscale::verification::ModelCheckingHelpRequestException	511
multiscale::verification::SpatialTemporalException	863
multiscale::UnimplementedMethodException	1039
multiscale::IOException	410
multiscale::FileOpenException	357
multiscale::InvalidInputException	407
multiscale::NumericException	592
multiscale::RuntimeException	772
multiscale::IndexOutOfBoundsException	404
multiscale::TestException	962
multiscale::ExceptionHandler	355
multiscale::Filesystem	360
multiscale::verification::FilterNumericMeasureAttribute	364
multiscale::verification::FilterSubsetAttribute	369
multiscale::verification::FutureLogicPropertyAttribute	371
multiscale::Geometry2D	372
multiscale::verification::GlobalLogicPropertyAttribute	388
grammar	
multiscale::verification::BinaryNumericMeasureGrammar< std::string::const_iterator >	146
multiscale::verification::BinaryStatisticalMeasureGrammar< std::string::const_iterator >	156
multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< std::string::const_iterator >	165
multiscale::verification::ChangeMeasureGrammar< std::string::const_iterator >	196
multiscale::verification::ComparatorGrammar< std::string::const_iterator >	274
multiscale::verification::LogicPropertyGrammar< std::string::const_iterator >	427
multiscale::verification::BinaryNumericMeasureGrammar< Iterator >	146
multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >	156
multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >	165
multiscale::verification::ChangeMeasureGrammar< Iterator >	196
multiscale::verification::ComparatorGrammar< Iterator >	274
multiscale::verification::LogicPropertyGrammar< Iterator >	427
multiscale::verification::NumericStateVariableGrammar< Iterator >	614
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >	678
multiscale::verification::SemanticTypeGrammar< Iterator >	778
multiscale::verification::SemanticTypeStringGrammar< Iterator >	782
multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >	831
multiscale::verification::TemporalNumericCollectionGrammar< Iterator >	930
multiscale::verification::TemporalNumericMeasureGrammar< Iterator >	944
multiscale::verification::UnaryNumericMeasureGrammar< Iterator >	1010
multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >	1021
multiscale::verification::NumericStateVariableGrammar< std::string::const_iterator >	614
multiscale::verification::SemanticTypeGrammar< std::string::const_iterator >	778
multiscale::verification::SemanticTypeStringGrammar< std::string::const_iterator >	782
multiscale::verification::SpatialMeasureCollectionGrammar< std::string::const_iterator >	831
multiscale::verification::TemporalNumericCollectionGrammar< std::string::const_iterator >	930
multiscale::verification::TemporalNumericMeasureGrammar< std::string::const_iterator >	944
multiscale::verification::UnaryNumericMeasureGrammar< std::string::const_iterator >	1010

multiscale::verification::UnaryStatisticalMeasureGrammar< std::string::const_iterator >	1021
multiscale::verification::HeterogeneousTimeseriesComponentAttribute	389
multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >	393
multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute	395
multiscale::verification::HomogeneousTimeseriesComponentAttribute	396
multiscale::verification::HomogeneousTimeseriesMeasureAttribute	400
multiscale::verification::ImplicationConstraintAttribute	403
multiscale::verification::ImplicationLogicPropertyAttribute	403
multiscale::verification::LogicPropertyAttribute	420
multiscale::verification::LogicPropertyDataReader	422
multiscale::analysis::MatFactory	457
multiscale::analysis::CircularMatFactory	206
multiscale::analysis::RectangularMatFactory	727
multiscale::MinEnclosingTriangleFinder	463
multiscale::verification::ModelChecker	489
multiscale::verification::ApproximateBayesianModelChecker	83
multiscale::verification::ApproximateProbabilisticModelChecker	102
multiscale::verification::BayesianModelChecker	117
multiscale::verification::ProbabilisticBlackBoxModelChecker	686
multiscale::verification::StatisticalModelChecker	882
multiscale::verification::ModelCheckerFactory	501
multiscale::verification::ApproximateBayesianModelCheckerFactory	95
multiscale::verification::ApproximateProbabilisticModelCheckerFactory	110
multiscale::verification::BayesianModelCheckerFactory	131
multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory	690
multiscale::verification::StatisticalModelCheckerFactory	895
multiscale::verification::ModelCheckingManager	514
multiscale::verification::ModelCheckingOutputWriter	529
multiscale::MultiplicationOperation	547
multiscale::verification::NextKLogicPropertyAttribute	553
multiscale::verification::NextLogicPropertyAttribute	554
multiscale::verification::Nil	555
multiscale::verification::NotConstraintAttribute	556
multiscale::verification::NotLogicPropertyAttribute	557
multiscale::NumberIterator	558
multiscale::LexicographicNumberIterator	413
multiscale::StandardNumberIterator	876
multiscale::Numeric	562
multiscale::verification::NumericEvaluator	589
multiscale::verification::NumericMeasureAttribute	594
multiscale::verification::NumericMeasureCollectionAttribute	595
multiscale::verification::NumericMeasureCollectionEvaluator	596
multiscale::NumericRangeManipulator	610
multiscale::verification::NumericSpatialMeasureAttribute	611
multiscale::verification::NumericStateVariableAttribute	612
multiscale::verification::NumericStateVariableId	619
multiscale::verification::NumericStatisticalMeasureAttribute	626
multiscale::OperatingSystem	635
multiscale::verification::OrConstraintAttribute	639
multiscale::verification::OrLogicPropertyAttribute	640
multiscale::verification::Parser	641
multiscale::verification::ParserGrammarExceptionHandler	645
multiscale::verification::ParserGrammarExtraInputException	648
multiscale::verification::ParserGrammarProbabilityException	650
multiscale::verification::ParserGrammarUnexpectedTokenException	653
multiscale::verification::ParserGrammarUnparseableInputException	655
multiscale::video::PolarCsvToInputFilesConverter	656

multiscale::video::PolarGnuplotScriptGenerator	668
multiscale::verification::PrimaryConstraintAttribute	676
multiscale::verification::PrimaryLogicPropertyAttribute	676
multiscale::verification::PrimaryNumericMeasureAttribute	677
multiscale::verification::ProbabilisticLogicPropertyAttribute	696
multiscale::verification::ProbabilityErrorHandler	699
multiscale::video::RectangularCsvToInputFilesConverter	701
multiscale::video::RectangularEntityCsvToInputFilesConverter	712
multiscale::video::RectangularGnuplotScriptGenerator	722
multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >	766
multiscale::verification::UnexpectedTokenErrorHandler::result< typename, typename, typename >	767
multiscale::RGBColourGenerator	768
multiscale::verification::SemanticType	775
multiscale::verification::SemanticTypeAttribute	777
multiscale::analysis::Silhouette	786
multiscale::verification::SimilarityMeasureAttribute	791
multiscale::verification::SimilarityTemporalNumericCollectionAttribute	794
multiscale::verification::SpatialEntity	809
multiscale::verification::Cluster	212
multiscale::verification::Region	733
multiscale::analysis::SpatialEntityPseudo3D	815
multiscale::analysis::Cluster	215
multiscale::analysis::Region	735
multiscale::verification::SpatialMeasureAttribute	828
multiscale::verification::SpatialMeasureCollectionAttribute	829
multiscale::verification::SpatialMeasureEvaluator	845
multiscale::verification::SpatialNumericComparisonAttribute	848
multiscale::verification::SpatialTemporalDataReader	849
multiscale::verification::SpatialTemporalTrace	866
multiscale::verification::StateVariableAttribute	880
static_visitor	
multiscale::verification::ConstraintVisitor	294
multiscale::verification::FilterNumericVisitor	365
multiscale::verification::LogicPropertyVisitor	440
multiscale::verification::NumericMeasureCollectionVisitor	598
multiscale::verification::NumericVisitor	627
multiscale::verification::SubsetVisitor	915
multiscale::verification::TemporalNumericVisitor	953
multiscale::verification::TimeseriesComponentVisitor	982
multiscale::StringManipulator	902
multiscale::verification::SubsetAttribute	908
multiscale::verification::SubsetOperationAttribute	909
multiscale::verification::SubsetSpecificAttribute	911
multiscale::verification::SubsetSubsetOperationAttribute	913
multiscale::SubtractionOperation	922
symbols	
multiscale::verification::BinaryNumericMeasureTypeParser	150
multiscale::verification::BinaryStatisticalMeasureTypeParser	160
multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser	169
multiscale::verification::ChangeMeasureTypeParser	200
multiscale::verification::ComparatorNonEqualTypeParser	277
multiscale::verification::ComparatorTypeParser	279
multiscale::verification::HeterogeneousTimeseriesComponentTypeParser	390
multiscale::verification::HomogeneousTimeseriesComponentTypeParser	397
multiscale::verification::HomogeneousTimeseriesMeasureTypeParser	400
multiscale::verification::SimilarityMeasureTypeParser	792
multiscale::verification::SpatialMeasureTypeParser	846
multiscale::verification::SubsetOperationTypeParser	909

multiscale::verification::SubsetSpecificTypeParser	912
multiscale::verification::TimeseriesMeasureTypeParser	987
multiscale::verification::UnaryNumericMeasureTypeParser	1014
multiscale::verification::UnaryStatisticalMeasureTypeParser	1025
multiscale::verification::TemporalDataReader	922
multiscale::verification::TemporalNumericCollectionAttribute	930
multiscale::verification::TemporalNumericComparisonAttribute	941
multiscale::verification::TemporalNumericMeasureAttribute	942
multiscale::verification::TemporalNumericMeasureCollectionAttribute	943
Test	
multiscaletest::MultiscaleTest	551
multiscaletest::MinEnclosingTriangleFinderTest	482
multiscaletest::ModelCheckerTest	502
multiscaletest::ApproximateBayesianModelCheckerTest	98
multiscaletest::ApproximateProbabilisticModelCheckerTest	114
multiscaletest::BayesianModelCheckerTest	134
multiscaletest::ProbabilisticBlackBoxModelCheckerTest	693
multiscaletest::StatisticalModelCheckerTest	898
multiscaletest::TraceEvaluationTest	990
multiscaletest::CompleteTraceTest	280
multiscaletest::EmptyTraceTest	341
multiscaletest::NumericStateVariableTraceTest	623
multiscaletest::SpatialEntitiesTraceTest	806
multiscale::verification::TimePoint	965
multiscale::verification::TimePointEvaluator	976
multiscale::verification::TimeseriesComponentAttribute	979
multiscale::verification::TimeseriesComponentEvaluator	980
multiscale::verification::TimeseriesMeasureAttribute	986
multiscale::verification::TimeseriesTimeseriesComponentAttribute	989
multiscale::verification::TypeSemanticsTable	998
multiscale::verification::UnaryNumericFilterAttribute	1008
multiscale::verification::UnaryNumericMeasureAttribute	1009
multiscale::verification::UnaryNumericNumericAttribute	1016
multiscale::verification::UnaryNumericTemporalAttribute	1017
multiscale::verification::UnarySpatialConstraintAttribute	1019
multiscale::verification::UnaryStatisticalMeasureAttribute	1020
multiscale::verification::UnaryStatisticalNumericAttribute	1028
multiscale::verification::UnaryStatisticalSpatialAttribute	1029
multiscale::verification::UnaryTypeConstraintAttribute	1030
multiscale::verification::UnexpectedTokenErrorHandler	1035
multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >	1037
multiscale::verification::UntilLogicPropertyAttribute	1042
multiscale::UserDefinedTypeName< T >	1043
multiscale::XmlValidator	1049

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

multiscale::verification::AbstractSyntaxTree	Class used for representing an abstract syntax tree	71
multiscale::AdditionOperation	Functor representing an addition operation	75
multiscale::AlgorithmException	Class for representing algorithm exceptions	76
multiscale::verification::AndConstraintAttribute	Class for representing an "and" constraint attribute	78
multiscale::verification::AndLogicPropertyAttribute	Class for representing an "and" logic property attribute	79
multiscale::video::AnnularSector	An annular sector is the basic element in the considered circular geometry	80
multiscale::verification::ApproximateBayesianModelChecker	Class used to run approximate Bayesian model checking tasks	83
multiscale::verification::ApproximateBayesianModelCheckerFactory	Class for creating ApproximateBayesianModelChecker instances	95
multiscaletest::ApproximateBayesianModelCheckerTest	Class for testing the approximate Bayesian model checker	98
multiscale::verification::ApproximateProbabilisticModelChecker	Class used to run approximate probabilistic model checking tasks	102
multiscale::verification::ApproximateProbabilisticModelCheckerFactory	Class for creating ApproximateProbabilisticModelChecker instances	110
multiscaletest::ApproximateProbabilisticModelCheckerTest	Class for testing the approximate probabilistic model checker	114
multiscale::verification::BayesianModelChecker	Class used to run Bayesian model checking tasks	117
multiscale::verification::BayesianModelCheckerFactory	Class for creating BayesianModelChecker instances	131
multiscaletest::BayesianModelCheckerTest	Class for testing the Bayesian model checker	134
multiscale::BetaDistribution	Class for analysing Beta distributed data	138
multiscale::verification::BinaryNumericFilterAttribute	Class for representing a binary numeric filter attribute	143
multiscale::verification::BinaryNumericMeasureAttribute	Class for representing a binary numeric measure attribute	145
multiscale::verification::BinaryNumericMeasureGrammar< Iterator >	The grammar for parsing binary numeric measure statements	146

multiscale::verification::BinaryNumericMeasureTypeParser	Symbol table and parser for the binary numeric measure type	150
multiscale::verification::BinaryNumericNumericAttribute	Class for representing a binary numeric numeric measure attribute	153
multiscale::verification::BinaryNumericTemporalAttribute	Class for representing a binary numeric temporal measure attribute	154
multiscale::verification::BinaryStatisticalMeasureAttribute	Class for representing a binary statistical measure attribute	155
multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >	The grammar for parsing binary statistical measure statements	156
multiscale::verification::BinaryStatisticalMeasureTypeParser	Symbol table and parser for the binary statistical measure type	160
multiscale::verification::BinaryStatisticalNumericAttribute	Class for representing a binary statistical numeric attribute	163
multiscale::verification::BinaryStatisticalQuantileMeasureAttribute	Class for representing a binary statistical quantile measure attribute	164
multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >	The grammar for parsing binary statistical quantile measure statements	165
multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser	Symbol table and parser for the binary statistical quantile measure type	169
multiscale::verification::BinaryStatisticalQuantileNumericAttribute	Class for representing a binary statistical quantile numeric attribute	172
multiscale::verification::BinaryStatisticalQuantileSpatialAttribute	Class for representing a binary statistical quantile spatial attribute	173
multiscale::verification::BinaryStatisticalSpatialAttribute	Class for representing a binary statistical spatial attribute	174
multiscale::BinomialDistribution	Class for analysing Binomial distributed data	176
multiscale::video::CartesianToConcentrationsConverter	Scale the values of the rectangular geometry grid cells	181
multiscale::video::CartesianToPolarConverter	Converter from the rectangular geometry grid cells to annular sectors	186
multiscale::verification::ChangeMeasureAttribute	Class for representing a change measure attribute	192
multiscale::verification::ChangeMeasureEvaluator	Class for evaluating change measure expressions	193
multiscale::verification::ChangeMeasureGrammar< Iterator >	The grammar for parsing change measure statements	196
multiscale::verification::ChangeMeasureTypeParser	Symbol table and parser for the change measure type	200
multiscale::verification::ChangeTemporalNumericCollectionAttribute	Class for representing a change temporal numeric collection attribute	202
multiscale::verification::ChangeTemporalNumericMeasureAttribute	Class for representing a change temporal numeric measure attribute	203
multiscale::analysis::CircularityMeasure	Class for computing the circularity measure for the given collection of points	205
multiscale::analysis::CircularMatFactory	Class for creating a Mat object considering a circular grid	206
multiscale::verification::Cluster	Class for representing a cluster	212
multiscale::analysis::Cluster	Class for representing a cluster of entities in an image	215
multiscale::analysis::ClusterDetector	Class for detecting clusters in 2D images	224
multiscale::verification::CommandLineModelChecking	Class for running model checkers from the command line	236
multiscale::verification::ComparatorAttribute	Class for representing a comparator attribute	272

multiscale::verification::ComparatorEvaluator	Class for evaluating comparison expressions	272
multiscale::verification::ComparatorGrammar< Iterator >	The grammar for parsing comparator statements	274
multiscale::verification::ComparatorNonEqualTypeParser	Symbol table and parser for the comparator type which does not accept the "=" symbol	277
multiscale::verification::ComparatorTypeParser	Symbol table and parser for the comparator type	279
multiscaletest::CompleteTraceTest	Class for testing evaluation of complete traces containing both numeric state variables and spatial entities	280
multiscale::ConsolePrinter	Class used to print (coloured) messages to the console	283
multiscale::verification::ConstraintAttribute	Class for representing a constraint attribute	292
multiscale::verification::ConstraintVisitor	Class used to evaluate constraints	294
multiscale::analysis::DataPoint	Class for representing a data point	307
multiscale::analysis::DBSCAN	Class which implements an improved version of the DBSCAN algorithm	310
multiscale::analysis::Detector	Abstract class for detecting entities of interest in images	316
multiscale::Distribution		336
multiscale::DivisionOperation	Functor representing a division operation	340
multiscaletest::EmptyTraceTest	Class for testing evaluation of empty traces	341
multiscale::analysis::Entity	Class for representing an entity in an image (e.g. cell, organism etc.)	344
multiscale::verification::EquivalenceConstraintAttribute	Class for representing an "equivalence" constraint attribute	351
multiscale::verification::EquivalenceLogicPropertyAttribute	Class for representing an "equivalence" logic property attribute	351
EuclideanDataPoint		353
multiscale::ExceptionHandler	Exception handler class	355
multiscale::FileOpenException	Class for representing exceptions when opening a file	357
multiscale::Filesystem	Class containing methods for interacting with the filesystem	360
multiscale::verification::FilterNumericMeasureAttribute	Class for representing a filter numeric measure	364
multiscale::verification::FilterNumericVisitor	Class for evaluating filter numeric measures	365
multiscale::verification::FilterSubsetAttribute	Class for representing a filter subset attribute	369
multiscale::verification::FutureLogicPropertyAttribute	Class for representing a "future" logic property attribute	371
multiscale::Geometry2D	Two-dimensional geometric operations	372
multiscale::verification::GlobalLogicPropertyAttribute	Class for representing a "globally" logic property attribute	388
multiscale::verification::HeterogeneousTimeseriesComponentAttribute	Class for representing a heterogeneous timeseries component attribute	389
multiscale::verification::HeterogeneousTimeseriesComponentTypeParser	Symbol table and parser for the heterogeneous timeseries component type	390

multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator	< Relation >	393
multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute	Class for representing a homogeneous homogeneous timeseries attribute	395
multiscale::verification::HomogeneousTimeseriesComponentAttribute	Class for representing a homogeneous timeseries component attribute	396
multiscale::verification::HomogeneousTimeseriesComponentTypeParser	Symbol table and parser for the homogeneous timeseries component type	397
multiscale::verification::HomogeneousTimeseriesMeasureAttribute	Class for representing a homogeneous timeseries measure attribute	400
multiscale::verification::HomogeneousTimeseriesMeasureTypeParser	Symbol table and parser for the homogeneous timeseries measure type	400
multiscale::verification::ImplicationConstraintAttribute	Class for representing an "implication" constraint attribute	403
multiscale::verification::ImplicationLogicPropertyAttribute	Class for representing an "implication" logic property attribute	403
multiscale::IndexOutOfBoundsException	Class for representing an index out of bounds exception	404
multiscale::InvalidInputException	Class for representing invalid input exceptions	407
multiscale::IOException	Class for representing input and output exceptions	410
multiscale::LexicographicNumberIterator	Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "_"	413
multiscale::verification::LogicPropertyAttribute	Class for representing a logic property attribute	420
multiscale::verification::LogicPropertyDataReader	Class used to input logic properties	422
multiscale::verification::LogicPropertyGrammar	< Iterator > The grammar for parsing logic properties	427
multiscale::verification::LogicPropertyVisitor	Class used to evaluate logic properties	440
multiscale::analysis::MatFactory	Class for creating a cv::Mat object	457
multiscaletest::MinEnclosingTriangleFinder	Class for computing the minimum area enclosing triangle for a given polygon	463
multiscaletest::MinEnclosingTriangleFinderTest	Class for testing the minimum enclosing triangle algorithm	482
multiscale::verification::ModelChecker	Abstract class representing a generic model checker	489
multiscale::verification::ModelCheckerFactory	Interface for different model checker factories	501
multiscaletest::ModelCheckerTest	Class for testing model checkers	502
multiscale::verification::ModelCheckingException	Class for representing a model checking exception	508
multiscale::verification::ModelCheckingHelpRequestException	Class for representing a model checking help request exception	511
multiscale::verification::ModelCheckingManager	Class for managing the model checking processes	514
multiscale::verification::ModelCheckingOutputWriter	Class used to output the model checkers progress	529
multiscale::MultiplicationOperation	Functor representing a multiplication operation	547
multiscale::MultiscaleException	Parent exception class for the project	548
multiscaletest::MultiscaleTest	551

multiscale::verification::NextKLogicPropertyAttribute	553
Class for representing a "next K" logic property attribute	
multiscale::verification::NextLogicPropertyAttribute	554
Class for representing a "next" logic property attribute	
multiscale::verification::Nil	555
A class used to avoid run-time errors when defining a variant type	
multiscale::verification::NotConstraintAttribute	556
Class for representing a "not" constraint attribute	
multiscale::verification::NotLogicPropertyAttribute	557
Class for representing a "not" logic property attribute	
multiscale::NumberIterator	558
Abstract class representing a number iterator	
multiscale::Numeric	562
Class for processing numeric (shorts, ints, floats, doubles etc.) expressions	
multiscale::verification::NumericEvaluator	589
Class for evaluating numeric expressions	
multiscale::NumericException	592
Class for representing algorithm exceptions	
multiscale::verification::NumericMeasureAttribute	594
Class for representing a numeric measure attribute	
multiscale::verification::NumericMeasureCollectionAttribute	595
Class for representing a numeric measure collection attribute	
multiscale::verification::NumericMeasureCollectionEvaluator	596
Class used to evaluate numeric measure collections	
multiscale::verification::NumericMeasureCollectionVisitor	598
Class for evaluating numeric measure collections	
multiscale::NumericRangeManipulator	610
Operations for ranges of numeric values	
multiscale::verification::NumericSpatialMeasureAttribute	611
Class for representing a numeric spatial measure attribute	
multiscale::verification::NumericStateVariableAttribute	612
Class for representing a numeric state variable attribute	
multiscale::verification::NumericStateVariableGrammar< Iterator >	614
The grammar for parsing numeric state variable statements	
multiscale::verification::NumericStateVariableId	619
Class for representing the identity (name, type) of a numeric state variable	
multiscaletest::NumericStateVariableTraceTest	623
Class for testing evaluation of numeric state variable-only traces	
multiscale::verification::NumericStatisticalMeasureAttribute	626
Class for representing a numeric statistical measure attribute	
multiscale::verification::NumericVisitor	627
Class for evaluating numeric measures	
multiscale::OperatingSystem	635
Class for executing operating system related functions	
multiscale::verification::OrConstraintAttribute	639
Class for representing an "or" constraint attribute	
multiscale::verification::OrLogicPropertyAttribute	640
Class for representing an "or" logic property attribute	
multiscale::verification::Parser	641
Class used for parsing (P)BLSTL logical queries	
multiscale::verification::ParserGrammarExceptionHandler	645
Class for handling parser grammar exceptions	
multiscale::verification::ParserGrammarExtraInputException	648
Class for representing "extra input" exceptions in the parsing process	
multiscale::verification::ParserGrammarProbabilityException	650
Class for representing "probability" exceptions in the parsing process	
multiscale::verification::ParserGrammarUnexpectedTokenException	653
Class for representing "unexpected token" exceptions in the parsing process	

multiscale::verification::ParserGrammarUnparseableInputException	655
Class for representing "unparseable input" exceptions in the parsing process	655
multiscale::video::PolarCsvToInputFilesConverter	656
Csv file to input file converter considering polar coordinates	656
multiscale::video::PolarGnuplotScriptGenerator	668
Gnuplot script generator from the provided annular sectors	668
multiscale::verification::PrimaryConstraintAttribute	676
Class for representing a primary constraint attribute	676
multiscale::verification::PrimaryLogicPropertyAttribute	676
Class for representing a primary logic property attribute	676
multiscale::verification::PrimaryNumericMeasureAttribute	677
Class for representing a primary numeric measure attribute	677
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >	678
The grammar for parsing primary numeric measure statements	678
multiscale::verification::ProbabilisticBlackBoxModelChecker	686
Class used to run probabilistic black-box model checking tasks	686
multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory	690
Class for creating ProbabilisticBlackBoxModelChecker instances	690
multiscaletest::ProbabilisticBlackBoxModelCheckerTest	693
Class for testing the probabilistic black-box model checker	693
multiscale::verification::ProbabilisticLogicPropertyAttribute	696
Class for representing a probabilistic logic property attribute	696
multiscale::verification::ProbabilityErrorHandler	699
Structure for defining the error handler for invalid probability errors	699
multiscale::video::RectangularCsvToInputFilesConverter	701
Csv file to input file converter considering cartesian coordinates	701
multiscale::video::RectangularEntityCsvToInputFilesConverter	712
Csv entity file to input file converter considering cartesian coordinates	712
multiscale::video::RectangularGnuplotScriptGenerator	722
Gnuplot script generator from the provided concentrations considering a rectangular geometry	722
multiscale::analysis::RectangularMatFactory	727
Class for creating a cv::Mat object considering a rectangular grid	727
multiscale::verification::Region	733
Class for representing a region	733
multiscale::analysis::Region	735
Class for representing a region	735
multiscale::analysis::RegionDetector	744
Class for detecting regions of high intensity in grayscale images	744
multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >	766
Structure for specifying the type of the result	766
multiscale::verification::UnexpectedTokenErrorHandler::result< typename, typename, typename >	767
Structure for specifying the type of the result	767
multiscale::RGBColourGenerator	768
Generate a RGB colour	768
multiscale::RuntimeException	772
Class for representing runtime exceptions	772
multiscale::verification::SemanticType	775
Enumeration for defining a semantic type	775
multiscale::verification::SemanticTypeAttribute	777
Class for representing a semantic type attribute	777
multiscale::verification::SemanticTypeGrammar< Iterator >	778
The grammar for parsing semantic type statements	778
multiscale::verification::SemanticTypeStringGrammar< Iterator >	782
The grammar for parsing semantic type string statements	782
multiscale::analysis::Silhouette	786
Class for computing the "Silhouette" clustering index	786
multiscale::verification::SimilarityMeasureAttribute	791
Class for representing a similarity measure attribute	791

multiscale::verification::SimilarityMeasureTypeParser	Symbol table and parser for the similarity measure type	792
multiscale::verification::SimilarityTemporalNumericCollectionAttribute	Class for representing a similarity temporal numeric collection attribute	794
multiscale::analysis::SimulationClusterDetector	Class for detecting clusters in 2D images obtained from simulations	795
multiscaletest::SpatialEntitiesTraceTest	Class for testing evaluation of spatial entities-only traces	806
multiscale::verification::SpatialEntity	Class for representing a pseudo-3D spatial entity	809
multiscale::analysis::SpatialEntityPseudo3D	Class for representing a pseudo-3D (explicit 2D + implicit height) object	815
multiscale::verification::SpatialMeasureAttribute	Class for representing a spatial measure attribute	828
multiscale::verification::SpatialMeasureCollectionAttribute	Class used to represent a spatial measure collection attribute	829
multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >	The grammar for parsing spatial measure collection statements	831
multiscale::verification::SpatialMeasureEvaluator	Class for evaluating spatial measures	845
multiscale::verification::SpatialMeasureTypeParser	Symbol table and parser for the spatial measure type	846
multiscale::verification::SpatialNumericComparisonAttribute	Class for representing a spatial numeric comparison attribute	848
multiscale::verification::SpatialTemporalDataReader	Class for reading spatial temporal trace data from input files	849
multiscale::verification::SpatialTemporalException	Class for representing a spatial temporal exception	863
multiscale::verification::SpatialTemporalTrace	Class for representing a spatial temporal trace	866
multiscale::StandardNumberIterator	Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached	876
multiscale::verification::StateVariableAttribute	Class for representing a state variable attribute	880
multiscale::verification::StatisticalModelChecker	Class used to run statistical model checking tasks	882
multiscale::verification::StatisticalModelCheckerFactory	Class for creating StatisticalModelChecker instances	895
multiscaletest::StatisticalModelCheckerTest	Class for testing the statistical model checker	898
multiscale::StringManipulator	Class for manipulating std::strings	902
multiscale::verification::SubsetAttribute	Class for representing a subset attribute	908
multiscale::verification::SubsetOperationAttribute	Class for representing a subset operation attribute	909
multiscale::verification::SubsetOperationTypeParser	Symbol table and parser for the subset operation type	909
multiscale::verification::SubsetSpecificAttribute	Class for representing a subset specific attribute	911
multiscale::verification::SubsetSpecificTypeParser	Symbol table and parser for a specific subset type	912
multiscale::verification::SubsetSubsetOperationAttribute	Class for representing a subset subset operation attribute	913
multiscale::verification::SubsetVisitor	Class used to evaluate subsets	915

multiscale::SubtractionOperation	Functor representing a subtraction operation	922
multiscale::verification::TemporalDataReader	Class for reading (non-spatial) timeseries data from a .csv file	922
multiscale::verification::TemporalNumericCollectionAttribute	Class for representing a temporal numeric collection attribute	930
multiscale::verification::TemporalNumericCollectionGrammar< Iterator >	The grammar for parsing temporal numeric collection statements	930
multiscale::verification::TemporalNumericComparisonAttribute	Class for representing a temporal numeric comparison attribute	941
multiscale::verification::TemporalNumericMeasureAttribute	Class for representing a temporal numeric measure attribute	942
multiscale::verification::TemporalNumericMeasureCollectionAttribute	Class for representing temporal numeric measure collection attributes	943
multiscale::verification::TemporalNumericMeasureGrammar< Iterator >	The grammar for parsing temporal numeric measure statements	944
multiscale::verification::TemporalNumericVisitor	Class for evaluating temporal numeric measures	953
multiscale::TestException	Class for representing testing exceptions	962
multiscale::verification::TimePoint	Class for representing a timepoint	965
multiscale::verification::TimePointEvaluator	Class used to evaluate timepoints	976
multiscale::verification::TimeseriesComponentAttribute	Class for representing a timeseries component attribute	979
multiscale::verification::TimeseriesComponentEvaluator	Class for evaluating timeseries components	980
multiscale::verification::TimeseriesComponentVisitor	Class for evaluating timeseries components	982
multiscale::verification::TimeseriesMeasureAttribute	Class for representing a timeseries measure attribute	986
multiscale::verification::TimeseriesMeasureTypeParser	Symbol table and parser for the timeseries measure type	987
multiscale::verification::TimeseriesTimeseriesComponentAttribute	Class for representing a timeseries timeseries component attribute	989
multiscaletest::TraceEvaluationTest	Class for testing evaluation of traces	990
multiscale::verification::TypeSemanticsTable	Class for defining a type semantics table	998
multiscale::verification::UnaryNumericFilterAttribute	Class for representing a unary numeric filter attribute	1008
multiscale::verification::UnaryNumericMeasureAttribute	Class for representing a unary numeric measure attribute	1009
multiscale::verification::UnaryNumericMeasureGrammar< Iterator >	The grammar for parsing unary numeric measure statements	1010
multiscale::verification::UnaryNumericMeasureTypeParser	Symbol table and parser for the unary numeric measure type	1014
multiscale::verification::UnaryNumericNumericAttribute	Class for representing a unary numeric numeric measure attribute	1016
multiscale::verification::UnaryNumericTemporalAttribute	Class for representing a unary numeric temporal measure attribute	1017
multiscale::verification::UnarySpatialConstraintAttribute	Class for representing a "unary" spatial constraint attribute	1019
multiscale::verification::UnaryStatisticalMeasureAttribute	Class for representing a unary statistical measure attribute	1020
multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >	The grammar for parsing unary statistical measure statements	1021

multiscale::verification::UnaryStatisticalMeasureTypeParser	Symbol table and parser for the unary statistical measure type	1025
multiscale::verification::UnaryStatisticalNumericAttribute	Class for representing a unary statistical numeric attribute	1028
multiscale::verification::UnaryStatisticalSpatialAttribute	Class for representing a unary statistical spatial attribute	1029
multiscale::verification::UnaryTypeConstraintAttribute	Class for representing a "unary" type constraint attribute	1030
multiscale::UnexpectedBehaviourException	Class for representing unexpected behaviour exceptions	1032
multiscale::verification::UnexpectedTokenErrorHandler	Structure for defining the error handler for unexpected token errors	1035
multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >	1037
multiscale::UnimplementedMethodException	Class for representing unimplemented method exceptions	1039
multiscale::verification::UntilLogicPropertyAttribute	Class for representing an "until" logic property attribute	1042
multiscale::UserDefinedTypeName< T >	Class for representing a user defined type name	1043
multiscale::XmlValidator::XmlValidationErrorHandler	Class used for handling errors during the xml file validation process	1044
multiscale::XmlValidator	Class used to validate xml files	1049

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/ Multiscale.hpp	1057
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/ MultiscaleTest.hpp	1058
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/ UserDefinedTypeName.hpp .	1058
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ AlgorithmException.hpp	1060
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ ExceptionHandler.hpp	1060
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ FileOpenException.hpp	1061
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ IndexOutOfBoundsException.hpp	1062
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ InvalidInputException.hpp	1063
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ IOException.hpp	1064
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ MultiscaleException.hpp	1065
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ NumericException.hpp	1068
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ RuntimeException.hpp	1068
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ TestException.hpp	1069
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ UnexpectedBehaviourException.hpp	1070
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ UnimplementedMethodException.hpp	1071
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ CircularityMeasure.hpp	1072
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Cluster.hpp	1073
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ ClusterDetector.hpp	1076
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ DataPoint.hpp	1077
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ DSCAN.hpp	1078
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Detector.hpp	1079
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Entity.hpp	1079
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ MatFactory.hpp	1082
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Region.hpp	1083

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/RegionDetector.hpp	1085
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Shape2D.hpp	1086
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Silhouette.hpp	1087
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp	1087
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/SpatialEntityPseudo3DType.hpp	1088
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp	1075
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/CircularMatFactory.hpp	1080
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/RectangularMatFactory.hpp	1081
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/CircularMatFactorySample.cpp	1088
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/LexicographicIteratorSample.cpp	1089
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/RectangularMatFactorySample.cpp	1090
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularDetectRegions.cpp	1091
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularityMeasure.cpp	1095
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Cluster.cpp	1095
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/ClusterDetector.cpp	1096
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/DBSCAN.cpp	1097
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Detector.cpp	1097
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Entity.cpp	1098
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/MatFactory.cpp	1100
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.cpp	1100
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Region.cpp	1104
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/RegionDetector.cpp	1104
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Silhouette.cpp	1105
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SimulationClusters.cpp	1105
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SpatialEntityPseudo3D.cpp	1108
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/SimulationClusterDetector.cpp	1096
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/CircularMatFactory.cpp	1098
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/RectangularMatFactory.cpp	1099
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/DBSCANTest.cpp	1109
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/ConsolePrinter.hpp	1111
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Filesystem.hpp	1112
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Geometry2D.hpp	1113
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/MinEnclosingTriangleFinder.hpp	1115
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumberIterator.hpp	1116
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Numeric.hpp	1116
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumericRangeManipulator.hpp	1117

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/OperatingSystem. .hpp	1118
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/RGBColourGenerator. .hpp	1118
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/StringManipulator. .hpp	1122
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/XmlValidator.hpp	1123
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/LexicographicNumberIterator. .hpp	1113
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/NumberIteratorType. .hpp	1114
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/StandardNumberIterator. .hpp	1114
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/BetaDistribution. .hpp	1119
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/BinomialDistribution. .hpp	1120
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/Distribution. .hpp	1121
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/ConsolePrinterSample.cpp	1124
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/ExecuteProgramSample.cpp	1125
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/LineCircleIntersectionSample. .cpp	1126
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/MinEnclosingTriangleFinderSample. .cpp	1127
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/RGBColourGeneratorSample. .cpp	1131
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/XmlValidatorSample.cpp	1132
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/ConsolePrinter.cpp	1133
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/Filesystem.cpp	1133
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/Geometry2D.cpp	1134
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/MinEnclosingTriangleFinder.cpp	1136
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/NumberIterator.cpp	1137
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/Numeric.cpp	1137
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/OperatingSystem.cpp	1138
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/RGBColourGenerator.cpp	1138
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/StringManipulator.cpp	1141
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/XmlValidator.cpp	1141
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/LexicographicNumberIterator.cpp	1135
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/StandardNumberIterator.cpp	1135
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/BetaDistribution.cpp	1139
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/BinomialDistribution.cpp	1139
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/Distribution.cpp	1140
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/Geometry2DTest.cpp	1142
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/MinEnclosingTriangleFinderTest.cpp	1143
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/NumericTest.cpp	1144
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/StatisticsTest.cpp	1148
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndConstraintAttribute.hpp	1149
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndLogicPropertyAttribute.hpp	1150
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericFilterAttribute.hpp	1151
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp	1152
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericNumericAttribute.hpp	1154

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ TemporalNumericMeasureCollectionAttribute.hpp	1214
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ TimeseriesComponentAttribute.hpp	1214
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ TimeseriesMeasureAttribute.hpp	1215
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ TimeseriesTimeseriesComponentAttribute.hpp	1216
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryNumericFilterAttribute.hpp	1217
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryNumericMeasureAttribute.hpp	1218
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryNumericNumericAttribute.hpp	1220
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryNumericTemporalAttribute.hpp	1220
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnarySpatialConstraintAttribute.hpp	1221
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryStatisticalMeasureAttribute.hpp	1222
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryStatisticalNumericAttribute.hpp	1224
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryStatisticalSpatialAttribute.hpp	1224
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UnaryTypeConstraintAttribute.hpp	1225
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ UntilLogicPropertyAttribute.hpp	1226
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ApproximateBayesianModelChecker.hpp	1227
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ApproximateBayesianModelCheckerFactory.hpp	1228
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ApproximateProbabilisticModelChecker.hpp	1229
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ApproximateProbabilisticModelCheckerFactory.hpp	1229
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ BayesianModelChecker.hpp	1230
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ BayesianModelCheckerFactory.hpp	1230
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ModelChecker.hpp	1231
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ModelCheckerFactory.hpp	1232
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ModelCheckingManager.hpp	1232
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ModelCheckingOutputWriter.hpp	1233
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ProbabilisticBlackBoxModelChecker.hpp	1234
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ ProbabilisticBlackBoxModelCheckerFactory.hpp	1234
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ StatisticalModelChecker.hpp	1235
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ StatisticalModelCheckerFactory.hpp	1236
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/data/ LogicPropertyDataReader.hpp	1236

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/ SpatialTemporalDataReader.hpp	1237
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/ TemporalDataReader.hpp	1238
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ModelCheckingException.hpp	1239
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ModelCheckingHelpRequestException.hpp	1240
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ParserGrammarExceptionHandler.hpp	1241
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ParserGrammarExtraInputException.hpp	1242
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ParserGrammarProbabilityException.hpp	1243
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ParserGrammarUnexpectedTokenException.hpp	1244
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ ParserGrammarUnparseableInputException.hpp	1245
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ SpatialTemporalException.hpp	1246
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/ CommandLineModelChecking.hpp	1247
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/ ProbabilityErrorHandler.hpp	1248
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/ UnexpectedTokenErrorHandler.hpp	1249
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ AbstractSyntaxTree.hpp	1250
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ Cluster.hpp	1074
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ NumericStateVariableId.hpp	1250
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ Region.hpp	1084
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ SemanticType.hpp	1251
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ SpatialEntity.hpp	1252
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ SpatialTemporalTrace.hpp	1253
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ TimePoint.hpp	1254
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/ TypeSemanticsTable.hpp	1255
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryNumericMeasureGrammar.hpp	1256
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryNumericMeasureGrammarDefinition.hpp	1257
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryStatisticalMeasureGrammar.hpp	1258
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryStatisticalMeasureGrammarDefinition.hpp	1258
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryStatisticalQuantileMeasureGrammar.hpp	1259
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ BinaryStatisticalQuantileMeasureGrammarDefinition.hpp	1260
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ ChangeMeasureGrammar.hpp	1260

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ ChangeMeasureGrammarDefinition.hpp	1261
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ ComparatorGrammar.hpp	1262
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ ComparatorGrammarDefinition.hpp	1262
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ LogicPropertyGrammar.hpp	1263
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ LogicPropertyGrammarDefinition.hpp	1264
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ NumericStateVariableGrammar.hpp	1265
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ NumericStateVariableGrammarDefinition.hpp	1265
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ Parser.hpp	1266
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ PrimaryNumericMeasureGrammar.hpp	1267
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ PrimaryNumericMeasureGrammarDefinition.hpp	1268
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SemanticTypeGrammar.hpp	1268
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SemanticTypeGrammarDefinition.hpp	1269
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SemanticTypeStringGrammar.hpp	1270
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SemanticTypeStringGrammarDefinition.hpp	1270
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SpatialMeasureCollectionGrammar.hpp	1271
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SpatialMeasureCollectionGrammarDefinition.hpp	1272
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SymbolTables.hpp	1272
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ SymbolTablesAutoGenerated.hpp	1274
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ TemporalNumericCollectionGrammar.hpp	1275
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ TemporalNumericCollectionGrammarDefinition.hpp	1276
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ TemporalNumericMeasureGrammar.hpp	1276
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ TemporalNumericMeasureGrammarDefinition.hpp	1277
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ UnaryNumericMeasureGrammar.hpp	1278
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ UnaryNumericMeasureGrammarDefinition.hpp	1279
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ UnaryStatisticalMeasureGrammar.hpp	1279
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ UnaryStatisticalMeasureGrammarDefinition.hpp	1280
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ ChangeMeasureEvaluator.hpp	1281
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ ComparatorEvaluator.hpp	1281
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ ConstraintVisitor.hpp	1282

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ FilterNumericVisitor.hpp	1283
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ LogicPropertyVisitor.hpp	1284
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ NumericEvaluator.hpp	1285
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ NumericMeasureCollectionEvaluator.hpp	1286
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ NumericMeasureCollectionVisitor.hpp	1287
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ NumericVisitor.hpp	1288
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ SpatialMeasureEvaluator.hpp	1289
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ SubsetVisitor.hpp	1290
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ TemporalNumericVisitor.hpp	1291
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ TimePointEvaluator.hpp	1292
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ TimeseriesComponentEvaluator.hpp	1293
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ TimeseriesComponentVisitor.hpp	1294
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ LogicPropertyDataReaderSample.cpp	1295
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ ParserEvaluationSample.cpp	1296
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ ParserSample.cpp	1297
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ PatternAnalysisSample.cpp	1298
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ SpatialTemporalDataReaderSample.cpp	1298
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ TemporalDataReaderSample.cpp	1300
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ TypeSemanticsTableDataReaderSample.cpp	1302
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/ Mule.cpp	1332
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ BinaryNumericMeasureAttribute.cpp	1303
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ BinaryStatisticalMeasureAttribute.cpp	1304
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ BinaryStatisticalQuantileMeasureAttribute.cpp	1304
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ ChangeMeasureAttribute.cpp	1305
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ ComparatorAttribute.cpp	1306
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ HeterogeneousTimeseriesComponentAttribute.cpp	1307
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ HomogeneousTimeseriesComponentAttribute.cpp	1308
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ HomogeneousTimeseriesMeasureAttribute.cpp	1309
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ ProbabilisticLogicPropertyAttribute.cpp	1310

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SimilarityMeasureAttribute.cpp	1310
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SpatialMeasureAttribute.cpp	1311
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SpatialMeasureAttributeAutoGenerated.cpp	1312
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SubsetOperationAttribute.cpp	1313
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SubsetSpecificAttribute.cpp	1314
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SubsetSpecificAttributeAutoGenerated.cpp	1315
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ TimeseriesMeasureAttribute.cpp	1316
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ UnaryNumericMeasureAttribute.cpp	1317
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ UnaryStatisticalMeasureAttribute.cpp	1318
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateBayesianModelChecker.cpp	1319
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateBayesianModelCheckerFactory.cpp	1320
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateProbabilisticModelChecker.cpp	1320
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateProbabilisticModelCheckerFactory.cpp	1320
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ BayesianModelChecker.cpp	1321
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ BayesianModelCheckerFactory.cpp	1321
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelChecker.cpp	1322
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelCheckingManager.cpp	1322
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelCheckingOutputWriter.cpp	1322
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ProbabilisticBlackBoxModelChecker.cpp	1323
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ProbabilisticBlackBoxModelCheckerFactory.cpp	1323
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelChecker.cpp	1324
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelCheckerFactory.cpp	1324
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ LogicPropertyDataReader.cpp	1325
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReader.cpp	1325
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReaderAutoGenerated.cpp	1326
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ TemporalDataReader.cpp	1327
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ ParserGrammarExceptionHandler.cpp	1327
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/ CommandLineModelChecking.cpp	1328
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ AbstractSyntaxTree.cpp	1329

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ NumericStateVariableId.cpp	1329
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SemanticType.cpp	1329
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialEntity.cpp	1330
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialTemporalTrace.cpp	1331
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ TimePoint.cpp	1331
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ TypeSemanticsTable.cpp	1332
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryNumericMeasureGrammar.cpp	1333
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryStatisticalMeasureGrammar.cpp	1333
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ BinaryStatisticalQuantileMeasureGrammar.cpp	1334
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ ChangeMeasureGrammar.cpp	1334
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ ComparatorGrammar.cpp	1335
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ LogicPropertyGrammar.cpp	1335
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ NumericStateVariableGrammar.cpp	1336
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ Parser.cpp	1336
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ PrimaryNumericMeasureGrammar.cpp	1337
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ SemanticTypeGrammar.cpp	1337
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ SemanticTypeStringGrammar.cpp	1338
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ SpatialMeasureCollectionGrammar.cpp	1339
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ TemporalNumericCollectionGrammar.cpp	1339
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ TemporalNumericMeasureGrammar.cpp	1340
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ UnaryNumericMeasureGrammar.cpp	1340
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ UnaryStatisticalMeasureGrammar.cpp	1341
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ApproximateBayesianModelCheckerTest.hpp	1341
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ApproximateProbabilisticModelCheckerTest.hpp	1342
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ BayesianModelCheckerTest.hpp	1343
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ModelCheckerTest.hpp	1343
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ModelCheckingTest.cpp	1344
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ProbabilisticBlackBoxModelCheckerTest.hpp	1345
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ StatisticalModelCheckerTest.hpp	1345

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp	1346
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp	1366
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp	1385
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.cpp	1405
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.hpp	1405
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp	1405
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.cpp	1425
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.hpp	1425
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/InputStringParser.hpp	1426
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.cpp	1427
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.hpp	1427
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/AnnularSector.hpp	1501
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/CartesianToPolarConverter.hpp	1502
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarCsvToInputFilesConverter.hpp	1503
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarGnuplotScriptGenerator.hpp	1503
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/AnnularSector.cpp	1504
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/CartesianToPolarConverter.cpp	1505
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/MapCartesianToPolarScript.cpp	1506
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarCsvToInputFilesConverter.cpp	1507
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarGnuplotScriptGenerator.cpp	1508
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarMapCsvToInputFiles.cpp	1509
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/CartesianToConcentrationsConverter.hpp	1511
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularCsvToInputFilesConverter.hpp	1511
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularEntityCsvToInputFilesConverter.hpp	1512
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularGnuplotScriptGenerator.hpp	1513
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/CartesianToConcentrationsConverter.cpp	1513
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/MapCartesianToScript.cpp	1514
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularCsvToInputFilesConverter.cpp	1515
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularEntityCsvToInputFilesConverter.cpp	1516

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/	RectangularGnuplot.h	1517
ScriptGenerator.cpp		
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/	RectangularMapCsv.h	1517
ToInputFiles.cpp		
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/	RectangularMap.h	1519
EntityCsvToInputFiles.cpp		

Chapter 6

Namespace Documentation

6.1 multiscale Namespace Reference

Namespaces

- [analysis](#)
- [verification](#)
- [video](#)

Classes

- class [AdditionOperation](#)
Functor representing an addition operation.
- class [AlgorithmException](#)
Class for representing algorithm exceptions.
- class [BetaDistribution](#)
Class for analysing Beta distributed data.
- class [BinomialDistribution](#)
Class for analysing Binomial distributed data.
- class [ConsolePrinter](#)
Class used to print (coloured) messages to the console.
- class [Distribution](#)
- class [DivisionOperation](#)
Functor representing a division operation.
- class [ExceptionHandler](#)
Exception handler class.
- class [FileOpenException](#)
Class for representing exceptions when opening a file.
- class [Filesystem](#)
Class containing methods for interacting with the filesystem.
- class [Geometry2D](#)
Two-dimensional geometric operations.
- class [IndexOutOfBoundsException](#)
Class for representing an index out of bounds exception.
- class [InvalidInputException](#)
Class for representing invalid input exceptions.
- class [IOException](#)

- class [LexicographicNumberIterator](#)

Class for representing input and output exceptions.
- class [MinEnclosingTriangleFinder](#)

Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "—".
- class [MultiplicationOperation](#)

Functor representing a multiplication operation.
- class [MultiscaleException](#)

Parent exception class for the project.
- class [NumberIterator](#)

Abstract class representing a number iterator.
- class [Numeric](#)

Class for processing numeric (shorts, ints, floats, doubles etc.) expressions.
- class [NumericException](#)

Class for representing algorithm exceptions.
- class [NumericRangeManipulator](#)

Operations for ranges of numeric values.
- class [OperatingSystem](#)

Class for executing operating system related functions.
- class [RGBColourGenerator](#)

Generate a RGB colour.
- class [RuntimeException](#)

Class for representing runtime exceptions.
- class [StandardNumberIterator](#)

Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.
- class [StringManipulator](#)

Class for manipulating std::strings.
- class [SubtractionOperation](#)

Functor representing a subtraction operation.
- class [TestException](#)

Class for representing testing exceptions.
- class [UnexpectedBehaviourException](#)

Class for representing unexpected behaviour exceptions.
- class [UnimplementedMethodException](#)

Class for representing unimplemented method exceptions.
- class [UserDefinedTypeName](#)

Class for representing a user defined type name.
- class [XmlValidator](#)

Class used to validate xml files.

Enumerations

- enum [UnixColourCode](#) : unsigned int {

 UnixColourCode::BLACK = 0, UnixColourCode::RED = 1, UnixColourCode::GREEN = 2, UnixColourCode::YELLOW = 3,

 UnixColourCode::BLUE = 4, UnixColourCode::MAGENTA = 5, UnixColourCode::CYAN = 6, UnixColourCode::WHITE = 7 }

- enum `WindowsColourCode` : unsigned int {
 `WindowsColourCode::BLACK` = 0, `WindowsColourCode::DARK_BLUE` = 1, `WindowsColourCode::DARK_GREEN` = 2, `WindowsColourCode::DARK_CYAN` = 3, `WindowsColourCode::DARK_RED` = 4, `WindowsColourCode::DARK_MAGENTA` = 5, `WindowsColourCode::DARK_YELLOW` = 6, `WindowsColourCode::DARK_WHITE` = 7, `WindowsColourCode::GRAY` = 8, `WindowsColourCode::BLUE` = 9, `WindowsColourCode::GREEN` = 10, `WindowsColourCode::CYAN` = 11, `WindowsColourCode::RED` = 12, `WindowsColourCode::MAGENTA` = 13, `WindowsColourCode::YELLOW` = 14, `WindowsColourCode::WHITE` = 15 }
- enum `ColourCode` : unsigned int {
 `ColourCode::BLACK` = 0, `ColourCode::RED` = 1, `ColourCode::GREEN` = 2, `ColourCode::YELLOW` = 3, `ColourCode::BLUE` = 4, `ColourCode::MAGENTA` = 5, `ColourCode::CYAN` = 6, `ColourCode::WHITE` = 7 }
- enum `NumberIteratorType` { `STANDARD` = 1, `LEXICOGRAPHIC` = 2 }

The type of the number iterator.

Variables

- const int `EXEC_SUCCESS_CODE` = 0
- const int `EXEC_ERR_CODE` = 1
- const std::string `ERR_MSG` = "An error occurred: "
- const std::string `ERR_UNDEFINED_ENUM_VALUE` = "The provided enumeration value is invalid. Please use one of the available enumeration values instead."
- const std::string `ERR_INDEX_OUT_OF_BOUNDS_BEGIN` = "The provided index value ("
- const std::string `ERR_INDEX_OUT_OF_BOUNDS_END` = ") is invalid. Please change."
- const std::string `ERR_UNIMPLEMENTED_METHOD` = "The method you tried to call is not implemented. Please change."

6.1.1 Enumeration Type Documentation

6.1.1.1 enum multiscale::ColourCode : unsigned int [strong]

Enumerator

- BLACK** Black non-colour
RED Red colour
GREEN Green colour
YELLOW Yellow colour
BLUE Blue colour
MAGENTA Magenta colour
CYAN Cyan colour
WHITE White non-colour

Definition at line 44 of file ConsolePrinter.hpp.

6.1.1.2 enum multiscale::NumberIteratorType

The type of the number iterator.

Enumerator

- STANDARD** Standard number iterator
LEXICOGRAPHIC Lexicographic number iterator

Definition at line 7 of file NumberIteratorType.hpp.

6.1.1.3 enum multiscale::UnixColourCode : unsigned int [strong]

Enumerator

BLACK Black non-colour

RED Red colour

GREEN Green colour

YELLOW Yellow colour

BLUE Blue colour

MAGENTA Magenta colour

CYAN Cyan colour

WHITE White non-colour

Definition at line 12 of file ConsolePrinter.hpp.

6.1.1.4 enum multiscale::WindowsColourCode : unsigned int [strong]

Enumerator

BLACK Black non-colour

DARK_BLUE Dark blue colour

DARK_GREEN Dark green colour

DARK_CYAN Dark cyan colour

DARK_RED Dark red colour

DARK_MAGENTA Dark magenta colour

DARK_YELLOW Dark yellow colour

DARK_WHITE White non-colour

GRAY Gray non-colour

BLUE Blue colour

GREEN Green colour

CYAN Cyan colour

RED Red colour

MAGENTA Magenta colour

YELLOW Yellow colour

WHITE Faint white non-colour

Definition at line 24 of file ConsolePrinter.hpp.

6.1.2 Variable Documentation

6.1.2.1 const std::string multiscale::ERR_INDEX_OUT_OF_BOUNDS_BEGIN = "The provided index value ("

Definition at line 30 of file Multiscale.hpp.

Referenced by multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException().

6.1.2.2 const std::string multiscale::ERR_INDEX_OUT_OF_BOUNDS_END = ") is invalid. Please change."

Definition at line 31 of file Multiscale.hpp.

Referenced by multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException().

6.1.2.3 const std::string multiscale::ERR_MSG = "An error occurred: "

Definition at line 25 of file Multiscale.hpp.

Referenced by isValidNrOfConcentrationsForPosition(), isValidOutputType(), multiscale::ExceptionHandler::printDetailedErrorMessage(), multiscale::ExceptionHandler::printRawErrorMessage(), and printWrongParameters().

6.1.2.4 const std::string multiscale::ERR_UNDEFINED_ENUM_VALUE = "The provided enumeration value is invalid. Please use one of the available enumeration values instead."

Definition at line 28 of file Multiscale.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::areSimilarValues(), multiscale::verification::ChangeMeasureEvaluator::computeNumericMeasureValueChange(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::TimeseriesComponentEvaluator::evaluate(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateHomogeneousHomogeneousTimeseries(), multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTimeseriesTimeseriesComponent(), multiscale::verification::spatialmeasure::getMaxValidSpatialMeasureValue(), multiscale::verification::spatialmeasure::getMinValidSpatialMeasureValue(), multiscale::verification::TimeseriesComponentVisitor::operator()(), multiscale::verification::spatialmeasure::validateSpatialMeasureType(), multiscale::verification::spatialmeasure::validateSpatialMeasureTypeIndex(), multiscale::verification::subsetsspecific::validateSubsetSpecificType(), and multiscale::verification::subsetsspecific::validateSubsetSpecificTypeIndex().

6.1.2.5 const std::string multiscale::ERR_UNIMPLEMENTED_METHOD = "The method you tried to call is not implemented. Please change."

Definition at line 31 of file UnimplementedMethodException.hpp.

6.1.2.6 const int multiscale::EXEC_ERR_CODE = 1

Definition at line 22 of file Multiscale.hpp.

Referenced by analysePatterns(), main(), and readAndPrintInputFile().

6.1.2.7 const int multiscale::EXEC_SUCCESS_CODE = 0

Definition at line 21 of file Multiscale.hpp.

Referenced by main(), and readAndPrintInputFile().

6.2 multiscale::analysis Namespace Reference

Classes

- class [CircularityMeasure](#)

Class for computing the circularity measure for the given collection of points.
- class [CircularMatFactory](#)

Class for creating a Mat object considering a circular grid.
- class [Cluster](#)

Class for representing a cluster of entities in an image.
- class [ClusterDetector](#)

Class for detecting clusters in 2D images.
- class [DataPoint](#)

- class [DBSCAN](#)

Class for representing a data point.
- class [Detector](#)

Class which implements an improved version of the [DBSCAN](#) algorithm.
- class [Entity](#)

Abstract class for detecting entities of interest in images.
- class [MatFactory](#)

Class for creating a cv::Mat object.
- class [RectangularMatFactory](#)

Class for creating a cv::Mat object considering a rectangular grid.
- class [Region](#)

Class for representing a region.
- class [RegionDetector](#)

Class for detecting regions of high intensity in grayscale images.
- class [Silhouette](#)

Class for computing the "Silhouette" clustering index.
- class [SimulationClusterDetector](#)

Class for detecting clusters in 2D images obtained from simulations.
- class [SpatialEntityPseudo3D](#)

Class for representing a pseudo-3D (explicit 2D + implicit height) object.

Typedefs

- `typedef std::pair< std::vector< cv::Point >, std::vector< std::vector< cv::Point > > > Polygon`

Enumerations

- enum [Shape2D](#) : unsigned int { [Shape2D::Triangle](#) = 0, [Shape2D::Rectangle](#), [Shape2D::Circle](#), [Shape2D::Undefined](#) }

Enumeration for determining the type of a 2D shape.

- enum [SpatialEntityPseudo3DType](#) : unsigned int { [SpatialEntityPseudo3DType::Cluster](#) = 0, [SpatialEntityPseudo3DType::Region](#) }

Enumeration for determining the type of a pseudo 3D entity.

6.2.1 Typedef Documentation

6.2.1.1 `typedef std::pair<std::vector<cv::Point>, std::vector<std::vector<cv::Point> > >` `multiscale::analysis::Polygon`

Define a wrapper for polygons i.e. pairs (o, i) where o = outer contour and i = collection of inner contours/holes
Definition at line 20 of file RegionDetector.hpp.

6.2.2 Enumeration Type Documentation

6.2.2.1 enum multiscale::analysis::Shape2D : unsigned int [strong]

Enumeration for determining the type of a 2D shape.

Enumerator

Triangle Triangular 2D shape

Rectangle Rectangular 2D shape

Circle Circular 2D shape

Undefined Undefined 2D shape

Definition at line 10 of file Shape2D.hpp.

6.2.2.2 enum multiscale::analysis::SpatialEntityPseudo3DType : unsigned int [strong]

Enumeration for determining the type of a pseudo 3D entity.

Enumerator

Cluster Cluster

Region Region

Definition at line 10 of file SpatialEntityPseudo3DType.hpp.

6.3 multiscale::verification Namespace Reference

Namespaces

- [spatialmeasure](#)
- [subsetsspecific](#)

Classes

- class [AbstractSyntaxTree](#)
Class used for representing an abstract syntax tree.
- class [AndConstraintAttribute](#)
Class for representing an "and" constraint attribute.
- class [AndLogicPropertyAttribute](#)
Class for representing an "and" logic property attribute.
- class [ApproximateBayesianModelChecker](#)
Class used to run approximate Bayesian model checking tasks.
- class [ApproximateBayesianModelCheckerFactory](#)
Class for creating [ApproximateBayesianModelChecker](#) instances.
- class [ApproximateProbabilisticModelChecker](#)
Class used to run approximate probabilistic model checking tasks.
- class [ApproximateProbabilisticModelCheckerFactory](#)
Class for creating [ApproximateProbabilisticModelChecker](#) instances.
- class [BayesianModelChecker](#)
Class used to run Bayesian model checking tasks.

- class [BayesianModelCheckerFactory](#)
Class for creating `BayesianModelChecker` instances.
- class [BinaryNumericFilterAttribute](#)
Class for representing a binary numeric filter attribute.
- class [BinaryNumericMeasureAttribute](#)
Class for representing a binary numeric measure attribute.
- class [BinaryNumericMeasureGrammar](#)
The grammar for parsing binary numeric measure statements.
- struct [BinaryNumericMeasureTypeParser](#)
Symbol table and parser for the binary numeric measure type.
- class [BinaryNumericNumericAttribute](#)
Class for representing a binary numeric numeric measure attribute.
- class [BinaryNumericTemporalAttribute](#)
Class for representing a binary numeric temporal measure attribute.
- class [BinaryStatisticalMeasureAttribute](#)
Class for representing a binary statistical measure attribute.
- class [BinaryStatisticalMeasureGrammar](#)
The grammar for parsing binary statistical measure statements.
- struct [BinaryStatisticalMeasureTypeParser](#)
Symbol table and parser for the binary statistical measure type.
- class [BinaryStatisticalNumericAttribute](#)
Class for representing a binary statistical numeric attribute.
- class [BinaryStatisticalQuantileMeasureAttribute](#)
Class for representing a binary statistical quantile measure attribute.
- class [BinaryStatisticalQuantileMeasureGrammar](#)
The grammar for parsing binary statistical quantile measure statements.
- struct [BinaryStatisticalQuantileMeasureTypeParser](#)
Symbol table and parser for the binary statistical quantile measure type.
- class [BinaryStatisticalQuantileNumericAttribute](#)
Class for representing a binary statistical quantile numeric attribute.
- class [BinaryStatisticalQuantileSpatialAttribute](#)
Class for representing a binary statistical quantile spatial attribute.
- class [BinaryStatisticalSpatialAttribute](#)
Class for representing a binary statistical spatial attribute.
- class [ChangeMeasureAttribute](#)
Class for representing a change measure attribute.
- class [ChangeMeasureEvaluator](#)
Class for evaluating change measure expressions.
- class [ChangeMeasureGrammar](#)
The grammar for parsing change measure statements.
- struct [ChangeMeasureTypeParser](#)
Symbol table and parser for the change measure type.
- class [ChangeTemporalNumericCollectionAttribute](#)
Class for representing a change temporal numeric collection attribute.
- class [ChangeTemporalNumericMeasureAttribute](#)
Class for representing a change temporal numeric measure attribute.
- class [Cluster](#)
Class for representing a cluster.
- class [CommandLineModelChecking](#)
Class for running model checkers from the command line.
- class [ComparatorAttribute](#)

- class [ComparatorEvaluator](#)
Class for representing a comparator attribute.
- class [ComparatorGrammar](#)
The grammar for parsing comparator statements.
- struct [ComparatorNonEqualTypeParser](#)
Symbol table and parser for the comparator type which does not accept the "=" symbol.
- struct [ComparatorTypeParser](#)
Symbol table and parser for the comparator type.
- class [ConstraintAttribute](#)
Class for representing a constraint attribute.
- class [ConstraintVisitor](#)
Class used to evaluate constraints.
- class [EquivalenceConstraintAttribute](#)
Class for representing an "equivalence" constraint attribute.
- class [EquivalenceLogicPropertyAttribute](#)
Class for representing an "equivalence" logic property attribute.
- class [FilterNumericMeasureAttribute](#)
Class for representing a filter numeric measure.
- class [FilterNumericVisitor](#)
Class for evaluating filter numeric measures.
- class [FilterSubsetAttribute](#)
Class for representing a filter subset attribute.
- class [FutureLogicPropertyAttribute](#)
Class for representing a "future" logic property attribute.
- class [GlobalLogicPropertyAttribute](#)
Class for representing a "globally" logic property attribute.
- class [HeterogeneousTimeseriesComponentAttribute](#)
Class for representing a heterogeneous timeseries component attribute.
- struct [HeterogeneousTimeseriesComponentTypeParser](#)
Symbol table and parser for the heterogeneous timeseries component type.
- class [HomogeneousHomogeneousTimeseriesAttribute](#)
Class for representing a homogeneous homogeneous timeseries attribute.
- class [HomogeneousTimeseriesComponentAttribute](#)
Class for representing a homogeneous timeseries component attribute.
- struct [HomogeneousTimeseriesComponentTypeParser](#)
Symbol table and parser for the homogeneous timeseries component type.
- class [HomogeneousTimeseriesMeasureAttribute](#)
Class for representing a homogeneous timeseries measure attribute.
- struct [HomogeneousTimeseriesMeasureTypeParser](#)
Symbol table and parser for the homogeneous timeseries measure type.
- class [ImplicationConstraintAttribute](#)
Class for representing an "implication" constraint attribute.
- class [ImplicationLogicPropertyAttribute](#)
Class for representing an "implication" logic property attribute.
- class [LogicPropertyAttribute](#)
Class for representing a logic property attribute.
- class [LogicPropertyDataReader](#)
Class used to input logic properties.
- class [LogicPropertyGrammar](#)
The grammar for parsing logic properties.

- class [LogicPropertyVisitor](#)
Class used to evaluate logic properties.
- class [ModelChecker](#)
Abstract class representing a generic model checker.
- class [ModelCheckerFactory](#)
Interface for different model checker factories.
- class [ModelCheckingException](#)
Class for representing a model checking exception.
- class [ModelCheckingHelpRequestException](#)
Class for representing a model checking help request exception.
- class [ModelCheckingManager](#)
Class for managing the model checking processes.
- class [ModelCheckingOutputWriter](#)
Class used to output the model checkers progress.
- class [NextKLogicPropertyAttribute](#)
Class for representing a "next K" logic property attribute.
- class [NextLogicPropertyAttribute](#)
Class for representing a "next" logic property attribute.
- class [Nil](#)
A class used to avoid run-time errors when defining a variant type.
- class [NotConstraintAttribute](#)
Class for representing a "not" constraint attribute.
- class [NotLogicPropertyAttribute](#)
Class for representing a "not" logic property attribute.
- class [NumericEvaluator](#)
Class for evaluating numeric expressions.
- class [NumericMeasureAttribute](#)
Class for representing a numeric measure attribute.
- class [NumericMeasureCollectionAttribute](#)
Class for representing a numeric measure collection attribute.
- class [NumericMeasureCollectionEvaluator](#)
Class used to evaluate numeric measure collections.
- class [NumericMeasureCollectionVisitor](#)
Class for evaluating numeric measure collections.
- class [NumericSpatialMeasureAttribute](#)
Class for representing a numeric spatial measure attribute.
- class [NumericStateVariableAttribute](#)
Class for representing a numeric state variable attribute.
- class [NumericStateVariableGrammar](#)
The grammar for parsing numeric state variable statements.
- class [NumericStateVariableId](#)
Class for representing the identity (name, type) of a numeric state variable.
- class [NumericStatisticalMeasureAttribute](#)
Class for representing a numeric statistical measure attribute.
- class [NumericVisitor](#)
Class for evaluating numeric measures.
- class [OrConstraintAttribute](#)
Class for representing an "or" constraint attribute.
- class [OrLogicPropertyAttribute](#)
Class for representing an "or" logic property attribute.
- class [Parser](#)

- class [ParserGrammarExceptionHandler](#)
Class used for parsing (P)BLSTL logical queries.
- class [ParserGrammarExtraInputException](#)
Class for handling parser grammar exceptions.
- class [ParserGrammarProbabilityException](#)
Class for representing "extra input" exceptions in the parsing process.
- class [ParserGrammarUnexpectedTokenException](#)
Class for representing "unexpected token" exceptions in the parsing process.
- class [ParserGrammarUnparseableInputException](#)
Class for representing "unparseable input" exceptions in the parsing process.
- class [PrimaryConstraintAttribute](#)
Class for representing a primary constraint attribute.
- class [PrimaryLogicPropertyAttribute](#)
Class for representing a primary logic property attribute.
- class [PrimaryNumericMeasureAttribute](#)
Class for representing a primary numeric measure attribute.
- singleton [PrimaryNumericMeasureGrammar](#)
The grammar for parsing primary numeric measure statements.
- class [ProbabilisticBlackBoxModelChecker](#)
Class used to run probabilistic black-box model checking tasks.
- class [ProbabilisticBlackBoxModelCheckerFactory](#)
Class for creating [ProbabilisticBlackBoxModelChecker](#) instances.
- class [ProbabilisticLogicPropertyAttribute](#)
Class for representing a probabilistic logic property attribute.
- struct [ProbabilityErrorHandler](#)
Structure for defining the error handler for invalid probability errors.
- class [Region](#)
Class for representing a region.
- class [SemanticType](#)
Enumeration for defining a semantic type.
- class [SemanticTypeAttribute](#)
Class for representing a semantic type attribute.
- class [SemanticTypeGrammar](#)
The grammar for parsing semantic type statements.
- class [SemanticTypeStringGrammar](#)
The grammar for parsing semantic type string statements.
- class [SimilarityMeasureAttribute](#)
Class for representing a similarity measure attribute.
- struct [SimilarityMeasureTypeParser](#)
Symbol table and parser for the similarity measure type.
- class [SimilarityTemporalNumericCollectionAttribute](#)
Class for representing a similarity temporal numeric collection attribute.
- class [SpatialEntity](#)
Class for representing a pseudo-3D spatial entity.
- class [SpatialMeasureAttribute](#)
Class for representing a spatial measure attribute.
- class [SpatialMeasureCollectionAttribute](#)
Class used to represent a spatial measure collection attribute.
- class [SpatialMeasureCollectionGrammar](#)
The grammar for parsing spatial measure collection statements.

- class [SpatialMeasureEvaluator](#)
Class for evaluating spatial measures.
- struct [SpatialMeasureTypeParser](#)
Symbol table and parser for the spatial measure type.
- class [SpatialNumericComparisonAttribute](#)
Class for representing a spatial numeric comparison attribute.
- class [SpatialTemporalDataReader](#)
Class for reading spatial temporal trace data from input files.
- class [SpatialTemporalException](#)
Class for representing a spatial temporal exception.
- class [SpatialTemporalTrace](#)
Class for representing a spatial temporal trace.
- class [StateVariableAttribute](#)
Class for representing a state variable attribute.
- class [StatisticalModelChecker](#)
Class used to run statistical model checking tasks.
- class [StatisticalModelCheckerFactory](#)
Class for creating [StatisticalModelChecker](#) instances.
- class [SubsetAttribute](#)
Class for representing a subset attribute.
- class [SubsetOperationAttribute](#)
Class for representing a subset operation attribute.
- struct [SubsetOperationTypeParser](#)
Symbol table and parser for the subset operation type.
- class [SubsetSpecificAttribute](#)
Class for representing a subset specific attribute.
- struct [SubsetSpecificTypeParser](#)
Symbol table and parser for a specific subset type.
- class [SubsetSubsetOperationAttribute](#)
Class for representing a subset subset operation attribute.
- class [SubsetVisitor](#)
Class used to evaluate subsets.
- class [TemporalDataReader](#)
Class for reading (non-spatial) timeseries data from a .csv file.
- class [TemporalNumericCollectionAttribute](#)
Class for representing a temporal numeric collection attribute.
- class [TemporalNumericCollectionGrammar](#)
The grammar for parsing temporal numeric collection statements.
- class [TemporalNumericComparisonAttribute](#)
Class for representing a temporal numeric comparison attribute.
- class [TemporalNumericMeasureAttribute](#)
Class for representing a temporal numeric measure attribute.
- class [TemporalNumericMeasureCollectionAttribute](#)
Class for representing temporal numeric measure collection attributes.
- class [TemporalNumericMeasureGrammar](#)
The grammar for parsing temporal numeric measure statements.
- class [TemporalNumericVisitor](#)
Class for evaluating temporal numeric measures.
- class [TimePoint](#)
Class for representing a timepoint.
- class [TimePointEvaluator](#)

- class [TimeseriesComponentAttribute](#)

Class used to evaluate timepoints.
- class [TimeseriesComponentEvaluator](#)

Class for representing a timeseries component attribute.
- class [TimeseriesComponentVisitor](#)

Class for evaluating timeseries components.
- class [TimeseriesMeasureAttribute](#)

Class for representing a timeseries measure attribute.
- struct [TimeseriesMeasureTypeParser](#)

Symbol table and parser for the timeseries measure type.
- class [TimeseriesTimeseriesComponentAttribute](#)

Class for representing a timeseries timeseries component attribute.
- class [TypeSemanticsTable](#)

Class for defining a type semantics table.
- class [UnaryNumericFilterAttribute](#)

Class for representing a unary numeric filter attribute.
- class [UnaryNumericMeasureAttribute](#)

Class for representing a unary numeric measure attribute.
- class [UnaryNumericMeasureGrammar](#)

The grammar for parsing unary numeric measure statements.
- struct [UnaryNumericMeasureTypeParser](#)

Symbol table and parser for the unary numeric measure type.
- class [UnaryNumericNumericAttribute](#)

Class for representing a unary numeric numeric measure attribute.
- class [UnaryNumericTemporalAttribute](#)

Class for representing a unary numeric temporal measure attribute.
- class [UnarySpatialConstraintAttribute](#)

Class for representing a "unary" spatial constraint attribute.
- class [UnaryStatisticalMeasureAttribute](#)

Class for representing a unary statistical measure attribute.
- class [UnaryStatisticalMeasureGrammar](#)

The grammar for parsing unary statistical measure statements.
- struct [UnaryStatisticalMeasureTypeParser](#)

Symbol table and parser for the unary statistical measure type.
- class [UnaryStatisticalNumericAttribute](#)

Class for representing a unary statistical numeric attribute.
- class [UnaryStatisticalSpatialAttribute](#)

Class for representing a unary statistical spatial attribute.
- class [UnaryTypeConstraintAttribute](#)

Class for representing a "unary" type constraint attribute.
- struct [UnexpectedTokenErrorHandler](#)

Structure for defining the error handler for unexpected token errors.
- class [UntilLogicPropertyAttribute](#)

Class for representing an "until" logic property attribute.

Typedefs

- ```
typedef boost::variant< Nil,
boost::recursive_wrapper
< ConstraintAttribute >
, boost::recursive_wrapper
< OrConstraintAttribute >
, boost::recursive_wrapper
< AndConstraintAttribute >
, boost::recursive_wrapper
< ImplicationConstraintAttribute >
, boost::recursive_wrapper
< EquivalenceConstraintAttribute >
, boost::recursive_wrapper
< PrimaryConstraintAttribute > > ConstraintAttributeType
```

*Variant for a constraint attribute type.*
- ```
typedef boost::variant
< SpatialMeasureAttribute,
boost::recursive_wrapper
< PrimaryNumericMeasureAttribute >
, boost::recursive_wrapper
< UnaryNumericFilterAttribute >
, boost::recursive_wrapper
< BinaryNumericFilterAttribute >
, boost::recursive_wrapper
< FilterNumericMeasureAttribute > > FilterNumericMeasureAttributeType
```

Variant for a filter numeric measure attribute.
- ```
typedef boost::variant< Nil,
boost::recursive_wrapper
< LogicPropertyAttribute >
, boost::recursive_wrapper
< OrLogicPropertyAttribute >
, boost::recursive_wrapper
< AndLogicPropertyAttribute >
, boost::recursive_wrapper
< ImplicationLogicPropertyAttribute >
, boost::recursive_wrapper
< EquivalenceLogicPropertyAttribute >
, boost::recursive_wrapper
< UntilLogicPropertyAttribute >
, boost::recursive_wrapper
< PrimaryLogicPropertyAttribute > > LogicPropertyAttributeType
```

*Variant for the logic property attribute.*
- ```
typedef boost::variant< double,
NumericStateVariableAttribute,
boost::recursive_wrapper
< NumericSpatialMeasureAttribute >
, boost::recursive_wrapper
< PrimaryNumericMeasureAttribute >
, boost::recursive_wrapper
< UnaryNumericNumericAttribute >
, boost::recursive_wrapper
< BinaryNumericNumericAttribute >
, boost::recursive_wrapper
< NumericMeasureAttribute > > NumericMeasureType
```

Variant for the numeric measure attribute.

- `typedef boost::variant< SpatialMeasureCollectionAttribute, TemporalNumericCollectionAttribute > NumericMeasureCollectionType`
Variant for the numeric measure collection attribute.
- `typedef boost::variant< UnaryStatisticalSpatialAttribute, BinaryStatisticalSpatialAttribute, BinaryStatisticalQuantileSpatialAttribute, boost::recursive_wrapper< NumericSpatialMeasureAttribute > > NumericSpatialMeasureType`
Variant for a numeric spatial measure attribute.
- `typedef boost::variant< UnaryStatisticalNumericAttribute, BinaryStatisticalNumericAttribute, BinaryStatisticalQuantileNumericAttribute > NumericStatisticalMeasureType`
Variant for the numeric statistical measure attribute.
- `typedef boost::variant< Nil, boost::recursive_wrapper< ConstraintAttribute >, boost::recursive_wrapper< NotConstraintAttribute >, boost::recursive_wrapper< UnarySpatialConstraintAttribute >, boost::recursive_wrapper< UnaryTypeConstraintAttribute > > PrimaryConstraintAttributeType`
Variant for a primary constraint attribute.
- `typedef boost::variant< TemporalNumericComparisonAttribute, ChangeTemporalNumericMeasureAttribute, SimilarityTemporalNumericCollectionAttribute, boost::recursive_wrapper< NotLogicPropertyAttribute >, boost::recursive_wrapper< FutureLogicPropertyAttribute >, boost::recursive_wrapper< GlobalLogicPropertyAttribute >, boost::recursive_wrapper< NextLogicPropertyAttribute >, boost::recursive_wrapper< NextKLogicPropertyAttribute >, boost::recursive_wrapper< LogicPropertyAttribute > > PrimaryLogicPropertyAttributeType`
Variant for representing a primary logic property type.
- `typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive_wrapper< NumericSpatialMeasureAttribute >, boost::recursive_wrapper< PrimaryNumericMeasureAttribute > > PrimaryNumericMeasureAttributeType`
Variant for the primary numeric measure attribute.
- `typedef boost::variant< SubsetSpecificAttribute, FilterSubsetAttribute, boost::recursive_wrapper< SubsetSubsetOperationAttribute >, boost::recursive_wrapper< SubsetAttribute > > SubsetAttributeType`

- *Variant for a subset attribute.*
- ```
typedef boost::variant<
 < TemporalNumericMeasureCollectionAttribute,
 boost::recursive_wrapper< ChangeTemporalNumericCollectionAttribute >>
, boost::recursive_wrapper< TimeseriesTimeseriesComponentAttribute >
, boost::recursive_wrapper< HomogeneousHomogeneousTimeseriesAttribute >> TemporalNumericCollectionType
```

*Variant for the temporal numeric collection attribute.*
- ```
typedef boost::variant< double,
    NumericStateVariableAttribute,
    NumericStatisticalMeasureAttribute,
    boost::recursive_wrapper< UnaryNumericTemporalAttribute >
, boost::recursive_wrapper< BinaryNumericTemporalAttribute >
, boost::recursive_wrapper< TemporalNumericMeasureAttribute >> TemporalNumericMeasureType
```

Variant for the temporal numeric measure attribute.
- ```
typedef boost::variant< HeterogeneousTimeseriesComponentAttribute,
 HomogeneousTimeseriesComponentAttribute > TimeseriesComponentType
```

*Variant for the timeseries component attribute.*

## Enumerations

- enum `BinaryNumericMeasureType` : unsigned int {
 `BinaryNumericMeasureType::Add` = 0, `BinaryNumericMeasureType::Div`, `BinaryNumericMeasureType::Log`,  
`BinaryNumericMeasureType::Mod`,  
`BinaryNumericMeasureType::Multiply`, `BinaryNumericMeasureType::Power`, `BinaryNumericMeasureType::Subtract` }
- Enumeration for representing a binary numeric measure type.*
- enum `BinaryStatisticalMeasureType` : unsigned int { `BinaryStatisticalMeasureType::Covar` = 0 }
- Enumeration for representing a binary statistical measure type.*
- enum `BinaryStatisticalQuantileMeasureType` : unsigned int { `BinaryStatisticalQuantileMeasureType::Percentile` = 0, `BinaryStatisticalQuantileMeasureType::Quartile` }
- Enumeration for representing a binary statistical quantile measure type.*
- enum `ChangeMeasureType` : unsigned int { `ChangeMeasureType::Derivative` = 0, `ChangeMeasureType::Ratio` }
- Enumeration for representing a change measure type.*
- enum `ComparatorType` : unsigned int {
 `ComparatorType::GreaterThan` = 0, `ComparatorType::GreaterThanOrEqual`, `ComparatorType::LessThan`,  
`ComparatorType::LessThanOrEqual`,  
`ComparatorType::Equal` }
- Enumeration for representing a comparator type.*
- enum `HeterogeneousTimeseriesComponentType` : unsigned int { `HeterogeneousTimeseriesComponentType::Peak` = 0, `HeterogeneousTimeseriesComponentType::Valley` }
- Enumeration for representing a heterogeneous timeseries component type.*
- enum `HomogeneousTimeseriesComponentType` : unsigned int {
 `HomogeneousTimeseriesComponentType::Ascent` = 0, `HomogeneousTimeseriesComponentType::Descent`,  
`HomogeneousTimeseriesComponentType::Plateau`, `HomogeneousTimeseriesComponentType::UniformAscent`,  
`HomogeneousTimeseriesComponentType::UniformDescent` }
- Enumeration for representing a homogeneous timeseries component type.*

- enum `HomogeneousTimeseriesMeasureType` : unsigned int { `HomogeneousTimeseriesMeasureType::TimeSpan` = 0, `HomogeneousTimeseriesMeasureType::Values` }
 

*Enumeration for representing a homogeneous timeseries measure type.*
- enum `SimilarityMeasureType` : unsigned int { `SimilarityMeasureType::Opposite` = 0, `SimilarityMeasureType::Similar` }
 

*Enumeration for representing a similarity measure type.*
- enum `SpatialMeasureType` : unsigned int {
 `SpatialMeasureType::Clusteredness` = 0, `SpatialMeasureType::Density`, `SpatialMeasureType::Area`, `SpatialMeasureType::Perimeter`,  
`SpatialMeasureType::DistanceFromOrigin`, `SpatialMeasureType::Angle`, `SpatialMeasureType::TriangleMeasure`,  
`SpatialMeasureType::RectangleMeasure`, `SpatialMeasureType::CircleMeasure`, `SpatialMeasureType::CentroidX`, `SpatialMeasureType::CentroidY`,  
`SpatialMeasureType::NrOfSpatialMeasureTypeEntries` }
 

*Enumeration for representing the types of spatial measures.*
- enum `SubsetOperationType` : unsigned int { `SubsetOperationType::Difference` = 0, `SubsetOperationType::Intersection`, `SubsetOperationType::Union` }
 

*Enumeration for representing the types of subset operations.*
- enum `SubsetSpecificType` : unsigned int { `SubsetSpecificType::Clusters` = 0, `SubsetSpecificType::Regions`, `SubsetSpecificType::NrOfSubsetSpecificTypeEntries` }
 

*Enumeration for representing a specific subset type.*
- enum `TimeseriesMeasureType` : unsigned int { `TimeseriesMeasureType::EnteringTime` = 0, `TimeseriesMeasureType::EnteringValue` }
 

*Enumeration for representing a timeseries measure type.*
- enum `UnaryNumericMeasureType` : unsigned int {
 `UnaryNumericMeasureType::Abs` = 0, `UnaryNumericMeasureType::Ceil`, `UnaryNumericMeasureType::Floor`,  
`UnaryNumericMeasureType::Round`, `UnaryNumericMeasureType::Sign`, `UnaryNumericMeasureType::Sqrt`, `UnaryNumericMeasureType::Trunc` }
 

*Enumeration for representing a unary numeric measure type.*
- enum `UnaryStatisticalMeasureType` : unsigned int {
 `UnaryStatisticalMeasureType::Avg` = 0, `UnaryStatisticalMeasureType::Count`, `UnaryStatisticalMeasureType::Geomean`,  
`UnaryStatisticalMeasureType::Harmean`, `UnaryStatisticalMeasureType::Kurt`, `UnaryStatisticalMeasureType::Max`, `UnaryStatisticalMeasureType::Median`,  
`UnaryStatisticalMeasureType::Min`, `UnaryStatisticalMeasureType::Mode`, `UnaryStatisticalMeasureType::Product`, `UnaryStatisticalMeasureType::Skew`,  
`UnaryStatisticalMeasureType::Stdev`, `UnaryStatisticalMeasureType::Sum`, `UnaryStatisticalMeasureType::Var` }
 

*Enumeration for representing a unary statistical measure type.*
- enum `ApproximateBayesianModelCheckingResult` : unsigned int { `ApproximateBayesianModelCheckingResult::TRUE` = 0, `ApproximateBayesianModelCheckingResult::FALSE`, `ApproximateBayesianModelCheckingResult::MORE_TRACES_REQUIRED` }
 

*Enumeration for representing the model checking result.*
- enum `BayesianModelCheckingResult` : unsigned int { `BayesianModelCheckingResult::TRUE` = 0, `BayesianModelCheckingResult::FALSE`, `BayesianModelCheckingResult::MORE_TRACES_REQUIRED` }
 

*Enumeration for representing the model checking result.*
- enum `StatisticalModelCheckingResult` : int { `StatisticalModelCheckingResult::TRUE` = 0, `StatisticalModelCheckingResult::FALSE`, `StatisticalModelCheckingResult::UNDECIDED`, `StatisticalModelCheckingResult::MORE_TRACES_REQUIRED` }
 

*Enumeration for representing the model checking result.*

## Functions

- `std::ostream & operator<< (std::ostream &out, const BinaryNumericMeasureType &binaryNumericMeasureType)`

*Overload the output stream operator for the enumeration.*

- `std::ostream & operator<< (std::ostream &out, const BinaryStatisticalMeasureType &binaryStatisticalMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const BinaryStatisticalQuantileMeasureType &binaryStatisticalQuantileMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const ChangeMeasureType &changeMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const ComparatorType &comparatorType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const HeterogeneousTimeseriesComponentType &heterogeneousTimeseriesComponentType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const HomogeneousTimeseriesComponentType &homogeneousTimeseriesComponentType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const HomogeneousTimeseriesMeasureType &homogeneousTimeseriesMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const SimilarityMeasureType &similarityMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const SpatialMeasureType &spatialMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const SubsetOperationType &subsetOperationType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const SubsetSpecificType &subsetSpecificType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const TimeseriesMeasureType &timeseriesMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const UnaryNumericMeasureType &unaryNumericMeasureType)`  
*Overload the output stream operator for the enumeration.*
- `std::ostream & operator<< (std::ostream &out, const UnaryStatisticalMeasureType &unaryStatisticalMeasureType)`  
*Overload the output stream operator for the enumeration.*

## Variables

- `static const std::size_t NR_SPATIAL_MEASURE_TYPES = static_cast<std::size_t>(SpatialMeasureType::NrOfSpatialMeasureEntries)`  
*An std::size\_t constant which stores the number of spatial measure type entries.*
- `static const std::size_t NR_SUBSET_SPECIFIC_TYPES = static_cast<std::size_t>(SubsetSpecificType::NrOfSubsetSpecificTypeEntries)`  
*An std::size\_t constant which stores the number of subset specific type entries.*
- phoenix::function  
`< UnexpectedErrorHandler >`  
`const handleUnexpectedErrorHandler = UnexpectedErrorHandler()`
- phoenix::function  
`< ProbabilityErrorHandler >`  
`const handleProbabilityError = ProbabilityErrorHandler()`
- `static const std::string WRN_LOGIC_PROPERTY_EVAL_FALSE = "The enclosing logic property was evaluated to the default value \"false\"."`
- `static const std::string WRN_OUTPUT_SEPARATOR = " "`

### 6.3.1 Typedef Documentation

6.3.1.1 `typedef boost::variant< Nil, boost::recursive_wrapper<ConstraintAttribute>, boost::recursive_wrapper<OrConstraintAttribute>, boost::recursive_wrapper<AndConstraintAttribute>, boost::recursive_wrapper<ImplicationConstraintAttribute>, boost::recursive_wrapper<EquivalenceConstraintAttribute>, boost::recursive_wrapper<PrimaryConstraintAttribute> > multiscale::verification::ConstraintAttributeType`

Variant for a constraint attribute type.

Definition at line 20 of file ConstraintAttribute.hpp.

6.3.1.2 `typedef boost::variant< SpatialMeasureAttribute, boost::recursive_wrapper<PrimaryNumericMeasureAttribute>, boost::recursive_wrapper<UnaryNumericFilterAttribute>, boost::recursive_wrapper<BinaryNumericFilterAttribute>, boost::recursive_wrapper<FilterNumericMeasureAttribute> > multiscale::verification::FilterNumericMeasureAttributeType`

Variant for a filter numeric measure attribute.

Definition at line 18 of file FilterNumericMeasureAttribute.hpp.

6.3.1.3 `typedef boost::variant< Nil, boost::recursive_wrapper<LogicPropertyAttribute>, boost::recursive_wrapper<OrLogicPropertyAttribute>, boost::recursive_wrapper<AndLogicPropertyAttribute>, boost::recursive_wrapper<ImplicationLogicPropertyAttribute>, boost::recursive_wrapper<EquivalenceLogicPropertyAttribute>, boost::recursive_wrapper<UntilLogicPropertyAttribute>, boost::recursive_wrapper<PrimaryLogicPropertyAttribute> > multiscale::verification::LogicPropertyAttributeType`

Variant for the logic property attribute.

Definition at line 23 of file LogicPropertyAttribute.hpp.

6.3.1.4 `typedef boost::variant< SpatialMeasureCollectionAttribute, TemporalNumericCollectionAttribute > multiscale::verification::NumericMeasureCollectionType`

Variant for the numeric measure collection attribute.

Definition at line 18 of file NumericMeasureCollectionAttribute.hpp.

6.3.1.5 `typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive_wrapper<NumericSpatialMeasureAttribute>, boost::recursive_wrapper<PrimaryNumericMeasureAttribute>, boost::recursive_wrapper<UnaryNumericNumericAttribute>, boost::recursive_wrapper<BinaryNumericNumericAttribute>, boost::recursive_wrapper<NumericMeasureAttribute> > multiscale::verification::NumericMeasureType`

Variant for the numeric measure attribute.

Definition at line 19 of file NumericMeasureAttribute.hpp.

6.3.1.6 `typedef boost::variant< UnaryStatisticalSpatialAttribute, BinaryStatisticalSpatialAttribute, BinaryStatisticalQuantileSpatialAttribute, boost::recursive_wrapper<NumericSpatialMeasureAttribute> > multiscale::verification::NumericSpatialMeasureType`

Variant for a numeric spatial measure attribute.

Definition at line 16 of file NumericSpatialMeasureAttribute.hpp.

---

**6.3.1.7 `typedef boost::variant< UnaryStatisticalNumericAttribute, BinaryStatisticalNumericAttribute, BinaryStatisticalQuantileNumericAttribute > multiscale::verification::NumericStatisticalMeasureType`**

Variant for the numeric statistical measure attribute.

Definition at line 20 of file `NumericStatisticalMeasureAttribute.hpp`.

**6.3.1.8 `typedef boost::variant< Nil, boost::recursive_wrapper<ConstraintAttribute>, boost::recursive_wrapper<NotConstraintAttribute>, boost::recursive_wrapper<UnarySpatialConstraintAttribute>, boost::recursive_wrapper<UnaryTypeConstraintAttribute> > multiscale::verification::PrimaryConstraintAttributeType`**

Variant for a primary constraint attribute.

Definition at line 18 of file `PrimaryConstraintAttribute.hpp`.

**6.3.1.9 `typedef boost::variant< TemporalNumericComparisonAttribute, ChangeTemporalNumericMeasureAttribute, SimilarityTemporalNumericCollectionAttribute, boost::recursive_wrapper<NotLogicPropertyAttribute>, boost::recursive_wrapper<FutureLogicPropertyAttribute>, boost::recursive_wrapper<GlobalLogicPropertyAttribute>, boost::recursive_wrapper<NextLogicPropertyAttribute>, boost::recursive_wrapper<NextKLogicPropertyAttribute>, boost::recursive_wrapper<LogicPropertyAttribute> > multiscale::verification::PrimaryLogicPropertyAttributeType`**

Variant for representing a primary logic property type.

Definition at line 22 of file `PrimaryLogicPropertyAttribute.hpp`.

**6.3.1.10 `typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive_wrapper<NumericSpatialMeasureAttribute>, boost::recursive_wrapper<PrimaryNumericMeasureAttribute> > multiscale::verification::PrimaryNumericMeasureAttributeType`**

Variant for the primary numeric measure attribute.

Definition at line 15 of file `PrimaryNumericMeasureAttribute.hpp`.

**6.3.1.11 `typedef boost::variant< SubsetSpecificAttribute, FilterSubsetAttribute, boost::recursive_wrapper<SubsetSubsetOperationAttribute>, boost::recursive_wrapper<SubsetAttribute> > multiscale::verification::SubsetAttributeType`**

Variant for a subset attribute.

Definition at line 16 of file `SubsetAttribute.hpp`.

**6.3.1.12 `typedef boost::variant< TemporalNumericMeasureCollectionAttribute, boost::recursive_wrapper<ChangeTemporalNumericCollectionAttribute>, boost::recursive_wrapper<TimeseriesTimeseriesComponentAttribute>, boost::recursive_wrapper<HomogeneousHomogeneousTimeseriesAttribute> > multiscale::verification::TemporalNumericCollectionType`**

Variant for the temporal numeric collection attribute.

Definition at line 16 of file `TemporalNumericCollectionAttribute.hpp`.

6.3.1.13 `typedef boost::variant< double, NumericStateVariableAttribute, NumericStatisticalMeasureAttribute, boost::recursive_wrapper<UnaryNumericTemporalAttribute>, boost::recursive_wrapper<BinaryNumericTemporalAttribute>, boost::recursive_wrapper<TemporalNumericMeasureAttribute> > multiscale::verification::TemporalNumericMeasureType`

Variant for the temporal numeric measure attribute.

Definition at line 17 of file TemporalNumericMeasureAttribute.hpp.

6.3.1.14 `typedef boost::variant< HeterogeneousTimeseriesComponentAttribute, HomogeneousTimeseriesComponentAttribute > multiscale::verification::TimeseriesComponentType`

Variant for the timeseries component attribute.

Definition at line 19 of file TimeseriesComponentAttribute.hpp.

## 6.3.2 Enumeration Type Documentation

6.3.2.1 `enum multiscale::verification::ApproximateBayesianModelCheckingResult : unsigned int [strong]`

Enumeration for representing the model checking result.

Enumerator

**TRUE** The logic property was evaluated to true

**FALSE** The logic property was evaluated to false

**MORE\_TRACES\_REQUIRED** More traces are required to determine the truth value of the logic property

Definition at line 15 of file ApproximateBayesianModelChecker.hpp.

6.3.2.2 `enum multiscale::verification::BayesianModelCheckingResult : unsigned int [strong]`

Enumeration for representing the model checking result.

Enumerator

**TRUE** The logic property was evaluated to true

**FALSE** The logic property was evaluated to false

**MORE\_TRACES\_REQUIRED** More traces are required to determine the truth value of the logic property

Definition at line 15 of file BayesianModelChecker.hpp.

6.3.2.3 `enum multiscale::verification::BinaryNumericMeasureType : unsigned int [strong]`

Enumeration for representing a binary numeric measure type.

Enumerator

**Add** Addition

**Div** Division

**Log** Logarithm

**Mod** Remainder of division

**Multiply** Multiplication

**Power** Raise to power

**Subtract** Subtraction

Definition at line 13 of file BinaryNumericMeasureAttribute.hpp.

---

**6.3.2.4 enum multiscale::verification::BinaryStatisticalMeasureType : unsigned int [strong]**

Enumeration for representing a binary statistical measure type.

**Enumerator**

**Covar** Covariance

Definition at line 13 of file BinaryStatisticalMeasureAttribute.hpp.

**6.3.2.5 enum multiscale::verification::BinaryStatisticalQuantileMeasureType : unsigned int [strong]**

Enumeration for representing a binary statistical quantile measure type.

**Enumerator**

**Percentile** The percentile

**Quartile** The quartile

Definition at line 13 of file BinaryStatisticalQuantileMeasureAttribute.hpp.

**6.3.2.6 enum multiscale::verification::ChangeMeasureType : unsigned int [strong]**

Enumeration for representing a change measure type.

**Enumerator**

**Derivative** Derivative representing rate of change

**Ratio** Ratio of value change over difference in time change

Definition at line 13 of file ChangeMeasureAttribute.hpp.

**6.3.2.7 enum multiscale::verification::ComparatorType : unsigned int [strong]**

Enumeration for representing a comparator type.

**Enumerator**

**GreaterThan** Greater than

**GreaterThanOrEqual** Greater than or equal

**LessThan** Less than

**LessThanOrEqual** Less than or equal

**Equal** Equal

Definition at line 13 of file ComparatorAttribute.hpp.

**6.3.2.8 enum multiscale::verification::HeterogeneousTimeseriesComponentType : unsigned int [strong]**

Enumeration for representing a heterogeneous timeseries component type.

**Enumerator**

**Peak** The peak of the timeseries

**Valley** The value of the timeseries

Definition at line 13 of file HeterogeneousTimeseriesComponentAttribute.hpp.

**6.3.2.9 enum multiscale::verification::HomogeneousTimeseriesComponentType : unsigned int [strong]**

Enumeration for representing a homogeneous timeseries component type.

**Enumerator**

**Ascent** The ascending timeseries type

**Descent** The descending timeseries type

**Plateau** The plateau timeseries type

**UniformAscent** The uniformly (constantly) ascending timeseries type

**UniformDescent** The uniformly (constantly) descending timeseries type

Definition at line 13 of file HomogeneousTimeseriesComponentAttribute.hpp.

**6.3.2.10 enum multiscale::verification::HomogeneousTimeseriesMeasureType : unsigned int [strong]**

Enumeration for representing a homogeneous timeseries measure type.

**Enumerator**

**TimeSpan** The time span covered by the homogeneous timeseries

**Values** The values defining the homogeneous timeseries

Definition at line 13 of file HomogeneousTimeseriesMeasureAttribute.hpp.

**6.3.2.11 enum multiscale::verification::SimilarityMeasureType : unsigned int [strong]**

Enumeration for representing a similarity measure type.

**Enumerator**

**Opposite** The opposite type

**Similar** The similar type

Definition at line 13 of file SimilarityMeasureAttribute.hpp.

**6.3.2.12 enum multiscale::verification::SpatialMeasureType : unsigned int [strong]**

Enumeration for representing the types of spatial measures.

**Enumerator**

**Clusteredness** The clusteredness of a spatial entity

**Density** The density of a spatial entity

**Area** The area of a spatial entity

**Perimeter** The perimeter of a spatial entity

**DistanceFromOrigin** The distance of a spatial entity from origin

**Angle** The angle

**TriangleMeasure** The measure indicating how triangular is the shape of the spatial entity

**RectangleMeasure** The measure indicating how rectangular is the shape of the spatial entity

**CircleMeasure** The measure indicating how circular is the shape of the spatial entity

**CentroidX** The x coordinate of the spatial entity centroid

**CentroidY** The y coordinate of the spatial entity centroid

**NrOfSpatialMeasureTypeEntries** Enumeration type used to store the number of elements in the enumeration. Always leave it last!

Definition at line 19 of file SpatialMeasureType.hpp.

**6.3.2.13 enum multiscale::verification::StatisticalModelCheckingResult : int [strong]**

Enumeration for representing the model checking result.

**Enumerator**

**TRUE** The logic property was evaluated to true

**FALSE** The logic property was evaluated to false

**UNDECIDED** The truth value of the logic property is undecided

**MORE\_TRACES\_REQUIRED** More traces are required to determine the truth value of the logic property

Definition at line 15 of file StatisticalModelChecker.hpp.

**6.3.2.14 enum multiscale::verification::SubsetOperationType : unsigned int [strong]**

Enumeration for representing the types of subset operations.

**Enumerator**

**Difference** Difference of two subsets

**Intersection** Intersection of two subsets

**Union** Union of two subsets

Definition at line 13 of file SubsetOperationAttribute.hpp.

**6.3.2.15 enum multiscale::verification::SubsetSpecificType : unsigned int [strong]**

Enumeration for representing a specific subset type.

**Enumerator**

**Clusters** Clusters

**Regions** Regions

**NrOfSubsetSpecificTypeEntries** Enumeration type used to store the number of elements in the enumeration.  
Always leave it last!

Definition at line 19 of file SubsetSpecificType.hpp.

**6.3.2.16 enum multiscale::verification::TimeseriesMeasureType : unsigned int [strong]**

Enumeration for representing a timeseries measure type.

**Enumerator**

**EnteringTime** The entering time

**EnteringValue** The entering value

Definition at line 13 of file TimeseriesMeasureAttribute.hpp.

## 6.3.2.17 enum multiscale::verification::UnaryNumericMeasureType : unsigned int [strong]

Enumeration for representing a unary numeric measure type.

Enumerator

- Abs** Absolute value
- Ceil** Ceiling
- Floor** Floor
- Round** Round
- Sign** Sign: -1 (-), +1 (+) or 0 (0)
- Sqrt** Square root
- Trunc** Truncation

Definition at line 13 of file UnaryNumericMeasureAttribute.hpp.

## 6.3.2.18 enum multiscale::verification::UnaryStatisticalMeasureType : unsigned int [strong]

Enumeration for representing a unary statistical measure type.

Enumerator

- Avg** The average (arithmetic mean)
- Count** The cardinality of a collection
- Geomean** The geometric mean
- Harmean** The harmonic mean
- Kurt** The kurtosis
- Max** The maximum
- Median** The median
- Min** The minimum
- Mode** The mode
- Product** The product
- Skew** The skew
- Stdev** The standard deviation
- Sum** The sum
- Var** The variance

Definition at line 13 of file UnaryStatisticalMeasureAttribute.hpp.

### 6.3.3 Function Documentation

## 6.3.3.1 std::ostream &amp; multiscale::verification::operator&lt;&lt;( std::ostream &amp; out, const BinaryStatisticalMeasureType &amp; binaryStatisticalMeasureType )

Overload the output stream operator for the enumeration.

Parameters

---

|                                                           |                                                       |
|-----------------------------------------------------------|-------------------------------------------------------|
| <i>out</i>                                                | Output stream                                         |
| <i>binary</i><br><i>Statistical</i><br><i>MeasureType</i> | The binary statistical measure type to be printed out |

Definition at line 8 of file BinaryStatisticalMeasureAttribute.cpp.

References Covar.

**6.3.3.2 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const HomogeneousTimeseriesMeasureType & *homogeneousTimeseriesMeasureType* )**

Overload the output stream operator for the enumeration.

Parameters

|                                                               |                                                           |
|---------------------------------------------------------------|-----------------------------------------------------------|
| <i>out</i>                                                    | Output stream                                             |
| <i>homogeneous</i><br><i>Timeseries</i><br><i>MeasureType</i> | The homogeneous timeseries measure type to be printed out |

Definition at line 8 of file HomogeneousTimeseriesMeasureAttribute.cpp.

References TimeSpan, and Values.

**6.3.3.3 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const SimilarityMeasureType & *similarityMeasureType* )**

Overload the output stream operator for the enumeration.

Parameters

|                                         |                                               |
|-----------------------------------------|-----------------------------------------------|
| <i>out</i>                              | Output stream                                 |
| <i>similarity</i><br><i>MeasureType</i> | The similarity measure type to be printed out |

Definition at line 8 of file SimilarityMeasureAttribute.cpp.

References Opposite, and Similar.

**6.3.3.4 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const HeterogeneousTimeseriesComponentType & *heterogeneousTimeseriesComponentType* )**

Overload the output stream operator for the enumeration.

Parameters

|                                                                   |                                                               |
|-------------------------------------------------------------------|---------------------------------------------------------------|
| <i>out</i>                                                        | Output stream                                                 |
| <i>heterogeneous</i><br><i>Timeseries</i><br><i>ComponentType</i> | The heterogeneous timeseries component type to be printed out |

Definition at line 8 of file HeterogeneousTimeseriesComponentAttribute.cpp.

References Peak, and Valley.

**6.3.3.5 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const TimeseriesMeasureType & *timeseriesMeasureType* )**

Overload the output stream operator for the enumeration.

## Parameters

|                                         |                                               |
|-----------------------------------------|-----------------------------------------------|
| <i>out</i>                              | Output stream                                 |
| <i>timeseries</i><br><i>MeasureType</i> | The timeseries measure type to be printed out |

Definition at line 8 of file TimeseriesMeasureAttribute.cpp.

References EnteringTime, and EnteringValue.

**6.3.3.6 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const ChangeMeasureType & *changeMeasureType* )**

Overload the output stream operator for the enumeration.

## Parameters

|                                     |                                           |
|-------------------------------------|-------------------------------------------|
| <i>out</i>                          | Output stream                             |
| <i>change</i><br><i>MeasureType</i> | The change measure type to be printed out |

Definition at line 5 of file ChangeMeasureAttribute.cpp.

**6.3.3.7 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const BinaryStatisticalQuantileMeasureType & *binaryStatisticalQuantileMeasureType* )**

Overload the output stream operator for the enumeration.

## Parameters

|                                                                              |                                                               |
|------------------------------------------------------------------------------|---------------------------------------------------------------|
| <i>out</i>                                                                   | Output stream                                                 |
| <i>binary</i><br><i>Statistical</i><br><i>Quantile</i><br><i>MeasureType</i> | The binary statistial quantile measure type to be printed out |

Definition at line 8 of file BinaryStatisticalQuantileMeasureAttribute.cpp.

References Percentile, and Quartile.

**6.3.3.8 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const SubsetOperationType & *subsetOperationType* )**

Overload the output stream operator for the enumeration.

## Parameters

|                                       |                                             |
|---------------------------------------|---------------------------------------------|
| <i>out</i>                            | Output stream                               |
| <i>subset</i><br><i>OperationType</i> | The subset operation type to be printed out |

Definition at line 7 of file SubsetOperationAttribute.cpp.

References Difference, Intersection, and Union.

**6.3.3.9 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const HomogeneousTimeseriesComponentType & *homogeneousTimeseriesComponentType* )**

Overload the output stream operator for the enumeration.

**Parameters**

|                                           |                                                             |
|-------------------------------------------|-------------------------------------------------------------|
| <i>out</i>                                | Output stream                                               |
| <i>homogeneousTimeseriesComponentType</i> | The homogeneous timeseries component type to be printed out |

Definition at line 8 of file HomogeneousTimeseriesComponentAttribute.cpp.

References Ascent, Descent, Plateau, UniformAscent, and UniformDescent.

### 6.3.3.10 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const ComparatorType & *comparatorType* )

Overload the output stream operator for the enumeration.

**Parameters**

|                       |                                       |
|-----------------------|---------------------------------------|
| <i>out</i>            | Output stream                         |
| <i>comparatorType</i> | The comparator type to be printed out |

Definition at line 7 of file ComparatorAttribute.cpp.

References Equal, GreaterThan, GreaterThanOrEqual, LessThan, and LessThanOrEqual.

### 6.3.3.11 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const UnaryNumericMeasureType & *unaryNumericMeasureType* )

Overload the output stream operator for the enumeration.

**Parameters**

|                                |                                                  |
|--------------------------------|--------------------------------------------------|
| <i>out</i>                     | Output stream                                    |
| <i>unaryNumericMeasureType</i> | The unary numeric measure type to be printed out |

Definition at line 7 of file UnaryNumericMeasureAttribute.cpp.

References Abs, Ceil, Floor, Round, Sign, Sqrt, and Trunc.

### 6.3.3.12 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const BinaryNumericMeasureType & *binaryNumericMeasureType* )

Overload the output stream operator for the enumeration.

**Parameters**

|                                 |                                                   |
|---------------------------------|---------------------------------------------------|
| <i>out</i>                      | Output stream                                     |
| <i>binaryNumericMeasureType</i> | The binary numeric measure type to be printed out |

Definition at line 7 of file BinaryNumericMeasureAttribute.cpp.

References Add, Div, Log, Mod, Multiply, Power, and Subtract.

### 6.3.3.13 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const UnaryStatisticalMeasureType & *unaryStatisticalMeasureType* )

Overload the output stream operator for the enumeration.

## Parameters

|                                                          |                                                      |
|----------------------------------------------------------|------------------------------------------------------|
| <i>out</i>                                               | Output stream                                        |
| <i>unary</i><br><i>Statistical</i><br><i>MeasureType</i> | The unary statistical measure type to be printed out |

Definition at line 7 of file UnaryStatisticalMeasureAttribute.cpp.

References Avg, Count, Geomean, Harmeant, Kurt, Max, Median, Min, Mode, Product, Skew, Stdev, Sum, and Var.

### 6.3.3.14 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const SubsetSpecificType & *subsetSpecificType* )

Overload the output stream operator for the enumeration.

Overload the output stream operator for the SubsetSpecificType enumeration.

## Parameters

|                                      |                                            |
|--------------------------------------|--------------------------------------------|
| <i>out</i>                           | Output stream                              |
| <i>subsetSpecific</i><br><i>Type</i> | The specific subset type to be printed out |

Definition at line 17 of file SubsetSpecificAttributeAutoGenerated.cpp.

References Clusters, and Regions.

### 6.3.3.15 std::ostream & multiscale::verification::operator<< ( std::ostream & *out*, const SpatialMeasureType & *spatialMeasureType* )

Overload the output stream operator for the enumeration.

Overload the output stream operator for the SpatialMeasureType enumeration.

## Parameters

|                                      |                                            |
|--------------------------------------|--------------------------------------------|
| <i>out</i>                           | Output stream                              |
| <i>spatial</i><br><i>MeasureType</i> | The spatial measure type to be printed out |

Definition at line 132 of file SpatialMeasureAttributeAutoGenerated.cpp.

References Angle, Area, CentroidX, CentroidY, CircleMeasure, Clusteredness, Density, DistanceFromOrigin, Perimeter, RectangleMeasure, and TriangleMeasure.

## 6.3.4 Variable Documentation

### 6.3.4.1 phoenix::function<ProbabilityErrorHandler> const multiscale::verification::handleProbabilityError = ProbabilityErrorHandler()

Definition at line 25 of file LogicPropertyGrammarDefinition.hpp.

Referenced by multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseProbabilisticLogicProperty< ErrorHandlingSupport() ).

### 6.3.4.2 phoenix::function< UnexpectedTokenErrorHandler > const multiscale::verification::handleUnexpectedToken< Error = UnexpectedTokenErrorHandler() )

Definition at line 25 of file ChangeMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseComposed< ConstraintErrorHandlingSupport(), multiscale::verification::LogicPropertyGrammar< Iterator >::initialise< ComposedLogicPropertyErrorHandlingSupport(), multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialiseErrorHandlingSupport(), multiscale::verification::NumericStateVariableGrammar< Iterator >::initialise< ErrorHandlingSupport(), multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseFilter< NumericMeasureErrorHandlingSupport(), multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseNumericMeasureErrorHandlingSupport(), multiscale::verification::PrimaryNumericMeasure< Grammar< Iterator >::initialiseNumericSpatialMeasureErrorHandlingSupport(), multiscale::verification::Temporal< NumericMeasureGrammar< Iterator >::initialiseNumericStatisticalMeasureErrorHandlingSupport(), multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialisePrimaryConstraintErrorHandlingSupport(), multiscale::verification::LogicPropertyGrammar< Iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseProbabilisticLogicPropertyErrorHandling< Support(), multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasure< CollectionErrorHandlingSupport(), multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSubsetErrorHandlingSupport(), multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTemporalNumericCollectionErrorHandlingSupport(), and multiscale::verification::TemporalNumeric< MeasureGrammar< Iterator >::initialiseTemporalNumericMeasureErrorHandlingSupport().

#### 6.3.4.3 const std::size\_t multiscale::verification::NR\_SPATIAL\_MEASURE\_TYPES = static\_cast<std::size\_t>(SpatialMeasureType::NrOfSpatialMeasureTypeEntries) [static]

An std::size\_t constant which stores the number of spatial measure type entries.

Definition at line 16 of file SpatialMeasureAttribute.hpp.

Referenced by multiscale::verification::SpatialEntity::operator<(), multiscale::verification::SpatialEntity::Spatial< Entity(), multiscale::verification::SpatialEntity::toString(), and multiscale::verification::spatialmeasure::validate< SpatialMeasureTypeIndex() .

#### 6.3.4.4 const std::size\_t multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES = static\_cast<std::size\_t>(SubsetSpecificType::NrOfSubsetSpecificTypeEntries) [static]

An std::size\_t constant which stores the number of subset specific type entries.

Definition at line 16 of file SubsetSpecificAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::evaluateSpatialMeasureConstraint(), multiscale::verification::ConstraintVisitor::evaluateTypeConstraint(), multiscale::verification::TimePoint::getConsidered< SpatialEntities(), multiscale::verification::TimePoint::numberOfSpatialEntities(), printTimePoint(), multiscale::verification::TimePoint::TimePoint(), multiscale::verification::TimePoint::updateSpatialEntities(), and multiscale::verification::subsetspecific::validateSubsetSpecificTypeIndex() .

#### 6.3.4.5 const std::string multiscale::verification::WRN\_LOGIC\_PROPERTY\_EVAL\_FALSE = "The enclosing logic property was evaluated to the default value \"false\"." [static]

Definition at line 19 of file LogicPropertyVisitor.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::printExceptionMessage() .

#### 6.3.4.6 const std::string multiscale::verification::WRN\_OUTPUT\_SEPARATOR = " " [static]

Definition at line 20 of file LogicPropertyVisitor.hpp.

## 6.4 multiscale::verification::spatialmeasure Namespace Reference

## Functions

- void [validateSpatialMeasureType](#) (const [SpatialMeasureType](#) &spatialMeasureType)  
*Check if the given spatial measure type is valid.*
- void [validateSpatialMeasureTypeIndex](#) (const std::size\_t &spatialMeasureTypeIndex)  
*Check if the given spatial measure type index is valid.*
- size\_t [computeSpatialMeasureTypeIndex](#) (const [SpatialMeasureType](#) &spatialMeasureType)  
*Compute the index of the spatial measure type.*
- [SpatialMeasureType computeSpatialMeasureType](#) (const std::size\_t &spatialMeasureTypeIndex)  
*Compute the spatial measure type from the given index.*
- double [getMinValidSpatialMeasureValue](#) (const [SpatialMeasureType](#) &spatialMeasureType)  
*Get the minimum valid value for the given spatial measure type.*
- double [getMaxValidSpatialMeasureValue](#) (const [SpatialMeasureType](#) &spatialMeasureType)  
*Get the maximum valid value for the given spatial measure type.*

### 6.4.1 Function Documentation

#### 6.4.1.1 [SpatialMeasureType multiscale::verification::spatialmeasure::computeSpatialMeasureType](#) ( const std::size\_t & spatialMeasureTypeIndex )

Compute the spatial measure type from the given index.

##### Parameters

|                                |                                      |
|--------------------------------|--------------------------------------|
| <i>spatialMeasureTypeIndex</i> | The given spatial measure type index |
|--------------------------------|--------------------------------------|

Definition at line 33 of file SpatialMeasureAttribute.cpp.

References validateSpatialMeasureTypeIndex().

#### 6.4.1.2 [size\\_t multiscale::verification::spatialmeasure::computeSpatialMeasureTypeIndex](#) ( const [SpatialMeasureType](#) & spatialMeasureType )

Compute the index of the spatial measure type.

##### Parameters

|                           |                                |
|---------------------------|--------------------------------|
| <i>spatialMeasureType</i> | The given spatial measure type |
|---------------------------|--------------------------------|

Definition at line 26 of file SpatialMeasureAttribute.cpp.

References validateSpatialMeasureType().

Referenced by multiscale::verification::SpatialEntity::getSpatialMeasureValue(), and multiscale::verification::SpatialEntity::setSpatialMeasureValue().

#### 6.4.1.3 [double multiscale::verification::spatialmeasure::getMaxValidSpatialMeasureValue](#) ( const [SpatialMeasureType](#) & spatialMeasureType )

Get the maximum valid value for the given spatial measure type.

**Parameters**

|                                      |                                |
|--------------------------------------|--------------------------------|
| <i>spatial</i><br><i>MeasureType</i> | The given spatial measure type |
|--------------------------------------|--------------------------------|

Definition at line 76 of file SpatialMeasureAttributeAutoGenerated.cpp.

References multiscale::verification::Angle, multiscale::verification::Area, multiscale::verification::CentroidX, multiscale::verification::CentroidY, multiscale::verification::CircleMeasure, multiscale::verification::Clusteredness, multiscale::verification::Density, multiscale::verification::DistanceFromOrigin, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, and multiscale::verification::TriangleMeasure.

Referenced by multiscale::verification::SpatialEntity::validateSpatialMeasureValue().

#### 6.4.1.4 double multiscale::verification::spatialmeasure::getMinValidSpatialMeasureValue ( const SpatialMeasureType & *spatialMeasureType* )

Get the minimum valid value for the given spatial measure type.

**Parameters**

|                                      |                                |
|--------------------------------------|--------------------------------|
| <i>spatial</i><br><i>MeasureType</i> | The given spatial measure type |
|--------------------------------------|--------------------------------|

Definition at line 21 of file SpatialMeasureAttributeAutoGenerated.cpp.

References multiscale::verification::Angle, multiscale::verification::Area, multiscale::verification::CentroidX, multiscale::verification::CentroidY, multiscale::verification::CircleMeasure, multiscale::verification::Clusteredness, multiscale::verification::Density, multiscale::verification::DistanceFromOrigin, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, and multiscale::verification::TriangleMeasure.

Referenced by multiscale::verification::SpatialEntity::validateSpatialMeasureValue().

#### 6.4.1.5 void multiscale::verification::spatialmeasure::validateSpatialMeasureType ( const SpatialMeasureType & *spatialMeasureType* )

Check if the given spatial measure type is valid.

**Parameters**

|                                      |                                |
|--------------------------------------|--------------------------------|
| <i>spatial</i><br><i>MeasureType</i> | The given spatial measure type |
|--------------------------------------|--------------------------------|

Definition at line 12 of file SpatialMeasureAttribute.cpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, and multiscale::verification::NrOfSpatialMeasureTypeEntries.

Referenced by computeSpatialMeasureTypeIndex(), multiscale::verification::SpatialMeasureEvaluator::evaluate(), and multiscale::verification::SpatialEntity::validateSpatialMeasureValue().

#### 6.4.1.6 void multiscale::verification::spatialmeasure::validateSpatialMeasureTypeIndex ( const std::size\_t & *spatialMeasureTypeIndex* )

Check if the given spatial measure type index is valid.

**Parameters**

|                                |                                      |
|--------------------------------|--------------------------------------|
| <i>spatialMeasureTypeIndex</i> | The given spatial measure type index |
|--------------------------------|--------------------------------------|

Definition at line 19 of file SpatialMeasureAttribute.cpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, and multiscale::verification::NR\_SPATIAL\_MEASURE\_TYPES.

Referenced by computeSpatialMeasureType().

## 6.5 multiscale::verification::subsetspecific Namespace Reference

**Functions**

- void [validateSubsetSpecificType](#) (const **SubsetSpecificType** &subsetSpecificType)  
*Check if the given subset specific type is valid.*
- void [validateSubsetSpecificTypeIndex](#) (const std::size\_t &subsetSpecificTypeIndex)  
*Check if the given subset specific type index is valid.*
- size\_t [computeSubsetSpecificTypeIndex](#) (const **SubsetSpecificType** &subsetSpecificType)  
*Compute the index of the subset specific type.*
- **SubsetSpecificType** [computeSubsetSpecificType](#) (const std::size\_t &subsetSpecificTypeIndex)  
*Compute the subset specific type from the given index.*

### 6.5.1 Function Documentation

#### 6.5.1.1 **SubsetSpecificType** multiscale::verification::subsetspecific::computeSubsetSpecificType ( const std::size\_t &subsetSpecificTypeIndex )

Compute the subset specific type from the given index.

**Parameters**

|                                |                                      |
|--------------------------------|--------------------------------------|
| <i>subsetSpecificTypeIndex</i> | The given subset specific type index |
|--------------------------------|--------------------------------------|

Definition at line 31 of file SubsetSpecificAttribute.cpp.

References validateSubsetSpecificTypeIndex().

Referenced by multiscale::verification::ConstraintVisitor::evaluateSpatialMeasureConstraint(), multiscale::verification::ConstraintVisitor::evaluateTypeConstraint(), printTimePoint(), and multiscale::verification::TimePoint::updateSpatialEntities().

#### 6.5.1.2 size\_t multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex ( const **SubsetSpecificType** &subsetSpecificType )

Compute the index of the subset specific type.

**Parameters**

|                           |                                |
|---------------------------|--------------------------------|
| <i>subsetSpecificType</i> | The given subset specific type |
|---------------------------|--------------------------------|

Definition at line 24 of file SubsetSpecificAttribute.cpp.

References validateSubsetSpecificType().

Referenced by multiscale::verification::TimePoint::addSpatialEntity(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), multiscale::verification::TimePoint::removeSpatialEntity(), and multiscale::verification::TimePoint::setConsideredSpatialEntityType().

#### 6.5.1.3 void multiscale::verification::subsetspecific::validateSubsetSpecificType ( const SubsetSpecificType & subsetSpecificType )

Check if the given subset specific type is valid.

**Parameters**

|                           |                                |
|---------------------------|--------------------------------|
| <i>subsetSpecificType</i> | The given subset specific type |
|---------------------------|--------------------------------|

Definition at line 10 of file SubsetSpecificAttribute.cpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, and multiscale::verification::NrOfSubsetSpecificTypeEntries.

Referenced by multiscale::verification::TimePoint::addSpatialEntity(), computeSubsetSpecificTypeIndex(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), multiscale::verification::TimePoint::removeSpatialEntity(), and multiscale::verification::TimePoint::setConsideredSpatialEntityType().

#### 6.5.1.4 void multiscale::verification::subsetspecific::validateSubsetSpecificTypeIndex ( const std::size\_t & subsetSpecificTypeIndex )

Check if the given subset specific type index is valid.

**Parameters**

|                                |                                      |
|--------------------------------|--------------------------------------|
| <i>subsetSpecificTypeIndex</i> | The given subset specific type index |
|--------------------------------|--------------------------------------|

Definition at line 17 of file SubsetSpecificAttribute.cpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, and multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES.

Referenced by computeSubsetSpecificType().

## 6.6 multiscale::video Namespace Reference

### Classes

- class [AnnularSector](#)  
*An annular sector is the basic element in the considered circular geometry.*
- class [CartesianToConcentrationsConverter](#)  
*Scale the values of the rectangular geometry grid cells.*
- class [CartesianToPolarConverter](#)  
*Converter from the rectangular geometry grid cells to annular sectors.*
- class [PolarCsvToInputFilesConverter](#)  
*Csv file to input file converter considering polar coordinates.*
- class [PolarGnuplotScriptGenerator](#)  
*Gnuplot script generator from the provided annular sectors.*
- class [RectangularCsvToInputFilesConverter](#)  
*Csv file to input file converter considering cartesian coordinates.*

- class [RectangularEntityCsvToInputFilesConverter](#)  
*Csv entity file to input file converter considering cartesian coordinates.*
- class [RectangularGnuplotScriptGenerator](#)  
*Gnuplot script generator from the provided concentrations considering a rectangular geometry.*

## 6.7 multiscaletest Namespace Reference

### Namespaces

- [verification](#)

### Classes

- class [ApproximateBayesianModelCheckerTest](#)  
*Class for testing the approximate Bayesian model checker.*
- class [ApproximateProbabilisticModelCheckerTest](#)  
*Class for testing the approximate probabilistic model checker.*
- class [BayesianModelCheckerTest](#)  
*Class for testing the Bayesian model checker.*
- class [CompleteTraceTest](#)  
*Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.*
- class [EmptyTraceTest](#)  
*Class for testing evaluation of empty traces.*
- class [MinEnclosingTriangleFinderTest](#)  
*Class for testing the minimum enclosing triangle algorithm.*
- class [ModelCheckerTest](#)  
*Class for testing model checkers.*
- class [MultiscaleTest](#)
- class [NumericStateVariableTraceTest](#)  
*Class for testing evaluation of numeric state variable-only traces.*
- class [ProbabilisticBlackBoxModelCheckerTest](#)  
*Class for testing the probabilistic black-box model checker.*
- class [SpatialEntitiesTraceTest](#)  
*Class for testing evaluation of spatial entities-only traces.*
- class [StatisticalModelCheckerTest](#)  
*Class for testing the statistical model checker.*
- class [TraceEvaluationTest](#)  
*Class for testing evaluation of traces.*

## 6.8 multiscaletest::verification Namespace Reference

### Functions

- bool [parseInputString](#) (const std::string &inputString)  
*Parse the input string and return the result of the parsing.*

#### 6.8.1 Function Documentation

##### 6.8.1.1 bool multiscaletest::verification::parseInputString ( const std::string & *inputString* )

Parse the input string and return the result of the parsing.

**Parameters**

|                    |                  |
|--------------------|------------------|
| <i>inputString</i> | The input string |
|--------------------|------------------|

Definition at line 27 of file InputStringParser.hpp.

References multiscale::verification::Parser::parse().

Referenced by TEST().

## Chapter 7

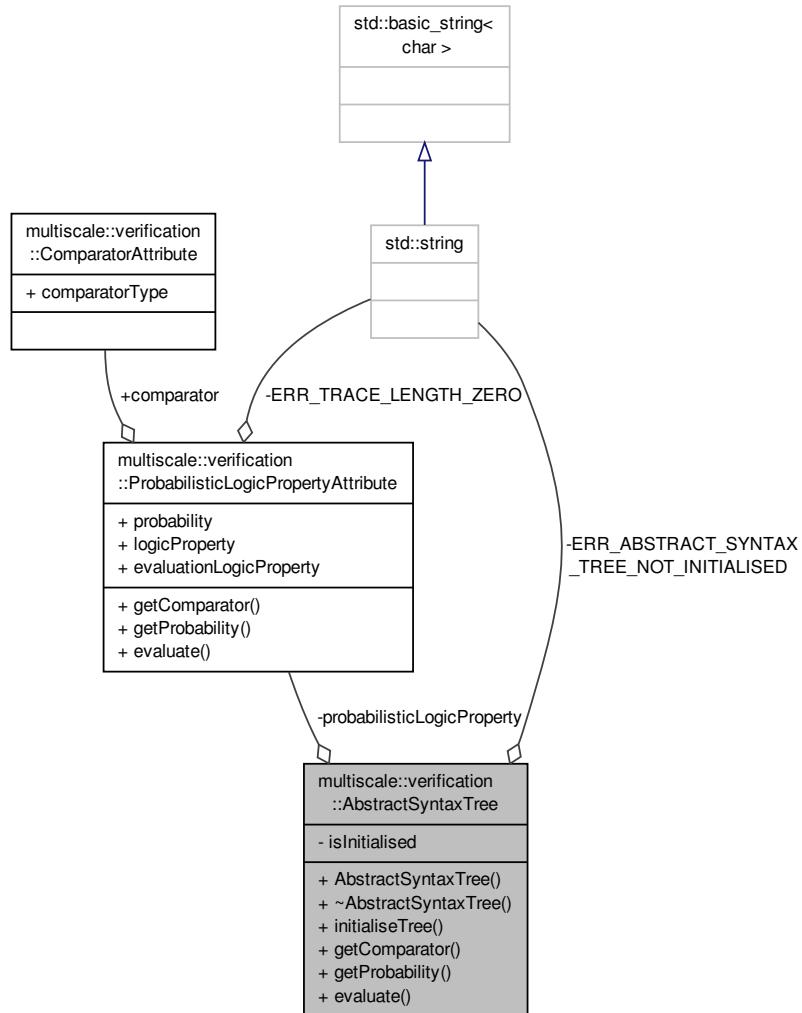
# Class Documentation

### 7.1 multiscale::verification::AbstractSyntaxTree Class Reference

Class used for representing an abstract syntax tree.

```
#include <AbstractSyntaxTree.hpp>
```

Collaboration diagram for multiscale::verification::AbstractSyntaxTree:



## Public Member Functions

- [AbstractSyntaxTree \(\)](#)
- [~AbstractSyntaxTree \(\)](#)
- void [initialiseTree \(const ProbabilisticLogicPropertyAttribute &probabilisticLogicPropertyAttribute\)](#)

*Initialise the abstract syntax tree using the given probabilistic logic property attribute.*
- [ComparatorType getComparator \(\)](#)

*Get the type of the comparator used in the probabilistic logical query.*
- double [getProbability \(\)](#)

*Get the value of the probability used in the probabilistic logical query.*
- bool [evaluate \(const SpatialTemporalTrace &spatialTemporalTrace, const TypeSemanticsTable &typeSemanticsTable\)](#)

*Evaluate the abstract syntax tree considering the given trace and type semantics table.*

## Private Attributes

- bool `isInitialised`
- `ProbabilisticLogicPropertyAttribute probabilisticLogicProperty`

## Static Private Attributes

- static const std::string `ERR_ABSTRACT_SYNTAX_TREE_NOT_INITIALISED` = "The abstract syntax tree was not initialised before evaluation. Call the method `initialiseTree(...)` before calling the method `evaluate(...)`."

### 7.1.1 Detailed Description

Class used for representing an abstract syntax tree.

Definition at line 14 of file AbstractSyntaxTree.hpp.

### 7.1.2 Constructor & Destructor Documentation

#### 7.1.2.1 AbstractSyntaxTree::AbstractSyntaxTree( )

Definition at line 7 of file AbstractSyntaxTree.cpp.

References `isInitialised`.

#### 7.1.2.2 AbstractSyntaxTree::~AbstractSyntaxTree( )

Definition at line 11 of file AbstractSyntaxTree.cpp.

### 7.1.3 Member Function Documentation

#### 7.1.3.1 bool AbstractSyntaxTree::evaluate ( const SpatialTemporalTrace & *spatialTemporalTrace*, const TypeSemanticsTable & *typeSemanticsTable* )

Evaluate the abstract syntax tree considering the given trace and type semantics table.

##### Parameters

|                                   |                                     |
|-----------------------------------|-------------------------------------|
| <code>spatialTemporalTrace</code> | The given spatial temporal trace    |
| <code>typeSemanticsTable</code>   | The considered type semantics table |

Definition at line 26 of file AbstractSyntaxTree.cpp.

References `ERR_ABSTRACT_SYNTAX_TREE_NOT_INITIALISED`, `multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluate()`, `isInitialised`, `MS_throw`, and `probabilisticLogicProperty`.

Referenced by `analysePatterns()`, `multiscale::verification::ModelChecker::evaluate()`, `main()`, and `multiscaletest::TraceEvaluationTest::RunTest()`.

#### 7.1.3.2 ComparatorType AbstractSyntaxTree::getComparator( )

Get the type of the comparator used in the probabilistic logical query.

Definition at line 18 of file AbstractSyntaxTree.cpp.

References multiscale::verification::ProbabilisticLogicPropertyAttribute::getComparator(), and probabilisticLogicProperty.

Referenced by multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator().

### 7.1.3.3 double AbstractSyntaxTree::getProbability( )

Get the value of the probability used in the probabilistic logical query.

Definition at line 22 of file AbstractSyntaxTree.cpp.

References multiscale::verification::ProbabilisticLogicPropertyAttribute::getProbability(), and probabilisticLogicProperty.

Referenced by multiscale::verification::ApproximateProbabilisticModelChecker::initialise(), multiscale::verification::StatisticalModelChecker::initialise(), multiscale::verification::BayesianModelChecker::initialise(), multiscale::verification::ApproximateBayesianModelChecker::initialise(), multiscale::verification::ModelChecker::updateHypothesesPValuesForGreaterThan(), and multiscale::verification::ModelChecker::updateHypothesesPValuesForLessThan().

### 7.1.3.4 void AbstractSyntaxTree::initialiseTree( const ProbabilisticLogicPropertyAttribute & probabilisticLogicPropertyAttribute )

Initialise the abstract syntax tree using the given probabilistic logic property attribute.

#### Parameters

|                                            |                                            |
|--------------------------------------------|--------------------------------------------|
| <i>probabilisticLogicPropertyAttribute</i> | The probabilistic logic property attribute |
|--------------------------------------------|--------------------------------------------|

Definition at line 13 of file AbstractSyntaxTree.cpp.

References isInitialised, and probabilisticLogicProperty.

Referenced by multiscale::verification::Parser::parseLogicalQuery().

## 7.1.4 Member Data Documentation

### 7.1.4.1 const std::string AbstractSyntaxTree::ERR\_ABSTRACT\_SYNTAX\_TREE\_NOT\_INITIALISED = "The abstract syntax tree was not initialised before evaluation. Call the method initialiseTree(...) before calling the method evaluate(...)." [static], [private]

Definition at line 53 of file AbstractSyntaxTree.hpp.

Referenced by evaluate().

### 7.1.4.2 bool multiscale::verification::AbstractSyntaxTree::isInitialised [private]

Flag for indicating if the abstract syntax tree was initialised

Definition at line 19 of file AbstractSyntaxTree.hpp.

Referenced by AbstractSyntaxTree(), evaluate(), and initialiseTree().

### 7.1.4.3 ProbabilisticLogicPropertyAttribute multiscale::verification::AbstractSyntaxTree::probabilisticLogicProperty [private]

The abstract syntax tree represented using a probabilistic logic property attribute

Definition at line 22 of file AbstractSyntaxTree.hpp.

Referenced by evaluate(), getComparator(), getProbability(), and initialiseTree().

The documentation for this class was generated from the following files:

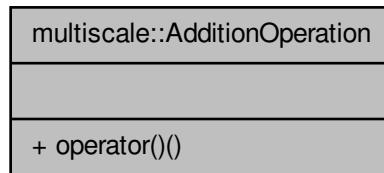
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[AbstractSyntaxTree.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[AbstractSyntaxTree.cpp](#)

## 7.2 multiscale::AdditionOperation Class Reference

Functor representing an addition operation.

```
#include <Numeric.hpp>
```

Collaboration diagram for multiscale::AdditionOperation:



### Public Member Functions

- template<typename Operand >  
Operand [operator\(\)](#) (Operand operand1, Operand operand2) const  
*Add the two operands.*

#### 7.2.1 Detailed Description

Functor representing an addition operation.

Definition at line 15 of file Numeric.hpp.

#### 7.2.2 Member Function Documentation

**7.2.2.1** template<typename Operand > Operand multiscale::AdditionOperation::operator() ( Operand *operand1*, Operand *operand2* ) const [inline]

Add the two operands.

##### Parameters

---

|                 |                    |
|-----------------|--------------------|
| <i>operand1</i> | The first operand  |
| <i>operand2</i> | The second operand |

Definition at line 25 of file Numeric.hpp.

The documentation for this class was generated from the following file:

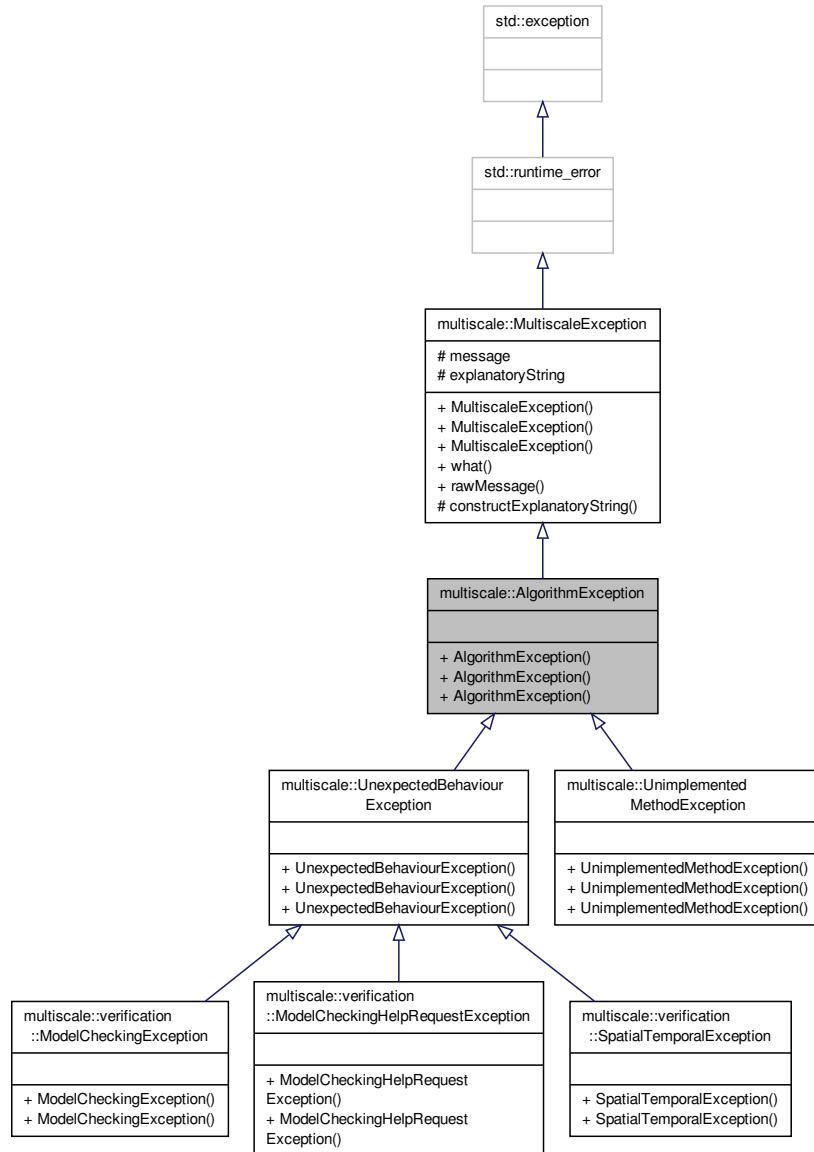
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Numeric.hpp

## 7.3 multiscale::AlgorithmException Class Reference

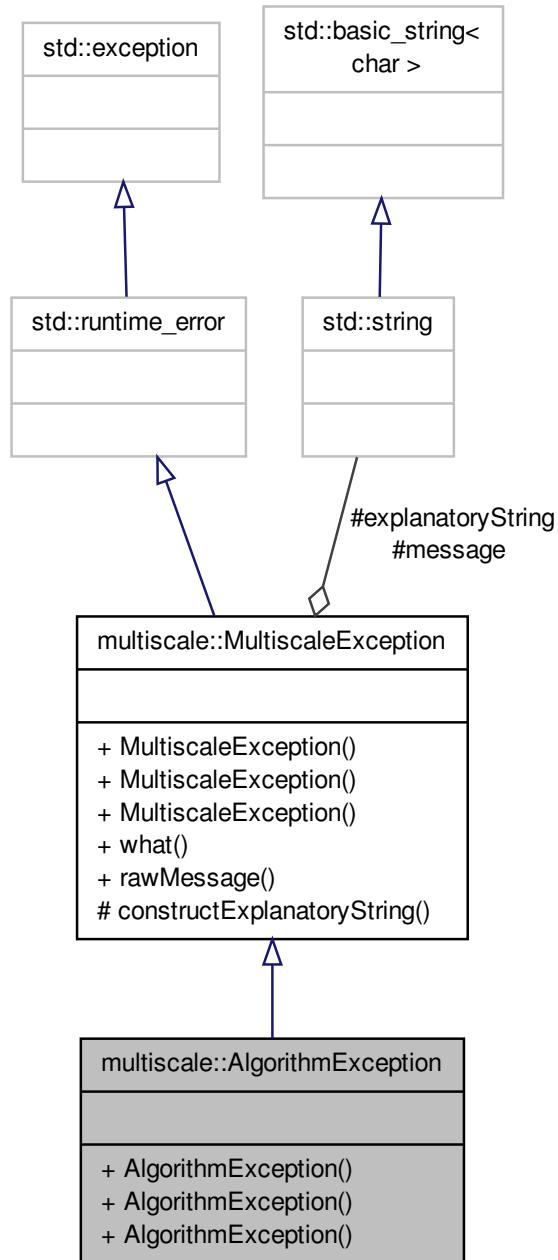
Class for representing algorithm exceptions.

```
#include <AlgorithmException.hpp>
```

Inheritance diagram for multiscale::AlgorithmException:



Collaboration diagram for multiscale::AlgorithmException:



## Public Member Functions

- [AlgorithmException \(\)](#)
- [AlgorithmException \(const std::string &file, int line, const std::string &msg\)](#)
- [AlgorithmException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.3.1 Detailed Description

Class for representing algorithm exceptions.

Definition at line 12 of file AlgorithmException.hpp.

### 7.3.2 Constructor & Destructor Documentation

#### 7.3.2.1 multiscale::AlgorithmException::AlgorithmException( ) [inline]

Definition at line 16 of file AlgorithmException.hpp.

#### 7.3.2.2 multiscale::AlgorithmException::AlgorithmException( const std::string & file, int line, const std::string & msg ) [inline], [explicit]

Definition at line 18 of file AlgorithmException.hpp.

#### 7.3.2.3 multiscale::AlgorithmException::AlgorithmException( const std::string & file, int line, const char \* msg ) [inline], [explicit]

Definition at line 23 of file AlgorithmException.hpp.

The documentation for this class was generated from the following file:

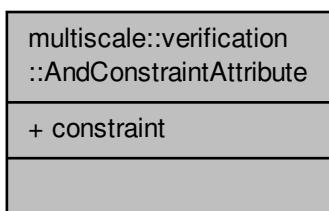
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/AlgorithmException.hpp

## 7.4 multiscale::verification::AndConstraintAttribute Class Reference

Class for representing an "and" constraint attribute.

```
#include <AndConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::AndConstraintAttribute:



### Public Attributes

- [ConstraintAttributeType constraint](#)

### 7.4.1 Detailed Description

Class for representing an "and" constraint attribute.

Definition at line 14 of file AndConstraintAttribute.hpp.

### 7.4.2 Member Data Documentation

#### 7.4.2.1 ConstraintAttributeType multiscale::verification::AndConstraintAttribute::constraint

The constraint following the "and" operator

Definition at line 18 of file AndConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

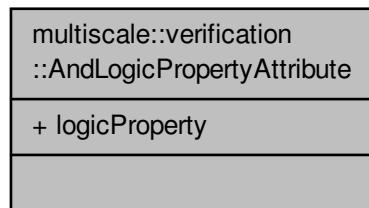
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndConstraintAttribute.hpp

## 7.5 multiscale::verification::AndLogicPropertyAttribute Class Reference

Class for representing an "and" logic property attribute.

```
#include <AndLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::AndLogicPropertyAttribute:



### Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.5.1 Detailed Description

Class for representing an "and" logic property attribute.

Definition at line 14 of file AndLogicPropertyAttribute.hpp.

### 7.5.2 Member Data Documentation

### 7.5.2.1 LogicPropertyAttributeType multiscale::verification::AndLogicPropertyAttribute::logicProperty

The logical property following the "and" operator

Definition at line 18 of file AndLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::operator()().

The documentation for this class was generated from the following file:

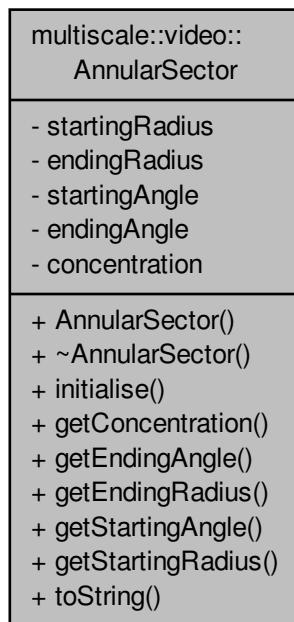
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndLogicPropertyAttribute.hpp

## 7.6 multiscale::video::AnnularSector Class Reference

An annular sector is the basic element in the considered circular geometry.

```
#include <AnnularSector.hpp>
```

Collaboration diagram for multiscale::video::AnnularSector:



### Public Member Functions

- [AnnularSector \(\)](#)
- [~AnnularSector \(\)](#)
- void [initialise](#) (double [startingRadius](#), double [endingRadius](#), double [startingAngle](#), double [endingAngle](#), double [concentration](#))

*Initialise the members of the class.*

- double [getConcentration \(\) const](#)

*Get the value of the concentration.*

- `double getEndingAngle () const`  
*Get the value of the ending angle.*
- `double getEndingRadius () const`  
*Get the value of the ending radius.*
- `double getStartingAngle () const`  
*Get the value of the starting angle.*
- `double getStartingRadius () const`  
*Get the value of the starting radius.*
- `std::string toString ()`  
*Get the std::string representation of the annular sector.*

## Private Attributes

- `double startingRadius`
- `double endingRadius`
- `double startingAngle`
- `double endingAngle`
- `double concentration`

### 7.6.1 Detailed Description

An annular sector is the basic element in the considered circular geometry.

More information about annuli and sectors of annuli can be found online (e.g. Wikipedia).

Definition at line 15 of file AnnularSector.hpp.

### 7.6.2 Constructor & Destructor Documentation

#### 7.6.2.1 AnnularSector::AnnularSector( )

Definition at line 11 of file AnnularSector.cpp.

References concentration, endingAngle, endingRadius, startingAngle, and startingRadius.

#### 7.6.2.2 AnnularSector::~AnnularSector( )

Definition at line 19 of file AnnularSector.cpp.

### 7.6.3 Member Function Documentation

#### 7.6.3.1 double AnnularSector::getConcentration( ) const

Get the value of the concentration.

Definition at line 30 of file AnnularSector.cpp.

References concentration.

#### 7.6.3.2 double AnnularSector::getEndingAngle( ) const

Get the value of the ending angle.

Definition at line 34 of file AnnularSector.cpp.

References endingAngle.

### 7.6.3.3 double AnnularSector::getEndingRadius ( ) const

Get the value of the ending radius.

Definition at line 38 of file AnnularSector.cpp.

References endingRadius.

### 7.6.3.4 double AnnularSector::getStartingAngle ( ) const

Get the value of the starting angle.

Definition at line 42 of file AnnularSector.cpp.

References startingAngle.

### 7.6.3.5 double AnnularSector::getStartingRadius ( ) const

Get the value of the starting radius.

Definition at line 46 of file AnnularSector.cpp.

References startingRadius.

### 7.6.3.6 void AnnularSector::initialise ( double *startingRadius*, double *endingRadius*, double *startingAngle*, double *endingAngle*, double *concentration* )

Initialise the members of the class.

#### Parameters

|                       |                 |
|-----------------------|-----------------|
| <i>startingRadius</i> | Starting radius |
| <i>endingRadius</i>   | Ending radius   |
| <i>startingAngle</i>  | Starting angle  |
| <i>endingAngle</i>    | Ending angle    |
| <i>concentration</i>  | Concentration   |

Definition at line 21 of file AnnularSector.cpp.

References concentration, endingAngle, endingRadius, startingAngle, and startingRadius.

### 7.6.3.7 std::string AnnularSector::toString ( )

Get the std::string representation of the annular sector.

Definition at line 50 of file AnnularSector.cpp.

References concentration, endingAngle, endingRadius, SEPARATOR, startingAngle, and startingRadius.

## 7.6.4 Member Data Documentation

### 7.6.4.1 double multiscale::video::AnnularSector::concentration [private]

Definition at line 23 of file AnnularSector.hpp.

Referenced by AnnularSector(), getConcentration(), initialise(), and toString().

### 7.6.4.2 double multiscale::video::AnnularSector::endingAngle [private]

Definition at line 22 of file AnnularSector.hpp.

Referenced by AnnularSector(), getEndingAngle(), initialise(), and toString().

#### 7.6.4.3 double multiscale::video::AnnularSector::endingRadius [private]

Definition at line 20 of file AnnularSector.hpp.

Referenced by AnnularSector(), getEndingRadius(), initialise(), and toString().

#### 7.6.4.4 double multiscale::video::AnnularSector::startingAngle [private]

Definition at line 21 of file AnnularSector.hpp.

Referenced by AnnularSector(), getStartingAngle(), initialise(), and toString().

#### 7.6.4.5 double multiscale::video::AnnularSector::startingRadius [private]

Definition at line 19 of file AnnularSector.hpp.

Referenced by AnnularSector(), getStartingRadius(), initialise(), and toString().

The documentation for this class was generated from the following files:

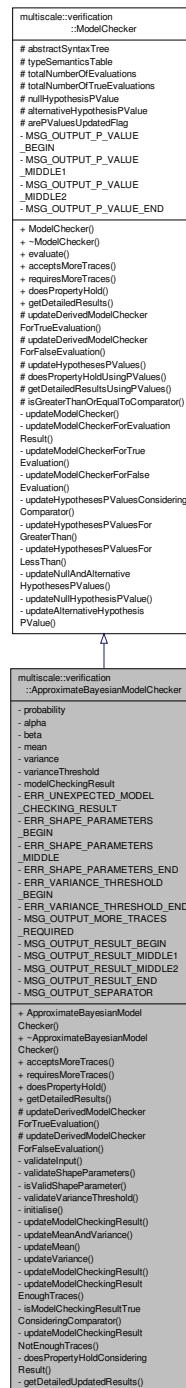
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/[AnnularSector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/[AnnularSector.cpp](#)

## 7.7 multiscale::verification::ApproximateBayesianModelChecker Class Reference

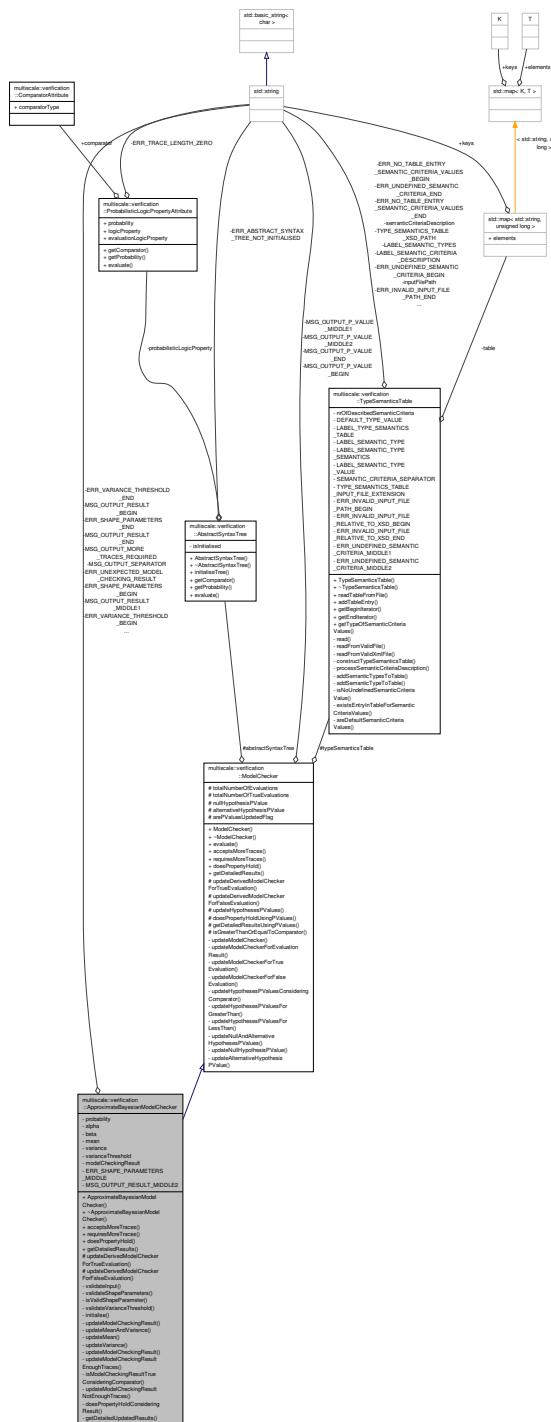
Class used to run approximate Bayesian model checking tasks.

```
#include <ApproximateBayesianModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::ApproximateBayesianModelChecker:



## Collaboration diagram for multiscale::verification::ApproximateBayesianModelChecker:



## Public Member Functions

- `ApproximateBayesianModelChecker` (const `AbstractSyntaxTree` &`abstractSyntaxTree`, const `TypeSemanticsTable` &`typeSemanticsTable`, double `alpha`, double `beta`, double `varianceThreshold`)
  - `~ApproximateBayesianModelChecker` ()
  - bool `acceptsMoreTraces` () override

*Check if more traces are accepted for evaluating the logic property.*

- bool `requiresMoreTraces ()` override  
*Check if more traces are required for evaluating the logic property.*
- bool `doesPropertyHold ()` override  
*Check if the given property holds.*
- std::string `getDetailedResults ()` override  
*Get the detailed description of the results.*

## Protected Member Functions

- void `updateDerivedModelCheckerForTrueEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- void `updateDerivedModelCheckerForFalseEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*

## Private Member Functions

- void `validateInput (double alpha, double beta, double varianceThreshold)`  
*Validate the input parameters  $\alpha$ ,  $\beta$  and the variance threshold.*
- void `validateShapeParameters (double alpha, double beta)`  
*Validate the shape parameters  $\alpha$  and  $\beta$ .*
- bool `isValidShapeParameter (double shapeParameter)`  
*Check if the given shape parameter value is valid.*
- void `validateVarianceThreshold (double varianceThreshold)`  
*Validate the variance threshold.*
- void `initialise ()`  
*Initialisation of some of the class members.*
- void `updateModelCheckingResult ()`  
*Update the result of the model checking task.*
- void `updateMeanAndVariance ()`  
*Update the value of the mean and variance estimates.*
- void `updateMean ()`  
*Update the value of the mean estimate.*
- void `updateVariance ()`  
*Update the value of the variance estimate.*
- void `updateModelCheckingResult (double variance)`  
*Update the result of the model checking task considering the given variance value.*
- void `updateModelCheckingResultEnoughTraces (double variance)`  
*Update the result of the model checking task considering that enough traces have been provided.*
- bool `isModelCheckingResultTrueConsideringComparator (double variance)`  
*Check if the result of the model checking task is true considering the probabilistic comparator (i.e.  $\leq$ ,  $\geq$ )*
- void `updateModelCheckingResultNotEnoughTraces ()`  
*Update the result of the model checking task considering that not enough traces have been provided.*
- bool `doesPropertyHoldConsideringResult ()`  
*Check if the given property holds considering the obtained model checking result.*
- std::string `getDetailedUpdatedResults ()`  
*Get the detailed description of the updated results.*

## Private Attributes

- double probability
- double alpha
- double beta
- double mean
- double variance
- double varianceThreshold
- ApproximateBayesianModelCheckingResult modelCheckingResult

## Static Private Attributes

- static const std::string `ERR_UNEXPECTED_MODEL_CHECKING_RESULT` = "An invalid Approximate Bayesian model checking result was obtained. Please check source code."
- static const std::string `ERR_SHAPE_PARAMETERS_BEGIN` = "The provided Beta distribution shape parameters `alpha` and `beta` ("
- static const std::string `ERR_SHAPE_PARAMETERS_MIDDLE` = ", "
- static const std::string `ERR_SHAPE_PARAMETERS_END` = ") should be greater than zero. Please change."
- static const std::string `ERR_VARIANCE_THRESHOLD_BEGIN` = "The provided `variance` threshold ("
- static const std::string `ERR_VARIANCE_THRESHOLD_END` = ") should be greater than zero. Please change."
- static const std::string `MSG_OUTPUT_MORE_TRACES_REQUIRED` = "More traces are required to provide a true/false answer assuming the given Beta distribution shape parameters and `variance` threshold value. Probabilistic black-box model checking was used instead to provide an answer."
- static const std::string `MSG_OUTPUT_RESULT_BEGIN` = "The provided answer is given for the Beta distribution shape parameters `alpha` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE1` = " and `beta` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE2` = ", and `variance` threshold value = "
- static const std::string `MSG_OUTPUT_RESULT_END` = ""
- static const std::string `MSG_OUTPUT_SEPARATOR` = " "

## Additional Inherited Members

### 7.7.1 Detailed Description

Class used to run approximate Bayesian model checking tasks.

The implementation of this class is (partially) based on the algorithms described in the following paper:

C. Langmead, 'Generalized Queries and Bayesian Statistical Model Checking in Dynamic Bayesian Networks : Application to Personalized Medicine', Computer Science Department, Aug. 2009.

In our implementation the variables in the original paper (right hand side of the assignments) have been given the following new names (left hand side of assignments):

`probability` =  $p$

`alpha` =  $\alpha$

`beta` =  $\beta$

`mean` =  $\hat{p}$

`variance` =  $\hat{v}$

`varianceThreshold` =  $T$

`totalNumberOfEvaluations` =  $n$

`totalNumberOfTrueEvaluations` =  $k$

Definition at line 51 of file ApproximateBayesianModelChecker.hpp.

## 7.7.2 Constructor & Destructor Documentation

7.7.2.1 `ApproximateBayesianModelChecker::ApproximateBayesianModelChecker ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable, double alpha, double beta, double varianceThreshold )`

Definition at line 11 of file ApproximateBayesianModelChecker.cpp.

References alpha, beta, initialise(), validateInput(), and varianceThreshold.

7.7.2.2 `ApproximateBayesianModelChecker::~ApproximateBayesianModelChecker ( )`

Definition at line 27 of file ApproximateBayesianModelChecker.cpp.

## 7.7.3 Member Function Documentation

7.7.3.1 `bool ApproximateBayesianModelChecker::acceptsMoreTraces ( ) [override], [virtual]`

Check if more traces are accepted for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 29 of file ApproximateBayesianModelChecker.cpp.

References modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, and updateModelCheckingResult().

Referenced by requiresMoreTraces().

7.7.3.2 `bool ApproximateBayesianModelChecker::doesPropertyHold ( ) [override], [virtual]`

Check if the given property holds.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 39 of file ApproximateBayesianModelChecker.cpp.

References doesPropertyHoldConsideringResult(), and updateModelCheckingResult().

7.7.3.3 `bool ApproximateBayesianModelChecker::doesPropertyHoldConsideringResult ( ) [private]`

Check if the given property holds considering the obtained model checking result.

Definition at line 147 of file ApproximateBayesianModelChecker.cpp.

References multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues(), ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT, multiscale::verification::FALSE, modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MS\_throw, and multiscale::verification::TRUE.

Referenced by doesPropertyHold().

7.7.3.4 `std::string ApproximateBayesianModelChecker::getDetailedResults ( ) [override], [virtual]`

Get the detailed description of the results.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 45 of file ApproximateBayesianModelChecker.cpp.

References getDetailedUpdatedResults(), and updateModelCheckingResult().

## 7.7.3.5 std::string ApproximateBayesianModelChecker::getDetailedUpdatedResults( ) [private]

Get the detailed description of the updated results.

Definition at line 166 of file ApproximateBayesianModelChecker.cpp.

References alpha, beta, multiscale::verification::ModelChecker::getDetailedResultsUsingPValues(), modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_RESULT\_BEGIN, MSG\_OUTPUT\_RESULT\_END, MSG\_OUTPUT\_RESULT\_MIDDLE1, MSG\_OUTPUT\_RESULT\_MIDDLE2, MSG\_OUTPUT\_SEPARATOR, multiscale::StringManipulator::toString(), and varianceThreshold.

Referenced by getDetailedResults().

## 7.7.3.6 void ApproximateBayesianModelChecker::initialise( ) [private]

Initialisation of some of the class members.

Definition at line 86 of file ApproximateBayesianModelChecker.cpp.

References multiscale::verification::ModelChecker::abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), mean, probability, and variance.

Referenced by ApproximateBayesianModelChecker().

## 7.7.3.7 bool ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator( double variance ) [private]

Check if the result of the model checking task is true considering the probabilistic comparator (i.e.  $\leq$ ,  $\geq$ )

For queries of type : a)  $P \geq \theta[\phi]$  the result is ( $mean \geq \theta$ ) b)  $P \leq \theta[\phi]$  the result is ( $mean \leq \theta$ )

**Parameters**

|                 |                          |
|-----------------|--------------------------|
| <i>variance</i> | The given variance value |
|-----------------|--------------------------|

Definition at line 135 of file ApproximateBayesianModelChecker.cpp.

References multiscale::Numeric::greaterOrEqual(), multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator(), multiscale::Numeric::lessOrEqual(), mean, and probability.

Referenced by updateModelCheckingResultEnoughTraces().

7.7.3.8 bool ApproximateBayesianModelChecker::isValidShapeParameter( double *shapeParameter* ) [private]

Check if the given shape parameter value is valid.

The shape parameter values should be greater than zero

**Parameters**

|                       |                           |
|-----------------------|---------------------------|
| <i>shapeParameter</i> | The given shape parameter |
|-----------------------|---------------------------|

Definition at line 71 of file ApproximateBayesianModelChecker.cpp.

Referenced by validateShapeParameters().

## 7.7.3.9 bool ApproximateBayesianModelChecker::requiresMoreTraces( ) [override], [virtual]

Check if more traces are required for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 35 of file ApproximateBayesianModelChecker.cpp.

References `acceptsMoreTraces()`.

**7.7.3.10 void ApproximateBayesianModelChecker::updateDerivedModelCheckerForFalseEvaluation( ) [override], [protected], [virtual]**

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 53 of file `ApproximateBayesianModelChecker.cpp`.

**7.7.3.11 void ApproximateBayesianModelChecker::updateDerivedModelCheckerForTrueEvaluation( ) [override], [protected], [virtual]**

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 51 of file `ApproximateBayesianModelChecker.cpp`.

**7.7.3.12 void ApproximateBayesianModelChecker::updateMean( ) [private]**

Update the value of the mean estimate.

Definition at line 103 of file `ApproximateBayesianModelChecker.cpp`.

References `alpha`, `beta`, `mean`, [multiscale::verification::ModelChecker::totalNumberOfEvaluations](#), and [multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations](#).

Referenced by `updateMeanAndVariance()`.

**7.7.3.13 void ApproximateBayesianModelChecker::updateMeanAndVariance( ) [private]**

Update the value of the mean and variance estimates.

Definition at line 98 of file `ApproximateBayesianModelChecker.cpp`.

References `updateMean()`, and `updateVariance()`.

Referenced by `updateModelCheckingResult()`.

**7.7.3.14 void ApproximateBayesianModelChecker::updateModelCheckingResult( ) [private]**

Update the result of the model checking task.

Definition at line 93 of file `ApproximateBayesianModelChecker.cpp`.

References `updateMeanAndVariance()`, and `variance`.

Referenced by `acceptsMoreTraces()`, `doesPropertyHold()`, and `getDetailedResults()`.

**7.7.3.15 void ApproximateBayesianModelChecker::updateModelCheckingResult( double variance ) [private]**

Update the result of the model checking task considering the given variance value.

## Parameters

|                 |                          |
|-----------------|--------------------------|
| <i>variance</i> | The given variance value |
|-----------------|--------------------------|

Definition at line 119 of file ApproximateBayesianModelChecker.cpp.

References updateModelCheckingResultEnoughTraces(), updateModelCheckingResultNotEnoughTraces(), and varianceThreshold.

#### 7.7.3.16 void ApproximateBayesianModelChecker::updateModelCheckingResultEnoughTraces ( double *variance* ) [private]

Update the result of the model checking task considering that enough traces have been provided.

## Parameters

|                 |                          |
|-----------------|--------------------------|
| <i>variance</i> | The given variance value |
|-----------------|--------------------------|

Definition at line 127 of file ApproximateBayesianModelChecker.cpp.

References multiscale::verification::FALSE, isModelCheckingResultTrueConsideringComparator(), model←CheckingResult, and multiscale::verification::TRUE.

Referenced by updateModelCheckingResult().

#### 7.7.3.17 void ApproximateBayesianModelChecker::updateModelCheckingResultNotEnoughTraces ( ) [private]

Update the result of the model checking task considering that not enough traces have been provided.

Definition at line 143 of file ApproximateBayesianModelChecker.cpp.

References modelCheckingResult, and multiscale::verification::MORE\_TRACES\_REQUIRED.

Referenced by updateModelCheckingResult().

#### 7.7.3.18 void ApproximateBayesianModelChecker::updateVariance ( ) [private]

Update the value of the variance estimate.

Definition at line 110 of file ApproximateBayesianModelChecker.cpp.

References alpha, beta, multiscale::verification::ModelChecker::totalNumberOfEvaluations, multiscale::verification←::ModelChecker::totalNumberOfTrueEvaluations, and variance.

Referenced by updateMeanAndVariance().

#### 7.7.3.19 void ApproximateBayesianModelChecker::validateInput ( double *alpha*, double *beta*, double *varianceThreshold* ) [private]

Validate the input parameters  $\alpha$ ,  $\beta$  and the variance threshold.

$\alpha$ ,  $\beta$  and variance threshold should be greater than zero

## Parameters

|                          |                                                        |
|--------------------------|--------------------------------------------------------|
| <i>alpha</i>             | The shape parameter $\alpha$ for the Beta distribution |
| <i>beta</i>              | The shape parameter $\beta$ for the Beta distribution  |
| <i>varianceThreshold</i> | The variance threshold                                 |

Definition at line 55 of file ApproximateBayesianModelChecker.cpp.

References validateShapeParameters(), and validateVarianceThreshold().

Referenced by ApproximateBayesianModelChecker().

### 7.7.3.20 void ApproximateBayesianModelChecker::validateShapeParameters ( double *alpha*, double *beta* ) [private]

Validate the shape parameters  $\alpha$  and  $\beta$ .

$\alpha$  and  $\beta$  should be greater than zero

#### Parameters

|              |                                                        |
|--------------|--------------------------------------------------------|
| <i>alpha</i> | The shape parameter $\alpha$ for the Beta distribution |
| <i>beta</i>  | The shape parameter $\beta$ for the Beta distribution  |

Definition at line 60 of file ApproximateBayesianModelChecker.cpp.

References ERR\_SHAPE\_PARAMETERS\_BEGIN, ERR\_SHAPE\_PARAMETERS\_END, ERR\_SHAPE\_PARAMETERS\_MIDDLE, isValidShapeParameter(), MS\_throw, and multiscale::StringManipulator::toString().

Referenced by validateInput().

### 7.7.3.21 void ApproximateBayesianModelChecker::validateVarianceThreshold ( double *varianceThreshold* ) [private]

Validate the variance threshold.

The variance threshold should be greater than 0

#### Parameters

|                          |                        |
|--------------------------|------------------------|
| <i>varianceThreshold</i> | The variance threshold |
|--------------------------|------------------------|

Definition at line 75 of file ApproximateBayesianModelChecker.cpp.

References ERR\_VARIANCE\_THRESHOLD\_BEGIN, ERR\_VARIANCE\_THRESHOLD\_END, multiscale::Numeric::lessOrEqual(), MS\_throw, and multiscale::StringManipulator::toString().

Referenced by validateInput().

## 7.7.4 Member Data Documentation

### 7.7.4.1 double multiscale::verification::ApproximateBayesianModelChecker::alpha [private]

The shape parameter  $\alpha$  for the Beta distribution prior

Definition at line 58 of file ApproximateBayesianModelChecker.hpp.

Referenced by ApproximateBayesianModelChecker(), getDetailedUpdatedResults(), updateMean(), and updateVariance().

### 7.7.4.2 double multiscale::verification::ApproximateBayesianModelChecker::beta [private]

The shape parameter  $\beta$  for the Beta distribution prior

Definition at line 59 of file ApproximateBayesianModelChecker.hpp.

Referenced by ApproximateBayesianModelChecker(), getDetailedUpdatedResults(), updateMean(), and updateVariance().

### 7.7.4.3 const std::string ApproximateBayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_BEGIN = "The provided Beta distribution shape parameters alpha and beta (" [static], [private]

Definition at line 184 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateShapeParameters().

7.7.4.4 const std::string ApproximateBayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_END = ") should be greater than zero. Please change." [static], [private]

Definition at line 186 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateShapeParameters().

7.7.4.5 const std::string ApproximateBayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_MIDDLE = ", " [static], [private]

Definition at line 185 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateShapeParameters().

7.7.4.6 const std::string ApproximateBayesianModelChecker::ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT = "An invalid ApproximateBayesian model checking result was obtained. Please check source code." [static], [private]

Definition at line 182 of file ApproximateBayesianModelChecker.hpp.

Referenced by doesPropertyHoldConsideringResult().

7.7.4.7 const std::string ApproximateBayesianModelChecker::ERR\_VARIANCE\_THRESHOLD\_BEGIN = "The provided variance threshold (" [static], [private]

Definition at line 188 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateVarianceThreshold().

7.7.4.8 const std::string ApproximateBayesianModelChecker::ERR\_VARIANCE\_THRESHOLD\_END = ") should be greater than zero. Please change." [static], [private]

Definition at line 189 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateVarianceThreshold().

7.7.4.9 double multiscale::verification::ApproximateBayesianModelChecker::mean [private]

The value of the mean

Definition at line 61 of file ApproximateBayesianModelChecker.hpp.

Referenced by initialise(), isModelCheckingResultTrueConsideringComparator(), and updateMean().

7.7.4.10 ApproximateBayesianModelCheckingResult multiscale::verification::ApproximateBayesianModelChecker::modelCheckingResult [private]

The result of the model checking task

Definition at line 66 of file ApproximateBayesianModelChecker.hpp.

Referenced by acceptsMoreTraces(), doesPropertyHoldConsideringResult(), getDetailedUpdatedResults(), updateModelCheckingResultEnoughTraces(), and updateModelCheckingResultNotEnoughTraces().

7.7.4.11 const std::string ApproximateBayesianModelChecker::MSG\_OUTPUT\_MORE\_TRACES\_REQUIRED = "More traces are required to provide a true/false answer assuming the given Beta distribution shape parameters and variance threshold value. Probabilistic black-box model checking was used instead to provide an answer." [static], [private]

Definition at line 191 of file ApproximateBayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.12 `const std::string ApproximateBayesianModelChecker::MSG_OUTPUT_RESULT_BEGIN = "The provided answer is given for the Beta distribution shape parameters alpha = "` [static], [private]

Definition at line 193 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.13 `const std::string ApproximateBayesianModelChecker::MSG_OUTPUT_RESULT_END = ""` [static], [private]

Definition at line 196 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.14 `const std::string ApproximateBayesianModelChecker::MSG_OUTPUT_RESULT_MIDDLE1 = " and beta = "` [static], [private]

Definition at line 194 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.15 `const std::string ApproximateBayesianModelChecker::MSG_OUTPUT_RESULT_MIDDLE2 = ", and variance threshold value = "` [static], [private]

Definition at line 195 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.16 `const std::string ApproximateBayesianModelChecker::MSG_OUTPUT_SEPARATOR = " "` [static], [private]

Definition at line 198 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `getDetailedUpdatedResults()`.

7.7.4.17 `double multiscale::verification::ApproximateBayesianModelChecker::probability` [private]

The probability specified by the user for the logic property to be evaluated

Definition at line 55 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `initialise()`, and `isModelCheckingResultTrueConsideringComparator()`.

7.7.4.18 `double multiscale::verification::ApproximateBayesianModelChecker::variance` [private]

The value of the variance

Definition at line 62 of file `ApproximateBayesianModelChecker.hpp`.

Referenced by `initialise()`, `updateModelCheckingResult()`, and `updateVariance()`.

7.7.4.19 `double multiscale::verification::ApproximateBayesianModelChecker::varianceThreshold` [private]

The variance threshold

Definition at line 64 of file ApproximateBayesianModelChecker.hpp.

Referenced by ApproximateBayesianModelChecker(), getDetailedUpdatedResults(), and updateModelCheckingResult().

The documentation for this class was generated from the following files:

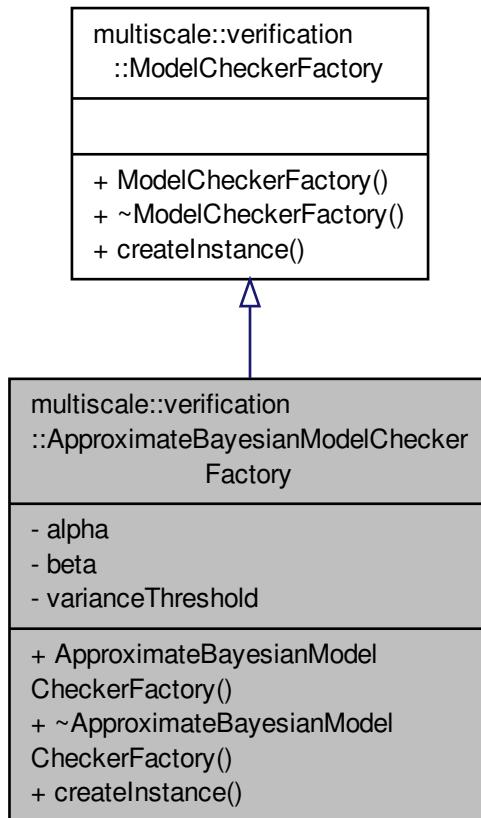
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelChecker.cpp

## 7.8 multiscale::verification::ApproximateBayesianModelCheckerFactory Class Reference

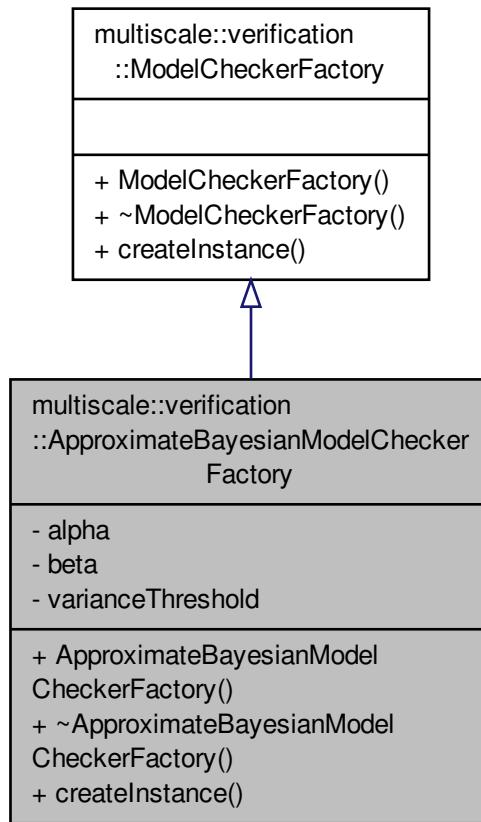
Class for creating [ApproximateBayesianModelChecker](#) instances.

```
#include <ApproximateBayesianModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::ApproximateBayesianModelCheckerFactory:



Collaboration diagram for multiscale::verification::ApproximateBayesianModelCheckerFactory:



## Public Member Functions

- [ApproximateBayesianModelCheckerFactory](#) (double `alpha`, double `beta`, double `varianceThreshold`)
- [~ApproximateBayesianModelCheckerFactory](#) ()
- std::shared\_ptr< ModelChecker > `createInstance` (const [AbstractSyntaxTree](#) &abstractSyntaxTree, const [TypeSemanticsTable](#) &typeSemanticsTable) override

*Create an instance of [ApproximateBayesianModelChecker](#).*

## Private Attributes

- double `alpha`
- double `beta`
- double `varianceThreshold`

### 7.8.1 Detailed Description

Class for creating [ApproximateBayesianModelChecker](#) instances.

Definition at line 12 of file `ApproximateBayesianModelChecker.hpp`.

## 7.8.2 Constructor & Destructor Documentation

7.8.2.1 `ApproximateBayesianModelCheckerFactory::ApproximateBayesianModelCheckerFactory ( double alpha, double beta, double varianceThreshold )`

Definition at line 7 of file ApproximateBayesianModelCheckerFactory.cpp.

7.8.2.2 `ApproximateBayesianModelCheckerFactory::~ApproximateBayesianModelCheckerFactory ( )`

Definition at line 12 of file ApproximateBayesianModelCheckerFactory.cpp.

## 7.8.3 Member Function Documentation

7.8.3.1 `std::shared_ptr< ModelChecker > ApproximateBayesianModelCheckerFactory::createInstance ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [override], [virtual]`

Create an instance of [ApproximateBayesianModelChecker](#).

### Parameters

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| <code>abstractSyntaxTree</code> | The abstract syntax tree representing the logic property to be checked                |
| <code>typeSemanticsTable</code> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implements [multiscale::verification::ModelCheckerFactory](#).

Definition at line 15 of file ApproximateBayesianModelCheckerFactory.cpp.

References alpha, beta, and varianceThreshold.

## 7.8.4 Member Data Documentation

7.8.4.1 `double multiscale::verification::ApproximateBayesianModelCheckerFactory::alpha [private]`

The shape parameter  $\alpha$  for the Beta distribution prior

Definition at line 16 of file ApproximateBayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

7.8.4.2 `double multiscale::verification::ApproximateBayesianModelCheckerFactory::beta [private]`

The shape parameter  $\beta$  for the Beta distribution prior

Definition at line 17 of file ApproximateBayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

7.8.4.3 `double multiscale::verification::ApproximateBayesianModelCheckerFactory::varianceThreshold [private]`

The variance threshold

Definition at line 19 of file ApproximateBayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

The documentation for this class was generated from the following files:

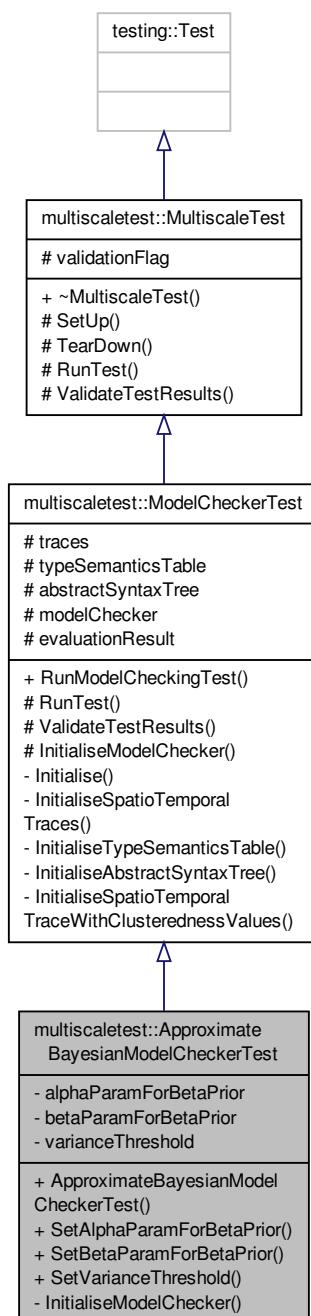
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ApproximateBayesianModelCheckerFactory.hpp](#)
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ApproximateBayesianModelCheckerFactory.cpp](#)

## 7.9 multiscaletest::ApproximateBayesianModelCheckerTest Class Reference

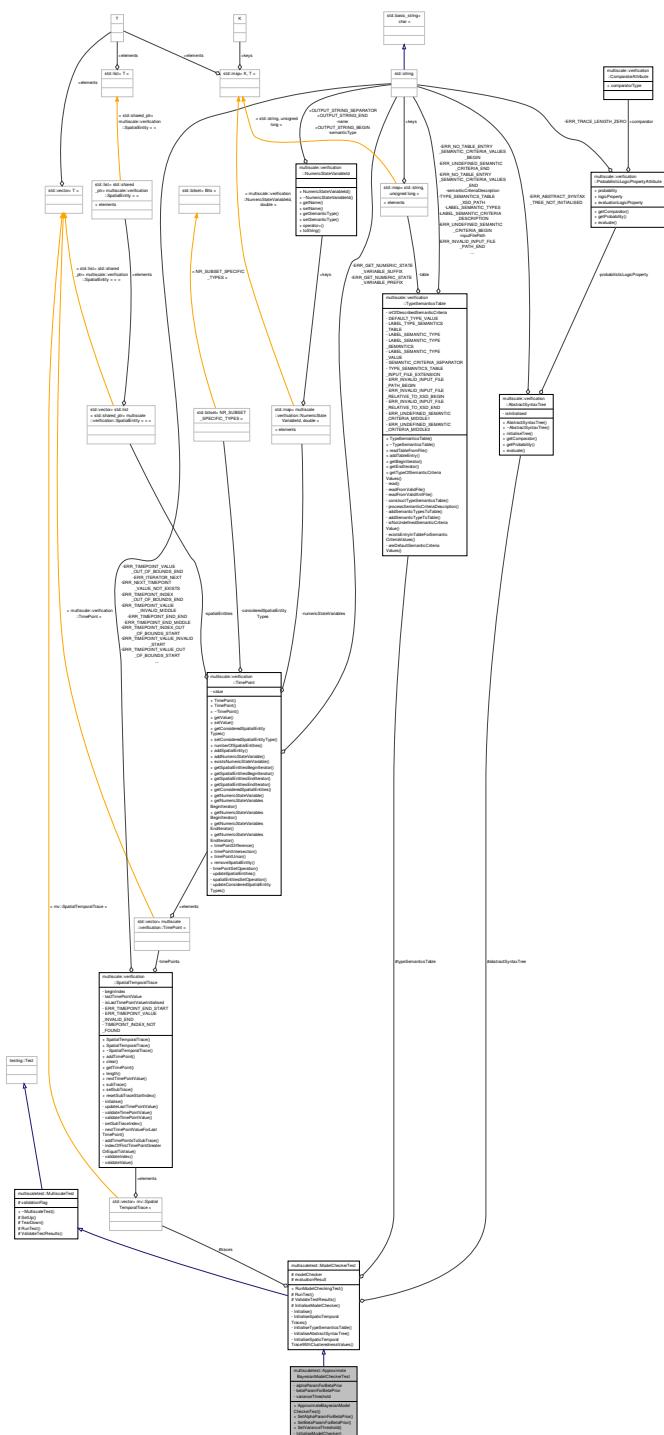
Class for testing the approximate Bayesian model checker.

```
#include <ApproximateBayesianModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::ApproximateBayesianModelCheckerTest:



## Collaboration diagram for multiscaletest::ApproximateBayesianModelCheckerTest:



## Public Member Functions

- `ApproximateBayesianModelCheckerTest ()`
  - `void SetAlphaParamForBetaPrior (double alphaParamForBetaPrior)`

*Set the value of the alpha parameter for the beta prior.*

- void SetBetaParamForBetaPrior (double betaParamForBetaPrior)

*Set the value of the beta parameter for the beta prior.*

- void [SetVarianceThreshold](#) (double *v*ariance*T*hreshold)  
*S*et the value of the variance threshold.

### Private Member Functions

- void [InitialiseModelChecker](#) () override  
*I*nitalise the model checker.

### Private Attributes

- double alphaParamForBetaPrior
- double betaParamForBetaPrior
- double varianceThreshold

### Additional Inherited Members

#### 7.9.1 Detailed Description

Class for testing the approximate Bayesian model checker.

Definition at line 15 of file ApproximateBayesianModelCheckerTest.hpp.

#### 7.9.2 Constructor & Destructor Documentation

##### 7.9.2.1 multiscaletest::ApproximateBayesianModelCheckerTest::ApproximateBayesianModelCheckerTest ( ) [inline]

Definition at line 26 of file ApproximateBayesianModelCheckerTest.hpp.

#### 7.9.3 Member Function Documentation

##### 7.9.3.1 void multiscaletest::ApproximateBayesianModelCheckerTest::InitialiseModelChecker ( ) [override], [private], [virtual]

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 68 of file ApproximateBayesianModelCheckerTest.hpp.

##### 7.9.3.2 void multiscaletest::ApproximateBayesianModelCheckerTest::SetAlphaParamForBetaPrior ( double *alphaParamForBetaPrior* )

Set the value of the alpha parameter for the beta prior.

#### Parameters

|                                   |                                        |
|-----------------------------------|----------------------------------------|
| <i>alphaParam</i><br>ForBetaPrior | The alpha parameter for the beta prior |
|-----------------------------------|----------------------------------------|

Definition at line 56 of file ApproximateBayesianModelCheckerTest.hpp.

##### 7.9.3.3 void multiscaletest::ApproximateBayesianModelCheckerTest::SetBetaParamForBetaPrior ( double *betaParamForBetaPrior* )

Set the value of the beta parameter for the beta prior.

**Parameters**

|                                         |                                       |
|-----------------------------------------|---------------------------------------|
| <i>betaParamFor</i><br><i>BetaPrior</i> | The beta parameter for the beta prior |
|-----------------------------------------|---------------------------------------|

Definition at line 60 of file ApproximateBayesianModelCheckerTest.hpp.

#### 7.9.3.4 void multiscaletest::ApproximateBayesianModelCheckerTest::SetVarianceThreshold ( double *varianceThreshold* )

Set the value of the variance threshold.

**Parameters**

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| <i>variance</i><br><i>Threshold</i> | The value of the variance threshold |
|-------------------------------------|-------------------------------------|

Definition at line 64 of file ApproximateBayesianModelCheckerTest.hpp.

### 7.9.4 Member Data Documentation

#### 7.9.4.1 double multiscaletest::ApproximateBayesianModelCheckerTest::alphaParamForBetaPrior [private]

The alpha parameter for the beta prior

Definition at line 19 of file ApproximateBayesianModelCheckerTest.hpp.

#### 7.9.4.2 double multiscaletest::ApproximateBayesianModelCheckerTest::betaParamForBetaPrior [private]

The beta parameter for the beta prior

Definition at line 20 of file ApproximateBayesianModelCheckerTest.hpp.

#### 7.9.4.3 double multiscaletest::ApproximateBayesianModelCheckerTest::varianceThreshold [private]

The considered variance threshold T

Definition at line 22 of file ApproximateBayesianModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

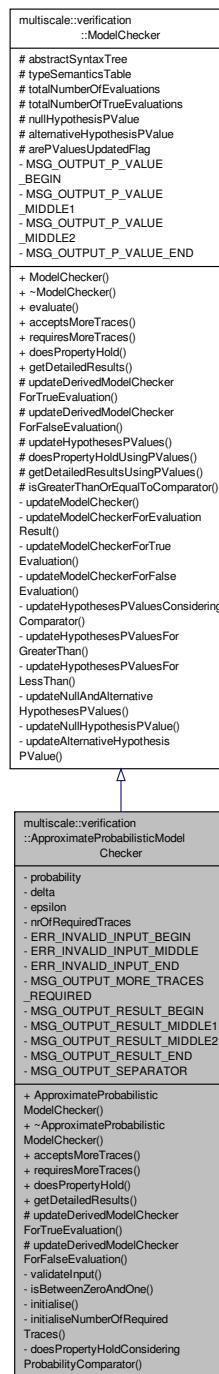
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateBayesianModelCheckerTest.hpp

## 7.10 multiscale::verification::ApproximateProbabilisticModelChecker Class Reference

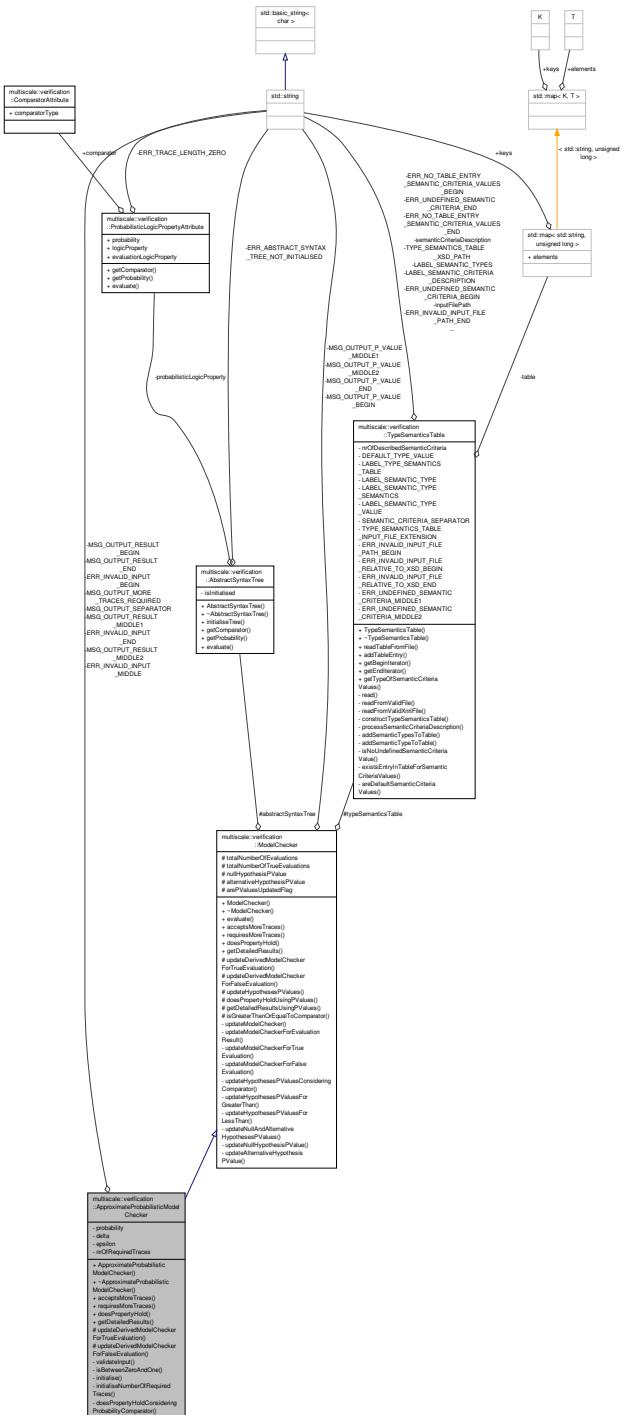
Class used to run approximate probabilistic model checking tasks.

```
#include <ApproximateProbabilisticModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::ApproximateProbabilisticModelChecker:



## Collaboration diagram for multiscale::verification::ApproximateProbabilisticModelChecker:



## Public Member Functions

- `ApproximateProbabilisticModelChecker` (const `AbstractSyntaxTree` &`abstractSyntaxTree`, const `TypeSemanticsTable` &`typeSemanticsTable`, double `delta`, double `epsilon`)
  - `~ApproximateProbabilisticModelChecker` ()
  - bool `acceptsMoreTraces` () override

*Check if more traces are accepted for evaluating the logic property.*

- bool `requiresMoreTraces ()` override  
*Check if more traces are required for evaluating the logic property.*
- bool `doesPropertyHold ()` override  
*Check if the given property holds.*
- std::string `getDetailedResults ()` override  
*Get the detailed description of the results.*

## Protected Member Functions

- void `updateDerivedModelCheckerForTrueEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- void `updateDerivedModelCheckerForFalseEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*

## Private Member Functions

- void `validateInput (double delta, double epsilon)`  
*Validate the input parameters `delta` and `epsilon`.*
- bool `isBetweenZeroAndOne (double value)`  
*Check if the given value is between zero and one (exclusive)*
- void `initialise ()`  
*Initialisation of some of the class members.*
- void `initialiseNumberOfRequiredTraces ()`  
*Initialise the number of required traces.*
- bool `doesPropertyHoldConsideringProbabilityComparator ()`  
*Check if the given property holds considering the probability comparator (i.e.  $\leq$ ,  $\geq$ )*

## Private Attributes

- double `probability`
- double `delta`
- double `epsilon`
- unsigned int `nrOfRequiredTraces`

## Static Private Attributes

- static const std::string `ERR_INVALID_INPUT_BEGIN` = "The values of the provided input parameters `delta` and `epsilon` ("
- static const std::string `ERR_INVALID_INPUT_MIDDLE` = ", "
- static const std::string `ERR_INVALID_INPUT_END` = ") must be between zero and one (exclusive). Please change."
- static const std::string `MSG_OUTPUT_MORE_TRACES_REQUIRED` = "More traces are required to provide a true/false answer assuming the given upper bound on the `probability` of the computed `probability` to deviate from the true probability. Probabilistic black-box model checking was used instead to provide an answer."
- static const std::string `MSG_OUTPUT_RESULT_BEGIN` = "The provided answer is given assuming the upper bound on the `probability` to deviate more than `epsilon` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE1` = " from the true `probability` is `delta` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE2` = ". The number of required samples was N = "
- static const std::string `MSG_OUTPUT_RESULT_END` = ""
- static const std::string `MSG_OUTPUT_SEPARATOR` = " "

## Additional Inherited Members

### 7.10.1 Detailed Description

Class used to run approximate probabilistic model checking tasks.

The implementation of this class is based on the algorithm described in the following paper:

T. Héault, R. Lassaigne, F. Magniette, and S. Peyronnet, ‘Approximate Probabilistic Model Checking’, in Verification, Model Checking, and Abstract Interpretation, B. Steffen and G. Levi, Eds. Springer Berlin Heidelberg, 2004, pp. 73–84.

Definition at line 23 of file ApproximateProbabilisticModelChecker.hpp.

### 7.10.2 Constructor & Destructor Documentation

#### 7.10.2.1 ApproximateProbabilisticModelChecker::ApproximateProbabilisticModelChecker ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable, double delta, double epsilon )

Definition at line 10 of file ApproximateProbabilisticModelChecker.cpp.

References delta, epsilon, initialise(), and validateInput().

#### 7.10.2.2 ApproximateProbabilisticModelChecker::~ApproximateProbabilisticModelChecker ( )

Definition at line 27 of file ApproximateProbabilisticModelChecker.cpp.

### 7.10.3 Member Function Documentation

#### 7.10.3.1 bool ApproximateProbabilisticModelChecker::acceptsMoreTraces ( ) [override], [virtual]

Check if more traces are accepted for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 29 of file ApproximateProbabilisticModelChecker.cpp.

References nrOfRequiredTraces, and multiscale::verification::ModelChecker::totalNumberOfEvaluations.

Referenced by requiresMoreTraces().

#### 7.10.3.2 bool ApproximateProbabilisticModelChecker::doesPropertyHold ( ) [override], [virtual]

Check if the given property holds.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 37 of file ApproximateProbabilisticModelChecker.cpp.

References doesPropertyHoldConsideringProbabilityComparator(), multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues(), and requiresMoreTraces().

#### 7.10.3.3 bool ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator ( ) [private]

Check if the given property holds considering the probability comparator (i.e.  $\leq$ ,  $\geq$ )

For queries of type : a)  $P \geq \theta[\phi]$  result =  $(nr_{true}, races / nr_races) - \epsilon \geq \theta$  b)  $P \leq \theta[\phi]$  result =  $(nr_{true}, races / nr_races) + \epsilon \leq \theta$

Definition at line 93 of file ApproximateProbabilisticModelChecker.cpp.

References epsilon, multiscale::Numeric::greaterOrEqual(), multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator(), multiscale::Numeric::lessOrEqual(), probability, multiscale::verification::ModelChecker::totalNumberOfEvaluations, and multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations.

Referenced by doesPropertyHold().

#### 7.10.3.4 std::string ApproximateProbabilisticModelChecker::getDetailedResults( ) [override], [virtual]

Get the detailed description of the results.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 45 of file ApproximateProbabilisticModelChecker.cpp.

References delta, epsilon, multiscale::verification::ModelChecker::getDetailedResultsUsingPValues(), MSG\_OUTPUT\_MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_RESULT\_BEGIN, MSG\_OUTPUT\_RESULT\_END, MSG\_OUTPUT\_RESULT\_MIDDLE1, MSG\_OUTPUT\_RESULT\_MIDDLE2, MSG\_OUTPUT\_SEPARATOR, nrOfRequiredTraces, requiresMoreTraces(), and multiscale::StringManipulator::toString().

#### 7.10.3.5 void ApproximateProbabilisticModelChecker::initialise( ) [private]

Initialisation of some of the class members.

Definition at line 80 of file ApproximateProbabilisticModelChecker.cpp.

References multiscale::verification::ModelChecker::abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), initialiseNumberOfRequiredTraces(), and probability.

Referenced by ApproximateProbabilisticModelChecker().

#### 7.10.3.6 void ApproximateProbabilisticModelChecker::initialiseNumberOfRequiredTraces( ) [private]

Initialise the number of required traces.

Precondition: The class members delta and epsilon are correctly initialised.

Definition at line 86 of file ApproximateProbabilisticModelChecker.cpp.

References delta, epsilon, and nrOfRequiredTraces.

Referenced by initialise().

#### 7.10.3.7 bool ApproximateProbabilisticModelChecker::isBetweenZeroAndOne( double value ) [private]

Check if the given value is between zero and one (exclusive)

Parameters

|                    |                 |
|--------------------|-----------------|
| <code>value</code> | The given value |
|--------------------|-----------------|

Definition at line 76 of file ApproximateProbabilisticModelChecker.cpp.

Referenced by validateInput().

#### 7.10.3.8 bool ApproximateProbabilisticModelChecker::requiresMoreTraces( ) [override], [virtual]

Check if more traces are required for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 33 of file ApproximateProbabilisticModelChecker.cpp.

References `acceptsMoreTraces()`.

Referenced by `doesPropertyHold()`, and `getDetailedResults()`.

#### 7.10.3.9 void ApproximateProbabilisticModelChecker::updateDerivedModelCheckerForFalseEvaluation( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Do not do anything

Implements `multiscale::verification::ModelChecker`.

Definition at line 63 of file `ApproximateProbabilisticModelChecker.cpp`.

#### 7.10.3.10 void ApproximateProbabilisticModelChecker::updateDerivedModelCheckerForTrueEvaluation( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Do not do anything

Implements `multiscale::verification::ModelChecker`.

Definition at line 61 of file `ApproximateProbabilisticModelChecker.cpp`.

#### 7.10.3.11 void ApproximateProbabilisticModelChecker::validateInput( double *delta*, double *epsilon* ) [private]

Validate the input parameters *delta* and *epsilon*.

Precondition:  $0 < \delta, \epsilon < 1$

Parameters

|                |                                                                                   |
|----------------|-----------------------------------------------------------------------------------|
| <i>delta</i>   | The upper bound on the probability to deviate from the true probability           |
| <i>epsilon</i> | The considered amount by which the probability deviates from the true probability |

Definition at line 65 of file `ApproximateProbabilisticModelChecker.cpp`.

References `ERR_INVALID_INPUT_BEGIN`, `ERR_INVALID_INPUT_END`, `ERR_INVALID_INPUT_MIDDLE`, `isBetweenZeroAndOne()`, `MS_throw`, and `multiscale::StringManipulator::toString()`.

Referenced by `ApproximateProbabilisticModelChecker()`.

### 7.10.4 Member Data Documentation

#### 7.10.4.1 double multiscale::verification::ApproximateProbabilisticModelChecker::delta [private]

The upper bound on the probability for the computed probability to deviate from the true probability

Definition at line 30 of file `ApproximateProbabilisticModelChecker.hpp`.

Referenced by `ApproximateProbabilisticModelChecker()`, `getDetailedResults()`, and `initialiseNumberOfRequiredTraces()`.

#### 7.10.4.2 double multiscale::verification::ApproximateProbabilisticModelChecker::epsilon [private]

The considered deviation from the true probability

Definition at line 32 of file `ApproximateProbabilisticModelChecker.hpp`.

Referenced by ApproximateProbabilisticModelChecker(), doesPropertyHoldConsideringProbabilityComparator(), getDetailedResults(), and initialiseNumberOfRequiredTraces().

7.10.4.3 `const std::string ApproximateProbabilisticModelChecker::ERR_INVALID_INPUT_BEGIN = "The values of the provided input parameters delta and epsilon (" [static], [private]`

Definition at line 104 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by validateInput().

7.10.4.4 `const std::string ApproximateProbabilisticModelChecker::ERR_INVALID_INPUT_END = ") must be between zero and one (exclusive). Please change." [static], [private]`

Definition at line 106 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by validateInput().

7.10.4.5 `const std::string ApproximateProbabilisticModelChecker::ERR_INVALID_INPUT_MIDDLE = ", " [static], [private]`

Definition at line 105 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by validateInput().

7.10.4.6 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_MORE_TRACES_REQUIRED = "More traces are required to provide a true/false answer assuming the given upper bound on the probability of the computed probability to deviate from the true probability. Probabilistic black-box model checking was used instead to provide an answer." [static], [private]`

Definition at line 108 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

7.10.4.7 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_RESULT_BEGIN = "The provided answer is given assuming the upper bound on the probability to deviate more than epsilon = " [static], [private]`

Definition at line 110 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

7.10.4.8 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_RESULT_END = "" [static], [private]`

Definition at line 113 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

7.10.4.9 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_RESULT_MIDDLE1 = " from the true probability is delta = " [static], [private]`

Definition at line 111 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

7.10.4.10 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_RESULT_MIDDLE2 = ". The number of required samples was N = " [static], [private]`

Definition at line 112 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by `getDetailedResults()`.

7.10.4.11 `const std::string ApproximateProbabilisticModelChecker::MSG_OUTPUT_SEPARATOR = " " [static], [private]`

Definition at line 115 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by `getDetailedResults()`.

7.10.4.12 `unsigned int multiscale::verification::ApproximateProbabilisticModelChecker::nrOfRequiredTraces [private]`

The number of required traces

Definition at line 34 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by `acceptsMoreTraces()`, `getDetailedResults()`, and `initialiseNumberOfRequiredTraces()`.

7.10.4.13 `double multiscale::verification::ApproximateProbabilisticModelChecker::probability [private]`

The probability specified by the user for the logic property to be evaluated

Definition at line 27 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by `doesPropertyHoldConsideringProbabilityComparator()`, and `initialise()`.

The documentation for this class was generated from the following files:

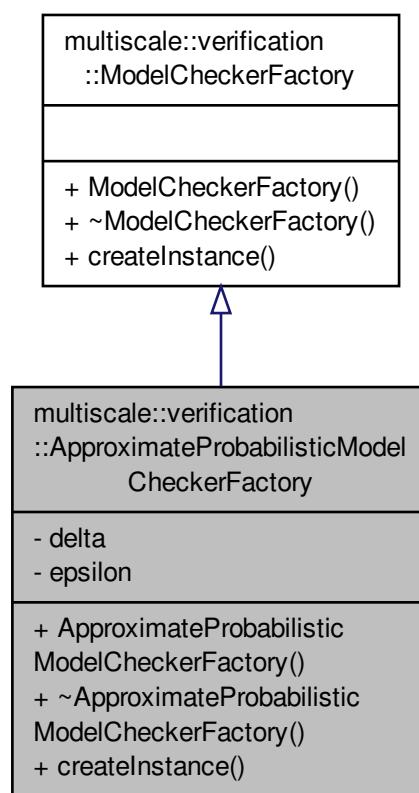
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ApproximateProbabilisticModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ApproximateProbabilisticModelChecker.cpp](#)

## 7.11 multiscale::verification::ApproximateProbabilisticModelCheckerFactory Class Reference

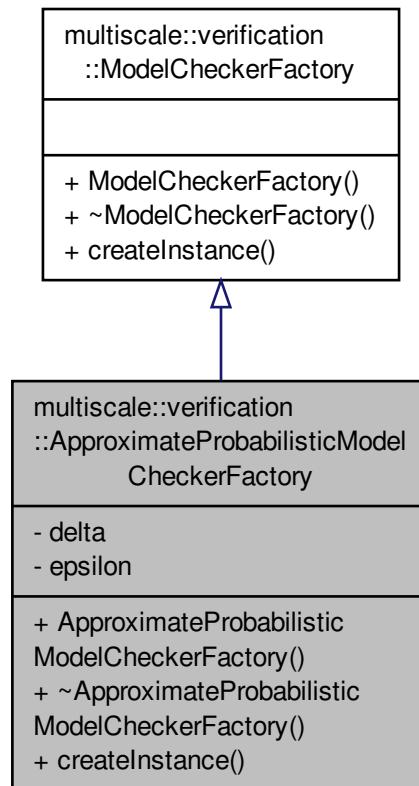
Class for creating [ApproximateProbabilisticModelChecker](#) instances.

```
#include <ApproximateProbabilisticModelCheckerFactory.hpp>
```

## Inheritance diagram for multiscale::verification::ApproximateProbabilisticModelCheckerFactory:



Collaboration diagram for multiscale::verification::ApproximateProbabilisticModelCheckerFactory:



## Public Member Functions

- `ApproximateProbabilisticModelCheckerFactory` (double `delta`, double `epsilon`)
- `~ApproximateProbabilisticModelCheckerFactory ()`
- `std::shared_ptr< ModelChecker > createInstance` (const `AbstractSyntaxTree` &`abstractSyntaxTree`, const `TypeSemanticsTable` &`typeSemanticsTable`) override

*Create an instance of `ApproximateProbabilisticModelChecker`.*

## Private Attributes

- double `delta`
- double `epsilon`

### 7.11.1 Detailed Description

Class for creating `ApproximateProbabilisticModelChecker` instances.

Definition at line 12 of file `ApproximateProbabilisticModelChecker.hpp`.

## 7.11.2 Constructor & Destructor Documentation

7.11.2.1 `ApproximateProbabilisticModelCheckerFactory::ApproximateProbabilisticModelCheckerFactory ( double delta, double epsilon )`

Definition at line 7 of file ApproximateProbabilisticModelCheckerFactory.cpp.

7.11.2.2 `ApproximateProbabilisticModelCheckerFactory::~ApproximateProbabilisticModelCheckerFactory ( )`

Definition at line 11 of file ApproximateProbabilisticModelCheckerFactory.cpp.

## 7.11.3 Member Function Documentation

7.11.3.1 `std::shared_ptr< ModelChecker > ApproximateProbabilisticModelCheckerFactory::createInstance ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [override], [virtual]`

Create an instance of [ApproximateProbabilisticModelChecker](#).

### Parameters

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| <code>abstractSyntaxTree</code> | The abstract syntax tree representing the logic property to be checked                |
| <code>typeSemanticsTable</code> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implements [multiscale::verification::ModelCheckerFactory](#).

Definition at line 14 of file ApproximateProbabilisticModelCheckerFactory.cpp.

References delta, and epsilon.

## 7.11.4 Member Data Documentation

7.11.4.1 `double multiscale::verification::ApproximateProbabilisticModelCheckerFactory::delta [private]`

The upper bound on the probability for the computed probability to deviate from the true probability

Definition at line 16 of file ApproximateProbabilisticModelCheckerFactory.hpp.

Referenced by `createInstance()`.

7.11.4.2 `double multiscale::verification::ApproximateProbabilisticModelCheckerFactory::epsilon [private]`

The considered deviation from the true probability

Definition at line 18 of file ApproximateProbabilisticModelCheckerFactory.hpp.

Referenced by `createInstance()`.

The documentation for this class was generated from the following files:

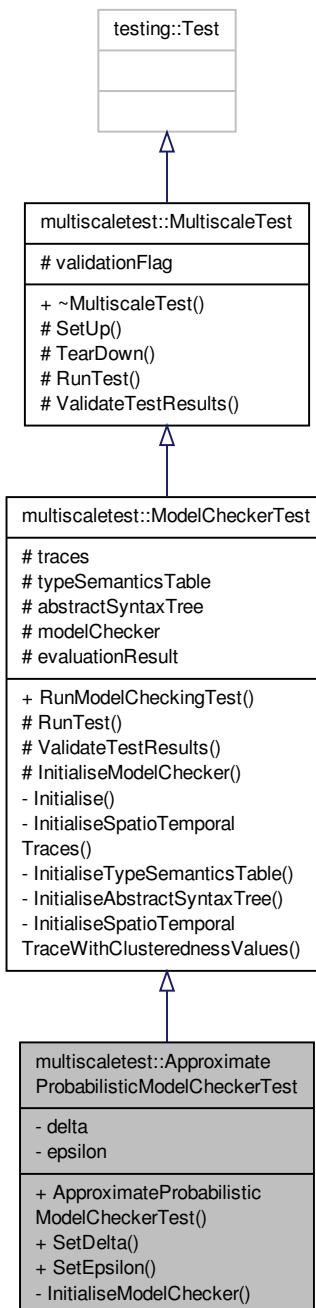
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ApproximateProbabilisticModelCheckerFactory.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ApproximateProbabilisticModelCheckerFactory.cpp](#)

## 7.12 multiscaletest::ApproximateProbabilisticModelCheckerTest Class Reference

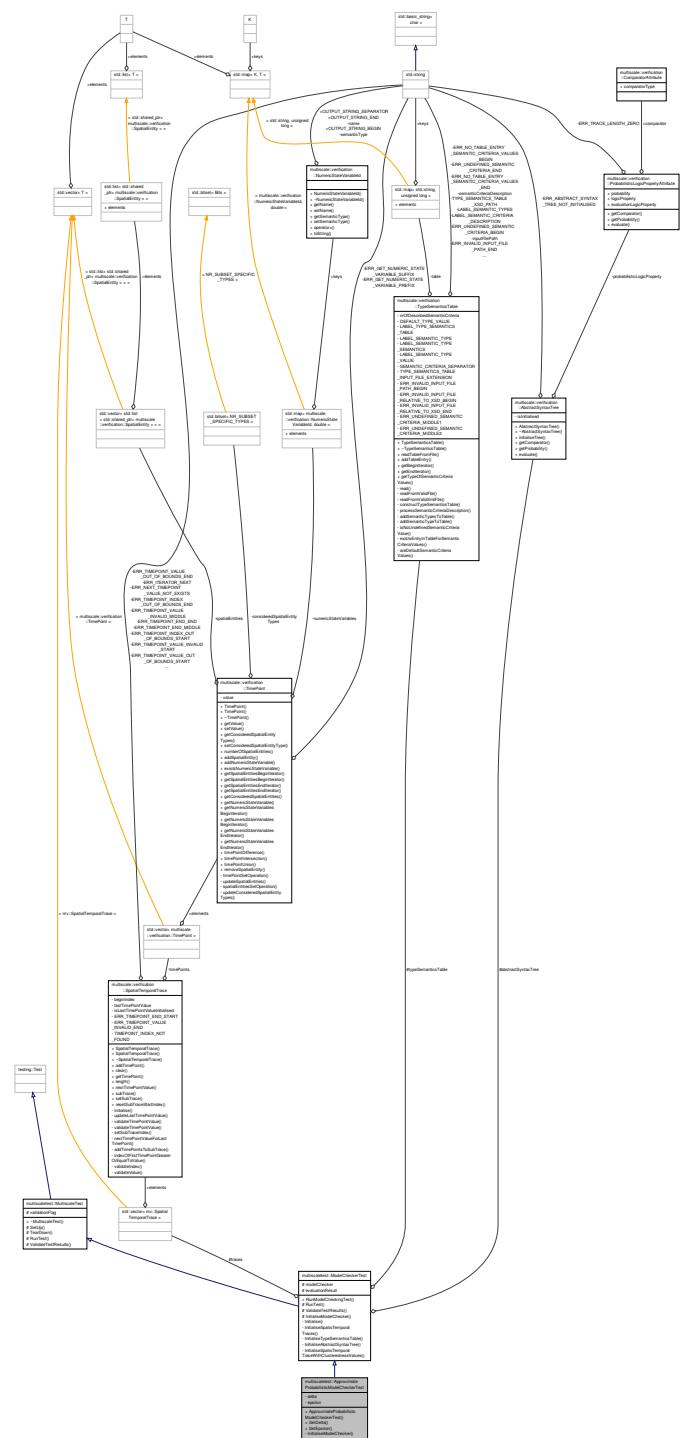
Class for testing the approximate probabilistic model checker.

```
#include <ApproximateProbabilisticModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::ApproximateProbabilisticModelCheckerTest:



## Collaboration diagram for multiscaletest::ApproximateProbabilisticModelCheckerTest:



## Public Member Functions

- `ApproximateProbabilisticModelCheckerTest()`
  - `void SetDelta (double delta)`  
*Set the value of delta.*
  - `void SetEpsilon (double epsilon)`  
*Set the value of epsilon.*

## Private Member Functions

- void [InitialiseModelChecker \(\) override](#)

*Initialise the model checker.*

## Private Attributes

- double delta
- double epsilon

## Additional Inherited Members

### 7.12.1 Detailed Description

Class for testing the approximate probabilistic model checker.

Definition at line 15 of file ApproximateProbabilisticModelCheckerTest.hpp.

### 7.12.2 Constructor & Destructor Documentation

#### 7.12.2.1 [multiscaletest::ApproximateProbabilisticModelCheckerTest::ApproximateProbabilisticModelCheckerTest \( \) \[inline\]](#)

Definition at line 24 of file ApproximateProbabilisticModelCheckerTest.hpp.

### 7.12.3 Member Function Documentation

#### 7.12.3.1 [void multiscaletest::ApproximateProbabilisticModelCheckerTest::InitialiseModelChecker \( \) \[override\], \[private\], \[virtual\]](#)

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 55 of file ApproximateProbabilisticModelCheckerTest.hpp.

#### 7.12.3.2 [void multiscaletest::ApproximateProbabilisticModelCheckerTest::SetDelta \( double delta \)](#)

Set the value of delta.

##### Parameters

|              |                    |
|--------------|--------------------|
| <i>delta</i> | The value of delta |
|--------------|--------------------|

Definition at line 47 of file ApproximateProbabilisticModelCheckerTest.hpp.

#### 7.12.3.3 [void multiscaletest::ApproximateProbabilisticModelCheckerTest::SetEpsilon \( double epsilon \)](#)

Set the value of epsilon.

##### Parameters

|                      |                      |
|----------------------|----------------------|
| <code>epsilon</code> | The value of epsilon |
|----------------------|----------------------|

Definition at line 51 of file ApproximateProbabilisticModelCheckerTest.hpp.

#### 7.12.4 Member Data Documentation

##### 7.12.4.1 double multiscaletest::ApproximateProbabilisticModelCheckerTest::delta [private]

The value of delta in the Chernoff-Hoeffding inequality

Definition at line 19 of file ApproximateProbabilisticModelCheckerTest.hpp.

##### 7.12.4.2 double multiscaletest::ApproximateProbabilisticModelCheckerTest::epsilon [private]

The value of epsilon in the Chernoff-Hoeffding inequality

Definition at line 20 of file ApproximateProbabilisticModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

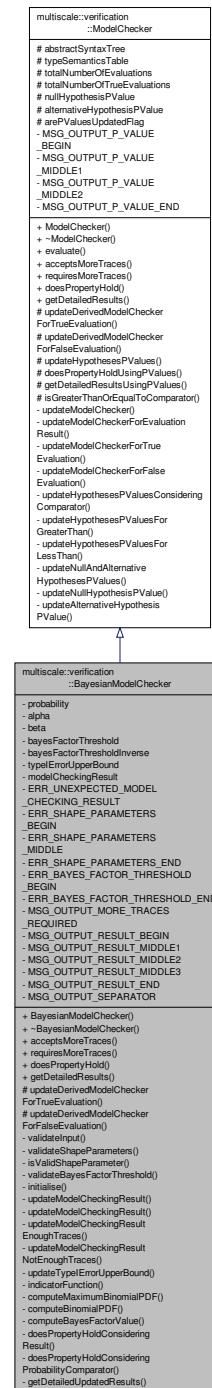
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateProbabilisticModelCheckerTest.hpp

## 7.13 multiscale::verification::BayesianModelChecker Class Reference

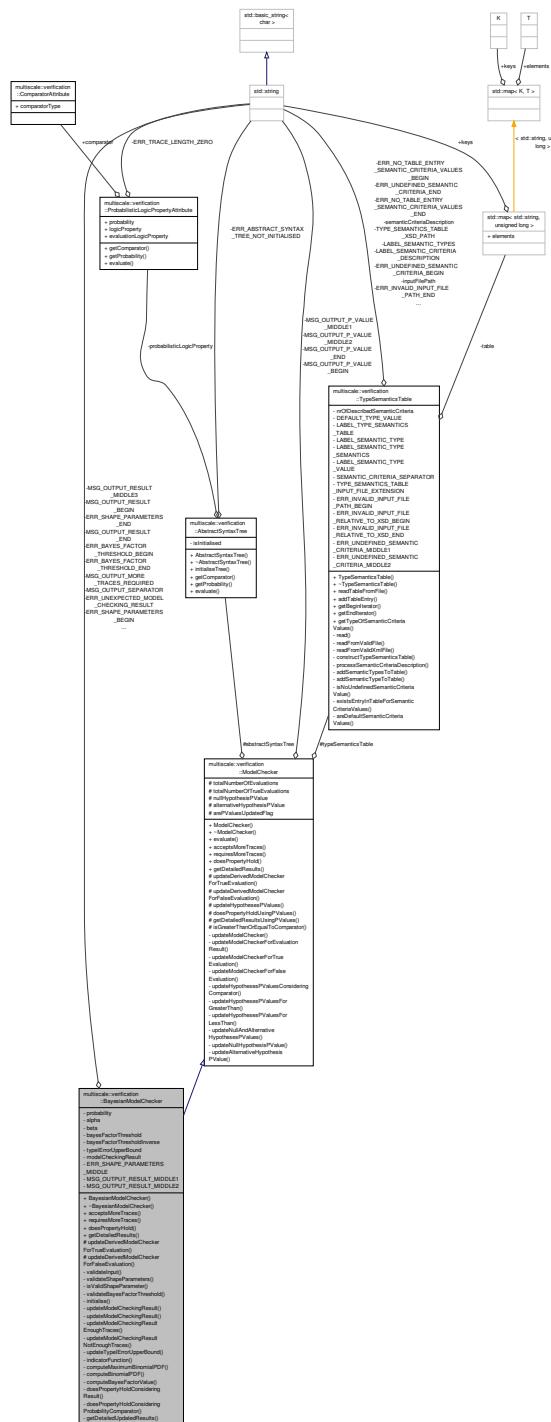
Class used to run Bayesian model checking tasks.

```
#include <BayesianModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::BayesianModelChecker:



## Collaboration diagram for multiscale::verification::BayesianModelChecker



## Public Member Functions

- `BayesianModelChecker` (const `AbstractSyntaxTree` &`abstractSyntaxTree`, const `TypeSemanticsTable` &`typeSemanticsTable`, double `alpha`, double `beta`, double `bayesFactorThreshold`)
  - `~BayesianModelChecker` ()
  - bool `acceptsMoreTraces` () override

*Check if more traces are accepted for evaluating the logic property.*

- bool `requiresMoreTraces ()` override  
*Check if more traces are required for evaluating the logic property.*
- bool `doesPropertyHold ()` override  
*Check if the given property holds.*
- std::string `getDetailedResults ()` override  
*Get the detailed description of the results.*

## Protected Member Functions

- void `updateDerivedModelCheckerForTrueEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- void `updateDerivedModelCheckerForFalseEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*

## Private Member Functions

- void `validateInput (double alpha, double beta, double bayesFactorThreshold)`  
*Validate the input parameters  $\alpha$ ,  $\beta$  and the Bayes factor threshold.*
- void `validateShapeParameters (double alpha, double beta)`  
*Validate the shape parameters  $\alpha$  and  $\beta$ .*
- bool `isValidShapeParameter (double shapeParameter)`  
*Check if the given shape parameter value is valid.*
- void `validateBayesFactorThreshold (double bayesFactorThreshold)`  
*Validate the Bayes factor threshold.*
- void `initialise ()`  
*Initialisation of some of the class members.*
- void `updateModelCheckingResult ()`  
*Update the result of the model checking task.*
- void `updateModelCheckingResult (double bayesFactor)`  
*Update the result of the model checking task considering the given Bayes factor value.*
- void `updateModelCheckingResultEnoughTraces (double bayesFactor)`  
*Update the result of the model checking task considering that enough traces have been provided.*
- void `updateModelCheckingResultNotEnoughTraces ()`  
*Update the result of the model checking task considering that not enough traces have been provided.*
- void `updateTypeIErrorUpperBound ()`  
*Update the value of the type I error upper bound.*
- bool `indicatorFunction (unsigned int nrOfSuccesses)`  
*Compute the value of the indicator function  $I_{\mathcal{B}(n,x) < 1/T}(x)$ .*
- double `computeMaximumBinomialPDF (unsigned int nrOfSuccesses)`  
*Compute the maximum value of the probability distribution function for the Binomial distribution.*
- double `computeBinomialPDF (unsigned int nrOfSuccesses, double probability)`  
*Compute the value of the probability distribution function for the Binomial distribution.*
- double `computeBayesFactorValue (unsigned int nrOfObservations, unsigned int nrOfSuccesses)`  
*Compute the value of the Bayes factor.*
- bool `doesPropertyHoldConsideringResult ()`  
*Check if the given property holds considering the obtained model checking result.*
- bool `doesPropertyHoldConsideringProbabilityComparator (bool isNullHypothesisTrue)`  
*Check if the given property holds considering the obtained answer and probability comparator (i.e.  $\leq$ ,  $\geq$ )*
- std::string `getDetailedUpdatedResults ()`  
*Get the detailed description of the updated results.*

## Private Attributes

- double probability
- double alpha
- double beta
- double bayesFactorThreshold
- double bayesFactorThresholdInverse
- double typeIErrorUpperBound
- BayesianModelCheckingResult modelCheckingResult

## Static Private Attributes

- static const std::string `ERR_UNEXPECTED_MODEL_CHECKING_RESULT` = "An invalid Bayesian model checking result was obtained. Please check source code."
- static const std::string `ERR_SHAPE_PARAMETERS_BEGIN` = "The provided Beta distribution shape parameters `alpha` and `beta` ("
- static const std::string `ERR_SHAPE_PARAMETERS_MIDDLE` = ", "
- static const std::string `ERR_SHAPE_PARAMETERS_END` = ") should be greater than zero. Please change."
- static const std::string `ERR_BAYES_FACTOR_THRESHOLD_BEGIN` = "The provided Bayes factor threshold ("
- static const std::string `ERR_BAYES_FACTOR_THRESHOLD_END` = ") should be greater than one. Please change."
- static const std::string `MSG_OUTPUT_MORE_TRACES_REQUIRED` = "More traces are required to provide a true/false answer assuming the given Beta distribution shape parameters and Bayes factor threshold value. Probabilistic black-box model checking was used instead to provide an answer."
- static const std::string `MSG_OUTPUT_RESULT_BEGIN` = "The provided answer is given for the Beta distribution shape parameters `alpha` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE1` = " and `beta` = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE2` = ", and Bayes factor threshold value = "
- static const std::string `MSG_OUTPUT_RESULT_MIDDLE3` = ". The type I error upper bound for the provided answer is = "
- static const std::string `MSG_OUTPUT_RESULT_END` = ""
- static const std::string `MSG_OUTPUT_SEPARATOR` = " "

## Additional Inherited Members

### 7.13.1 Detailed Description

Class used to run Bayesian model checking tasks.

The implementation of this class is (partially) based on the algorithms described in the following paper:

S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, and P. Zuliani, ‘A Bayesian Approach to Model Checking Biological Systems’, in Computational Methods in Systems Biology, P. Degano and R. Gorrieri, Eds. Springer Berlin Heidelberg, 2009, pp. 218–234.

In our implementation the variables in the original paper (right hand side of the assignments) have been given the following new names (left hand side of assignments):

`probability` =  $\theta$

`alpha` =  $\alpha$

`beta` =  $\beta$

`bayesFactor` =  $\mathcal{B}_n$

`bayesFactorThreshold` =  $T$

`totalNumberOfEvaluations` =  $n$

`totalNumberOfTrueEvaluations = x`

Definition at line 50 of file BayesianModelChecker.hpp.

### 7.13.2 Constructor & Destructor Documentation

7.13.2.1 `BayesianModelChecker::BayesianModelChecker ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable, double alpha, double beta, double bayesFactorThreshold )`

Definition at line 15 of file BayesianModelChecker.cpp.

References `alpha`, `bayesFactorThreshold`, `beta`, `initialise()`, and `validateInput()`.

7.13.2.2 `BayesianModelChecker::~BayesianModelChecker ( )`

Definition at line 28 of file BayesianModelChecker.cpp.

### 7.13.3 Member Function Documentation

7.13.3.1 `bool BayesianModelChecker::acceptsMoreTraces ( ) [override], [virtual]`

Check if more traces are accepted for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 30 of file BayesianModelChecker.cpp.

References `modelCheckingResult`, `multiscale::verification::MORE_TRACES_REQUIRED`, and `updateModelCheckingResult()`.

Referenced by `requiresMoreTraces()`.

7.13.3.2 `double BayesianModelChecker::computeBayesFactorValue ( unsigned int nrOfObservations, unsigned int nrOfSuccesses ) [private]`

Compute the value of the Bayes factor.

According to the original paper the Bayes factor can be computed as follows:  $\mathcal{B} = \frac{1}{(F_{x+\alpha, n-x+\beta})(\theta)} - 1$

#### Parameters

|                               |                                  |
|-------------------------------|----------------------------------|
| <code>nrOfObservations</code> | The total number of observations |
| <code>nrOfSuccesses</code>    | The total number of successes    |

Definition at line 158 of file BayesianModelChecker.cpp.

References `multiscale::Numeric::almostEqual()`, `alpha`, `beta`, `multiscale::BetaDistribution::cdf()`, and `probability`.

Referenced by `indicatorFunction()`, and `updateModelCheckingResult()`.

7.13.3.3 `double BayesianModelChecker::computeBinomialPDF ( unsigned int nrOfSuccesses, double probability ) [private]`

Compute the value of the probability distribution function for the Binomial distribution.

## Parameters

|                      |                                              |
|----------------------|----------------------------------------------|
| <i>nrOfSuccesses</i> | The number of successful observations/trials |
| <i>probability</i>   | The probability of success                   |

Definition at line 152 of file BayesianModelChecker.cpp.

References multiscale::BinomialDistribution::pdf(), and multiscale::verification::ModelChecker::totalNumberOfEvaluations.

Referenced by computeMaximumBinomialPDF().

#### 7.13.3.4 double BayesianModelChecker::computeMaximumBinomialPDF ( unsigned int *nrOfSuccesses* ) [private]

Compute the maximum value of the probability distribution function for the Binomial distribution.

The maximum value is reached when  $p = \theta$  or  $p = \frac{2k}{n}$

## Parameters

|                      |                                              |
|----------------------|----------------------------------------------|
| <i>nrOfSuccesses</i> | The number of successful observations/trials |
|----------------------|----------------------------------------------|

Definition at line 142 of file BayesianModelChecker.cpp.

References computeBinomialPDF(), probability, and multiscale::verification::ModelChecker::totalNumberOfEvaluations.

Referenced by updateTypeIErrorUpperBound().

#### 7.13.3.5 bool BayesianModelChecker::doesPropertyHold ( ) [override], [virtual]

Check if the given property holds.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 40 of file BayesianModelChecker.cpp.

References doesPropertyHoldConsideringResult(), and updateModelCheckingResult().

#### 7.13.3.6 bool BayesianModelChecker::doesPropertyHoldConsideringProbabilityComparator ( bool *isNullHypothesisTrue* ) [private]

Check if the given property holds considering the obtained answer and probability comparator (i.e.  $\leq$ ,  $\geq$ )

For queries of type : a)  $P \geq \theta[\phi]$  the *isNullHypothesisTrue* flag value is returned b)  $P \leq \theta[\phi]$  the  $!(\text{isNullHypothesisTrue})$  flag value is returned

## Parameters

|                             |                                                                                    |
|-----------------------------|------------------------------------------------------------------------------------|
| <i>isNullHypothesisTrue</i> | Flag indicating if the null hypothesis is true considering a $P \geq [\phi]$ query |
|-----------------------------|------------------------------------------------------------------------------------|

Definition at line 189 of file BayesianModelChecker.cpp.

References multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator().

Referenced by doesPropertyHoldConsideringResult().

#### 7.13.3.7 bool BayesianModelChecker::doesPropertyHoldConsideringResult ( ) [private]

Check if the given property holds considering the obtained model checking result.

Definition at line 170 of file BayesianModelChecker.cpp.

References doesPropertyHoldConsideringProbabilityComparator(), multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues(), ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT, multiscale::verification::FA\_LSE, modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MS\_throw, and multiscale::verification::TRUE.

Referenced by doesPropertyHold().

#### 7.13.3.8 std::string BayesianModelChecker::getDetailedResults( ) [override], [virtual]

Get the detailed description of the results.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 46 of file BayesianModelChecker.cpp.

References getDetailedUpdatedResults(), and updateModelCheckingResult().

#### 7.13.3.9 std::string BayesianModelChecker::getDetailedUpdatedResults( ) [private]

Get the detailed description of the updated results.

Definition at line 197 of file BayesianModelChecker.cpp.

References alpha, bayesFactorThreshold, beta, multiscale::verification::ModelChecker::getDetailedResultsUsingPValues(), modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_MIDDLE\_TRACES\_REQUIRED, MSG\_OUTPUT\_RESULT\_BEGIN, MSG\_OUTPUT\_RESULT\_END, MSG\_OUTPUT\_RESULT\_MIDDLE1, MSG\_OUTPUT\_RESULT\_MIDDLE2, MSG\_OUTPUT\_RESULT\_MIDDLE3, MSG\_OUTPUT\_SEPARATOR, multiscale::StringManipulator::toString(), and typeIErrorUpperBound.

Referenced by getDetailedResults().

#### 7.13.3.10 bool BayesianModelChecker::indicatorFunction( unsigned int nrOfSuccesses ) [private]

Compute the value of the indicator function  $I_{\mathcal{B}(n,x) < 1/T}(x)$ .

##### Parameters

|                      |                                              |
|----------------------|----------------------------------------------|
| <i>nrOfSuccesses</i> | The number of successful observations/trials |
|----------------------|----------------------------------------------|

Definition at line 136 of file BayesianModelChecker.cpp.

References bayesFactorThresholdInverse, computeBayesFactorValue(), and multiscale::verification::ModelChecker::totalNumberOfEvaluations.

Referenced by updateTypeIErrorUpperBound().

#### 7.13.3.11 void BayesianModelChecker::initialise( ) [private]

Initialisation of some of the class members.

Definition at line 87 of file BayesianModelChecker.cpp.

References multiscale::verification::ModelChecker::abstractSyntaxTree, bayesFactorThreshold, bayesFactorThresholdInverse, multiscale::verification::AbstractSyntaxTree::getProbability(), probability, and typeIErrorUpperBound.

Referenced by BayesianModelChecker().

#### 7.13.3.12 bool BayesianModelChecker::isValidShapeParameter( double shapeParameter ) [private]

Check if the given shape parameter value is valid.

The shape parameter values should be greater than zero

**Parameters**

|                       |                           |
|-----------------------|---------------------------|
| <i>shapeParameter</i> | The given shape parameter |
|-----------------------|---------------------------|

Definition at line 72 of file BayesianModelChecker.cpp.

Referenced by validateShapeParameters().

#### 7.13.3.13 bool BayesianModelChecker::requiresMoreTraces ( ) [override], [virtual]

Check if more traces are required for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 36 of file BayesianModelChecker.cpp.

References acceptsMoreTraces().

#### 7.13.3.14 void BayesianModelChecker::updateDerivedModelCheckerForFalseEvaluation ( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 54 of file BayesianModelChecker.cpp.

#### 7.13.3.15 void BayesianModelChecker::updateDerivedModelCheckerForTrueEvaluation ( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 52 of file BayesianModelChecker.cpp.

#### 7.13.3.16 void BayesianModelChecker::updateModelCheckingResult ( ) [private]

Update the result of the model checking task.

Definition at line 97 of file BayesianModelChecker.cpp.

References computeBayesFactorValue(), multiscale::verification::ModelChecker::totalNumberOfEvaluations, multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations, and updateTypeIErrorUpperBound().

Referenced by acceptsMoreTraces(), doesPropertyHold(), and getDetailedResults().

#### 7.13.3.17 void BayesianModelChecker::updateModelCheckingResult ( double *bayesFactor* ) [private]

Update the result of the model checking task considering the given Bayes factor value.

**Parameters**

|                    |                              |
|--------------------|------------------------------|
| <i>bayesFactor</i> | The given Bayes factor value |
|--------------------|------------------------------|

Definition at line 105 of file BayesianModelChecker.cpp.

References `bayesFactorThreshold`, `bayesFactorThresholdInverse`, `updateModelCheckingResultEnoughTraces()`, and `updateModelCheckingResultNotEnoughTraces()`.

#### 7.13.3.18 void BayesianModelChecker::updateModelCheckingResultEnoughTraces ( double *bayesFactor* ) [private]

Update the result of the model checking task considering that enough traces have been provided.

##### Parameters

|                    |                              |
|--------------------|------------------------------|
| <i>bayesFactor</i> | The given Bayes factor value |
|--------------------|------------------------------|

Definition at line 114 of file BayesianModelChecker.cpp.

References `bayesFactorThreshold`, `bayesFactorThresholdInverse`, `multiscale::verification::FALSE`, `modelCheckingResult`, and `multiscale::verification::TRUE`.

Referenced by `updateModelCheckingResult()`.

#### 7.13.3.19 void BayesianModelChecker::updateModelCheckingResultNotEnoughTraces ( ) [private]

Update the result of the model checking task considering that not enough traces have been provided.

Definition at line 122 of file BayesianModelChecker.cpp.

References `modelCheckingResult`, and `multiscale::verification::MORE_TRACES_REQUIRED`.

Referenced by `updateModelCheckingResult()`.

#### 7.13.3.20 void BayesianModelChecker::updateTypeIErrorUpperBound ( ) [private]

Update the value of the type I error upper bound.

Definition at line 126 of file BayesianModelChecker.cpp.

References `computeMaximumBinomialPDF()`, `indicatorFunction()`, `multiscale::verification::ModelChecker::totalNumberOfEvaluations`, and `typeIErrorUpperBound`.

Referenced by `updateModelCheckingResult()`.

#### 7.13.3.21 void BayesianModelChecker::validateBayesFactorThreshold ( double *bayesFactorThreshold* ) [private]

Validate the Bayes factor threshold.

The Bayes factor threshold should be greater than 1

##### Parameters

|                             |                            |
|-----------------------------|----------------------------|
| <i>bayesFactorThreshold</i> | The Bayes factor threshold |
|-----------------------------|----------------------------|

Definition at line 76 of file BayesianModelChecker.cpp.

References `ERR_BAYES_FACTOR_THRESHOLD_BEGIN`, `ERR_BAYES_FACTOR_THRESHOLD_END`, `multiscale::Numeric::lessOrEqual()`, `MS_throw`, and `multiscale::StringManipulator::toString()`.

Referenced by `validateInput()`.

7.13.3.22 void BayesianModelChecker::validateInput ( double *alpha*, double *beta*, double *bayesFactorThreshold* )  
[private]

Validate the input parameters  $\alpha$ ,  $\beta$  and the Bayes factor threshold.

$\alpha$  and  $\beta$  should be greater than zero, and Bayes factor threshold should be greater than 1

#### Parameters

|                             |                                                        |
|-----------------------------|--------------------------------------------------------|
| <i>alpha</i>                | The shape parameter $\alpha$ for the Beta distribution |
| <i>beta</i>                 | The shape parameter $\beta$ for the Beta distribution  |
| <i>bayesFactorThreshold</i> | The Bayes factor threshold                             |

Definition at line 56 of file BayesianModelChecker.cpp.

References validateBayesFactorThreshold(), and validateShapeParameters().

Referenced by BayesianModelChecker().

7.13.3.23 void BayesianModelChecker::validateShapeParameters ( double *alpha*, double *beta* ) [private]

Validate the shape parameters  $\alpha$  and  $\beta$ .

$\alpha$  and  $\beta$  should be greater than zero

#### Parameters

|              |                                                        |
|--------------|--------------------------------------------------------|
| <i>alpha</i> | The shape parameter $\alpha$ for the Beta distribution |
| <i>beta</i>  | The shape parameter $\beta$ for the Beta distribution  |

Definition at line 61 of file BayesianModelChecker.cpp.

References ERR\_SHAPE\_PARAMETERS\_BEGIN, ERR\_SHAPE\_PARAMETERS\_END, ERR\_SHAPE\_PARAMETERS\_MIDDLE, isValidShapeParameter(), MS\_throw, and multiscale::StringManipulator::toString().

Referenced by validateInput().

## 7.13.4 Member Data Documentation

7.13.4.1 double multiscale::verification::BayesianModelChecker::alpha [private]

The shape parameter  $\alpha$  for the Beta distribution prior

Definition at line 57 of file BayesianModelChecker.hpp.

Referenced by BayesianModelChecker(), computeBayesFactorValue(), and getDetailedUpdatedResults().

7.13.4.2 double multiscale::verification::BayesianModelChecker::bayesFactorThreshold [private]

The Bayes factor threshold

Definition at line 60 of file BayesianModelChecker.hpp.

Referenced by BayesianModelChecker(), getDetailedUpdatedResults(), initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.13.4.3 double multiscale::verification::BayesianModelChecker::bayesFactorThresholdInverse [private]

The Bayes factor threshold to the power "-1"

Definition at line 61 of file BayesianModelChecker.hpp.

Referenced by indicatorFunction(), initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

#### 7.13.4.4 double multiscale::verification::BayesianModelChecker::beta [private]

The shape parameter  $\beta$  for the Beta distribution prior

Definition at line 58 of file BayesianModelChecker.hpp.

Referenced by BayesianModelChecker(), computeBayesFactorValue(), and getDetailedUpdatedResults().

#### 7.13.4.5 const std::string BayesianModelChecker::ERR\_BAYES\_FACTOR\_THRESHOLD\_BEGIN = "The provided Bayes factor threshold (" [static], [private]

Definition at line 210 of file BayesianModelChecker.hpp.

Referenced by validateBayesFactorThreshold().

#### 7.13.4.6 const std::string BayesianModelChecker::ERR\_BAYES\_FACTOR\_THRESHOLD\_END = ") should be greater than one. Please change." [static], [private]

Definition at line 211 of file BayesianModelChecker.hpp.

Referenced by validateBayesFactorThreshold().

#### 7.13.4.7 const std::string BayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_BEGIN = "The provided Beta distribution shape parameters alpha and beta (" [static], [private]

Definition at line 206 of file BayesianModelChecker.hpp.

Referenced by validateShapeParameters().

#### 7.13.4.8 const std::string BayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_END = ") should be greater than zero. Please change." [static], [private]

Definition at line 208 of file BayesianModelChecker.hpp.

Referenced by validateShapeParameters().

#### 7.13.4.9 const std::string BayesianModelChecker::ERR\_SHAPE\_PARAMETERS\_MIDDLE = ", " [static], [private]

Definition at line 207 of file BayesianModelChecker.hpp.

Referenced by validateShapeParameters().

#### 7.13.4.10 const std::string BayesianModelChecker::ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT = "An invalid Bayesian model checking result was obtained. Please check source code." [static], [private]

Definition at line 204 of file BayesianModelChecker.hpp.

Referenced by doesPropertyHoldConsideringResult().

#### 7.13.4.11 BayesianModelCheckingResult multiscale::verification::BayesianModelChecker::modelCheckingResult [private]

The result of the model checking task

Definition at line 65 of file BayesianModelChecker.hpp.

Referenced by acceptsMoreTraces(), doesPropertyHoldConsideringResult(), getDetailedUpdatedResults(), updateModelCheckingResultEnoughTraces(), and updateModelCheckingResultNotEnoughTraces().

7.13.4.12 `const std::string BayesianModelChecker::MSG_OUTPUT_MORE_TRACES_REQUIRED = "More traces are required to provide a true/false answer assuming the given Beta distribution shape parameters and Bayes factor threshold value. Probabilistic black-box model checking was used instead to provide an answer." [static], [private]`

Definition at line 213 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.13 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_BEGIN = "The provided answer is given for the Beta distribution shape parameters alpha = " [static], [private]`

Definition at line 215 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.14 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_END = "" [static], [private]`

Definition at line 219 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.15 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_MIDDLE1 = " and beta = " [static], [private]`

Definition at line 216 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.16 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_MIDDLE2 = ", and Bayes factor threshold value = " [static], [private]`

Definition at line 217 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.17 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_MIDDLE3 = ". The type I error upper bound for the provided answer is = " [static], [private]`

Definition at line 218 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.18 `const std::string BayesianModelChecker::MSG_OUTPUT_SEPARATOR = " " [static], [private]`

Definition at line 221 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.19 `double multiscale::verification::BayesianModelChecker::probability [private]`

The probability specified by the user for the logic property to be evaluated

Definition at line 54 of file BayesianModelChecker.hpp.

Referenced by computeBayesFactorValue(), computeMaximumBinomialPDF(), and initialise().

#### 7.13.4.20 double multiscale::verification::BayesianModelChecker::typeIErrorUpperBound [private]

The type I error upper bound

Definition at line 63 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults(), initialise(), and updateTypeIErrorUpperBound().

The documentation for this class was generated from the following files:

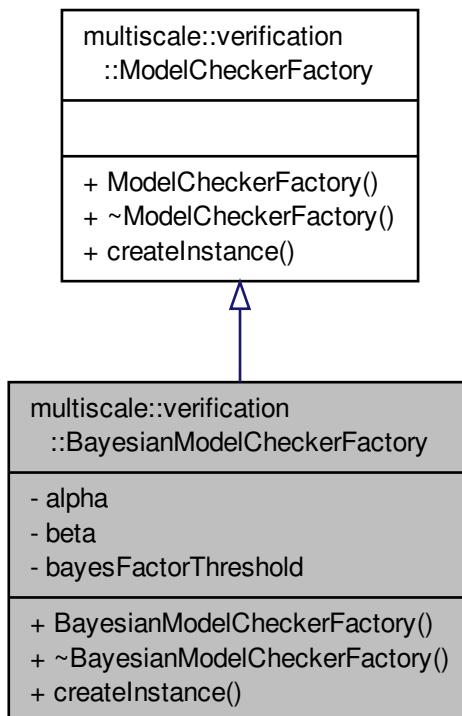
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[BayesianModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[BayesianModelChecker.cpp](#)

## 7.14 multiscale::verification::BayesianModelCheckerFactory Class Reference

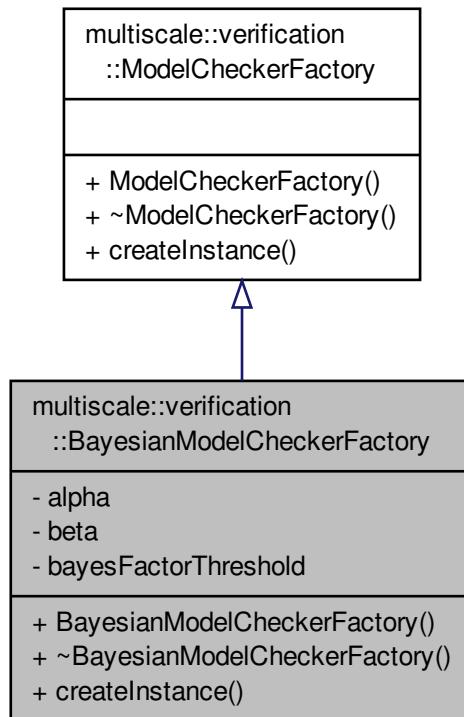
Class for creating [BayesianModelChecker](#) instances.

```
#include <BayesianModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::BayesianModelCheckerFactory:



Collaboration diagram for multiscale::verification::BayesianModelCheckerFactory:



## Public Member Functions

- [BayesianModelCheckerFactory](#) (double `alpha`, double `beta`, double `bayesFactorThreshold`)
- [~BayesianModelCheckerFactory \(\)](#)
- [std::shared\\_ptr< ModelChecker > createInstance](#) (const [AbstractSyntaxTree](#) &`abstractSyntaxTree`, const [TypeSemanticsTable](#) &`typeSemanticsTable`) override

*Create an instance of [BayesianModelChecker](#).*

## Private Attributes

- double `alpha`
- double `beta`
- double `bayesFactorThreshold`

### 7.14.1 Detailed Description

Class for creating [BayesianModelChecker](#) instances.

Definition at line 12 of file BayesianModelCheckerFactory.hpp.

## 7.14.2 Constructor & Destructor Documentation

7.14.2.1 `BayesianModelCheckerFactory::BayesianModelCheckerFactory ( double alpha, double beta, double bayesFactorThreshold )`

Definition at line 7 of file BayesianModelCheckerFactory.cpp.

7.14.2.2 `BayesianModelCheckerFactory::~BayesianModelCheckerFactory ( )`

Definition at line 12 of file BayesianModelCheckerFactory.cpp.

## 7.14.3 Member Function Documentation

7.14.3.1 `std::shared_ptr< ModelChecker > BayesianModelCheckerFactory::createInstance ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [override], [virtual]`

Create an instance of [BayesianModelChecker](#).

### Parameters

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| <code>abstractSyntaxTree</code> | The abstract syntax tree representing the logic property to be checked                |
| <code>typeSemanticsTable</code> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implements [multiscale::verification::ModelCheckerFactory](#).

Definition at line 15 of file BayesianModelCheckerFactory.cpp.

References alpha, bayesFactorThreshold, and beta.

## 7.14.4 Member Data Documentation

7.14.4.1 `double multiscale::verification::BayesianModelCheckerFactory::alpha [private]`

The shape parameter  $\alpha$  for the Beta distribution prior

Definition at line 16 of file BayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

7.14.4.2 `double multiscale::verification::BayesianModelCheckerFactory::bayesFactorThreshold [private]`

The Bayes factor threshold

Definition at line 19 of file BayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

7.14.4.3 `double multiscale::verification::BayesianModelCheckerFactory::beta [private]`

The shape parameter  $\beta$  for the Beta distribution prior

Definition at line 17 of file BayesianModelCheckerFactory.hpp.

Referenced by `createInstance()`.

The documentation for this class was generated from the following files:

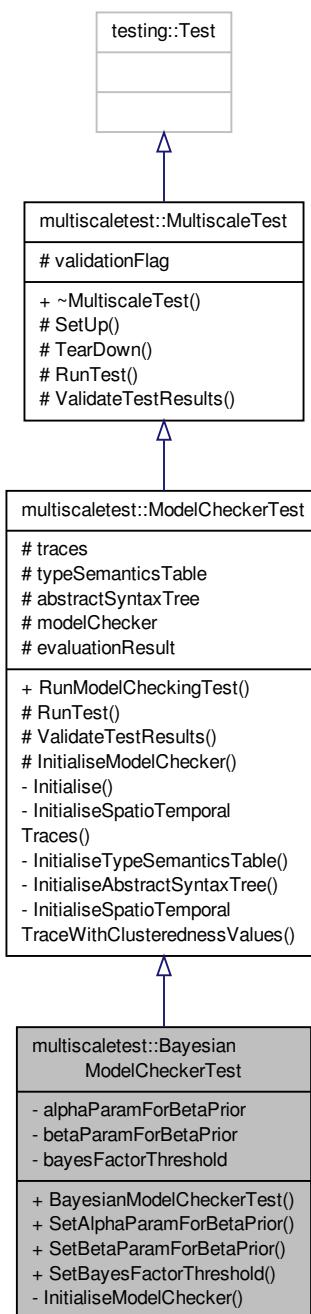
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[BayesianModelCheckerFactory.hpp](#)
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[BayesianModelCheckerFactory.cpp](#)

## 7.15 multiscaletest::BayesianModelCheckerTest Class Reference

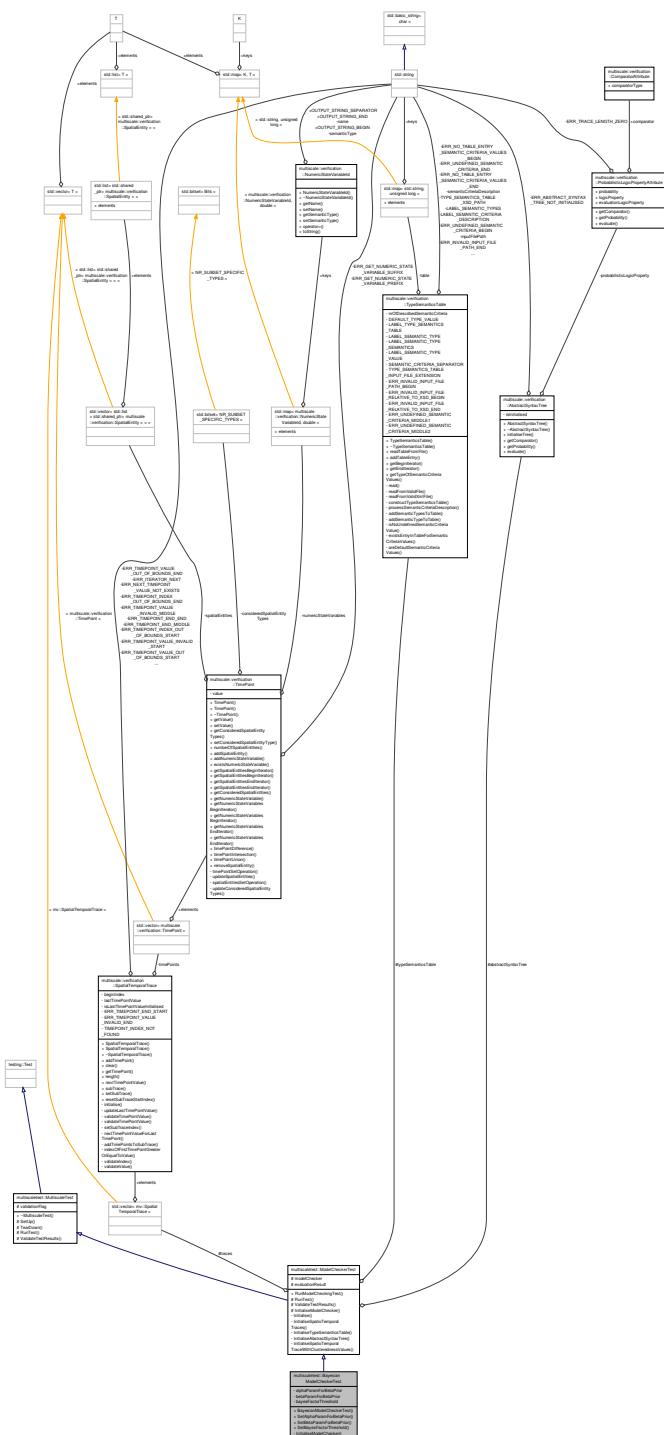
Class for testing the Bayesian model checker.

```
#include <BayesianModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::BayesianModelCheckerTest:



## Collaboration diagram for multiscaletest::BayesianModelCheckerTest:



## Public Member Functions

- `BayesianModelCheckerTest ()`
  - `void SetAlphaParamForBetaPrior (double alphaParamForBetaPrior)`

*Set the value of the alpha parameter for the beta prior.*

- void SetBetaParamForBetaPrior (double betaParamForBetaPrior)

*Set the value of the beta parameter for the beta prior.*

- void [SetBayesFactorThreshold \(double bayesFactorThreshold\)](#)  
*Set the value of the Bayes factor threshold.*

## Private Member Functions

- void [InitialiseModelChecker \(\) override](#)  
*Initialise the model checker.*

## Private Attributes

- double alphaParamForBetaPrior
- double betaParamForBetaPrior
- double bayesFactorThreshold

## Additional Inherited Members

### 7.15.1 Detailed Description

Class for testing the Bayesian model checker.

Definition at line 15 of file BayesianModelCheckerTest.hpp.

### 7.15.2 Constructor & Destructor Documentation

#### 7.15.2.1 multiscaletest::BayesianModelCheckerTest::BayesianModelCheckerTest ( ) [inline]

Definition at line 26 of file BayesianModelCheckerTest.hpp.

### 7.15.3 Member Function Documentation

#### 7.15.3.1 void multiscaletest::BayesianModelCheckerTest::InitialiseModelChecker ( ) [override], [private], [virtual]

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 68 of file BayesianModelCheckerTest.hpp.

#### 7.15.3.2 void multiscaletest::BayesianModelCheckerTest::SetAlphaParamForBetaPrior ( double alphaParamForBetaPrior )

Set the value of the alpha parameter for the beta prior.

##### Parameters

|                                     |                                        |
|-------------------------------------|----------------------------------------|
| <code>alphaParamForBetaPrior</code> | The alpha parameter for the beta prior |
|-------------------------------------|----------------------------------------|

Definition at line 56 of file BayesianModelCheckerTest.hpp.

#### 7.15.3.3 void multiscaletest::BayesianModelCheckerTest::SetBayesFactorThreshold ( double bayesFactorThreshold )

Set the value of the Bayes factor threshold.

**Parameters**

|                             |                                         |
|-----------------------------|-----------------------------------------|
| <i>bayesFactorThreshold</i> | The value of the Bayes factor threshold |
|-----------------------------|-----------------------------------------|

Definition at line 64 of file BayesianModelCheckerTest.hpp.

#### 7.15.3.4 void multiscaletest::BayesianModelCheckerTest::SetBetaParamForBetaPrior ( double *betaParamForBetaPrior* )

Set the value of the beta parameter for the beta prior.

**Parameters**

|                              |                                       |
|------------------------------|---------------------------------------|
| <i>betaParamForBetaPrior</i> | The beta parameter for the beta prior |
|------------------------------|---------------------------------------|

Definition at line 60 of file BayesianModelCheckerTest.hpp.

### 7.15.4 Member Data Documentation

#### 7.15.4.1 double multiscaletest::BayesianModelCheckerTest::alphaParamForBetaPrior [private]

The alpha parameter for the beta prior

Definition at line 19 of file BayesianModelCheckerTest.hpp.

#### 7.15.4.2 double multiscaletest::BayesianModelCheckerTest::bayesFactorThreshold [private]

The considered bayes factor threshold

Definition at line 22 of file BayesianModelCheckerTest.hpp.

#### 7.15.4.3 double multiscaletest::BayesianModelCheckerTest::betaParamForBetaPrior [private]

The beta parameter for the beta prior

Definition at line 20 of file BayesianModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

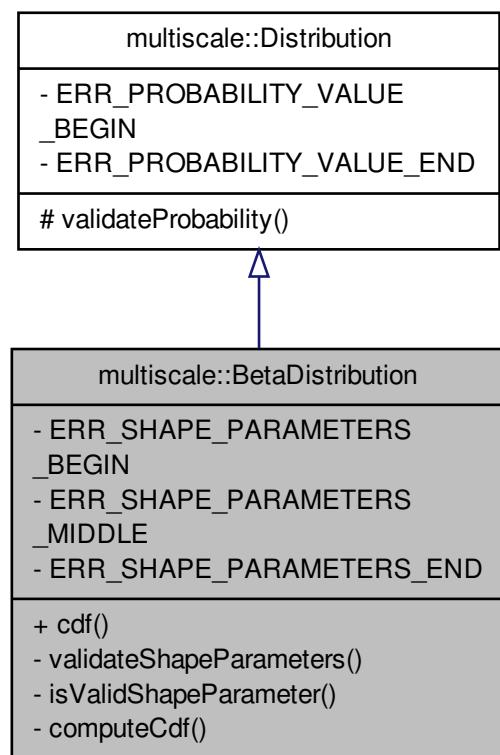
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/BayesianModelCheckerTest.hpp

## 7.16 multiscale::BetaDistribution Class Reference

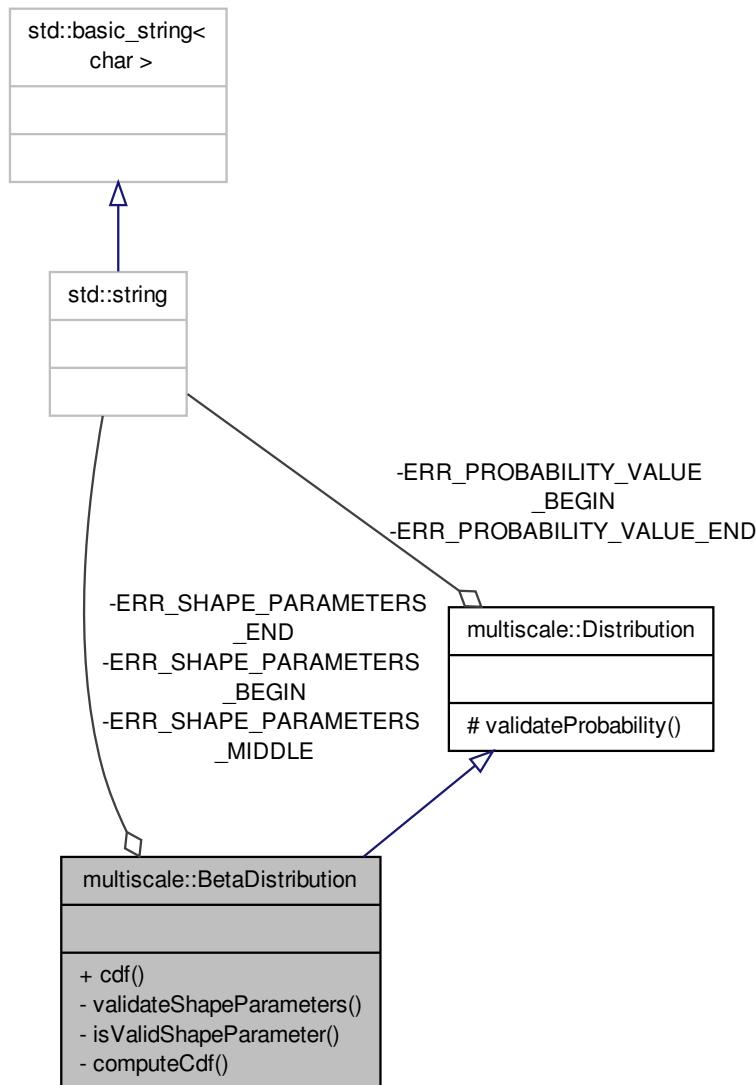
Class for analysing Beta distributed data.

```
#include <BetaDistribution.hpp>
```

Inheritance diagram for multiscale::BetaDistribution:



Collaboration diagram for multiscale::BetaDistribution:



## Static Public Member Functions

- static double [cdf](#) (double alpha, double beta, double probability)  
*Compute the value of the cumulative distribution function (cdf) for a Beta distribution.*

## Static Private Member Functions

- static void [validateShapeParameters](#) (double alpha, double beta)  
*Validate the shape parameters  $\alpha$  and  $\beta$ .*
- static bool [isValidShapeParameter](#) (double shapeParameter)  
*Check if the given shape parameter value is valid.*
- static double [computeCdf](#) (double alpha, double beta, double probability)

*Compute the value of the cumulative distribution function (cdf) for a Beta distribution considering that the parameters are valid.*

## Static Private Attributes

- static const std::string `ERR_SHAPE_PARAMETERS_BEGIN` = "The provided Beta distribution shape parameters alpha and beta ("
- static const std::string `ERR_SHAPE_PARAMETERS_MIDDLE` = ", "
- static const std::string `ERR_SHAPE_PARAMETERS_END` = ") should be greater than zero. Please change."

## Additional Inherited Members

### 7.16.1 Detailed Description

Class for analysing Beta distributed data.

Definition at line 10 of file BetaDistribution.hpp.

### 7.16.2 Member Function Documentation

#### 7.16.2.1 double BetaDistribution::cdf ( double *alpha*, double *beta*, double *probability* ) [static]

Compute the value of the cumulative distribution function (cdf) for a Beta distribution.

The value of the cumulative distribution function (cdf) is computed considering the given probability and shape parameters.

#### Parameters

|                    |                                                                |
|--------------------|----------------------------------------------------------------|
| <i>alpha</i>       | Shape parameter <i>alpha</i>                                   |
| <i>beta</i>        | Shape parameter <i>beta</i>                                    |
| <i>probability</i> | The considered probability when computing the value of the cdf |

Definition at line 10 of file BetaDistribution.cpp.

References `computeCdf()`, `multiscale::Distribution::validateProbability()`, and `validateShapeParameters()`.

Referenced by `multiscale::verification::BayesianModelChecker::computeBayesFactorValue()`, and `TEST()`.

#### 7.16.2.2 double BetaDistribution::computeCdf ( double *alpha*, double *beta*, double *probability* ) [static], [private]

Compute the value of the cumulative distribution function (cdf) for a Beta distribution considering that the parameters are valid.

#### Parameters

|                    |                                                                |
|--------------------|----------------------------------------------------------------|
| <i>alpha</i>       | Shape parameter <i>alpha</i>                                   |
| <i>beta</i>        | Shape parameter <i>beta</i>                                    |
| <i>probability</i> | The considered probability when computing the value of the cdf |

Definition at line 32 of file BetaDistribution.cpp.

Referenced by `cdf()`.

#### 7.16.2.3 bool BetaDistribution::isValidShapeParameter ( double *shapeParameter* ) [static], [private]

Check if the given shape parameter value is valid.

The shape parameter values should be greater than zero

**Parameters**

|                       |                           |
|-----------------------|---------------------------|
| <i>shapeParameter</i> | The given shape parameter |
|-----------------------|---------------------------|

Definition at line 28 of file BetaDistribution.cpp.

Referenced by validateShapeParameters().

**7.16.2.4 void BetaDistribution::validateShapeParameters ( double *alpha*, double *beta* ) [static], [private]**

Validate the shape parameters  $\alpha$  and  $\beta$ .

$\alpha$  and  $\beta$  should be greater than zero

**Parameters**

|              |                                                        |
|--------------|--------------------------------------------------------|
| <i>alpha</i> | The shape parameter $\alpha$ for the Beta distribution |
| <i>beta</i>  | The shape parameter $\beta$ for the Beta distribution  |

Definition at line 17 of file BetaDistribution.cpp.

References ERR\_SHAPE\_PARAMETERS\_BEGIN, ERR\_SHAPE\_PARAMETERS\_END, ERR\_SHAPE\_PARAMETERS\_MIDDLE, isValidShapeParameter(), MS\_throw, and multiscale::StringManipulator::toString().

Referenced by cdf().

**7.16.3 Member Data Documentation****7.16.3.1 const std::string BetaDistribution::ERR\_SHAPE\_PARAMETERS\_BEGIN = "The provided Beta distribution shape parameters alpha and beta (" [static], [private]**

Definition at line 51 of file BetaDistribution.hpp.

Referenced by validateShapeParameters().

**7.16.3.2 const std::string BetaDistribution::ERR\_SHAPE\_PARAMETERS\_END = ") should be greater than zero. Please change." [static], [private]**

Definition at line 53 of file BetaDistribution.hpp.

Referenced by validateShapeParameters().

**7.16.3.3 const std::string BetaDistribution::ERR\_SHAPE\_PARAMETERS\_MIDDLE = "," [static], [private]**

Definition at line 52 of file BetaDistribution.hpp.

Referenced by validateShapeParameters().

The documentation for this class was generated from the following files:

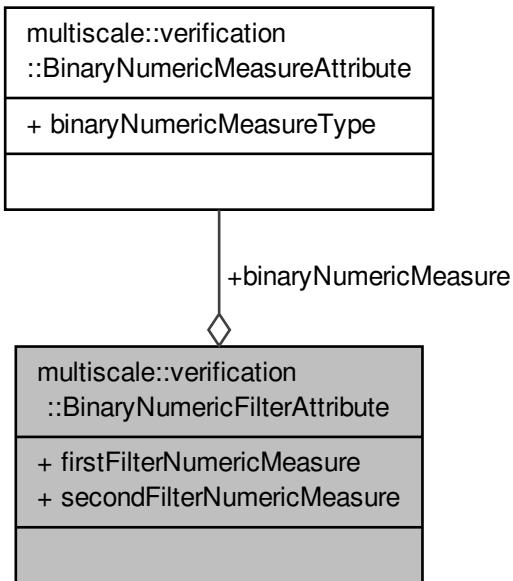
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/[BetaDistribution.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/[BetaDistribution.cpp](#)

**7.17 multiscale::verification::BinaryNumericFilterAttribute Class Reference**

Class for representing a binary numeric filter attribute.

```
#include <BinaryNumericFilterAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryNumericFilterAttribute:



## Public Attributes

- `BinaryNumericMeasureAttribute` `binaryNumericMeasure`
- `FilterNumericMeasureAttributeType` `firstFilterNumericMeasure`
- `FilterNumericMeasureAttributeType` `secondFilterNumericMeasure`

### 7.17.1 Detailed Description

Class for representing a binary numeric filter attribute.

Definition at line 15 of file `BinaryNumericFilterAttribute.hpp`.

### 7.17.2 Member Data Documentation

#### 7.17.2.1 `BinaryNumericMeasureAttribute` `multiscale::verification::BinaryNumericFilterAttribute::binaryNumericMeasure`

The binary numeric measure

Definition at line 20 of file `BinaryNumericFilterAttribute.hpp`.

Referenced by `multiscale::verification::FilterNumericVisitor::operator()()`.

#### 7.17.2.2 `FilterNumericMeasureAttributeType` `multiscale::verification::BinaryNumericFilterAttribute::firstFilterNumericMeasure`

The first filter numeric measure

Definition at line 22 of file `BinaryNumericFilterAttribute.hpp`.

Referenced by multiscale::verification::FilterNumericVisitor::operator()().

### 7.17.2.3 FilterNumericMeasureAttributeType multiscale::verification::BinaryNumericFilterAttribute::secondFilter< NumericMeasure

The second filter numeric measure

Definition at line 24 of file BinaryNumericFilterAttribute.hpp.

Referenced by multiscale::verification::FilterNumericVisitor::operator()().

The documentation for this class was generated from the following file:

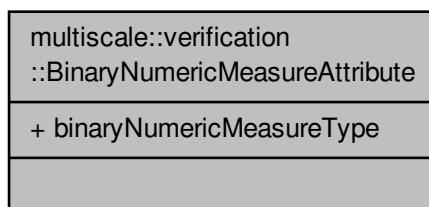
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[BinaryNumericFilterAttribute.hpp](#)

## 7.18 multiscale::verification::BinaryNumericMeasureAttribute Class Reference

Class for representing a binary numeric measure attribute.

```
#include <BinaryNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryNumericMeasureAttribute:



### Public Attributes

- [BinaryNumericMeasureType binaryNumericMeasureType](#)

### 7.18.1 Detailed Description

Class for representing a binary numeric measure attribute.

Definition at line 33 of file BinaryNumericMeasureAttribute.hpp.

### 7.18.2 Member Data Documentation

#### 7.18.2.1 BinaryNumericMeasureType multiscale::verification::BinaryNumericMeasureAttribute::binaryNumericMeasureType

The binary numeric measure type

Definition at line 37 of file BinaryNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::FilterNumericVisitor::operator()().

The documentation for this class was generated from the following file:

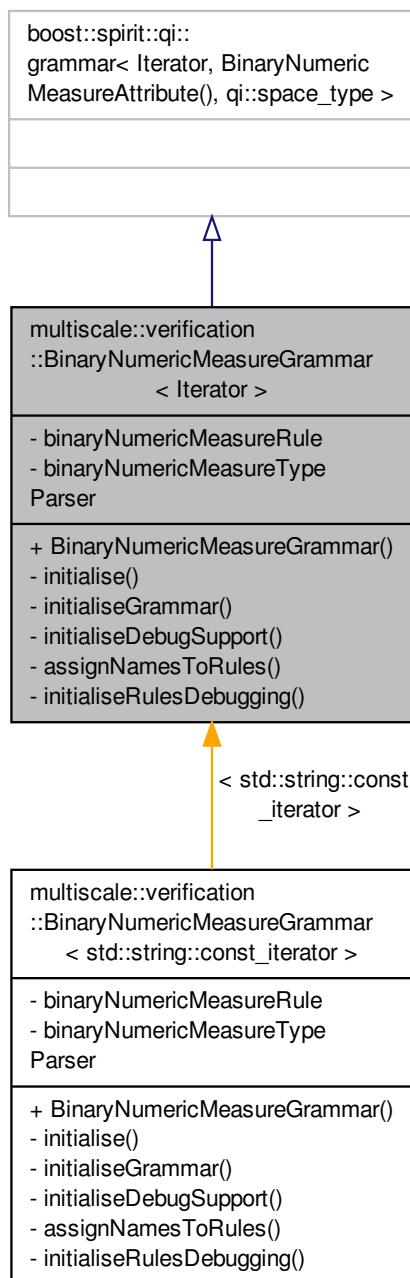
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[BinaryNumericMeasureAttribute.hpp](#)

## 7.19 multiscale::verification::BinaryNumericMeasureGrammar< Iterator > Class Template Reference

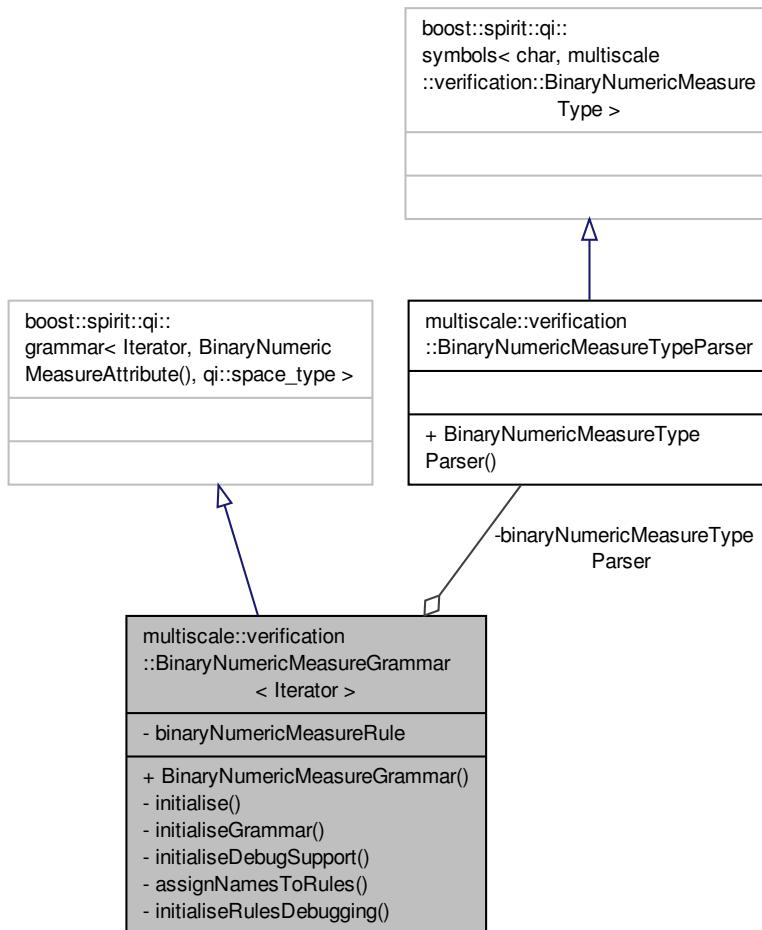
The grammar for parsing binary numeric measure statements.

```
#include <BinaryNumericMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::BinaryNumericMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::BinaryNumericMeasureGrammar< Iterator >:



## Public Member Functions

- [BinaryNumericMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*

## Private Attributes

- qi::rule< Iterator,&br/>    BinaryNumericMeasureAttribute(),  
    qi::space\_type > binaryNumericMeasureRule
- BinaryNumericMeasureTypeParser binaryNumericMeasureTypeParser

### 7.19.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::BinaryNumericMeasureGrammar< Iterator >
```

The grammar for parsing binary numeric measure statements.

Definition at line 30 of file BinaryNumericMeasureGrammar.hpp.

### 7.19.2 Constructor & Destructor Documentation

7.19.2.1 template<typename Iterator > multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::BinaryNumericMeasureGrammar( )

Definition at line 23 of file BinaryNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::initialise().

### 7.19.3 Member Function Documentation

7.19.3.1 template<typename Iterator > void multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::assignNamesToRules( ) [private]

Assign names to the rules.

Definition at line 56 of file BinaryNumericMeasureGrammarDefinition.hpp.

7.19.3.2 template<typename Iterator > void multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::initialise( ) [private]

Initialisation function.

Definition at line 33 of file BinaryNumericMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::BinaryNumericMeasureGrammar().

7.19.3.3 template<typename Iterator > void multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::initialiseDebugSupport( ) [private]

Initialise debug support.

Definition at line 47 of file BinaryNumericMeasureGrammarDefinition.hpp.

7.19.3.4 template<typename Iterator > void multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 40 of file BinaryNumericMeasureGrammarDefinition.hpp.

---

7.19.3.5 `template<typename Iterator > void multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]`

Initialise the debugging of rules.

Definition at line 62 of file `BinaryNumericMeasureGrammarDefinition.hpp`.

#### 7.19.4 Member Data Documentation

7.19.4.1 `template<typename Iterator> qi::rule<Iterator, BinaryNumericMeasureAttribute(), qi::space_type> multiscale::verification::BinaryNumericMeasureGrammar< Iterator >::binaryNumericMeasureRule [private]`

The rule for parsing a binary numeric measure

Definition at line 36 of file `BinaryNumericMeasureGrammar.hpp`.

7.19.4.2 `template<typename Iterator> BinaryNumericMeasureTypeParser multiscale::verification< ::BinaryNumericMeasureGrammar< Iterator >::binaryNumericMeasureTypeParser [private]`

The binary numeric measure type parser

Definition at line 41 of file `BinaryNumericMeasureGrammar.hpp`.

The documentation for this class was generated from the following files:

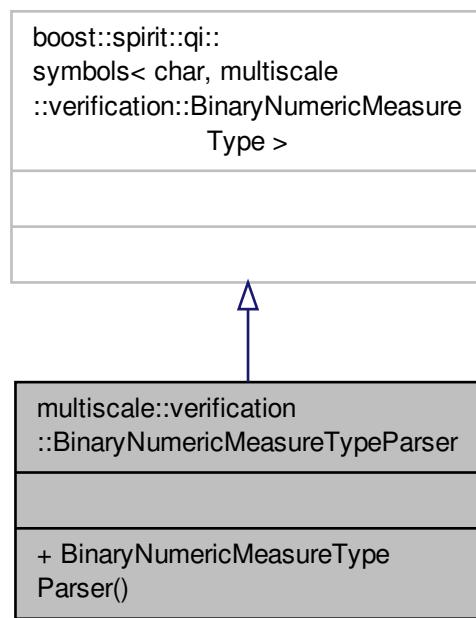
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryNumericMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryNumericMeasureGrammarDefinition.hpp](#)

## 7.20 `multiscale::verification::BinaryNumericMeasureTypeParser` Struct Reference

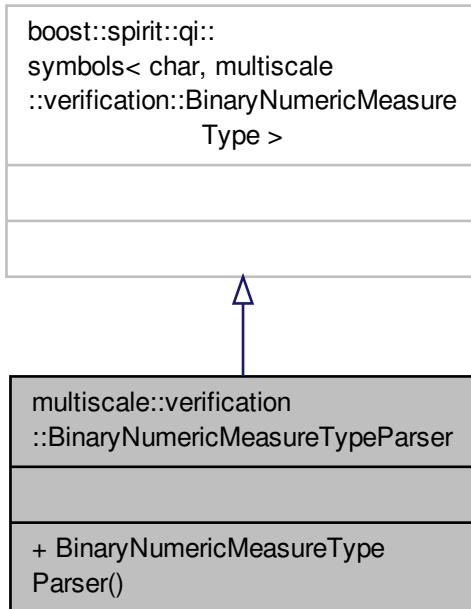
Symbol table and parser for the binary numeric measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::BinaryNumericMeasureTypeParser:



Collaboration diagram for multiscale::verification::BinaryNumericMeasureTypeParser:



## Public Member Functions

- [BinaryNumericMeasureTypeParser \(\)](#)

### 7.20.1 Detailed Description

Symbol table and parser for the binary numeric measure type.

Definition at line 27 of file [SymbolTables.hpp](#).

### 7.20.2 Constructor & Destructor Documentation

#### 7.20.2.1 multiscale::verification::BinaryNumericMeasureTypeParser::BinaryNumericMeasureTypeParser ( ) [inline]

Definition at line 30 of file [SymbolTables.hpp](#).

References [multiscale::verification::Add](#), [multiscale::verification::Div](#), [multiscale::verification::Log](#), [multiscale::verification::Mod](#), [multiscale::verification::Multiply](#), [multiscale::verification::Power](#), and [multiscale::verification::Subtract](#).

The documentation for this struct was generated from the following file:

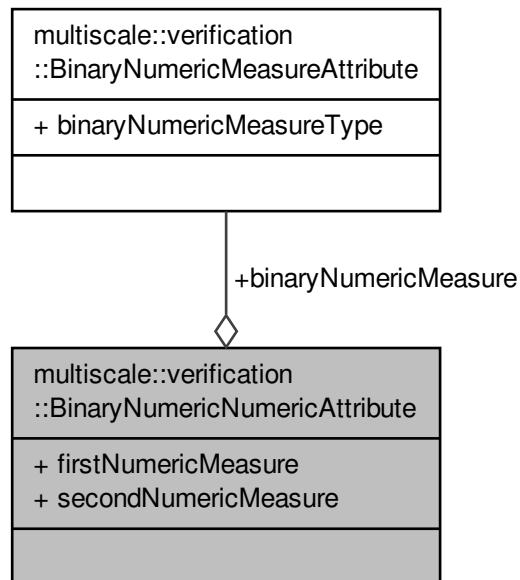
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.21 multiscale::verification::BinaryNumericNumericAttribute Class Reference

Class for representing a binary numeric numeric measure attribute.

```
#include <BinaryNumericNumericAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryNumericNumericAttribute:



### Public Attributes

- `BinaryNumericMeasureAttribute binaryNumericMeasure`
- `NumericMeasureType firstNumericMeasure`
- `NumericMeasureType secondNumericMeasure`

#### 7.21.1 Detailed Description

Class for representing a binary numeric numeric measure attribute.

Definition at line 15 of file `BinaryNumericNumericAttribute.hpp`.

#### 7.21.2 Member Data Documentation

##### 7.21.2.1 `BinaryNumericMeasureAttribute multiscale::verification::BinaryNumericNumericAttribute::binaryNumericMeasure`

The binary numeric measure

Definition at line 19 of file `BinaryNumericNumericAttribute.hpp`.

### 7.21.2.2 NumericMeasureType multiscale::verification::BinaryNumericNumericAttribute::firstNumericMeasure

The first numeric measure

Definition at line 20 of file `BinaryNumericNumericAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

### 7.21.2.3 NumericMeasureType multiscale::verification::BinaryNumericNumericAttribute::secondNumericMeasure

The second numeric measure

Definition at line 21 of file `BinaryNumericNumericAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

The documentation for this class was generated from the following file:

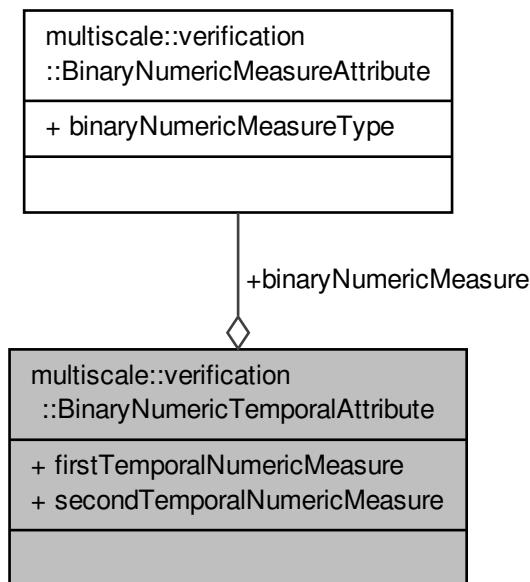
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericNumericAttribute.hpp`

## 7.22 multiscale::verification::BinaryNumericTemporalAttribute Class Reference

Class for representing a binary numeric temporal measure attribute.

```
#include <BinaryNumericTemporalAttribute.hpp>
```

Collaboration diagram for `multiscale::verification::BinaryNumericTemporalAttribute`:



### Public Attributes

- `BinaryNumericMeasureAttribute binaryNumericMeasure`

- [TemporalNumericMeasureType firstTemporalNumericMeasure](#)
- [TemporalNumericMeasureType secondTemporalNumericMeasure](#)

### 7.22.1 Detailed Description

Class for representing a binary numeric temporal measure attribute.

Definition at line 15 of file [BinaryNumericTemporalAttribute.hpp](#).

### 7.22.2 Member Data Documentation

#### 7.22.2.1 [BinaryNumericMeasureAttribute multiscale::verification::BinaryNumericTemporalAttribute::binaryNumericMeasure](#)

The binary numeric measure

Definition at line 20 of file [BinaryNumericTemporalAttribute.hpp](#).

#### 7.22.2.2 [TemporalNumericMeasureType multiscale::verification::BinaryNumericTemporalAttribute::firstTemporalNumericMeasure](#)

The first temporal numeric measure

Definition at line 22 of file [BinaryNumericTemporalAttribute.hpp](#).

Referenced by [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#).

#### 7.22.2.3 [TemporalNumericMeasureType multiscale::verification::BinaryNumericTemporalAttribute::secondTemporalNumericMeasure](#)

The second temporal numeric measure

Definition at line 24 of file [BinaryNumericTemporalAttribute.hpp](#).

Referenced by [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

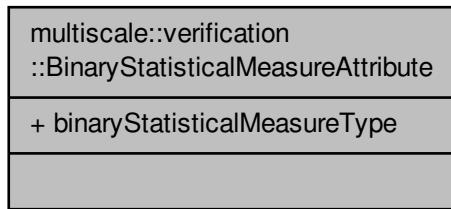
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericTemporalAttribute.hpp](#)

## 7.23 multiscale::verification::BinaryStatisticalMeasureAttribute Class Reference

Class for representing a binary statistical measure attribute.

```
#include <BinaryStatisticalMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalMeasureAttribute:



## Public Attributes

- [BinaryStatisticalMeasureType binaryStatisticalMeasureType](#)

### 7.23.1 Detailed Description

Class for representing a binary statistical measure attribute.

Definition at line 28 of file [BinaryStatisticalMeasureAttribute.hpp](#).

### 7.23.2 Member Data Documentation

#### 7.23.2.1 [BinaryStatisticalMeasureType multiscale::verification::BinaryStatisticalMeasureAttribute::binaryStatisticalMeasureType](#)

The binary statistical measure type

Definition at line 32 of file [BinaryStatisticalMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::NumericVisitor::operator\(\)\(\)](#), and [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

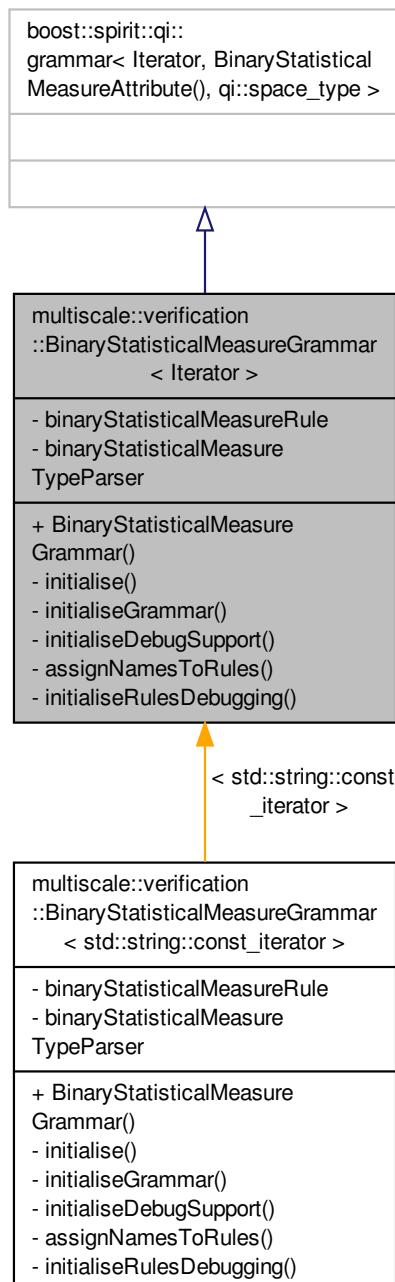
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp](#)

## 7.24 [multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >](#) Class Template Reference

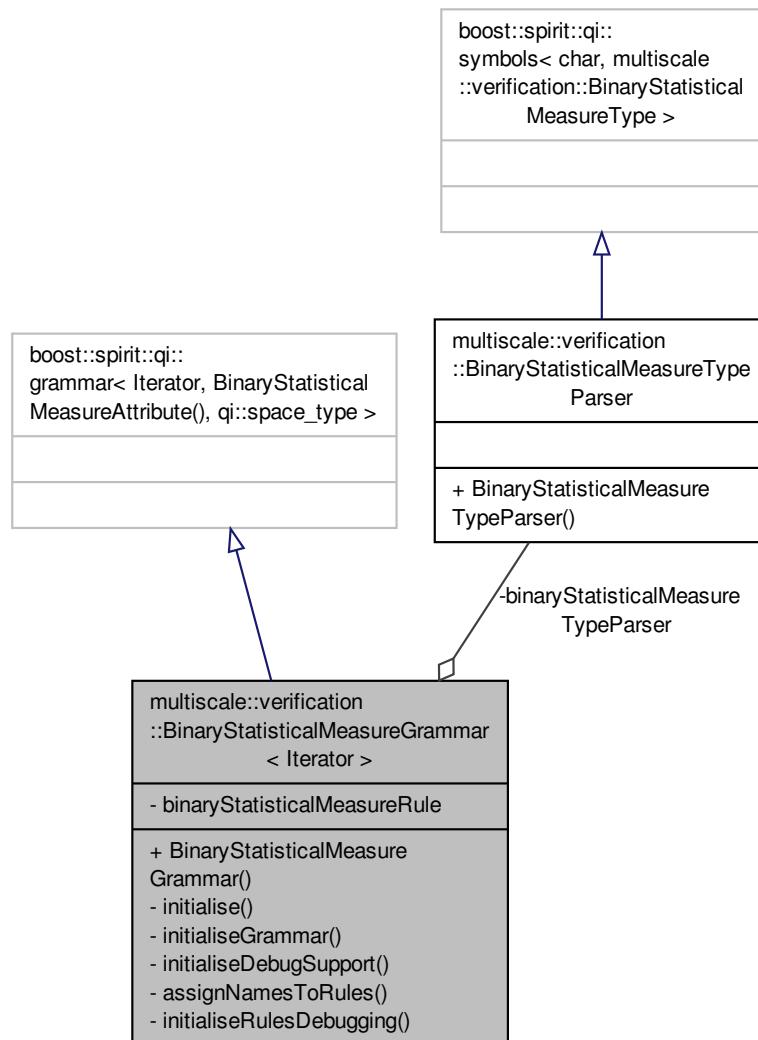
The grammar for parsing binary statistical measure statements.

```
#include <BinaryStatisticalMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >:



## Public Member Functions

- `BinaryStatisticalMeasureGrammar ()`

## Private Member Functions

- `void initialise ()`  
*Initialisation function.*
- `void initialiseGrammar ()`  
*Initialise the grammar.*
- `void initialiseDebugSupport ()`  
*Initialise debug support.*
- `void assignNamesToRules ()`  
*Assign names to the rules.*

- void [initialiseRulesDebugging \(\)](#)

*Initialise the debugging of rules.*

## Private Attributes

- qi::rule< Iterator,  
[BinaryStatisticalMeasureAttribute\(\)](#),  
 qi::space\_type > [binaryStatisticalMeasureRule](#)
- [BinaryStatisticalMeasureTypeParser](#) [binaryStatisticalMeasureTypeParser](#)

### 7.24.1 Detailed Description

`template<typename Iterator>class multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >`

The grammar for parsing binary statistical measure statements.

Definition at line 30 of file `BinaryStatisticalMeasureGrammar.hpp`.

### 7.24.2 Constructor & Destructor Documentation

7.24.2.1 `template<typename Iterator > multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::BinaryStatisticalMeasureGrammar ( )`

Definition at line 23 of file `BinaryStatisticalMeasureGrammarDefinition.hpp`.

References `multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::initialise()`.

### 7.24.3 Member Function Documentation

7.24.3.1 `template<typename Iterator > void multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::assignNamesToRules ( ) [private]`

Assign names to the rules.

Definition at line 56 of file `BinaryStatisticalMeasureGrammarDefinition.hpp`.

7.24.3.2 `template<typename Iterator > void multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::initialise ( ) [private]`

Initialisation function.

Definition at line 33 of file `BinaryStatisticalMeasureGrammarDefinition.hpp`.

Referenced by `multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::BinaryStatisticalMeasureGrammar()`.

7.24.3.3 `template<typename Iterator > void multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::initialiseDebugSupport ( ) [private]`

Initialise debug support.

Definition at line 47 of file `BinaryStatisticalMeasureGrammarDefinition.hpp`.

---

7.24.3.4 template<typename Iterator > void multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 40 of file BinaryStatisticalMeasureGrammarDefinition.hpp.

7.24.3.5 template<typename Iterator > void multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 62 of file BinaryStatisticalMeasureGrammarDefinition.hpp.

#### 7.24.4 Member Data Documentation

7.24.4.1 template<typename Iterator> qi::rule<Iterator, BinaryStatisticalMeasureAttribute(), qi::space\_type> multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::binaryStatisticalMeasureRule [private]

The rule for parsing a binary statistical measure

Definition at line 36 of file BinaryStatisticalMeasureGrammar.hpp.

7.24.4.2 template<typename Iterator> BinaryStatisticalMeasureTypeParser multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >::binaryStatisticalMeasureTypeParser [private]

The binary statistical measure type parser

Definition at line 42 of file BinaryStatisticalMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

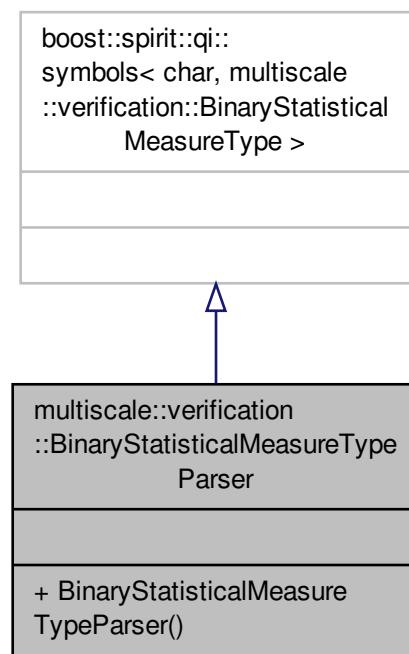
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryStatisticalMeasureGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryStatisticalMeasureGrammarDefinition.hpp](#)

### 7.25 multiscale::verification::BinaryStatisticalMeasureTypeParser Struct Reference

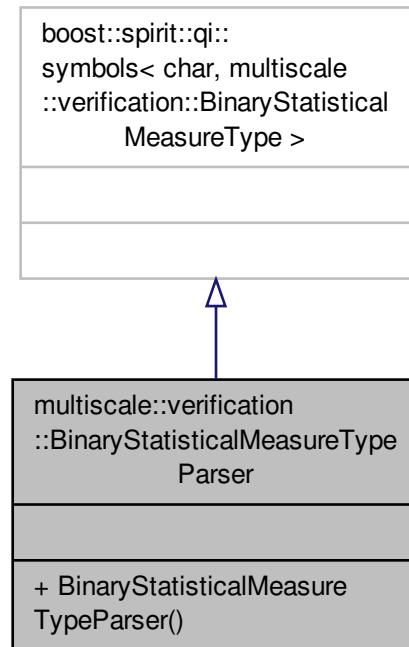
Symbol table and parser for the binary statistical measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::BinaryStatisticalMeasureTypeParser:



Collaboration diagram for multiscale::verification::BinaryStatisticalMeasureTypeParser:



## Public Member Functions

- [BinaryStatisticalMeasureTypeParser \(\)](#)

### 7.25.1 Detailed Description

Symbol table and parser for the binary statistical measure type.

Definition at line 45 of file [SymbolTables.hpp](#).

### 7.25.2 Constructor & Destructor Documentation

#### 7.25.2.1 multiscale::verification::BinaryStatisticalMeasureTypeParser::BinaryStatisticalMeasureTypeParser( ) [inline]

Definition at line 48 of file [SymbolTables.hpp](#).

References [multiscale::verification::Covar](#).

The documentation for this struct was generated from the following file:

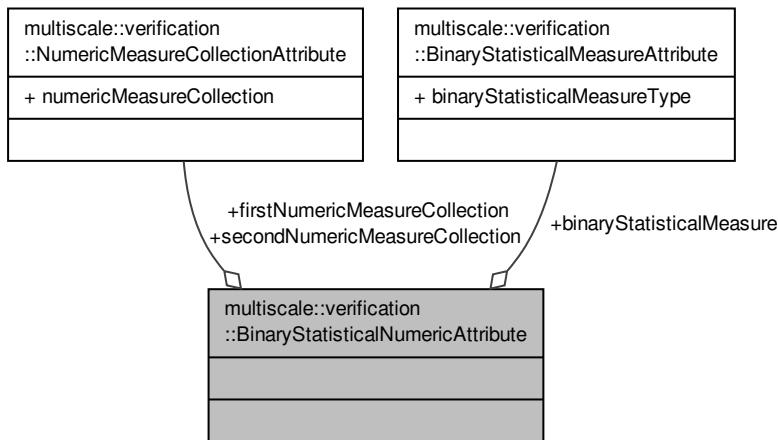
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.hpp](#)

## 7.26 multiscale::verification::BinaryStatisticalNumericAttribute Class Reference

Class for representing a binary statistical numeric attribute.

```
#include <BinaryStatisticalNumericAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalNumericAttribute:



### Public Attributes

- [BinaryStatisticalMeasureAttribute binaryStatisticalMeasure](#)
- [NumericMeasureCollectionAttribute firstNumericMeasureCollection](#)
- [NumericMeasureCollectionAttribute secondNumericMeasureCollection](#)

#### 7.26.1 Detailed Description

Class for representing a binary statistical numeric attribute.

Definition at line 15 of file [BinaryStatisticalNumericAttribute.hpp](#).

#### 7.26.2 Member Data Documentation

##### 7.26.2.1 BinaryStatisticalMeasureAttribute multiscale::verification::BinaryStatisticalNumericAttribute::binaryStatisticalMeasure

The binary statistical subset measure

Definition at line 20 of file [BinaryStatisticalNumericAttribute.hpp](#).

Referenced by [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#).

##### 7.26.2.2 NumericMeasureCollectionAttribute multiscale::verification::BinaryStatisticalNumericAttribute::firstNumericMeasureCollection

The first considered numeric measure collection

Definition at line 22 of file [BinaryStatisticalNumericAttribute.hpp](#).

Referenced by multiscale::verification::TemporalNumericVisitor::operator()().

### 7.26.2.3 NumericMeasureCollectionAttribute multiscale::verification::BinaryStatisticalNumericAttribute::second ← NumericMeasureCollection

The second considered numeric measure collection

Definition at line 24 of file BinaryStatisticalNumericAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()().

The documentation for this class was generated from the following file:

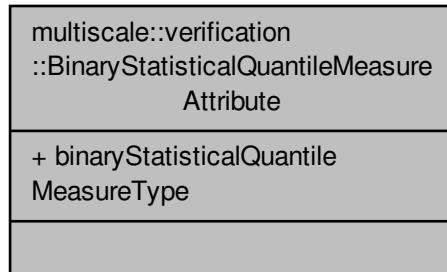
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalNumericAttribute.hpp

## 7.27 multiscale::verification::BinaryStatisticalQuantileMeasureAttribute Class Reference

Class for representing a binary statistical quantile measure attribute.

```
#include <BinaryStatisticalQuantileMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalQuantileMeasureAttribute:



### Public Attributes

- [BinaryStatisticalQuantileMeasureType binaryStatisticalQuantileMeasureType](#)

#### 7.27.1 Detailed Description

Class for representing a binary statistical quantile measure attribute.

Definition at line 29 of file BinaryStatisticalQuantileMeasureAttribute.hpp.

#### 7.27.2 Member Data Documentation

7.27.2.1 **BinaryStatisticalQuantileMeasureType** multiscale::verification::BinaryStatisticalQuantileMeasureAttribute  
  ::binaryStatisticalQuantileMeasureType

The binary statistical quantile measure type

Definition at line 34 of file BinaryStatisticalQuantileMeasureAttribute.hpp.

The documentation for this class was generated from the following file:

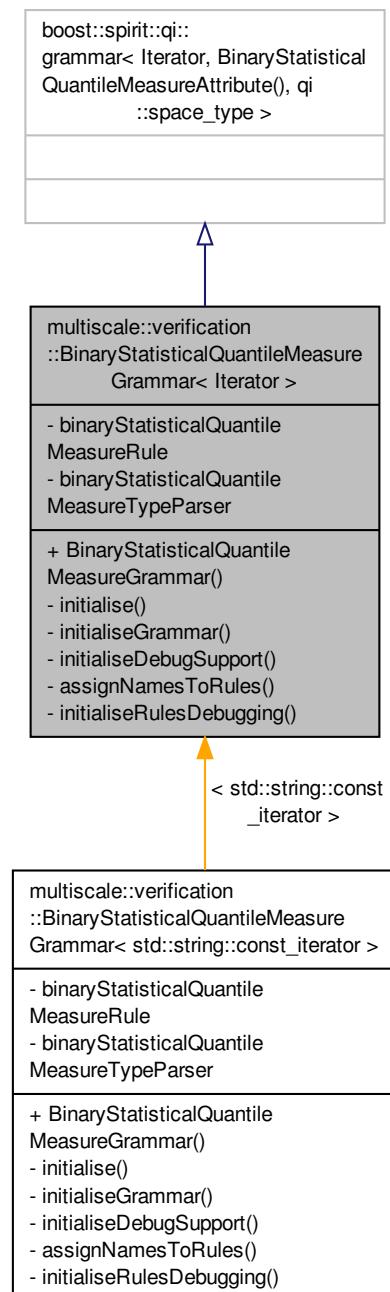
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp

## 7.28 multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator > Class Template Reference

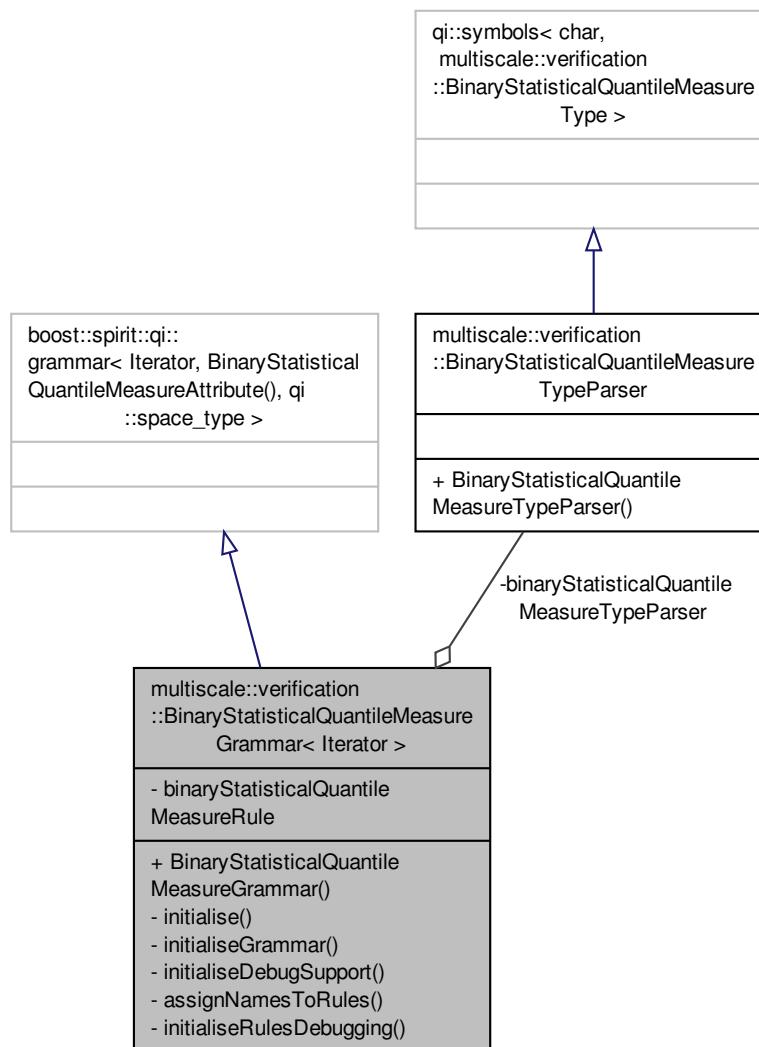
The grammar for parsing binary statistical quantile measure statements.

```
#include <BinaryStatisticalQuantileMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >:



## Public Member Functions

- [BinaryStatisticalQuantileMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)

- `Assign names to the rules.`
- void `initialiseRulesDebugging ()`  
*Initialise the debugging of rules.*

## Private Attributes

- `qi::rule< Iterator,`  
`BinaryStatisticalQuantileMeasureAttribute(),`  
`qi::space_type > binaryStatisticalQuantileMeasureRule`
- `BinaryStatisticalQuantileMeasureTypeParser binaryStatisticalQuantileMeasureTypeParser`

### 7.28.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >
```

The grammar for parsing binary statistical quantile measure statements.

Definition at line 30 of file `BinaryStatisticalQuantileMeasureGrammar.hpp`.

### 7.28.2 Constructor & Destructor Documentation

```
7.28.2.1 template<typename Iterator > multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::BinaryStatisticalQuantileMeasureGrammar ()
```

Definition at line 23 of file `BinaryStatisticalQuantileMeasureGrammarDefinition.hpp`.

References `multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::initialise()`.

### 7.28.3 Member Function Documentation

```
7.28.3.1 template<typename Iterator > void multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::assignNamesToRules () [private]
```

Assign names to the rules.

Definition at line 56 of file `BinaryStatisticalQuantileMeasureGrammarDefinition.hpp`.

```
7.28.3.2 template<typename Iterator > void multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::initialise () [private]
```

Initialisation function.

Definition at line 33 of file `BinaryStatisticalQuantileMeasureGrammarDefinition.hpp`.

Referenced by `multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::BinaryStatisticalQuantileMeasureGrammar()`.

```
7.28.3.3 template<typename Iterator > void multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::initialiseDebugSupport () [private]
```

Initialise debug support.

Definition at line 47 of file `BinaryStatisticalQuantileMeasureGrammarDefinition.hpp`.

7.28.3.4 template<typename Iterator > void multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 40 of file BinaryStatisticalQuantileMeasureGrammarDefinition.hpp.

7.28.3.5 template<typename Iterator > void multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 62 of file BinaryStatisticalQuantileMeasureGrammarDefinition.hpp.

## 7.28.4 Member Data Documentation

7.28.4.1 template<typename Iterator> qi::rule<Iterator, BinaryStatisticalQuantileMeasureAttribute(), qi::space\_type> multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::binaryStatisticalQuantileMeasureRule [private]

The rule for parsing a binary statistical quantile measure

Definition at line 36 of file BinaryStatisticalQuantileMeasureGrammar.hpp.

7.28.4.2 template<typename Iterator> BinaryStatisticalQuantileMeasureTypeParser multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >::binaryStatisticalQuantileMeasureTypeParser [private]

The binary statistical quantile measure type parser

Definition at line 42 of file BinaryStatisticalQuantileMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

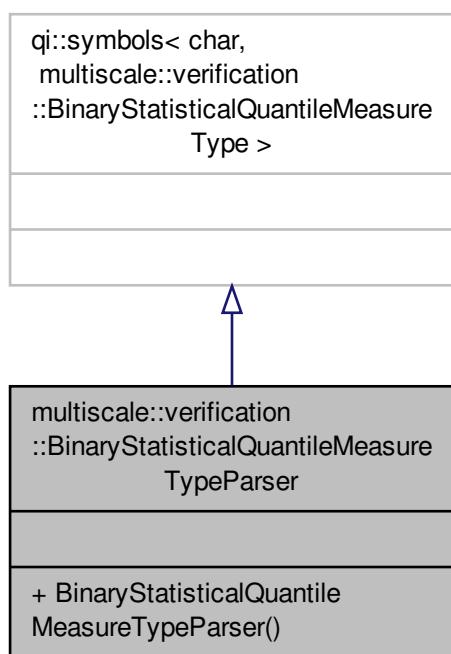
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryStatisticalQuantileMeasureGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[BinaryStatisticalQuantileMeasureGrammarDefinition.hpp](#)

## 7.29 multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser Struct Reference

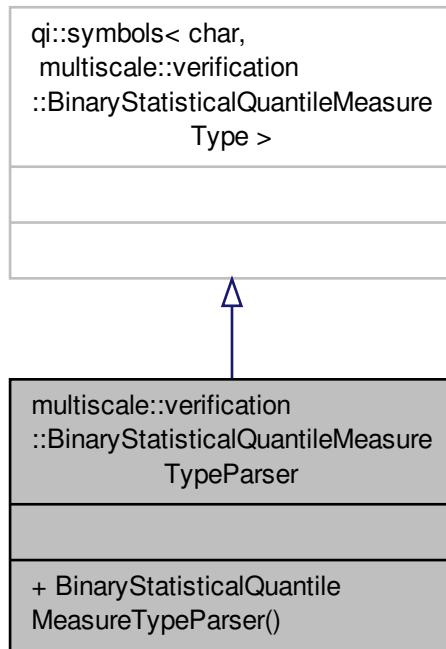
Symbol table and parser for the binary statistical quantile measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser:



Collaboration diagram for multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser:



## Public Member Functions

- [BinaryStatisticalQuantileMeasureTypeParser \(\)](#)

### 7.29.1 Detailed Description

Symbol table and parser for the binary statistical quantile measure type.

Definition at line 57 of file [SymbolTables.hpp](#).

### 7.29.2 Constructor & Destructor Documentation

**7.29.2.1 multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser::BinaryStatisticalQuantileMeasureTypeParser ( ) [inline]**

Definition at line 60 of file [SymbolTables.hpp](#).

References [multiscale::verification::Percentile](#), and [multiscale::verification::Quartile](#).

The documentation for this struct was generated from the following file:

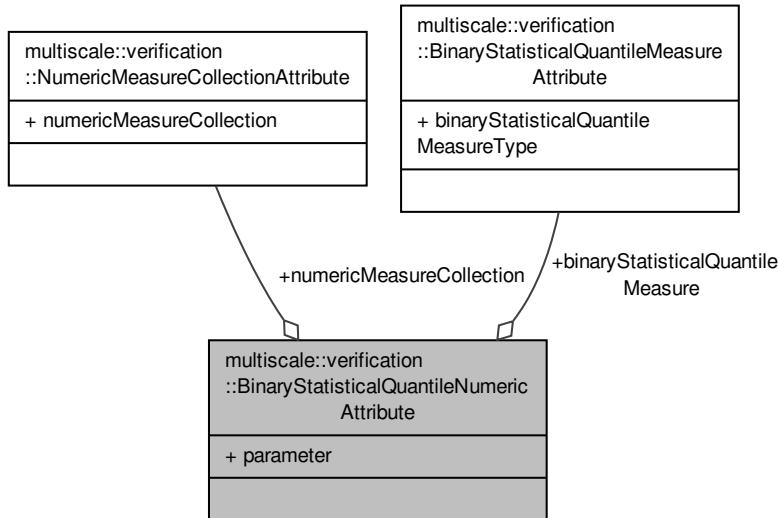
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.30 multiscale::verification::BinaryStatisticalQuantileNumericAttribute Class Reference

Class for representing a binary statistical quantile numeric attribute.

```
#include <BinaryStatisticalQuantileNumericAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalQuantileNumericAttribute:



### Public Attributes

- `BinaryStatisticalQuantileMeasureAttribute binaryStatisticalQuantileMeasure`
- `NumericMeasureCollectionAttribute numericMeasureCollection`
- `double parameter`

#### 7.30.1 Detailed Description

Class for representing a binary statistical quantile numeric attribute.

Definition at line 15 of file `BinaryStatisticalQuantileNumericAttribute.hpp`.

#### 7.30.2 Member Data Documentation

##### 7.30.2.1 `BinaryStatisticalQuantileMeasureAttribute multiscale::verification::BinaryStatisticalQuantileNumericAttribute::binaryStatisticalQuantileMeasure`

The binary statistical quantile measure

Definition at line 20 of file `BinaryStatisticalQuantileNumericAttribute.hpp`.

Referenced by `multiscale::verification::TemporalNumericVisitor::operator()()`.

### 7.30.2.2 NumericMeasureCollectionAttribute multiscale::verification::BinaryStatisticalQuantileNumericAttribute<-->::numericMeasureCollection

The considered numeric measure collection

Definition at line 22 of file BinaryStatisticalQuantileNumericAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()().

### 7.30.2.3 double multiscale::verification::BinaryStatisticalQuantileNumericAttribute::parameter

The considered parameter

Definition at line 24 of file BinaryStatisticalQuantileNumericAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()().

The documentation for this class was generated from the following file:

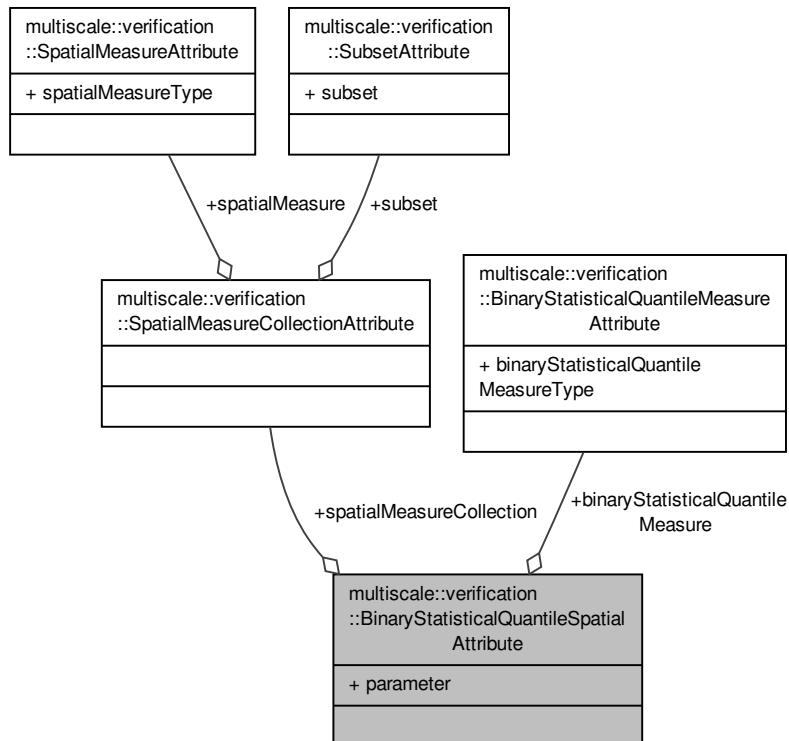
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileNumericAttribute.hpp

## 7.31 multiscale::verification::BinaryStatisticalQuantileSpatialAttribute Class Reference

Class for representing a binary statistical quantile spatial attribute.

```
#include <BinaryStatisticalQuantileSpatialAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalQuantileSpatialAttribute:



## Public Attributes

- `BinaryStatisticalQuantileMeasureAttribute binaryStatisticalQuantileMeasure`
- `SpatialMeasureCollectionAttribute spatialMeasureCollection`
- `double parameter`

### 7.31.1 Detailed Description

Class for representing a binary statistical quantile spatial attribute.

Definition at line 15 of file `BinaryStatisticalQuantileSpatialAttribute.hpp`.

### 7.31.2 Member Data Documentation

#### 7.31.2.1 `BinaryStatisticalQuantileMeasureAttribute multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::binaryStatisticalQuantileMeasure`

The binary statistical quantile measure

Definition at line 20 of file `BinaryStatisticalQuantileSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

#### 7.31.2.2 `double multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::parameter`

The considered parameter

Definition at line 24 of file `BinaryStatisticalQuantileSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

#### 7.31.2.3 `SpatialMeasureCollectionAttribute multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::spatialMeasureCollection`

The considered spatial measure collection

Definition at line 22 of file `BinaryStatisticalQuantileSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

The documentation for this class was generated from the following file:

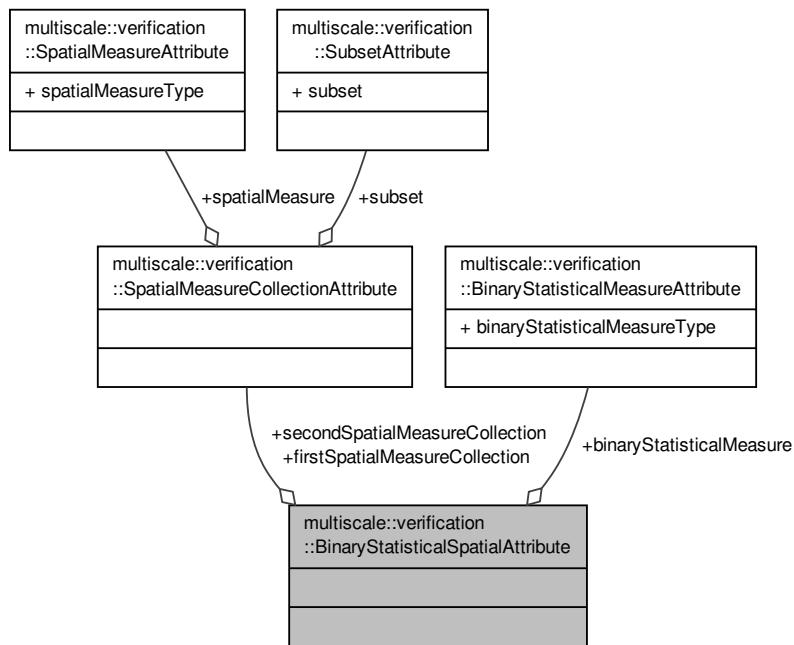
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileSpatialAttribute.hpp`

## 7.32 `multiscale::verification::BinaryStatisticalSpatialAttribute` Class Reference

Class for representing a binary statistical spatial attribute.

```
#include <BinaryStatisticalSpatialAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinaryStatisticalSpatialAttribute:



## Public Attributes

- `BinaryStatisticalMeasureAttribute binaryStatisticalMeasure`
- `SpatialMeasureCollectionAttribute firstSpatialMeasureCollection`
- `SpatialMeasureCollectionAttribute secondSpatialMeasureCollection`

### 7.32.1 Detailed Description

Class for representing a binary statistical spatial attribute.

Definition at line 15 of file `BinaryStatisticalSpatialAttribute.hpp`.

### 7.32.2 Member Data Documentation

#### 7.32.2.1 `BinaryStatisticalMeasureAttribute multiscale::verification::BinaryStatisticalSpatialAttribute::binaryStatisticalMeasure`

The binary statistical subset measure

Definition at line 20 of file `BinaryStatisticalSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

#### 7.32.2.2 `SpatialMeasureCollectionAttribute multiscale::verification::BinaryStatisticalSpatialAttribute::firstSpatialMeasureCollection`

The first considered spatial measure collection

Definition at line 22 of file `BinaryStatisticalSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

### 7.32.2.3 `SpatialMeasureCollectionAttribute multiscale::verification::BinaryStatisticalSpatialAttribute::secondSpatialMeasureCollection`

The second considered spatial measure collection

Definition at line 24 of file `BinaryStatisticalSpatialAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

The documentation for this class was generated from the following file:

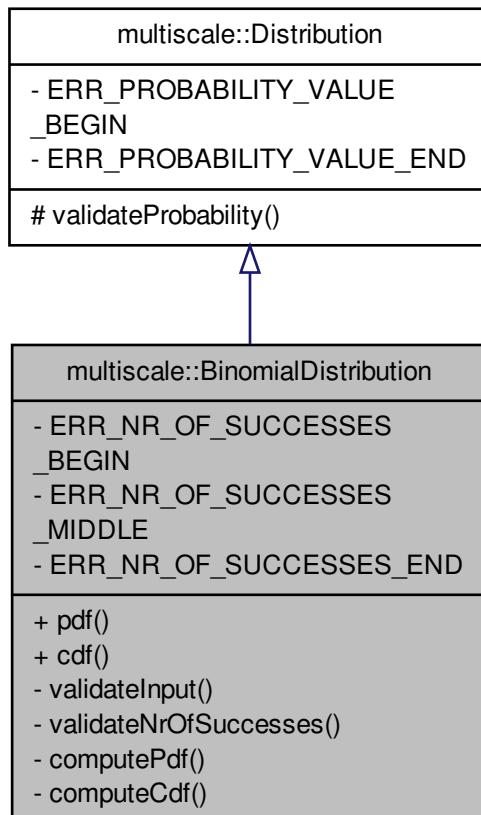
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalSpatialAttribute.hpp`

## 7.33 `multiscale::BinomialDistribution` Class Reference

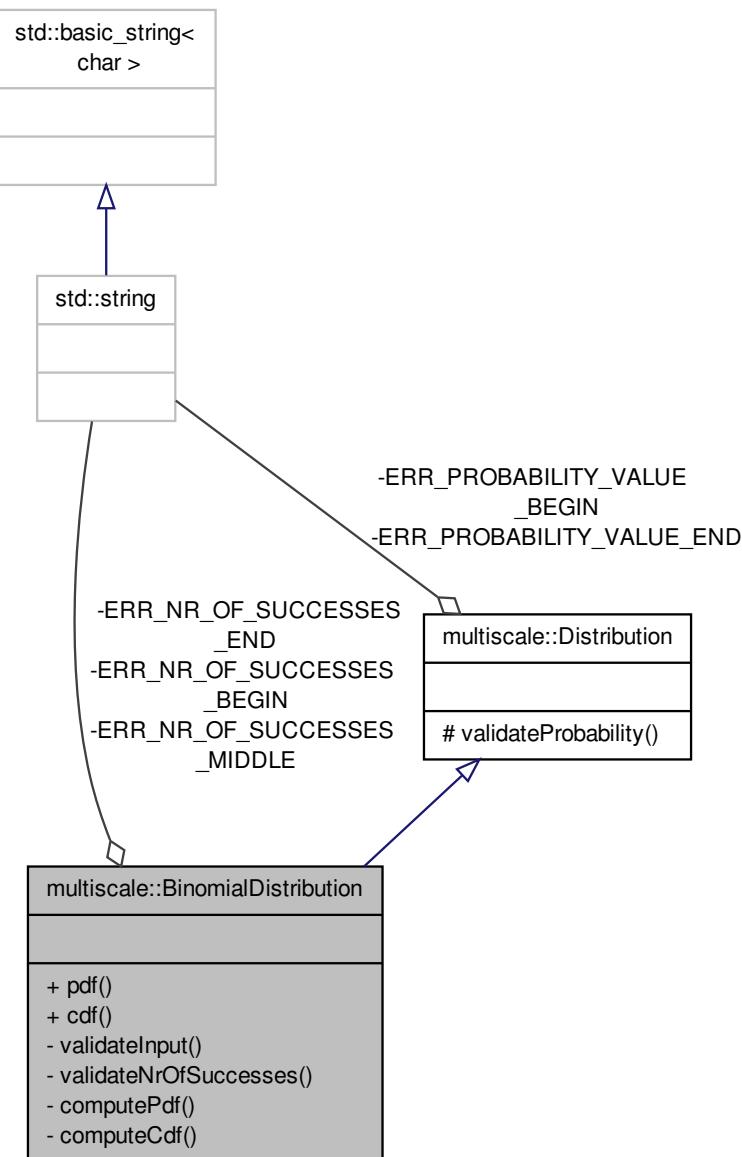
Class for analysing Binomial distributed data.

```
#include <BinomialDistribution.hpp>
```

Inheritance diagram for `multiscale::BinomialDistribution`:



Collaboration diagram for multiscale::BinomialDistribution:



## Static Public Member Functions

- static double [pdf](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability)  
*Compute the value of the probability distribution/mass function (pdf) for a binomial distribution.*
- static double [cdf](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability)  
*Compute the value of the cumulative distribution function (cdf) for a binomial distribution.*

## Static Private Member Functions

- static void [validateInput](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability)

- static void [validateNrOfSuccesses](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses)
 

*Validate the given input data.*

*Check if the number of true observations is less than or equal to the total number of observations.*
- static double [computePdf](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability)
 

*Compute the value of the probability distribution function for a binomial distribution.*
- static double [computeCdf](#) (unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability)
 

*Compute the value of the cumulative distribution function for a binomial distribution.*

## Static Private Attributes

- static const std::string [ERR\\_NR\\_OF\\_SUCCESSES\\_BEGIN](#) = "The given number of successes ("
- static const std::string [ERR\\_NR\\_OF\\_SUCCESSES\\_MIDDLE](#) = ") should be less than or equal to the total number of observations ("
- static const std::string [ERR\\_NR\\_OF\\_SUCCESSES\\_END](#) = ")."

## Additional Inherited Members

### 7.33.1 Detailed Description

Class for analysing Binomial distributed data.

Definition at line 12 of file BinomialDistribution.hpp.

### 7.33.2 Member Function Documentation

#### 7.33.2.1 double BinomialDistribution::cdf ( *unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability* ) [static]

Compute the value of the cumulative distribution function (cdf) for a binomial distribution.

##### Parameters

|                         |                                                                |
|-------------------------|----------------------------------------------------------------|
| <i>nrOfObservations</i> | The total number of observations                               |
| <i>nrOfSuccesses</i>    | The number of successes                                        |
| <i>probability</i>      | The probability p used by the cumulative distribution function |

Definition at line 17 of file BinomialDistribution.cpp.

References [computeCdf\(\)](#), and [validateInput\(\)](#).

Referenced by [TEST\(\)](#), [multiscale::verification::ModelChecker::updateAlternativeHypothesisPValue\(\)](#), and [multiscale::verification::ModelChecker::updateNullHypothesisPValue\(\)](#).

#### 7.33.2.2 double BinomialDistribution::computeCdf ( *unsigned int nrOfObservations, unsigned int nrOfSuccesses, double probability* ) [static], [private]

Compute the value of the cumulative distribution function for a binomial distribution.

##### Parameters

|                         |                                  |
|-------------------------|----------------------------------|
| <i>nrOfObservations</i> | The total number of observations |
|-------------------------|----------------------------------|

|                      |                                                                |
|----------------------|----------------------------------------------------------------|
| <i>nrOfSuccesses</i> | The number of successes                                        |
| <i>probability</i>   | The probability p used by the cumulative distribution function |

Definition at line 50 of file BinomialDistribution.cpp.

Referenced by cdf().

**7.33.2.3 double BinomialDistribution::computePdf ( unsigned int *nrOfObservations*, unsigned int *nrOfSuccesses*, double *probability* ) [static], [private]**

Compute the value of the probability distribution function for a binomial distribution.

Parameters

|                         |                                                                |
|-------------------------|----------------------------------------------------------------|
| <i>nrOfObservations</i> | The total number of observations                               |
| <i>nrOfSuccesses</i>    | The number of successes                                        |
| <i>probability</i>      | The probability p used by the cumulative distribution function |

Definition at line 43 of file BinomialDistribution.cpp.

Referenced by pdf().

**7.33.2.4 double BinomialDistribution::pdf ( unsigned int *nrOfObservations*, unsigned int *nrOfSuccesses*, double *probability* ) [static]**

Compute the value of the probability distribution/mass function (pdf) for a binomial distribution.

Parameters

|                         |                                                                |
|-------------------------|----------------------------------------------------------------|
| <i>nrOfObservations</i> | The total number of observations                               |
| <i>nrOfSuccesses</i>    | The number of successes                                        |
| <i>probability</i>      | The probability p used by the cumulative distribution function |

Definition at line 10 of file BinomialDistribution.cpp.

References computePdf(), and validateInput().

Referenced by multiscale::verification::BayesianModelChecker::computeBinomialPDF(), and TEST().

**7.33.2.5 void BinomialDistribution::validateInput ( unsigned int *nrOfObservations*, unsigned int *nrOfSuccesses*, double *probability* ) [static], [private]**

Validate the given input data.

Parameters

|                         |                                                                |
|-------------------------|----------------------------------------------------------------|
| <i>nrOfObservations</i> | The total number of observations                               |
| <i>nrOfSuccesses</i>    | The number of successes                                        |
| <i>probability</i>      | The probability p used by the cumulative distribution function |

Definition at line 24 of file BinomialDistribution.cpp.

References validateNrOfSuccesses(), and multiscale::Distribution::validateProbability().

Referenced by cdf(), and pdf().

7.33.2.6 void BinomialDistribution::validateNrOfSuccesses ( unsigned int *nrOfObservations*, unsigned int *nrOfSuccesses* )  
[static], [private]

Check if the number of true observations is less than or equal to the total number of observations.

**Parameters**

|                         |                                  |
|-------------------------|----------------------------------|
| <i>nrOfObservations</i> | The total number of observations |
| <i>nrOfSuccesses</i>    | The number of successes          |

Definition at line 30 of file BinomialDistribution.cpp.

References ERR\_NR\_OF\_SUCCESSES\_BEGIN, ERR\_NR\_OF\_SUCCESSES\_END, ERR\_NR\_OF\_SUCCESSES\_MIDDLE, MS\_throw, and multiscale::StringManipulator::toString().

Referenced by validateInput().

### 7.33.3 Member Data Documentation

**7.33.3.1 const std::string BinomialDistribution::ERR\_NR\_OF\_SUCCESSES\_BEGIN = "The given number of successes (" [static], [private]**

Definition at line 73 of file BinomialDistribution.hpp.

Referenced by validateNrOfSuccesses().

**7.33.3.2 const std::string BinomialDistribution::ERR\_NR\_OF\_SUCCESSES\_END = ")." [static], [private]**

Definition at line 75 of file BinomialDistribution.hpp.

Referenced by validateNrOfSuccesses().

**7.33.3.3 const std::string BinomialDistribution::ERR\_NR\_OF\_SUCCESSES\_MIDDLE = ") should be less than or equal to the total number of observations (" [static], [private]**

Definition at line 74 of file BinomialDistribution.hpp.

Referenced by validateNrOfSuccesses().

The documentation for this class was generated from the following files:

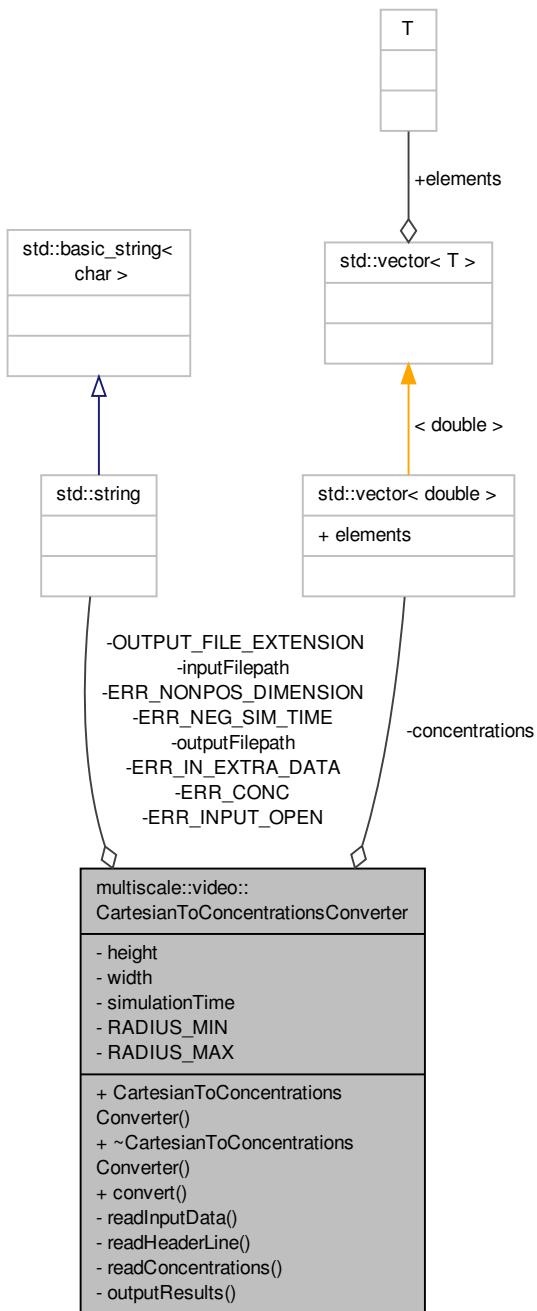
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/[BinomialDistribution.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/[BinomialDistribution.cpp](#)

## 7.34 multiscale::video::CartesianToConcentrationsConverter Class Reference

Scale the values of the rectangular geometry grid cells.

```
#include <CartesianToConcentrationsConverter.hpp>
```

Collaboration diagram for multiscale::video::CartesianToConcentrationsConverter:



## Public Member Functions

- `CartesianToConcentrationsConverter (const std::string &inputfilepath, const std::string &outputfilepath)`
- `~CartesianToConcentrationsConverter ()`
- `void convert ()`

*Start the conversion.*

## Private Member Functions

- void `readInputData ()`  
*Read the input data.*
- void `readHeaderLine (std::ifstream &fin)`  
*Read the header line.*
- void `readConcentrations (std::ifstream &fin)`  
*Read the concentrations.*
- void `outputResults ()`  
*Output the results.*

## Private Attributes

- `std::vector< double > concentrations`
- `unsigned long height`
- `unsigned long width`
- `double simulationTime`
- `std::string inputfilepath`
- `std::string outputfilepath`

## Static Private Attributes

- static const `std::string ERR_CONC` = "All `concentrations` have to be between 0 and 1."
- static const `std::string ERR_NONPOS_DIMENSION` = "The dimensions N and M must be positive."
- static const `std::string ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const `std::string ERR_INPUT_OPEN` = "The input file could not be opened"
- static const `std::string ERR_IN_EXTRA_DATA` = "The input file contains more data than required."
- static const `std::string OUTPUT_FILE_EXTENSION` = ".out"
- static const `double RADIUS_MIN` = 0.001
- static const `double RADIUS_MAX` = 0.3

### 7.34.1 Detailed Description

Scale the values of the rectangular geometry grid cells.

Definition at line 13 of file `CartesianToConcentrationsConverter.hpp`.

### 7.34.2 Constructor & Destructor Documentation

#### 7.34.2.1 `CartesianToConcentrationsConverter::CartesianToConcentrationsConverter ( const std::string & inputfilepath, const std::string & outputfilepath )`

Definition at line 16 of file `CartesianToConcentrationsConverter.cpp`.

References `height`, `simulationTime`, and `width`.

#### 7.34.2.2 `CartesianToConcentrationsConverter::~CartesianToConcentrationsConverter ( )`

Definition at line 26 of file `CartesianToConcentrationsConverter.cpp`.

### 7.34.3 Member Function Documentation

#### 7.34.3.1 void CartesianToConcentrationsConverter::convert( )

Start the conversion.

Definition at line 28 of file `CartesianToConcentrationsConverter.cpp`.

References `outputResults()`, and `readInputData()`.

Referenced by `main()`.

#### 7.34.3.2 void CartesianToConcentrationsConverter::outputResults( ) [private]

Output the results.

Definition at line 86 of file `CartesianToConcentrationsConverter.cpp`.

References `concentrations`, `multiscale::video::RectangularGnuplotScriptGenerator::generateScript()`, `height`, `outputFilepath`, `simulationTime`, and `width`.

Referenced by `convert()`.

#### 7.34.3.3 void CartesianToConcentrationsConverter::readConcentrations( std::ifstream & fin ) [private]

Read the concentrations.

Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 66 of file `CartesianToConcentrationsConverter.cpp`.

References `concentrations`, `ERR_CONC`, `height`, `MS_throw`, and `width`.

Referenced by `readInputData()`.

#### 7.34.3.4 void CartesianToConcentrationsConverter::readHeaderLine( std::ifstream & fin ) [private]

Read the header line.

The header line contains values for number of concentric circles, number of sectors and simulation time

Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 57 of file `CartesianToConcentrationsConverter.cpp`.

References `ERR_NEG_SIM_TIME`, `ERR_NONPOS_DIMENSION`, `height`, `MS_throw`, `simulationTime`, and `width`.

Referenced by `readInputData()`.

#### 7.34.3.5 void CartesianToConcentrationsConverter::readInputData( ) [private]

Read the input data.

Definition at line 33 of file `CartesianToConcentrationsConverter.cpp`.

References `ERR_IN_EXTRA_DATA`, `ERR_INPUT_OPEN`, `inputFilepath`, `MS_throw`, `readConcentrations()`, and `readHeaderLine()`.

Referenced by `convert()`.

#### 7.34.4 Member Data Documentation

7.34.4.1 `std::vector<double> multiscale::video::CartesianToConcentrationsConverter::concentrations` [private]

Concentrations received as input

Definition at line 17 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `outputResults()`, and `readConcentrations()`.

7.34.4.2 `const std::string CartesianToConcentrationsConverter::ERR_CONC = "All concentrations have to be between 0 and 1."` [static], [private]

Definition at line 61 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readConcentrations()`.

7.34.4.3 `const std::string CartesianToConcentrationsConverter::ERR_IN_EXTRA_DATA = "The input file contains more data than required."` [static], [private]

Definition at line 65 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

7.34.4.4 `const std::string CartesianToConcentrationsConverter::ERR_INPUT_OPEN = "The input file could not be opened"` [static], [private]

Definition at line 64 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

7.34.4.5 `const std::string CartesianToConcentrationsConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative."` [static], [private]

Definition at line 63 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readHeaderLine()`.

7.34.4.6 `const std::string CartesianToConcentrationsConverter::ERR_NONPOS_DIMENSION = "The dimensions N and M must be positive."` [static], [private]

Definition at line 62 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readHeaderLine()`.

7.34.4.7 `unsigned long multiscale::video::CartesianToConcentrationsConverter::height` [private]

Height of the grid

Definition at line 19 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `CartesianToConcentrationsConverter()`, `outputResults()`, `readConcentrations()`, and `readHeaderLine()`.

7.34.4.8 `std::string multiscale::video::CartesianToConcentrationsConverter::filepath` [private]

Path to the input file

Definition at line 23 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

**7.34.4.9** `const std::string CartesianToConcentrationsConverter::OUTPUT_FILE_EXTENSION = ".out" [static], [private]`

Definition at line 67 of file `CartesianToConcentrationsConverter.hpp`.

**7.34.4.10** `std::string multiscale::video::CartesianToConcentrationsConverter::outputFilepath [private]`

Path to the output file

Definition at line 24 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `outputResults()`.

**7.34.4.11** `const double CartesianToConcentrationsConverter::RADIUS_MAX = 0.3 [static], [private]`

Definition at line 70 of file `CartesianToConcentrationsConverter.hpp`.

**7.34.4.12** `const double CartesianToConcentrationsConverter::RADIUS_MIN = 0.001 [static], [private]`

Definition at line 69 of file `CartesianToConcentrationsConverter.hpp`.

**7.34.4.13** `double multiscale::video::CartesianToConcentrationsConverter::simulationTime [private]`

Simulation time

Definition at line 21 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `CartesianToConcentrationsConverter()`, `outputResults()`, and `readHeaderLine()`.

**7.34.4.14** `unsigned long multiscale::video::CartesianToConcentrationsConverter::width [private]`

Width of the grid

Definition at line 20 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `CartesianToConcentrationsConverter()`, `outputResults()`, `readConcentrations()`, and `readHeaderLine()`.

The documentation for this class was generated from the following files:

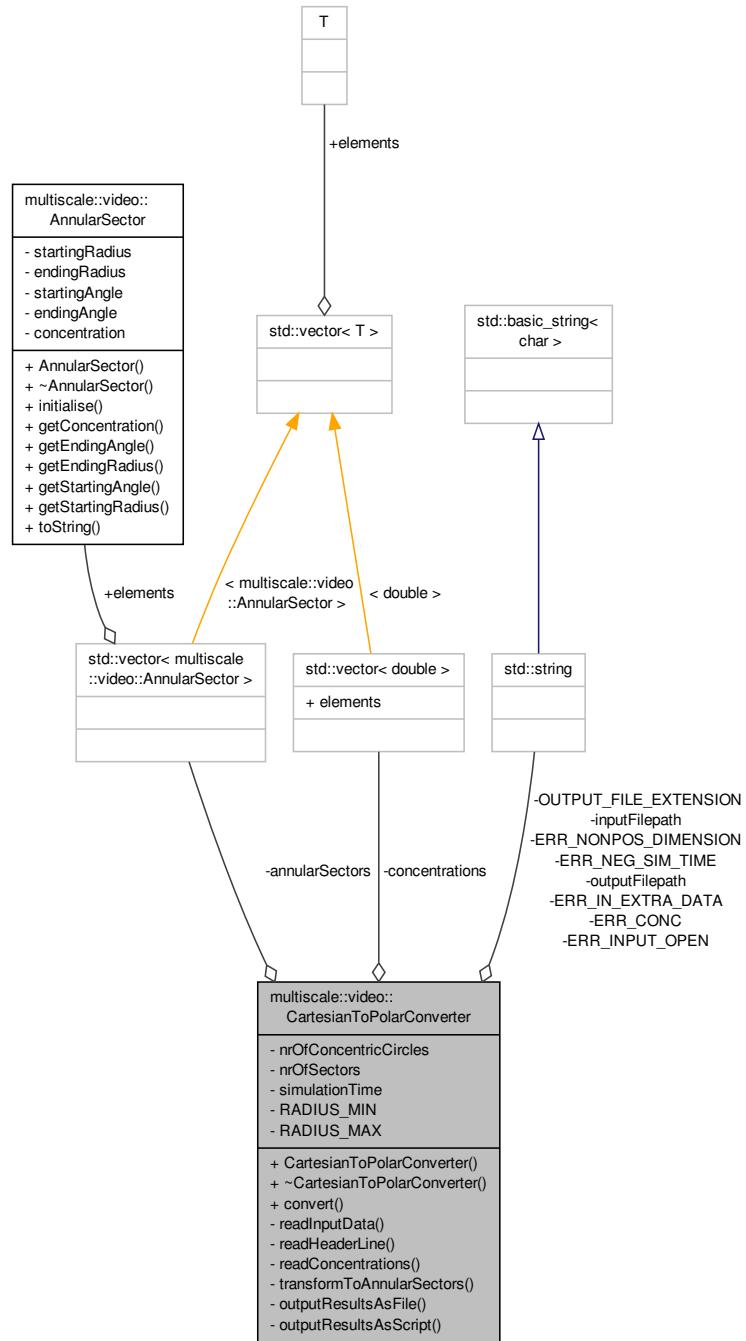
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/[CartesianToConcentrationsConverter.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/[CartesianToConcentrationsConverter.cpp](#)

## 7.35 multiscale::video::CartesianToPolarConverter Class Reference

Converter from the rectangular geometry grid cells to annular sectors.

```
#include <CartesianToPolarConverter.hpp>
```

Collaboration diagram for multiscale::video::CartesianToPolarConverter:



## Public Member Functions

- `CartesianToPolarConverter (const std::string &inputfilepath, const std::string &outputfilepath)`
- `~CartesianToPolarConverter ()`
- `void convert (bool outputToScript)`

*Start the conversion.*

## Private Member Functions

- void `readInputData ()`  
*Read the input data.*
- void `readHeaderLine (std::ifstream &fin)`  
*Read the header line.*
- void `readConcentrations (std::ifstream &fin)`  
*Read the concentrations.*
- void `transformToAnnularSectors ()`  
*Convert the concentrations to annular sectors.*
- void `outputResultsAsFile ()`  
*Output the results as a plain file.*
- void `outputResultsAsScript ()`  
*Output the results as a gnuplot script.*

## Private Attributes

- std::vector< `AnnularSector` > `annularSectors`
- std::vector< double > `concentrations`
- unsigned long `nrOfConcentricCircles`
- unsigned long `nrOfSectors`
- double `simulationTime`
- std::string `inputFilepath`
- std::string `outputFilepath`

## Static Private Attributes

- static const std::string `ERR_CONC` = "All `concentrations` have to be between 0 and 1."
- static const std::string `ERR_NONPOS_DIMENSION` = "The dimensions N and M must be positive."
- static const std::string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const std::string `ERR_INPUT_OPEN` = "The input file could not be opened"
- static const std::string `ERR_IN_EXTRA_DATA` = "The input file contains more data than required."
- static const std::string `OUTPUT_FILE_EXTENSION` = ".out"
- static const double `RADIUS_MIN` = 0.001
- static const double `RADIUS_MAX` = 0.3

### 7.35.1 Detailed Description

Converter from the rectangular geometry grid cells to annular sectors.

Definition at line 15 of file `CartesianToPolarConverter.hpp`.

### 7.35.2 Constructor & Destructor Documentation

#### 7.35.2.1 `CartesianToPolarConverter::CartesianToPolarConverter ( const std::string & inputFilepath, const std::string & outputFilepath )`

Definition at line 16 of file `CartesianToPolarConverter.cpp`.

References `nrOfConcentricCircles`, `nrOfSectors`, and `simulationTime`.

7.35.2.2 `CartesianToPolarConverter::~CartesianToPolarConverter( )`

Definition at line 26 of file `CartesianToPolarConverter.cpp`.

## 7.35.3 Member Function Documentation

7.35.3.1 `void CartesianToPolarConverter::convert( bool outputToScript )`

Start the conversion.

## Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <code>outputToScript</code> | Output to script or to plain file |
|-----------------------------|-----------------------------------|

Definition at line 28 of file `CartesianToPolarConverter.cpp`.

References `outputResultsAsFile()`, `outputResultsAsScript()`, `readInputData()`, and `transformToAnnularSectors()`.

Referenced by `main()`.

7.35.3.2 `void CartesianToPolarConverter::outputResultsAsFile( ) [private]`

Output the results as a plain file.

Definition at line 123 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `OUTPUT_FILE_EXTENSION`, and `outputFilepath`.

Referenced by `convert()`.

7.35.3.3 `void CartesianToPolarConverter::outputResultsAsScript( ) [private]`

Output the results as a gnuplot script.

Definition at line 138 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `multiscale::video::PolarGnuplotScriptGenerator::generateScript()`, `outputFilepath`, and `simulationTime`.

Referenced by `convert()`.

7.35.3.4 `void CartesianToPolarConverter::readConcentrations( std::ifstream & fin ) [private]`

Read the concentrations.

## Parameters

|                  |                   |
|------------------|-------------------|
| <code>fin</code> | Input file stream |
|------------------|-------------------|

Definition at line 72 of file `CartesianToPolarConverter.cpp`.

References `concentrations`, `ERR_CONC`, `MS_throw`, `nrOfConcentricCircles`, and `nrOfSectors`.

Referenced by `readInputData()`.

7.35.3.5 `void CartesianToPolarConverter::readHeaderLine( std::ifstream & fin ) [private]`

Read the header line.

The header line contains values for number of concentric circles, number of sectors and simulation time

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 63 of file `CartesianToPolarConverter.cpp`.

References `ERR_NEG_SIM_TIME`, `ERR_NONPOS_DIMENSION`, `MS_throw`, `nrOfConcentricCircles`, `nrOfSectors`, and `simulationTime`.

Referenced by `readInputData()`.

#### 7.35.3.6 void `CartesianToPolarConverter::readInputData( )` [private]

Read the input data.

Definition at line 39 of file `CartesianToPolarConverter.cpp`.

References `ERR_IN_EXTRA_DATA`, `ERR_INPUT_OPEN`, `inputFilepath`, `MS_throw`, `readConcentrations()`, and `readHeaderLine()`.

Referenced by `convert()`.

#### 7.35.3.7 void `CartesianToPolarConverter::transformToAnnularSectors( )` [private]

Convert the concentrations to annular sectors.

Definition at line 92 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `concentrations`, `nrOfConcentricCircles`, `nrOfSectors`, `RADIUS_MAX`, and `RADIUS_MIN`.

Referenced by `convert()`.

### 7.35.4 Member Data Documentation

#### 7.35.4.1 std::vector<`AnnularSector`> `multiscale::video::CartesianToPolarConverter::annularSectors` [private]

Resulting annular sectors

Definition at line 20 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`, `outputResultsAsScript()`, and `transformToAnnularSectors()`.

#### 7.35.4.2 std::vector<double> `multiscale::video::CartesianToPolarConverter::concentrations` [private]

Concentrations received as input

Definition at line 22 of file `CartesianToPolarConverter.hpp`.

Referenced by `readConcentrations()`, and `transformToAnnularSectors()`.

#### 7.35.4.3 const std::string `CartesianToPolarConverter::ERR_CONC` = "All concentrations have to be between 0 and 1." [static], [private]

Definition at line 79 of file `CartesianToPolarConverter.hpp`.

Referenced by `readConcentrations()`.

7.35.4.4 `const std::string CartesianToPolarConverter::ERR_IN_EXTRA_DATA = "The input file contains more data than required." [static], [private]`

Definition at line 83 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.35.4.5 `const std::string CartesianToPolarConverter::ERR_INPUT_OPEN = "The input file could not be opened" [static], [private]`

Definition at line 82 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.35.4.6 `const std::string CartesianToPolarConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative." [static], [private]`

Definition at line 81 of file `CartesianToPolarConverter.hpp`.

Referenced by `readHeaderLine()`.

7.35.4.7 `const std::string CartesianToPolarConverter::ERR_NONPOS_DIMENSION = "The dimensions N and M must be positive." [static], [private]`

Definition at line 80 of file `CartesianToPolarConverter.hpp`.

Referenced by `readHeaderLine()`.

7.35.4.8 `std::string multiscale::video::CartesianToPolarConverter::inputFilepath [private]`

Path to the input file

Definition at line 32 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.35.4.9 `unsigned long multiscale::video::CartesianToPolarConverter::nrOfConcentricCircles [private]`

Number of concentric circles

Definition at line 25 of file `CartesianToPolarConverter.hpp`.

Referenced by `CartesianToPolarConverter()`, `readConcentrations()`, `readHeaderLine()`, and `transformToAnnularSectors()`.

7.35.4.10 `unsigned long multiscale::video::CartesianToPolarConverter::nrOfSectors [private]`

Number of sectors

Definition at line 27 of file `CartesianToPolarConverter.hpp`.

Referenced by `CartesianToPolarConverter()`, `readConcentrations()`, `readHeaderLine()`, and `transformToAnnularSectors()`.

7.35.4.11 `const std::string CartesianToPolarConverter::OUTPUT_FILE_EXTENSION = ".out" [static], [private]`

Definition at line 85 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`.

#### 7.35.4.12 std::string multiscale::video::CartesianToPolarConverter::outputFilepath [private]

Path to the output file

Definition at line 34 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`, and `outputResultsAsScript()`.

#### 7.35.4.13 const double CartesianToPolarConverter::RADIUS\_MAX = 0.3 [static], [private]

Definition at line 88 of file `CartesianToPolarConverter.hpp`.

Referenced by `transformToAnnularSectors()`.

#### 7.35.4.14 const double CartesianToPolarConverter::RADIUS\_MIN = 0.001 [static], [private]

Definition at line 87 of file `CartesianToPolarConverter.hpp`.

Referenced by `transformToAnnularSectors()`.

#### 7.35.4.15 double multiscale::video::CartesianToPolarConverter::simulationTime [private]

Simulation time corresponding to the input data

Definition at line 29 of file `CartesianToPolarConverter.hpp`.

Referenced by `CartesianToPolarConverter()`, `outputResultsAsScript()`, and `readHeaderLine()`.

The documentation for this class was generated from the following files:

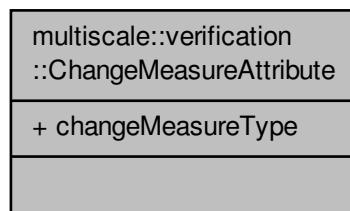
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/`CartesianToPolarConverter.hpp`](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/`CartesianToPolarConverter.cpp`](#)

## 7.36 multiscale::verification::ChangeMeasureAttribute Class Reference

Class for representing a change measure attribute.

```
#include <ChangeMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ChangeMeasureAttribute:



## Public Attributes

- [ChangeMeasureType changeMeasureType](#)

### 7.36.1 Detailed Description

Class for representing a change measure attribute.

Definition at line 28 of file ChangeMeasureAttribute.hpp.

### 7.36.2 Member Data Documentation

#### 7.36.2.1 ChangeMeasureType multiscale::verification::ChangeMeasureAttribute::changeMeasureType

The change measure type

Definition at line 32 of file ChangeMeasureAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure(), and multiscale::verification::NumericMeasureCollectionVisitor::evaluateChangeTemporalNumericCollection().

The documentation for this class was generated from the following file:

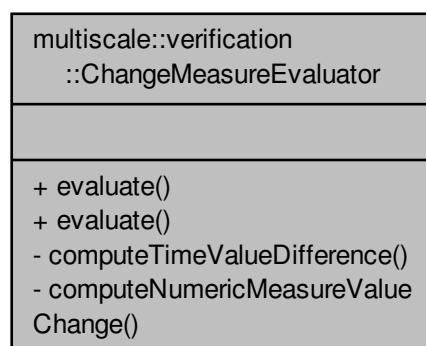
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[ChangeMeasureAttribute.hpp](#)

## 7.37 multiscale::verification::ChangeMeasureEvaluator Class Reference

Class for evaluating change measure expressions.

```
#include <ChangeMeasureEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::ChangeMeasureEvaluator:



## Static Public Member Functions

- static double [evaluate](#) (const [ChangeMeasureType](#) &changeMeasureType, double temporalNumericMeasureCollectionFirstValue, double temporalNumericMeasureCollectionSecondValue)

*Compute the change measure value considering temporal numeric measure collection and time values.*

- static double `evaluate` (const `ChangeMeasureType` &changeMeasureType, double `temporalNumericMeasureFirstTimepoint`, double `temporalNumericMeasureSecondTimepoint`, unsigned long `timeValueFirstTimepoint`, unsigned long `timeValueSecondTimepoint`)

*Compute the change measure value considering the given temporal numeric measure and time values.*

## Static Private Member Functions

- static double `computeTimeValueDifference` (unsigned long `timeValueFirstTimepoint`, unsigned long `timeValueSecondTimepoint`)

*Compute the time value difference considering the given time values.*

- static double `computeNumericMeasureValueChange` (const `ChangeMeasureType` &changeMeasureType, double `temporalNumericMeasureFirstTimepoint`, double `temporalNumericMeasureSecondTimepoint`)

*Compute the numeric measure value change considering the given change measure and numeric values.*

### 7.37.1 Detailed Description

Class for evaluating change measure expressions.

Definition at line 15 of file `ChangeMeasureEvaluator.hpp`.

### 7.37.2 Member Function Documentation

- 7.37.2.1 static double `multiscale::verification::ChangeMeasureEvaluator::computeNumericMeasureValueChange` ( const `ChangeMeasureType` & `changeMeasureType`, double `temporalNumericMeasureFirstTimepoint`, double `temporalNumericMeasureSecondTimepoint` ) [inline], [static], [private]

Compute the numeric measure value change considering the given change measure and numeric values.

#### Parameters

|                                                    |                                                                                                    |
|----------------------------------------------------|----------------------------------------------------------------------------------------------------|
| <code>changeMeasureType</code>                     | The type of the change measure                                                                     |
| <code>temporalNumericMeasureFirstTimepoint</code>  | The temporal numeric measure value corresponding to the trace starting from the initial time-point |
| <code>temporalNumericMeasureSecondTimepoint</code> | The temporal numeric measure value corresponding to the trace starting from the second timepoint   |

Definition at line 90 of file `ChangeMeasureEvaluator.hpp`.

References `multiscale::verification::Derivative`, `multiscale::ERR_UNDEFINED_ENUM_VALUE`, `MS_throw`, and `multiscale::verification::Ratio`.

Referenced by `evaluate()`.

- 7.37.2.2 static double `multiscale::verification::ChangeMeasureEvaluator::computeTimeValueDifference` ( unsigned long `timeValueFirstTimepoint`, unsigned long `timeValueSecondTimepoint` ) [inline], [static], [private]

Compute the time value difference considering the given time values.

Time difference = (second timepoint value) - (first timepoint value)

**Parameters**

|                                            |                                                      |
|--------------------------------------------|------------------------------------------------------|
| <i>timeValueFirst</i><br><i>Timepoint</i>  | The time value corresponding to the first timepoint  |
| <i>timeValueSecond</i><br><i>Timepoint</i> | The time value corresponding to the second timepoint |

Definition at line 77 of file ChangeMeasureEvaluator.hpp.

Referenced by evaluate().

**7.37.2.3 static double multiscale::verification::ChangeMeasureEvaluator::evaluate ( const ChangeMeasureType & changeMeasureType, double temporalNumericMeasureCollectionFirstValue, double temporalNumericMeasureCollectionSecondValue ) [inline], [static]**

Compute the change measure value considering temporal numeric measure collection and time values.

**Parameters**

|                                                    |                                                                                                                                       |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <i>changeMeasureType</i>                           | The type of the change measure                                                                                                        |
| <i>temporalNumericMeasureCollectionFirstValue</i>  | The temporal numeric measure collection value corresponding to the trace starting from the initial timepoint                          |
| <i>temporalNumericMeasureCollectionSecondValue</i> | The temporal numeric measure collection value corresponding to the trace starting from the timepoint succeeding the initial timepoint |

Definition at line 29 of file ChangeMeasureEvaluator.hpp.

References computeNumericMeasureValueChange().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure(), and multiscale::verification::NumericMeasureCollectionVisitor::evaluateChangeTemporalNumericCollection().

**7.37.2.4 static double multiscale::verification::ChangeMeasureEvaluator::evaluate ( const ChangeMeasureType & changeMeasureType, double temporalNumericMeasureFirstTimepoint, double temporalNumericMeasureSecondTimepoint, unsigned long timeValueFirstTimepoint, unsigned long timeValueSecondTimepoint ) [inline], [static]**

Compute the change measure value considering the given temporal numeric measure and time values.

**Parameters**

|                                             |                                                                                                   |
|---------------------------------------------|---------------------------------------------------------------------------------------------------|
| <i>changeMeasureType</i>                    | The type of the change measure                                                                    |
| <i>temporalNumericMeasureFirstTimepoint</i> | The temporal numeric measure value corresponding to the trace starting from the initial timepoint |

|                                                                                                  |                                                                                                  |
|--------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| <i>temporal</i> ↵<br><i>Numeric</i> ↵<br><i>Measure</i> ↵<br><i>Second</i> ↵<br><i>Timepoint</i> | The temporal numeric measure value corresponding to the trace starting from the second timepoint |
| <i>timeValueFirst</i> ↵<br><i>Timepoint</i>                                                      | The time value corresponding to the first timepoint                                              |
| <i>timeValue</i> ↵<br><i>Second</i> ↵<br><i>Timepoint</i>                                        | The time value corresponding to the second timepoint                                             |

Definition at line 49 of file ChangeMeasureEvaluator.hpp.

References computeNumericMeasureValueChange(), and computeTimeValueDifference().

The documentation for this class was generated from the following file:

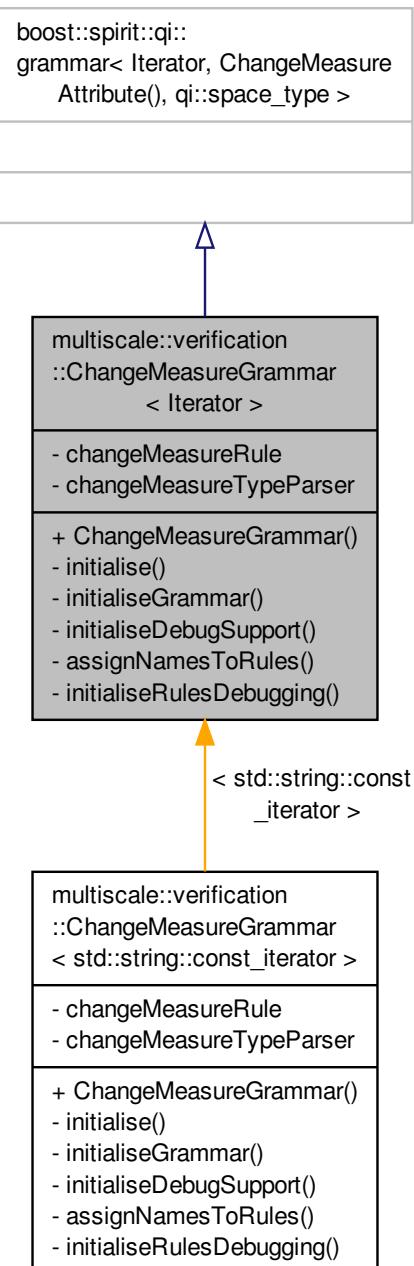
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[ChangeMeasureEvaluator.hpp](#)

## 7.38 multiscale::verification::ChangeMeasureGrammar< Iterator > Class Template Reference

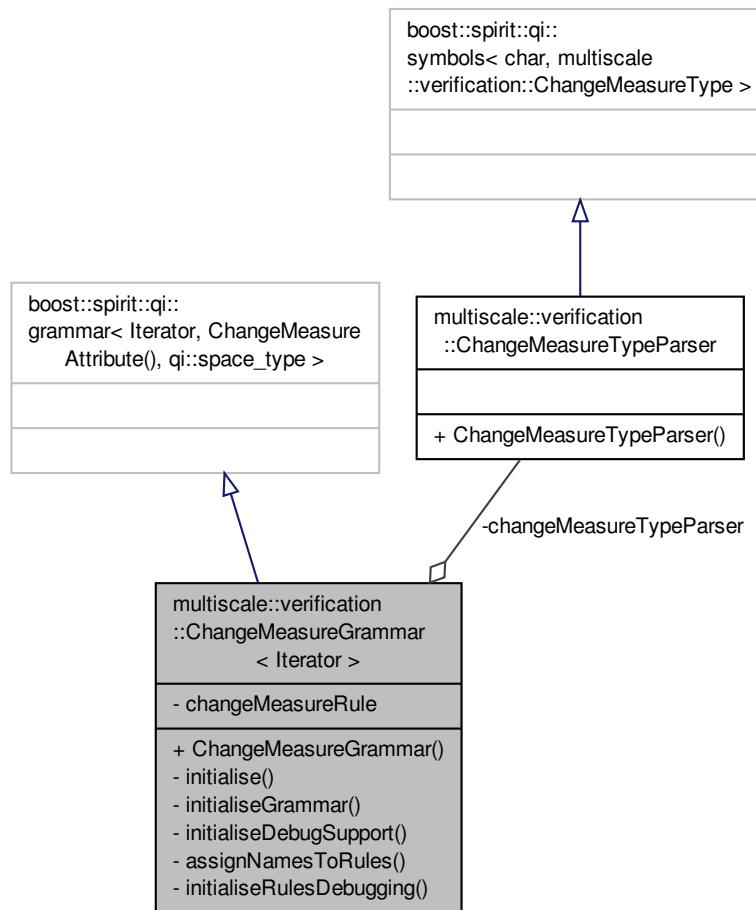
The grammar for parsing change measure statements.

```
#include <ChangeMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::ChangeMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::ChangeMeasureGrammar< Iterator >:



## Public Member Functions

- [ChangeMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*

## Private Attributes

- qi::rule< Iterator,  
[ChangeMeasureAttribute\(\)](#),  
qi::space\_type > changeMeasureRule
- [ChangeMeasureTypeParser](#) changeMeasureTypeParser

### 7.38.1 Detailed Description

```
template<typename Iterator> class multiscale::verification::ChangeMeasureGrammar< Iterator >
```

The grammar for parsing change measure statements.

Definition at line 30 of file ChangeMeasureGrammar.hpp.

### 7.38.2 Constructor & Destructor Documentation

```
7.38.2.1 template<typename Iterator > multiscale::verification::ChangeMeasureGrammar< Iterator >::ChangeMeasureGrammar()
```

Definition at line 29 of file ChangeMeasureGrammarDefinition.hpp.

References multiscale::verification::ChangeMeasureGrammar< Iterator >::initialise().

### 7.38.3 Member Function Documentation

```
7.38.3.1 template<typename Iterator > void multiscale::verification::ChangeMeasureGrammar< Iterator >::assignNamesToRules() [private]
```

Assign names to the rules.

Definition at line 59 of file ChangeMeasureGrammarDefinition.hpp.

```
7.38.3.2 template<typename Iterator > void multiscale::verification::ChangeMeasureGrammar< Iterator >::initialise() [private]
```

Initialisation function.

Definition at line 36 of file ChangeMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::ChangeMeasureGrammar< Iterator >::ChangeMeasureGrammar().

```
7.38.3.3 template<typename Iterator > void multiscale::verification::ChangeMeasureGrammar< Iterator >::initialiseDebugSupport() [private]
```

Initialise debug support.

Definition at line 50 of file ChangeMeasureGrammarDefinition.hpp.

```
7.38.3.4 template<typename Iterator > void multiscale::verification::ChangeMeasureGrammar< Iterator >::initialiseGrammar() [private]
```

Initialise the grammar.

Definition at line 43 of file ChangeMeasureGrammarDefinition.hpp.

---

7.38.3.5 template<typename Iterator > void multiscale::verification::ChangeMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 65 of file ChangeMeasureGrammarDefinition.hpp.

#### 7.38.4 Member Data Documentation

7.38.4.1 template<typename Iterator> qi::rule<Iterator, ChangeMeasureAttribute(), qi::space\_type> multiscale::verification::ChangeMeasureGrammar< Iterator >::changeMeasureRule [private]

The rule for parsing a change measure

Definition at line 36 of file ChangeMeasureGrammar.hpp.

7.38.4.2 template<typename Iterator> ChangeMeasureTypeParser multiscale::verification::ChangeMeasureGrammar< Iterator >::changeMeasureTypeParser [private]

The change measure type parser

Definition at line 41 of file ChangeMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

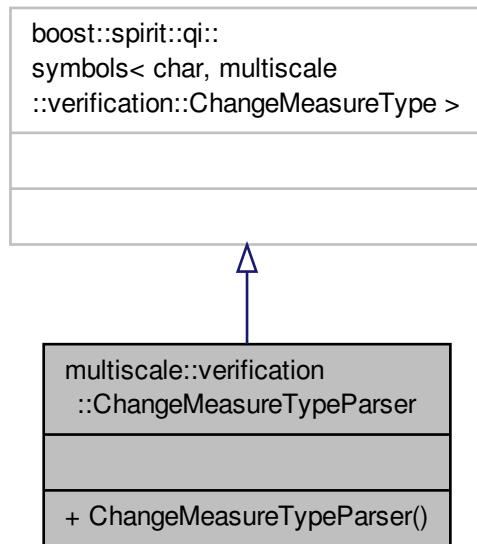
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[ChangeMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[ChangeMeasureGrammarDefinition.hpp](#)

#### 7.39 multiscale::verification::ChangeMeasureTypeParser Struct Reference

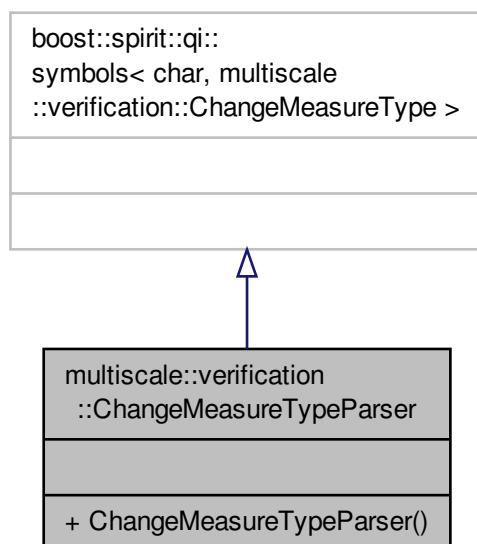
Symbol table and parser for the change measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::ChangeMeasureTypeParser:



Collaboration diagram for multiscale::verification::ChangeMeasureTypeParser:



## Public Member Functions

- [ChangeMeasureTypeParser \(\)](#)

### 7.39.1 Detailed Description

Symbol table and parser for the change measure type.

Definition at line 70 of file [SymbolTables.hpp](#).

### 7.39.2 Constructor & Destructor Documentation

#### 7.39.2.1 multiscale::verification::ChangeMeasureTypeParser::ChangeMeasureTypeParser ( ) [inline]

Definition at line 73 of file [SymbolTables.hpp](#).

References multiscale::verification::Derivative, and multiscale::verification::Ratio.

The documentation for this struct was generated from the following file:

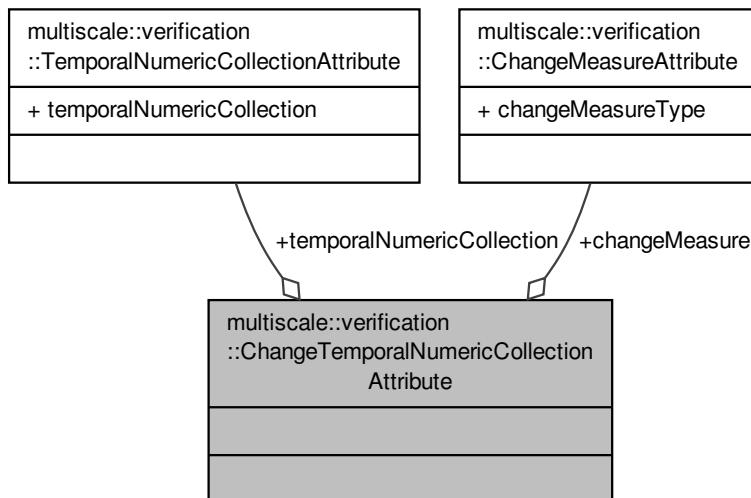
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.40 multiscale::verification::ChangeTemporalNumericCollectionAttribute Class Reference

Class for representing a change temporal numeric collection attribute.

```
#include <ChangeTemporalNumericCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ChangeTemporalNumericCollectionAttribute:



## Public Attributes

- [ChangeMeasureAttribute changeMeasure](#)
- [TemporalNumericCollectionAttribute temporalNumericCollection](#)

### 7.40.1 Detailed Description

Class for representing a change temporal numeric collection attribute.

Definition at line 16 of file ChangeTemporalNumericCollectionAttribute.hpp.

### 7.40.2 Member Data Documentation

#### 7.40.2.1 ChangeMeasureAttribute multiscale::verification::ChangeTemporalNumericCollectionAttribute::changeMeasure

The change measure

Definition at line 21 of file ChangeTemporalNumericCollectionAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

#### 7.40.2.2 TemporalNumericCollectionAttribute multiscale::verification::ChangeTemporalNumericCollectionAttribute::temporalNumericCollection

The temporal numeric collection

Definition at line 23 of file ChangeTemporalNumericCollectionAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

The documentation for this class was generated from the following file:

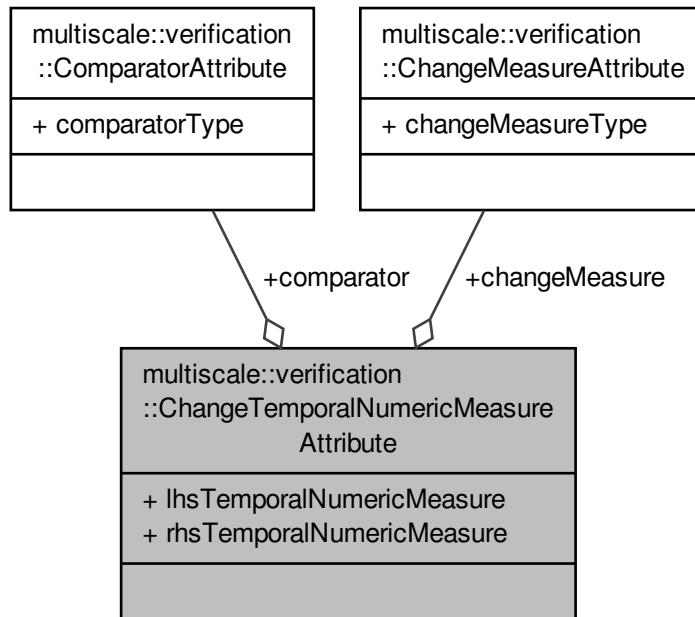
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericCollectionAttribute.hpp

## 7.41 multiscale::verification::ChangeTemporalNumericMeasureAttribute Class Reference

Class for representing a change temporal numeric measure attribute.

```
#include <ChangeTemporalNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ChangeTemporalNumericMeasureAttribute:



## Public Attributes

- `ChangeMeasureAttribute changeMeasure`
- `TemporalNumericMeasureType lhsTemporalNumericMeasure`
- `ComparatorAttribute comparator`
- `TemporalNumericMeasureType rhsTemporalNumericMeasure`

### 7.41.1 Detailed Description

Class for representing a change temporal numeric measure attribute.

Definition at line 17 of file `ChangeTemporalNumericMeasureAttribute.hpp`.

### 7.41.2 Member Data Documentation

#### 7.41.2.1 ChangeMeasureAttribute multiscale::verification::ChangeTemporalNumericMeasureAttribute::changeMeasure

The change measure

Definition at line 22 of file `ChangeTemporalNumericMeasureAttribute.hpp`.

Referenced by `multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure()`.

#### 7.41.2.2 ComparatorAttribute multiscale::verification::ChangeTemporalNumericMeasureAttribute::comparator

The comparator

Definition at line 26 of file ChangeTemporalNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeTemporalNumericMeasure().

#### 7.41.2.3 TemporalNumericMeasureType multiscale::verification::ChangeTemporalNumericMeasureAttribute::lhs ← TemporalNumericMeasure

The left hand side temporal numeric measure

Definition at line 24 of file ChangeTemporalNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure().

#### 7.41.2.4 TemporalNumericMeasureType multiscale::verification::ChangeTemporalNumericMeasureAttribute::rhs ← TemporalNumericMeasure

The right hand side temporal numeric measure

Definition at line 28 of file ChangeTemporalNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeTemporalNumericMeasure().

The documentation for this class was generated from the following file:

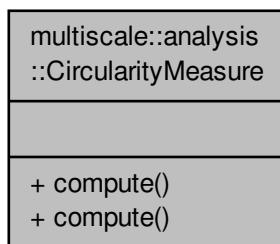
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericMeasureAttribute.hpp

## 7.42 multiscale::analysis::CircularityMeasure Class Reference

Class for computing the circularity measure for the given collection of points.

```
#include <CircularityMeasure.hpp>
```

Collaboration diagram for multiscale::analysis::CircularityMeasure:



### Static Public Member Functions

- static double **compute** (const std::vector< cv::Point2f > &points)  
*Compute circularity measure for the given collection of points.*
- static double **compute** (const std::vector< cv::Point > &points)  
*Compute circularity measure for the given collection of points.*

### 7.42.1 Detailed Description

Class for computing the circularity measure for the given collection of points.

Definition at line 15 of file CircularityMeasure.hpp.

### 7.42.2 Member Function Documentation

#### 7.42.2.1 double CircularityMeasure::compute ( const std::vector< cv::Point2f > & points ) [static]

Compute circularity measure for the given collection of points.

The circularity measure is equal to the standard circularity measure described in the following paper:

Joviša Žunić, Kaoru Hirota, Paul L. Rosin, A Hu moment invariant as a shape circularity measure, Pattern Recognition, Volume 43, Issue 1, January 2010, Pages 47-57, ISSN 0031-3203, <http://dx.doi.org/10.1016/j.patcog.2009.06.017>.

Definition at line 7 of file CircularityMeasure.cpp.

References multiscale::Geometry2D::PI.

#### 7.42.2.2 double CircularityMeasure::compute ( const std::vector< cv::Point > & points ) [static]

Compute circularity measure for the given collection of points.

The circularity measure is equal to the standard circularity measure described in the following paper:

Joviša Žunić, Kaoru Hirota, Paul L. Rosin, A Hu moment invariant as a shape circularity measure, Pattern Recognition, Volume 43, Issue 1, January 2010, Pages 47-57, ISSN 0031-3203, <http://dx.doi.org/10.1016/j.patcog.2009.06.017>.

Definition at line 23 of file CircularityMeasure.cpp.

References multiscale::Geometry2D::PI.

The documentation for this class was generated from the following files:

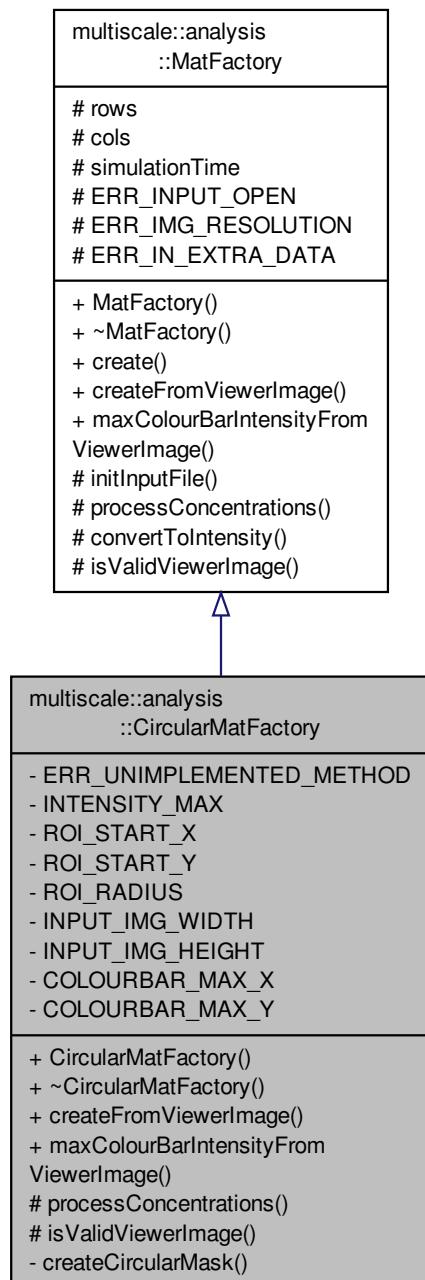
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/CircularityMeasure.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularityMeasure.cpp

## 7.43 multiscale::analysis::CircularMatFactory Class Reference

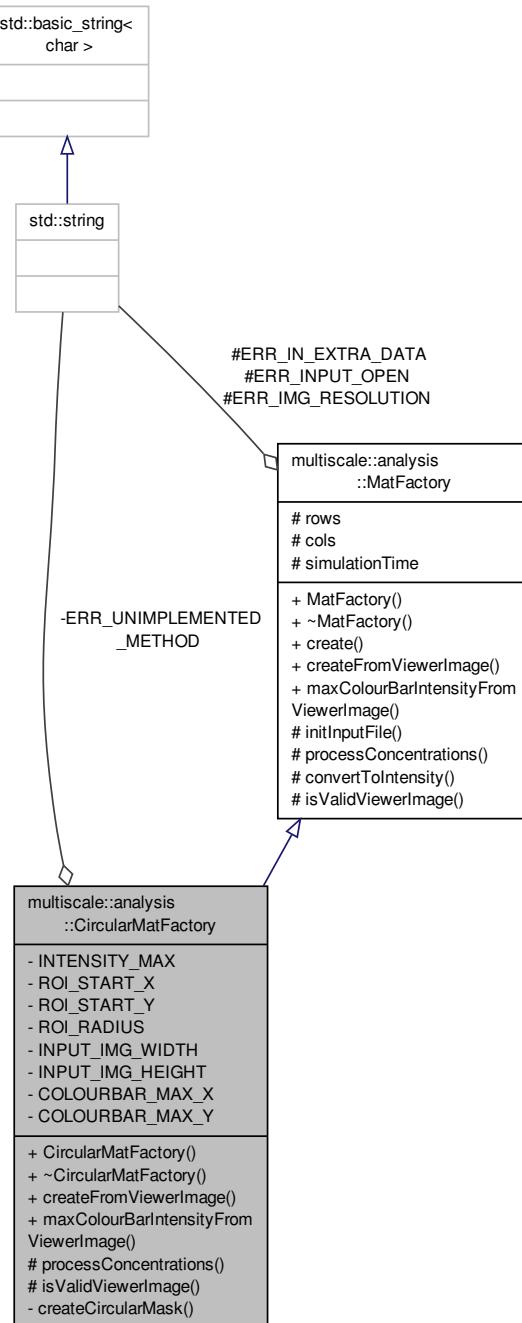
Class for creating a Mat object considering a circular grid.

```
#include <CircularMatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::CircularMatFactory:



Collaboration diagram for multiscale::analysis::CircularMatFactory:



## Public Member Functions

- `CircularMatFactory ()`
- `~CircularMatFactory ()`
- `cv::Mat createFromViewerImage (const std::string &inputFile) override`

*Create a Mat object from the image file obtained from the CircularGeometryViewer.*

- `double maxColourBarIntensityFromViewerImage (const std::string &inputFile) override`

*Get the maximum grayscale intensity of the colour bar in the image.*

## Protected Member Functions

- `unsigned char * processConcentrations (std::ifstream &fin) override`  
*Process the concentrations from the input file.*
- `bool isValidViewerImage (const cv::Mat &image) override`  
*Check if the image generated by the viewer has the required resolution.*

## Private Member Functions

- `cv::Mat createCircularMask (unsigned int originX, unsigned int originY, unsigned int radius, const cv::Mat &image)`  
*Create a mask inside the circle with origin at (originX, originY) and the given radius.*

## Static Private Attributes

- `static const std::string ERR_UNIMPLEMENTED_METHOD = "The method you called is not implemented."`
- `static const int INTENSITY_MAX = 255`
- `static const int ROI_START_X = 1024`
- `static const int ROI_START_Y = 786`
- `static const int ROI_RADIUS = 615`
- `static const int INPUT_IMG_WIDTH = 2048`
- `static const int INPUT_IMG_HEIGHT = 1572`
- `static const int COLOURBAR_MAX_X = 1775`
- `static const int COLOURBAR_MAX_Y = 56`

## Additional Inherited Members

### 7.43.1 Detailed Description

Class for creating a Mat object considering a circular grid.

Definition at line 12 of file CircularMatFactory.hpp.

### 7.43.2 Constructor & Destructor Documentation

#### 7.43.2.1 CircularMatFactory::CircularMatFactory ( )

Definition at line 10 of file CircularMatFactory.cpp.

#### 7.43.2.2 CircularMatFactory::~CircularMatFactory ( )

Definition at line 12 of file CircularMatFactory.cpp.

### 7.43.3 Member Function Documentation

#### 7.43.3.1 cv::Mat CircularMatFactory::createCircularMask ( `unsigned int originX, unsigned int originY, unsigned int radius,` `const cv::Mat & image ) [private]`

Create a mask inside the circle with origin at (originX, originY) and the given radius.

Create a mask with 255 intensity pixels inside the circle with origin at (originX, originY) and the given radius. All the other pixels have intensity zero.

The original image is provided only for computing its size correctly.

#### Parameters

|                |                                 |
|----------------|---------------------------------|
| <i>originX</i> | The x coordinate for the origin |
| <i>originY</i> | The y coordinate for the origin |
| <i>radius</i>  | The size of the radius          |
| <i>image</i>   | The original image              |

Definition at line 48 of file CircularMatFactory.cpp.

References INTENSITY\_MAX.

Referenced by createFromViewerImage().

**7.43.3.2 cv::Mat CircularMatFactory::createFromViewerImage ( const std::string & *inputFile* ) [override], [virtual]**

Create a Mat object from the image file obtained from the CircularGeometryViewer.

Create the Mat instance from the given image file

#### Parameters

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 14 of file CircularMatFactory.cpp.

References createCircularMask(), isValidViewerImage(), ROI\_RADIUS, ROI\_START\_X, and ROI\_START\_Y.

Referenced by main().

**7.43.3.3 bool CircularMatFactory::isValidViewerImage ( const cv::Mat & *image* ) [override], [protected], [virtual]**

Check if the image generated by the viewer has the required resolution.

#### Parameters

|              |                               |
|--------------|-------------------------------|
| <i>image</i> | Image generated by the viewer |
|--------------|-------------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 58 of file CircularMatFactory.cpp.

References multiscale::analysis::MatFactory::ERR\_IMG\_RESOLUTION, multiscale::analysis::MatFactory::ERR\_← INPUT\_OPEN, INPUT\_IMG\_HEIGHT, INPUT\_IMG\_WIDTH, and MS\_throw.

Referenced by createFromViewerImage(), and maxColourBarIntensityFromViewerImage().

**7.43.3.4 double CircularMatFactory::maxColourBarIntensityFromViewerImage ( const std::string & *inputFile* ) [override], [virtual]**

Get the maximum grayscale intensity of the colour bar in the image.

#### Parameters

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 33 of file CircularMatFactory.cpp.

References COLOURBAR\_MAX\_X, COLOURBAR\_MAX\_Y, and [isValidViewerImage\(\)](#).

Referenced by [main\(\)](#).

**7.43.3.5 `unsigned char * CircularMatFactory::processConcentrations ( std::ifstream & fin ) [override], [protected], [virtual]`**

Process the concentrations from the input file.

REMARK: This method is not implemented and throws an error when called.

#### Parameters

|            |                                                          |
|------------|----------------------------------------------------------|
| <i>fin</i> | Input file stream from which the concentrations are read |
|------------|----------------------------------------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 41 of file CircularMatFactory.cpp.

References [ERR\\_UNIMPLEMENTED\\_METHOD](#), and [MS\\_throw](#).

### 7.43.4 Member Data Documentation

**7.43.4.1 `const int CircularMatFactory::COLOURBAR_MAX_X = 1775 [static], [private]`**

Definition at line 79 of file CircularMatFactory.hpp.

Referenced by [maxColourBarIntensityFromViewerImage\(\)](#).

**7.43.4.2 `const int CircularMatFactory::COLOURBAR_MAX_Y = 56 [static], [private]`**

Definition at line 80 of file CircularMatFactory.hpp.

Referenced by [maxColourBarIntensityFromViewerImage\(\)](#).

**7.43.4.3 `const std::string CircularMatFactory::ERR_UNIMPLEMENTED_METHOD = "The method you called is not implemented." [static], [private]`**

Definition at line 68 of file CircularMatFactory.hpp.

Referenced by [processConcentrations\(\)](#).

**7.43.4.4 `const int CircularMatFactory::INPUT_IMG_HEIGHT = 1572 [static], [private]`**

Definition at line 77 of file CircularMatFactory.hpp.

Referenced by [isValidViewerImage\(\)](#).

**7.43.4.5 `const int CircularMatFactory::INPUT_IMG_WIDTH = 2048 [static], [private]`**

Definition at line 76 of file CircularMatFactory.hpp.

Referenced by [isValidViewerImage\(\)](#).

7.43.4.6 `const int CircularMatFactory::INTENSITY_MAX = 255` [static], [private]

Definition at line 70 of file CircularMatFactory.hpp.

Referenced by `createCircularMask()`.

7.43.4.7 `const int CircularMatFactory::ROI_RADIUS = 615` [static], [private]

Definition at line 74 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.43.4.8 `const int CircularMatFactory::ROI_START_X = 1024` [static], [private]

Definition at line 72 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.43.4.9 `const int CircularMatFactory::ROI_START_Y = 786` [static], [private]

Definition at line 73 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

The documentation for this class was generated from the following files:

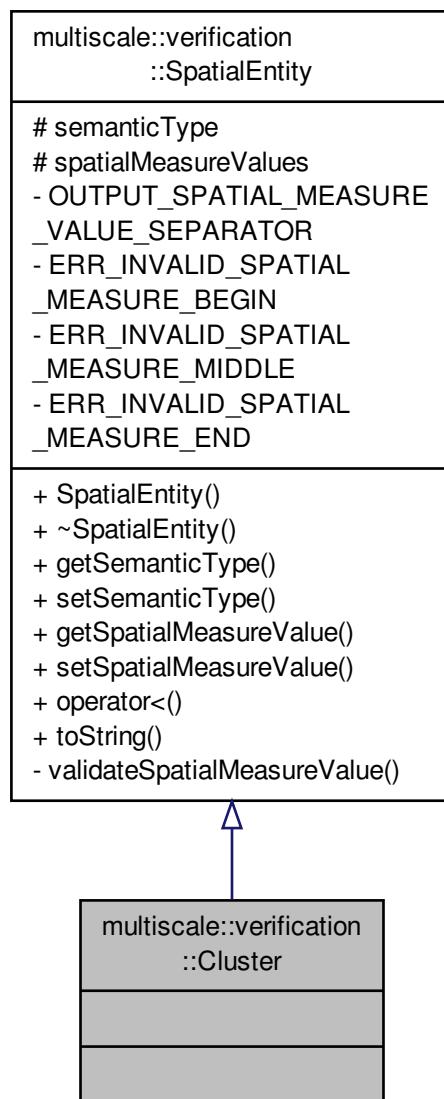
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/[CircularMatFactory.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/[CircularMatFactory.cpp](#)

## 7.44 multiscale::verification::Cluster Class Reference

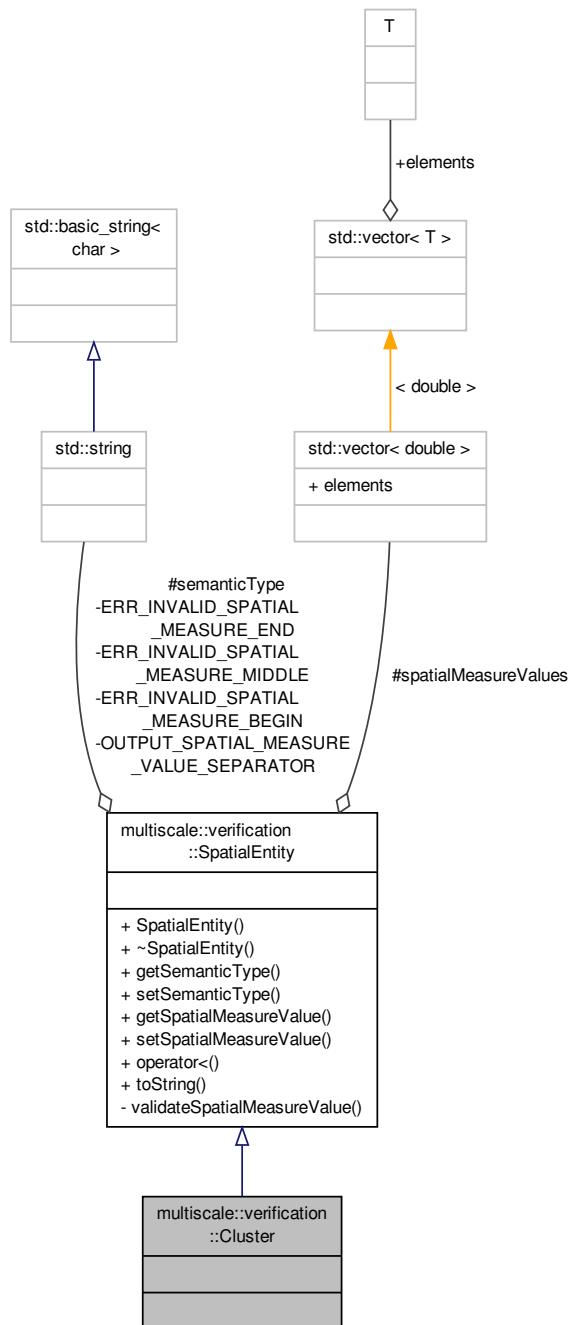
Class for representing a cluster.

```
#include <Cluster.hpp>
```

Inheritance diagram for multiscale::verification::Cluster:



Collaboration diagram for multiscale::verification::Cluster:



## Additional Inherited Members

### 7.44.1 Detailed Description

Class for representing a cluster.

Definition at line 21 of file Cluster.hpp.

The documentation for this class was generated from the following file:

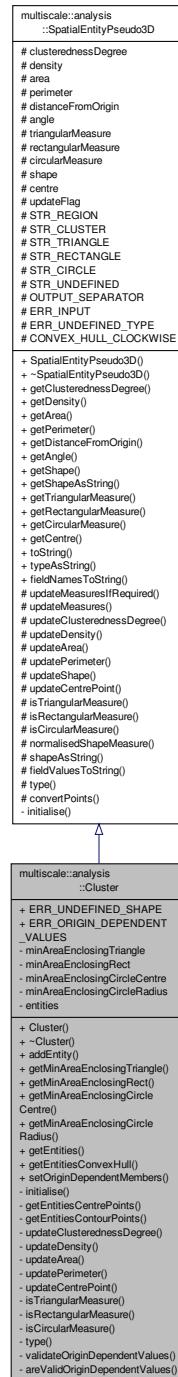
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[Cluster.hpp](#)

## 7.45 multiscale::analysis::Cluster Class Reference

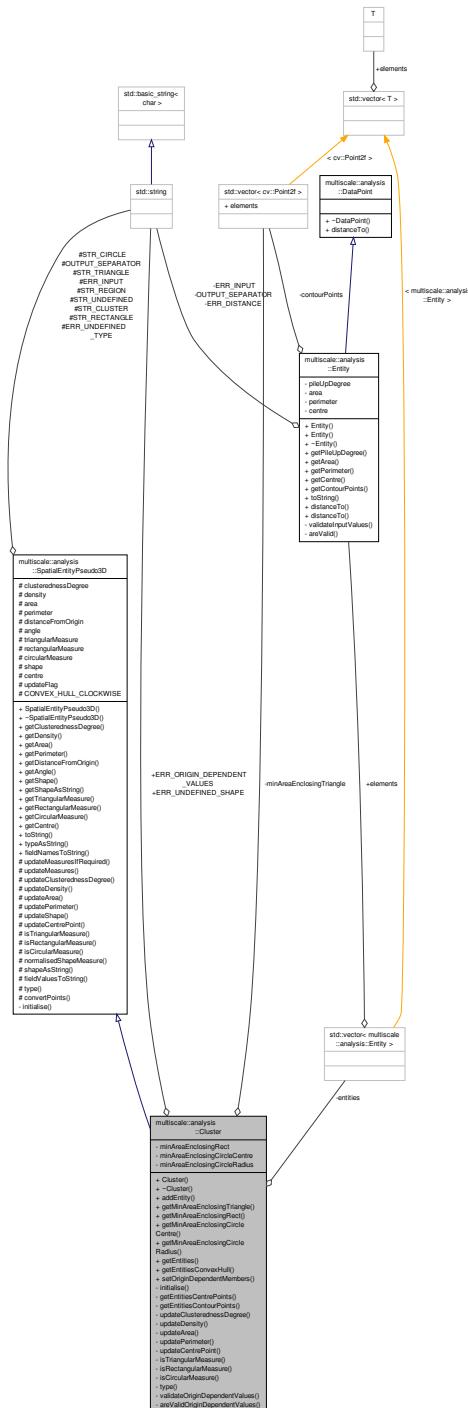
Class for representing a cluster of entities in an image.

```
#include <Cluster.hpp>
```

Inheritance diagram for multiscale::analysis::Cluster:



Collaboration diagram for multiscale::analysis::Cluster:



## Public Member Functions

- `Cluster ()`
- `~Cluster ()`
- void `addEntity (const Entity &entity)`

*Add a new entity to the cluster.*

- `std::vector< cv::Point2f > getMinAreaEnclosingTriangle ()`

- Get the minimum area enclosing triangle.
- cv::RotatedRect **getMinAreaEnclosingRect ()**  
Get the minimum area enclosing rectangle.
- cv::Point2f **getMinAreaEnclosingCircleCentre ()**  
Get the minimum area enclosing circle centre.
- float **getMinAreaEnclosingCircleRadius ()**  
Get the minimum area enclosing circle radius.
- std::vector< Entity > **getEntities () const**  
Get the collection of underlying entities.
- std::vector< cv::Point2f > **getEntitiesConvexHull ()**  
Get the convex hull enclosing the collection of entities' contour points.
- void **setOriginDependentMembers (double distanceFromOrigin, double angleWrtOrigin)**  
Set the values of the origin dependent members.

## Static Public Attributes

- static const std::string **ERR\_UNDEFINED\_SHAPE** = "The shape of the given cluster is undefined."
- static const std::string **ERR\_ORIGIN\_DEPENDENT\_VALUES** = "The origin dependent values are invalid (i.e. negative)."

## Private Member Functions

- void **initialise ()**  
Initialisation function for the class.
- std::vector< cv::Point2f > **getEntitiesCentrePoints ()**  
Get the collection of entities' centres.
- std::vector< cv::Point2f > **getEntitiesContourPoints ()**  
Get the collection of entities' contour points.
- void **updateClusterednessDegree () override**  
Update the value of the clusteredness degree.
- void **updateDensity () override**  
Update the value of the pile up degree.
- void **updateArea () override**  
Update the value of the area.
- void **updatePerimeter () override**  
Update the value of the perimeter.
- void **updateCentrePoint () override**  
Update the point defining the centre of the cluster.
- double **isTriangularMeasure () override**  
Get the measure that the cluster has a triangular shape.
- double **isRectangularMeasure () override**  
Get the measure that the cluster has a rectangular shape.
- double **isCircularMeasure () override**  
Get the measure that the cluster has a circular shape.
- SpatialEntityPseudo3DType **type () override**  
Return the type of the pseudo 3D spatial entity.
- void **validateOriginDependentValues (double distanceFromOrigin, double angleWrtOrigin)**  
Validate the origin dependent values (i.e. non-negative)
- bool **areValidOriginDependentValues (double distanceFromOrigin, double angleWrtOrigin)**  
Check if the origin dependent values are valid (i.e. non-negative)

## Private Attributes

- std::vector< cv::Point2f > `minAreaEnclosingTriangle`
- cv::RotatedRect `minAreaEnclosingRect`
- cv::Point2f `minAreaEnclosingCircleCentre`
- float `minAreaEnclosingCircleRadius`
- std::vector< Entity > `entities`

## Additional Inherited Members

### 7.45.1 Detailed Description

Class for representing a cluster of entities in an image.

Definition at line 19 of file Cluster.hpp.

### 7.45.2 Constructor & Destructor Documentation

#### 7.45.2.1 Cluster::Cluster( )

Definition at line 11 of file Cluster.cpp.

References initialise().

#### 7.45.2.2 Cluster::~Cluster( )

Definition at line 15 of file Cluster.cpp.

### 7.45.3 Member Function Documentation

#### 7.45.3.1 void Cluster::addEntity( const Entity & entity )

Add a new entity to the cluster.

Definition at line 17 of file Cluster.cpp.

References entities, and multiscale::analysis::SpatialEntityPseudo3D::updateFlag.

#### 7.45.3.2 bool Cluster::isValidOriginDependentValues( double *distanceFromOrigin*, double *angleWrtOrigin* ) [private]

Check if the origin dependent values are valid (i.e. non-negative)

##### Parameters

|                           |                                  |
|---------------------------|----------------------------------|
| <i>distanceFromOrigin</i> | Distance from the origin         |
| <i>angleWrtOrigin</i>     | Angle with respect to the origin |

Definition at line 199 of file Cluster.cpp.

References multiscale::Numeric::greaterOrEqual().

Referenced by validateOriginDependentValues().

#### 7.45.3.3 `std::vector< Entity > Cluster::getEntities( ) const`

Get the collection of underlying entities.

Definition at line 47 of file Cluster.cpp.

References entities.

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterToImage().

#### 7.45.3.4 `std::vector< cv::Point2f > Cluster::getEntitiesCentrePoints( ) [private]`

Get the collection of entities' centres.

Definition at line 84 of file Cluster.cpp.

References entities.

#### 7.45.3.5 `std::vector< cv::Point2f > Cluster::getEntitiesContourPoints( ) [private]`

Get the collection of entities' contour points.

Definition at line 94 of file Cluster.cpp.

References entities.

Referenced by getEntitiesConvexHull(), isCircularMeasure(), and isRectangularMeasure().

#### 7.45.3.6 `std::vector< cv::Point2f > Cluster::getEntitiesConvexHull( )`

Get the convex hull enclosing the collection of entities' contour points.

Definition at line 51 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::CONVEX\_HULL\_CLOCKWISE, entities, and get← EntitiesContourPoints().

Referenced by multiscale::analysis::ClusterDetector::getClusterConvexHull(), isTriangularMeasure(), update← CentrePoint(), and updatePerimeter().

#### 7.45.3.7 `cv::Point2f Cluster::getMinAreaEnclosingCircleCentre( )`

Get the minimum area enclosing circle centre.

Definition at line 35 of file Cluster.cpp.

References minAreaEnclosingCircleCentre, and multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIf← Required().

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterCircularShape().

#### 7.45.3.8 `float Cluster::getMinAreaEnclosingCircleRadius( )`

Get the minimum area enclosing circle radius.

Definition at line 41 of file Cluster.cpp.

References minAreaEnclosingCircleRadius, and multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIf← Required().

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterCircularShape().

**7.45.3.9 cv::RotatedRect Cluster::getMinAreaEnclosingRect( )**

Get the minimum area enclosing rectangle.

Definition at line 29 of file Cluster.cpp.

References minAreaEnclosingRect, and multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIfRequired().

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterRectangularShape().

**7.45.3.10 std::vector< cv::Point2f > Cluster::getMinAreaEnclosingTriangle( )**

Get the minimum area enclosing triangle.

Definition at line 23 of file Cluster.cpp.

References minAreaEnclosingTriangle, and multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIfRequired().

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterTriangularShape().

**7.45.3.11 void Cluster::initialise( ) [private]**

Initialisation function for the class.

Definition at line 69 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::angle, multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree, multiscale::analysis::SpatialEntityPseudo3D::density, multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin, entities, minAreaEnclosingCircleRadius, minAreaEnclosingTriangle, and multiscale::analysis::SpatialEntityPseudo3D::updateFlag.

Referenced by Cluster().

**7.45.3.12 double Cluster::isCircularMeasure( ) [override], [private], [virtual]**

Get the measure that the cluster has a circular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 178 of file Cluster.cpp.

References getEntitiesContourPoints(), minAreaEnclosingCircleCentre, minAreaEnclosingCircleRadius, multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), and multiscale::Geometry2D::PI.

**7.45.3.13 double Cluster::isRectangularMeasure( ) [override], [private], [virtual]**

Get the measure that the cluster has a rectangular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 167 of file Cluster.cpp.

References getEntitiesContourPoints(), minAreaEnclosingRect, and multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure().

**7.45.3.14 double Cluster::isTriangularMeasure( ) [override], [private], [virtual]**

Get the measure that the cluster has a triangular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 159 of file Cluster.cpp.

References multiscale::MinEnclosingTriangleFinder::find(), getEntitiesConvexHull(), minAreaEnclosingTriangle, and multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure().

#### 7.45.3.15 void Cluster::setOriginDependentMembers ( double *distanceFromOrigin*, double *angleWrtOrigin* )

Set the values of the origin dependent members.

##### Parameters

|                           |                                  |
|---------------------------|----------------------------------|
| <i>distanceFromOrigin</i> | Distance from the origin         |
| <i>angleWrtOrigin</i>     | Angle with respect to the origin |

Definition at line 62 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::angle, multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin, and validateOriginDependentValues().

Referenced by multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

#### 7.45.3.16 SpatialEntityPseudo3DType Cluster::type ( ) [override], [private], [virtual]

Return the type of the pseudo 3D spatial entity.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 189 of file Cluster.cpp.

References multiscale::analysis::Cluster.

#### 7.45.3.17 void Cluster::updateArea ( ) [override], [private], [virtual]

Update the value of the area.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 136 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::area, and entities.

#### 7.45.3.18 void Cluster::updateCentrePoint ( ) [override], [private], [virtual]

Update the point defining the centre of the cluster.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 150 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::centre, and getEntitiesConvexHull().

#### 7.45.3.19 void Cluster::updateClusterednessDegree ( ) [override], [private], [virtual]

Update the value of the clusteredness degree.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 106 of file Cluster.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree, and entities.

**7.45.3.20 void Cluster::updateDensity( ) [override], [private], [virtual]**

Update the value of the pile up degree.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 126 of file Cluster.cpp.

References [multiscale::analysis::SpatialEntityPseudo3D::density](#), and [entities](#).

**7.45.3.21 void Cluster::updatePerimeter( ) [override], [private], [virtual]**

Update the value of the perimeter.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 144 of file Cluster.cpp.

References [getEntitiesConvexHull\(\)](#), and [multiscale::analysis::SpatialEntityPseudo3D::perimeter](#).

**7.45.3.22 void Cluster::validateOriginDependentValues( double *distanceFromOrigin*, double *angleWrtOrigin* ) [private]**

Validate the origin dependent values (i.e. non-negative)

**Parameters**

|                           |                                  |
|---------------------------|----------------------------------|
| <i>distanceFromOrigin</i> | Distance from the origin         |
| <i>angleWrtOrigin</i>     | Angle with respect to the origin |

Definition at line 193 of file Cluster.cpp.

References [isValidOriginDependentValues\(\)](#), [ERR\\_ORIGIN\\_DEPENDENT\\_VALUES](#), and [MS\\_throw](#).

Referenced by [setOriginDependentMembers\(\)](#).

## 7.45.4 Member Data Documentation

**7.45.4.1 std::vector<Entity> multiscale::analysis::Cluster::entities [private]**

Entities which belong to this cluster

Definition at line 35 of file Cluster.hpp.

Referenced by [addEntity\(\)](#), [getEntities\(\)](#), [getEntitiesCentrePoints\(\)](#), [getEntitiesContourPoints\(\)](#), [getEntitiesConvexHull\(\)](#), [initialise\(\)](#), [updateArea\(\)](#), [updateClusterednessDegree\(\)](#), and [updateDensity\(\)](#).

**7.45.4.2 const std::string Cluster::ERR\_ORIGIN\_DEPENDENT\_VALUES = "The origin dependent values are invalid (i.e. negative)." [static]**

Definition at line 126 of file Cluster.hpp.

Referenced by [validateOriginDependentValues\(\)](#).

**7.45.4.3 const std::string Cluster::ERR\_UNDEFINED\_SHAPE = "The shape of the given cluster is undefined." [static]**

Definition at line 125 of file Cluster.hpp.

Referenced by [multiscale::analysis::SimulationClusterDetector::outputClusterShape\(\)](#).

#### 7.45.4.4 cv::Point2f multiscale::analysis::Cluster::minAreaEnclosingCircleCentre [private]

The minimum area enclosing circle centre point

Definition at line 30 of file Cluster.hpp.

Referenced by getMinAreaEnclosingCircleCentre(), and isCircularMeasure().

#### 7.45.4.5 float multiscale::analysis::Cluster::minAreaEnclosingCircleRadius [private]

The minimum area enclosing circle radius

Definition at line 32 of file Cluster.hpp.

Referenced by getMinAreaEnclosingCircleRadius(), initialise(), and isCircularMeasure().

#### 7.45.4.6 cv::RotatedRect multiscale::analysis::Cluster::minAreaEnclosingRect [private]

The minimum area enclosing rectangle

Definition at line 27 of file Cluster.hpp.

Referenced by getMinAreaEnclosingRect(), and isRectangularMeasure().

#### 7.45.4.7 std::vector<cv::Point2f> multiscale::analysis::Cluster::minAreaEnclosingTriangle [private]

The minimum area enclosing triangle

Definition at line 24 of file Cluster.hpp.

Referenced by getMinAreaEnclosingTriangle(), initialise(), and isTriangularMeasure().

The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[Cluster.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[Cluster.cpp](#)

## 7.46 multiscale::analysis::ClusterDetector Class Reference

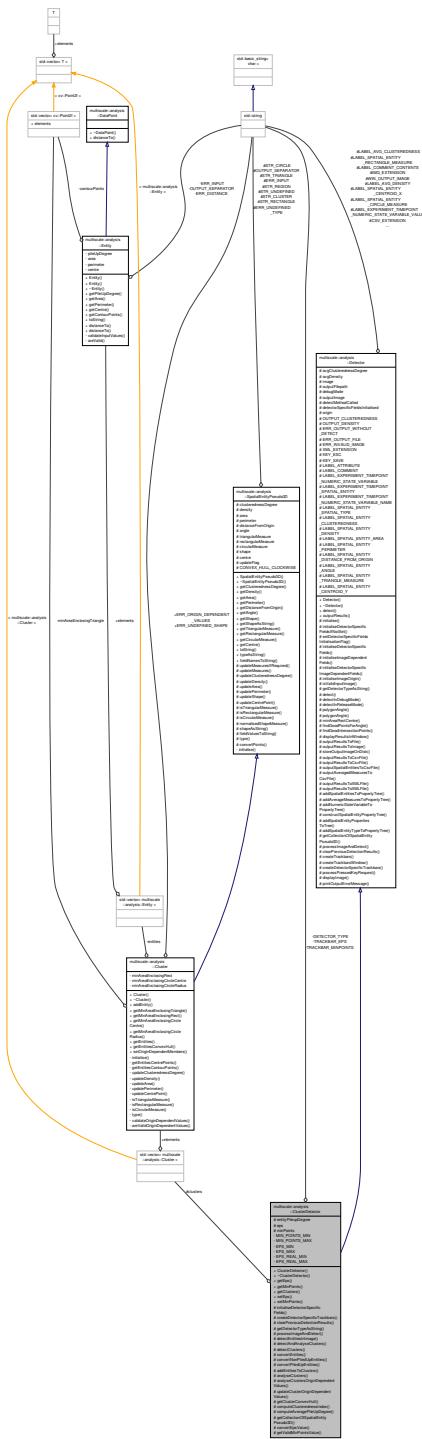
Class for detecting clusters in 2D images.

```
#include <ClusterDetector.hpp>
```

## Inheritance diagram for multiscale::analysis::ClusterDetector:



## Collaboration diagram for multiscale::analysis::ClusterDetector:



## Public Member Functions

- `ClusterDetector` (int maxPileupNumber, double maxPileupIntensity, bool `debugMode`=false)
  - virtual `~ClusterDetector` ()
  - double `getEps` ()

*Get the value of the clustering algorithm parameter  $\text{eps}$ .*

- int **getMinPoints** ()

- `std::vector< Cluster > const & getClusters ()`  
*Get the value of the clustering algorithm parameter MinPoints.*
- `void setEps (double eps)`  
*Set the value of the clustering algorithm parameter eps.*
- `void setMinPoints (int minPoints)`  
*Set the value of the clustering algorithm parameter MinPoints.*

## Protected Member Functions

- `void initialiseDetectorSpecificFields () override`  
*Initialise clustering values.*
- `void createDetectorSpecificTrackbars () override`  
*Create the trackbars.*
- `void clearPreviousDetectionResults () override`  
*Clear the clusters from the previous detection.*
- `std::string getDetectorTypeAsString () override`  
*Get the type of the detector as a std::string.*
- `void processImageAndDetect () override`  
*Process the provided image and detect clusters in it.*
- `virtual void detectEntitiesInImage (std::vector< Entity > &entities)=0`  
*Detect the entities in the image.*
- `void detectAndAnalyseClusters (const std::vector< Entity > &entities, std::vector< Cluster > &clusters)`  
*Detect and analyse the clusters of entities in the image.*
- `void detectClusters (const std::vector< Entity > &entities, std::vector< int > &clusterIndexes, int &nrofClusters)`  
*Detect the clusters of entities in the image.*
- `std::vector< std::shared_ptr< DataPoint > > convertEntities (const std::vector< Entity > &entities)`  
*Convert the entities to the format required by the DBSCAN class.*
- `void convertNonPiledUpEntities (const std::vector< Entity > &entities, std::vector< std::shared_ptr< DataPoint > > &dataPoints)`  
*Convert the non pile up entities to the format required by the DBSCAN class.*
- `void convertPiledUpEntities (const std::vector< Entity > &entities, std::vector< std::shared_ptr< DataPoint > > &dataPoints)`  
*Convert the entities to the required format by the DBSCAN class.*
- `void addEntitiesToClusters (const std::vector< Entity > &entities, const std::vector< int > &clusterIndexes, int nrOfClusters, std::vector< Cluster > &clusters)`  
*Add the entities to the clusters as indicated by the clusterIndexes parameter.*
- `void analyseClusters (std::vector< Cluster > &clusters)`  
*Analyse the clusters.*
- `void analyseClustersOriginDependentValues (std::vector< Cluster > &clusters)`  
*Analyse the clusters and compute the origin dependent values.*
- `void updateClusterOriginDependentValues (Cluster &cluster, const std::vector< cv::Point > &clusterConvexHull)`  
*Update the cluster and compute the origin dependent values considering the convex hull.*
- `std::vector< cv::Point > getClusterConvexHull (Cluster &cluster)`  
*Return the convex hull of the given cluster.*
- `double computeClusterednessIndex (const std::vector< Cluster > &clusters)`  
*Compute the clusteredness index for all the entities detected in the image.*
- `double computeAveragePileUpDegree (std::vector< Cluster > &clusters)`

- Compute the average pile up degree for all entities in the image.*
- std::vector< std::shared\_ptr< SpatialEntityPseudo3D > > `getCollectionOfSpatialEntityPseudo3D () override`  
*Get the collection of clusters detected in the image.*
  - double `convertEpsValue ()`  
*Convert the value of eps from integer to double.*
  - int `getValidMinPointsValue ()`  
*Return non-zero value for minPoints.*

## Protected Attributes

- double `entityPileupDegree`
- int `eps`
- int `minPoints`
- std::vector< `Cluster` > `clusters`

## Static Private Attributes

- static const std::string `DETECTOR_TYPE` = "Clusters"
- static const std::string `TRACKBAR_EPS` = "Eps (Multiplied by 10)"
- static const std::string `TRACKBAR_MINPOINTS` = "Minimum number of points"
- static const int `MIN_POINTS_MIN` = 0
- static const int `MIN_POINTS_MAX` = 100
- static const int `EPS_MIN` = 0
- static const int `EPS_MAX` = 10000
- static const int `EPS_REAL_MIN` = 0
- static const int `EPS_REAL_MAX` = 1000

## Additional Inherited Members

### 7.46.1 Detailed Description

Class for detecting clusters in 2D images.

Definition at line 17 of file ClusterDetector.hpp.

### 7.46.2 Constructor & Destructor Documentation

#### 7.46.2.1 ClusterDetector::ClusterDetector ( int `maxPileupNumber`, double `maxPileupIntensity`, bool `debugMode = false` )

##### Parameters

|                                       |                                                                                  |
|---------------------------------------|----------------------------------------------------------------------------------|
| <code>debugMode</code>                | Flag indicating if detector should run in debug mode or not                      |
| <code>maxPileup←<br/>Number</code>    | The maximum number of entities which can occupy a grid position at the same time |
| <code>maxPileup←<br/>Intensity</code> | The grayscale intensity of a maximally piled up grid position                    |

Definition at line 14 of file ClusterDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, entityPileupDegree, eps, and minPoints.

## 7.46.2.2 ClusterDetector::~ClusterDetector( ) [virtual]

Definition at line 25 of file ClusterDetector.cpp.

## 7.46.3 Member Function Documentation

## 7.46.3.1 void ClusterDetector::addEntitiesToClusters ( const std::vector&lt; Entity &gt; &amp; entities, const std::vector&lt; int &gt; &amp; clusterIndexes, int nrOfClusters, std::vector&lt; Cluster &gt; &amp; clusters ) [protected]

Add the entities to the clusters as indicated by the clusterIndexes parameter.

Add the entities to the clusters as indicated by the clusterIndexes parameter

The "noise" cluster will be ignored.

## Parameters

|                       |                                                            |
|-----------------------|------------------------------------------------------------|
| <i>entities</i>       | Entities detected in the image                             |
| <i>clusterIndexes</i> | Indexes to which cluster each entity belongs               |
| <i>nrOfClusters</i>   | Total number of clusters                                   |
| <i>clusters</i>       | Collection of clusters, each one with the updated measures |

Definition at line 127 of file ClusterDetector.cpp.

Referenced by detectAndAnalyseClusters().

## 7.46.3.2 void ClusterDetector::analyseClusters ( std::vector&lt; Cluster &gt; &amp; clusters ) [protected]

Analyse the clusters.

Analyse the clusters and compute the angle and distance from the centre, average clusteredness degree and pile up degree

## Parameters

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <i>clusters</i> | Collection of clusters, each one with the updated measures |
|-----------------|------------------------------------------------------------|

Definition at line 146 of file ClusterDetector.cpp.

References analyseClustersOriginDependentValues(), multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, computeAveragePileUpDegree(), and computeClusterednessIndex().

Referenced by detectAndAnalyseClusters().

## 7.46.3.3 void ClusterDetector::analyseClustersOriginDependentValues ( std::vector&lt; Cluster &gt; &amp; clusters ) [protected]

Analyse the clusters and compute the origin dependent values.

The values which depend on the origin point are the distance of the cluster from the centre and the angle

## Parameters

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <i>clusters</i> | Collection of clusters, each one with the updated measures |
|-----------------|------------------------------------------------------------|

Definition at line 153 of file ClusterDetector.cpp.

References getClusterConvexHull(), and updateClusterOriginDependentValues().

Referenced by analyseClusters().

#### 7.46.3.4 void ClusterDetector::clearPreviousDetectionResults( ) [override], [protected], [virtual]

Clear the clusters from the previous detection.

Implements [multiscale::analysis::Detector](#).

Definition at line 63 of file ClusterDetector.cpp.

References clusters.

#### 7.46.3.5 double ClusterDetector::computeAveragePileUpDegree ( std::vector< Cluster > & clusters ) [protected]

Compute the average pile up degree for all entities in the image.

Compute the average pile up degree for all entities in the image as the sum of the average pile up degrees of all clusters divided by the number of clusters

Parameters

|                 |                                            |
|-----------------|--------------------------------------------|
| <i>clusters</i> | Clusters of entities detected in the image |
|-----------------|--------------------------------------------|

Definition at line 188 of file ClusterDetector.cpp.

Referenced by analyseClusters().

#### 7.46.3.6 double ClusterDetector::computeClusterednessIndex ( const std::vector< Cluster > & clusters ) [protected]

Compute the clusteredness index for all the entities detected in the image.

Compute the clusteredness index for all the entities detected in the image using [Silhouette](#) cluster validity index

Parameters

|                 |                                                            |
|-----------------|------------------------------------------------------------|
| <i>clusters</i> | Collection of clusters, each one with the updated measures |
|-----------------|------------------------------------------------------------|

Definition at line 183 of file ClusterDetector.cpp.

References multiscale::analysis::Silhouette::computeOverallAverageMeasure().

Referenced by analyseClusters().

#### 7.46.3.7 std::vector< std::shared\_ptr< DataPoint > > ClusterDetector::convertEntities ( const std::vector< Entity > & entities ) [protected]

Convert the entities to the format required by the [DBSCAN](#) class.

Parameters

|                 |                                |
|-----------------|--------------------------------|
| <i>entities</i> | Entities detected in the image |
|-----------------|--------------------------------|

Definition at line 99 of file ClusterDetector.cpp.

References convertNonPiledUpEntities(), and convertPiledUpEntities().

Referenced by detectClusters().

#### 7.46.3.8 double ClusterDetector::convertEpsValue ( ) [protected]

Convert the value of eps from integer to double.

Definition at line 211 of file ClusterDetector.cpp.

References eps, EPS\_MAX, EPS\_MIN, EPS\_REAL\_MAX, and EPS\_REAL\_MIN.

Referenced by detectClusters(), and getEps().

7.46.3.9 void ClusterDetector::convertNonPiledUpEntities ( const std::vector< Entity > & entities, std::vector< std::shared\_ptr< DataPoint > > & dataPoints ) [protected]

Convert the non pile up entities to the format required by the [DBSCAN](#) class.

**Parameters**

|                   |                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------|
| <i>entities</i>   | Entities detected in the image                                                                 |
| <i>dataPoints</i> | Collection of <a href="#">DataPoint</a> instances required by the <a href="#">DBSCAN</a> class |

Definition at line 108 of file ClusterDetector.cpp.

Referenced by convertEntities().

7.46.3.10 void ClusterDetector::convertPiledUpEntities ( const std::vector< Entity > & entities, std::vector< std::shared\_ptr< DataPoint > > & dataPoints ) [protected]

Convert the entities to the required format by the [DBSCAN](#) class.

**Parameters**

|                   |                                                                                                |
|-------------------|------------------------------------------------------------------------------------------------|
| <i>entities</i>   | Entities detected in the image                                                                 |
| <i>dataPoints</i> | Collection of <a href="#">DataPoint</a> instances required by the <a href="#">DBSCAN</a> class |

Definition at line 115 of file ClusterDetector.cpp.

Referenced by convertEntities().

7.46.3.11 void ClusterDetector::createDetectorSpecificTrackbars ( ) [override], [protected], [virtual]

Create the trackbars.

Implements [multiscale::analysis::Detector](#).

Definition at line 58 of file ClusterDetector.cpp.

References eps, EPS\_MAX, MIN\_POINTS\_MAX, minPoints, TRACKBAR\_EPS, TRACKBAR\_MINPOINTS, and [multiscale::analysis::Detector::WIN\\_OUTPUT\\_IMAGE](#).

7.46.3.12 void ClusterDetector::detectAndAnalyseClusters ( const std::vector< Entity > & entities, std::vector< Cluster > & clusters ) [protected]

Detect and analyse the clusters of entities in the image.

Detect and analyse the clusters of entities in the image

Remark: The "noise" cluster will be ignored.

**Parameters**

|                 |                                            |
|-----------------|--------------------------------------------|
| <i>entities</i> | Entities detected in the image             |
| <i>clusters</i> | Clusters of entities detected in the image |

Definition at line 78 of file ClusterDetector.cpp.

References addEntitiesToClusters(), analyseClusters(), [multiscale::analysis::DBSCAN::CLUSTERING\\_UNCLASIFIED](#), and detectClusters().

Referenced by processImageAndDetect().

**7.46.3.13 void ClusterDetector::detectClusters ( const std::vector< Entity > & entities, std::vector< int > & clusterIndexes, int & nrOfClusters ) [protected]**

Detect the clusters of entities in the image.

Detect the clusters of entities in the image using Density Based scan (DBscan) clustering algorithm Clusters start from index 1, because cluster 0 contains only noise data/points.

#### Parameters

|                       |                                              |
|-----------------------|----------------------------------------------|
| <i>entities</i>       | Entities detected in the image               |
| <i>clusterIndexes</i> | Indexes to which cluster each entity belongs |
| <i>nrOfClusters</i>   | Total number of clusters                     |

Definition at line 88 of file ClusterDetector.cpp.

References convertEntities(), convertEpsValue(), getValidMinPointsValue(), and multiscale::analysis::DBSCAN::run().

Referenced by detectAndAnalyseClusters().

**7.46.3.14 virtual void multiscale::analysis::ClusterDetector::detectEntitiesInImage ( std::vector< Entity > & entities ) [protected], [pure virtual]**

Detect the entities in the image.

Detect the entities in the image, compute their centre point and degree of pile up

#### Parameters

|                 |                                |
|-----------------|--------------------------------|
| <i>entities</i> | Entities detected in the image |
|-----------------|--------------------------------|

Implemented in [multiscale::analysis::SimulationClusterDetector](#).

Referenced by processImageAndDetect().

**7.46.3.15 std::vector< cv::Point > ClusterDetector::getClusterConvexHull ( Cluster & cluster ) [protected]**

Return the convex hull of the given cluster.

#### Parameters

|                |                   |
|----------------|-------------------|
| <i>cluster</i> | The given cluster |
|----------------|-------------------|

Definition at line 173 of file ClusterDetector.cpp.

References multiscale::analysis::Cluster::getEntitiesConvexHull().

Referenced by analyseClustersOriginDependentValues().

**7.46.3.16 std::vector< Cluster > const & ClusterDetector::getClusters ( )**

Get a const reference to the std::vector of detected clusters.

Definition at line 35 of file ClusterDetector.cpp.

References clusters.

**7.46.3.17 std::vector< std::shared\_ptr< SpatialEntityPseudo3D > > ClusterDetector::getCollectionOfSpatialEntityPseudo3D ( ) [override], [protected], [virtual]**

Get the collection of clusters detected in the image.

Implements [multiscale::analysis::Detector](#).

Definition at line 201 of file ClusterDetector.cpp.

References multiscale::analysis::Cluster, and clusters.

#### 7.46.3.18 std::string ClusterDetector::getDetectorTypeAsString( ) [override], [protected], [virtual]

Get the type of the detector as a std::string.

Implements [multiscale::analysis::Detector](#).

Definition at line 67 of file ClusterDetector.cpp.

References DETECTOR\_TYPE.

#### 7.46.3.19 double ClusterDetector::getEps( )

Get the value of the clustering algorithm parameter eps.

Definition at line 27 of file ClusterDetector.cpp.

References convertEpsValue().

Referenced by saveDetectorParameterValues().

#### 7.46.3.20 int ClusterDetector::getMinPoints( )

Get the value of the clustering algorithm parameter MinPoints.

Definition at line 31 of file ClusterDetector.cpp.

References minPoints.

Referenced by saveDetectorParameterValues().

#### 7.46.3.21 int ClusterDetector::getValidMinPointsValue( ) [protected]

Return non-zero value for minPoints.

Definition at line 219 of file ClusterDetector.cpp.

References minPoints.

Referenced by detectClusters().

#### 7.46.3.22 void ClusterDetector::initialiseDetectorSpecificFields( ) [override], [protected], [virtual]

Initialise clustering values.

Implements [multiscale::analysis::Detector](#).

Definition at line 53 of file ClusterDetector.cpp.

References eps, and minPoints.

#### 7.46.3.23 void ClusterDetector::processImageAndDetect( ) [override], [protected], [virtual]

Process the provided image and detect clusters in it.

Implements [multiscale::analysis::Detector](#).

Definition at line 71 of file ClusterDetector.cpp.

References clusters, detectAndAnalyseClusters(), and detectEntitiesInImage().

#### 7.46.3.24 void ClusterDetector::setEps ( double eps )

Set the value of the clustering algorithm parameter eps.

**Parameters**

|            |                                                 |
|------------|-------------------------------------------------|
| <i>eps</i> | Value of the clustering algorithm parameter eps |
|------------|-------------------------------------------------|

Definition at line 39 of file ClusterDetector.cpp.

References *eps*, EPS\_MAX, EPS\_MIN, EPS\_REAL\_MAX, EPS\_REAL\_MIN, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

#### 7.46.3.25 void ClusterDetector::setMinPoints ( int minPoints )

Set the value of the clustering algorithm parameter MinPoints.

**Parameters**

|                  |                                                       |
|------------------|-------------------------------------------------------|
| <i>minPoints</i> | Value of the clustering algorithm parameter MinPoints |
|------------------|-------------------------------------------------------|

Definition at line 47 of file ClusterDetector.cpp.

References *minPoints*, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

#### 7.46.3.26 void ClusterDetector::updateClusterOriginDependentValues ( Cluster & cluster, const std::vector< cv::Point > & clusterConvexHull ) [protected]

Update the cluster and compute the origin dependent values considering the convex hull.

The values which depend on the origin point are the distance of the cluster from the centre and the angle

**Parameters**

|                          |                            |
|--------------------------|----------------------------|
| <i>cluster</i>           | Cluster                    |
| <i>clusterConvexHull</i> | Convex hull of the cluster |

Definition at line 163 of file ClusterDetector.cpp.

References multiscale::Geometry2D::distanceBtwPoints(), multiscale::Geometry2D::minimumDistancePointIndex(), multiscale::analysis::Detector::origin, multiscale::analysis::Detector::polygonAngle(), and multiscale::analysis::Cluster::setOriginDependentMembers().

Referenced by analyseClustersOriginDependentValues().

### 7.46.4 Member Data Documentation

#### 7.46.4.1 std::vector<Cluster> multiscale::analysis::ClusterDetector::clusters [protected]

Clusters found in the image

Definition at line 29 of file ClusterDetector.hpp.

Referenced by clearPreviousDetectionResults(), getClusters(), getCollectionOfSpatialEntityPseudo3D(), multiscale::analysis::SimulationClusterDetector::outputResultsToImage(), and processImageAndDetect().

7.46.4.2 `const std::string ClusterDetector::DETECTOR_TYPE = "Clusters" [static], [private]`

Definition at line 206 of file ClusterDetector.hpp.

Referenced by `getDetectorTypeAsString()`.

7.46.4.3 `double multiscale::analysis::ClusterDetector::entityPileupDegree [protected]`

The pile up degree (intensity) of a grid position occupied by only one entity

Definition at line 21 of file ClusterDetector.hpp.

Referenced by `ClusterDetector()`, and `multiscale::analysis::SimulationClusterDetector::computePileUpDegreeAtPosition()`.

7.46.4.4 `int multiscale::analysis::ClusterDetector::eps [protected]`

[DBSCAN](#) algorithm parameter for specifying the maximum radius of the neighbourhood

Definition at line 24 of file ClusterDetector.hpp.

Referenced by `ClusterDetector()`, `convertEpsValue()`, `createDetectorSpecificTrackbars()`, `initialiseDetectorSpecificFields()`, and `setEps()`.

7.46.4.5 `const int ClusterDetector::EPS_MAX = 10000 [static], [private]`

Definition at line 215 of file ClusterDetector.hpp.

Referenced by `convertEpsValue()`, `createDetectorSpecificTrackbars()`, and `setEps()`.

7.46.4.6 `const int ClusterDetector::EPS_MIN = 0 [static], [private]`

Definition at line 214 of file ClusterDetector.hpp.

Referenced by `convertEpsValue()`, and `setEps()`.

7.46.4.7 `const int ClusterDetector::EPS_REAL_MAX = 1000 [static], [private]`

Definition at line 217 of file ClusterDetector.hpp.

Referenced by `convertEpsValue()`, and `setEps()`.

7.46.4.8 `const int ClusterDetector::EPS_REAL_MIN = 0 [static], [private]`

Definition at line 216 of file ClusterDetector.hpp.

Referenced by `convertEpsValue()`, and `setEps()`.

7.46.4.9 `const int ClusterDetector::MIN_POINTS_MAX = 100 [static], [private]`

Definition at line 212 of file ClusterDetector.hpp.

Referenced by `createDetectorSpecificTrackbars()`.

7.46.4.10 `const int ClusterDetector::MIN_POINTS_MIN = 0 [static], [private]`

Definition at line 211 of file ClusterDetector.hpp.

7.46.4.11 int multiscale::analysis::ClusterDetector::minPoints [protected]

DBSCAN algorithm parameter for specifying the minimum number of points in an eps-neighbourhood of that point

Definition at line 26 of file ClusterDetector.hpp.

Referenced by ClusterDetector(), createDetectorSpecificTrackbars(), getMinPoints(), getValidMinPointsValue(), initialiseDetectorSpecificFields(), and setMinPoints().

7.46.4.12 const std::string ClusterDetector::TRACKBAR\_EPS = "Eps (Multiplied by 10)" [static], [private]

Definition at line 208 of file ClusterDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.46.4.13 const std::string ClusterDetector::TRACKBAR\_MINPOINTS = "Minimum number of points" [static], [private]

Definition at line 209 of file ClusterDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[ClusterDetector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[ClusterDetector.cpp](#)

## 7.47 multiscale::verification::CommandLineModelChecking Class Reference

Class for running model checkers from the command line.

```
#include <CommandLineModelChecking.hpp>
```

## Collaboration diagram for multiscale::verification::CommandLineModelChecking



## Public Member Functions

- `CommandLineModelChecking ()`
  - `~CommandLineModelChecking ()`
  - `void initialise (int argc, char **argv)`

*Initialise the class with the given command line arguments.*

- void execute ()

*Execute the model checking task.*

## Private Member Functions

- `bool areValidArguments (int argc, char **argv)`

*Check if the provided command line arguments are valid.*
- `void initialiseAllowedArgumentsConfiguration ()`

*Initialise the configuration of allowed command line arguments.*
- `void initialiseRequiredArgumentsConfiguration ()`

*Initialise the configuration of required command line arguments.*
- `void initialiseOptionalArgumentsConfiguration ()`

*Initialise the configuration of optional command line arguments.*
- `void initialiseModelCheckerTypeSpecificArgumentsConfiguration ()`

*Initialise the configuration of model checker type specific command line arguments.*
- `po::options_description initialiseStatisticalModelCheckerArgumentsConfiguration ()`

*Initialise the configuration of the statistical model checker command line arguments.*
- `po::options_description initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration ()`

*Initialise the configuration of the approximate probabilistic model checker command line arguments.*
- `po::options_description initialiseBayesianModelCheckerArgumentsConfiguration ()`

*Initialise the configuration of the Bayesian model checker command line arguments.*
- `po::options_description initialiseApproximateBayesianModelCheckerArgumentsConfiguration ()`

*Initialise the configuration of the approximate Bayesian model checker command line arguments.*
- `bool areValidArgumentsConsideringConfiguration (int argc, char **argv)`

*Check if the provided command line arguments are valid.*
- `po::parsed_options parseAndStoreArgumentsValues (int argc, char **argv)`

*Parse and store the command line arguments' values in a variables map.*
- `bool areInvalidExecutionArguments (const po::parsed_options &parsedArguments)`

*Check if any invalid execution arguments were provided.*
- `bool isHelpArgumentPresent ()`

*Check if the help command line argument is present.*
- `void handleHelpRequest ()`

*Handle the help request i.e. if the –help flag was provided.*
- `void printHelpMessage ()`

*Print help message to the console.*
- `void printHelpIntroMessage ()`

*Print the help intro message to the console.*
- `void printHelpContentsMessage ()`

*Print the help contents message to the console.*
- `void printHelpClosingMessage ()`

*Print the help closing message to the console.*
- `bool areUnrecognizedArgumentsPresent (const po::parsed_options &parsedArguments)`

*Check if any unrecognized command line arguments are present.*
- `bool areInvalidModelCheckingArguments ()`

*Check if any invalid model checker type dependent arguments are present.*
- `bool areInvalidModelCheckingArgumentsPresent ()`

*Check if any model checker type dependent arguments are invalid.*
- `void removeRequiredArguments (po::variables_map &variablesMap)`

*Remove the required arguments from the given variables\_map.*
- `void removeOptionalArguments (po::variables_map &variablesMap)`

*Remove the optional arguments from the given variables\_map.*

- `bool areInvalidModelCheckingTypeSpecificArguments (unsigned int modelCheckerType, po::variables_map &variablesMap)`  
*Check if the model checking type specific arguments from the given variables\_map are invalid.*
- `bool areModelCheckingTypeSpecificArgumentsPresent (unsigned int modelCheckerType, const po::variables_map &variablesMap)`  
*Check if all model checking type specific arguments are present.*
- `bool areStatisticalModelCheckingArgumentsPresent (const po::variables_map &variablesMap)`  
*Check if the arguments specific to statistical model checking are present.*
- `bool areApproximateProbabilisticModelCheckingArgumentsPresent (const po::variables_map &variablesMap)`  
*Check if the arguments specific to approximate probabilistic model checking are present.*
- `bool areBayesianModelCheckingArgumentsPresent (const po::variables_map &variablesMap)`  
*Check if the arguments specific to Bayesian model checking are present.*
- `bool areApproximateBayesianModelCheckingArgumentsPresent (const po::variables_map &variablesMap)`  
*Check if the arguments specific to approximate Bayesian model checking are present.*
- `void removeModelCheckingTypeSpecificArguments (unsigned int modelCheckerType, po::variables_map &variablesMap)`  
*Remove the model checking type specific arguments from the given variables\_map.*
- `void removeStatisticalModelCheckingArguments (po::variables_map &variablesMap)`  
*Remove the statistical model checking arguments from the given variables\_map.*
- `void removeApproximateProbabilisticModelCheckingArguments (po::variables_map &variablesMap)`  
*Remove the approximate probabilistic model checking arguments from the given variables\_map.*
- `void removeBayesianModelCheckingArguments (po::variables_map &variablesMap)`  
*Remove the Bayesian model checking arguments from the given variables\_map.*
- `void removeApproximateBayesianModelCheckingArguments (po::variables_map &variablesMap)`  
*Remove the approximate Bayesian model checking arguments from the given variables\_map.*
- `void initialiseClassMembers ()`  
*Initialise the class members using the command line arguments.*
- `void initialiseRequiredArgumentsDependentClassMembers ()`  
*Initialise the class members dependent on required command line arguments.*
- `void initialiseOptionalArgumentsDependentClassMembers ()`  
*Initialise the class members dependent on optional command line arguments.*
- `void initialiseModelCheckerTypeDependentClassMembers ()`  
*Initialise the class members dependent on the model checker type.*
- `void initialiseModelChecker ()`  
*Initialise the model checker.*
- `void initialiseProbabilisticBlackBoxModelChecker ()`  
*Initialise the probabilistic black box model checker.*
- `void initialiseStatisticalModelChecker ()`  
*Initialise the statistical model checker.*
- `void initialiseApproximateProbabilisticModelChecker ()`  
*Initialise the approximate probabilistic model checker.*
- `void initialiseBayesianModelChecker ()`  
*Initialise the Bayesian model checker.*
- `void initialiseApproximateBayesianModelChecker ()`  
*Initialise the approximate Bayesian model checker.*
- `void initialiseModelCheckingManager ()`  
*Initialise the model checking manager.*
- `void printModelCheckingInitialisationMessage ()`  
*Print the model checking initialisation message.*

## Private Attributes

- std::string `logicQueriesFilepath`
- std::string `tracesFolderPath`
- unsigned int `modelCheckerType`
- unsigned long `extraEvaluationTime`
- std::string `extraEvaluationProgramPath`
- std::string `typeSemanticsTableFilepath`
- bool `shouldVerboseDetailedResults`
- po::variables\_map `variablesMap`
- po::options\_description `allowedArguments`
- po::options\_description `requiredArguments`
- po::options\_description `optionalArguments`
- po::options\_description `modelCheckerTypeSpecificArguments`
- std::string `modelCheckerTypeName`
- std::string `modelCheckerParameters`
- std::shared\_ptr<ModelCheckerFactory> `modelCheckerFactory`
- std::shared\_ptr<ModelCheckingManager> `modelCheckingManager`

## Static Private Attributes

- static const std::string `ERR_INVALID_COMMAND_LINE_ARGUMENTS` = "Invalid command line arguments were provided and the model checker execution was stopped."
- static const std::string `ERR_INVALID_MODEL_CHECKING_ARGUMENTS` = "The command line arguments provided for the chosen model checking type are invalid. Please run Mule with the --help flag to determine which arguments you should use."
- static const std::string `ERR_INVALID_MODEL_CHECKING_TYPE` = "The provided model checking type is invalid. Please run Mule with the --help flag to determine which values you can use."
- static const std::string `ARG_LOGIC_QUERIES_NAME_LONG` = "logic-queries"
- static const std::string `ARG_LOGIC_QUERIES_NAME_BOTH` = `ARG_LOGIC_QUERIES_NAME_LONG + ",q"`
- static const std::string `ARG_LOGIC_QUERIES_DESCRIPTION` = "the path to the spatio-temporal queries input file"
- static const std::string `ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG` = "spatial-temporal-traces"
- static const std::string `ARG_SPATIAL_TEMPORAL_TRACES_NAME_BOTH` = `ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG + "t"`
- static const std::string `ARG_SPATIAL_TEMPORAL_TRACES_DESCRIPTION` = "the path to the folder containing spatio-temporal traces"
- static const std::string `ARG_EXTRA_EVALUATION_TIME_NAME_LONG` = "extra-evaluation-time"
- static const std::string `ARG_EXTRA_EVALUATION_TIME_NAME_BOTH` = `ARG_EXTRA_EVALUATION_TIME_NAME_LONG + ",e"`
- static const std::string `ARG_EXTRA_EVALUATION_TIME_DESCRIPTION` = "the maximum number of minutes the application can wait before finishing evaluation"
- static const std::string `ARG_MODEL_CHECKER_TYPE_NAME_LONG` = "model-checker-type"
- static const std::string `ARG_MODEL_CHECKER_TYPE_NAME_BOTH` = `ARG_MODEL_CHECKER_TYPE_NAME_LONG + ",m"`
- static const std::string `ARG_MODEL_CHECKER_TYPE_DESCRIPTION` = "the type of the model checker (0 = Probabilistic black-box, 1 = Frequentist statistical, 2 = Frequentist approximate probabilistic (Chernoff-Hoeffding), 3 = Bayesian (statistical hypothesis testing), 4 = Approximate Bayesian (mean and variance estimation))"
- static const std::string `ARG_HELP_NAME_LONG` = "help"
- static const std::string `ARG_HELP_NAME_BOTH` = `ARG_HELP_NAME_LONG + ",h"`
- static const std::string `ARG_HELP_DESCRIPTION` = "display help message (describing the meaning and usage of each command line argument)"

- static const std::string `ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG` = "extra-evaluation-program"
- static const std::string `ARG_EXTRA_EVALUATION_PROGRAM_NAME_BOTH` = `ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG` + ",p"
- static const std::string `ARG_EXTRA_EVALUATION_PROGRAM_DESCRIPTION` = "the program which will be executed whenever extra evaluation (and input traces) is required"
- static const std::string `ARG_TYPE_SEMATICS_TABLE_NAME_LONG` = "type-semantics-table"
- static const std::string `ARG_TYPE_SEMATICS_TABLE_NAME_BOTH` = `ARG_TYPE_SEMATICS_TABLE_NAME_LONG` + ",s"
- static const std::string `ARG_TYPE_SEMATICS_TABLE_DESCRIPTION` = "the type semantics table mapping semantic criteria values (e.g. Organ.Heart) to abstract positive natural numbers"
- static const std::string `ARG_VERBOSE_NAME_LONG` = "verbose"
- static const std::string `ARG_VERBOSE_NAME_BOTH` = `ARG_VERBOSE_NAME_LONG` + ",v"
- static const std::string `ARG_VERBOSE_DESCRIPTION` = "if this flag is set detailed evaluation results will be displayed"
- static const std::string `ARG_TYPE_I_ERROR_NAME_LONG` = "type-I-error"
- static const std::string `ARG_TYPE_I_ERROR_DESCRIPTION` = "the probability of type I errors"
- static const std::string `ARG_TYPE_II_ERROR_NAME_LONG` = "type-II-error"
- static const std::string `ARG_TYPE_II_ERROR_DESCRIPTION` = "the probability of type II errors"
- static const std::string `ARG_DELTA_NAME_LONG` = "delta"
- static const std::string `ARG_DELTA_DESCRIPTION` = "the upper bound on the probability to deviate from the true probability"
- static const std::string `ARG_EPSILON_NAME_LONG` = "epsilon"
- static const std::string `ARG_EPSILON_DESCRIPTION` = "the considered deviation from the true probability"
- static const std::string `ARG_BAYESIAN_ALPHA_NAME_LONG` = "bayesian-alpha"
- static const std::string `ARG_BAYESIAN_ALPHA_DESCRIPTION` = "the alpha shape parameter of the Beta distribution prior"
- static const std::string `ARG_BAYESIAN_BETA_NAME_LONG` = "bayesian-beta"
- static const std::string `ARG_BAYESIAN_BETA_DESCRIPTION` = "the beta shape parameter of the Beta distribution prior"
- static const std::string `ARG_BAYES_FACTOR_THRESHOLD_NAME_LONG` = "bayes-factor-threshold"
- static const std::string `ARG_BAYES_FACTOR_THRESHOLD_DESCRIPTION` = "the Bayes factor threshold used to fix the confidence level of the answer"
- static const std::string `ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG` = "approximate-bayesian-alpha"
- static const std::string `ARG_APPROXIMATE_BAYESIAN_ALPHA_DESCRIPTION` = "the alpha shape parameter of the Beta distribution prior"
- static const std::string `ARG_APPROXIMATE_BAYESIAN_BETA_NAME_LONG` = "approximate-bayesian-beta"
- static const std::string `ARG_APPROXIMATE_BAYESIAN_BETA_DESCRIPTION` = "the beta shape parameter of the Beta distribution prior"
- static const std::string `ARG_VARIANCE_THRESHOLD_NAME_LONG` = "variance-threshold"
- static const std::string `ARG_VARIANCE_THRESHOLD_DESCRIPTION` = "the variance threshold used to fix the confidence level of the answer"
- static const std::string `HELP_NAME_LABEL` = "NAME:"
- static const std::string `HELP_NAME_MSG` = " Mule - Multidimensional multiscale model checker"
- static const std::string `HELP_USAGE_LABEL` = "USAGE:"
- static const std::string `HELP_USAGE_MSG` = " Mule <required-arguments> [<optional-arguments>] <model-checking-type-specific-arguments>"
- static const std::string `HELP_DESCRIPTION_LABEL` = "DESCRIPTION:"
- static const std::string `HELP_DESCRIPTION_MSG` = " Mule is a multidimensional (spatial-temporal) multiscale approximate probabilistic model checker. It can be used for two different types of applications. First of all Mule can be employed to validate logic properties against multidimensional multiscale models. Secondly it can be used in reverse mode as a method to query time series data generated by in vivo/vitro experiments. Properties of interest are formalised using a multiscale spatio-temporal logic and their validity is checked using Mule."

- static const std::string `HELP_AUTHOR_LABEL` = "AUTHOR:"
- static const std::string `HELP_AUTHOR_MSG` = " The author of this software is Ovidiu Parvu."
- static const std::string `HELP_COPYRIGHT_LABEL` = "COPYRIGHT:"
- static const std::string `HELP_COPYRIGHT_MSG` = " Copyright Ovidiu Parvu 2014."
- static const std::string `HELP_REPORTING_BUGS_LABEL` = "REPORTING BUGS:"
- static const std::string `HELP_REPORTING_BUGS_MSG` = " Please send requests for fixing bugs or recommendations to <ovidiu.parvu[AT]gmail.com>."
- static const std::string `MSG_MODEL_CHECKING_HELP_REQUESTED` = "A request for displaying help information was issued."
- static const unsigned int `MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX` = 0
- static const unsigned int `MODEL_CHECKER_TYPE_STATISTICAL` = 1
- static const unsigned int `MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC` = 2
- static const unsigned int `MODEL_CHECKER_TYPE_BAYESIAN` = 3
- static const unsigned int `MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN` = 4
- static const std::string `MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME` = "Probabilistic black-box"
- static const std::string `MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_PARAMETERS` = "None"
- static const std::string `MODEL_CHECKER_STATISTICAL_NAME` = "Frequentist statistical"
- static const std::string `MODEL_CHECKER_STATISTICAL_PARAMETERS_BEGIN` = "Probability of type I errors (false negatives) = "
- static const std::string `MODEL_CHECKER_STATISTICAL_PARAMETERS_MIDDLE` = " and of type II errors (false positives) = "
- static const std::string `MODEL_CHECKER_STATISTICAL_PARAMETERS_END` = ":"
- static const std::string `MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME` = "Frequentist approximate probabilistic (Chernoff-Hoeffding)"
- static const std::string `MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_BEGIN` = "Upper bound on probability to deviate more than epsilon = "
- static const std::string `MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_MIDDLE` = " from the true probability is delta = "
- static const std::string `MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_END` = ":"
- static const std::string `MODEL_CHECKER_BAYESIAN_NAME` = "Bayesian (statistical hypothesis testing)"
- static const std::string `MODEL_CHECKER_BAYESIAN_PARAMETERS_BEGIN` = "Beta distribution prior shape parameters alpha = "
- static const std::string `MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE1` = " and beta = "
- static const std::string `MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE2` = ". Bayes factor threshold = "
- static const std::string `MODEL_CHECKER_BAYESIAN_PARAMETERS_END` = ":"
- static const std::string `MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME` = "Approximate Bayesian (mean and variance estimate)"
- static const std::string `MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_BEGIN` = "Beta distribution prior shape parameters alpha = "
- static const std::string `MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_MIDDLE1` = " and beta = "
- static const std::string `MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_MIDDLE2` = ". Variance threshold = "
- static const std::string `MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_END` = ":"
- static const std::string `CONFIG_CAPTION_ALLOWED_ARGUMENTS` = ""
- static const std::string `CONFIG_CAPTION_REQUIRED_ARGUMENTS` = "REQUIRED ARGUMENTS"
- static const std::string `CONFIG_CAPTION_OPTIONAL_ARGUMENTS` = "OPTIONAL ARGUMENTS"
- static const std::string `CONFIG_CAPTION_MODEL_CHECKER_TYPE_SPECIFIC_ARGUMENTS` = "MODEL CHECKING TYPE SPECIFIC ARGUMENTS"
- static const std::string `CONFIG_CAPTION_PROBABILISTIC_BLACK_BOX_MODEL_CHECKER_ARGUMENTS` = `MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME`
- static const std::string `CONFIG_CAPTION_STATISTICAL_MODEL_CHECKER_ARGUMENTS` = `MODEL_CHECKER_STATISTICAL_NAME`

- static const std::string `CONFIG_CAPTION_APPROXIMATE_PROBABILISTIC_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME`
- static const std::string `CONFIG_CAPTION_BAYESIAN_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_BAYESIAN_NAME`
- static const std::string `CONFIG_CAPTION_APPROXIMATE_BAYESIAN_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME`

### 7.47.1 Detailed Description

Class for running model checkers from the command line.

Definition at line 22 of file CommandLineModelChecking.hpp.

### 7.47.2 Constructor & Destructor Documentation

#### 7.47.2.1 CommandLineModelChecking::CommandLineModelChecking( )

Definition at line 25 of file CommandLineModelChecking.cpp.

#### 7.47.2.2 CommandLineModelChecking::~CommandLineModelChecking( )

Definition at line 34 of file CommandLineModelChecking.cpp.

### 7.47.3 Member Function Documentation

#### 7.47.3.1 bool CommandLineModelChecking::areApproximateBayesianModelCheckingArgumentsPresent( const po::variables\_map & variablesMap ) [private]

Check if the arguments specific to approximate Bayesian model checking are present.

##### Parameters

|                           |                                                      |
|---------------------------|------------------------------------------------------|
| <code>variablesMap</code> | The map containing all parsed command line arguments |
|---------------------------|------------------------------------------------------|

Definition at line 314 of file CommandLineModelChecking.cpp.

References ARG\_APPROXIMATE\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_APPROXIMATE\_BAYESIAN\_BEATA\_NAME\_LONG, and ARG\_VARIANCE\_THRESHOLD\_NAME\_LONG.

Referenced by areModelCheckingTypeSpecificArgumentsPresent().

#### 7.47.3.2 bool CommandLineModelChecking::areApproximateProbabilisticModelCheckingArgumentsPresent( const po::variables\_map & variablesMap ) [private]

Check if the arguments specific to approximate probabilistic model checking are present.

##### Parameters

|                           |                                                      |
|---------------------------|------------------------------------------------------|
| <code>variablesMap</code> | The map containing all parsed command line arguments |
|---------------------------|------------------------------------------------------|

Definition at line 299 of file CommandLineModelChecking.cpp.

References ARG\_DELTA\_NAME\_LONG, and ARG\_EPSILON\_NAME\_LONG.

Referenced by areModelCheckingTypeSpecificArgumentsPresent().

```
7.47.3.3 bool CommandLineModelChecking::areBayesianModelCheckingArgumentsPresent (const po::variables_map &
variablesMap) [private]
```

Check if the arguments specific to Bayesian model checking are present.

**Parameters**

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 306 of file CommandLineModelChecking.cpp.

References ARG\_BAYES\_FACTOR\_THRESHOLD\_NAME\_LONG, ARG\_BAYESIAN\_ALPHA\_NAME\_LONG, and ARG\_BAYESIAN\_BETA\_NAME\_LONG.

Referenced by areModelCheckingTypeSpecificArgumentsPresent().

#### 7.47.3.4 bool CommandLineModelChecking::areInvalidExecutionArguments ( const po::parsed\_options & *parsedArguments* ) [private]

Check if any invalid execution arguments were provided.

**Parameters**

|                        |                                   |
|------------------------|-----------------------------------|
| <i>parsedArguments</i> | The parsed command line arguments |
|------------------------|-----------------------------------|

Definition at line 153 of file CommandLineModelChecking.cpp.

References areUnrecognizedArgumentsPresent(), and isHelpArgumentPresent().

Referenced by areValidArgumentsConsideringConfiguration().

#### 7.47.3.5 bool CommandLineModelChecking::areInvalidModelCheckingArguments ( ) [private]

Check if any invalid model checker type dependent arguments are present.

Definition at line 212 of file CommandLineModelChecking.cpp.

References areInvalidModelCheckingArgumentsPresent(), ERR\_INVALID\_MODEL\_CHECKING\_ARGUMENTS, and MS\_throw.

Referenced by areValidArgumentsConsideringConfiguration().

#### 7.47.3.6 bool CommandLineModelChecking::areInvalidModelCheckingArgumentsPresent ( ) [private]

Check if any model checker type dependent arguments are invalid.

Definition at line 220 of file CommandLineModelChecking.cpp.

References areInvalidModelCheckingTypeSpecificArguments(), ARG\_MODEL\_CHECKER\_TYPE\_NAME\_LONG, modelCheckerType, removeOptionalArguments(), removeRequiredArguments(), and variablesMap.

Referenced by areInvalidModelCheckingArguments().

#### 7.47.3.7 bool CommandLineModelChecking::areInvalidModelCheckingTypeSpecificArguments ( unsigned int *modelCheckerType*, po::variables\_map & *variablesMap* ) [private]

Check if the model checking type specific arguments from the given variables\_map are invalid.

**Parameters**

|                         |                               |
|-------------------------|-------------------------------|
| <i>modelCheckerType</i> | The type of the model checker |
|-------------------------|-------------------------------|

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 255 of file CommandLineModelChecking.cpp.

References `areModelCheckingTypeSpecificArgumentsPresent()`, and `removeModelCheckingTypeSpecificArguments()`.

Referenced by `areInvalidModelCheckingArgumentsPresent()`.

**7.47.3.8 bool CommandLineModelChecking::areModelCheckingTypeSpecificArgumentsPresent ( unsigned int modelCheckerType, const po::variables\_map & variablesMap ) [private]**

Check if all model checking type specific arguments are present.

#### Parameters

|                         |                                                      |
|-------------------------|------------------------------------------------------|
| <i>modelCheckerType</i> | The type of the model checker                        |
| <i>variablesMap</i>     | The map containing all parsed command line arguments |

Definition at line 266 of file CommandLineModelChecking.cpp.

References `areApproximateBayesianModelCheckingArgumentsPresent()`, `areApproximateProbabilisticModelCheckingArgumentsPresent()`, `areBayesianModelCheckingArgumentsPresent()`, `areStatisticalModelCheckingArgumentsPresent()`, `ERR_INVALID_MODEL_CHECKING_TYPE`, `MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN`, `MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC`, `MODEL_CHECKER_TYPE_BAYESIAN`, `MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX`, `MODEL_CHECKER_TYPE_STATISTICAL`, and `MS_throw`.

Referenced by `areInvalidModelCheckingTypeSpecificArguments()`.

**7.47.3.9 bool CommandLineModelChecking::areStatisticalModelCheckingArgumentsPresent ( const po::variables\_map & variablesMap ) [private]**

Check if the arguments specific to statistical model checking are present.

#### Parameters

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 292 of file CommandLineModelChecking.cpp.

References `ARG_TYPE_I_ERROR_NAME_LONG`, and `ARG_TYPE_II_ERROR_NAME_LONG`.

Referenced by `areModelCheckingTypeSpecificArgumentsPresent()`.

**7.47.3.10 bool CommandLineModelChecking::areUnrecognizedArgumentsPresent ( const po::parsed\_options & parsedArguments ) [private]**

Check if any unrecognized command line arguments are present.

#### Parameters

|                        |                                   |
|------------------------|-----------------------------------|
| <i>parsedArguments</i> | The parsed command line arguments |
|------------------------|-----------------------------------|

Definition at line 205 of file CommandLineModelChecking.cpp.

Referenced by `areInvalidExecutionArguments()`.

7.47.3.11 bool CommandLineModelChecking::isValidArguments ( int *argc*, char \*\* *argv* ) [private]

Check if the provided command line arguments are valid.

**Parameters**

|             |                                               |
|-------------|-----------------------------------------------|
| <i>argc</i> | The number of provided command line arguments |
| <i>argv</i> | The collection of command line arguments      |

Definition at line 51 of file CommandLineModelChecking.cpp.

References areValidArgumentsConsideringConfiguration(), and initialiseAllowedArgumentsConfiguration().

Referenced by initialise().

**7.47.3.12 bool CommandLineModelChecking::areValidArgumentsConsideringConfiguration ( int *argc*, char \*\* *argv* ) [private]**

Check if the provided command line arguments are valid.

**Parameters**

|             |                                               |
|-------------|-----------------------------------------------|
| <i>argc</i> | The number of provided command line arguments |
| <i>argv</i> | The collection of command line arguments      |

Definition at line 131 of file CommandLineModelChecking.cpp.

References areInvalidExecutionArguments(), areInvalidModelCheckingArguments(), parseAndStoreArgumentsValues(), and variablesMap.

Referenced by areValidArguments().

**7.47.3.13 void CommandLineModelChecking::execute ( )**

Execute the model checking task.

Definition at line 47 of file CommandLineModelChecking.cpp.

References modelCheckerFactory, and modelCheckingManager.

Referenced by runModelCheckingTask().

**7.47.3.14 void CommandLineModelChecking::handleHelpRequest ( ) [private]**

Handle the help request i.e. if the –help flag was provided.

Definition at line 164 of file CommandLineModelChecking.cpp.

References MS\_throw, MSG\_MODEL\_CHECKING\_HELP\_REQUESTED, and printHelpMessage().

Referenced by initialise().

**7.47.3.15 void CommandLineModelChecking::initialise ( int *argc*, char \*\* *argv* )**

Initialise the class with the given command line arguments.

**Parameters**

|             |                                               |
|-------------|-----------------------------------------------|
| <i>argc</i> | The number of provided command line arguments |
| <i>argv</i> | The collection of command line arguments      |

Definition at line 36 of file CommandLineModelChecking.cpp.

References areValidArguments(), ERR\_INVALID\_COMMAND\_LINE\_ARGUMENTS, handleHelpRequest(), initialiseClassMembers(), isHelpArgumentPresent(), MS\_throw, and printModelCheckingInitialisationMessage().

Referenced by runModelCheckingTask().

---

**7.47.3.16 void CommandLineModelChecking::initialiseAllowedArgumentsConfiguration( ) [private]**

Initialise the configuration of allowed command line arguments.

Definition at line 57 of file CommandLineModelChecking.cpp.

References allowedArguments, initialiseModelCheckerTypeSpecificArgumentsConfiguration(), initialiseOptionalArgumentsConfiguration(), initialiseRequiredArgumentsConfiguration(), modelCheckerTypeSpecificArguments, optionalArguments, and requiredArguments.

Referenced by areValidArguments().

**7.47.3.17 void CommandLineModelChecking::initialiseApproximateBayesianModelChecker( ) [private]**

Initialise the approximate Bayesian model checker.

Definition at line 488 of file CommandLineModelChecking.cpp.

References ARG\_APPROXIMATE\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_APPROXIMATE\_BAYESIAN\_BETA\_NAME\_LONG, ARG\_VARIANCE\_THRESHOLD\_NAME\_LONG, MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_NAME, MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_BEGIN, MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_END, MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_MIDDLE1, MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_MIDDLE2, modelCheckerFactory, modelCheckerParameters, modelCheckerTypeName, multiscale::StringManipulator::toString(), and variablesMap.

Referenced by initialiseModelChecker().

**7.47.3.18 po::options\_description CommandLineModelChecking::initialiseApproximateBayesianModelCheckerArgumentsConfiguration( ) [private]**

Initialise the configuration of the approximate Bayesian model checker command line arguments.

Definition at line 121 of file CommandLineModelChecking.cpp.

References ARG\_APPROXIMATE\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_APPROXIMATE\_BAYESIAN\_BETA\_NAME\_LONG, ARG\_BAYESIAN\_ALPHA\_DESCRIPTION, ARG\_BAYESIAN\_BETA\_DESCRIPTION, ARG\_VARIANCE\_THRESHOLD\_DESCRIPTION, ARG\_VARIANCE\_THRESHOLD\_NAME\_LONG, and CONFIG\_OPTION\_APPROXIMATE\_BAYESIAN\_MODEL\_CHECKER\_ARGUMENTS.

Referenced by initialiseModelCheckerTypeSpecificArgumentsConfiguration().

**7.47.3.19 void CommandLineModelChecking::initialiseApproximateProbabilisticModelChecker( ) [private]**

Initialise the approximate probabilistic model checker.

Definition at line 453 of file CommandLineModelChecking.cpp.

References ARG\_DELTA\_NAME\_LONG, ARG\_EPSILON\_NAME\_LONG, MODEL\_CHECKER\_APPROXIMATE\_PROBABILISTIC\_NAME, MODEL\_CHECKER\_APPROXIMATE\_PROBABILISTIC\_PARAMETERS\_BEGIN, MODEL\_CHECKER\_APPROXIMATE\_PROBABILISTIC\_PARAMETERS\_END, MODEL\_CHECKER\_APPROXIMATE\_PROBABILISTIC\_PARAMETERS\_MIDDLE, modelCheckerFactory, modelCheckerParameters, modelCheckerTypeName, multiscale::StringManipulator::toString(), and variablesMap.

Referenced by initialiseModelChecker().

**7.47.3.20 po::options\_description CommandLineModelChecking::initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration( ) [private]**

Initialise the configuration of the approximate probabilistic model checker command line arguments.

Definition at line 102 of file CommandLineModelChecking.cpp.

References ARG\_DELTA\_DESCRIPTION, ARG\_DELTA\_NAME\_LONG, ARG\_EPSILON\_DESCRIPTION, ARG\_EPSILON\_NAME\_LONG, and CONFIG\_CAPTION\_APPROXIMATE\_PROBABILISTIC\_MODEL\_CHECKER\_ARGUMENTS.

Referenced by initialiseModelCheckerTypeSpecificArgumentsConfiguration().

#### 7.47.3.21 void CommandLineModelChecking::initialiseBayesianModelChecker( ) [private]

Initialise the Bayesian model checker.

Definition at line 469 of file CommandLineModelChecking.cpp.

References ARG\_BAYES\_FACTOR\_THRESHOLD\_NAME\_LONG, ARG\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_BAYESIAN\_BETA\_NAME\_LONG, MODEL\_CHECKER\_BAYESIAN\_NAME, MODEL\_CHECKER\_BAYESIAN\_PARAMETERS\_BEGIN, MODEL\_CHECKER\_BAYESIAN\_PARAMETERS\_END, MODEL\_CHECKER\_BAYESIAN\_PARAMETERS\_MIDDLE1, MODEL\_CHECKER\_BAYESIAN\_PARAMETERS\_MIDDLE2, modelCheckerFactory, modelCheckerParameters, modelCheckerTypeName, multiscale::StringManipulator::toString(), and variablesMap.

Referenced by initialiseModelChecker().

#### 7.47.3.22 po::options\_description CommandLineModelChecking::initialiseBayesianModelCheckerArgumentsConfiguration( ) [private]

Initialise the configuration of the Bayesian model checker command line arguments.

Definition at line 111 of file CommandLineModelChecking.cpp.

References ARG\_BAYES\_FACTOR\_THRESHOLD\_DESCRIPTION, ARG\_BAYES\_FACTOR\_THRESHOLD\_NAME\_LONG, ARG\_BAYESIAN\_ALPHA\_DESCRIPTION, ARG\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_BAYESIAN\_BETA\_DESCRIPTION, ARG\_BAYESIAN\_BETA\_NAME\_LONG, and CONFIG\_CAPTION\_BAYESIAN\_MODEL\_CHECKER\_ARGUMENTS.

Referenced by initialiseModelCheckerTypeSpecificArgumentsConfiguration().

#### 7.47.3.23 void CommandLineModelChecking::initialiseClassMembers( ) [private]

Initialise the class members using the command line arguments.

Definition at line 371 of file CommandLineModelChecking.cpp.

References initialiseModelCheckerTypeDependentClassMembers(), initialiseOptionalArgumentsDependentClassMembers(), and initialiseRequiredArgumentsDependentClassMembers().

Referenced by initialise().

#### 7.47.3.24 void CommandLineModelChecking::initialiseModelChecker( ) [private]

Initialise the model checker.

Definition at line 403 of file CommandLineModelChecking.cpp.

References ERR\_INVALID\_MODEL\_CHECKING\_TYPE, initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseProbabilisticBlackBoxModelChecker(), initialiseStatisticalModelChecker(), MODEL\_CHECKER\_TYPE\_APPROXIMATE\_BAYESIAN, MODEL\_CHECKER\_TYPE\_APPROXIMATE\_PROBABILISTIC, MODEL\_CHECKER\_TYPE\_BAYESIAN, MODEL\_CHECKER\_TYPE\_PROBABILISTIC\_BLACK\_BOX, MODEL\_CHECKER\_TYPE\_STATISTICAL, modelCheckerType, and MS\_THROW.

Referenced by initialiseModelCheckerTypeDependentClassMembers().

**7.47.3.25 void CommandLineModelChecking::initialiseModelCheckerTypeDependentClassMembers( ) [private]**

Initialise the class members dependent on the model checker type.

Definition at line 398 of file CommandLineModelChecking.cpp.

References initialiseModelChecker(), and initialiseModelCheckingManager().

Referenced by initialiseClassMembers().

**7.47.3.26 void CommandLineModelChecking::initialiseModelCheckerTypeSpecificArgumentsConfiguration( ) [private]**

Initialise the configuration of model checker type specific command line arguments.

Definition at line 81 of file CommandLineModelChecking.cpp.

References initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration(), initialiseBayesianModelCheckerArgumentsConfiguration(), initialiseStatisticalModelCheckerArgumentsConfiguration(), and modelCheckerTypeSpecificArguments.

Referenced by initialiseAllowedArgumentsConfiguration().

**7.47.3.27 void CommandLineModelChecking::initialiseModelCheckingManager( ) [private]**

Initialise the model checking manager.

Definition at line 509 of file CommandLineModelChecking.cpp.

References extraEvaluationProgramPath, extraEvaluationTime, logicQueriesFilepath, modelCheckingManager, shouldVerboseDetailedResults, tracesFolderPath, and typeSemanticsTableFilepath.

Referenced by initialiseModelCheckerTypeDependentClassMembers().

**7.47.3.28 void CommandLineModelChecking::initialiseOptionalArgumentsConfiguration( ) [private]**

Initialise the configuration of optional command line arguments.

Definition at line 74 of file CommandLineModelChecking.cpp.

References ARG\_EXTRA\_EVALUATION\_PROGRAM\_DESCRIPTION, ARG\_EXTRA\_EVALUATION\_PROGRAM\_NAME\_BOTH, ARG\_HELP\_DESCRIPTION, ARG\_HELP\_NAME\_BOTH, ARG\_TYPE\_SEMATICS\_TABLE\_DESCRIPTION, ARG\_TYPE\_SEMATICS\_TABLE\_NAME\_BOTH, ARG\_VERBOSE\_DESCRIPTION, ARG\_VERBOSE\_NAME\_BOTH, and optionalArguments.

Referenced by initialiseAllowedArgumentsConfiguration().

**7.47.3.29 void CommandLineModelChecking::initialiseOptionalArgumentsDependentClassMembers( ) [private]**

Initialise the class members dependent on optional command line arguments.

Definition at line 384 of file CommandLineModelChecking.cpp.

References ARG\_EXTRA\_EVALUATION\_PROGRAM\_NAME\_LONG, ARG\_TYPE\_SEMATICS\_TABLE\_NAME\_LONG, ARG\_VERBOSE\_NAME\_LONG, extraEvaluationProgramPath, shouldVerboseDetailedResults, typeSemanticsTableFilepath, and variablesMap.

Referenced by initialiseClassMembers().

**7.47.3.30 void CommandLineModelChecking::initialiseProbabilisticBlackBoxModelChecker( ) [private]**

Initialise the probabilistic black box model checker.

Definition at line 430 of file CommandLineModelChecking.cpp.

References MODEL\_CHECKER\_PROBABILISTIC\_BLACK\_BOX\_NAME, MODEL\_CHECKER\_PROBABILISTIC\_BLACK\_BOX\_PARAMETERS, modelCheckerFactory, modelCheckerParameters, and modelCheckerTypeName.

Referenced by initialiseModelChecker().

#### 7.47.3.31 void CommandLineModelChecking::initialiseRequiredArgumentsConfiguration( ) [private]

Initialise the configuration of required command line arguments.

Definition at line 67 of file CommandLineModelChecking.cpp.

References ARG\_EXTRA\_EVALUATION\_TIME\_DESCRIPTION, ARG\_EXTRA\_EVALUATION\_TIME\_NAME\_BOTH, ARG\_LOGIC\_QUERIES\_DESCRIPTION, ARG\_LOGIC\_QUERIES\_NAME\_BOTH, ARG\_MODEL\_CHECKER\_TYPE\_DESCRIPTION, ARG\_MODEL\_CHECKER\_TYPE\_NAME\_BOTH, ARG\_SPATIAL\_TEMPORAL\_TRACES\_DESCRIPTION, ARG\_SPATIAL\_TEMPORAL\_TRACES\_NAME\_BOTH, and requiredArguments.

Referenced by initialiseAllowedArgumentsConfiguration().

#### 7.47.3.32 void CommandLineModelChecking::initialiseRequiredArgumentsDependentClassMembers( ) [private]

Initialise the class members dependent on required command line arguments.

Definition at line 377 of file CommandLineModelChecking.cpp.

References ARG\_EXTRA\_EVALUATION\_TIME\_NAME\_LONG, ARG\_LOGIC\_QUERIES\_NAME\_LONG, ARG\_MODEL\_CHECKER\_TYPE\_NAME\_LONG, ARG\_SPATIAL\_TEMPORAL\_TRACES\_NAME\_LONG, extraEvaluationTime, logicQueriesFilepath, modelCheckerType, tracesFolderPath, and variablesMap.

Referenced by initialiseClassMembers().

#### 7.47.3.33 void CommandLineModelChecking::initialiseStatisticalModelChecker( ) [private]

Initialise the statistical model checker.

Definition at line 437 of file CommandLineModelChecking.cpp.

References ARG\_TYPE\_I\_ERROR\_NAME\_LONG, ARG\_TYPE\_II\_ERROR\_NAME\_LONG, MODEL\_CHECKER\_STATISTICAL\_NAME, MODEL\_CHECKER\_STATISTICAL\_PARAMETERS\_BEGIN, MODEL\_CHECKER\_STATISTICAL\_PARAMETERS\_END, MODEL\_CHECKER\_STATISTICAL\_PARAMETERS\_MIDDLE, modelCheckerFactory, modelCheckerParameters, modelCheckerTypeName, multiscale::StringManipulator::toString(), and variablesMap.

Referenced by initialiseModelChecker().

#### 7.47.3.34 po::options\_description CommandLineModelChecking::initialiseStatisticalModelCheckerArgumentsConfiguration( ) [private]

Initialise the configuration of the statistical model checker command line arguments.

Definition at line 93 of file CommandLineModelChecking.cpp.

References ARG\_TYPE\_I\_ERROR\_DESCRIPTION, ARG\_TYPE\_I\_ERROR\_NAME\_LONG, ARG\_TYPE\_II\_ERROR\_DESCRIPTION, ARG\_TYPE\_II\_ERROR\_NAME\_LONG, and CONFIG\_CAPTION\_STATISTICAL\_MODEL\_CHECKER\_ARGUMENTS.

Referenced by initialiseModelCheckerTypeSpecificArgumentsConfiguration().

**7.47.3.35 bool CommandLineModelChecking::isHelpArgumentPresent( ) [private]**

Check if the help command line argument is present.

Definition at line 160 of file CommandLineModelChecking.cpp.

References ARG\_HELP\_NAME\_LONG, and variablesMap.

Referenced by areInvalidExecutionArguments(), and initialise().

**7.47.3.36 po::parsed\_options CommandLineModelChecking::parseAndStoreArgumentsValues( int argc, char \*\* argv ) [private]**

Parse and store the command line arguments' values in a variables map.

**Parameters**

|             |                                               |
|-------------|-----------------------------------------------|
| <i>argc</i> | The number of provided command line arguments |
| <i>argv</i> | The collection of command line arguments      |

Definition at line 144 of file CommandLineModelChecking.cpp.

References allowedArguments, and variablesMap.

Referenced by areValidArgumentsConsideringConfiguration().

**7.47.3.37 void CommandLineModelChecking::printHelpClosingMessage( ) [private]**

Print the help closing message to the console.

Definition at line 192 of file CommandLineModelChecking.cpp.

References HELP\_AUTHOR\_LABEL, HELP\_AUTHOR\_MSG, HELP\_COPYRIGHT\_LABEL, HELP\_COPYRIGHT\_MSG, HELP\_REPORTING\_BUGS\_LABEL, and HELP\_REPORTING\_BUGS\_MSG.

Referenced by printHelpMessage().

**7.47.3.38 void CommandLineModelChecking::printHelpContentsMessage( ) [private]**

Print the help contents message to the console.

Definition at line 188 of file CommandLineModelChecking.cpp.

References allowedArguments.

Referenced by printHelpMessage().

**7.47.3.39 void CommandLineModelChecking::printHelpIntroMessage( ) [private]**

Print the help intro message to the console.

Definition at line 176 of file CommandLineModelChecking.cpp.

References HELP\_DESCRIPTION\_LABEL, HELP\_DESCRIPTION\_MSG, HELP\_NAME\_LABEL, HELP\_NAME\_MSG, HELP\_USAGE\_LABEL, and HELP\_USAGE\_MSG.

Referenced by printHelpMessage().

**7.47.3.40 void CommandLineModelChecking::printHelpMessage( ) [private]**

Print help message to the console.

Definition at line 170 of file CommandLineModelChecking.cpp.

References printHelpClosingMessage(), printHelpContentsMessage(), and printHelpIntroMessage().

Referenced by handleHelpRequest().

#### 7.47.3.41 void CommandLineModelChecking::printModelCheckingInitialisationMessage( ) [private]

Print the model checking initialisation message.

Definition at line 519 of file CommandLineModelChecking.cpp.

References extraEvaluationTime, logicQueriesFilepath, modelCheckerParameters, modelCheckerTypeName, multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage(), multiscale::verification::ModelCheckingOutputWriter::printIntroductionMessage(), and tracesFolderPath.

Referenced by initialise().

#### 7.47.3.42 void CommandLineModelChecking::removeApproximateBayesianModelCheckingArguments( po::variables\_map & variablesMap ) [private]

Remove the approximate Bayesian model checking arguments from the given variables\_map.

Parameters

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 365 of file CommandLineModelChecking.cpp.

References ARG\_APPROXIMATE\_BAYESIAN\_ALPHA\_NAME\_LONG, ARG\_APPROXIMATE\_BAYESIAN\_BE→TA\_NAME\_LONG, and ARG\_VARIANCE\_THRESHOLD\_NAME\_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

#### 7.47.3.43 void CommandLineModelChecking::removeApproximateProbabilisticModelCheckingArguments( po::variables\_map & variablesMap ) [private]

Remove the approximate probabilistic model checking arguments from the given variables\_map.

Parameters

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 354 of file CommandLineModelChecking.cpp.

References ARG\_DELTA\_NAME\_LONG, and ARG\_EPSILON\_NAME\_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

#### 7.47.3.44 void CommandLineModelChecking::removeBayesianModelCheckingArguments( po::variables\_map & variablesMap ) [private]

Remove the Bayesian model checking arguments from the given variables\_map.

Parameters

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 359 of file CommandLineModelChecking.cpp.

References ARG\_BAYES\_FACTOR\_THRESHOLD\_NAME\_LONG, ARG\_BAYESIAN\_ALPHA\_NAME\_LONG, and ARG\_BAYESIAN\_BETA\_NAME\_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

7.47.3.45 void CommandLineModelChecking::removeModelCheckingTypeSpecificArguments ( unsigned int *modelCheckerType*, po::variables\_map & *variablesMap* ) [private]

Remove the model checking type specific arguments from the given variables\_map.

**Parameters**

|                             |                                                      |
|-----------------------------|------------------------------------------------------|
| <i>modelChecker</i><br>Type | The type of the model checker                        |
| <i>variablesMap</i>         | The map containing all parsed command line arguments |

Definition at line 322 of file CommandLineModelChecking.cpp.

References `ERR_INVALID_MODEL_CHECKING_TYPE`, `MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN`, `MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC`, `MODEL_CHECKER_TYPE_BAYESIAN`, `MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX`, `MODEL_CHECKER_TYPE_STATISTICAL`, `MS_throw`, `removeApproximateBayesianModelCheckingArguments()`, `removeApproximateProbabilisticModelCheckingArguments()`, `removeBayesianModelCheckingArguments()`, and `removeStatisticalModelCheckingArguments()`.

Referenced by `areInvalidModelCheckingTypeSpecificArguments()`.

**7.47.3.46 void CommandLineModelChecking::removeOptionalArguments ( po::variables\_map & variablesMap ) [private]**

Remove the optional arguments from the given variables\_map.

**Parameters**

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 237 of file CommandLineModelChecking.cpp.

References `ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG`, `ARG_HELP_NAME_LONG`, `ARG_TYPE_SEMATICS_TABLE_NAME_LONG`, and `ARG_VERBOSE_NAME_LONG`.

Referenced by `areInvalidModelCheckingArgumentsPresent()`.

**7.47.3.47 void CommandLineModelChecking::removeRequiredArguments ( po::variables\_map & variablesMap ) [private]**

Remove the required arguments from the given variables\_map.

**Parameters**

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 230 of file CommandLineModelChecking.cpp.

References `ARG_EXTRA_EVALUATION_TIME_NAME_LONG`, `ARG_LOGIC_QUERIES_NAME_LONG`, `ARG_MODEL_CHECKER_TYPE_NAME_LONG`, and `ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG`.

Referenced by `areInvalidModelCheckingArgumentsPresent()`.

**7.47.3.48 void CommandLineModelChecking::removeStatisticalModelCheckingArguments ( po::variables\_map & variablesMap ) [private]**

Remove the statistical model checking arguments from the given variables\_map.

**Parameters**

|                     |                                                      |
|---------------------|------------------------------------------------------|
| <i>variablesMap</i> | The map containing all parsed command line arguments |
|---------------------|------------------------------------------------------|

Definition at line 349 of file CommandLineModelChecking.cpp.

References `ARG_TYPE_I_ERROR_NAME_LONG`, and `ARG_TYPE_II_ERROR_NAME_LONG`.

Referenced by `removeModelCheckingTypeSpecificArguments()`.

#### 7.47.4 Member Data Documentation

7.47.4.1 `po::options_description multiscale::verification::CommandLineModelChecking::allowedArguments [private]`

The configuration indicating which command line arguments are allowed

Definition at line 52 of file CommandLineModelChecking.hpp.

Referenced by `initialiseAllowedArgumentsConfiguration()`, `parseAndStoreArgumentsValues()`, and `printHelpContentsMessage()`.

7.47.4.2 `const std::string CommandLineModelChecking::ARG_APPROXIMATE_BAYESIAN_ALPHA_DESCRIPTION = "the alpha shape parameter of the Beta distribution prior" [static], [private]`

Definition at line 353 of file CommandLineModelChecking.hpp.

7.47.4.3 `const std::string CommandLineModelChecking::ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG = "approximate-bayesian-alpha" [static], [private]`

Definition at line 352 of file CommandLineModelChecking.hpp.

Referenced by `areApproximateBayesianModelCheckingArgumentsPresent()`, `initialiseApproximateBayesianModelChecker()`, `initialiseApproximateBayesianModelCheckerArgumentsConfiguration()`, and `removeApproximateBayesianModelCheckingArguments()`.

7.47.4.4 `const std::string CommandLineModelChecking::ARG_APPROXIMATE_BAYESIAN_BETA_DESCRIPTION = "the beta shape parameter of the Beta distribution prior" [static], [private]`

Definition at line 356 of file CommandLineModelChecking.hpp.

7.47.4.5 `const std::string CommandLineModelChecking::ARG_APPROXIMATE_BAYESIAN_BETA_NAME_LONG = "approximate-bayesian-beta" [static], [private]`

Definition at line 355 of file CommandLineModelChecking.hpp.

Referenced by `areApproximateBayesianModelCheckingArgumentsPresent()`, `initialiseApproximateBayesianModelChecker()`, `initialiseApproximateBayesianModelCheckerArgumentsConfiguration()`, and `removeApproximateBayesianModelCheckingArguments()`.

7.47.4.6 `const std::string CommandLineModelChecking::ARG_BAYES_FACTOR_THRESHOLD_DESCRIPTION = "the Bayes factor threshold used to fix the confidence level of the answer" [static], [private]`

Definition at line 350 of file CommandLineModelChecking.hpp.

Referenced by `initialiseBayesianModelCheckerArgumentsConfiguration()`.

7.47.4.7 `const std::string CommandLineModelChecking::ARG_BAYES_FACTOR_THRESHOLD_NAME_LONG = "bayes-factor-threshold" [static], [private]`

Definition at line 349 of file CommandLineModelChecking.hpp.

Referenced by `areBayesianModelCheckingArgumentsPresent()`, `initialiseBayesianModelChecker()`, `initialiseBayesianModelCheckerArgumentsConfiguration()`, and `removeBayesianModelCheckingArguments()`.

7.47.4.8 `const std::string CommandLineModelChecking::ARG_BAYESIAN_ALPHA_DESCRIPTION = "the alpha shape parameter of the Beta distribution prior" [static], [private]`

Definition at line 344 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and initialiseBayesianModelCheckerArgumentsConfiguration().

7.47.4.9 `const std::string CommandLineModelChecking::ARG_BAYESIAN_ALPHA_NAME_LONG = "bayesian-alpha" [static], [private]`

Definition at line 343 of file CommandLineModelChecking.hpp.

Referenced by areBayesianModelCheckingArgumentsPresent(), initialiseBayesianModelChecker(), initialiseBayesianModelCheckerArgumentsConfiguration(), and removeBayesianModelCheckingArguments().

7.47.4.10 `const std::string CommandLineModelChecking::ARG_BAYESIAN_BETA_DESCRIPTION = "the beta shape parameter of the Beta distribution prior" [static], [private]`

Definition at line 347 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and initialiseBayesianModelCheckerArgumentsConfiguration().

7.47.4.11 `const std::string CommandLineModelChecking::ARG_BAYESIAN_BETA_NAME_LONG = "bayesian-beta" [static], [private]`

Definition at line 346 of file CommandLineModelChecking.hpp.

Referenced by areBayesianModelCheckingArgumentsPresent(), initialiseBayesianModelChecker(), initialiseBayesianModelCheckerArgumentsConfiguration(), and removeBayesianModelCheckingArguments().

7.47.4.12 `const std::string CommandLineModelChecking::ARG_DELTA_DESCRIPTION = "the upper bound on the probability to deviate from the true probability" [static], [private]`

Definition at line 338 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

7.47.4.13 `const std::string CommandLineModelChecking::ARG_DELTA_NAME_LONG = "delta" [static], [private]`

Definition at line 337 of file CommandLineModelChecking.hpp.

Referenced by areApproximateProbabilisticModelCheckingArgumentsPresent(), initialiseApproximateProbabilisticModelChecker(), initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration(), and removeApproximateProbabilisticModelCheckingArguments().

7.47.4.14 `const std::string CommandLineModelChecking::ARG_EPSILON_DESCRIPTION = "the considered deviation from the true probability" [static], [private]`

Definition at line 341 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

```
7.47.4.15 const std::string CommandLineModelChecking::ARG_EPSILON_NAME_LONG = "epsilon" [static],
[private]
```

Definition at line 340 of file CommandLineModelChecking.hpp.

Referenced by areApproximateProbabilisticModelCheckingArgumentsPresent(), initialiseApproximateProbabilisticModelChecker(), initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration(), and removeApproximateProbabilisticModelCheckingArguments().

```
7.47.4.16 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_PROGRAM_DESCRIPTION = "the
program which will be executed whenever extra evaluation (and input traces) is required" [static],
[private]
```

Definition at line 321 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.47.4.17 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_PROGRAM_NAME_BOTH =
ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG + ",p" [static], [private]
```

Definition at line 320 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.47.4.18 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG =
"extra-evaluation-program" [static], [private]
```

Definition at line 319 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsDependentClassMembers(), and removeOptionalArguments().

```
7.47.4.19 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_TIME_DESCRIPTION = "the maximum
number of minutes the application can wait before finishing evaluation" [static], [private]
```

Definition at line 309 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.47.4.20 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_TIME_NAME_BOTH =
ARG_EXTRA_EVALUATION_TIME_NAME_LONG + ",e" [static], [private]
```

Definition at line 308 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.47.4.21 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_TIME_NAME_LONG =
"extra-evaluation-time" [static], [private]
```

Definition at line 307 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

```
7.47.4.22 const std::string CommandLineModelChecking::ARG_HELP_DESCRIPTION = "display help message (describing the
meaning and usage of each command line argument)" [static], [private]
```

Definition at line 317 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.23 `const std::string CommandLineModelChecking::ARG_HELP_NAME_BOTH = ARG_HELP_NAME_LONG + "h"` [static], [private]

Definition at line 316 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.24 `const std::string CommandLineModelChecking::ARG_HELP_NAME_LONG = "help"` [static], [private]

Definition at line 315 of file CommandLineModelChecking.hpp.

Referenced by isHelpArgumentPresent(), and removeOptionalArguments().

7.47.4.25 `const std::string CommandLineModelChecking::ARG_LOGIC_QUERIES_DESCRIPTION = "the path to the spatio-temporal queries input file"` [static], [private]

Definition at line 301 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.26 `const std::string CommandLineModelChecking::ARG_LOGIC_QUERIES_NAME_BOTH = ARG_LOGIC_QUERIES_NAME_LONG + "q"` [static], [private]

Definition at line 300 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.27 `const std::string CommandLineModelChecking::ARG_LOGIC_QUERIES_NAME_LONG = "logic-queries"` [static], [private]

Definition at line 299 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.47.4.28 `const std::string CommandLineModelChecking::ARG_MODEL_CHECKER_TYPE_DESCRIPTION = "the type of the model checker (0 = Probabilistic black-box, 1 = Frequentist statistical, 2 = Frequentist approximate probabilistic (Chernoff-Hoeffding), 3 = Bayesian (statistical hypothesis testing), 4 = Approximate Bayesian (mean and variance estimation))"` [static], [private]

Definition at line 313 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.29 `const std::string CommandLineModelChecking::ARG_MODEL_CHECKER_TYPE_NAME_BOTH = ARG_MODEL_CHECKER_TYPE_NAME_LONG + ",m"` [static], [private]

Definition at line 312 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.30 `const std::string CommandLineModelChecking::ARG_MODEL_CHECKER_TYPE_NAME_LONG = "model-checker-type"` [static], [private]

Definition at line 311 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArgumentsPresent(), initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.47.4.31 `const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_DESCRIPTION = "the path to the folder containing spatio-temporal traces" [static], [private]`

Definition at line 305 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.32 `const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_NAME_BOTH = ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG + ",t" [static], [private]`

Definition at line 304 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.47.4.33 `const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG = "spatial-temporal-traces" [static], [private]`

Definition at line 303 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.47.4.34 `const std::string CommandLineModelChecking::ARG_TYPE_I_ERROR_DESCRIPTION = "the probability of type I errors" [static], [private]`

Definition at line 332 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

7.47.4.35 `const std::string CommandLineModelChecking::ARG_TYPE_I_ERROR_NAME_LONG = "type-I-error" [static], [private]`

Definition at line 331 of file CommandLineModelChecking.hpp.

Referenced by areStatisticalModelCheckingArgumentsPresent(), initialiseStatisticalModelChecker(), initialiseStatisticalModelCheckerArgumentsConfiguration(), and removeStatisticalModelCheckingArguments().

7.47.4.36 `const std::string CommandLineModelChecking::ARG_TYPE_II_ERROR_DESCRIPTION = "the probability of type II errors" [static], [private]`

Definition at line 335 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

7.47.4.37 `const std::string CommandLineModelChecking::ARG_TYPE_II_ERROR_NAME_LONG = "type-II-error" [static], [private]`

Definition at line 334 of file CommandLineModelChecking.hpp.

Referenced by areStatisticalModelCheckingArgumentsPresent(), initialiseStatisticalModelChecker(), initialiseStatisticalModelCheckerArgumentsConfiguration(), and removeStatisticalModelCheckingArguments().

---

---

7.47.4.38 `const std::string CommandLineModelChecking::ARG_TYPE_SEMATICS_TABLE_DESCRIPTION = "the type semantics table mapping semantic criteria values (e.g. Organ.Heart) to abstract positive natural numbers"` [static], [private]

Definition at line 325 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.39 `const std::string CommandLineModelChecking::ARG_TYPE_SEMATICS_TABLE_NAME_BOTH = ARG_TYPE_SEMATICS_TABLE_NAME_LONG + "s"` [static], [private]

Definition at line 324 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.40 `const std::string CommandLineModelChecking::ARG_TYPE_SEMATICS_TABLE_NAME_LONG = "type-semantics-table"` [static], [private]

Definition at line 323 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsDependentClassMembers(), and removeOptionalArguments().

7.47.4.41 `const std::string CommandLineModelChecking::ARG_VARIANCE_THRESHOLD_DESCRIPTION = "the variance threshold used to fix the confidence level of the answer"` [static], [private]

Definition at line 359 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration().

7.47.4.42 `const std::string CommandLineModelChecking::ARG_VARIANCE_THRESHOLD_NAME_LONG = "variance-threshold"` [static], [private]

Definition at line 358 of file CommandLineModelChecking.hpp.

Referenced by areApproximateBayesianModelCheckingArgumentsPresent(), initialiseApproximateBayesianModelChecker(), initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and removeApproximateBayesianModelCheckingArguments().

7.47.4.43 `const std::string CommandLineModelChecking::ARG_VERBOSE_DESCRIPTION = "if this flag is set detailed evaluation results will be displayed"` [static], [private]

Definition at line 329 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.44 `const std::string CommandLineModelChecking::ARG_VERBOSE_NAME_BOTH = ARG_VERBOSE_NAME_LONG + "v"` [static], [private]

Definition at line 328 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.47.4.45 `const std::string CommandLineModelChecking::ARG_VERBOSE_NAME_LONG = "verbose"` [static], [private]

Definition at line 327 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsDependentClassMembers(), and removeOptionalArguments().

7.47.4.46 `const std::string CommandLineModelChecking::CONFIG_CAPTION_ALLOWED_ARGUMENTS = "" [static], [private]`

Definition at line 407 of file CommandLineModelChecking.hpp.

7.47.4.47 `const std::string CommandLineModelChecking::CONFIG_CAPTION_APPROXIMATE_BAYESIAN_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME [static], [private]`

Definition at line 416 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration().

7.47.4.48 `const std::string CommandLineModelChecking::CONFIG_CAPTION_APPROXIMATE_PROBABILISTIC_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME [static], [private]`

Definition at line 414 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

7.47.4.49 `const std::string CommandLineModelChecking::CONFIG_CAPTION_BAYESIAN_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_BAYESIAN_NAME [static], [private]`

Definition at line 415 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelCheckerArgumentsConfiguration().

7.47.4.50 `const std::string CommandLineModelChecking::CONFIG_CAPTION_MODEL_CHECKER_TYPE_SPECIFIC_ARGUMENTS = "MODEL CHECKING TYPE SPECIFIC ARGUMENTS" [static], [private]`

Definition at line 410 of file CommandLineModelChecking.hpp.

7.47.4.51 `const std::string CommandLineModelChecking::CONFIG_CAPTION_OPTIONAL_ARGUMENTS = "OPTIONAL ARGUMENTS" [static], [private]`

Definition at line 409 of file CommandLineModelChecking.hpp.

7.47.4.52 `const std::string CommandLineModelChecking::CONFIG_CAPTION_PROBABILISTIC_BLACK_BOX_MODEL_CHECKER_ARGUMENTS = MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME [static], [private]`

Definition at line 412 of file CommandLineModelChecking.hpp.

7.47.4.53 `const std::string CommandLineModelChecking::CONFIG_CAPTION_REQUIRED_ARGUMENTS = "REQUIRED ARGUMENTS" [static], [private]`

Definition at line 408 of file CommandLineModelChecking.hpp.

---

7.47.4.54 const std::string CommandLineModelChecking::CONFIG\_CAPTION\_STATISTICAL\_MODEL\_CHECKER\_ARGUMENTS = **MODEL\_CHECKER\_STATISTICAL\_NAME** [static], [private]

Definition at line 413 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

7.47.4.55 const std::string CommandLineModelChecking::ERR\_INVALID\_COMMAND\_LINE\_ARGUMENTS = "Invalid command line arguments were provided and the model checker execution was stopped." [static], [private]

Definition at line 294 of file CommandLineModelChecking.hpp.

Referenced by initialise().

7.47.4.56 const std::string CommandLineModelChecking::ERR\_INVALID\_MODEL\_CHECKING\_ARGUMENTS = "The command line arguments provided for the chosen model checking type are invalid. Please run Mule with the --help flag to determine which arguments you should use." [static], [private]

Definition at line 295 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArguments().

7.47.4.57 const std::string CommandLineModelChecking::ERR\_INVALID\_MODEL\_CHECKING\_TYPE = "The provided model checking type is invalid. Please run Mule with the --help flag to determine which values you can use." [static], [private]

Definition at line 297 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

7.47.4.58 std::string multiscale::verification::CommandLineModelChecking::extraEvaluationProgramPath [private]

The path to the program which will be executed whenever more traces are required

Definition at line 38 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), and initialiseOptionalArgumentsDependentClassMembers().

7.47.4.59 unsigned long multiscale::verification::CommandLineModelChecking::extraEvaluationTime [private]

The number of minutes for which the application waits for new traces to be produced

Definition at line 35 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArgumentsDependentClassMembers(), and printModelCheckingInitialisationMessage().

7.47.4.60 const std::string CommandLineModelChecking::HELP\_AUTHOR\_LABEL = "AUTHOR:" [static], [private]

Definition at line 367 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.47.4.61 const std::string CommandLineModelChecking::HELP_AUTHOR_MSG = " The author of this software is Ovidiu
Parvu." [static], [private]
```

Definition at line 368 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.47.4.62 const std::string CommandLineModelChecking::HELP_COPYRIGHT_LABEL = "COPYRIGHT:" [static],
[private]
```

Definition at line 369 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.47.4.63 const std::string CommandLineModelChecking::HELP_COPYRIGHT_MSG = " Copyright Ovidiu Parvu 2014."
[static], [private]
```

Definition at line 370 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.47.4.64 const std::string CommandLineModelChecking::HELP_DESCRIPTION_LABEL = "DESCRIPTION:" [static],
[private]
```

Definition at line 365 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.47.4.65 const std::string CommandLineModelChecking::HELP_DESCRIPTION_MSG = " Mule is a multidimensional
(spatial-temporal) multiscale approximate probabilistic model checker. It can be used for two different types of
applications. First of all Mule can be employed to validate logic properties against multidimensional multiscale
models. Secondly it can be used in reverse mode as a method to query time series data generated by in vivo/vitro
experiments. Properties of interest are formalised using a multiscale spatio-temporal logic and their validity is
checked using Mule." [static], [private]
```

Definition at line 366 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.47.4.66 const std::string CommandLineModelChecking::HELP_NAME_LABEL = "NAME:" [static], [private]
```

Definition at line 361 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.47.4.67 const std::string CommandLineModelChecking::HELP_NAME_MSG = " Mule - Multidimensional multiscale model
checker" [static], [private]
```

Definition at line 362 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.47.4.68 const std::string CommandLineModelChecking::HELP_REPORTING_BUGS_LABEL = "REPORTING BUGS:"
[static], [private]
```

Definition at line 371 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

7.47.4.69 const std::string CommandLineModelChecking::HELP\_REPORTING\_BUGS\_MSG = " Please send requests for fixing bugs or recommendations to <ovidiu.parvu[AT]gmail.com>." [static], [private]

Definition at line 372 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

7.47.4.70 const std::string CommandLineModelChecking::HELP\_USAGE\_LABEL = "USAGE:" [static], [private]

Definition at line 363 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

7.47.4.71 const std::string CommandLineModelChecking::HELP\_USAGE\_MSG = " Mule <required-arguments> [<optional-arguments>] <model-checking-type-specific-arguments>" [static], [private]

Definition at line 364 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

7.47.4.72 std::string multiscale::verification::CommandLineModelChecking::logicQueriesFilepath [private]

The path to the logic queries file

Definition at line 27 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArgumentsDependentClassMembers(), and printModelCheckingInitialisationMessage().

7.47.4.73 const std::string CommandLineModelChecking::MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_NAME = "Approximate Bayesian (mean and variance estimate)" [static], [private]

Definition at line 401 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

7.47.4.74 const std::string CommandLineModelChecking::MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_N\_PARAMETERS\_BEGIN = "Beta distribution prior shape parameters alpha = " [static], [private]

Definition at line 402 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

7.47.4.75 const std::string CommandLineModelChecking::MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_END = ":" [static], [private]

Definition at line 405 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

7.47.4.76 const std::string CommandLineModelChecking::MODEL\_CHECKER\_APPROXIMATE\_BAYESIAN\_PARAMETERS\_MI\_DDLE1 = " and beta = " [static], [private]

Definition at line 403 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

7.47.4.77 `const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_BAYESIAN_PARAMETERS_MI_DDLE2 = ". Variance threshold = " [static], [private]`

Definition at line 404 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

7.47.4.78 `const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME = "Frequentist approximate probabilistic (Chernoff-Hoeffding)" [static], [private]`

Definition at line 390 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

7.47.4.79 `const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_BEGIN = "Upper bound on probability to deviate more than epsilon = " [static], [private]`

Definition at line 391 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

7.47.4.80 `const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_END = "." [static], [private]`

Definition at line 393 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

7.47.4.81 `const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_PARAMETERS_MIDDLE = " from the true probability is delta = " [static], [private]`

Definition at line 392 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

7.47.4.82 `const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_NAME = "Bayesian (statistical hypothesis testing)" [static], [private]`

Definition at line 395 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

7.47.4.83 `const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_PARAMETERS_BEGIN = "Beta distribution prior shape parameters alpha = " [static], [private]`

Definition at line 396 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

---

```
7.47.4.84 const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_PARAMETERS_END = "."
[static], [private]
```

Definition at line 399 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.47.4.85 const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE1 = " and
beta = " [static], [private]
```

Definition at line 397 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.47.4.86 const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE2 = ". Bayes
factor threshold = " [static], [private]
```

Definition at line 398 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.47.4.87 const std::string CommandLineModelChecking::MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME =
"Probabilistic black-box" [static], [private]
```

Definition at line 382 of file CommandLineModelChecking.hpp.

Referenced by initialiseProbabilisticBlackBoxModelChecker().

```
7.47.4.88 const std::string CommandLineModelChecking::MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_PARAMETERS =
"None" [static], [private]
```

Definition at line 383 of file CommandLineModelChecking.hpp.

Referenced by initialiseProbabilisticBlackBoxModelChecker().

```
7.47.4.89 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_NAME = "Frequentist statistical"
[static], [private]
```

Definition at line 385 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.47.4.90 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_BEGIN =
"Probability of type I errors (false negatives) = " [static], [private]
```

Definition at line 386 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.47.4.91 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_END = "."
[static], [private]
```

Definition at line 388 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.47.4.92 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_MIDDLE = " and of
type II errors (false positives) = " [static], [private]
```

Definition at line 387 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.47.4.93 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN = 4 [static], [private]
```

Definition at line 380 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.47.4.94 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC = 2 [static], [private]
```

Definition at line 378 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.47.4.95 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_BAYESIAN = 3 [static], [private]
```

Definition at line 379 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.47.4.96 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX = 0 [static], [private]
```

Definition at line 376 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.47.4.97 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_STATISTICAL = 1 [static], [private]
```

Definition at line 377 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.47.4.98 std::shared_ptr<ModelCheckerFactory> multiscale::verification::CommandLineModelChecking::modelCheckerFactory [private]
```

The model checker

Definition at line 70 of file CommandLineModelChecking.hpp.

Referenced by execute(), initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseProbabilisticBlackBoxModelChecker(), and initialiseStatisticalModelChecker().

7.47.4.99 `std::string multiscale::verification::CommandLineModelChecking::modelCheckerParameters` [private]

The parameters specific to the model checker

Definition at line 67 of file `CommandLineModelChecking.hpp`.

Referenced by `initialiseApproximateBayesianModelChecker()`, `initialiseApproximateProbabilisticModelChecker()`, `initialiseBayesianModelChecker()`, `initialiseProbabilisticBlackBoxModelChecker()`, `initialiseStatisticalModelChecker()`, and `printModelCheckingInitialisationMessage()`.

7.47.4.100 `unsigned int multiscale::verification::CommandLineModelChecking::modelCheckerType` [private]

The type of the model checker

Definition at line 32 of file `CommandLineModelChecking.hpp`.

Referenced by `areInvalidModelCheckingArgumentsPresent()`, `initialiseModelChecker()`, and `initialiseRequiredArgumentsDependentClassMembers()`.

7.47.4.101 `std::string multiscale::verification::CommandLineModelChecking::modelCheckerTypeName` [private]

The name of the model checker type

Definition at line 65 of file `CommandLineModelChecking.hpp`.

Referenced by `initialiseApproximateBayesianModelChecker()`, `initialiseApproximateProbabilisticModelChecker()`, `initialiseBayesianModelChecker()`, `initialiseProbabilisticBlackBoxModelChecker()`, `initialiseStatisticalModelChecker()`, and `printModelCheckingInitialisationMessage()`.

7.47.4.102 `po::options_description multiscale::verification::CommandLineModelChecking::modelCheckerTypeSpecificArguments` [private]

The configuration indicating which command line arguments are allowed

Definition at line 61 of file `CommandLineModelChecking.hpp`.

Referenced by `initialiseAllowedArgumentsConfiguration()`, and `initialiseModelCheckerTypeSpecificArgumentsConfiguration()`.

7.47.4.103 `std::shared_ptr<ModelCheckingManager> multiscale::verification::CommandLineModelChecking::modelCheckingManager` [private]

The model checking task manager

Definition at line 72 of file `CommandLineModelChecking.hpp`.

Referenced by `execute()`, and `initialiseModelCheckingManager()`.

7.47.4.104 `const std::string CommandLineModelChecking::MSG_MODEL_CHECKING_HELP_REQUESTED = "A request for displaying help information was issued."` [static], [private]

Definition at line 374 of file `CommandLineModelChecking.hpp`.

Referenced by `handleHelpRequest()`.

7.47.4.105 `po::options_description multiscale::verification::CommandLineModelChecking::optionalArguments` [private]

The configuration indicating which command line arguments are allowed

Definition at line 58 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), and initialiseOptionalArgumentsConfiguration().

#### 7.47.4.106 `po::options_description multiscale::verification::CommandLineModelChecking::requiredArguments [private]`

The configuration indicating which command line arguments are allowed

Definition at line 55 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), and initialiseRequiredArgumentsConfiguration().

#### 7.47.4.107 `bool multiscale::verification::CommandLineModelChecking::shouldVerboseDetailedResults [private]`

The flag indicating if detailed results should be printed out

Definition at line 45 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), and initialiseOptionalArgumentsDependentClassMembers().

#### 7.47.4.108 `std::string multiscale::verification::CommandLineModelChecking::tracesFolderPath [private]`

The path to the folder containing traces

Definition at line 29 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArgumentsDependentClassMembers(), and printModelCheckingInitialisationMessage().

#### 7.47.4.109 `std::string multiscale::verification::CommandLineModelChecking::typeSemanticsTableFilepath [private]`

The path to the type semantics table

Definition at line 42 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), and initialiseOptionalArgumentsDependentClassMembers().

#### 7.47.4.110 `po::variables_map multiscale::verification::CommandLineModelChecking::variablesMap [private]`

The map containing  $\langle a, v \rangle$  pairs where  $a$  = command line argument and  $v$  = value

Definition at line 49 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArgumentsPresent(), areValidArgumentsConsideringConfiguration(), initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseOptionalArgumentsDependentClassMembers(), initialiseRequiredArgumentsDependentClassMembers(), initialiseStatisticalModelChecker(), isHelpArgumentPresent(), and parseAndStoreArgumentsValues().

The documentation for this class was generated from the following files:

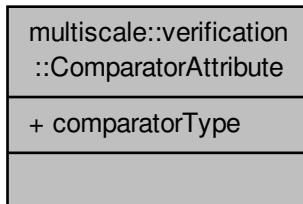
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/[CommandLineModelChecking.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/[CommandLineModelChecking.cpp](#)

## 7.48 multiscale::verification::ComparatorAttribute Class Reference

Class for representing a comparator attribute.

```
#include <ComparatorAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ComparatorAttribute:



### Public Attributes

- [ComparatorType comparatorType](#)

#### 7.48.1 Detailed Description

Class for representing a comparator attribute.

Definition at line 31 of file ComparatorAttribute.hpp.

#### 7.48.2 Member Data Documentation

##### 7.48.2.1 ComparatorType multiscale::verification::ComparatorAttribute::comparatorType

The comparator type

Definition at line 35 of file ComparatorAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeTemporalNumericMeasure(), multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericComparison(), multiscale::verification::ProbabilisticLogicPropertyAttribute::getComparator(), and multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

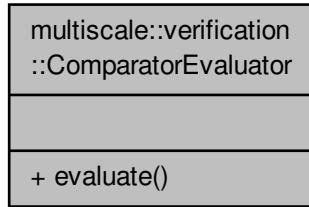
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[ComparatorAttribute.hpp](#)

## 7.49 multiscale::verification::ComparatorEvaluator Class Reference

Class for evaluating comparison expressions.

```
#include <ComparatorEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::ComparatorEvaluator:



## Static Public Member Functions

- template<typename T >  
static bool **evaluate** (T lhsElement, const **ComparatorType** &comparator, T rhsElement)  
*Compare two elements using a ComparatorType comparator.*

### 7.49.1 Detailed Description

Class for evaluating comparison expressions.

Definition at line 13 of file ComparatorEvaluator.hpp.

### 7.49.2 Member Function Documentation

- 7.49.2.1 template<typename T > static bool multiscale::verification::ComparatorEvaluator::evaluate ( T lhsElement, const ComparatorType & comparator, T rhsElement ) [inline], [static]**

Compare two elements using a ComparatorType comparator.

#### Parameters

|                   |                                                               |
|-------------------|---------------------------------------------------------------|
| <i>lhsElement</i> | The element which is on the left hand side of the comparator  |
| <i>comparator</i> | The comparator type used to compare the elements              |
| <i>rhsElement</i> | The element which is on the right hand side of the comparator |

Definition at line 24 of file ComparatorEvaluator.hpp.

References multiscale::Numeric::almostEqual(), multiscale::verification::Equal, multiscale::ERR\_UNDEFIN← D\_ENUM\_VALUE, multiscale::Numeric::greaterOrEqual(), multiscale::verification::GreaterThan, multiscale← :verification::GreaterThanOrEqual, multiscale::Numeric::lessOrEqual(), multiscale::verification::LessThan, multiscale::verification::LessThanOrEqual, and MS\_throw.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeTemporalNumericMeasure(), multiscale← :verification::LogicPropertyVisitor::evaluateTemporalNumericComparison(), multiscale::verification::Constraint← Visitor::filterSpatialEntitiesWrtSpatialMeasure(), and multiscale::verification::ConstraintVisitor::filterSpatialEntities← WrtTypeConsideringNonEqualComparator().

The documentation for this class was generated from the following file:

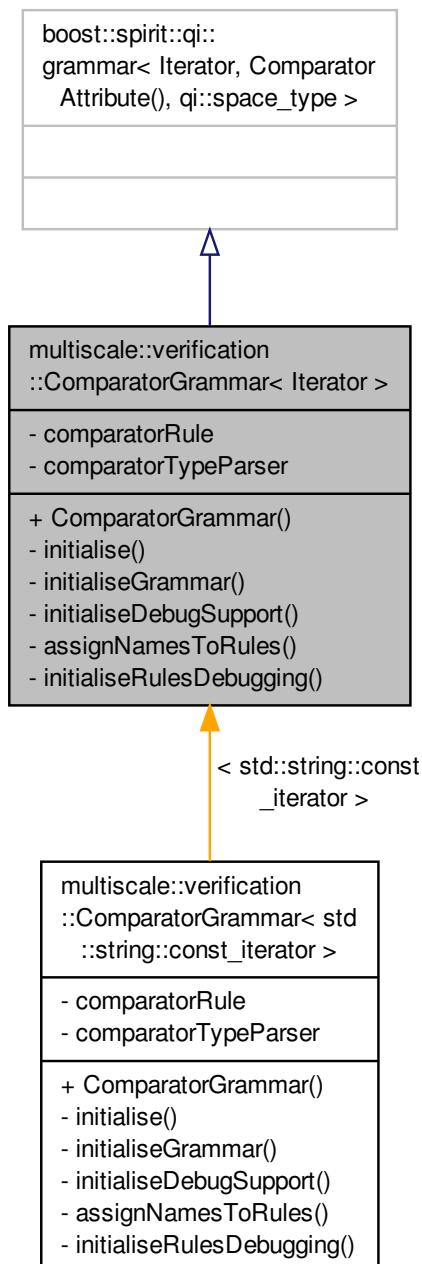
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/**ComparatorEvaluator.hpp**

## 7.50 multiscale::verification::ComparatorGrammar< Iterator > Class Template Reference

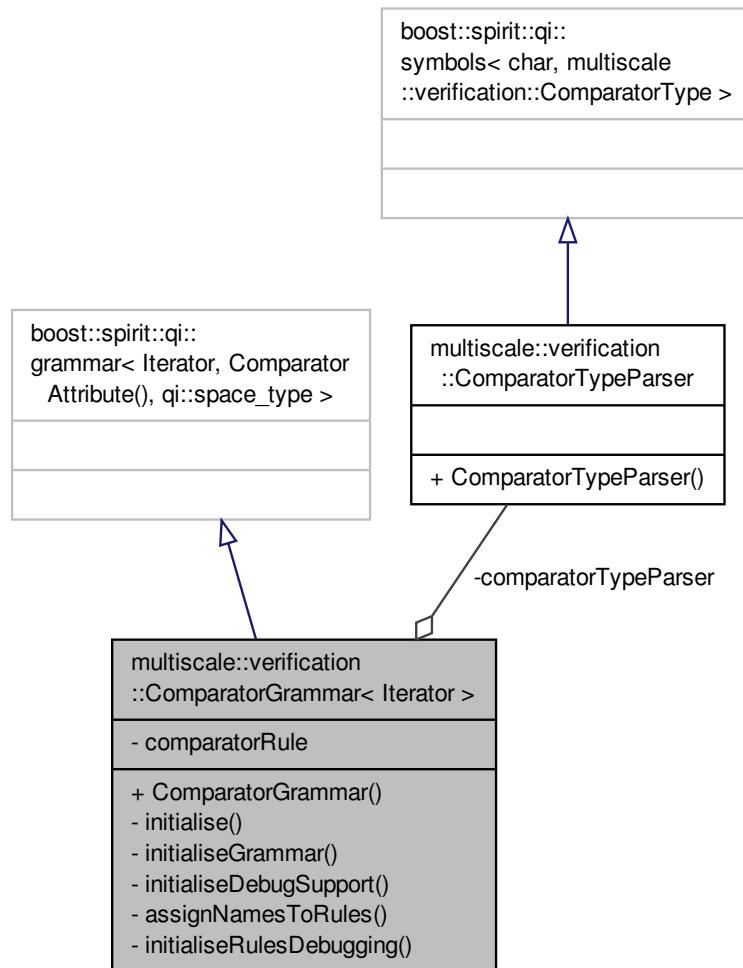
The grammar for parsing comparator statements.

```
#include <ComparatorGrammar.hpp>
```

Inheritance diagram for multiscale::verification::ComparatorGrammar< Iterator >:



Collaboration diagram for multiscale::verification::ComparatorGrammar< Iterator >:



## Public Member Functions

- [ComparatorGrammar \(\)](#)

## Private Member Functions

- `void initialise ()`  
*Initialisation function.*
- `void initialiseGrammar ()`  
*Initialise the grammar.*
- `void initialiseDebugSupport ()`  
*Initialise debug support.*
- `void assignNamesToRules ()`  
*Assign names to the rules.*
- `void initialiseRulesDebugging ()`  
*Initialise the debugging of rules.*

## Private Attributes

- `qi::rule< Iterator,`  
`ComparatorAttribute(),`  
`qi::space_type > comparatorRule`
- `ComparatorTypeParser comparatorTypeParser`

### 7.50.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::ComparatorGrammar< Iterator >
```

The grammar for parsing comparator statements.

Definition at line 30 of file ComparatorGrammar.hpp.

### 7.50.2 Constructor & Destructor Documentation

```
7.50.2.1 template<typename Iterator > multiscale::verification::ComparatorGrammar< Iterator >
>::ComparatorGrammar()
```

Definition at line 29 of file ComparatorGrammarDefinition.hpp.

References `multiscale::verification::ComparatorGrammar< Iterator >::initialise()`.

### 7.50.3 Member Function Documentation

```
7.50.3.1 template<typename Iterator > void multiscale::verification::ComparatorGrammar< Iterator >
>::assignNamesToRules() [private]
```

Assign names to the rules.

Definition at line 59 of file ComparatorGrammarDefinition.hpp.

```
7.50.3.2 template<typename Iterator > void multiscale::verification::ComparatorGrammar< Iterator >::initialise()
[private]
```

Initialisation function.

Definition at line 36 of file ComparatorGrammarDefinition.hpp.

Referenced by `multiscale::verification::ComparatorGrammar< Iterator >::ComparatorGrammar()`.

```
7.50.3.3 template<typename Iterator > void multiscale::verification::ComparatorGrammar< Iterator >
>::initialiseDebugSupport() [private]
```

Initialise debug support.

Definition at line 50 of file ComparatorGrammarDefinition.hpp.

```
7.50.3.4 template<typename Iterator > void multiscale::verification::ComparatorGrammar< Iterator >
>::initialiseGrammar() [private]
```

Initialise the grammar.

Definition at line 43 of file ComparatorGrammarDefinition.hpp.

---

```
7.50.3.5 template<typename Iterator > void multiscale::verification::ComparatorGrammar< Iterator
>::initialiseRulesDebugging() [private]
```

Initialise the debugging of rules.

Definition at line 65 of file ComparatorGrammarDefinition.hpp.

#### 7.50.4 Member Data Documentation

```
7.50.4.1 template<typename Iterator> qi::rule<Iterator, ComparatorAttribute(), qi::space_type>
multiscale::verification::ComparatorGrammar< Iterator >::comparatorRule [private]
```

The rule for parsing a comparator

Definition at line 36 of file ComparatorGrammar.hpp.

```
7.50.4.2 template<typename Iterator> ComparatorTypeParser multiscale::verification::ComparatorGrammar<
Iterator >::comparatorTypeParser [private]
```

The comparator type parser

Definition at line 41 of file ComparatorGrammar.hpp.

The documentation for this class was generated from the following files:

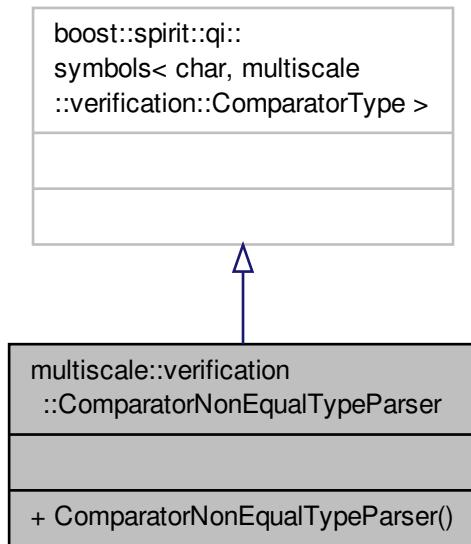
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[ComparatorGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[ComparatorGrammarDefinition.hpp](#)

## 7.51 multiscale::verification::ComparatorNonEqualTypeParser Struct Reference

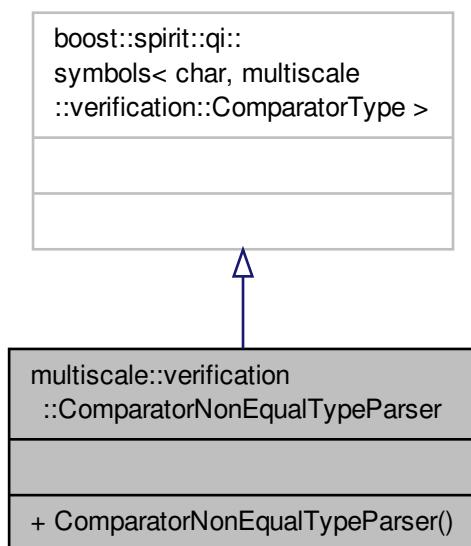
Symbol table and parser for the comparator type which does not accept the "=" symbol.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::ComparatorNonEqualTypeParser:



Collaboration diagram for multiscale::verification::ComparatorNonEqualTypeParser:



## Public Member Functions

- [ComparatorNonEqualTypeParser \(\)](#)

### 7.51.1 Detailed Description

Symbol table and parser for the comparator type which does not accept the "=" symbol.

Definition at line 83 of file [SymbolTables.hpp](#).

### 7.51.2 Constructor & Destructor Documentation

#### 7.51.2.1 multiscale::verification::ComparatorNonEqualTypeParser::ComparatorNonEqualTypeParser( ) [inline]

Definition at line 86 of file [SymbolTables.hpp](#).

References [multiscale::verification::GreaterThan](#), [multiscale::verification::GreaterThanOrEqual](#), [multiscale::verification::LessThan](#), and [multiscale::verification::LessThanOrEqual](#).

The documentation for this struct was generated from the following file:

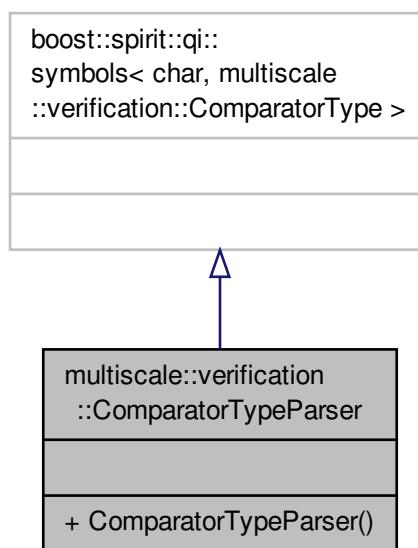
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.52 multiscale::verification::ComparatorTypeParser Struct Reference

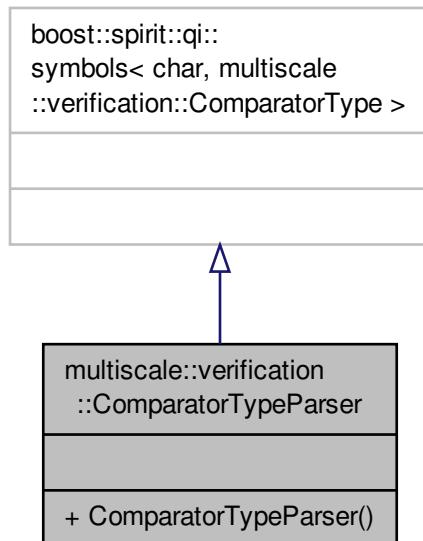
Symbol table and parser for the comparator type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::ComparatorTypeParser:



Collaboration diagram for multiscale::verification::ComparatorTypeParser:



## Public Member Functions

- [ComparatorTypeParser \(\)](#)

### 7.52.1 Detailed Description

Symbol table and parser for the comparator type.

Definition at line 98 of file [SymbolTables.hpp](#).

### 7.52.2 Constructor & Destructor Documentation

#### 7.52.2.1 multiscale::verification::ComparatorTypeParser::ComparatorTypeParser ( ) [inline]

Definition at line 101 of file [SymbolTables.hpp](#).

References `multiscale::verification::Equal`, `multiscale::verification::GreaterThan`, `multiscale::verification::GreaterThanOrEqualTo`, `multiscale::verification::LessThan`, and `multiscale::verification::LessThanOrEqualTo`.

The documentation for this struct was generated from the following file:

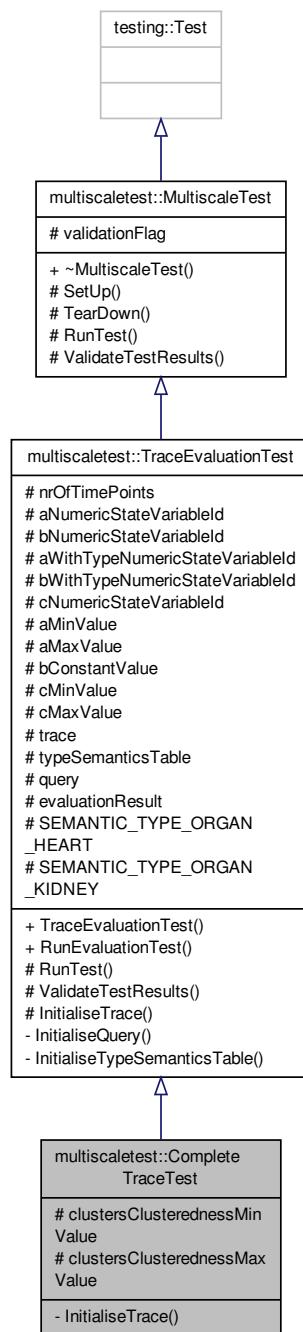
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp](#)

## 7.53 multiscaletest::CompleteTraceTest Class Reference

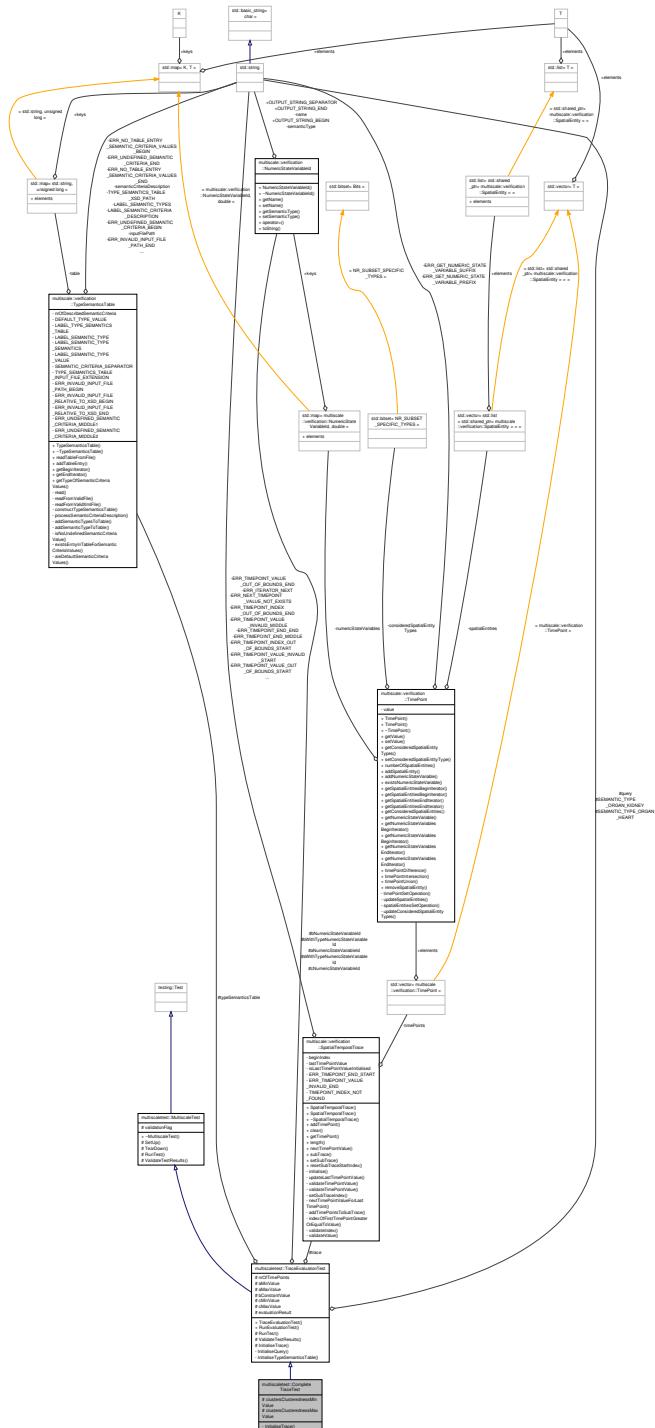
Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.

```
#include <CompleteTraceTest.hpp>
```

Inheritance diagram for multiscaletest::CompleteTraceTest:



## Collaboration diagram for multiscaletest::CompleteTraceTest::



## Protected Attributes

- double *clustersClusterednessMinValue*
  - double *clustersClusterednessMaxValue*

## Private Member Functions

- virtual void [InitialiseTrace \(\) override](#)

*Initialise the trace.*

## Additional Inherited Members

### 7.53.1 Detailed Description

Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.

Definition at line 27 of file CompleteTraceTest.hpp.

### 7.53.2 Member Function Documentation

#### 7.53.2.1 void multiscaletest::CompleteTraceTest::InitialiseTrace ( ) [override], [private], [virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 41 of file CompleteTraceTest.hpp.

### 7.53.3 Member Data Documentation

#### 7.53.3.1 double multiscaletest::CompleteTraceTest::clustersClusterednessMaxValue [protected]

The maximum clusteredness value for the cluster spatial entity type

Definition at line 32 of file CompleteTraceTest.hpp.

#### 7.53.3.2 double multiscaletest::CompleteTraceTest::clustersClusterednessMinValue [protected]

The minimum clusteredness value for the cluster spatial entity type

Definition at line 31 of file CompleteTraceTest.hpp.

The documentation for this class was generated from the following file:

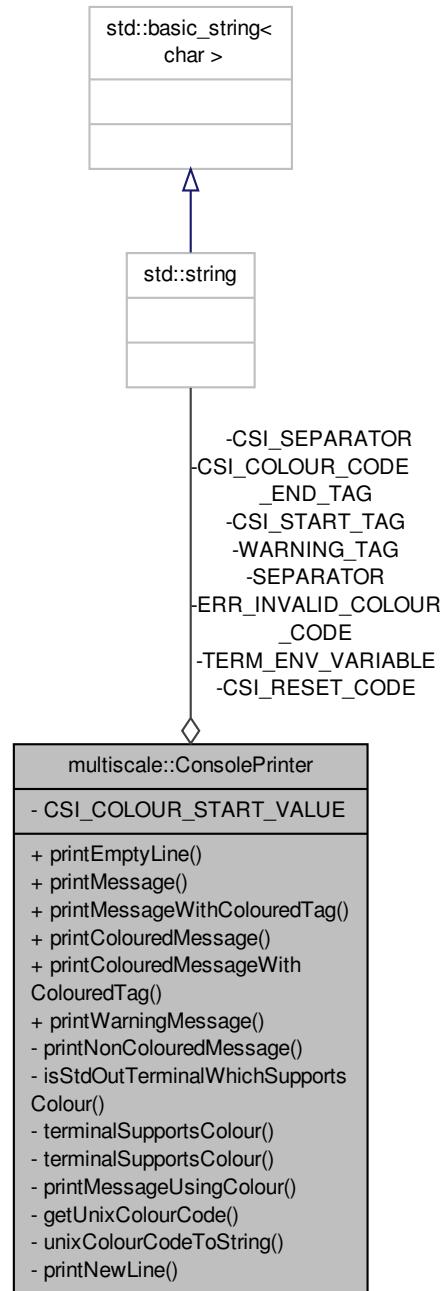
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[CompleteTraceTest.hpp](#)

## 7.54 multiscale::ConsolePrinter Class Reference

Class used to print (coloured) messages to the console.

```
#include <ConsolePrinter.hpp>
```

Collaboration diagram for multiscale::ConsolePrinter:



## Static Public Member Functions

- static void `printEmptyLine ()`  
*Print a new empty line.*
- static void `printMessage (const std::string &message)`  
*Print a message to the standard output.*

- static void `printMessageWithColouredTag` (const std::string &message, const std::string &tag, const `ColourCode` &tagColour)
 

*Print a message with a coloured tag to the standard output.*
- static void `printColouredMessage` (const std::string &message, const `ColourCode` &colourCode)
 

*Print a coloured message to the standard output.*
- static void `printColouredMessageWithColouredTag` (const std::string &message, const `ColourCode` &messageColour, const std::string &tag, const `ColourCode` &tagColour)
 

*Print a coloured message with a coloured tag to the standard output.*
- static void `printWarningMessage` (const std::string &message)
 

*Print a warning containing the given message string to the standard output.*

## Static Private Member Functions

- static void `printNonColouredMessage` (const std::string &message, bool appendNewLineAtEnd=true)
 

*Print a (non-coloured) message to the standard output.*
- static bool `isStdOutTerminalWhichSupportsColour` ()
 

*Check if the standard output is a terminal which supports colour.*
- static bool `terminalSupportsColour` (bool isTerminal)
 

*Check if the terminal supports colour.*
- static bool `terminalSupportsColour` ()
 

*Check if the terminal supports colour.*
- static void `printMessageUsingColour` (const std::string &message, const `ColourCode` &colourCode, bool appendNewLineAtEnd=true)
 

*Print a coloured message to the standard output.*
- static std::string `getUnixColourCode` (const `UnixColourCode` &unixColourCode)
 

*Get the CSI string representation corresponding to the given UNIX colour code.*
- static std::string `unixColourCodeToString` (const `UnixColourCode` &unixColourCode)
 

*Get the string representation corresponding to the given UNIX colour code.*
- static void `print.NewLine` (bool shouldPrint=true)
 

*Get the CSI string representation for resetting all attributes (including colour)*

## Static Private Attributes

- static const std::string `SEPARATOR` = " "
- static const std::string `WARNING_TAG` = "[ WARNING ]"
- static const std::string `CSI_START_TAG` = "\033["
  - static const std::string `CSI_COLOUR_CODE_END_TAG` = "m"
  - static const std::string `CSI_RESET_CODE` = "0"
  - static const std::string `CSI_SEPARATOR` = ";"
  - static const int `CSI_COLOUR_START_VALUE` = 30
  - static const std::string `TERM_ENV_VARIABLE` = "TERM"
- static const std::string `ERR_INVALID_COLOUR_CODE` = "The provided colour code is invalid. Please provide a valid colour code instead (see documentation for more details)."

### 7.54.1 Detailed Description

Class used to print (coloured) messages to the console.

Definition at line 57 of file ConsolePrinter.hpp.

## 7.54.2 Member Function Documentation

7.54.2.1 `std::string ConsolePrinter::getUnixColourCode ( const UnixColourCode & unixColourCode ) [static], [private]`

Get the CSI string representation corresponding to the given UNIX colour code.

**Parameters**

|                       |                            |
|-----------------------|----------------------------|
| <i>unixColourCode</i> | The given UNIX colour code |
|-----------------------|----------------------------|

Definition at line 205 of file ConsolePrinter.cpp.

References CSI\_COLOUR\_CODE\_END\_TAG, CSI\_RESET\_CODE, CSI\_SEPARATOR, CSI\_START\_TAG, and unixColourCodeToString().

Referenced by printMessageUsingColour().

**7.54.2.2 bool ConsolePrinter::isStdOutTerminalWhichSupportsColour( ) [static], [private]**

Check if the standard output is a terminal which supports colour.

Definition at line 76 of file ConsolePrinter.cpp.

References terminalSupportsColour().

Referenced by printColouredMessage(), printColouredMessageWithColouredTag(), and printMessageWithColouredTag().

**7.54.2.3 void ConsolePrinter::printColouredMessage( const std::string & message, const ColourCode & colourCode ) [static]**

Print a coloured message to the standard output.

The message will be printed in colour if and only if the standard output is a terminal. Otherwise it will be printed without changing colour.

**Parameters**

|                   |                                               |
|-------------------|-----------------------------------------------|
| <i>message</i>    | The given message                             |
| <i>colourCode</i> | The colour code used for printing the message |

Definition at line 42 of file ConsolePrinter.cpp.

References isStdOutTerminalWhichSupportsColour(), printMessageUsingColour(), and printNonColouredMessage().

Referenced by main(), multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResultsForLogicProperties(), multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResultsIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printEvaluationResultsSummary(), multiscale::verification::ModelCheckingOutputWriter::printFailedMessage(), multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage(), multiscale::verification::ModelCheckingOutputWriter::printIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage(), multiscale::verification::ModelCheckingOutputWriter::printResultTag(), multiscale::verification::ModelCheckingOutputWriter::printSeparatorTag(), multiscale::verification::ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage(), and multiscale::verification::ModelCheckingOutputWriter::printSuccessMessage().

**7.54.2.4 void ConsolePrinter::printColouredMessageWithColouredTag( const std::string & message, const ColourCode & messageColour, const std::string & tag, const ColourCode & tagColour ) [static]**

Print a coloured message with a coloured tag to the standard output.

**Parameters**

|                      |                                 |
|----------------------|---------------------------------|
| <i>message</i>       | The given message               |
| <i>messageColour</i> | The colour of the given message |
| <i>tag</i>           | The given tag                   |
| <i>tagColour</i>     | The colour of the given tag     |

Definition at line 51 of file ConsolePrinter.cpp.

References `isStdOutTerminalWhichSupportsColour()`, `printMessageUsingColour()`, `printNonColouredMessage()`, and `SEPARATOR`.

Referenced by `main()`.

#### 7.54.2.5 void ConsolePrinter::printEmptyLine( ) [static]

Print a new empty line.

Definition at line 22 of file ConsolePrinter.cpp.

References `print.NewLine()`.

Referenced by `multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResultsIntroductionMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printIntroductionMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage()`, and `multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesEndMessage()`.

#### 7.54.2.6 void ConsolePrinter::printMessage( const std::string & message ) [static]

Print a message to the standard output.

##### Parameters

|                |                   |
|----------------|-------------------|
| <i>message</i> | The given message |
|----------------|-------------------|

Definition at line 26 of file ConsolePrinter.cpp.

References `printNonColouredMessage()`.

Referenced by `main()`.

#### 7.54.2.7 void ConsolePrinter::printMessageUsingColour( const std::string & message, const ColourCode & colourCode, bool appendNewLineAtEnd = true ) [static], [private]

Print a coloured message to the standard output.

The message will be printed in colour if and only if the standard output is a terminal. Otherwise it will be printed using default colour.

##### Parameters

|                           |                                                                      |
|---------------------------|----------------------------------------------------------------------|
| <i>message</i>            | The given message                                                    |
| <i>colourCode</i>         | The given colour code                                                |
| <i>appendNewLineAtEnd</i> | Flag indicating if a new line character should be printed in the end |

Definition at line 111 of file ConsolePrinter.cpp.

References `getUnixColourCode()`, and `print.NewLine()`.

Referenced by `printColouredMessage()`, `printColouredMessageWithColouredTag()`, and `printMessageWithColouredTag()`.

7.54.2.8 void ConsolePrinter::printMessageWithColouredTag ( const std::string & *message*, const std::string & *tag*, const ColourCode & *tagColour* ) [static]

Print a message with a coloured tag to the standard output.

**Parameters**

|                  |                       |
|------------------|-----------------------|
| <i>message</i>   | The given message     |
| <i>tag</i>       | The given tag         |
| <i>tagColour</i> | The colour of the tag |

Definition at line 30 of file ConsolePrinter.cpp.

References `isStdOutTerminalWhichSupportsColour()`, `printMessageUsingColour()`, `printNonColouredMessage()`, and `SEPARATOR`.

Referenced by `main()`, `multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResults`←  
`IntroductionMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printEvaluationResultsSummary()`,  
`multiscale::verification::ModelCheckingOutputWriter::printExecuteExtraEvaluationProgramMessage()`, `multiscale`←  
`::verification::ModelCheckingOutputWriter::printInitialisationMessage()`, `multiscale::verification::ModelChecking`←  
`OutputWriter::printIntroductionMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printLogic`←  
`PropertyWithTag()`, `multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroduction`←  
`Message()`, `multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage()`,  
`multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage()`, `multiscale::verification`←  
`::ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage()`, `multiscale::verification::Model`←  
`CheckingOutputWriter::printStartTraceEvaluationMessage()`, `multiscale::verification::ModelCheckingOutput`←  
`Writer::printTimeoutMessage()`, `multiscale::verification::ModelCheckingOutputWriter::printTruthValueDependent`←  
`Message()`, and `printWarningMessage()`.

#### 7.54.2.9 void ConsolePrinter::print.NewLine ( bool *shouldPrint* = true ) [static], [private]

Get the CSI string representation for resetting all attributes (including colour)

Print new line character if *shouldPrint* flag is true

**Parameters**

|                    |                                                                          |
|--------------------|--------------------------------------------------------------------------|
| <i>shouldPrint</i> | Flag indicating if a new line character should be printed to the console |
|--------------------|--------------------------------------------------------------------------|

Definition at line 224 of file ConsolePrinter.cpp.

Referenced by `printEmptyLine()`, `printMessageUsingColour()`, and `printNonColouredMessage()`.

#### 7.54.2.10 void ConsolePrinter::printNonColouredMessage ( const std::string & *message*, bool *appendNewLineAtEnd* = true ) [static], [private]

Print a (non-coloured) message to the standard output.

**Parameters**

|                           |                                                                      |
|---------------------------|----------------------------------------------------------------------|
| <i>message</i>            | The given message                                                    |
| <i>appendNewLineAtEnd</i> | Flag indicating if a new line character should be printed in the end |

Definition at line 69 of file ConsolePrinter.cpp.

References `printNewLine()`.

Referenced by `printColouredMessage()`, `printColouredMessageWithColouredTag()`, `printMessage()`, and `printMessageWithColouredTag()`.

#### 7.54.2.11 void ConsolePrinter::printWarningMessage ( const std::string & *message* ) [static]

Print a warning containing the given message string to the standard output.

**Parameters**

|                |                   |
|----------------|-------------------|
| <i>message</i> | The given message |
|----------------|-------------------|

Definition at line 65 of file ConsolePrinter.cpp.

References printMessageWithColouredTag(), WARNING\_TAG, and multiscale::YELLOW.

Referenced by multiscale::OperatingSystem::executeProgram(), multiscale::Numeric::numberInverse(), multiscale::verification::LogicPropertyVisitor::printExceptionMessage(), and multiscale::Numeric::printNoValuesWarningMessage().

**7.54.2.12 bool ConsolePrinter::terminalSupportsColour ( bool *isTerminal* ) [static], [private]**

Check if the terminal supports colour.

**Parameters**

|                   |                                                      |
|-------------------|------------------------------------------------------|
| <i>isTerminal</i> | Flag indicating if the standard output is a terminal |
|-------------------|------------------------------------------------------|

Definition at line 89 of file ConsolePrinter.cpp.

References terminalSupportsColour().

**7.54.2.13 bool ConsolePrinter::terminalSupportsColour ( ) [static], [private]**

Check if the terminal supports colour.

Assumption: Standard output is a terminal

Definition at line 97 of file ConsolePrinter.cpp.

References multiscale::OperatingSystem::getEnvironmentVariable(), and TERM\_ENV\_VARIABLE.

Referenced by isStdOutTerminalWhichSupportsColour(), and terminalSupportsColour().

**7.54.2.14 std::string ConsolePrinter::unixColourCodeToString ( const UnixColourCode & *unixColourCode* ) [static], [private]**

Get the string representation corresponding to the given UNIX colour code.

**Parameters**

|                       |                            |
|-----------------------|----------------------------|
| <i>unixColourCode</i> | The given UNIX colour code |
|-----------------------|----------------------------|

Definition at line 210 of file ConsolePrinter.cpp.

References CSI\_COLOUR\_START\_VALUE, and multiscale::StringManipulator::toString().

Referenced by getUnixColourCode().

## 7.54.3 Member Data Documentation

**7.54.3.1 const std::string ConsolePrinter::CSI\_COLOUR\_CODE\_END\_TAG = "m" [static], [private]**

Definition at line 187 of file ConsolePrinter.hpp.

Referenced by getUnixColourCode().

**7.54.3.2 const int ConsolePrinter::CSI\_COLOUR\_START\_VALUE = 30 [static], [private]**

Definition at line 191 of file ConsolePrinter.hpp.

Referenced by unixColourCodeToString().

**7.54.3.3 const std::string ConsolePrinter::CSI\_RESET\_CODE = "0" [static], [private]**

Definition at line 188 of file ConsolePrinter.hpp.

Referenced by getUnixColourCode().

**7.54.3.4 const std::string ConsolePrinter::CSI\_SEPARATOR = ";" [static], [private]**

Definition at line 189 of file ConsolePrinter.hpp.

Referenced by getUnixColourCode().

**7.54.3.5 const std::string ConsolePrinter::CSI\_START\_TAG = "\033[" [static], [private]**

Definition at line 186 of file ConsolePrinter.hpp.

Referenced by getUnixColourCode().

**7.54.3.6 const std::string ConsolePrinter::ERR\_INVALID\_COLOUR\_CODE = "The provided colour code is invalid. Please provide a valid colour code instead (see documentation for more details)." [static], [private]**

Definition at line 195 of file ConsolePrinter.hpp.

**7.54.3.7 const std::string ConsolePrinter::SEPARATOR = " " [static], [private]**

Definition at line 182 of file ConsolePrinter.hpp.

Referenced by printColouredMessageWithColouredTag(), and printMessageWithColouredTag().

**7.54.3.8 const std::string ConsolePrinter::TERM\_ENV\_VARIABLE = "TERM" [static], [private]**

Definition at line 193 of file ConsolePrinter.hpp.

Referenced by terminalSupportsColour().

**7.54.3.9 const std::string ConsolePrinter::WARNING\_TAG = "[ WARNING ]" [static], [private]**

Definition at line 184 of file ConsolePrinter.hpp.

Referenced by printWarningMessage().

The documentation for this class was generated from the following files:

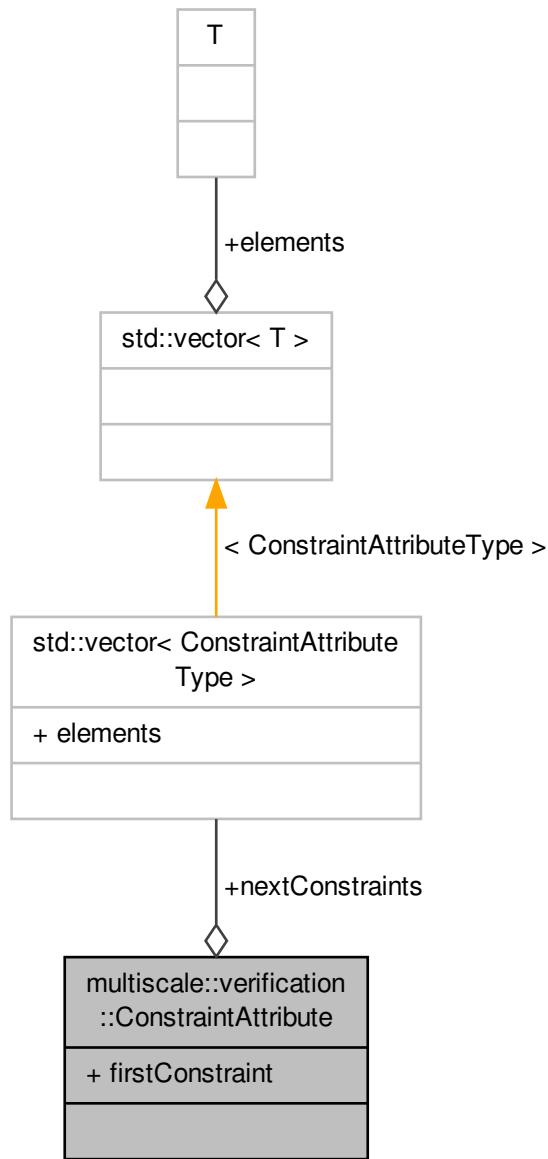
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[ConsolePrinter.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[ConsolePrinter.cpp](#)

## 7.55 multiscale::verification::ConstraintAttribute Class Reference

Class for representing a constraint attribute.

```
#include <ConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ConstraintAttribute:



## Public Attributes

- `ConstraintAttributeType firstConstraint`
- `std::vector< ConstraintAttributeType > nextConstraints`

### 7.55.1 Detailed Description

Class for representing a constraint attribute.

Definition at line 36 of file ConstraintAttribute.hpp.

## 7.55.2 Member Data Documentation

### 7.55.2.1 ConstraintAttributeType multiscale::verification::ConstraintAttribute::firstConstraint

The first constraint

Definition at line 40 of file ConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

### 7.55.2.2 std::vector<ConstraintAttributeType> multiscale::verification::ConstraintAttribute::nextConstraints

The next constraints

Definition at line 41 of file ConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::evaluateNextConstraints().

The documentation for this class was generated from the following file:

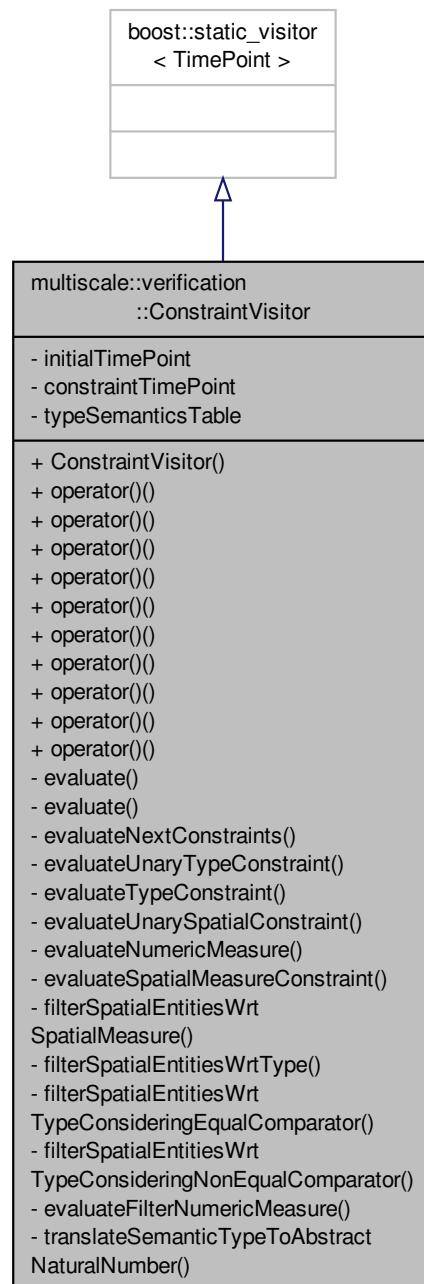
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp

## 7.56 multiscale::verification::ConstraintVisitor Class Reference

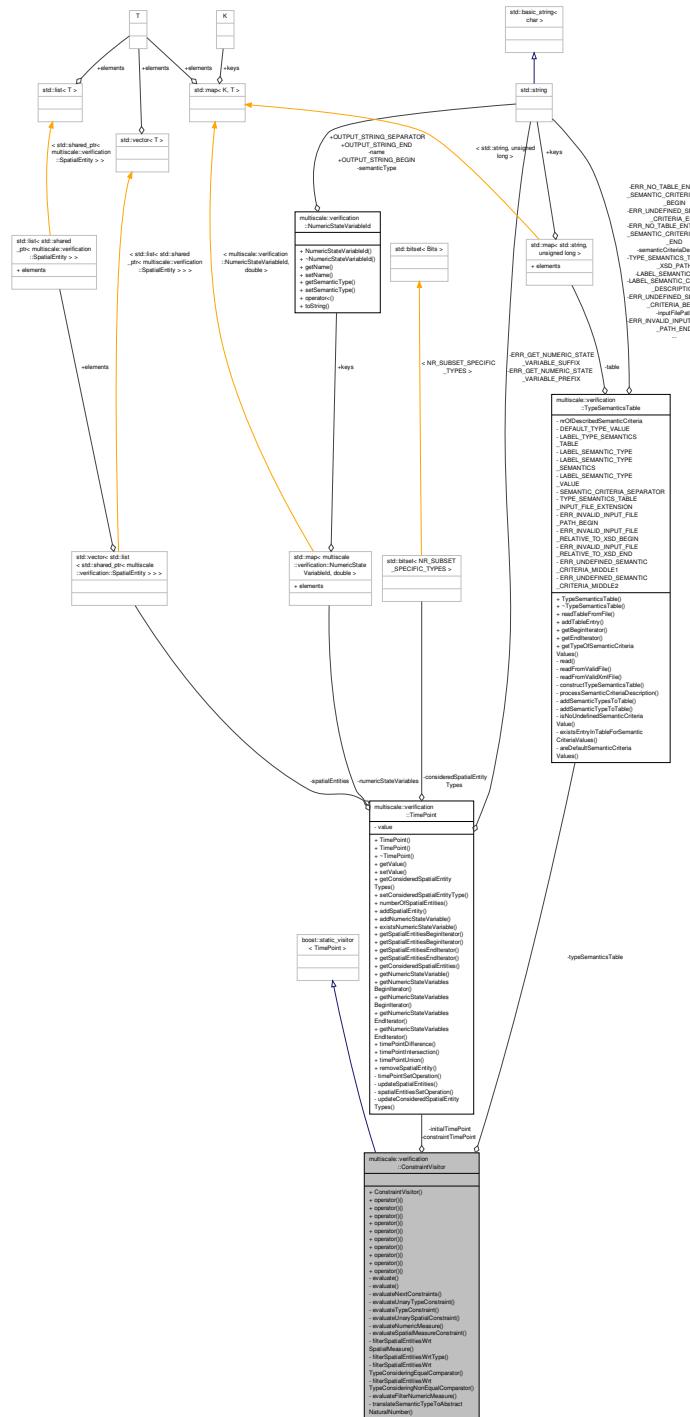
Class used to evaluate constraints.

```
#include <ConstraintVisitor.hpp>
```

Inheritance diagram for multiscale::verification::ConstraintVisitor:



## Collaboration diagram for multiscale::verification::ConstraintVisitor::



## Public Member Functions

- `ConstraintVisitor` (const `TimePoint` &`initialTimePoint`, const `TimePoint` &`constraintTimePoint`, const `Type` ←  
`SemanticsTable` &`typeSemanticsTable`)
  - `TimePoint operator()` (const `Nil` &`constraint`) const  
  
Overloading the "`()`" operator for the `Nil` alternative.
  - `TimePoint operator()` (const `ConstraintAttribute` &`constraint`) const

- Overloading the "()" operator for the `ConstraintAttribute` alternative.
- `TimePoint operator()` (const `OrConstraintAttribute` &constraint) const
  - Overloading the "()" operator for the `OrConstraintAttribute` alternative.
- `TimePoint operator()` (const `AndConstraintAttribute` &constraint) const
  - Overloading the "()" operator for the `AndConstraintAttribute` alternative.
- `TimePoint operator()` (const `ImplicationConstraintAttribute` &constraint) const
  - Overloading the "()" operator for the `ImplicationConstraintAttribute` alternative.
- `TimePoint operator()` (const `EquivalenceConstraintAttribute` &constraint) const
  - Overloading the "()" operator for the `EquivalenceConstraintAttribute` alternative.
- `TimePoint operator()` (const `PrimaryConstraintAttribute` &primaryConstraint) const
  - Overloading the "()" operator for the `PrimaryConstraintAttribute` alternative.
- `TimePoint operator()` (const `NotConstraintAttribute` &primaryConstraint) const
  - Overloading the "()" operator for the `NotConstraintAttribute` alternative.
- `TimePoint operator()` (const `UnaryTypeConstraintAttribute` &primaryConstraint) const
  - Overloading the "()" operator for the `UnaryTypeConstraintAttribute` alternative.
- `TimePoint operator()` (const `UnarySpatialConstraintAttribute` &primaryConstraint) const
  - Overloading the "()" operator for the `UnarySpatialConstraintAttribute` alternative.

## Private Member Functions

- `TimePoint evaluate` (const `ConstraintAttributeType` &constraint, const `TimePoint` &timePoint) const
  - Evaluate the constraint considering the given timepoint.
- `TimePoint evaluate` (const `PrimaryConstraintAttributeType` &primaryConstraint, const `TimePoint` &timePoint) const
  - Evaluate the primary constraint considering the given timepoints.
- `TimePoint evaluateNextConstraints` (const `ConstraintAttribute` &constraint, const `TimePoint` &timePoint) const
  - Evaluate the next constraints.
- `TimePoint evaluateUnaryTypeConstraint` (const `ComparatorType` &comparator, const `SemanticTypeAttribute` &semanticType, const `TimePoint` &timePoint) const
  - Evaluate the unary type constraint.
- `void evaluateTypeConstraint` (`TimePoint` &timePoint, const `ComparatorType` &comparator, const `SemanticTypeAttribute` &semanticType) const
  - Filter the timepoint's spatial entities considering the type of each spatial entity.
- `TimePoint evaluateUnarySpatialConstraint` (const `SpatialMeasureType` &spatialMeasure, const `ComparatorType` &comparator, const `FilterNumericMeasureAttributeType` &filterNumericMeasure, const `TimePoint` &timePoint) const
  - Evaluate the unary spatial constraint.
- `double evaluateNumericMeasure` (const `NumericMeasureType` &numericMeasure, const `TimePoint` &timePoint) const
  - Evaluate the numeric measure considering the given timepoint.
- `void evaluateSpatialMeasureConstraint` (`TimePoint` &timePoint, const `SpatialMeasureType` &spatialMeasure, const `ComparatorType` &comparator, const `FilterNumericMeasureAttributeType` &filterNumericMeasure) const
  - Filter the timepoint's spatial entities considering the given spatial measure constraint.
- `void filterSpatialEntitiesWrtSpatialMeasure` (`TimePoint` &timePoint, const `SubsetSpecificType` &spatialEntityType, const `SpatialMeasureType` &spatialMeasure, const `ComparatorType` &comparator, const `FilterNumericMeasureAttributeType` &filterNumericMeasure) const
  - Remove from the timepoint the spatial entities which fail to meet the spatial measure constraint.
- `void filterSpatialEntitiesWrtType` (`TimePoint` &timePoint, const `SubsetSpecificType` &spatialEntityType, const `ComparatorType` &comparator, const `SemanticTypeAttribute` &semanticType) const
  - Remove from the timepoint the spatial entities which fail to meet the type constraint.

- void `filterSpatialEntitiesWrtTypeConsideringEqualComparator` (`TimePoint` &`timePoint`, const `SubsetSpecificType` &`spatialEntityType`, const std::string &`rhsSemanticType`) const  
*Remove from the timepoint the spatial entities which fail to meet the type constraint.*
- void `filterSpatialEntitiesWrtTypeConsideringNonEqualComparator` (`TimePoint` &`timePoint`, const `SubsetSpecificType` &`spatialEntityType`, const `ComparatorType` &`comparator`, const std::string &`rhsSemanticType`) const  
*Remove from the timepoint the spatial entities which fail to meet the type constraint.*
- double `evaluateFilterNumericMeasure` (const `FilterNumericMeasureAttributeType` &`filterNumericMeasure`, const `TimePoint` &`timePoint`, const `SpatialEntity` &`spatialEntity`) const  
*Evaluate the filter numeric measure considering the provided timepoint and spatial entity.*
- double `translateSemanticTypeToAbstractNaturalNumber` (const std::string &`semanticType`) const  
*Translate the given semantic type to an abstract natural number.*

## Private Attributes

- const `TimePoint` & `initialTimePoint`
- const `TimePoint` & `constraintTimePoint`
- const `TypeSemanticsTable` & `typeSemanticsTable`

### 7.56.1 Detailed Description

Class used to evaluate constraints.

Definition at line 19 of file ConstraintVisitor.hpp.

### 7.56.2 Constructor & Destructor Documentation

7.56.2.1 `multiscale::verification::ConstraintVisitor::ConstraintVisitor` ( const `TimePoint` & `initialTimePoint`, const `TimePoint` & `constraintTimePoint`, const `TypeSemanticsTable` & `typeSemanticsTable` ) [inline]

Definition at line 29 of file ConstraintVisitor.hpp.

Referenced by `evaluate()`, and `evaluateNextConstraints()`.

### 7.56.3 Member Function Documentation

7.56.3.1 `TimePoint multiscale::verification::ConstraintVisitor::evaluate` ( const `ConstraintAttributeType` & `constraint`, const `TimePoint` & `timePoint` ) const [inline], [private]

Evaluate the constraint considering the given timepoint.

#### Parameters

|                         |                      |
|-------------------------|----------------------|
| <code>constraint</code> | The given constraint |
| <code>timePoint</code>  | The given timepoint  |

Definition at line 165 of file ConstraintVisitor.hpp.

References `ConstraintVisitor()`.

Referenced by `operator()`.

7.56.3.2 `TimePoint multiscale::verification::ConstraintVisitor::evaluate` ( const `PrimaryConstraintAttributeType` & `primaryConstraint`, const `TimePoint` & `timePoint` ) const [inline], [private]

Evaluate the primary constraint considering the given timepoints.

**Parameters**

|                          |                              |
|--------------------------|------------------------------|
| <i>primaryConstraint</i> | The given primary constraint |
| <i>timePoint</i>         | The given timepoint          |

Definition at line 175 of file ConstraintVisitor.hpp.

References ConstraintVisitor().

**7.56.3.3 double multiscale::verification::ConstraintVisitor::evaluateFilterNumericMeasure ( const FilterNumericMeasureAttributeType & *filterNumericMeasure*, const TimePoint & *timePoint*, const SpatialEntity & *spatialEntity* ) const [inline], [private]**

Evaluate the filter numeric measure considering the provided timepoint and spatial entity.

**Parameters**

|                             |                               |
|-----------------------------|-------------------------------|
| <i>filterNumericMeasure</i> | The filter numeric measure    |
| <i>timePoint</i>            | The considered timepoint      |
| <i>spatialEntity</i>        | The considered spatial entity |

Definition at line 430 of file ConstraintVisitor.hpp.

Referenced by filterSpatialEntitiesWrtSpatialMeasure().

**7.56.3.4 TimePoint multiscale::verification::ConstraintVisitor::evaluateNextConstraints ( const ConstraintAttribute & *constraint*, const TimePoint & *timePoint* ) const [inline], [private]**

Evaluate the next constraints.

Evaluate the next constraints considering the given constraint and timepoints

**Parameters**

|                   |                                                                                      |
|-------------------|--------------------------------------------------------------------------------------|
| <i>constraint</i> | The given constraint                                                                 |
| <i>timePoint</i>  | The resulting timepoint after applying the first constraint to the initial timepoint |

Definition at line 189 of file ConstraintVisitor.hpp.

References ConstraintVisitor(), and multiscale::verification::ConstraintAttribute::nextConstraints.

Referenced by operator()().

**7.56.3.5 double multiscale::verification::ConstraintVisitor::evaluateNumericMeasure ( const NumericMeasureType & *numericMeasure*, const TimePoint & *timePoint* ) const [inline], [private]**

Evaluate the numeric measure considering the given timepoint.

**Parameters**

|                       |                     |
|-----------------------|---------------------|
| <i>numericMeasure</i> | The numeric measure |
| <i>timePoint</i>      | The given timepoint |

Definition at line 269 of file ConstraintVisitor.hpp.

**7.56.3.6 void multiscale::verification::ConstraintVisitor::evaluateSpatialMeasureConstraint ( TimePoint & *timePoint*, const SpatialMeasureType & *spatialMeasure*, const ComparatorType & *comparator*, const FilterNumericMeasureAttributeType & *filterNumericMeasure* ) const [inline], [private]**

Filter the timepoint's spatial entities considering the given spatial measure constraint.

All considered spatial entities which fail to meet the constraints will be removed from the given timepoint.

**Parameters**

|                             |                                                                                 |
|-----------------------------|---------------------------------------------------------------------------------|
| <i>timePoint</i>            | The timepoint storing the collection of spatial entities which will be filtered |
| <i>spatialMeasure</i>       | The type of the spatial measure                                                 |
| <i>comparator</i>           | The type of the comparator                                                      |
| <i>filterNumericMeasure</i> | The filter numeric measure                                                      |

Definition at line 286 of file ConstraintVisitor.hpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::TimePoint::getConsideredSpatialEntityTypes(), and multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES.

Referenced by evaluateUnarySpatialConstraint().

**7.56.3.7 void multiscale::verification::ConstraintVisitor::evaluateTypeConstraint ( TimePoint & *timePoint*, const ComparatorType & *comparator*, const SemanticTypeAttribute & *semanticType* ) const [inline], [private]**

Filter the timepoint's spatial entities considering the type of each spatial entity.

**Parameters**

|                     |                                                                                 |
|---------------------|---------------------------------------------------------------------------------|
| <i>timePoint</i>    | The timepoint storing the collection of spatial entities which will be filtered |
| <i>comparator</i>   | The type of the comparator                                                      |
| <i>semanticType</i> | The semantic type                                                               |

Definition at line 228 of file ConstraintVisitor.hpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), filterSpatialEntitiesWrtType(), multiscale::verification::TimePoint::getConsideredSpatialEntityTypes(), and multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES.

Referenced by evaluateUnaryTypeConstraint().

**7.56.3.8 TimePoint multiscale::verification::ConstraintVisitor::evaluateUnarySpatialConstraint ( const SpatialMeasureType & *spatialMeasure*, const ComparatorType & *comparator*, const FilterNumericMeasureAttributeType & *filterNumericMeasure*, const TimePoint & *timePoint* ) const [inline], [private]**

Evaluate the unary spatial constraint.

Evaluate the unary spatial constraint considering the given spatial measure, comparator, numeric measure and timepoint

**Parameters**

|                             |                            |
|-----------------------------|----------------------------|
| <i>spatialMeasure</i>       | The spatial measure type   |
| <i>comparator</i>           | The comparator type        |
| <i>filterNumericMeasure</i> | The filter numeric measure |
| <i>timePoint</i>            | The considered timepoint   |

Definition at line 252 of file ConstraintVisitor.hpp.

References evaluateSpatialMeasureConstraint().

Referenced by operator()().

**7.56.3.9 TimePoint multiscale::verification::ConstraintVisitor::evaluateUnaryTypeConstraint ( const ComparatorType & comparator, const SemanticTypeAttribute & semanticType, const TimePoint & timePoint ) const [inline], [private]**

Evaluate the unary type constraint.

Evaluate the unary type constraint considering the given spatial measure, comparator, semantic type and timepoint

#### Parameters

|                     |                          |
|---------------------|--------------------------|
| <i>comparator</i>   | The comparator type      |
| <i>semanticType</i> | The semantic type        |
| <i>timePoint</i>    | The considered timepoint |

Definition at line 211 of file ConstraintVisitor.hpp.

References evaluateTypeConstraint().

Referenced by operator()().

**7.56.3.10 void multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtSpatialMeasure ( TimePoint & timePoint, const SubsetSpecificType & spatialEntityType, const SpatialMeasureType & spatialMeasure, const ComparatorType & comparator, const FilterNumericMeasureAttributeType & filterNumericMeasure ) const [inline], [private]**

Remove from the timepoint the spatial entities which fail to meet the spatial measure constraint.

#### Parameters

|                             |                                      |
|-----------------------------|--------------------------------------|
| <i>timePoint</i>            | The timepoint which will be filtered |
| <i>spatialEntityType</i>    | The considered spatial entity type   |
| <i>spatialMeasure</i>       | The type of the spatial measure      |
| <i>comparator</i>           | The type of the comparator           |
| <i>filterNumericMeasure</i> | The filter numeric measure           |

Definition at line 312 of file ConstraintVisitor.hpp.

References multiscale::verification::SpatialMeasureEvaluator::evaluate(), multiscale::verification::ComparatorEvaluator::evaluate(), evaluateFilterNumericMeasure(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), and multiscale::verification::TimePoint::removeSpatialEntity().

Referenced by evaluateSpatialMeasureConstraint().

**7.56.3.11 void multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtType ( TimePoint & timePoint, const SubsetSpecificType & spatialEntityType, const ComparatorType & comparator, const SemanticTypeAttribute & semanticType ) const [inline], [private]**

Remove from the timepoint the spatial entities which fail to meet the type constraint.

#### Parameters

|                          |                                      |
|--------------------------|--------------------------------------|
| <i>timePoint</i>         | The timepoint which will be filtered |
| <i>spatialEntityType</i> | The considered spatial entity type   |
| <i>comparator</i>        | The type of the comparator           |
| <i>semanticType</i>      | The semantic type                    |

Definition at line 345 of file ConstraintVisitor.hpp.

References multiscale::verification::Equal, filterSpatialEntitiesWrtTypeConsideringEqualComparator(), filterSpatialEntitiesWrtTypeConsideringNonEqualComparator(), and multiscale::verification::SemanticTypeAttribute::semanticType.

Referenced by evaluateTypeConstraint().

```
7.56.3.12 void multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringEqualComparator (
 TimePoint & timePoint, const SubsetSpecificType & spatialEntityType, const std::string & rhsSemanticType)
 const [inline], [private]
```

Remove from the timepoint the spatial entities which fail to meet the type constraint.

The assumption for this method is that the considered comparator is "=".

In this case the type semantics table is NOT used.

#### Parameters

|                          |                                                  |
|--------------------------|--------------------------------------------------|
| <i>timePoint</i>         | The timepoint which will be filtered             |
| <i>spatialEntityType</i> | The considered spatial entity type               |
| <i>rhsSemanticType</i>   | The semantic type on the right of the comparator |

Definition at line 372 of file ConstraintVisitor.hpp.

References multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), and multiscale::verification::TimePoint::removeSpatialEntity().

Referenced by filterSpatialEntitiesWrtType().

```
7.56.3.13 void multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringNonEqualComparator (
 TimePoint & timePoint, const SubsetSpecificType & spatialEntityType, const ComparatorType &
 comparator, const std::string & rhsSemanticType) const [inline], [private]
```

Remove from the timepoint the spatial entities which fail to meet the type constraint.

The assumption for this method is that the considered comparator is different from "=".

In this case the type semantics table is used.

#### Parameters

|                          |                                                  |
|--------------------------|--------------------------------------------------|
| <i>timePoint</i>         | The timepoint which will be filtered             |
| <i>spatialEntityType</i> | The considered spatial entity type               |
| <i>comparator</i>        | The type of the comparator                       |
| <i>rhsSemanticType</i>   | The semantic type on the right of the comparator |

Definition at line 401 of file ConstraintVisitor.hpp.

References multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), multiscale::verification::TimePoint::removeSpatialEntity(), and translateSemanticTypeToAbstractNaturalNumber().

Referenced by filterSpatialEntitiesWrtType().

```
7.56.3.14 TimePoint multiscale::verification::ConstraintVisitor::operator() (const Nil & constraint) const [inline]
```

Overloading the "()" operator for the [Nil](#) alternative.

#### Parameters

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 38 of file ConstraintVisitor.hpp.

References initialTimePoint.

7.56.3.15 **TimePoint multiscale::verification::ConstraintVisitor::operator()** ( **const ConstraintAttribute & constraint** ) const  
[inline]

Overloading the "(" operator for the [ConstraintAttribute](#) alternative.

**Parameters**

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 46 of file ConstraintVisitor.hpp.

References evaluate(), evaluateNextConstraints(), and multiscale::verification::ConstraintAttribute::firstConstraint.

**7.56.3.16 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const OrConstraintAttribute & *constraint* ) const [inline]**

Overloading the "(" operator for the [OrConstraintAttribute](#) alternative.

**Parameters**

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 56 of file ConstraintVisitor.hpp.

References multiscale::verification::OrConstraintAttribute::constraint, evaluate(), and multiscale::verification::TimePoint::timePointUnion().

**7.56.3.17 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const AndConstraintAttribute & *constraint* ) const [inline]**

Overloading the "(" operator for the [AndConstraintAttribute](#) alternative.

**Parameters**

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 68 of file ConstraintVisitor.hpp.

References multiscale::verification::AndConstraintAttribute::constraint, evaluate(), and multiscale::verification::TimePoint::timePointIntersection().

**7.56.3.18 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const ImplicationConstraintAttribute & *constraint* ) const [inline]**

Overloading the "(" operator for the [ImplicationConstraintAttribute](#) alternative.

**Parameters**

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 80 of file ConstraintVisitor.hpp.

References multiscale::verification::ImplicationConstraintAttribute::constraint, evaluate(), multiscale::verification::TimePoint::timePointDifference(), and multiscale::verification::TimePoint::timePointUnion().

**7.56.3.19 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const EquivalenceConstraintAttribute & *constraint* ) const [inline]**

Overloading the "(" operator for the [EquivalenceConstraintAttribute](#) alternative.

**Parameters**

|                   |                |
|-------------------|----------------|
| <i>constraint</i> | The constraint |
|-------------------|----------------|

Definition at line 95 of file ConstraintVisitor.hpp.

References multiscale::verification::EquivalenceConstraintAttribute::constraint, evaluate(), multiscale::verification::TimePoint::timePointDifference(), multiscale::verification::TimePoint::timePointIntersection(), and multiscale::

::verification::TimePoint::timePointUnion().

**7.56.3.20 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const PrimaryConstraintAttribute & primaryConstraint ) const [inline]**

Overloading the "(" operator for the [PrimaryConstraintAttribute](#) alternative.

**Parameters**

|                          |                        |
|--------------------------|------------------------|
| <i>primaryConstraint</i> | The primary constraint |
|--------------------------|------------------------|

Definition at line 118 of file ConstraintVisitor.hpp.

References [evaluate\(\)](#), and [multiscale::verification::PrimaryConstraintAttribute::primaryConstraint](#).

**7.56.3.21 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const NotConstraintAttribute & primaryConstraint ) const [inline]**

Overloading the "(" operator for the [NotConstraintAttribute](#) alternative.

**Parameters**

|                          |                        |
|--------------------------|------------------------|
| <i>primaryConstraint</i> | The primary constraint |
|--------------------------|------------------------|

Definition at line 126 of file ConstraintVisitor.hpp.

References [multiscale::verification::NotConstraintAttribute::constraint](#), [evaluate\(\)](#), and [multiscale::verification::TimePoint::timePointDifference\(\)](#).

**7.56.3.22 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const UnaryTypeConstraintAttribute & primaryConstraint ) const [inline]**

Overloading the "(" operator for the [UnaryTypeConstraintAttribute](#) alternative.

**Parameters**

|                          |                        |
|--------------------------|------------------------|
| <i>primaryConstraint</i> | The primary constraint |
|--------------------------|------------------------|

Definition at line 139 of file ConstraintVisitor.hpp.

References [multiscale::verification::UnaryTypeConstraintAttribute::comparator](#), [multiscale::verification::ComparatorAttribute::comparatorType](#), [evaluateUnaryTypeConstraint\(\)](#), and [multiscale::verification::UnaryTypeConstraintAttribute::semanticType](#).

**7.56.3.23 TimePoint multiscale::verification::ConstraintVisitor::operator() ( const UnarySpatialConstraintAttribute & primaryConstraint ) const [inline]**

Overloading the "(" operator for the [UnarySpatialConstraintAttribute](#) alternative.

**Parameters**

|                          |                        |
|--------------------------|------------------------|
| <i>primaryConstraint</i> | The primary constraint |
|--------------------------|------------------------|

Definition at line 150 of file ConstraintVisitor.hpp.

References [multiscale::verification::UnarySpatialConstraintAttribute::comparator](#), [multiscale::verification::ComparatorAttribute::comparatorType](#), [evaluateUnarySpatialConstraint\(\)](#), [multiscale::verification::UnarySpatialAttribute::semanticType](#).

ConstraintAttribute::filterNumericMeasure, multiscale::verification::UnarySpatialConstraintAttribute::spatialMeasure, and multiscale::verification::SpatialMeasureAttribute::spatialMeasureType.

**7.56.3.24 double multiscale::verification::ConstraintVisitor::translateSemanticTypeToAbstractNaturalNumber ( const std::string & *semanticType* ) const [inline], [private]**

Translate the given semantic type to an abstract natural number.

The type semantics table is used to compute the natural number corresponding to the semantic type.

#### Parameters

|                     |                              |
|---------------------|------------------------------|
| <i>semanticType</i> | The considered semantic type |
|---------------------|------------------------------|

Definition at line 443 of file ConstraintVisitor.hpp.

References multiscale::verification::TypeSemanticsTable::getTypeOfSemanticCriteriaValues().

Referenced by filterSpatialEntitiesWrtTypeConsideringNonEqualComparator().

## 7.56.4 Member Data Documentation

**7.56.4.1 const TimePoint& multiscale::verification::ConstraintVisitor::constraintTimePoint [private]**

The currently obtained constraint timepoint

Definition at line 24 of file ConstraintVisitor.hpp.

**7.56.4.2 const TimePoint& multiscale::verification::ConstraintVisitor::initialTimePoint [private]**

A copy of the initial timepoint

Definition at line 23 of file ConstraintVisitor.hpp.

Referenced by operator()().

**7.56.4.3 const TypeSemanticsTable& multiscale::verification::ConstraintVisitor::typeSemanticsTable [private]**

The considered type semantics table

Definition at line 25 of file ConstraintVisitor.hpp.

The documentation for this class was generated from the following file:

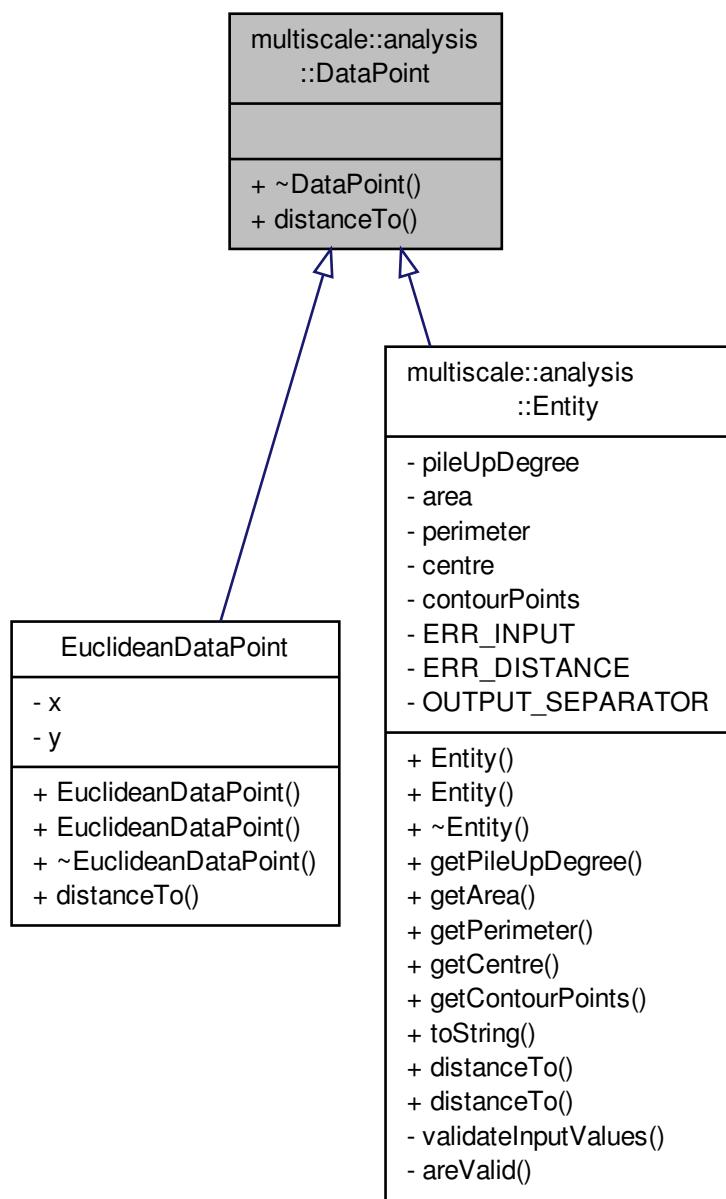
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[ConstraintVisitor.hpp](#)

## 7.57 multiscale::analysis::DataPoint Class Reference

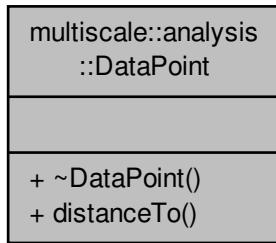
Class for representing a data point.

```
#include <DataPoint.hpp>
```

Inheritance diagram for multiscale::analysis::DataPoint:



Collaboration diagram for multiscale::analysis::DataPoint:



## Public Member Functions

- virtual `~DataPoint ()`
- virtual double `distanceTo (std::shared_ptr< DataPoint > point)=0`

*Compute the distance between this data point and another one.*

### 7.57.1 Detailed Description

Class for representing a data point.

Definition at line 12 of file DataPoint.hpp.

### 7.57.2 Constructor & Destructor Documentation

#### 7.57.2.1 virtual multiscale::analysis::DataPoint::~DataPoint ( ) [inline], [virtual]

Definition at line 16 of file DataPoint.hpp.

### 7.57.3 Member Function Documentation

#### 7.57.3.1 virtual double multiscale::analysis::DataPoint::distanceTo ( std::shared\_ptr< DataPoint > point ) [pure virtual]

Compute the distance between this data point and another one.

##### Parameters

|                    |                                              |
|--------------------|----------------------------------------------|
| <code>point</code> | Data point to which the distance is measured |
|--------------------|----------------------------------------------|

Implemented in [multiscale::analysis::Entity](#), and [EuclideanDataPoint](#).

The documentation for this class was generated from the following file:

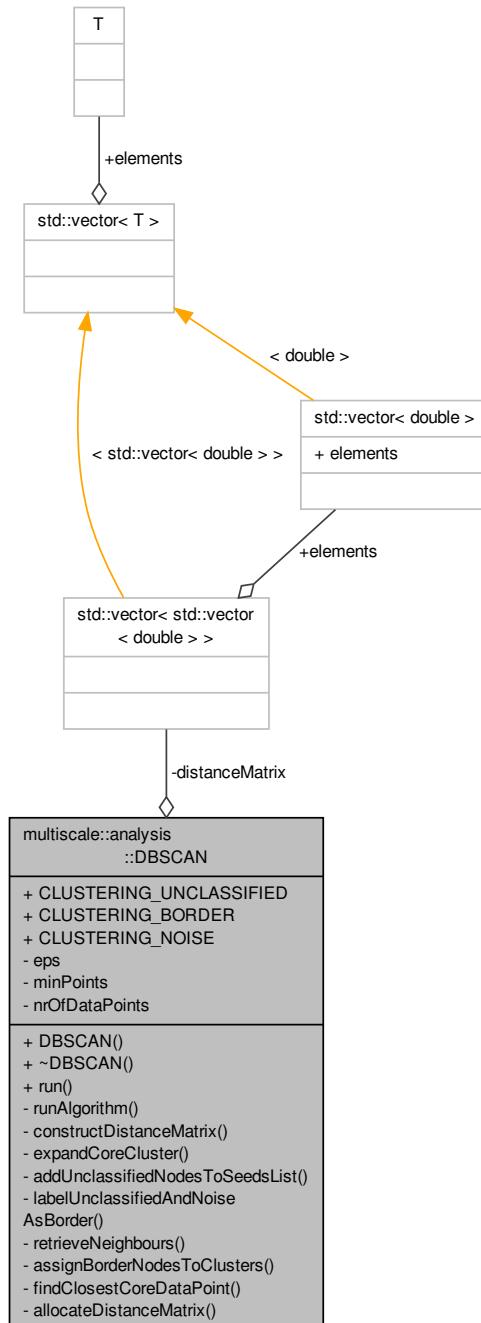
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[DataPoint.hpp](#)

## 7.58 multiscale::analysis::DBSCAN Class Reference

Class which implements an improved version of the [DBSCAN](#) algorithm.

```
#include <DBSCAN.hpp>
```

Collaboration diagram for multiscale::analysis::DBSCAN:



## Public Member Functions

- `DBSCAN ()`
- `~DBSCAN ()`
- `void run (const std::vector< std::shared_ptr< DataPoint >> &dataPoints, std::vector< int > &clusterIndexes, int &nrOfClusters, double eps, int minPoints)`

*Run the improved DBSCAN algorithm on the provided set of points.*

## Static Public Attributes

- `static const int CLUSTERING_UNCLASSIFIED = -2`
- `static const int CLUSTERING_BORDER = -1`
- `static const int CLUSTERING_NOISE = 0`

## Private Member Functions

- `void runAlgorithm (const std::vector< std::shared_ptr< DataPoint >> &dataPoints, std::vector< int > &clusterIndexes, int &nrOfClusters)`

*Run the improved DBSCAN algorithm on the provided set of points.*
- `void constructDistanceMatrix (const std::vector< std::shared_ptr< DataPoint >> &dataPoints)`

*Construct the distance matrix between any two data points.*
- `bool expandCoreCluster (std::vector< int > &clusterIndexes, int coreDataPointIndex, int clusterId)`

*Expand the cluster around the given core data point.*
- `void addUnclassifiedNodesToSeedsList (const std::vector< int > &neighbours, const std::vector< int > &clusterIndexes, std::vector< int > &seeds)`

*Add all unclassified neighbour nodes to the seeds list.*
- `void labelUnclassifiedAndNoiseAsBorder (const std::vector< int > &neighbours, std::vector< int > &clusterIndexes)`

*Label all unclassified and noise neighbour nodes as border nodes.*
- `std::vector< int > retrieveNeighbours (int dataPointIndex)`

*Retrieve the list of neighbour indexes which are at a distance < eps far from the given data point.*
- `void assignBorderNodesToClusters (std::vector< int > &clusterIndexes)`

*Assign the border nodes to the clusters to which the closest core objects belong.*
- `int findClosestCoreDataPoint (const std::vector< int > &neighbours, int borderDataPointIndex, const std::vector< int > &clusterIndexes)`

*Find the closest core data point from the given set of neighbours to the given border data point.*
- `void allocateDistanceMatrix ()`

*Allocate the distance matrix.*

## Private Attributes

- `double eps`
- `unsigned int minPoints`
- `unsigned int nrOfDataPoints`
- `std::vector< std::vector< double > > distanceMatrix`

### 7.58.1 Detailed Description

Class which implements an improved version of the DBSCAN algorithm.

Definition at line 15 of file DBSCAN.hpp.

## 7.58.2 Constructor & Destructor Documentation

### 7.58.2.1 DBSCAN::DBSCAN( )

Definition at line 9 of file DBSCAN.cpp.

### 7.58.2.2 DBSCAN::~DBSCAN( )

Definition at line 11 of file DBSCAN.cpp.

References distanceMatrix.

## 7.58.3 Member Function Documentation

### 7.58.3.1 void DBSCAN::addUnclassifiedNodesToSeedsList ( const std::vector< int > & neighbours, const std::vector< int > & clusterIndexes, std::vector< int > & seeds ) [private]

Add all unclassified neighbour nodes to the seeds list.

Parameters

|                       |                                                      |
|-----------------------|------------------------------------------------------|
| <i>neighbours</i>     | Neighbour nodes                                      |
| <i>clusterIndexes</i> | Indexes to which cluster each data point belongs     |
| <i>seeds</i>          | List of seeds (see <a href="#">DBSCAN</a> algorithm) |

Definition at line 87 of file DBSCAN.cpp.

References CLUSTERING\_UNCLASSIFIED.

Referenced by expandCoreCluster().

### 7.58.3.2 void DBSCAN::allocateDistanceMatrix( ) [private]

Allocate the distance matrix.

Definition at line 152 of file DBSCAN.cpp.

References distanceMatrix, and nrOfDataPoints.

Referenced by constructDistanceMatrix().

### 7.58.3.3 void DBSCAN::assignBorderNodesToClusters ( std::vector< int > & clusterIndexes ) [private]

Assign the border nodes to the clusters to which the closest core objects belong.

Parameters

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| <i>clusterIndexes</i> | Indexes to which cluster each data point belongs |
|-----------------------|--------------------------------------------------|

Definition at line 121 of file DBSCAN.cpp.

References CLUSTERING\_BORDER, findClosestCoreDataPoint(), nrOfDataPoints, and retrieveNeighbours().

Referenced by runAlgorithm().

### 7.58.3.4 void DBSCAN::constructDistanceMatrix ( const std::vector< std::shared\_ptr< DataPoint >> & dataPoints ) [private]

Construct the distance matrix between any two data points.

## Parameters

|                   |             |
|-------------------|-------------|
| <i>dataPoints</i> | Data points |
|-------------------|-------------|

Definition at line 46 of file DBSCAN.cpp.

References allocateDistanceMatrix(), distanceMatrix, and nrOfDataPoints.

Referenced by run().

**7.58.3.5 bool DBSCAN::expandCoreCluster ( std::vector< int > & *clusterIndexes*, int *coreDataPointIndex*, int *clusterId* ) [private]**

Expand the cluster around the given core data point.

## Parameters

|                           |                                                        |
|---------------------------|--------------------------------------------------------|
| <i>clusterIndexes</i>     | Indexes to which cluster each data point belongs       |
| <i>coreDataPointIndex</i> | Core data point index                                  |
| <i>clusterId</i>          | Id of the cluster to which the core data point belongs |

Definition at line 59 of file DBSCAN.cpp.

References addUnclassifiedNodesToSeedsList(), CLUSTERING\_NOISE, labelUnclassifiedAndNoiseAsBorder(), minPoints, and retrieveNeighbours().

Referenced by runAlgorithm().

**7.58.3.6 int DBSCAN::findClosestCoreDataPoint ( const std::vector< int > & *neighbours*, int *borderDataPointIndex*, const std::vector< int > & *clusterIndexes* ) [private]**

Find the closest core data point from the given set of neighbours to the given border data point.

## Parameters

|                             |                                                  |
|-----------------------------|--------------------------------------------------|
| <i>neighbours</i>           | Set of neighbours                                |
| <i>borderDataPointIndex</i> | Index of the border data point                   |
| <i>clusterIndexes</i>       | Indexes to which cluster each data point belongs |

Definition at line 132 of file DBSCAN.cpp.

References distanceMatrix.

Referenced by assignBorderNodesToClusters().

**7.58.3.7 void DBSCAN::labelUnclassifiedAndNoiseAsBorder ( const std::vector< int > & *neighbours*, std::vector< int > & *clusterIndexes* ) [private]**

Label all unclassified and noise neighbour nodes as border nodes.

## Parameters

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| <i>neighbours</i>     | Neighbour nodes                                  |
| <i>clusterIndexes</i> | Indexes to which cluster each data point belongs |

Definition at line 97 of file DBSCAN.cpp.

References CLUSTERING\_BORDER, CLUSTERING\_NOISE, and CLUSTERING\_UNCLASSIFIED.

Referenced by expandCoreCluster().

**7.58.3.8 std::vector< int > DBSCAN::retrieveNeighbours ( int *dataPointIndex* ) [private]**

Retrieve the list of neighbour indexes which are at a distance < *eps* far from the given data point.

**Parameters**

|                       |                                                                    |
|-----------------------|--------------------------------------------------------------------|
| <i>dataPointIndex</i> | Index of the data point for which the neighbours will be retrieved |
|-----------------------|--------------------------------------------------------------------|

Definition at line 107 of file DBSCAN.cpp.

References distanceMatrix, eps, and nrOfDataPoints.

Referenced by assignBorderNodesToClusters(), and expandCoreCluster().

**7.58.3.9 void DBSCAN::run ( const std::vector< std::shared\_ptr< **DataPoint** >> & *dataPoints*, std::vector< int > & *clusterIndexes*, int & *nrOfClusters*, double *eps*, int *minPoints* )**

Run the improved **DBSCAN** algorithm on the provided set of points.

The implementation of the improved **DBSCAN** algorithm is based on the paper: T. N. Tran, K. Drab, and M. Daszykowski, ‘Revised **DBSCAN** algorithm to cluster data with dense adjacent clusters’, Chemometrics and Intelligent Laboratory Systems, vol. 120, pp. 92–96, Jan. 2013.

Clusters start from index 1, because cluster 0 contains only noise data/points.

**Parameters**

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| <i>dataPoints</i>     | Collection of data points                        |
| <i>clusterIndexes</i> | Indexes to which cluster each data point belongs |
| <i>nrOfClusters</i>   | Total number of clusters                         |
| <i>eps</i>            | Maximum distance between two neighbours          |
| <i>minPoints</i>      | Minimum number of points in one cluster          |

Definition at line 15 of file DBSCAN.cpp.

References constructDistanceMatrix(), eps, minPoints, nrOfDataPoints, and runAlgorithm().

Referenced by multiscale::analysis::ClusterDetector::detectClusters(), and runTest().

**7.58.3.10 void DBSCAN::runAlgorithm ( const std::vector< std::shared\_ptr< **DataPoint** >> & *dataPoints*, std::vector< int > & *clusterIndexes*, int & *nrOfClusters* ) [private]**

Run the improved **DBSCAN** algorithm on the provided set of points.

The implementation of the improved **DBSCAN** algorithm is based on the paper: T. N. Tran, K. Drab, and M. Daszykowski, ‘Revised **DBSCAN** algorithm to cluster data with dense adjacent clusters’, Chemometrics and Intelligent Laboratory Systems, vol. 120, pp. 92–96, Jan. 2013.

Clusters start from index 1, because cluster 0 contains only noise data/points.

**Parameters**

|                       |                                                  |
|-----------------------|--------------------------------------------------|
| <i>dataPoints</i>     | Collection of data points                        |
| <i>clusterIndexes</i> | Indexes to which cluster each data point belongs |
| <i>nrOfClusters</i>   | Total number of clusters                         |

Definition at line 27 of file DBSCAN.cpp.

References assignBorderNodesToClusters(), CLUSTERING\_UNCLASSIFIED, expandCoreCluster(), and nrOfDataPoints.

Referenced by run().

## 7.58.4 Member Data Documentation

**7.58.4.1 const int DBSCAN::CLUSTERING\_BORDER = -1 [static]**

Definition at line 130 of file DBSCAN.hpp.

Referenced by `assignBorderNodesToClusters()`, and `labelUnclassifiedAndNoiseAsBorder()`.

#### 7.58.4.2 `const int DBSCAN::CLUSTERING_NOISE = 0 [static]`

Definition at line 131 of file `DBSCAN.hpp`.

Referenced by `expandCoreCluster()`, and `labelUnclassifiedAndNoiseAsBorder()`.

#### 7.58.4.3 `const int DBSCAN::CLUSTERING_UNCLASSIFIED = -2 [static]`

Definition at line 129 of file `DBSCAN.hpp`.

Referenced by `addUnclassifiedNodesToSeedsList()`, `multiscale::analysis::ClusterDetector::detectAndAnalyseClusters()`, `labelUnclassifiedAndNoiseAsBorder()`, and `runAlgorithm()`.

#### 7.58.4.4 `std::vector<std::vector<double>> multiscale::analysis::DBSCAN::distanceMatrix [private]`

The matrix containing the distances between any two data points

Definition at line 27 of file `DBSCAN.hpp`.

Referenced by `allocateDistanceMatrix()`, `constructDistanceMatrix()`, `findClosestCoreDataPoint()`, `retrieveNeighbours()`, and `~DBSCAN()`.

#### 7.58.4.5 `double multiscale::analysis::DBSCAN::eps [private]`

`DBSCAN` algorithm parameter for specifying the maximum radius of the neighbourhood

Definition at line 19 of file `DBSCAN.hpp`.

Referenced by `retrieveNeighbours()`, and `run()`.

#### 7.58.4.6 `unsigned int multiscale::analysis::DBSCAN::minPoints [private]`

`DBSCAN` algorithm parameter for specifying the minimum number of points in an `eps`-neighbourhood of that point

Definition at line 21 of file `DBSCAN.hpp`.

Referenced by `expandCoreCluster()`, and `run()`.

#### 7.58.4.7 `unsigned int multiscale::analysis::DBSCAN::nrOfDataPoints [private]`

Number of data points in the data set

Definition at line 24 of file `DBSCAN.hpp`.

Referenced by `allocateDistanceMatrix()`, `assignBorderNodesToClusters()`, `constructDistanceMatrix()`, `retrieveNeighbours()`, `run()`, and `runAlgorithm()`.

The documentation for this class was generated from the following files:

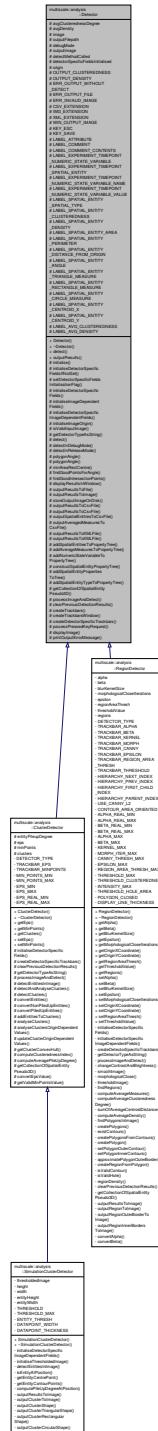
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/DBSCAN.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/DBSCAN.cpp](#)

## 7.59 multiscale::analysis::Detector Class Reference

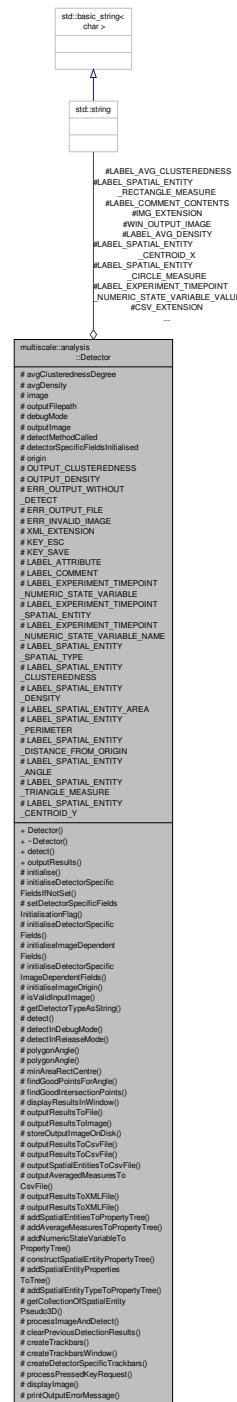
Abstract class for detecting entities of interest in images.

```
#include <Detector.hpp>
```

## Inheritance diagram for multiscale::analysis::Detector:



## Collaboration diagram for multiscale::analysis::Detector::



## Public Member Functions

- `Detector` (`bool debugMode=false`)
  - `virtual ~Detector ()`
  - `void detect (const cv::Mat &inputImage)`

*Run the detection procedure on the given image.*

- void outputResults (const std::string &outputFilepath)

*Output the results to the given file.*

## Protected Member Functions

- void **initialise ()**  
*Initialisation function for the class.*
- void **initialiseDetectorSpecificFieldsIfNotSet ()**  
*Initialisation of the detector specific values in case they were not set.*
- void **setDetectorSpecificFieldsInitialisationFlag (bool flag=true)**  
*Set the detector specific fields initialisation flag to true.*
- virtual void **initialiseDetectorSpecificFields ()=0**  
*Initialisation of the detector specific values.*
- void **initialiseImageDependentFields ()**  
*Initialisation of the image dependent values.*
- virtual void **initialiseDetectorSpecificImageDependentFields ()=0**  
*Initialisation of the detector specific image dependent values.*
- void **initialiseImageOrigin ()**
- bool **isValidInputImage (const cv::Mat &inputImage)**  
*Check if the image is valid.*
- virtual std::string **getDetectorTypeAsString ()=0**  
*Get the type of the employed detector as a std::string.*
- void **detect ()**  
*Run the detection procedure.*
- void **detectInDebugMode ()**  
*Run the detection procedure when in debug mode.*
- void **detectInReleaseMode ()**  
*Run the detection procedure when in release mode (i.e. non-debug mode)*
- double **polygonAngle (const std::vector< cv::Point > &polygon, unsigned int closestPointIndex)**  
*Compute the angle of the polygon.*
- double **polygonAngle (const std::vector< cv::Point > &polygonConvexHull, const cv::Point &closestPoint)**  
*Compute the angle of the polygon.*
- void **minAreaRectCentre (const std::vector< cv::Point > &polygon, cv::Point &centre)**  
*Get the centre of the minimum area bounding rectangle.*
- void **findGoodPointsForAngle (const std::vector< cv::Point > &polygonConvexHull, const cv::Point &boundingRectCentre, const cv::Point &closestPoint, std::vector< cv::Point > &goodPointsForAngle)**  
*Find the points for determining the angle of the polygon.*
- void **findGoodIntersectionPoints (const std::vector< cv::Point > &polygonConvexHull, const cv::Point &edgePointA, const cv::Point &edgePointB, std::vector< cv::Point > &goodPointsForAngle)**  
*Find good intersection points for computing the angle of the polygon.*
- void **displayResultsInWindow ()**  
*Display the results in a window.*
- void **outputResultsToFile ()**  
*Output the results to file(s).*
- virtual void **outputResultsToImage ()=0**  
*Output the results to an image.*
- void **storeOutputImageOnDisk ()**  
*Store the image with the output results on disk.*
- void **outputResultsToCsvFile ()**  
*Output the results to a csv file.*
- void **outputResultsToCsvFile (std::ofstream &fout)**  
*Output the results to a file using the provided output stream.*

- void `outputSpatialEntitiesToCsvFile` (std::ofstream &fout)  
*Output the pseudo 3D spatial entities to a csv file.*
- void `outputAveragedMeasuresToCsvFile` (std::ofstream &fout)  
*Output the averaged measures to a csv file.*
- void `outputResultsToXMLFile` ()  
*Output the results to an xml file.*
- void `outputResultsToXMLFile` (const std::string &filepath)  
*Output the clusters and averaged measures to an xml file.*
- void `addSpatialEntitiesToPropertyTree` (pt::ptree &propertyTree)  
*Add the pseudo 3D spatial entities to the property tree.*
- void `addAverageMeasuresToPropertyTree` (pt::ptree &propertyTree)  
*Add the average clusteredness and average density to the property tree.*
- void `addNumericStateVariableToPropertyTree` (pt::ptree &propertyTree, const std::string &name, double value)  
*Add a numeric state variable to the property tree.*
- pt::ptree `constructSpatialEntityPropertyTree` (SpatialEntityPseudo3D &spatialEntity)  
*Construct the property tree corresponding to the given spatial entity.*
- void `addSpatialEntityPropertiesToTree` (SpatialEntityPseudo3D &spatialEntity, pt::ptree &propertyTree)  
*Add the properties of the spatial entity to the property tree.*
- void `addSpatialEntityTypeToPropertyTree` (SpatialEntityPseudo3D &spatialEntity, pt::ptree &propertyTree)  
*Add the type of the spatial entity to the property tree.*
- virtual std::vector  
< std::shared\_ptr  
< SpatialEntityPseudo3D > > `getCollectionOfSpatialEntityPseudo3D` ()=0  
*Get the collection of pseudo 3D entities detected in the image.*
- virtual void `processImageAndDetect` ()=0  
*Process the input image and detect objects/entities of interest.*
- virtual void `clearPreviousDetectionResults` ()=0  
*Clear the results from the previous detection.*
- void `createTrackbars` ()  
*Create the trackbars which allow the user to change the values of the parameters.*
- void `createTrackbarsWindow` ()  
*Create the window in which the trackbars are placed.*
- virtual void `createDetectorSpecificTrackbars` ()=0  
*Create the trackbars specific to the used detector.*
- void `processPressedKeyRequest` (char &pressedKey)  
*Process the request of the user by pressing the key.*
- void `displayImage` (const cv::Mat &image, const std::string &windowName)  
*Display an image in a particular window.*
- void `printOutputErrorMessage` ()  
*Print error message, because the detect method was not called before calling the output method.*

## Protected Attributes

- double avgClusterednessDegree
- double avgDensity
- cv::Mat image
- std::string outputPath
- bool debugMode
- cv::Mat outputImage
- bool detectMethodCalled
- bool detectorSpecificFieldsInitialised
- cv::Point origin

## Static Protected Attributes

- static const std::string **OUTPUT\_CLUSTEREDNESS** = "Average clusteredness degree: "
- static const std::string **OUTPUT\_DENSITY** = "Average density: "
- static const std::string **ERR\_OUTPUT\_WITHOUT\_DETECT** = "Unable to output results if the `detect` method was not called previously."
- static const std::string **ERR\_OUTPUT\_FILE** = "Unable to create output file."
- static const std::string **ERR\_INVALID\_IMAGE** = "The input `image` is invalid."
- static const std::string **CSV\_EXTENSION** = ".out"
- static const std::string **IMG\_EXTENSION** = ".png"
- static const std::string **XML\_EXTENSION** = ".xml"
- static const std::string **WIN\_OUTPUT\_IMAGE** = "Output `image`"
- static const int **KEY\_ESC** = 27
- static const int **KEY\_SAVE** = 115
- static const std::string **LABEL\_ATTRIBUTE** = "<xmlattr>"
- static const std::string **LABEL\_COMMENT** = "<xmlcomment>"
- static const std::string **LABEL\_COMMENT\_CONTENTS** = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."
- static const std::string **LABEL\_EXPERIMENT\_TIMEPOINT\_NUMERIC\_STATE\_VARIABLE** = "experiment.timepoint.numericStateVariable"
- static const std::string **LABEL\_EXPERIMENT\_TIMEPOINT\_SPATIAL\_ENTITY** = "experiment.timepoint.spatialEntity"
- static const std::string **LABEL\_EXPERIMENT\_TIMEPOINT\_NUMERIC\_STATE\_VARIABLE\_NAME** = "name"
- static const std::string **LABEL\_EXPERIMENT\_TIMEPOINT\_NUMERIC\_STATE\_VARIABLE\_VALUE** = "value"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_SPATIAL\_TYPE** = "spatialType"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_CLUSTEREDNESS** = "clusteredness"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_DENSITY** = "density"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_AREA** = "area"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_PERIMETER** = "perimeter"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_DISTANCE\_FROM\_ORIGIN** = "distanceFromOrigin"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_ANGLE** = "angle"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_TRIANGLE\_MEASURE** = "triangleMeasure"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_RECTANGLE\_MEASURE** = "rectangleMeasure"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_CIRCLE\_MEASURE** = "circleMeasure"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_CENTROID\_X** = "centroidX"
- static const std::string **LABEL\_SPATIAL\_ENTITY\_CENTROID\_Y** = "centroidY"
- static const std::string **LABEL\_AVG\_CLUSTEREDNESS** = "avgClusteredness"
- static const std::string **LABEL\_AVG\_DENSITY** = "avgDensity"

### 7.59.1 Detailed Description

Abstract class for detecting entities of interest in images.

Definition at line 22 of file Detector.hpp.

### 7.59.2 Constructor & Destructor Documentation

#### 7.59.2.1 `Detector::Detector( bool debugMode = false )`

Definition at line 11 of file Detector.cpp.

References avgClusterednessDegree, avgDensity, debugMode, detectMethodCalled, and detectorSpecificFields. Initialised.

### 7.59.2.2 `Detector::~Detector( ) [virtual]`

Definition at line 21 of file Detector.cpp.

References image, and outputImage.

## 7.59.3 Member Function Documentation

### 7.59.3.1 `void Detector::addAverageMeasuresToPropertyTree( pt::ptree & propertyTree ) [protected]`

Add the average clusteredness and average density to the property tree.

#### Parameters

|                           |                   |
|---------------------------|-------------------|
| <code>propertyTree</code> | The property tree |
|---------------------------|-------------------|

Definition at line 255 of file Detector.cpp.

References addNumericStateVariableToPropertyTree(), avgClusterednessDegree, avgDensity, getDetectorType<→ AsString(), LABEL\_AVG\_CLUSTEREDNESS, and LABEL\_AVG\_DENSITY.

Referenced by outputResultsToXMLFile().

### 7.59.3.2 `void Detector::addNumericStateVariableToPropertyTree( pt::ptree & propertyTree, const std::string & name, double value ) [protected]`

Add a numeric state variable to the property tree.

#### Parameters

|                           |                                         |
|---------------------------|-----------------------------------------|
| <code>propertyTree</code> | The property tree                       |
| <code>name</code>         | The name of the numeric state variable  |
| <code>value</code>        | The value of the numeric state variable |

Definition at line 264 of file Detector.cpp.

References LABEL\_EXPERIMENT\_TIMEPOINT\_NUMERIC\_STATE\_VARIABLE, LABEL\_EXPERIMENT\_TIME<→ POINT\_NUMERIC\_STATE\_VARIABLE\_NAME, and LABEL\_EXPERIMENT\_TIMEPOINT\_NUMERIC\_STATE\_V<→ ARIABLE\_VALUE.

Referenced by addAverageMeasuresToPropertyTree().

### 7.59.3.3 `void Detector::addSpatialEntitiesToPropertyTree( pt::ptree & propertyTree ) [protected]`

Add the pseudo 3D spatial entities to the property tree.

#### Parameters

|                           |                   |
|---------------------------|-------------------|
| <code>propertyTree</code> | The property tree |
|---------------------------|-------------------|

Definition at line 245 of file Detector.cpp.

References constructSpatialEntityPropertyTree(), getCollectionOfSpatialEntityPseudo3D(), and LABEL\_EXPERI<→ MENT\_TIMEPOINT\_SPATIAL\_ENTITY.

Referenced by outputResultsToXMLFile().

### 7.59.3.4 `void Detector::addSpatialEntityPropertiesToTree( SpatialEntityPseudo3D & spatialEntity, pt::ptree & propertyTree ) [protected]`

Add the properties of the spatial entity to the property tree.

**Parameters**

|                      |                |
|----------------------|----------------|
| <i>spatialEntity</i> | Spatial entity |
| <i>propertyTree</i>  | Property tree  |

Definition at line 283 of file Detector.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::getAngle(), multiscale::analysis::SpatialEntityPseudo3D::getArea(), multiscale::analysis::SpatialEntityPseudo3D::getCentre(), multiscale::analysis::SpatialEntityPseudo3D::getCircularMeasure(), multiscale::analysis::SpatialEntityPseudo3D::getClusterednessDegree(), multiscale::analysis::SpatialEntityPseudo3D::getDensity(), multiscale::analysis::SpatialEntityPseudo3D::getDistanceFromOrigin(), multiscale::analysis::SpatialEntityPseudo3D::getPerimeter(), multiscale::analysis::SpatialEntityPseudo3D::getRectangularMeasure(), multiscale::analysis::SpatialEntityPseudo3D::getTriangularMeasure(), LABEL\_SPATIAL\_ENTITY\_ANGLE, LABEL\_SPATIAL\_ENTITY\_AREA, LABEL\_SPATIAL\_ENTITY\_CENTROID\_X, LABEL\_SPATIAL\_ENTITY\_CENTROID\_Y, LABEL\_SPATIAL\_ENTITY\_CIRCLE\_MEASURE, LABEL\_SPATIAL\_ENTITY\_CLUSTEREDNESS, LABEL\_SPATIAL\_ENTITY\_DENSITY, LABEL\_SPATIAL\_ENTITY\_DISTANCE\_FROM\_ORIGIN, LABEL\_SPATIAL\_ENTITY\_PERIMETER, LABEL\_SPATIAL\_ENTITY\_RECTANGLE\_MEASURE, and LABEL\_SPATIAL\_ENTITY\_TRIANGLE\_MEASURE.

Referenced by constructSpatialEntityPropertyTree().

#### 7.59.3.5 void Detector::addSpatialEntityTypeToPropertyTree ( *SpatialEntityPseudo3D & spatialEntity*, *pt::ptree & propertyTree* ) [protected]

Add the type of the spatial entity to the property tree.

**Parameters**

|                      |                |
|----------------------|----------------|
| <i>spatialEntity</i> | Spatial entity |
| <i>propertyTree</i>  | Property tree  |

Definition at line 297 of file Detector.cpp.

References LABEL\_ATTRIBUTE, LABEL\_SPATIAL\_ENTITY\_SPATIAL\_TYPE, and multiscale::analysis::SpatialEntityPseudo3D::typeAsString().

Referenced by constructSpatialEntityPropertyTree().

#### 7.59.3.6 virtual void multiscale::analysis::Detector::clearPreviousDetectionResults ( ) [protected], [pure virtual]

Clear the results from the previous detection.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

Referenced by detectInDebugMode(), and detectInReleaseMode().

#### 7.59.3.7 *pt::ptree* Detector::constructSpatialEntityPropertyTree ( *SpatialEntityPseudo3D & spatialEntity* ) [protected]

Construct the property tree corresponding to the given spatial entity.

**Parameters**

|                      |                                    |
|----------------------|------------------------------------|
| <i>spatialEntity</i> | The spatial entity to be converted |
|----------------------|------------------------------------|

Definition at line 274 of file Detector.cpp.

References addSpatialEntityPropertiesToTree(), and addSpatialEntityTypeToPropertyTree().

Referenced by addSpatialEntitiesToPropertyTree().

7.59.3.8 `virtual void multiscale::analysis::Detector::createDetectorSpecificTrackbars( ) [protected], [pure virtual]`

Create the trackbars specific to the used detector.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

Referenced by `createTrackbars()`.

7.59.3.9 `void Detector::createTrackbars( ) [protected]`

Create the trackbars which allow the user to change the values of the parameters.

Definition at line 306 of file `Detector.cpp`.

References `createDetectorSpecificTrackbars()`, and `createTrackbarsWindow()`.

Referenced by `detectInDebugMode()`.

7.59.3.10 `void Detector::createTrackbarsWindow( ) [protected]`

Create the window in which the trackbars are placed.

Definition at line 311 of file `Detector.cpp`.

References `WIN_OUTPUT_IMAGE`.

Referenced by `createTrackbars()`.

7.59.3.11 `void Detector::detect( const cv::Mat & inputImage )`

Run the detection procedure on the given image.

Parameters

|                                |                 |
|--------------------------------|-----------------|
| <code><i>inputImage</i></code> | The input image |
|--------------------------------|-----------------|

Definition at line 26 of file `Detector.cpp`.

References `detect()`, `ERR_INVALID_IMAGE`, `image`, `initialise()`, `isValidInputImage()`, and `MS_throw`.

Referenced by `main()`.

7.59.3.12 `void Detector::detect( ) [protected]`

Run the detection procedure.

Definition at line 85 of file `Detector.cpp`.

References `debugMode`, `detectInDebugMode()`, `detectInReleaseMode()`, and `detectMethodCalled`.

Referenced by `detect()`.

7.59.3.13 `void Detector::detectInDebugMode( ) [protected]`

Run the detection procedure when in debug mode.

Definition at line 95 of file `Detector.cpp`.

References `clearPreviousDetectionResults()`, `createTrackbars()`, `displayResultsInWindow()`, `KEY_ESC`, `processImageAndDetect()`, and `processPressedKeyRequest()`.

Referenced by `detect()`.

## 7.59.3.14 void Detector::detectInReleaseMode( ) [protected]

Run the detection procedure when in release mode (i.e. non-debug mode)

Definition at line 110 of file Detector.cpp.

References clearPreviousDetectionResults(), and processImageAndDetect().

Referenced by detect().

7.59.3.15 void Detector::displayImage( const cv::Mat & *image*, const std::string & *windowName* ) [protected]

Display an image in a particular window.

**Parameters**

|                   |                        |
|-------------------|------------------------|
| <i>image</i>      | The image              |
| <i>windowName</i> | The name of the window |

Definition at line 322 of file Detector.cpp.

Referenced by displayResultsInWindow().

## 7.59.3.16 void Detector::displayResultsInWindow( ) [protected]

Display the results in a window.

Definition at line 171 of file Detector.cpp.

References displayImage(), outputImage, outputResultsToImage(), and WIN\_OUTPUT\_IMAGE.

Referenced by detectInDebugMode().

7.59.3.17 void Detector::findGoodIntersectionPoints( const std::vector< cv::Point > & *polygonConvexHull*, const cv::Point & *edgePointA*, const cv::Point & *edgePointB*, std::vector< cv::Point > & *goodPointsForAngle* ) [protected]

Find good intersection points for computing the angle of the polygon.

**Parameters**

|                           |                                           |
|---------------------------|-------------------------------------------|
| <i>polygonConvexHull</i>  | The convex hull of the polygon            |
| <i>edgePointA</i>         | cv::Point A on the edge                   |
| <i>edgePointB</i>         | cv::Point B on the edge                   |
| <i>goodPointsForAngle</i> | The "good" points for computing the angle |

Definition at line 155 of file Detector.cpp.

References multiscale::Geometry2D::lineSegmentIntersection().

Referenced by findGoodPointsForAngle().

7.59.3.18 void Detector::findGoodPointsForAngle( const std::vector< cv::Point > & *polygonConvexHull*, const cv::Point & *boundingRectCentre*, const cv::Point & *closestPoint*, std::vector< cv::Point > & *goodPointsForAngle* ) [protected]

Find the points for determining the angle of the polygon.

## Parameters

|                           |                                                                         |
|---------------------------|-------------------------------------------------------------------------|
| <i>polygonConvexHull</i>  | Convex hull of polygon                                                  |
| <i>boundingRectCentre</i> | Centre of the rotated rectangle enclosing the polygon convex hull       |
| <i>closestPoint</i>       | Closest point to the origin from the set of points defining the polygon |
| <i>goodPointsForAngle</i> | The points which are relevant for computing the angle                   |

Definition at line 143 of file Detector.cpp.

References [findGoodIntersectionPoints\(\)](#), [image](#), and [multiscale::Geometry2D::orthogonalLineToAnotherLineEdgePoints\(\)](#).

Referenced by [polygonAngle\(\)](#).

**7.59.3.19** `virtual std::vector<std::shared_ptr<SpatialEntityPseudo3D>> multiscale::analysis::Detector::getCollectionOfSpatialEntityPseudo3D ( ) [protected], [pure virtual]`

Get the collection of pseudo 3D entities detected in the image.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

Referenced by [addSpatialEntitiesToPropertyTree\(\)](#), and [outputSpatialEntitiesToCsvFile\(\)](#).

**7.59.3.20** `virtual std::string multiscale::analysis::Detector::getDetectorTypeAsString ( ) [protected], [pure virtual]`

Get the type of the employed detector as a std::string.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

Referenced by [addAverageMeasuresToPropertyTree\(\)](#).

**7.59.3.21** `void Detector::initialise ( ) [protected]`

Initialisation function for the class.

Definition at line 47 of file Detector.cpp.

References [initialiseDetectorSpecificFieldsIfNotSet\(\)](#), and [initialiseImageDependentFields\(\)](#).

Referenced by [detect\(\)](#).

**7.59.3.22** `virtual void multiscale::analysis::Detector::initialiseDetectorSpecificFields ( ) [protected], [pure virtual]`

Initialisation of the detector specific values.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

Referenced by [initialiseDetectorSpecificFieldsIfNotSet\(\)](#).

**7.59.3.23** `void Detector::initialiseDetectorSpecificFieldsIfNotSet ( ) [protected]`

Initialisation of the detector specific values in case they were not set.

Definition at line 52 of file Detector.cpp.

References [detectorSpecificFieldsInitialised](#), and [initialiseDetectorSpecificFields\(\)](#).

Referenced by initialise().

**7.59.3.24 virtual void multiscale::analysis::Detector::initialiseDetectorSpecificImageDependentFields( ) [protected], [pure virtual]**

Initialisation of the detector specific image dependent values.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::SimulationClusterDetector](#).

Referenced by initialiseImageDependentFields().

**7.59.3.25 void Detector::initialiseImageDependentFields( ) [protected]**

Initialisation of the image dependent values.

Definition at line 64 of file Detector.cpp.

References initialiseDetectorSpecificImageDependentFields(), and initialiseImageOrigin().

Referenced by initialise().

**7.59.3.26 void Detector::initialiseImageOrigin( ) [protected]**

Definition at line 69 of file Detector.cpp.

References image, and origin.

Referenced by initialiseImageDependentFields().

**7.59.3.27 bool Detector::isValidInputImage( const cv::Mat & *inputImage* ) [protected]**

Check if the image is valid.

Check if the number of dimensions = 2, if the number of rows and number of columns is greater than one and if the image is of type CV\_8UC1

#### Parameters

|                   |                 |
|-------------------|-----------------|
| <i>inputImage</i> | The input image |
|-------------------|-----------------|

Definition at line 76 of file Detector.cpp.

Referenced by detect().

**7.59.3.28 void Detector::minAreaRectCentre( const std::vector< cv::Point > & *polygon*, cv::Point & *centre* ) [protected]**

Get the centre of the minimum area bounding rectangle.

#### Parameters

|                |                                      |
|----------------|--------------------------------------|
| <i>polygon</i> | The polygon                          |
| <i>centre</i>  | The centre of the bounding rectangle |

Definition at line 137 of file Detector.cpp.

Referenced by polygonAngle().

**7.59.3.29 void Detector::outputAveragedMeasuresToCsvFile( std::ofstream & *fout* ) [protected]**

Output the averaged measures to a csv file.

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>fout</i> | Output file stream |
|-------------|--------------------|

Definition at line 222 of file Detector.cpp.

References avgClusterednessDegree, avgDensity, OUTPUT\_CLUSTEREDNESS, and OUTPUT\_DENSITY.

Referenced by outputResultsToCsvFile().

**7.59.3.30 void Detector::outputResults ( const std::string & *outputfilepath* )**

Output the results to the given file.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>outputfilepath</i> | Path to the output file |
|-----------------------|-------------------------|

Definition at line 37 of file Detector.cpp.

References detectMethodCalled, outputFilepath, outputResultsToFile(), and printOutputErrorMessage().

Referenced by main().

**7.59.3.31 void Detector::outputResultsToCsvFile ( ) [protected]**

Output the results to a csv file.

Definition at line 190 of file Detector.cpp.

References CSV\_EXTENSION, ERR\_OUTPUT\_FILE, MS\_throw, and outputFilepath.

Referenced by outputResultsToFile().

**7.59.3.32 void Detector::outputResultsToCsvFile ( std::ofstream & *fout* ) [protected]**

Output the results to a file using the provided output file stream.

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>fout</i> | Output file stream |
|-------------|--------------------|

Definition at line 202 of file Detector.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::fieldNamesToString(), outputAveragedMeasuresToCsvFile(), and outputSpatialEntitiesToCsvFile().

**7.59.3.33 void Detector::outputResultsToFile ( ) [protected]**

Output the results to file(s)

Definition at line 176 of file Detector.cpp.

References outputResultsToCsvFile(), outputResultsToImage(), outputResultsToXMLFile(), and storeOutputImageOnDisk().

Referenced by outputResults().

**7.59.3.34 virtual void multiscale::analysis::Detector::outputResultsToImage ( ) [protected], [pure virtual]**

Output the results to an image.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::SimulationClusterDetector](#).

Referenced by `displayResultsInWindow()`, and `outputResultsToFile()`.

#### 7.59.3.35 void Detector::outputResultsToXMLFile( ) [protected]

Output the results to an xml file.

Definition at line 227 of file `Detector.cpp`.

References `outputFilepath`, and `XML_EXTENSION`.

Referenced by `outputResultsToFile()`.

#### 7.59.3.36 void Detector::outputResultsToXMLFile( const std::string & *filepath* ) [protected]

Output the clusters and averaged measures to an xml file.

##### Parameters

|                 |                  |
|-----------------|------------------|
| <i>filepath</i> | Output file path |
|-----------------|------------------|

Definition at line 231 of file `Detector.cpp`.

References `addAverageMeasuresToPropertyTree()`, `addSpatialEntitiesToPropertyTree()`, `LABEL_COMMENT`, and `LABEL_COMMENT_CONTENTS`.

#### 7.59.3.37 void Detector::outputSpatialEntitiesToCsvFile( std::ofstream & *fout* ) [protected]

Output the pseudo 3D spatial entities to a csv file.

##### Parameters

|             |                    |
|-------------|--------------------|
| <i>fout</i> | Output file stream |
|-------------|--------------------|

Definition at line 214 of file `Detector.cpp`.

References `getCollectionOfSpatialEntityPseudo3D()`.

Referenced by `outputResultsToCsvFile()`.

#### 7.59.3.38 double Detector::polygonAngle( const std::vector< cv::Point > & *polygon*, unsigned int *closestPointIndex* ) [protected]

Compute the angle of the polygon.

Compute the angle determined by the closest point to the origin and the points P1 and P2. These points are obtained from the intersection of the polygon with the line which is orthogonal to the line AB where:

- Point A is the polygon point closest to the origin;
- Point B is the centre point of the bounding rotated rectangle.

##### Parameters

|                          |                                                                                      |
|--------------------------|--------------------------------------------------------------------------------------|
| <i>polygon</i>           | Given polygon                                                                        |
| <i>closestPointIndex</i> | Index of the closest point to the origin from the set of points defining the polygon |

Definition at line 115 of file `Detector.cpp`.

Referenced by `multiscale::analysis::RegionDetector::createRegionFromPolygon()`, and `multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues()`.

7.59.3.39 double Detector::polygonAngle ( const std::vector< cv::Point > & *polygonConvexHull*, const cv::Point & *closestPoint* ) [protected]

Compute the angle of the polygon.

Compute the angle determined by the closest point to the origin and the points P1 and P2. These points are obtained from the intersection of the convex hull with the line AB, determined by points A and B. Points A and B are the middle points of the sides of the rotated rectangle enclosing the polygon that are orthogonal to the line which is the nearest to the closestPoint.

## Parameters

|                                 |                                                                         |
|---------------------------------|-------------------------------------------------------------------------|
| $polygon \leftarrow ConvexHull$ | Convex hull of polygon                                                  |
| $closestPoint$                  | Closest point to the origin from the set of points defining the polygon |

Definition at line 123 of file Detector.cpp.

References `multiscale::Geometry2D::angleBtwPoints()`, `findGoodPointsForAngle()`, and `minAreaRectCentre()`.

**7.59.3.40 void Detector::printOutputErrorMessage( ) [protected]**

Print error message, because the detect method was not called before calling the output method.

Definition at line 327 of file Detector.cpp.

References ERR\_OUTPUT\_WITHOUT\_DETECT.

Referenced by outputResults().

7.59.3.41 virtual void multiscale::analysis::Detector::processImageAndDetect( ) [protected], [pure virtual]

Process the input image and detect objects/entities of interest.

Implemented in `multiscale::analysis::RegionDetector`, and `multiscale::analysis::ClusterDetector`.

Referenced by detectInDebugMode(), and detectInReleaseMode().

7.59.3.42 void Detector::processPressedKeyRequest ( char & pressedKey ) [protected]

Process the request of the user by pressing the key.

## Parameters

`pressedKey` Key pressed by the user, if a key was pressed, or "-1", otherwise

Definition at line 316 of file Detector.cpp.

Referenced by `detectInDebugMode()`.

**7.59.3.43 void Detector::setDetectorSpecificFieldsInitialisationFlag ( bool flag = true ) [protected]**

Set the detector specific fields initialisation flag to true.

Definition at line 60 of file Detector.cpp

References detectorSpecificFieldsInitialised

Referenced by multiscale::analysis::RegionDetector::setAlpha(), multiscale::analysis::RegionDetector::setBeta(), multiscale::analysis::RegionDetector::setBlurKernelSize(), multiscale::analysis::ClusterDetector::setEps(), multiscale::analysis::RegionDetector::setEpsilon(), multiscale::analysis::ClusterDetector::setMinPoints(), multiscale::analysis::RegionDetector::setMorphologicalCloselterations(). multiscale::analysis::RegionDetector::

::setOriginXCoordinate(), multiscale::analysis::RegionDetector::setOriginYCoordinate(), multiscale::analysis::RegionDetector::setRegionAreaThresh(), and multiscale::analysis::RegionDetector::setThresholdValue().

#### 7.59.3.44 void Detector::storeOutputImageOnDisk( ) [protected]

Store the image with the output results on disk.

Definition at line 184 of file Detector.cpp.

References IMG\_EXTENSION, outputFilepath, and outputImage.

Referenced by outputResultsToFile().

### 7.59.4 Member Data Documentation

#### 7.59.4.1 double multiscale::analysis::Detector::avgClusterednessDegree [protected]

For regions: Average degree of clusteredness of all regions

For clusters: Index of clusteredness for all clusters

Definition at line 26 of file Detector.hpp.

Referenced by addAverageMeasuresToPropertyTree(), multiscale::analysis::ClusterDetector::analyseClusters(), multiscale::analysis::ClusterDetector::ClusterDetector(), multiscale::analysis::RegionDetector::computeAverageClusterednessDegree(), Detector(), outputAveragedMeasuresToCsvFile(), multiscale::analysis::RegionDetector::RegionDetector(), and multiscale::analysis::RegionDetector::sumOfAverageCentroidDistances().

#### 7.59.4.2 double multiscale::analysis::Detector::avgDensity [protected]

For regions: Average density of all regions

For clusters: Average pile up degree of all clusters

Definition at line 31 of file Detector.hpp.

Referenced by addAverageMeasuresToPropertyTree(), multiscale::analysis::ClusterDetector::analyseClusters(), multiscale::analysis::ClusterDetector::ClusterDetector(), multiscale::analysis::RegionDetector::computeAverageDensity(), Detector(), outputAveragedMeasuresToCsvFile(), and multiscale::analysis::RegionDetector::RegionDetector().

#### 7.59.4.3 const std::string Detector::CSV\_EXTENSION = ".out" [static], [protected]

Definition at line 296 of file Detector.hpp.

Referenced by outputResultsToCsvFile().

#### 7.59.4.4 bool multiscale::analysis::Detector::debugMode [protected]

Flag for indicating if debug mode is set

Definition at line 39 of file Detector.hpp.

Referenced by detect(), and Detector().

#### 7.59.4.5 bool multiscale::analysis::Detector::detectMethodCalled [protected]

Flag for indicating if the detect method was called

Definition at line 43 of file Detector.hpp.

Referenced by detect(), Detector(), and outputResults().

**7.59.4.6 bool multiscale::analysis::Detector::detectorSpecificFieldsInitialised [protected]**

Flag for indicating if the parameters were

Definition at line 44 of file Detector.hpp.

Referenced by Detector(), initialiseDetectorSpecificFieldsIfNotSet(), and setDetectorSpecificFieldsInitialisation←Flag().

**7.59.4.7 const std::string Detector::ERR\_INVALID\_IMAGE = "The input image is invalid." [static], [protected]**

Definition at line 294 of file Detector.hpp.

Referenced by detect().

**7.59.4.8 const std::string Detector::ERR\_OUTPUT\_FILE = "Unable to create output file." [static], [protected]**

Definition at line 293 of file Detector.hpp.

Referenced by outputResultsToCsvFile().

**7.59.4.9 const std::string Detector::ERR\_OUTPUT\_WITHOUT\_DETECT = "Unable to output results if the detect method was not called previously." [static], [protected]**

Definition at line 292 of file Detector.hpp.

Referenced by printOutputErrorMessage().

**7.59.4.10 cv::Mat multiscale::analysis::Detector::image [protected]**

Input image

Definition at line 37 of file Detector.hpp.

Referenced by multiscale::analysis::RegionDetector::changeContrastAndBrightness(), multiscale::analysis::←SimulationClusterDetector::computePileUpDegreeAtPosition(), detect(), findGoodPointsForAngle(), multiscale::analysis::←SimulationClusterDetector::initialiseDetectorSpecificImageDependentFields(), initialiseImageOrigin(), multiscale::analysis::←SimulationClusterDetector::initialiseThresholdedImage(), multiscale::analysis::←SimulationClusterDetector::outputResultsToImage(), multiscale::analysis::RegionDetector::outputResultsToImage(), multiscale::analysis::←RegionDetector::regionDensity(), and ~Detector().

**7.59.4.11 const std::string Detector::IMG\_EXTENSION = ".png" [static], [protected]**

Definition at line 297 of file Detector.hpp.

Referenced by storeOutputImageOnDisk().

**7.59.4.12 const int Detector::KEY\_ESC = 27 [static], [protected]**

Definition at line 302 of file Detector.hpp.

Referenced by detectInDebugMode().

7.59.4.13 `const int Detector::KEY_SAVE = 115 [static], [protected]`

Definition at line 303 of file Detector.hpp.

7.59.4.14 `const std::string Detector::LABEL_ATTRIBUTE = "<xmllattr>" [static], [protected]`

Definition at line 305 of file Detector.hpp.

Referenced by addSpatialEntityTypeToPropertyTree().

7.59.4.15 `const std::string Detector::LABEL_AVG_CLUSTEREDNESS = "avgClusteredness" [static], [protected]`

Definition at line 329 of file Detector.hpp.

Referenced by addAverageMeasuresToPropertyTree().

7.59.4.16 `const std::string Detector::LABEL_AVG_DENSITY = "avgDensity" [static], [protected]`

Definition at line 330 of file Detector.hpp.

Referenced by addAverageMeasuresToPropertyTree().

7.59.4.17 `const std::string Detector::LABEL_COMMENT = "<xmlcomment>" [static], [protected]`

Definition at line 306 of file Detector.hpp.

Referenced by outputResultsToXMLFile().

7.59.4.18 `const std::string Detector::LABEL_COMMENT_CONTENTS = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library." [static], [protected]`

Definition at line 308 of file Detector.hpp.

Referenced by outputResultsToXMLFile().

7.59.4.19 `const std::string Detector::LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE = "experiment.timepoint.numericStateVariable" [static], [protected]`

Definition at line 310 of file Detector.hpp.

Referenced by addNumericStateVariableToPropertyTree().

7.59.4.20 `const std::string Detector::LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_NAME = "name" [static], [protected]`

Definition at line 313 of file Detector.hpp.

Referenced by addNumericStateVariableToPropertyTree().

7.59.4.21 `const std::string Detector::LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_VALUE = "value" [static], [protected]`

Definition at line 314 of file Detector.hpp.

Referenced by addNumericStateVariableToPropertyTree().

```
7.59.4.22 const std::string Detector::LABEL_EXPERIMENT_TIMEPOINT_SPATIAL_ENTITY =
 "experiment.timepoint.spatialEntity" [static], [protected]
```

Definition at line 311 of file Detector.hpp.

Referenced by addSpatialEntitiesToPropertyTree().

```
7.59.4.23 const std::string Detector::LABEL_SPATIAL_ENTITY_ANGLE = "angle" [static], [protected]
```

Definition at line 322 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.24 const std::string Detector::LABEL_SPATIAL_ENTITY_AREA = "area" [static], [protected]
```

Definition at line 319 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.25 const std::string Detector::LABEL_SPATIAL_ENTITY_CENTROID_X = "centroidX" [static], [protected]
```

Definition at line 326 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.26 const std::string Detector::LABEL_SPATIAL_ENTITY_CENTROID_Y = "centroidY" [static], [protected]
```

Definition at line 327 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.27 const std::string Detector::LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE = "circleMeasure" [static],
 [protected]
```

Definition at line 325 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.28 const std::string Detector::LABEL_SPATIAL_ENTITY_CLUSTEREDNESS = "clusteredness" [static],
 [protected]
```

Definition at line 317 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.29 const std::string Detector::LABEL_SPATIAL_ENTITY_DENSITY = "density" [static], [protected]
```

Definition at line 318 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

```
7.59.4.30 const std::string Detector::LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN = "distanceFromOrigin"
 [static], [protected]
```

Definition at line 321 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

7.59.4.31 `const std::string Detector::LABEL_SPATIAL_ENTITY_PERIMETER = "perimeter" [static], [protected]`

Definition at line 320 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

7.59.4.32 `const std::string Detector::LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE = "rectangleMeasure" [static], [protected]`

Definition at line 324 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

7.59.4.33 `const std::string Detector::LABEL_SPATIAL_ENTITY_SPATIAL_TYPE = "spatialType" [static], [protected]`

Definition at line 316 of file Detector.hpp.

Referenced by addSpatialEntityTypeToPropertyTree().

7.59.4.34 `const std::string Detector::LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE = "triangleMeasure" [static], [protected]`

Definition at line 323 of file Detector.hpp.

Referenced by addSpatialEntityPropertiesToTree().

7.59.4.35 `cv::Point multiscale::analysis::Detector::origin [protected]`

The point representing the origin

Definition at line 46 of file Detector.hpp.

Referenced by multiscale::analysis::RegionDetector::createRegionFromPolygon(), multiscale::analysis::RegionDetector::getOriginXCoordinate(), multiscale::analysis::RegionDetector::getOriginYCoordinate(), initialisImageOrigin(), multiscale::analysis::RegionDetector::setOriginXCoordinate(), multiscale::analysis::RegionDetector::setOriginYCoordinate(), and multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

7.59.4.36 `const std::string Detector::OUTPUT_CLUSTEREDNESS = "Average clusteredness degree: " [static], [protected]`

Definition at line 289 of file Detector.hpp.

Referenced by outputAveragedMeasuresToCsvFile().

7.59.4.37 `const std::string Detector::OUTPUT_DENSITY = "Average density: " [static], [protected]`

Definition at line 290 of file Detector.hpp.

Referenced by outputAveragedMeasuresToCsvFile().

7.59.4.38 `std::string multiscale::analysis::Detector::outputFilepath [protected]`

Path of the output file

Definition at line 38 of file Detector.hpp.

Referenced by `outputResults()`, `outputResultsToCsvFile()`, `outputResultsToXMLFile()`, and `storeOutputImageOnDisk()`.

#### 7.59.4.39 `cv::Mat multiscale::analysis::Detector::outputImage` [protected]

Image for displaying the results

Definition at line 41 of file `Detector.hpp`.

Referenced by `displayResultsInWindow()`, `multiscale::analysis::RegionDetector::outputRegionToImage()`, `multiscale::analysis::SimulationClusterDetector::outputResultsToImage()`, `multiscale::analysis::RegionDetector::outputResultsToImage()`, `storeOutputImageOnDisk()`, and `~Detector()`.

#### 7.59.4.40 `const std::string Detector::WIN_OUTPUT_IMAGE = "Output image"` [static], [protected]

Definition at line 300 of file `Detector.hpp`.

Referenced by `multiscale::analysis::ClusterDetector::createDetectorSpecificTrackbars()`, `multiscale::analysis::RegionDetector::createDetectorSpecificTrackbars()`, `createTrackbarsWindow()`, and `displayResultsInWindow()`.

#### 7.59.4.41 `const std::string Detector::XML_EXTENSION = ".xml"` [static], [protected]

Definition at line 298 of file `Detector.hpp`.

Referenced by `outputResultsToXMLFile()`.

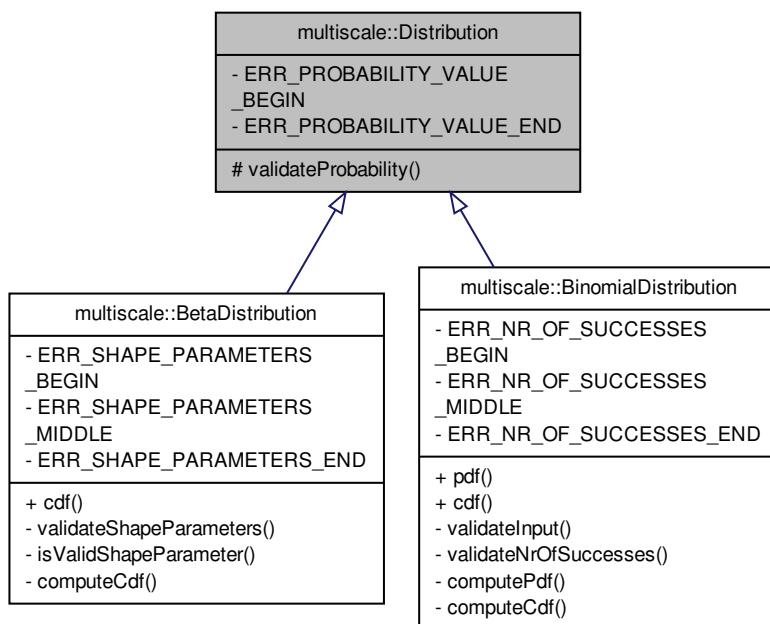
The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[Detector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[Detector.cpp](#)

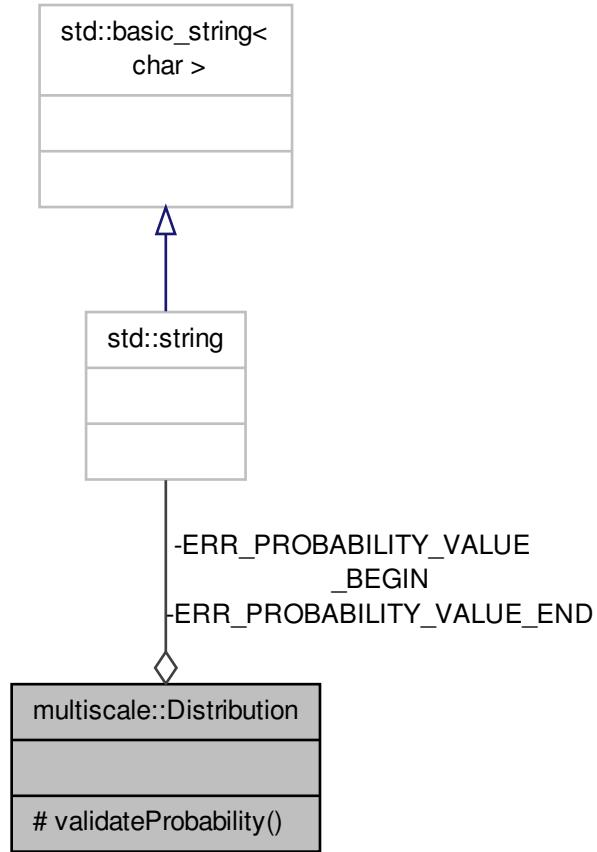
## 7.60 multiscale::Distribution Class Reference

```
#include <Distribution.hpp>
```

Inheritance diagram for multiscale::Distribution:



## Collaboration diagram for multiscale::Distribution:



## Static Protected Member Functions

- static void validateProbability (double probability)

*Check if the value of the probability is valid.*

## Static Private Attributes

- static const std::string ERR\_PROBABILITY\_VALUE\_BEGIN = "The given probability value ("
  - static const std::string ERR\_PROBABILITY\_VALUE\_END = ") should be between 0 and 1."

### **7.60.1 Detailed Description**

Definition at line 10 of file Distribution.hpp.

## 7.60.2 Member Function Documentation

7.60.2.1 void Distribution::validateProbability( double *probability* ) [static], [protected]

Check if the value of the probability is valid.

**Parameters**

|                    |                              |
|--------------------|------------------------------|
| <i>probability</i> | The value of the probability |
|--------------------|------------------------------|

Definition at line 8 of file Distribution.cpp.

References ERR\_PROBABILITY\_VALUE\_BEGIN, ERR\_PROBABILITY\_VALUE\_END, MS\_throw, and multiscale::StringManipulator::toString().

Referenced by multiscale::BetaDistribution::cdf(), and multiscale::BinomialDistribution::validateInput().

### 7.60.3 Member Data Documentation

7.60.3.1 `const std::string Distribution::ERR_PROBABILITY_VALUE_BEGIN = "The given probability value (" [static], [private]`

Definition at line 23 of file Distribution.hpp.

Referenced by validateProbability().

7.60.3.2 `const std::string Distribution::ERR_PROBABILITY_VALUE_END = ") should be between 0 and 1." [static], [private]`

Definition at line 24 of file Distribution.hpp.

Referenced by validateProbability().

The documentation for this class was generated from the following files:

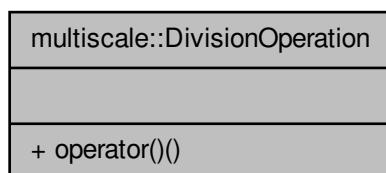
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/[Distribution.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/[Distribution.cpp](#)

## 7.61 multiscale::DivisionOperation Class Reference

Functor representing a division operation.

```
#include <Numeric.hpp>
```

Collaboration diagram for multiscale::DivisionOperation:



### Public Member Functions

- template<typename Operand >  
Operand [operator\(\)](#) (Operand operand1, Operand operand2) const

Divide the two operands.

### 7.61.1 Detailed Description

Functor representing a division operation.

Definition at line 32 of file Numeric.hpp.

### 7.61.2 Member Function Documentation

7.61.2.1 template<typename Operand > Operand multiscale::DivisionOperation::operator() ( Operand *operand1*, Operand *operand2* ) const [inline]

Divide the two operands.

#### Parameters

|                 |                    |
|-----------------|--------------------|
| <i>operand1</i> | The first operand  |
| <i>operand2</i> | The second operand |

Definition at line 42 of file Numeric.hpp.

The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[Numeric.hpp](#)

## 7.62 multiscaletest::EmptyTraceTest Class Reference

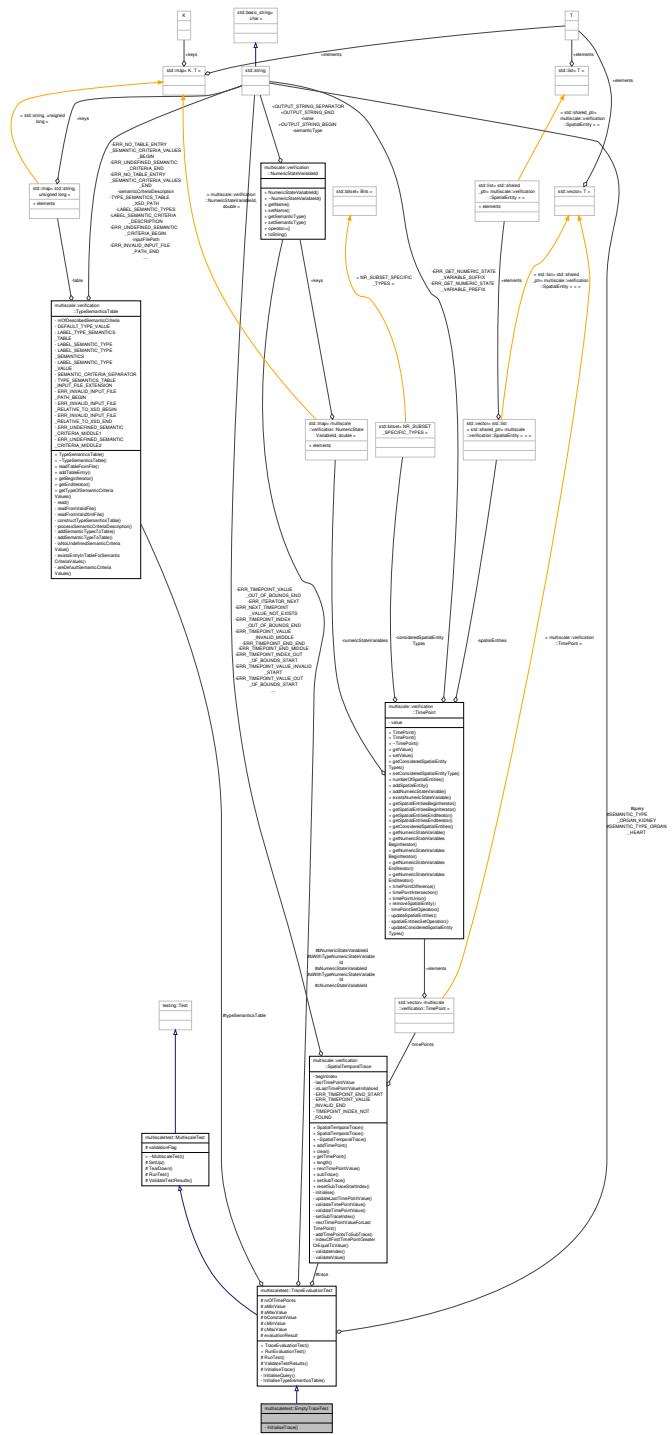
Class for testing evaluation of empty traces.

```
#include <EmptyTraceTest.hpp>
```

Inheritance diagram for multiscaletest::EmptyTraceTest:



## Collaboration diagram for multiscaletest::EmptyTraceTest:



## Private Member Functions

- virtual void **InitialiseTrace** () override  
*Initialise the trace.*

### **Additional Inherited Members**

### 7.62.1 Detailed Description

Class for testing evaluation of empty traces.

Definition at line 22 of file `EmptyTraceTest.hpp`.

### 7.62.2 Member Function Documentation

#### 7.62.2.1 `void multiscaletest::EmptyTraceTest::InitialiseTrace( ) [override], [private], [virtual]`

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 31 of file `EmptyTraceTest.hpp`.

The documentation for this class was generated from the following file:

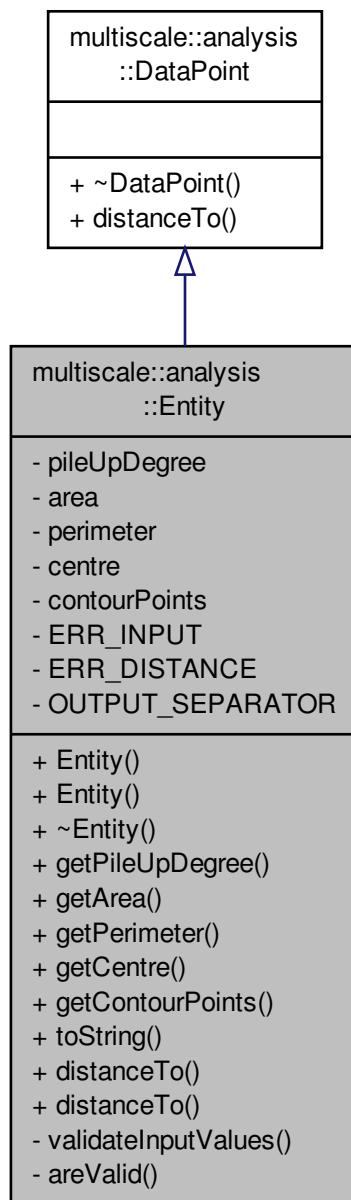
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[EmptyTraceTest.hpp](#)

## 7.63 multiscale::analysis::Entity Class Reference

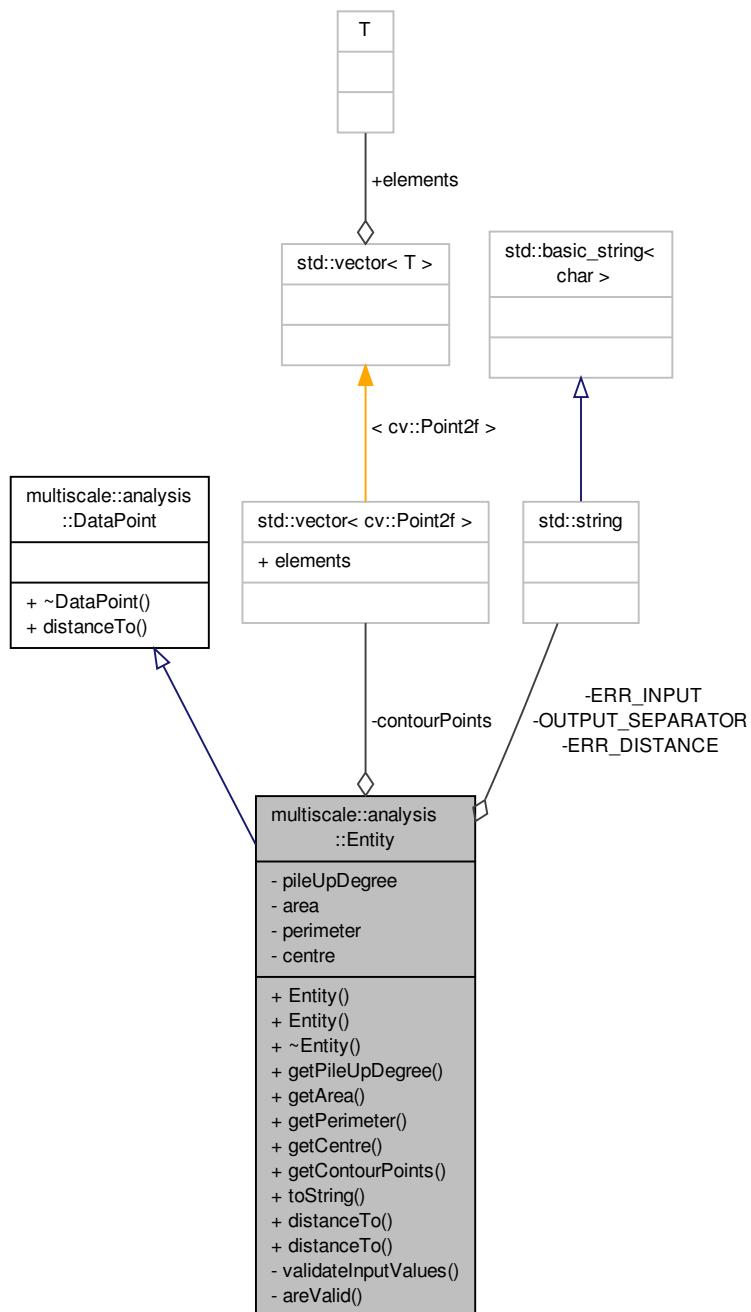
Class for representing an entity in an image (e.g. cell, organism etc.)

```
#include <Entity.hpp>
```

Inheritance diagram for multiscale::analysis::Entity:



Collaboration diagram for multiscale::analysis::Entity:



## Public Member Functions

- `Entity` (unsigned int `pileUpDegree`, double `area`, double `perimeter`, const `cv::Point2f &centre`, const `std::vector<cv::Point2f > &contourPoints`)
- `Entity` (const `Entity` &`entity`)
- `~Entity ()`
- unsigned int `getPileUpDegree () const`

- `double getArea () const`  
*Get the degree of pile up.*
- `double getPerimeter () const`  
*Get the area.*
- `cv::Point2f getCentre () const`  
*Get the perimeter.*
- `cv::Point2f getCentre () const`  
*Get the point defining the centre of the entity.*
- `std::vector< cv::Point2f > getContourPoints () const`  
*Get the set of points defining the contour of the entity.*
- `std::string toString ()`  
*Get a std::string representation of all the field values.*
- `double distanceTo (std::shared_ptr< DataPoint > point) override`  
*Get the distance between this entity and another one.*
- `double distanceTo (const Entity &entity)`  
*Get the distance between this entity and another one.*

## Private Member Functions

- `void validateInputValues (unsigned int pileUpDegree, double area, double perimeter, const cv::Point2f &centre, const std::vector< cv::Point2f > &contourPoints)`
  - `bool isValid (unsigned int pileUpDegree, double area, double perimeter, const cv::Point2f &centre, const std::vector< cv::Point2f > &contourPoints)`
- Check if the provided degree of pile up, area, centre and contour points are valid.*

## Private Attributes

- `unsigned int pileUpDegree`
- `double area`
- `double perimeter`
- `cv::Point2f centre`
- `std::vector< cv::Point2f > contourPoints`

## Static Private Attributes

- `static const std::string ERR_INPUT = "Invalid input parameters were provided to the constructor."`
- `static const std::string ERR_DISTANCE = "The distance to an object of a different type cannot be computed."`
- `static const std::string OUTPUT_SEPARATOR = ","`

### 7.63.1 Detailed Description

Class for representing an entity in an image (e.g. cell, organism etc.)

Definition at line 16 of file Entity.hpp.

### 7.63.2 Constructor & Destructor Documentation

#### 7.63.2.1 Entity::Entity ( `unsigned int pileUpDegree, double area, double perimeter, const cv::Point2f & centre, const std::vector< cv::Point2f > & contourPoints` )

Definition at line 9 of file Entity.cpp.

References area, centre, contourPoints, perimeter, pileUpDegree, and validateInputValues().

### 7.63.2.2 Entity::Entity ( const Entity & entity )

Definition at line 20 of file Entity.cpp.

References area, centre, contourPoints, perimeter, pileUpDegree, and validateInputValues().

### 7.63.2.3 Entity::~Entity ( )

Definition at line 30 of file Entity.cpp.

## 7.63.3 Member Function Documentation

### 7.63.3.1 bool Entity::isValid ( unsigned int pileUpDegree, double area, double perimeter, const cv::Point2f & centre, const std::vector<cv::Point2f> & contourPoints ) [private]

Check if the provided degree of pile up, area, centre and contour points are valid.

#### Parameters

|                      |                                           |
|----------------------|-------------------------------------------|
| <i>pileUpDegree</i>  | Degree of pile up                         |
| <i>area</i>          | Area                                      |
| <i>perimeter</i>     | Perimeter                                 |
| <i>centre</i>        | Centre of the entity                      |
| <i>contourPoints</i> | Points defining the contour of the entity |

Definition at line 75 of file Entity.cpp.

References multiscale::Numeric::greaterOrEqual().

Referenced by validateInputValues().

### 7.63.3.2 double Entity::distanceTo ( std::shared\_ptr<DataPoint> point ) [override], [virtual]

Get the distance between this entity and another one.

Implements [multiscale::analysis::DataPoint](#).

Definition at line 58 of file Entity.cpp.

References centre, and multiscale::Geometry2D::distanceBtwPoints().

### 7.63.3.3 double Entity::distanceTo ( const Entity & entity )

Get the distance between this entity and another one.

Definition at line 64 of file Entity.cpp.

References centre, and multiscale::Geometry2D::distanceBtwPoints().

### 7.63.3.4 double Entity::getArea ( ) const

Get the area.

Definition at line 36 of file Entity.cpp.

References area.

### 7.63.3.5 cv::Point2f Entity::getCentre ( ) const

Get the point defining the centre of the entity.

Definition at line 44 of file Entity.cpp.

References centre.

#### 7.63.3.6 `std::vector< cv::Point2f > Entity::getContourPoints( ) const`

Get the set of points defining the contour of the entity.

Definition at line 48 of file Entity.cpp.

References contourPoints.

#### 7.63.3.7 `double Entity::getPerimeter( ) const`

Get the perimeter.

Definition at line 40 of file Entity.cpp.

References perimeter.

#### 7.63.3.8 `unsigned int Entity::getPileUpDegree( ) const`

Get the degree of pile up.

Definition at line 32 of file Entity.cpp.

References pileUpDegree.

#### 7.63.3.9 `std::string Entity::toString( )`

Get a std::string representation of all the field values.

Definition at line 52 of file Entity.cpp.

References centre, OUTPUT\_SEPARATOR, and pileUpDegree.

#### 7.63.3.10 `void Entity::validateInputValues( unsigned int pileUpDegree, double area, double perimeter, const cv::Point2f & centre, const std::vector< cv::Point2f > & contourPoints ) [private]`

##### Parameters

|                            |                                           |
|----------------------------|-------------------------------------------|
| <code>pileUpDegree</code>  | Degree of pile up                         |
| <code>area</code>          | Area                                      |
| <code>perimeter</code>     | Perimeter                                 |
| <code>centre</code>        | Centre of the entity                      |
| <code>contourPoints</code> | Points defining the contour of the entity |

Definition at line 68 of file Entity.cpp.

References isValid(), ERR\_INPUT, and MS\_throw.

Referenced by Entity().

## 7.63.4 Member Data Documentation

#### 7.63.4.1 `double multiscale::analysis::Entity::area [private]`

Area of the entity

Definition at line 23 of file Entity.hpp.

Referenced by Entity(), and getArea().

#### 7.63.4.2 `cv::Point2f multiscale::analysis::Entity::centre` [private]

Point defining the centre of the entity

Definition at line 28 of file Entity.hpp.

Referenced by distanceTo(), Entity(), getCentre(), and toString().

#### 7.63.4.3 `std::vector<cv::Point2f> multiscale::analysis::Entity::contourPoints` [private]

Set of points defining the contour of the entity

Definition at line 30 of file Entity.hpp.

Referenced by Entity(), and getContourPoints().

#### 7.63.4.4 `const std::string Entity::ERR_DISTANCE = "The distance to an object of a different type cannot be computed."` [static], [private]

Definition at line 93 of file Entity.hpp.

#### 7.63.4.5 `const std::string Entity::ERR_INPUT = "Invalid input parameters were provided to the constructor."` [static], [private]

Definition at line 92 of file Entity.hpp.

Referenced by validateInputValues().

#### 7.63.4.6 `const std::string Entity::OUTPUT_SEPARATOR = ","` [static], [private]

Definition at line 95 of file Entity.hpp.

Referenced by toString().

#### 7.63.4.7 `double multiscale::analysis::Entity::perimeter` [private]

Perimeter of the entity

Definition at line 25 of file Entity.hpp.

Referenced by Entity(), and getPerimeter().

#### 7.63.4.8 `unsigned int multiscale::analysis::Entity::pileUpDegree` [private]

Degree of pile up (relevant only if entities can pile up onto each other)

Definition at line 21 of file Entity.hpp.

Referenced by Entity(), getPileUpDegree(), and toString().

The documentation for this class was generated from the following files:

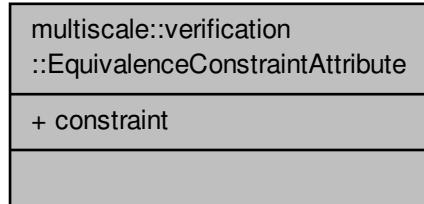
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[Entity.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[Entity.cpp](#)

## 7.64 multiscale::verification::EquivalenceConstraintAttribute Class Reference

Class for representing an "equivalence" constraint attribute.

```
#include <EquivalenceConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::EquivalenceConstraintAttribute:



### Public Attributes

- [ConstraintAttributeType constraint](#)

#### 7.64.1 Detailed Description

Class for representing an "equivalence" constraint attribute.

Definition at line 14 of file EquivalenceConstraintAttribute.hpp.

#### 7.64.2 Member Data Documentation

##### 7.64.2.1 ConstraintAttributeType multiscale::verification::EquivalenceConstraintAttribute::constraint

The constraint following the "equivalence" operator

Definition at line 18 of file EquivalenceConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

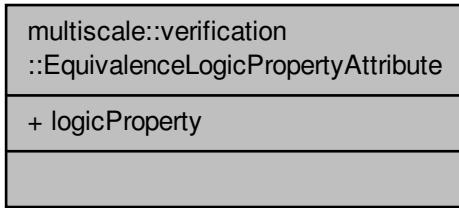
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/EquivalenceConstraintAttribute.hpp](#)

## 7.65 multiscale::verification::EquivalenceLogicPropertyAttribute Class Reference

Class for representing an "equivalence" logic property attribute.

```
#include <EquivalenceLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::EquivalenceLogicPropertyAttribute:



## Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.65.1 Detailed Description

Class for representing an "equivalence" logic property attribute.

Definition at line 14 of file EquivalenceLogicPropertyAttribute.hpp.

### 7.65.2 Member Data Documentation

#### 7.65.2.1 LogicPropertyAttributeType multiscale::verification::EquivalenceLogicPropertyAttribute::logicProperty

The logic property following the "equivalence" operator

Definition at line 18 of file EquivalenceLogicPropertyAttribute.hpp.

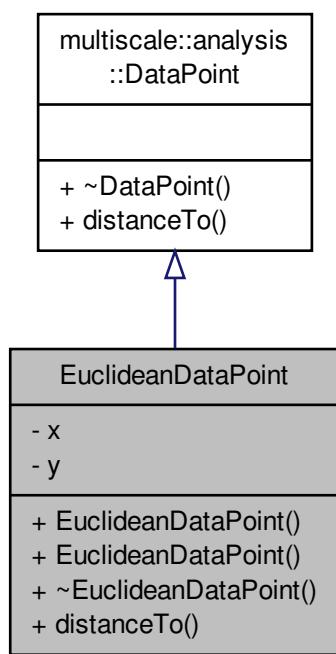
Referenced by multiscale::verification::LogicPropertyVisitor::operator()().

The documentation for this class was generated from the following file:

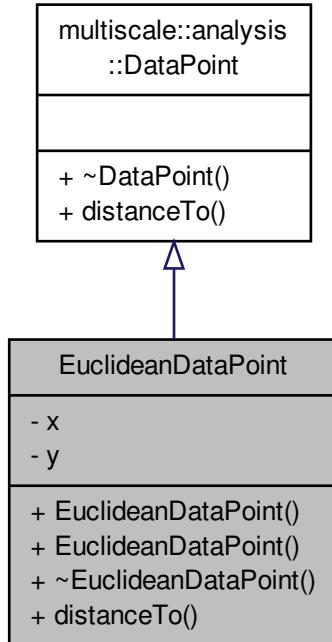
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[EquivalenceLogicPropertyAttribute.hpp](#)

## 7.66 EuclideanDataPoint Class Reference

Inheritance diagram for EuclideanDataPoint:



Collaboration diagram for EuclideanDataPoint:



## Public Member Functions

- `EuclideanDataPoint (double x, double y)`
- `EuclideanDataPoint (const EuclideanDataPoint &point)`
- `~EuclideanDataPoint ()`
- `double distanceTo (std::shared_ptr< DataPoint > point) override`

*Compute the distance between this data point and another one.*

## Private Attributes

- `double x`
- `double y`

### 7.66.1 Detailed Description

Definition at line 14 of file DBSCANTest.cpp.

### 7.66.2 Constructor & Destructor Documentation

#### 7.66.2.1 EuclideanDataPoint::EuclideanDataPoint ( double x, double y ) [inline]

Definition at line 21 of file DBSCANTest.cpp.

7.66.2.2 EuclideanDataPoint::EuclideanDataPoint ( const EuclideanDataPoint & *point* ) [inline]

Definition at line 22 of file DBSCANTest.cpp.

## 7.66.2.3 EuclideanDataPoint::~EuclideanDataPoint ( ) [inline]

Definition at line 23 of file DBSCANTest.cpp.

## 7.66.3 Member Function Documentation

7.66.3.1 double EuclideanDataPoint::distanceTo ( std::shared\_ptr< DataPoint > *point* ) [inline], [override], [virtual]

Compute the distance between this data point and another one.

Parameters

|              |                                              |
|--------------|----------------------------------------------|
| <i>point</i> | Data point to which the distance is measured |
|--------------|----------------------------------------------|

Implements [multiscale::analysis::DataPoint](#).

Definition at line 25 of file DBSCANTest.cpp.

References [multiscale::Geometry2D::distanceBtwPoints\(\)](#).

## 7.66.4 Member Data Documentation

## 7.66.4.1 double EuclideanDataPoint::x [private]

Definition at line 17 of file DBSCANTest.cpp.

## 7.66.4.2 double EuclideanDataPoint::y [private]

Definition at line 18 of file DBSCANTest.cpp.

The documentation for this class was generated from the following file:

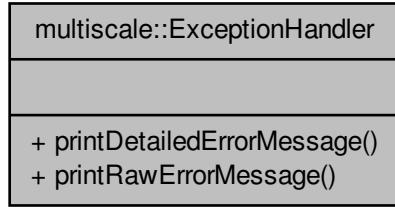
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/[DBSCANTest.cpp](#)

## 7.67 multiscale::ExceptionHandler Class Reference

Exception handler class.

```
#include <ExceptionHandler.hpp>
```

Collaboration diagram for multiscale::ExceptionHandler:



## Static Public Member Functions

- static void [printDetailedErrorMessage](#) (const std::exception &ex)  
*Print the detailed error message.*
- static void [printRawErrorMessage](#) (const MultiscaleException &ex)  
*Print the raw error message.*

### 7.67.1 Detailed Description

Exception handler class.

Definition at line 14 of file `ExceptionHandler.hpp`.

### 7.67.2 Member Function Documentation

**7.67.2.1 static void multiscale::ExceptionHandler::printDetailedErrorMessage ( const std::exception & ex ) [inline], [static]**

Print the detailed error message.

The error message is printed using the `ex.what()` method

#### Parameters

|                 |           |
|-----------------|-----------|
| <code>ex</code> | Exception |
|-----------------|-----------|

Definition at line 23 of file `ExceptionHandler.hpp`.

References `multiscale::ERR_MSG`.

Referenced by `analysePatterns()`, `multiscale::OperatingSystem::executeProgram()`, `main()`, `multiscale::verification::ModelCheckingManager::parseLogicProperty()`, `printParsingResult()`, `readAndPrintInputFile()`, `readAndPrintTypeSemanticsTable()`, `readQueriesFromFile()`, `readValidXmlFilesFromFolder()`, and `validateXmlFile()`.

**7.67.2.2 static void multiscale::ExceptionHandler::printRawErrorMessage ( const MultiscaleException & ex ) [inline], [static]**

Print the raw error message.

The error message is printed using the `ex.rawMessage()` method

**Parameters**

|    |           |
|----|-----------|
| ex | Exception |
|----|-----------|

Definition at line 34 of file ExceptionHandler.hpp.

References multiscale::ERR\_MSG, and multiscale::MultiscaleException::rawMessage().

Referenced by main().

The documentation for this class was generated from the following file:

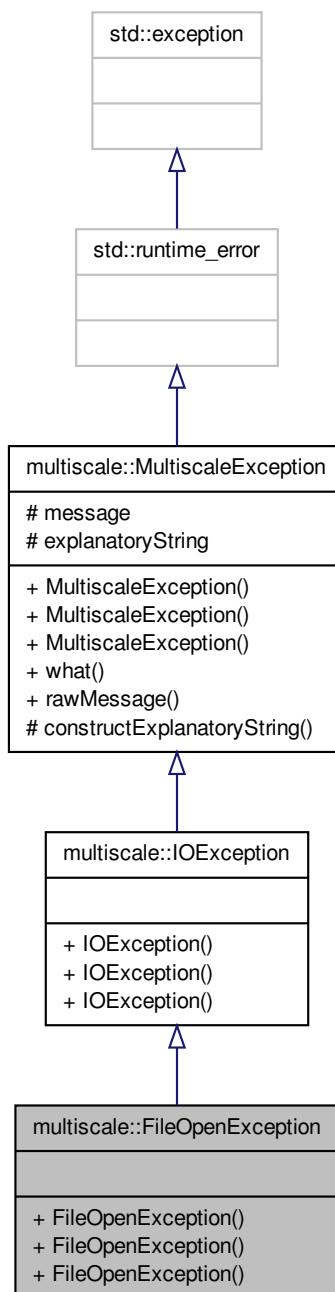
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/[ExceptionHandler.hpp](#)

## 7.68 multiscale::FileOpenException Class Reference

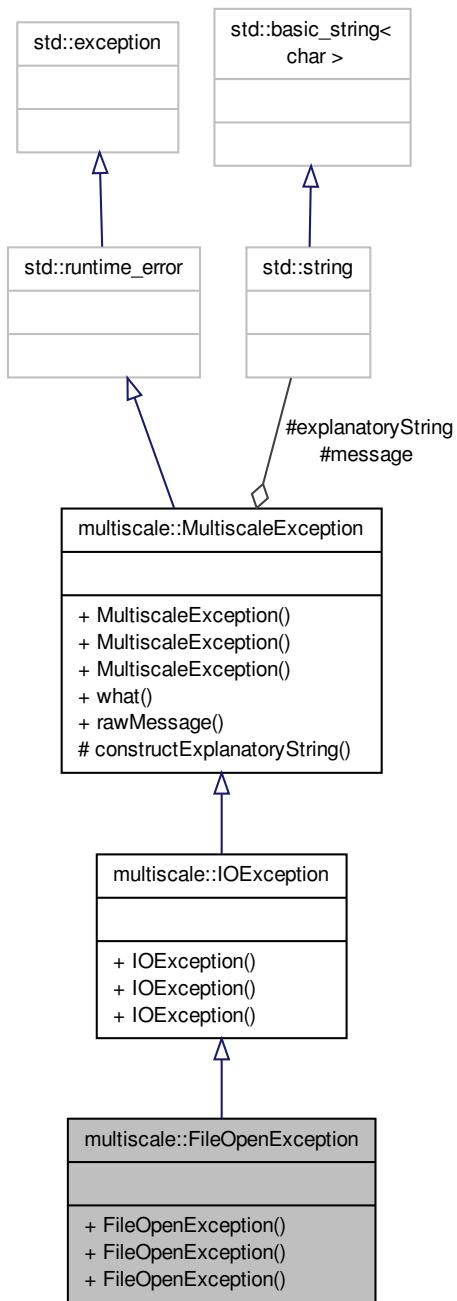
Class for representing exceptions when opening a file.

```
#include <FileOpenException.hpp>
```

Inheritance diagram for multiscale::FileOpenException:



Collaboration diagram for multiscale::FileOpenException:



## Public Member Functions

- [FileOpenException \(\)](#)
- [FileOpenException \(const std::string &file, int line, const std::string &msg\)](#)
- [FileOpenException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.68.1 Detailed Description

Class for representing exceptions when opening a file.

Definition at line 12 of file FileOpenException.hpp.

### 7.68.2 Constructor & Destructor Documentation

#### 7.68.2.1 multiscale::FileOpenException::FileOpenException( ) [inline]

Definition at line 16 of file FileOpenException.hpp.

#### 7.68.2.2 multiscale::FileOpenException::FileOpenException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 18 of file FileOpenException.hpp.

#### 7.68.2.3 multiscale::FileOpenException::FileOpenException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 23 of file FileOpenException.hpp.

The documentation for this class was generated from the following file:

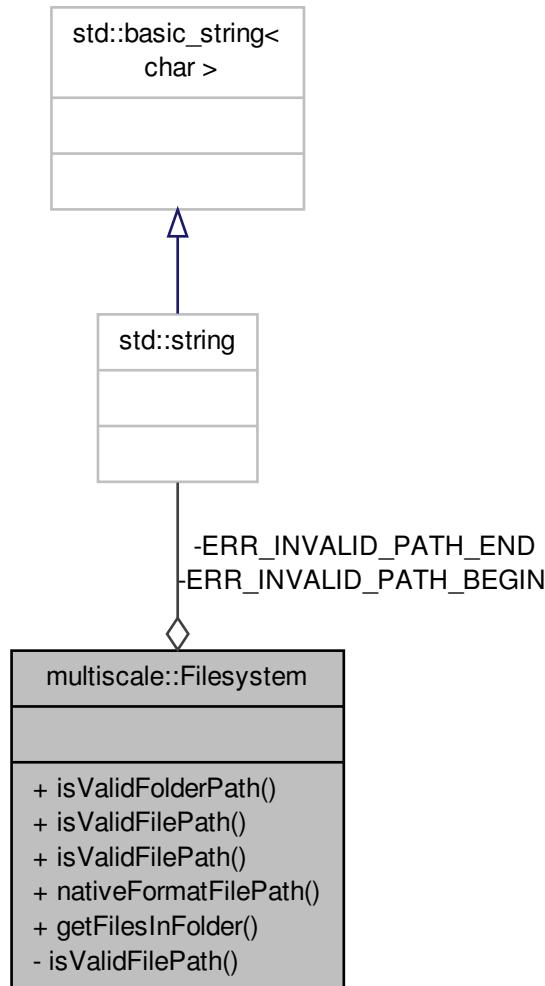
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/FileOpenException.hpp

## 7.69 multiscale::Filesystem Class Reference

Class containing methods for interacting with the filesystem.

```
#include <Filesystem.hpp>
```

Collaboration diagram for multiscale::Filesystem:



## Static Public Member Functions

- static bool [isValidFolderPath](#) (const std::string &path)
 

*Check if the given path is a valid folder path.*
- static bool [isValidFilePath](#) (const std::string &path)
 

*Check if the given path is a valid file path.*
- static bool [isValidFilePath](#) (const std::string &path, const std::string &extension)
 

*Check if the given path is a valid file path and has the given extension.*
- static std::string [nativeFormatFilePath](#) (const std::string &path)
 

*Return the given path as an absolute path in native format.*
- static std::vector< std::string > [getFilesInFolder](#) (const std::string &folderPath, const std::string &extension)
 

*Get the list of files with the given extension in the provided folder.*

## Static Private Member Functions

- static bool `isValidFilePath` (const `fs::path &path)`

*Check if the given path is a valid file path.*

## Static Private Attributes

- static const std::string `ERR_INVALID_PATH_BEGIN` = "The given input file path ("
- static const std::string `ERR_INVALID_PATH_END` = ") does not exist or does not point to a regular file.  
Please change."

### 7.69.1 Detailed Description

Class containing methods for interacting with the filesystem.

This class is using the Boost::Filesystem library.

Definition at line 18 of file Filesystem.hpp.

### 7.69.2 Member Function Documentation

#### 7.69.2.1 `std::vector< std::string > Filesystem::getFilesInFolder ( const std::string & folderPath, const std::string & extension ) [static]`

Get the list of files with the given extension in the provided folder.

Precondition: The provided folder path points to a directory.

##### Parameters

|                         |                        |
|-------------------------|------------------------|
| <code>folderPath</code> | The path to the folder |
| <code>extension</code>  | The given extension    |

Definition at line 46 of file Filesystem.cpp.

References `isValidFolderPath()`.

Referenced by `multiscale::verification::SpatialTemporalDataReader::getFilesInFolder()`.

#### 7.69.2.2 `bool Filesystem::isValidFilePath ( const std::string & path ) [static]`

Check if the given path is a valid file path.

A file path is valid if it points to a regular file.

##### Parameters

|                   |                |
|-------------------|----------------|
| <code>path</code> | The given path |
|-------------------|----------------|

Definition at line 17 of file Filesystem.cpp.

Referenced by `multiscale::OperatingSystem::executeProgramAndVerifyPath()`, `isValidFilePath()`, `multiscale::verification::TemporalDataReader::read()`, `multiscale::verification::TypeSemanticsTable::read()`, `multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile()`, `multiscale::XmlValidator::validateXmlFilepath()`, and `multiscale::XmlValidator::validateXmlSchemaPath()`.

#### 7.69.2.3 `bool Filesystem::isValidFilePath ( const std::string & path, const std::string & extension ) [static]`

Check if the given path is a valid file path and has the given extension.

A file path is valid if it points to a regular file with the given extension.

**Parameters**

|                  |                     |
|------------------|---------------------|
| <i>path</i>      | The given path      |
| <i>extension</i> | The given extension |

Definition at line 23 of file Filesystem.cpp.

References isValidFilePath().

**7.69.2.4 bool Filesystem::isValidFilePath ( const fs::path & *path* ) [static], [private]**

Check if the given path is a valid file path.

**Parameters**

|             |                |
|-------------|----------------|
| <i>path</i> | The given path |
|-------------|----------------|

Definition at line 64 of file Filesystem.cpp.

**7.69.2.5 bool Filesystem::isValidFolderPath ( const std::string & *path* ) [static]**

Check if the given path is a valid folder path.

A folder path is valid if it points to a directory.

**Parameters**

|             |                |
|-------------|----------------|
| <i>path</i> | The given path |
|-------------|----------------|

Definition at line 7 of file Filesystem.cpp.

Referenced by getFilesInFolder(), and multiscale::verification::SpatialTemporalDataReader::validateFolderPath().

**7.69.2.6 std::string Filesystem::nativeFormatFilePath ( const std::string & *path* ) [static]**

Return the given path as an absolute path in native format.

Precondition: The given path points to a regular file.

**Parameters**

|             |                |
|-------------|----------------|
| <i>path</i> | The given path |
|-------------|----------------|

Definition at line 33 of file Filesystem.cpp.

References ERR\_INVALID\_PATH\_BEGIN, ERR\_INVALID\_PATH\_END, and MS\_throw.

**7.69.3 Member Data Documentation****7.69.3.1 const std::string Filesystem::ERR\_INVALID\_PATH\_BEGIN = "The given input file path (" [static], [private]**

Definition at line 75 of file Filesystem.hpp.

Referenced by nativeFormatFilePath().

**7.69.3.2 const std::string Filesystem::ERR\_INVALID\_PATH\_END = ") does not exist or does not point to a regular file. Please change." [static], [private]**

Definition at line 76 of file Filesystem.hpp.

Referenced by nativeFormatFilePath().

The documentation for this class was generated from the following files:

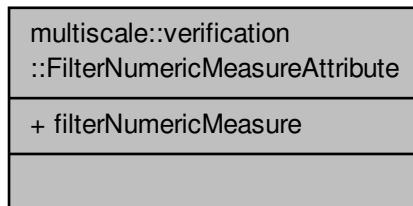
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[Filesystem.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[Filesystem.cpp](#)

## 7.70 multiscale::verification::FilterNumericMeasureAttribute Class Reference

Class for representing a filter numeric measure.

```
#include <FilterNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::FilterNumericMeasureAttribute:



### Public Attributes

- [FilterNumericMeasureAttributeType filterNumericMeasure](#)

#### 7.70.1 Detailed Description

Class for representing a filter numeric measure.

Definition at line 32 of file FilterNumericMeasureAttribute.hpp.

#### 7.70.2 Member Data Documentation

##### 7.70.2.1 [FilterNumericMeasureAttributeType multiscale::verification::FilterNumericMeasureAttribute::filterNumericMeasure](#)

The filter numeric measure

Definition at line 36 of file FilterNumericMeasureAttribute.hpp.

Referenced by [multiscale::verification::FilterNumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

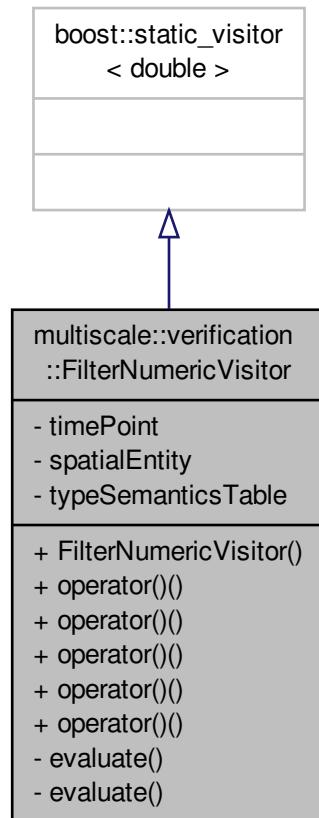
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[FilterNumericMeasureAttribute.hpp](#)

## 7.71 multiscale::verification::FilterNumericVisitor Class Reference

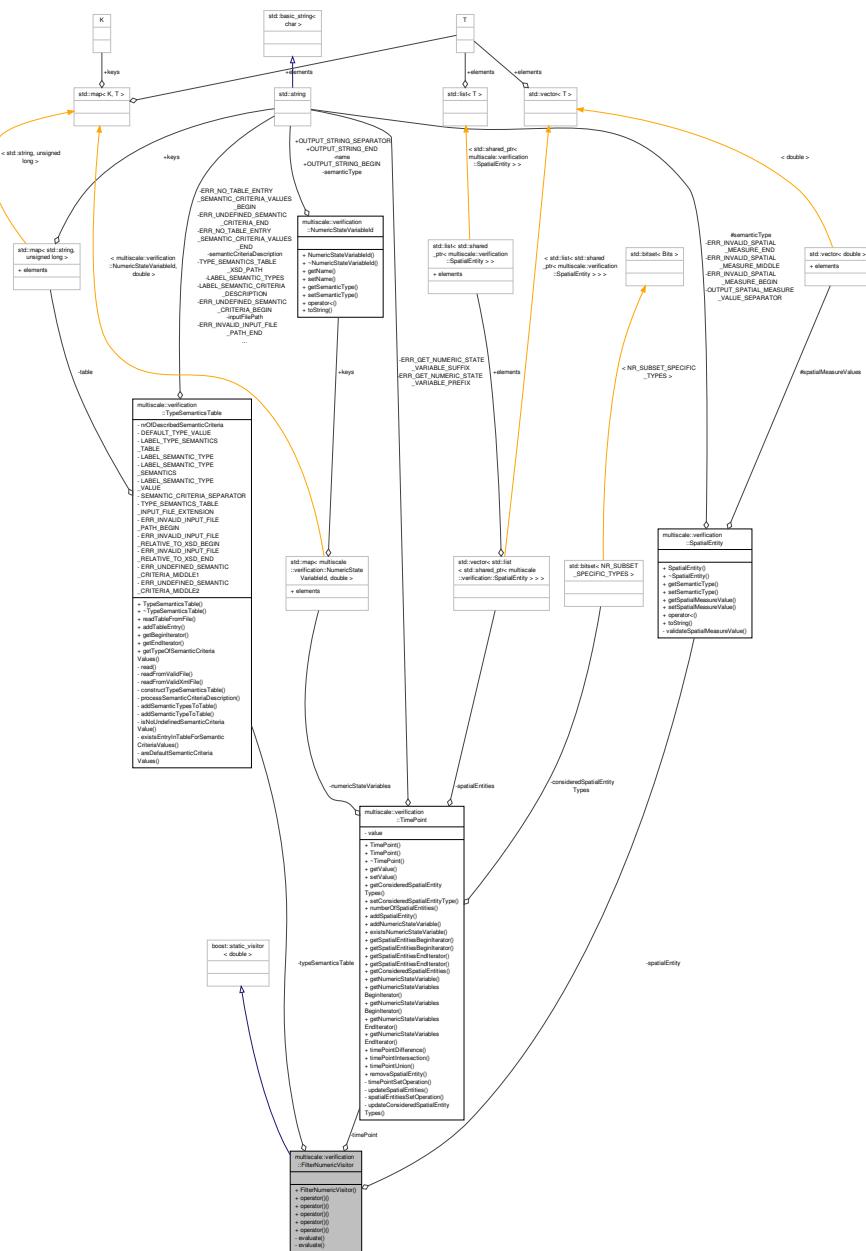
Class for evaluating filter numeric measures.

```
#include <FilterNumericVisitor.hpp>
```

Inheritance diagram for multiscale::verification::FilterNumericVisitor:



## Collaboration diagram for multiscale::verification::FilterNumericVisitor:



## Public Member Functions

- `FilterNumericVisitor` (const `TimePoint` &`timePoint`, const `SpatialEntity` &`spatialEntity`, const `TypeSemanticsTable` &`typeSemanticsTable`)
  - double `operator()` (const `FilterNumericMeasureAttribute` &`filterNumericMeasure`) const  
    *Overloading the "()" operator for the `FilterNumericMeasureAttribute` alternative.*
  - double `operator()` (const `PrimaryNumericMeasureAttribute` &`primaryNumericMeasure`) const  
    *Overloading the "()" operator for the `PrimaryNumericMeasureAttribute` alternative.*
  - double `operator()` (const `SpatialMeasureAttribute` &`spatialMeasure`) const  
    *Overloading the "()" operator for the `SpatialMeasureAttribute` alternative.*
  - double `operator()` (const `UnaryNumericFilterAttribute` &`unaryNumericFilter`) const  
    *Overloading the "()" operator for the `UnaryNumericFilterAttribute` alternative.*

- double `operator()` (const `BinaryNumericFilterAttribute` &binaryNumericFilter) const  
*Overloading the "()" operator for the `BinaryNumericFilterAttribute` alternative.*

## Private Member Functions

- double `evaluate` (const `FilterNumericMeasureAttributeType` &filterNumericMeasure) const  
*Evaluate the given filter numeric measure considering the timePoint and spatialEntity fields.*
- double `evaluate` (const `PrimaryNumericMeasureAttributeType` &primaryNumericMeasure) const  
*Evaluate the given primary numeric measure considering the timePoint field.*

## Private Attributes

- const `TimePoint` & `timePoint`
- const `SpatialEntity` & `spatialEntity`
- const `TypeSemanticsTable` & `typeSemanticsTable`

### 7.71.1 Detailed Description

Class for evaluating filter numeric measures.

Definition at line 16 of file FilterNumericVisitor.hpp.

### 7.71.2 Constructor & Destructor Documentation

7.71.2.1 `multiscale::verification::FilterNumericVisitor::FilterNumericVisitor ( const TimePoint & timePoint, const SpatialEntity & spatialEntity, const TypeSemanticsTable & typeSemanticsTable ) [inline]`

Definition at line 26 of file FilterNumericVisitor.hpp.

Referenced by `evaluate()`.

### 7.71.3 Member Function Documentation

7.71.3.1 `double multiscale::verification::FilterNumericVisitor::evaluate ( const FilterNumericMeasureAttributeType & filterNumericMeasure ) const [inline], [private]`

Evaluate the given filter numeric measure considering the timePoint and spatialEntity fields.

#### Parameters

|                                   |                                  |
|-----------------------------------|----------------------------------|
| <code>filterNumericMeasure</code> | The given filter numeric measure |
|-----------------------------------|----------------------------------|

Definition at line 84 of file FilterNumericVisitor.hpp.

References `FilterNumericVisitor()`.

Referenced by `operator()()`.

7.71.3.2 `double multiscale::verification::FilterNumericVisitor::evaluate ( const PrimaryNumericMeasureAttributeType & primaryNumericMeasure ) const [inline], [private]`

Evaluate the given primary numeric measure considering the timePoint field.

**Parameters**

|                                                    |                                   |
|----------------------------------------------------|-----------------------------------|
| <i>primary</i><br><i>Numeric</i><br><i>Measure</i> | The given primary numeric measure |
|----------------------------------------------------|-----------------------------------|

Definition at line 93 of file FilterNumericVisitor.hpp.

**7.71.3.3 double multiscale::verification::FilterNumericVisitor::operator()** ( **const FilterNumericMeasureAttribute & filterNumericMeasure** ) const [inline]

Overloading the "()" operator for the [FilterNumericMeasureAttribute](#) alternative.

**Parameters**

|                                        |                            |
|----------------------------------------|----------------------------|
| <i>filterNumeric</i><br><i>Measure</i> | The filter numeric measure |
|----------------------------------------|----------------------------|

Definition at line 35 of file FilterNumericVisitor.hpp.

References [evaluate\(\)](#), and [multiscale::verification::FilterNumericMeasureAttribute::filterNumericMeasure](#).

**7.71.3.4 double multiscale::verification::FilterNumericVisitor::operator()** ( **const PrimaryNumericMeasureAttribute & primaryNumericMeasure** ) const [inline]

Overloading the "()" operator for the [PrimaryNumericMeasureAttribute](#) alternative.

**Parameters**

|                                                    |                             |
|----------------------------------------------------|-----------------------------|
| <i>primary</i><br><i>Numeric</i><br><i>Measure</i> | The primary numeric measure |
|----------------------------------------------------|-----------------------------|

Definition at line 43 of file FilterNumericVisitor.hpp.

References [evaluate\(\)](#), and [multiscale::verification::PrimaryNumericMeasureAttribute::primaryNumericMeasure](#).

**7.71.3.5 double multiscale::verification::FilterNumericVisitor::operator()** ( **const SpatialMeasureAttribute & spatialMeasure** ) const [inline]

Overloading the "()" operator for the [SpatialMeasureAttribute](#) alternative.

**Parameters**

|                       |                     |
|-----------------------|---------------------|
| <i>spatialMeasure</i> | The spatial measure |
|-----------------------|---------------------|

Definition at line 51 of file FilterNumericVisitor.hpp.

References [multiscale::verification::SpatialMeasureEvaluator::evaluate\(\)](#), and [multiscale::verification::SpatialMeasureAttribute::spatialMeasureType](#).

**7.71.3.6 double multiscale::verification::FilterNumericVisitor::operator()** ( **const UnaryNumericFilterAttribute & unaryNumericFilter** ) const [inline]

Overloading the "()" operator for the [UnaryNumericFilterAttribute](#) alternative.

**Parameters**

|                           |                          |
|---------------------------|--------------------------|
| <i>unaryNumericFilter</i> | The unary numeric filter |
|---------------------------|--------------------------|

Definition at line 59 of file FilterNumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluate(), multiscale::verification::UnaryNumericFilterAttribute::filterNumericMeasure, multiscale::verification::UnaryNumericFilterAttribute::unaryNumericMeasure, and multiscale::verification::UnaryNumericMeasureAttribute::unaryNumericMeasureType.

#### 7.71.3.7 double multiscale::verification::FilterNumericVisitor::operator() ( const BinaryNumericFilterAttribute & *binaryNumericFilter* ) const [inline]

Overloading the "(" operator for the [BinaryNumericFilterAttribute](#) alternative.

##### Parameters

|                            |                           |
|----------------------------|---------------------------|
| <i>binaryNumericFilter</i> | The binary numeric filter |
|----------------------------|---------------------------|

Definition at line 70 of file FilterNumericVisitor.hpp.

References multiscale::verification::BinaryNumericFilterAttribute::binaryNumericMeasure, multiscale::verification::BinaryNumericMeasureAttribute::binaryNumericMeasureType, multiscale::verification::NumericEvaluator::evaluate(), evaluate(), multiscale::verification::BinaryNumericFilterAttribute::firstFilterNumericMeasure, and multiscale::verification::BinaryNumericFilterAttribute::secondFilterNumericMeasure.

## 7.71.4 Member Data Documentation

### 7.71.4.1 const SpatialEntity& multiscale::verification::FilterNumericVisitor::spatialEntity [private]

The considered spatial entity

Definition at line 21 of file FilterNumericVisitor.hpp.

### 7.71.4.2 const TimePoint& multiscale::verification::FilterNumericVisitor::timePoint [private]

The considered timepoint

Definition at line 20 of file FilterNumericVisitor.hpp.

### 7.71.4.3 const TypeSemanticsTable& multiscale::verification::FilterNumericVisitor::typeSemanticsTable [private]

The considered type semantics table

Definition at line 22 of file FilterNumericVisitor.hpp.

The documentation for this class was generated from the following file:

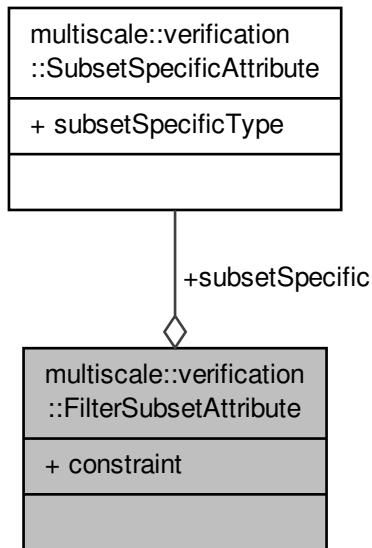
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[FilterNumericVisitor.hpp](#)

## 7.72 multiscale::verification::FilterSubsetAttribute Class Reference

Class for representing a filter subset attribute.

```
#include <FilterSubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::FilterSubsetAttribute:



## Public Attributes

- `SubsetSpecificAttribute subsetSpecific`
- `ConstraintAttributeType constraint`

### 7.72.1 Detailed Description

Class for representing a filter subset attribute.

Definition at line 15 of file `FilterSubsetAttribute.hpp`.

### 7.72.2 Member Data Documentation

#### 7.72.2.1 ConstraintAttributeType multiscale::verification::FilterSubsetAttribute::constraint

The constraint

Definition at line 20 of file `FilterSubsetAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

#### 7.72.2.2 SubsetSpecificAttribute multiscale::verification::FilterSubsetAttribute::subsetSpecific

The specific subset to consider

Definition at line 19 of file `FilterSubsetAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

The documentation for this class was generated from the following file:

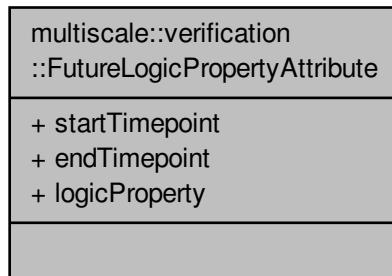
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FilterSubsetAttribute.hpp

## 7.73 multiscale::verification::FutureLogicPropertyAttribute Class Reference

Class for representing a "future" logic property attribute.

```
#include <FutureLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::FutureLogicPropertyAttribute:



### Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [LogicPropertyAttributeType](#) [logicProperty](#)

### 7.73.1 Detailed Description

Class for representing a "future" logic property attribute.

Definition at line 14 of file FutureLogicPropertyAttribute.hpp.

### 7.73.2 Member Data Documentation

#### 7.73.2.1 unsigned long multiscale::verification::FutureLogicPropertyAttribute::endTimepoint

The considered end timepoint

Definition at line 19 of file FutureLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty().

#### 7.73.2.2 LogicPropertyAttributeType multiscale::verification::FutureLogicPropertyAttribute::logicProperty

The logic property following the "future" operator

Definition at line 20 of file FutureLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty().

### 7.73.2.3 `unsigned long multiscale::verification::FutureLogicPropertyAttribute::startTimestep`

The considered start timestep

Definition at line 18 of file `FutureLogicPropertyAttribute.hpp`.

Referenced by `multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty()`.

The documentation for this class was generated from the following file:

- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FutureLogicPropertyAttribute.hpp`

## 7.74 `multiscale::Geometry2D Class Reference`

Two-dimensional geometric operations.

```
#include <Geometry2D.hpp>
```

Collaboration diagram for multiscale::Geometry2D:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| multiscale::Geometry2D                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| + PI<br>+ MATRIX_START_INDEX                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| + angleOfLineWrtOxAxis()<br>+ isAngleBetween()<br>+ isOppositeAngleBetween()<br>+ isAngleBetweenNonReflex()<br>+ isOppositeAngleBetweenNonReflex()<br>+ oppositeAngle()<br>+ slopeOfLine()<br>+ distanceBtwPoints()<br>+ distanceBtwPoints()<br>+ distanceFromPointToLine()<br>+ middlePoint()<br>+ orthogonalLineToAnotherLineEdgePoints()<br>+ areOnTheSameSideOfLine()<br>+ lineEquationDeterminedByPoints()<br>+ areIdenticalLines()<br>+ areIdenticalLines()<br>+ lineIntersection()<br>+ lineIntersection()<br>+ lineIntersection()<br>+ lineSegmentIntersection()<br>+ lineCircleIntersection()<br>+ lineSegmentCircleIntersection()<br>+ angleBtwPoints()<br>+ findPointsOnEdge()<br>+ minimumDistancePointIndex()<br>+ areaOfTriangle()<br>+ isPointOnLineSegment()<br>+ areEqualPoints()<br>+ areCollinear()<br>- isPointOnEdge()<br>- isBetweenCoordinates()<br>- translate()<br>- inverseTranslate()<br>- lineCircleTwoIntersectionPoints()<br>- lineCircleOneIntersectionPoint() |

## Static Public Member Functions

- static double [angleOfLineWrtOxAxis](#) (const cv::Point2f &a, const cv::Point2f &b)  
*Get the angle of the line measured from the Ox axis in counterclockwise direction.*
- static bool [isAngleBetween](#) (double angle1, double angle2, double angle3)  
*Check if angle1 lies between angles 2 and 3.*
- static bool [isOppositeAngleBetween](#) (double angle1, double angle2, double angle3)

- static bool **isAngleBetweenNonReflex** (double angle1, double angle2, double angle3)
 

*Check if the opposite of angle1, ((angle1 + 180) % 360), lies between angles 2 and 3.*
- static bool **isOppositeAngleBetweenNonReflex** (double angle1, double angle2, double angle3)
 

*Check if angle1 lies between non reflex angle determined by angles 2 and 3.*
- static double **oppositeAngle** (double angle)
 

*Return the angle opposite to the given angle.*
- static bool **slopeOfLine** (const cv::Point2f &a, const cv::Point2f &b, double &slope)
 

*Compute the slope of the line defined by points "a" and "b".*
- static double **distanceBtwPoints** (const cv::Point2f &a, const cv::Point2f &b)
 

*Compute the distance between two points.*
- static double **distanceBtwPoints** (double x1, double y1, double x2, double y2)
 

*Compute the distance between two points.*
- static double **distanceFromPointToLine** (const cv::Point2f &a, const cv::Point2f &linePointB, const cv::Point2f &linePointC)
 

*Compute the distance from a point "a" to a line specified by two points "B" and "C".*
- static cv::Point2f **middlePoint** (const cv::Point2f &a, const cv::Point2f &b)
 

*Get the point in the middle of the segment determined by points "a" and "b".*
- static void **orthogonalLineToAnotherLineEdgePoints** (const cv::Point &a1, const cv::Point &b1, cv::Point &a2, cv::Point &b2, int nrOfRows, int nrOfCols)
 

*Find the points which are on the edge and on the line orthogonal to the line defined by 2 given points.*
- static bool **areOnTheSameSideOfLine** (const cv::Point2f &p1, const cv::Point2f &p2, const cv::Point2f &a, const cv::Point2f &b)
 

*Check if p1 and p2 are on the same side of the line determined by points a and b.*
- static void **lineEquationDeterminedByPoints** (const cv::Point2f &p, const cv::Point2f &q, double &a, double &b, double &c)
 

*Get the values of "a", "b" and "c" of the line equation  $ax + by + c = 0$ .*
- static bool **areIdenticalLines** (double a1, double b1, double c1, double a2, double b2, double c2)
 

*Check if two lines are identical.*
- static bool **areIdenticalLines** (const cv::Point2f &a1, const cv::Point2f &b1, const cv::Point2f &a2, const cv::Point2f &b2)
 

*Check if two lines are identical.*
- static bool **lineIntersection** (const cv::Point2f &a1, const cv::Point2f &b1, const cv::Point2f &a2, const cv::Point2f &b2, cv::Point2f &intersection)
 

*Determine the intersection point of two lines, if this point exists.*
- static bool **lineIntersection** (const cv::Point &a1, const cv::Point &b1, const cv::Point &a2, const cv::Point &b2, cv::Point &intersection)
 

*Determine the intersection point of two lines, if this point exists.*
- static bool **lineIntersection** (double a1, double b1, double c1, double a2, double b2, double c2, cv::Point2f &intersection)
 

*Determine the intersection point of two lines, if this point exists.*
- static bool **lineSegmentIntersection** (const cv::Point &a1, const cv::Point &b1, const cv::Point &a2, const cv::Point &b2, cv::Point &intersection)
 

*Determine the intersection point of two line segments, if this point exists.*
- static bool **lineCircleIntersection** (cv::Point2f a, cv::Point2f b, const cv::Point2f &circleOrigin, double radius, std::vector<cv::Point2f> &intersectionPoints)
 

*Determine if a line and a circle intersect and return the intersection points if they exist.*
- static bool **lineSegmentCircleIntersection** (const cv::Point2f &a, const cv::Point2f &b, const cv::Point2f &circleOrigin, double radius, std::vector<cv::Point2f> &intersectionPoints)
 

*Determine if a line segment and a circle intersect and return the intersection points if they exist.*
- static double **angleBtwPoints** (const cv::Point2f &a, const cv::Point2f &b, const cv::Point2f &c)
 

*Compute the angle between three points.*

- static std::vector< cv::Point2f > [findPointsOnEdge](#) (const std::vector< cv::Point2f > &points, unsigned int nrOfRows, unsigned int nrOfCols)
 

*Find the subset of points from the given set of points which lie on the edge.*
- static unsigned int [minimumDistancePointIndex](#) (const std::vector< cv::Point > &points, const cv::Point2f &origin)
 

*Get the index of the point which is the closest to the origin.*
- static double [areaOfTriangle](#) (const cv::Point2f &a, const cv::Point2f &b, const cv::Point2f &c)
 

*Compute the area of a triangle defined by three points.*
- static bool [isPointOnLineSegment](#) (const cv::Point2f &point, const cv::Point2f &lineSegmentStart, const cv::Point2f &lineSegmentEnd)
 

*Check if one point lies between two other points.*
- static bool [areEqualPoints](#) (const cv::Point2f &point1, const cv::Point2f &point2)
 

*Check if points point1 and point2 are equal or not.*
- static bool [areCollinear](#) (const cv::Point2f &point1, const cv::Point2f &point2, const cv::Point2f &point3)
 

*Check if the three points are collinear.*

## Static Public Attributes

- static const double [PI](#) = 3.14159265358979323846264338327950288419716939937510
- static const int [MATRIX\\_START\\_INDEX](#) = 1

## Static Private Member Functions

- static bool [isPointOnEdge](#) (const cv::Point2f &p, int nrOfRows, int nrOfCols)
 

*Check if the given point is on the edge.*
- template<typename T, typename U>  
static bool [isBetweenCoordinates](#) (T c, U c1, U c2)
 

*Check if the coordinate c lies between c1 and c2.*
- static void [translate](#) (cv::Point2f &point, const cv::Point2f &translation)
 

*Translate a point by the given values.*
- static void [inverseTranslate](#) (cv::Point2f &point, const cv::Point2f &translation)
 

*Inverse translate a point by the given values.*
- static void [lineCircleTwoIntersectionPoints](#) (const cv::Point2f &circleOrigin, double A, double B, double C, double delta, std::vector< cv::Point2f > &intersectionPoints)
 

*Treat the case when the line and circle intersect in two points.*
- static void [lineCircleOneIntersectionPoint](#) (const cv::Point2f &circleOrigin, double A, double B, double C, double delta, std::vector< cv::Point2f > &intersectionPoints)
 

*Treat the case when the line and circle intersect in one point.*

### 7.74.1 Detailed Description

Two-dimensional geometric operations.

Definition at line 13 of file Geometry2D.hpp.

### 7.74.2 Member Function Documentation

#### 7.74.2.1 double Geometry2D::angleBtwPoints ( const cv::Point2f & a, const cv::Point2f & b, const cv::Point2f & c ) [static]

Compute the angle between three points.

Compute the angle between the lines determined by points A, B and B, C

**Parameters**

|          |               |
|----------|---------------|
| <i>a</i> | cv::Point2f a |
| <i>b</i> | cv::Point2f b |
| <i>c</i> | cv::Point2f c |

Definition at line 338 of file Geometry2D.cpp.

References PI.

Referenced by multiscale::analysis::Detector::polygonAngle().

#### 7.74.2.2 double Geometry2D::angleOfLineWrtOxAxis ( const cv::Point2f & *a*, const cv::Point2f & *b* ) [static]

Get the angle of the line measured from the Ox axis in counterclockwise direction.

The line is specified by points "a" and "b". The value of the angle is expressed in degrees.

**Parameters**

|          |         |
|----------|---------|
| <i>a</i> | Point a |
| <i>b</i> | Point b |

Definition at line 10 of file Geometry2D.cpp.

References PI.

Referenced by multiscale::MinEnclosingTriangleFinder::intersects(), multiscale::MinEnclosingTriangleFinder::intersectsAbove(), and multiscale::MinEnclosingTriangleFinder::intersectsBelow().

#### 7.74.2.3 double Geometry2D::areaOfTriangle ( const cv::Point2f & *a*, const cv::Point2f & *b*, const cv::Point2f & *c* ) [static]

Compute the area of a triangle defined by three points.

The area is computed using the determinant method. An example is presented at <http://demonstrations.wolfram.com/TheAreaOfATriangleUsingADeterminant/> (Last access: 10.07.2013)

**Parameters**

|          |         |
|----------|---------|
| <i>a</i> | Point a |
| <i>b</i> | Point b |
| <i>c</i> | Point c |

Definition at line 384 of file Geometry2D.cpp.

Referenced by multiscale::MinEnclosingTriangleFinder::returnMinEnclosingTriangle(), TEST(), and multiscale::MinEnclosingTriangleFinder::updateMinEnclosingTriangle().

#### 7.74.2.4 bool Geometry2D::areCollinear ( const cv::Point2f & *point1*, const cv::Point2f & *point2*, const cv::Point2f & *point3* ) [static]

Check if the three points are collinear.

**Parameters**

|               |                  |
|---------------|------------------|
| <i>point1</i> | The first point  |
| <i>point2</i> | The second point |
| <i>point3</i> | The third point  |

Definition at line 409 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

7.74.2.5 bool Geometry2D::areEqualPoints ( const cv::Point2f & *point1*, const cv::Point2f & *point2* ) [static]

Check if points *point1* and *point2* are equal or not.

**Parameters**

|               |                 |
|---------------|-----------------|
| <i>point1</i> | One point       |
| <i>point2</i> | The other point |

Definition at line 402 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle(), and lineEquationDeterminedByPoints().

**7.74.2.6 bool Geometry2D::areIdenticalLines ( double *a1*, double *b1*, double *c1*, double *a2*, double *b2*, double *c2* ) [static]**

Check if two lines are identical.

Lines are be specified in the following form:  $A_1x + B_1x = C_1$   $A_2x + B_2x = C_2$

If  $(A_1/A_2) == (B_1/B_2) == (C_1/C_2)$ , then the lines are identical else they are not

**Parameters**

|           |    |
|-----------|----|
| <i>a1</i> | A1 |
| <i>b1</i> | B1 |
| <i>c1</i> | C1 |
| <i>a2</i> | A2 |
| <i>b2</i> | B2 |
| <i>c2</i> | C2 |

Definition at line 178 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::areIdenticalLines().

**7.74.2.7 bool Geometry2D::areIdenticalLines ( const cv::Point2f & *a1*, const cv::Point2f & *b1*, const cv::Point2f & *a2*, const cv::Point2f & *b2* ) [static]**

Check if two lines are identical.

The lines are specified by a pair of points each. If they are identical, then the function returns true, else it returns false.

Lines can be specified in the following form:  $A_1x + B_1x = C_1$   $A_2x + B_2x = C_2$

If  $(A_1/A_2) == (B_1/B_2) == (C_1/C_2)$ , then the lines are identical else they are not

**Parameters**

|           |                                              |
|-----------|----------------------------------------------|
| <i>a1</i> | First point for determining the first line   |
| <i>b1</i> | Second point for determining the first line  |
| <i>a2</i> | First point for determining the second line  |
| <i>b2</i> | Second point for determining the second line |

Definition at line 193 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

**7.74.2.8 bool Geometry2D::areOnTheSameSideOfLine ( const cv::Point2f & *p1*, const cv::Point2f & *p2*, const cv::Point2f & *a*, const cv::Point2f & *b* ) [static]**

Check if *p1* and *p2* are on the same side of the line determined by points *a* and *b*.

**Parameters**

|           |                                   |
|-----------|-----------------------------------|
| <i>p1</i> | Point p1                          |
| <i>p2</i> | Point p2                          |
| <i>a</i>  | First point for determining line  |
| <i>b</i>  | Second point for determining line |

Definition at line 156 of file Geometry2D.cpp.

References lineEquationDeterminedByPoints(), and multiscale::Numeric::sign().

Referenced by multiscale::MinEnclosingTriangleFinder::findVertexCOnSideB(), and multiscale::MinEnclosingTriangleFinder::gamma().

#### 7.74.2.9 double Geometry2D::distanceBtwPoints ( const cv::Point2f & *a*, const cv::Point2f & *b* ) [static]

Compute the distance between two points.

Compute the Euclidean distance between two points

**Parameters**

|          |         |
|----------|---------|
| <i>a</i> | Point a |
| <i>b</i> | Point b |

Definition at line 76 of file Geometry2D.cpp.

Referenced by multiscale::analysis::Silhouette::computeAverageDissimilarityBtwEntityAndCluster(), multiscale::analysis::Silhouette::computeAverageDissimilarityWithinCluster(), multiscale::analysis::RegionDetector::createRegionFromPolygon(), EuclideanDataPoint::distanceTo(), multiscale::analysis::Entity::distanceTo(), isPointOnLineSegment(), minimumDistancePointIndex(), multiscale::analysis::RegionDetector::sumOfAverageCentroidDistances(), and multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

#### 7.74.2.10 double Geometry2D::distanceBtwPoints ( double *x1*, double *y1*, double *x2*, double *y2* ) [static]

Compute the distance between two points.

Compute the Euclidean distance between two points

**Parameters**

|           |                                      |
|-----------|--------------------------------------|
| <i>x1</i> | The x-coordinate of the first point  |
| <i>y1</i> | The y-coordinate of the first point  |
| <i>x2</i> | The x-coordinate of the second point |
| <i>y2</i> | The y-coordinate of the second point |

Definition at line 83 of file Geometry2D.cpp.

#### 7.74.2.11 double Geometry2D::distanceFromPointToLine ( const cv::Point2f & *a*, const cv::Point2f & *linePointB*, const cv::Point2f & *linePointC* ) [static]

Compute the distance from a point "a" to a line specified by two points "B" and "C".

Formula used:

$$\text{distance} = \frac{|(x_c - x_b)(y_b - y_a) - (x_b - x_a)(y_c - y_b)|}{\sqrt{(x_c - x_b)^2 + (y_c - y_b)^2}}$$

Reference: <http://mathworld.wolfram.com/cv::Point-LineDistance2-Dimensional.html>

**Parameters**

|                   |                                               |
|-------------------|-----------------------------------------------|
| <i>a</i>          | The point from which the distance is measures |
| <i>linePointB</i> | One of the points determining the line        |
| <i>linePointC</i> | One of the points determining the line        |

Definition at line 90 of file Geometry2D.cpp.

Referenced by multiscale::MinEnclosingTriangleFinder::height().

**7.74.2.12 std::vector< cv::Point2f > Geometry2D::findPointsOnEdge ( const std::vector< cv::Point2f > & *points*, unsigned int *nrOfRows*, unsigned int *nrOfCols* ) [static]**

Find the subset of points from the given set of points which lie on the edge.

A point "p" is considered to be on the edge if: ((p.x == 1) && (p.y > 1) && (p.y < nrOfCols)) OR ((p.x == nrOfRows) && (p.y > 1) && (p.y < nrOfCols)) OR ((p.y == 1) && (p.x > 1) && (p.x < nrOfRows)) OR ((p.y == nrOfCols) && (p.x > 1) && (p.x < nrOfRows))

**Parameters**

|                 |                       |
|-----------------|-----------------------|
| <i>points</i>   | The set of points     |
| <i>nrOfRows</i> | The number of rows    |
| <i>nrOfCols</i> | The number of columns |

Definition at line 350 of file Geometry2D.cpp.

References isPointOnEdge().

**7.74.2.13 void Geometry2D::inverseTranslate ( cv::Point2f & *point*, const cv::Point2f & *translation* ) [static], [private]**

Inverse translate a point by the given values.

**Parameters**

|                    |                    |
|--------------------|--------------------|
| <i>point</i>       | The point          |
| <i>translation</i> | Translation values |

Definition at line 435 of file Geometry2D.cpp.

Referenced by lineCircleOneIntersectionPoint(), and lineCircleTwoIntersectionPoints().

**7.74.2.14 bool Geometry2D::isAngleBetween ( double *angle1*, double *angle2*, double *angle3* ) [static]**

Check if angle1 lies between angles 2 and 3.

**Parameters**

|               |                                                       |
|---------------|-------------------------------------------------------|
| <i>angle1</i> | The angle which lies between angle2 and angle3 or not |
| <i>angle2</i> | One of the boundary angles                            |
| <i>angle3</i> | The other boundary angle                              |

Definition at line 20 of file Geometry2D.cpp.

Referenced by isAngleBetweenNonReflex(), and isOppositeAngleBetween().

**7.74.2.15 bool Geometry2D::isAngleBetweenNonReflex ( double *angle1*, double *angle2*, double *angle3* ) [static]**

Check if angle1 lies between non reflex angle determined by angles 2 and 3.

**Parameters**

|               |                                                       |
|---------------|-------------------------------------------------------|
| <i>angle1</i> | The angle which lies between angle2 and angle3 or not |
| <i>angle2</i> | One of the boundary angles                            |
| <i>angle3</i> | The other boundary angle                              |

Definition at line 34 of file Geometry2D.cpp.

References isAngleBetween(), and multiscale::Numeric::lessOrEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor(), multiscale::MinEnclosingTriangleFinder::isGammaAngleBetween(), and isOppositeAngleBetweenNonReflex().

#### 7.74.2.16 template<typename T , typename U > bool Geometry2D::isBetweenCoordinates ( T c, U c1, U c2 ) [static], [private]

Check if the coordinate c lies between c1 and c2.

**Parameters**

|           |               |
|-----------|---------------|
| <i>c</i>  | Coordinate c  |
| <i>c1</i> | Coordinate c1 |
| <i>c2</i> | Coordinate c2 |

Definition at line 426 of file Geometry2D.cpp.

#### 7.74.2.17 bool Geometry2D::isOppositeAngleBetween ( double *angle1*, double *angle2*, double *angle3* ) [static]

Check if the opposite of angle1, ((angle1 + 180) % 360), lies between angles 2 and 3.

**Parameters**

|               |                                                                              |
|---------------|------------------------------------------------------------------------------|
| <i>angle1</i> | The angle for which the opposite angle lies between angle2 and angle3 or not |
| <i>angle2</i> | One of the boundary angles                                                   |
| <i>angle3</i> | The other boundary angle                                                     |

Definition at line 28 of file Geometry2D.cpp.

References isAngleBetween(), and oppositeAngle().

#### 7.74.2.18 bool Geometry2D::isOppositeAngleBetweenNonReflex ( double *angle1*, double *angle2*, double *angle3* ) [static]

Check if the opposite of angle1 lies between non reflex angle determined by angles 2 and 3.

Check if the opposite of angle1, ((angle1 + 180) % 360), lies between non reflex angle determined by angles 2 and 3

**Parameters**

|               |                                                       |
|---------------|-------------------------------------------------------|
| <i>angle1</i> | The angle which lies between angle2 and angle3 or not |
| <i>angle2</i> | One of the boundary angles                            |
| <i>angle3</i> | The other boundary angle                              |

Definition at line 52 of file Geometry2D.cpp.

References isAngleBetweenNonReflex(), and oppositeAngle().

Referenced by multiscale::MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor().

---

```
7.74.2.19 bool Geometry2D::isPointOnEdge (const cv::Point2f & p, int nrOfRows, int nrOfCols) [static],
[private]
```

Check if the given point is on the edge.

A point "p" is considered to be on the edge if: ((p.x == 1) && (p.y > 1) && (p.y < nrOfCols)) OR ((p.x == nrOfRows) && (p.y > 1) && (p.y < nrOfCols)) OR ((p.y == 1) && (p.x > 1) && (p.x < nrOfRows)) OR ((p.y == nrOfCols) && (p.x > 1) && (p.x < nrOfRows))

#### Parameters

|                 |                       |
|-----------------|-----------------------|
| <i>p</i>        | The point p           |
| <i>nrOfRows</i> | The number of rows    |
| <i>nrOfCols</i> | The number of columns |

Definition at line 416 of file Geometry2D.cpp.

References MATRIX\_START\_INDEX.

Referenced by findPointsOnEdge(), and orthogonalLineToAnotherLineEdgePoints().

---

```
7.74.2.20 bool Geometry2D::isPointOnLineSegment (const cv::Point2f & point, const cv::Point2f & lineSegmentStart, const
cv::Point2f & lineSegmentEnd) [static]
```

Check if one point lies between two other points.

#### Parameters

|                         |                                               |
|-------------------------|-----------------------------------------------|
| <i>point</i>            | Point lying possibly outside the line segment |
| <i>lineSegmentStart</i> | First point determining the line segment      |
| <i>lineSegmentEnd</i>   | Second point determining the line segment     |

Definition at line 393 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual(), and distanceBtwPoints().

Referenced by multiscaletest::MinEnclosingTriangleFinderTest::IsOneEdgeFlush(), isOneEdgeFlush(), multiscaletest::MinEnclosingTriangleFinderTest::IsTriangleTouchingPolygon(), isTriangleTouchingPolygon(), multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle(), and TEST().

---

```
7.74.2.21 bool Geometry2D::lineCircleIntersection (cv::Point2f a, cv::Point2f b, const cv::Point2f & circleOrigin, double
radius, std::vector<cv::Point2f> & intersectionPoints) [static]
```

Determine if a line and a circle intersect and return the intersection points if they exist.

We translate all the points such that the circle origin coincides with the origin of the coordinate system. When returning the results, the intersection points are inverse translated.

#### Parameters

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| <i>a</i>                  | First point for determining the line                    |
| <i>b</i>                  | Second point for determining the line                   |
| <i>circleOrigin</i>       | Origin of the circle                                    |
| <i>radius</i>             | Radius of the circle                                    |
| <i>intersectionPoints</i> | The intersection points between the circle and the line |

< Two intersection points

< One intersection point

Definition at line 289 of file Geometry2D.cpp.

References lineCircleOneIntersectionPoint(), lineCircleTwoIntersectionPoints(), and translate().

Referenced by lineSegmentCircleIntersection(), and main().

**7.74.2.22 void Geometry2D::lineCircleOneIntersectionPoint ( const cv::Point2f & *circleOrigin*, double *A*, double *B*, double *C*, double *delta*, std::vector< cv::Point2f > & *intersectionPoints* ) [static], [private]**

Treat the case when the line and circle intersect in one point.

#### Parameters

|                           |                                                             |
|---------------------------|-------------------------------------------------------------|
| <i>circleOrigin</i>       | Origin of the circle                                        |
| <i>A</i>                  | $y_2 - y_1$                                                 |
| <i>B</i>                  | $x_1 - x_2$                                                 |
| <i>C</i>                  | $A*x_1 + B*y_1$                                             |
| <i>delta</i>              | $(4 * B^2 * C^2) - (4 * (A^2 + B^2) * (C^2 - (R^2 * A^2)))$ |
| <i>intersectionPoints</i> | Intersection points                                         |

Definition at line 459 of file Geometry2D.cpp.

References inverseTranslate().

Referenced by lineCircleIntersection().

**7.74.2.23 void Geometry2D::lineCircleTwoIntersectionPoints ( const cv::Point2f & *circleOrigin*, double *A*, double *B*, double *C*, double *delta*, std::vector< cv::Point2f > & *intersectionPoints* ) [static], [private]**

Treat the case when the line and circle intersect in two points.

#### Parameters

|                           |                                                             |
|---------------------------|-------------------------------------------------------------|
| <i>circleOrigin</i>       | Origin of the circle                                        |
| <i>A</i>                  | $y_2 - y_1$                                                 |
| <i>B</i>                  | $x_1 - x_2$                                                 |
| <i>C</i>                  | $A*x_1 + B*y_1$                                             |
| <i>delta</i>              | $(4 * B^2 * C^2) - (4 * (A^2 + B^2) * (C^2 - (R^2 * A^2)))$ |
| <i>intersectionPoints</i> | Intersection points                                         |

Definition at line 440 of file Geometry2D.cpp.

References inverseTranslate().

Referenced by lineCircleIntersection().

**7.74.2.24 void Geometry2D::lineEquationDeterminedByPoints ( const cv::Point2f & *p*, const cv::Point2f & *q*, double & *a*, double & *b*, double & *c* ) [static]**

Get the values of "a", "b" and "c" of the line equation  $ax + by + c = 0$ .

Get the values of "a", "b" and "c" of the line equation  $ax + by + c = 0$  knowing that point "p" and "q" are on the line

$$a = q.y - p.y \quad b = p.x - q.x \quad c = - (p.x * a) - (p.y * b)$$

#### Parameters

|          |                                      |
|----------|--------------------------------------|
| <i>p</i> | Point p                              |
| <i>q</i> | Point q                              |
| <i>a</i> | Parameter "a" from the line equation |

|          |                                      |
|----------|--------------------------------------|
| <i>b</i> | Parameter "b" from the line equation |
| <i>c</i> | Parameter "c" from the line equation |

Definition at line 169 of file Geometry2D.cpp.

References areEqualPoints().

Referenced by areOnTheSameSideOfLine(), and multiscale::MinEnclosingTriangleFinder::lineEquation←Parameters().

**7.74.2.25** `bool Geometry2D::lineIntersection ( const cv::Point2f & a1, const cv::Point2f & b1, const cv::Point2f & a2, const cv::Point2f & b2, cv::Point2f & intersection ) [static]`

Determine the intersection point of two lines, if this point exists.

Two lines intersect if they are not parallel (Parallel lines intersect at +/- infinity, but we do not consider this case here).

The lines are specified by a pair of points each. If they intersect, then the function returns true, else it returns false.

Lines can be specified in the following form:  $A_1x + B_1x = C_1$   $A_2x + B_2x = C_2$

If  $\det (= A_1xB_2 - A_2xB_1) == 0$ , then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with  $P(x, y)$  where:  $x = (C_1xB_2 - C_2xB_1) / (\det)$   $y = (C_2xA_1 - C_1xA_2) / (\det)$

#### Parameters

|                     |                                              |
|---------------------|----------------------------------------------|
| <i>a1</i>           | First point for determining the first line   |
| <i>b1</i>           | Second point for determining the first line  |
| <i>a2</i>           | First point for determining the second line  |
| <i>b2</i>           | Second point for determining the second line |
| <i>intersection</i> | The intersection point, if this point exists |

Definition at line 239 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::areIntersectingLines(), multiscale::MinEnclosingTriangleFinder::isLocalMinimalTriangle(), lineSegmentIntersection(), and multiscale::MinEnclosingTriangleFinder::middlePointOfSideB().

**7.74.2.26** `bool Geometry2D::lineIntersection ( const cv::Point & a1, const cv::Point & b1, const cv::Point & a2, const cv::Point & b2, cv::Point & intersection ) [static]`

Determine the intersection point of two lines, if this point exists.

Two lines intersect if they are not parallel (Parallel lines intersect at +/- infinity, but we do not consider this case here).

The lines are specified by a pair of points each. If they intersect, then the function returns true, else it returns false.

Lines can be specified in the following form:  $A_1x + B_1x = C_1$   $A_2x + B_2x = C_2$

If  $\det (= A_1xB_2 - A_2xB_1) == 0$ , then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with  $P(x, y)$  where:  $x = (C_1xB_2 - C_2xB_1) / (\det)$   $y = (C_2xA_1 - C_1xA_2) / (\det)$

#### Parameters

|                     |                                              |
|---------------------|----------------------------------------------|
| <i>a1</i>           | First point for determining the first line   |
| <i>b1</i>           | Second point for determining the first line  |
| <i>a2</i>           | First point for determining the second line  |
| <i>b2</i>           | Second point for determining the second line |
| <i>intersection</i> | The intersection point, if this point exists |

Definition at line 217 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

**7.74.2.27** `bool Geometry2D::lineIntersection ( double a1, double b1, double c1, double a2, double b2, double c2, cv::Point2f & intersection ) [static]`

Determine the intersection point of two lines, if this point exists.

Two lines intersect if they are not parallel (Parallel lines intersect at +/- infinity, but we do not consider this case here).

The lines are specified in the following form:  $A_1x + B_1x = C_1$   $A_2x + B_2x = C_2$

If  $\det (= A_1xB_2 - A_2xB_1) == 0$ , then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with  $P(x, y)$  where:  $x = (C_1xB_2 - C_2xB_1) / (\det)$   $y = (C_2xA_1 - C_1xA_2) / (\det)$

#### Parameters

|                     |                                              |
|---------------------|----------------------------------------------|
| <i>a1</i>           | A1                                           |
| <i>b1</i>           | B1                                           |
| <i>c1</i>           | C1                                           |
| <i>a2</i>           | A2                                           |
| <i>b2</i>           | B2                                           |
| <i>c2</i>           | C2                                           |
| <i>intersection</i> | The intersection point, if this point exists |

Definition at line 261 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

**7.74.2.28** `bool Geometry2D::lineSegmentCircleIntersection ( const cv::Point2f & a, const cv::Point2f & b, const cv::Point2f & circleOrigin, double radius, std::vector<cv::Point2f> & intersectionPoints ) [static]`

Determine if a line segment and a circle intersect and return the intersection points if they exist.

We translate all the points such that the circle origin coincides with the origin of the coordinate system. When returning the results, the intersection points are inverse translated.

#### Parameters

|                           |                                                         |
|---------------------------|---------------------------------------------------------|
| <i>a</i>                  | First point for determining the line                    |
| <i>b</i>                  | Second point for determining the line                   |
| <i>circleOrigin</i>       | Origin of the circle                                    |
| <i>radius</i>             | Radius of the circle                                    |
| <i>intersectionPoints</i> | The intersection points between the circle and the line |

Definition at line 318 of file Geometry2D.cpp.

References lineCircleIntersection().

Referenced by main().

**7.74.2.29** `bool Geometry2D::lineSegmentIntersection ( const cv::Point & a1, const cv::Point & b1, const cv::Point & a2, const cv::Point & b2, cv::Point & intersection ) [static]`

Determine the intersection point of two line segments, if this point exists.

Find the intersection point of the lines, if this point exists. Let us assume that this point exists and let us denote it with P(x, y). Then, in order for the point to be the intersection of the segments and not of the lines, we have to verify the following conditions:

1.  $\min(a1.x, b1.x) \leq x \leq \max(a1.x, b1.x)$  – x coordinate is valid for first line segment
2.  $\min(a2.x, b2.x) \leq x \leq \max(a2.x, b2.x)$  – x coordinate is valid for second line segment
3.  $\min(a1.y, b1.y) \leq y \leq \max(a1.y, b1.y)$  – y coordinate is valid for first line segment
4.  $\min(a2.y, b2.y) \leq y \leq \max(a2.y, b2.y)$  – y coordinate is valid for second line segment

#### Parameters

|                     |                                              |
|---------------------|----------------------------------------------|
| <i>a1</i>           | First point for determining the first line   |
| <i>b1</i>           | Second point for determining the first line  |
| <i>a2</i>           | First point for determining the second line  |
| <i>b2</i>           | Second point for determining the second line |
| <i>intersection</i> | The intersection point, if this point exists |

Definition at line 275 of file Geometry2D.cpp.

References lineIntersection().

Referenced by multiscale::analysis::Detector::findGoodIntersectionPoints().

**7.74.2.30** `cv::Point2f Geometry2D::middlePoint ( const cv::Point2f & a, const cv::Point2f & b ) [static]`

Get the point in the middle of the segment determined by points "a" and "b".

#### Parameters

|          |         |
|----------|---------|
| <i>a</i> | Point a |
| <i>b</i> | Point b |

Definition at line 103 of file Geometry2D.cpp.

Referenced by multiscaletest::MinEnclosingTriangleFinderTest::IsTriangleTouchingPolygon(), isTriangleTouching← Polygon(), multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle(), and multiscale::MinEnclosing← TriangleFinder::middlePointOfSideB().

**7.74.2.31** `unsigned int Geometry2D::minimumDistancePointIndex ( const std::vector< cv::Point > & points, const cv::Point2f & origin ) [static]`

Get the index of the point which is the closest to the origin.

Get the index of the point P from the given set of points, such that for any point A from the set of points  $\text{dist}(A, \text{origin}) \geq \text{dist}(P, \text{origin})$ .

#### Parameters

|               |                   |
|---------------|-------------------|
| <i>points</i> | The set of points |
| <i>origin</i> | The origin        |

Definition at line 364 of file Geometry2D.cpp.

References distanceBtwPoints().

Referenced by multiscale::analysis::RegionDetector::createRegionFromPolygon(), and multiscale::analysis::← ClusterDetector::updateClusterOriginDependentValues().

7.74.2.32 double Geometry2D::oppositeAngle ( double *angle* ) [static]

Return the angle opposite to the given angle.

```
if (angle < 180) then return (angle + 180); else return (angle - 180); endif
```

## Parameters

|              |       |
|--------------|-------|
| <i>angle</i> | Angle |
|--------------|-------|

Definition at line 58 of file Geometry2D.cpp.

Referenced by multiscale::MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor(), isOppositeAngleBetween(), and isOppositeAngleBetweenNonReflex().

7.74.2.33 void Geometry2D::orthogonalLineToAnotherLineEdgePoints ( const cv::Point & *a1*, const cv::Point & *b1*, cv::Point & *a2*, cv::Point & *b2*, int *nrOfRows*, int *nrOfCols* ) [static]

Find the points which are on the edge and on the line orthogonal to the line defined by 2 given points.

## Parameters

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <i>a1</i>       | First point for determining the first line         |
| <i>b1</i>       | Second point for determining the first line        |
| <i>a2</i>       | First point for determining the second line        |
| <i>b2</i>       | Second point for determining the second line       |
| <i>nrOfRows</i> | Maximum number of rows in the considered matrix    |
| <i>nrOfCols</i> | Maximum number of columns in the considered matrix |

Definition at line 110 of file Geometry2D.cpp.

References isPointOnEdge().

Referenced by multiscale::analysis::Detector::findGoodPointsForAngle().

7.74.2.34 bool Geometry2D::slopeOfLine ( const cv::Point2f & *a*, const cv::Point2f & *b*, double & *slope* ) [static]

Compute the slope of the line defined by points "a" and "b".

Returns true if the slope of the line can be computed and false otherwise.

## Parameters

|              |                                                         |
|--------------|---------------------------------------------------------|
| <i>a</i>     | Point a                                                 |
| <i>b</i>     | Point b                                                 |
| <i>slope</i> | Slope of the line if it is different from (+/-)infinity |

Definition at line 63 of file Geometry2D.cpp.

7.74.2.35 void Geometry2D::translate ( cv::Point2f & *point*, const cv::Point2f & *translation* ) [static], [private]

Translate a point by the given values.

## Parameters

|                    |                    |
|--------------------|--------------------|
| <i>point</i>       | The point          |
| <i>translation</i> | Translation values |

Definition at line 430 of file Geometry2D.cpp.

Referenced by lineCircleIntersection().

### 7.74.3 Member Data Documentation

7.74.3.1 `const int Geometry2D::MATRIX_START_INDEX = 1 [static]`

Definition at line 469 of file `Geometry2D.hpp`.

Referenced by `isPointOnEdge()`.

7.74.3.2 `const double Geometry2D::PI = 3.14159265358979323846264338327950288419716939937510 [static]`

Definition at line 468 of file `Geometry2D.hpp`.

Referenced by `angleBtwPoints()`, `angleOfLineWrtOxAxis()`, `multiscale::analysis::CircularityMeasure::compute()`, `multiscale::analysis::Cluster::isCircularMeasure()`, and `multiscale::analysis::Region::isCircularMeasure()`.

The documentation for this class was generated from the following files:

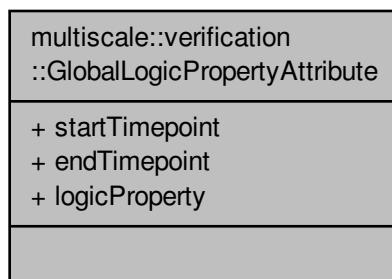
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Geometry2D.hpp`
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/Geometry2D.cpp`

## 7.75 multiscale::verification::GlobalLogicPropertyAttribute Class Reference

Class for representing a "globally" logic property attribute.

```
#include <GlobalLogicPropertyAttribute.hpp>
```

Collaboration diagram for `multiscale::verification::GlobalLogicPropertyAttribute`:



### Public Attributes

- `unsigned long startTimepoint`
- `unsigned long endTimepoint`
- `LogicPropertyAttributeType logicProperty`

### 7.75.1 Detailed Description

Class for representing a "globally" logic property attribute.

Definition at line 14 of file `GlobalLogicPropertyAttribute.hpp`.

## 7.75.2 Member Data Documentation

### 7.75.2.1 unsigned long multiscale::verification::GlobalLogicPropertyAttribute::endTimepoint

The considered end timepoint

Definition at line 19 of file GlobalLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty().

### 7.75.2.2 LogicPropertyAttributeType multiscale::verification::GlobalLogicPropertyAttribute::logicProperty

The logic property following the "globally" operator

Definition at line 20 of file GlobalLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty().

### 7.75.2.3 unsigned long multiscale::verification::GlobalLogicPropertyAttribute::startTimepoint

The considered start timepoint

Definition at line 18 of file GlobalLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty().

The documentation for this class was generated from the following file:

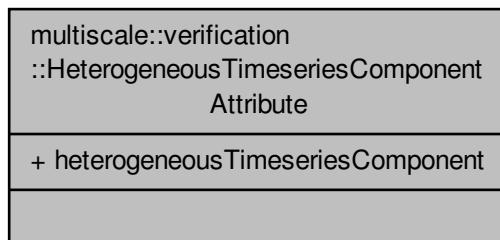
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/GlobalLogicPropertyAttribute.hpp

## 7.76 multiscale::verification::HeterogeneousTimeseriesComponentAttribute Class Reference

Class for representing a heterogeneous timeseries component attribute.

```
#include <HeterogeneousTimeseriesComponentAttribute.hpp>
```

Collaboration diagram for multiscale::verification::HeterogeneousTimeseriesComponentAttribute:



## Public Attributes

- [HeterogeneousTimeseriesComponentType heterogeneousTimeseriesComponent](#)

### 7.76.1 Detailed Description

Class for representing a heterogeneous timeseries component attribute.

Definition at line 28 of file `HeterogeneousTimeseriesComponentAttribute.hpp`.

### 7.76.2 Member Data Documentation

#### 7.76.2.1 `HeterogeneousTimeseriesComponentType multiscale::verification::HeterogeneousTimeseriesComponentAttribute::heterogeneousTimeseriesComponent`

The heterogeneous timeseries component

Definition at line 32 of file `HeterogeneousTimeseriesComponentAttribute.hpp`.

Referenced by `multiscale::verification::TimeseriesComponentVisitor::operator()()`.

The documentation for this class was generated from the following file:

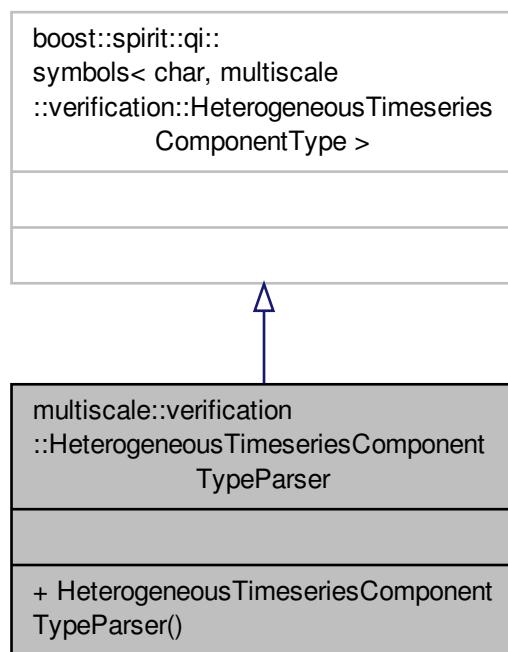
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HeterogeneousTimeseriesComponentAttribute.hpp`

## 7.77 `multiscale::verification::HeterogeneousTimeseriesComponentTypeParser` Struct Reference

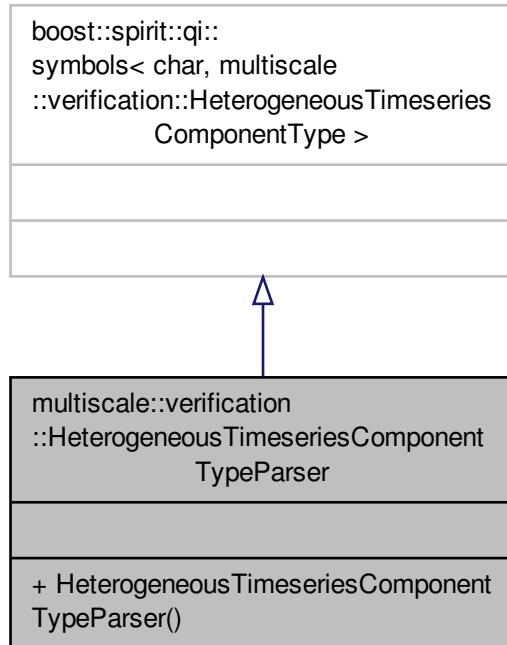
Symbol table and parser for the heterogeneous timeseries component type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::HeterogeneousTimeseriesComponentTypeParser:



Collaboration diagram for multiscale::verification::HeterogeneousTimeseriesComponentTypeParser:



## Public Member Functions

- [HeterogeneousTimeseriesComponentTypeParser \(\)](#)

### 7.77.1 Detailed Description

Symbol table and parser for the heterogeneous timeseries component type.

Definition at line 114 of file [SymbolTables.hpp](#).

### 7.77.2 Constructor & Destructor Documentation

#### 7.77.2.1 multiscale::verification::HeterogeneousTimeseriesComponentTypeParser::HeterogeneousTimeseriesComponentTypeParser( ) [inline]

Definition at line 117 of file [SymbolTables.hpp](#).

References [multiscale::verification::Peak](#), and [multiscale::verification::Valley](#).

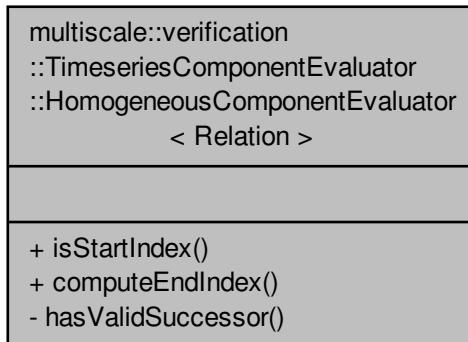
The documentation for this struct was generated from the following file:

- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp](#)

## 7.78 multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation > Class Template Reference

```
#include <TimeseriesComponentEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >:



### Public Member Functions

- bool `isStartIndex` (std::size\_t index, std::size\_t nrOfValues, const std::vector<double> &values, const Relation &relation) const

*Check if the given index is a homogeneous component start index.*

- std::size\_t `computeEndIndex` (std::size\_t startIndex, std::size\_t nrOfValues, const std::vector<double> &values, const Relation &relation) const

*Compute the homogeneous component end index considering the given homogeneous component start index.*

### Private Member Functions

- std::size\_t `hasValidSuccessor` (std::size\_t index, std::size\_t nrOfValues, const std::vector<double> &values, const Relation &relation) const

*Check if the provided homogeneous component index has a valid successor.*

#### 7.78.1 Detailed Description

```
template<typename Relation>class multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >
```

Definition at line 117 of file TimeseriesComponentEvaluator.hpp.

#### 7.78.2 Member Function Documentation

---

7.78.2.1 template<typename Relation > std::size\_t multiscale::verification::TimeseriesComponentEvaluator::←  
HomogeneousComponentEvaluator< Relation >::computeEndIndex ( std::size\_t startIndex, std::size\_t  
nrOfValues, const std::vector< double > & values, const Relation & relation ) const [inline]

Compute the homogeneous component end index considering the given homogeneous component start index.

A value located at index i in the collection,  $0 < i < (\text{values.size()} - 1)$ , is a nonuniform homogeneous component ending index if and only if  $\text{relation}(\text{values}[i - 1], \text{values}[i])$  and  $\text{!relation}(\text{values}[i], \text{values}[i + 1])$ , respectively  $\text{relation}(\text{values}[i - 1], \text{values}[i])$  if  $i = (\text{values.size()} - 1)$ .

#### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>startIndex</i> | The given start index           |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |
| <i>relation</i>   | The considered relation         |

Definition at line 167 of file TimeseriesComponentEvaluator.hpp.

References multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::isValidSuccessor(), and multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::isStartIndex().

7.78.2.2 template<typename Relation > std::size\_t multiscale::verification::TimeseriesComponentEvaluator::←  
HomogeneousComponentEvaluator< Relation >::hasValidSuccessor ( std::size\_t index, std::size\_t  
nrOfValues, const std::vector< double > & values, const Relation & relation ) const [inline], [private]

Check if the provided homogeneous component index has a valid successor.

#### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>index</i>      | The given index                 |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |
| <i>relation</i>   | The considered relation         |

Definition at line 192 of file TimeseriesComponentEvaluator.hpp.

Referenced by multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::computeEndIndex(), and multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::isStartIndex().

7.78.2.3 template<typename Relation > bool multiscale::verification::TimeseriesComponentEvaluator::←  
HomogeneousComponentEvaluator< Relation >::isStartIndex ( std::size\_t index, std::size\_t nrOfValues,  
const std::vector< double > & values, const Relation & relation ) const [inline]

Check if the given index is a homogeneous component start index.

A value located at index i in the collection,  $0 < i < (\text{values.size()} - 1)$ , is a nonuniform homogeneous component starting index if and only if  $\text{!relation}(\text{values}[i - 1], \text{values}[i])$  and  $\text{relation}(\text{values}[i], \text{values}[i + 1])$ , respectively  $\text{relation}(\text{values}[i], \text{values}[i + 1])$  if  $i = 0$ .

#### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>index</i>      | The given index                 |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |

|                 |                         |
|-----------------|-------------------------|
| <i>relation</i> | The considered relation |
|-----------------|-------------------------|

Definition at line 132 of file TimeseriesComponentEvaluator.hpp.

References multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::isValidSuccessor().

Referenced by multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >::computeEndIndex().

The documentation for this class was generated from the following file:

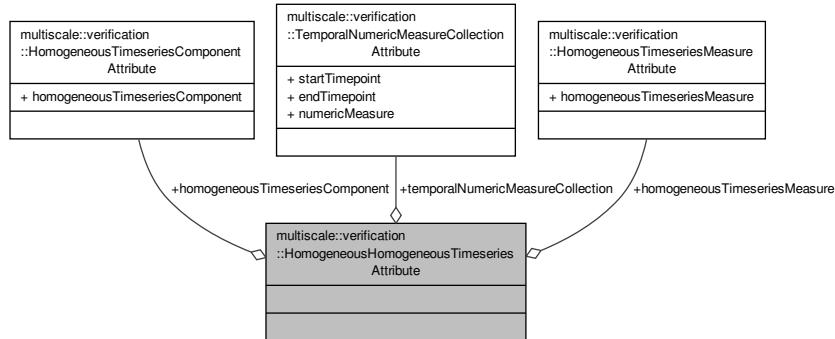
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp

## 7.79 multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute Class Reference

Class for representing a homogeneous homogeneous timeseries attribute.

```
#include <HomogeneousHomogeneousTimeseriesAttribute.hpp>
```

Collaboration diagram for multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute:



### Public Attributes

- `HomogeneousTimeseriesMeasureAttribute homogeneousTimeseriesMeasure`
- `HomogeneousTimeseriesComponentAttribute homogeneousTimeseriesComponent`
- `TemporalNumericMeasureCollectionAttribute temporalNumericMeasureCollection`

#### 7.79.1 Detailed Description

Class for representing a homogeneous homogeneous timeseries attribute.

Definition at line 17 of file HomogeneousHomogeneousTimeseriesAttribute.hpp.

#### 7.79.2 Member Data Documentation

**7.79.2.1 HomogeneousTimeseriesComponentAttribute multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::homogeneousTimeseriesComponent**

The homogeneous timeseries component

Definition at line 24 of file `HomogeneousHomogeneousTimeseriesAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

**7.79.2.2 HomogeneousTimeseriesMeasureAttribute multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::homogeneousTimeseriesMeasure**

The homogeneous timeseries measure

Definition at line 22 of file `HomogeneousHomogeneousTimeseriesAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

**7.79.2.3 TemporalNumericMeasureCollectionAttribute multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::temporalNumericMeasureCollection**

The temporal numeric measure collection

Definition at line 26 of file `HomogeneousHomogeneousTimeseriesAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

The documentation for this class was generated from the following file:

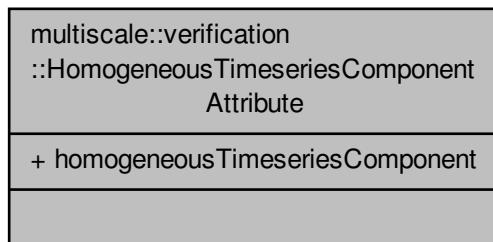
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousHomogeneousTimeseriesAttribute.hpp](#)

## 7.80 multiscale::verification::HomogeneousTimeseriesComponentAttribute Class Reference

Class for representing a homogeneous timeseries component attribute.

```
#include <HomogeneousTimeseriesComponentAttribute.hpp>
```

Collaboration diagram for `multiscale::verification::HomogeneousTimeseriesComponentAttribute`:



## Public Attributes

- [HomogeneousTimeseriesComponentType homogeneousTimeseriesComponent](#)

### 7.80.1 Detailed Description

Class for representing a homogeneous timeseries component attribute.

Definition at line 31 of file HomogeneousTimeseriesComponentAttribute.hpp.

### 7.80.2 Member Data Documentation

#### 7.80.2.1 [HomogeneousTimeseriesComponentType multiscale::verification::HomogeneousTimeseriesComponentAttribute::homogeneousTimeseriesComponent](#)

The homogeneous timeseries component

Definition at line 35 of file HomogeneousTimeseriesComponentAttribute.hpp.

Referenced by multiscale::verification::TimeseriesComponentEvaluator::evaluate().

The documentation for this class was generated from the following file:

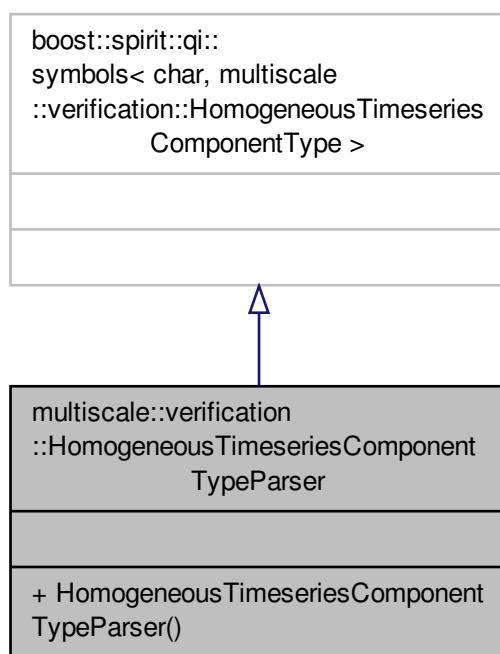
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp

## 7.81 multiscale::verification::HomogeneousTimeseriesComponentTypeParser Struct Reference

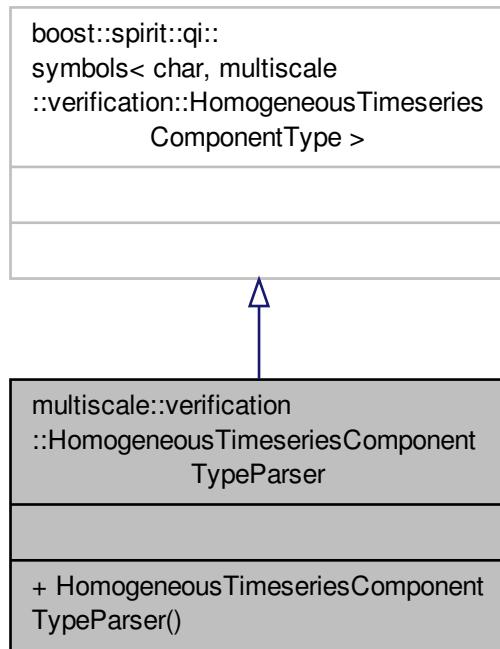
Symbol table and parser for the homogeneous timeseries component type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::HomogeneousTimeseriesComponentTypeParser:



Collaboration diagram for multiscale::verification::HomogeneousTimeseriesComponentTypeParser:



## Public Member Functions

- [HomogeneousTimeseriesComponentTypeParser \(\)](#)

### 7.81.1 Detailed Description

Symbol table and parser for the homogeneous timeseries component type.

Definition at line 127 of file SymbolTables.hpp.

### 7.81.2 Constructor & Destructor Documentation

#### 7.81.2.1 multiscale::verification::HomogeneousTimeseriesComponentTypeParser::HomogeneousTimeseriesComponentTypeParser( ) [inline]

Definition at line 130 of file SymbolTables.hpp.

References multiscale::verification::Ascent, multiscale::verification::Descent, multiscale::verification::Plateau, multiscale::verification::UniformAscent, and multiscale::verification::UniformDescent.

The documentation for this struct was generated from the following file:

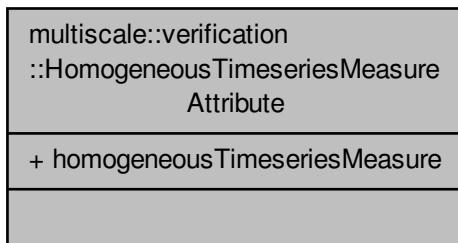
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp](#)

## 7.82 multiscale::verification::HomogeneousTimeseriesMeasureAttribute Class Reference

Class for representing a homogeneous timeseries measure attribute.

```
#include <HomogeneousTimeseriesMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::HomogeneousTimeseriesMeasureAttribute:



### Public Attributes

- [HomogeneousTimeseriesMeasureType homogeneousTimeseriesMeasure](#)

#### 7.82.1 Detailed Description

Class for representing a homogeneous timeseries measure attribute.

Definition at line 28 of file [HomogeneousTimeseriesMeasureAttribute.hpp](#).

#### 7.82.2 Member Data Documentation

##### 7.82.2.1 HomogeneousTimeseriesMeasureType multiscale::verification::HomogeneousTimeseriesMeasureAttribute<-->::homogeneousTimeseriesMeasure

The homogeneous timeseries measure

Definition at line 32 of file [HomogeneousTimeseriesMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::NumericMeasureCollectionVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

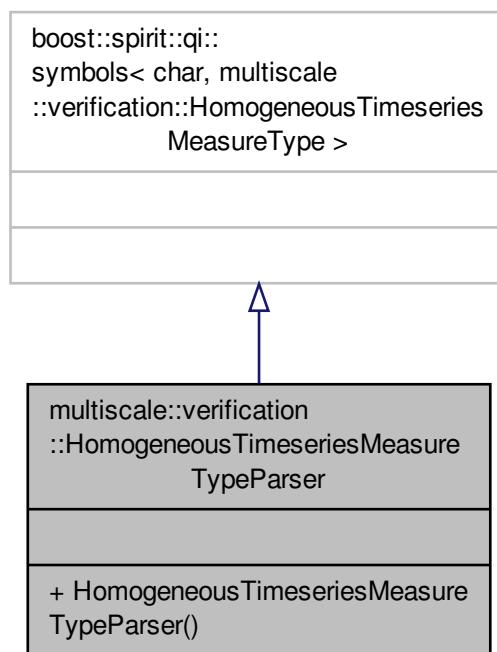
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesMeasureAttribute.hpp](#)

## 7.83 multiscale::verification::HomogeneousTimeseriesMeasureTypeParser Struct Reference

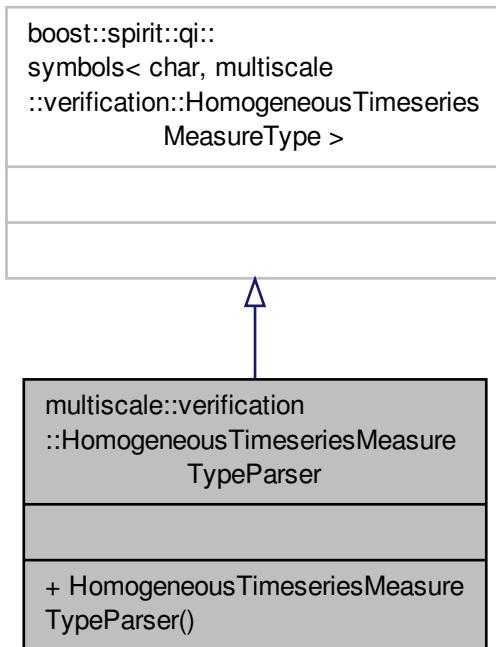
Symbol table and parser for the homogeneous timeseries measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::HomogeneousTimeseriesMeasureTypeParser:



Collaboration diagram for multiscale::verification::HomogeneousTimeseriesMeasureTypeParser:



## Public Member Functions

- [HomogeneousTimeseriesMeasureTypeParser \(\)](#)

### 7.83.1 Detailed Description

Symbol table and parser for the homogeneous timeseries measure type.

Definition at line 143 of file SymbolTables.hpp.

### 7.83.2 Constructor & Destructor Documentation

#### 7.83.2.1 multiscale::verification::HomogeneousTimeseriesMeasureTypeParser::HomogeneousTimeseriesMeasureTypeParser ( ) [inline]

Definition at line 146 of file SymbolTables.hpp.

References multiscale::verification::TimeSpan, and multiscale::verification::Values.

The documentation for this struct was generated from the following file:

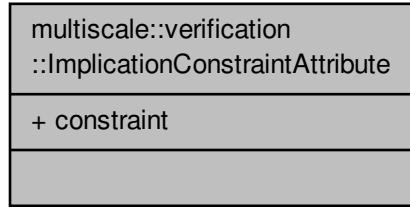
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.hpp](#)

## 7.84 multiscale::verification::ImplicationConstraintAttribute Class Reference

Class for representing an "implication" constraint attribute.

```
#include <ImplicationConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ImplicationConstraintAttribute:



### Public Attributes

- [ConstraintAttributeType constraint](#)

#### 7.84.1 Detailed Description

Class for representing an "implication" constraint attribute.

Definition at line 14 of file ImplicationConstraintAttribute.hpp.

#### 7.84.2 Member Data Documentation

##### 7.84.2.1 ConstraintAttributeType multiscale::verification::ImplicationConstraintAttribute::constraint

The constraint following the "implication" operator

Definition at line 18 of file ImplicationConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

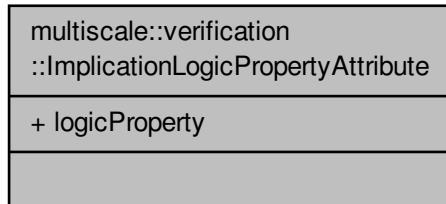
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationConstraintAttribute.hpp](#)

## 7.85 multiscale::verification::ImplicationLogicPropertyAttribute Class Reference

Class for representing an "implication" logic property attribute.

```
#include <ImplicationLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ImplicationLogicPropertyAttribute:



## Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.85.1 Detailed Description

Class for representing an "implication" logic property attribute.

Definition at line 14 of file [ImplicationLogicPropertyAttribute.hpp](#).

### 7.85.2 Member Data Documentation

#### 7.85.2.1 LogicPropertyAttributeType multiscale::verification::ImplicationLogicPropertyAttribute::logicProperty

The logical property following the "implication" operator

Definition at line 18 of file [ImplicationLogicPropertyAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

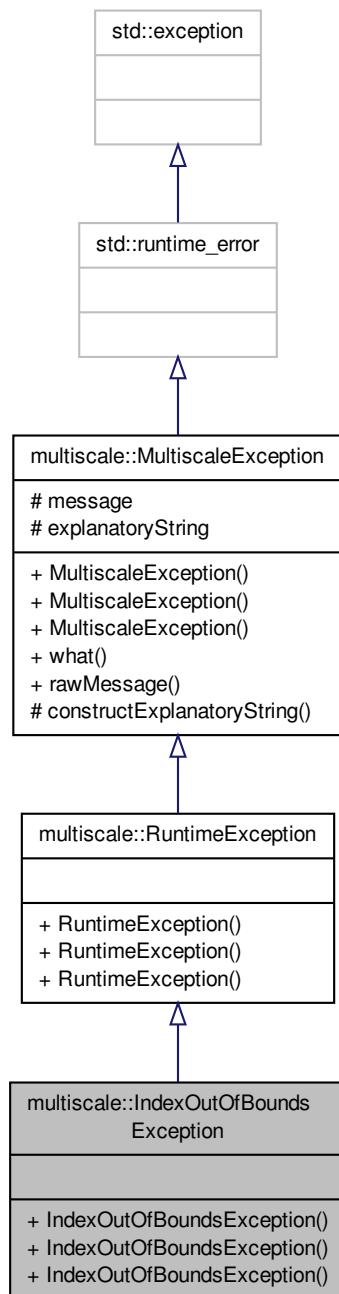
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/\*\*ImplicationLogicPropertyAttribute.hpp\*\*](#)

## 7.86 multiscale::IndexOutOfBoundsException Class Reference

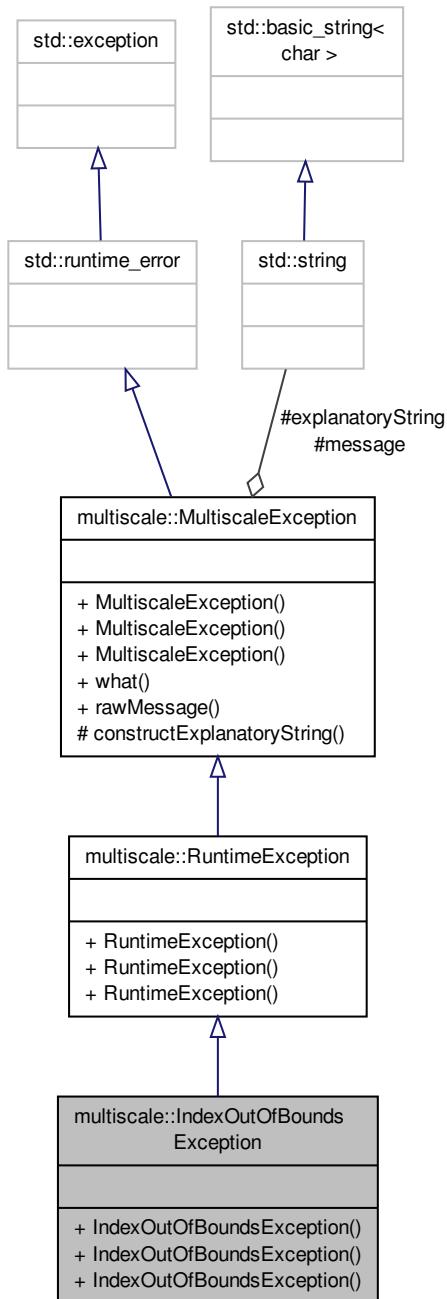
Class for representing an index out of bounds exception.

```
#include <IndexOutOfBoundsException.hpp>
```

Inheritance diagram for multiscale::IndexOutOfBoundsException:



Collaboration diagram for multiscale::IndexOutOfBoundsException:



## Public Member Functions

- `IndexOutOfBoundsException ()`
- `IndexOutOfBoundsException (const std::string &file, int line, const std::string &msg)`
- `IndexOutOfBoundsException (const std::string &file, int line, const char *msg)`

## Additional Inherited Members

### 7.86.1 Detailed Description

Class for representing an index out of bounds exception.

Definition at line 12 of file IndexOutOfBoundsException.hpp.

### 7.86.2 Constructor & Destructor Documentation

#### 7.86.2.1 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException( ) [inline]

Definition at line 16 of file IndexOutOfBoundsException.hpp.

#### 7.86.2.2 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 23 of file IndexOutOfBoundsException.hpp.

References multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_BEGIN, and multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_END.

#### 7.86.2.3 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 34 of file IndexOutOfBoundsException.hpp.

References multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_BEGIN, and multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_END.

The documentation for this class was generated from the following file:

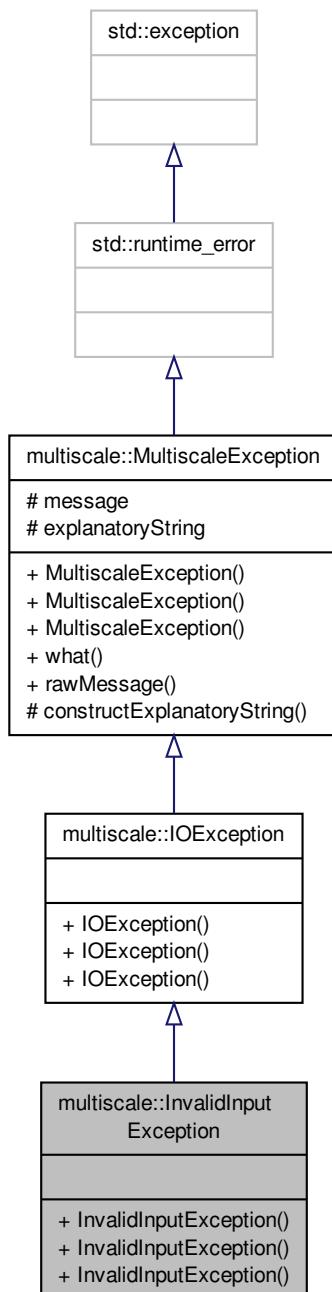
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/IndexOutOfBoundsException.hpp

## 7.87 multiscale::InvalidInputException Class Reference

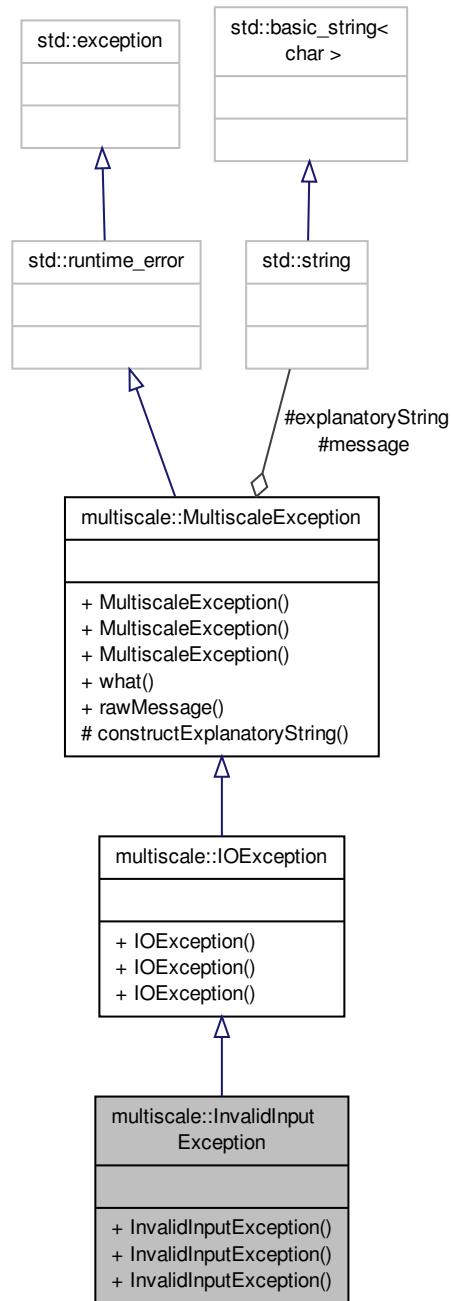
Class for representing invalid input exceptions.

```
#include <InvalidInputException.hpp>
```

Inheritance diagram for multiscale::InvalidInputException:



## Collaboration diagram for multiscale::InvalidInputException:



## Public Member Functions

- `InvalidInputException ()`
  - `InvalidInputException (const std::string &file, int line, const std::string &msg)`
  - `InvalidInputException (const std::string &file, int line, const char *msg)`

## Additional Inherited Members

### 7.87.1 Detailed Description

Class for representing invalid input exceptions.

Definition at line 12 of file InvalidInputException.hpp.

### 7.87.2 Constructor & Destructor Documentation

#### 7.87.2.1 multiscale::InvalidInputException::InvalidInputException( ) [inline]

Definition at line 16 of file InvalidInputException.hpp.

#### 7.87.2.2 multiscale::InvalidInputException::InvalidInputException( const std::string & file, int line, const std::string & msg ) [inline], [explicit]

Definition at line 18 of file InvalidInputException.hpp.

#### 7.87.2.3 multiscale::InvalidInputException::InvalidInputException( const std::string & file, int line, const char \* msg ) [inline], [explicit]

Definition at line 23 of file InvalidInputException.hpp.

The documentation for this class was generated from the following file:

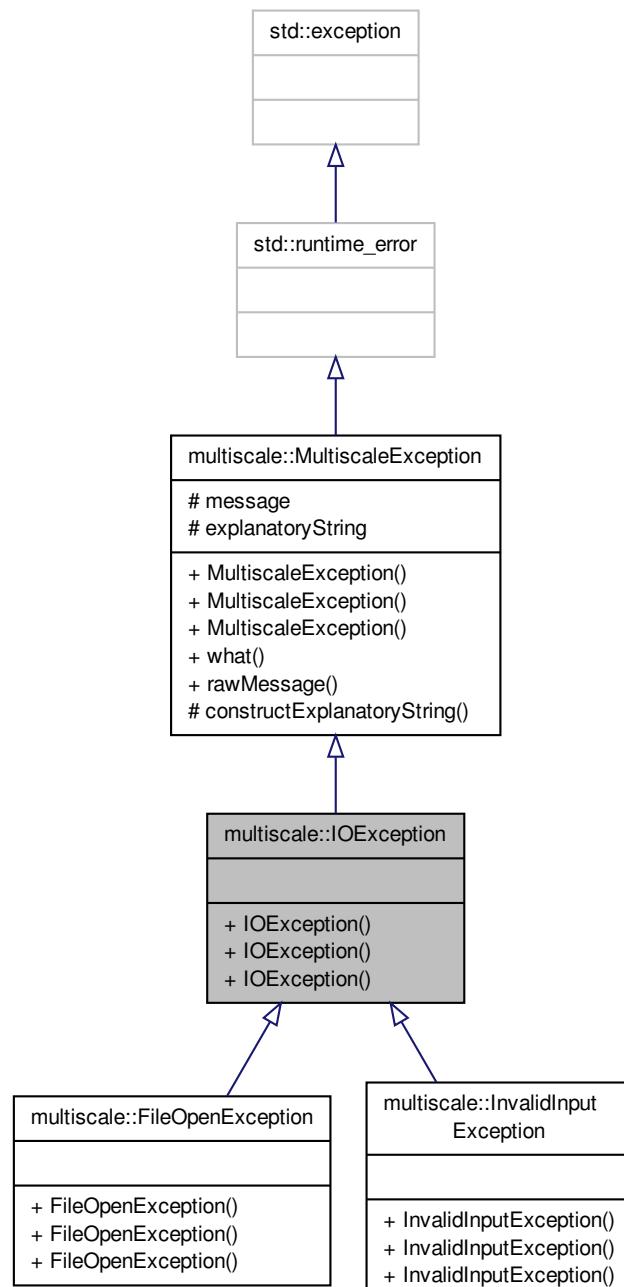
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/InvalidInputException.hpp

## 7.88 multiscale::IOException Class Reference

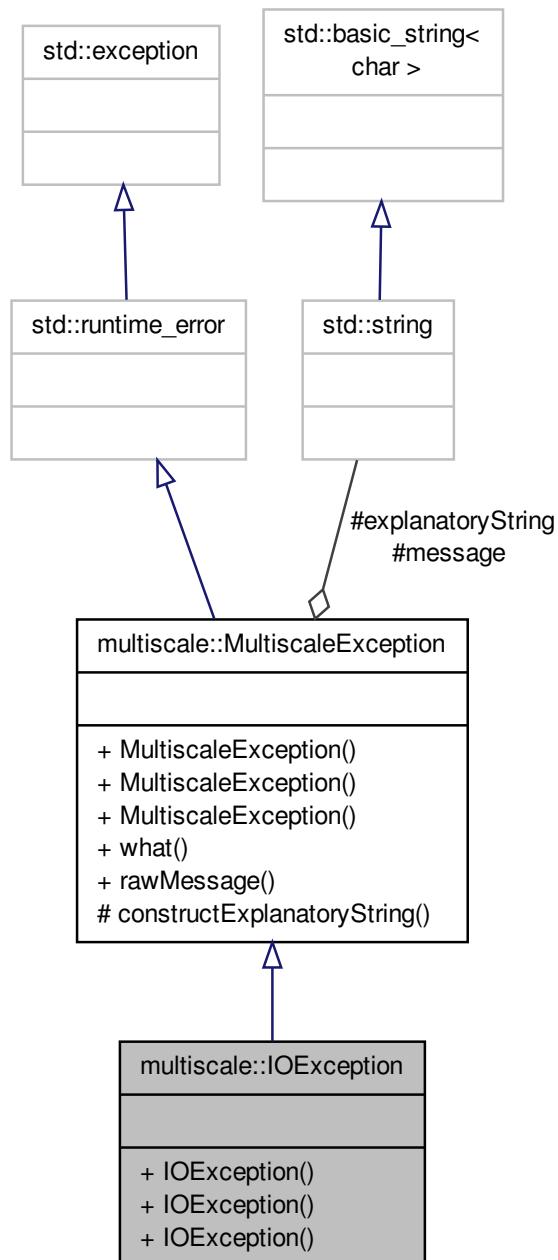
Class for representing input and output exceptions.

```
#include <IOException.hpp>
```

Inheritance diagram for multiscale::IOException:



Collaboration diagram for multiscale::IOException:



## Public Member Functions

- `IOException ()`
- `IOException (const std::string &file, int line, const std::string &msg)`
- `IOException (const std::string &file, int line, const char *msg)`

## Additional Inherited Members

### 7.88.1 Detailed Description

Class for representing input and output exceptions.

Definition at line 12 of file IOException.hpp.

### 7.88.2 Constructor & Destructor Documentation

#### 7.88.2.1 multiscale::IOException::IOException( ) [inline]

Definition at line 16 of file IOException.hpp.

#### 7.88.2.2 multiscale::IOException::IOException( const std::string & file, int line, const std::string & msg ) [inline], [explicit]

Definition at line 18 of file IOException.hpp.

#### 7.88.2.3 multiscale::IOException::IOException( const std::string & file, int line, const char \* msg ) [inline], [explicit]

Definition at line 23 of file IOException.hpp.

The documentation for this class was generated from the following file:

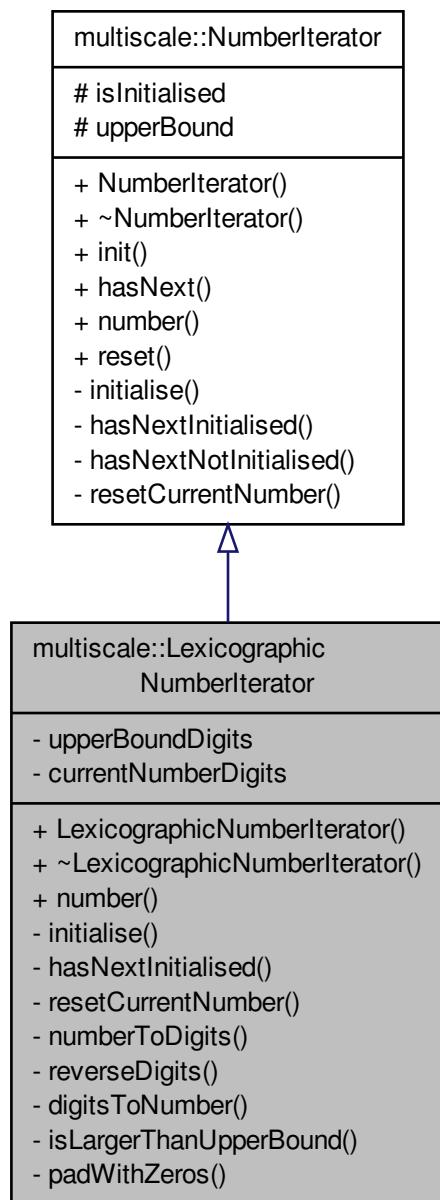
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/IOException.hpp

## 7.89 multiscale::LexicographicNumberIterator Class Reference

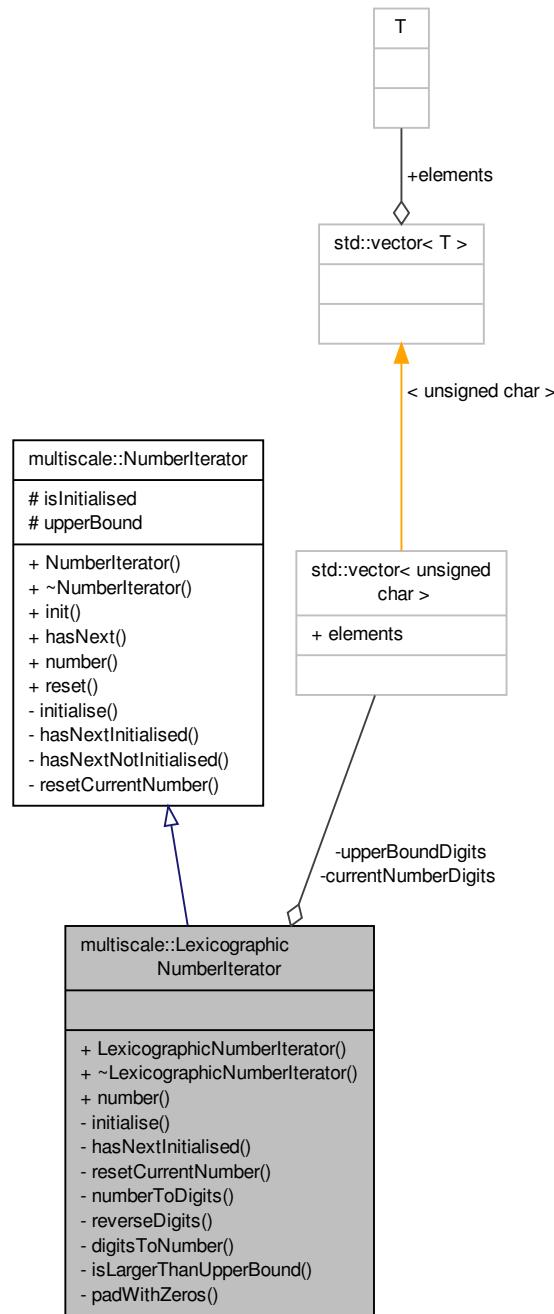
Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "\_" .

```
#include <LexicographicNumberIterator.hpp>
```

Inheritance diagram for multiscale::LexicographicNumberIterator:



Collaboration diagram for multiscale::LexicographicNumberIterator:



## Public Member Functions

- `LexicographicNumberIterator (unsigned int upperBound)`
- `~LexicographicNumberIterator ()`
- `unsigned int number ()`

*Get the number pointed by the iterator.*

## Private Member Functions

- void `initialise ()`  
*Initialise the std::vectors of digits.*
- bool `hasNextInitialised ()`  
*Check if there is a next number when in initialised state.*
- void `resetCurrentNumber ()`  
*Reset the digits of the current number to the initial value.*
- void `numberToDigits (unsigned int number, std::vector< unsigned char > &digits)`  
*Convert the number to a std::vector of digits.*
- void `reverseDigits (std::vector< unsigned char > &digits)`  
*Reverse the order of the digits.*
- unsigned int `digitsToNumber (std::vector< unsigned char > &digits)`  
*Convert the std::vector of digits to the number they represent.*
- bool `isLargerThanUpperBound (unsigned char lastDigit)`  
*Check if the current number with the provided last digit is greater than the upper bound.*
- void `padWithZeros ()`  
*Pad the current number with zeros.*

## Private Attributes

- std::vector< unsigned char > `upperBoundDigits`
- std::vector< unsigned char > `currentNumberDigits`

## Additional Inherited Members

### 7.89.1 Detailed Description

Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an `"_"`.

Definition at line 12 of file LexicographicNumberIterator.hpp.

### 7.89.2 Constructor & Destructor Documentation

#### 7.89.2.1 LexicographicNumberIterator::LexicographicNumberIterator ( `unsigned int upperBound` )

Definition at line 6 of file LexicographicNumberIterator.cpp.

References `initialise()`, and `multiscale::NumberIterator::reset()`.

#### 7.89.2.2 LexicographicNumberIterator::~LexicographicNumberIterator ( )

Definition at line 11 of file LexicographicNumberIterator.cpp.

References `currentNumberDigits`, and `upperBoundDigits`.

### 7.89.3 Member Function Documentation

#### 7.89.3.1 `unsigned int LexicographicNumberIterator::digitsToNumber ( std::vector< unsigned char > & digits )` [private]

Convert the std::vector of digits to the number they represent.

**Parameters**

|               |            |
|---------------|------------|
| <i>digits</i> | The digits |
|---------------|------------|

Definition at line 74 of file LexicographicNumberIterator.cpp.

References number().

Referenced by number(), and padWithZeros().

**7.89.3.2 bool LexicographicNumberIterator::hasNextInitialised( ) [private], [virtual]**

Check if there is a next number when in initialised state.

Implements [multiscale::NumberIterator](#).

Definition at line 26 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, isLargerThanUpperBound(), and padWithZeros().

**7.89.3.3 void LexicographicNumberIterator::initialise( ) [private], [virtual]**

Initialise the std::vectors of digits.

Implements [multiscale::NumberIterator](#).

Definition at line 20 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, numberToDigits(), multiscale::NumberIterator::upperBound, and upperBoundDigits.

Referenced by LexicographicNumberIterator().

**7.89.3.4 bool LexicographicNumberIterator::isLargerThanUpperBound( unsigned char *lastDigit* ) [private]**

Check if the current number with the provided last digit is greater than the upper bound.

Check if the current number is greater than the upper bound when replacing the last digit of the current number with the provided digit

**Parameters**

|                  |                |
|------------------|----------------|
| <i>lastDigit</i> | The last digit |
|------------------|----------------|

Definition at line 86 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, and upperBoundDigits.

Referenced by hasNextInitialised().

**7.89.3.5 unsigned int LexicographicNumberIterator::number( ) [virtual]**

Get the number pointed by the iterator.

Implements [multiscale::NumberIterator](#).

Definition at line 16 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, and digitsToNumber().

Referenced by digitsToNumber(), and main().

7.89.3.6 void LexicographicNumberIterator::numberToDigits ( *unsigned int number*, std::vector< unsigned char > & *digits* )  
[private]

Convert the number to a std::vector of digits.

## Parameters

|               |                          |
|---------------|--------------------------|
| <i>number</i> | The number               |
| <i>digits</i> | The digits of the number |

Definition at line 53 of file LexicographicNumberIterator.cpp.

References reverseDigits().

Referenced by initialise().

#### 7.89.3.7 void LexicographicNumberIterator::padWithZeros( ) [private]

Pad the current number with zeros.

Pad the current number with the maximum number of zeros such that it does not become larger than the upper bound

Definition at line 107 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, digitsToNumber(), multiscale::NumberIterator::upperBound, and upperBoundDigits.

Referenced by hasNextInitialised().

#### 7.89.3.8 void LexicographicNumberIterator::resetCurrentNumber( ) [private], [virtual]

Reset the digits of the current number to the initial value.

Implements [multiscale::NumberIterator](#).

Definition at line 42 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, and upperBoundDigits.

#### 7.89.3.9 void LexicographicNumberIterator::reverseDigits( std::vector< unsigned char > & *digits* ) [private]

Reverse the order of the digits.

Reverse the order of the digits such that the first one is swapped with the last one, the second one is swapped with the last but one and so on.

## Parameters

|               |            |
|---------------|------------|
| <i>digits</i> | The digits |
|---------------|------------|

Definition at line 63 of file LexicographicNumberIterator.cpp.

Referenced by numberToDigits().

### 7.89.4 Member Data Documentation

#### 7.89.4.1 std::vector< unsigned char > multiscale::LexicographicNumberIterator::currentNumberDigits [private]

The digits of the number to which the iterator points

Definition at line 17 of file LexicographicNumberIterator.hpp.

Referenced by hasNextInitialised(), initialise(), isLargerThanUpperBound(), number(), padWithZeros(), resetCurrentNumber(), and ~LexicographicNumberIterator().

### 7.89.4.2 std::vector<unsigned char> multiscale::LexicographicNumberIterator::upperBoundDigits [private]

The digits of the upper bound

Definition at line 16 of file LexicographicNumberIterator.hpp.

Referenced by initialise(), isLargerThanUpperBound(), padWithZeros(), resetCurrentNumber(), and ~LexicographicNumberIterator().

The documentation for this class was generated from the following files:

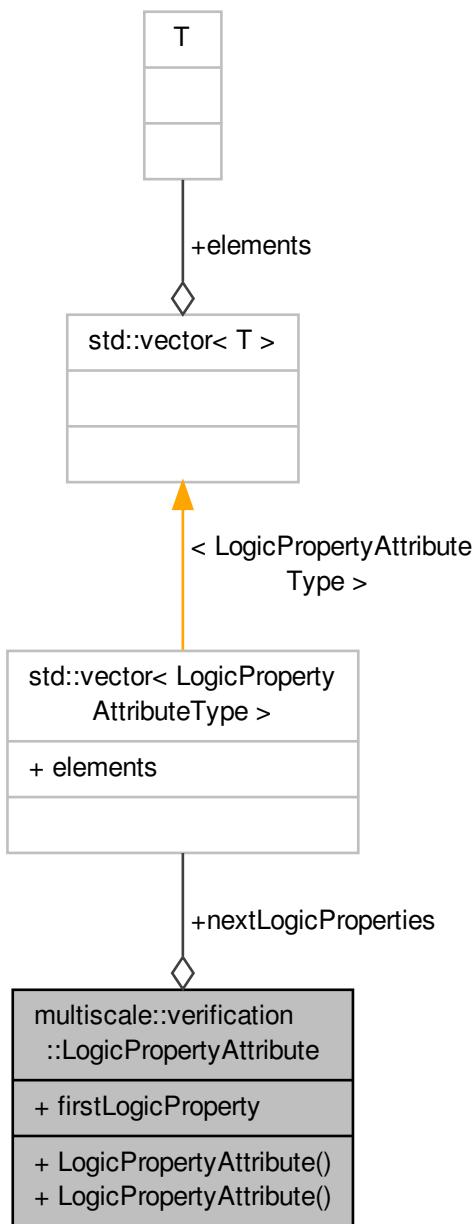
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/[LexicographicNumberIterator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/[LexicographicNumberIterator.cpp](#)

## 7.90 multiscale::verification::LogicPropertyAttribute Class Reference

Class for representing a logic property attribute.

```
#include <LogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::LogicPropertyAttribute:



## Public Member Functions

- `LogicPropertyAttribute ()`
- `LogicPropertyAttribute (const LogicPropertyAttributeType &firstLogicProperty, const std::vector< LogicPropertyAttributeType > &nextLogicProperties)`

## Public Attributes

- `LogicPropertyAttributeType firstLogicProperty`
- `std::vector< LogicPropertyAttributeType > nextLogicProperties`

### 7.90.1 Detailed Description

Class for representing a logic property attribute.

Definition at line 40 of file `LogicPropertyAttribute.hpp`.

### 7.90.2 Constructor & Destructor Documentation

#### 7.90.2.1 multiscale::verification::LogicPropertyAttribute::LogicPropertyAttribute( ) [inline]

Definition at line 49 of file `LogicPropertyAttribute.hpp`.

#### 7.90.2.2 multiscale::verification::LogicPropertyAttribute::LogicPropertyAttribute( const LogicPropertyAttributeType & firstLogicProperty, const std::vector< LogicPropertyAttributeType > & nextLogicProperties ) [inline]

Definition at line 51 of file `LogicPropertyAttribute.hpp`.

References `firstLogicProperty`, and `nextLogicProperties`.

### 7.90.3 Member Data Documentation

#### 7.90.3.1 LogicPropertyAttributeType multiscale::verification::LogicPropertyAttribute::firstLogicProperty

The first logic property

Definition at line 44 of file `LogicPropertyAttribute.hpp`.

Referenced by `multiscale::verification::LogicPropertyVisitor::constructEvaluationLogicProperty()`, `LogicPropertyAttribute()`, and `multiscale::verification::LogicPropertyVisitor::operator()`.

#### 7.90.3.2 std::vector<LogicPropertyAttributeType> multiscale::verification::LogicPropertyAttribute::nextLogicProperties

The next logic properties

Definition at line 45 of file `LogicPropertyAttribute.hpp`.

Referenced by `multiscale::verification::LogicPropertyVisitor::evaluateNextLogicProperties()`, and `LogicPropertyAttribute()`.

The documentation for this class was generated from the following file:

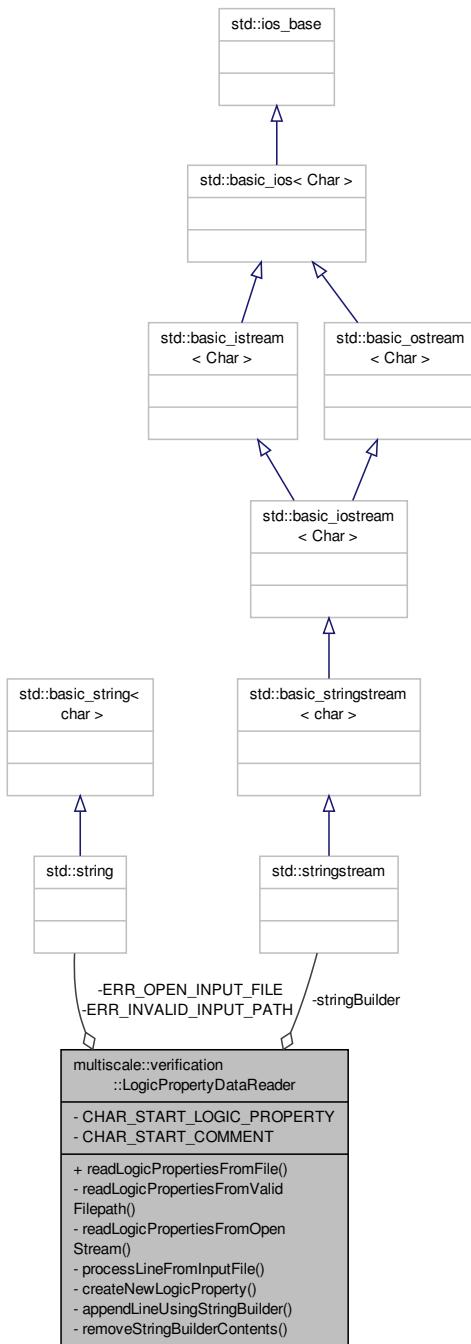
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp`

## 7.91 multiscale::verification::LogicPropertyDataReader Class Reference

Class used to input logic properties.

```
#include <LogicPropertyDataReader.hpp>
```

Collaboration diagram for multiscale::verification::LogicPropertyDataReader:



## Public Member Functions

- `std::vector< std::string > readLogicPropertiesFromFile (const std::string &filepath)`

*Return the logic properties read from a file.*

## Private Member Functions

- std::vector< std::string > **readLogicPropertiesFromValidFilepath** (const std::string &fin)  
*Read the logic properties from the given file.*
- std::vector< std::string > **readLogicPropertiesFromOpenStream** (std::ifstream &fin)  
*Read the logic properties from the given already opened input stream.*
- void **processLineFromInputFile** (const std::string &line, std::vector< std::string > &logicProperties)  
*Process a line from the input file.*
- void **createNewLogicProperty** (std::vector< std::string > &logicProperties)  
*Create a new logic property from the string builder contents.*
- void **appendLineUsingStringBuilder** (const std::string &line)  
*Append the given line to the string builder contents.*
- void **removeStringBuilderContents** ()  
*Remove the contents of the string builder.*

## Private Attributes

- std::stringstream **stringBuilder**

## Static Private Attributes

- static const std::string **ERR\_INVALID\_INPUT\_PATH** = "The path to the file containing the logic queries is invalid. Please change."
- static const std::string **ERR\_OPEN\_INPUT\_FILE** = "The file containing the logic queries could not be opened. Please make sure it is not used by another process."
- static const char **CHAR\_START\_LOGIC\_PROPERTY** = 'P'
- static const char **CHAR\_START\_COMMENT** = '#'

### 7.91.1 Detailed Description

Class used to input logic properties.

Definition at line 15 of file LogicPropertyDataReader.hpp.

### 7.91.2 Member Function Documentation

#### 7.91.2.1 void LogicPropertyDataReader::appendLineUsingStringBuilder ( const std::string & line ) [private]

Append the given line to the string builder contents.

##### Parameters

|             |                |
|-------------|----------------|
| <i>line</i> | The given line |
|-------------|----------------|

Definition at line 70 of file LogicPropertyDataReader.cpp.

References stringBuilder.

Referenced by processLineFromInputFile().

#### 7.91.2.2 void LogicPropertyDataReader::createNewLogicProperty ( std::vector< std::string > & logicProperties ) [private]

Create a new logic property from the string builder contents.

A new logic property is created only if the size of the string builder contents is greater than 0.

**Parameters**

|                        |                                                                 |
|------------------------|-----------------------------------------------------------------|
| <i>logicProperties</i> | The collection of logic properties obtained from the input file |
|------------------------|-----------------------------------------------------------------|

Definition at line 58 of file LogicPropertyDataReader.cpp.

References removeStringBuilderContents(), and stringBuilder.

Referenced by processLineFromInputFile(), and readLogicPropertiesFromOpenStream().

**7.91.2.3 void LogicPropertyDataReader::processLineFromInputFile ( const std::string & *line*, std::vector< std::string > & *logicProperties* ) [private]**

Process a line from the input file.

**Parameters**

|                        |                                                                 |
|------------------------|-----------------------------------------------------------------|
| <i>line</i>            | The line read from the input file                               |
| <i>logicProperties</i> | The collection of logic properties obtained from the input file |

Definition at line 44 of file LogicPropertyDataReader.cpp.

References appendLineUsingStringBuilder(), CHAR\_START\_COMMENT, CHAR\_START\_LOGIC\_PROPERTY, and createNewLogicProperty().

Referenced by readLogicPropertiesFromOpenStream().

**7.91.2.4 std::vector< std::string > LogicPropertyDataReader::readLogicPropertiesFromFile ( const std::string & *inputfilepath* )**

Return the logic properties read from a file.

All lines which start with "#" are used to write comments. All lines which start with "P" introduce a new logic property.

**Parameters**

|                      |                            |
|----------------------|----------------------------|
| <i>inputfilepath</i> | The path to the input file |
|----------------------|----------------------------|

Definition at line 9 of file LogicPropertyDataReader.cpp.

References ERR\_INVALID\_INPUT\_PATH, multiscale::Filesystem::isValidFilePath(), MS\_throw, and readLogicPropertiesFromValidFilepath().

Referenced by multiscale::verification::ModelCheckingManager::initialiseLogicProperties(), and readQueriesFromFile().

**7.91.2.5 std::vector< std::string > LogicPropertyDataReader::readLogicPropertiesFromOpenStream ( std::ifstream & *fin* ) [private]**

Read the logic properties from the given already opened input stream.

**Parameters**

|            |                  |
|------------|------------------|
| <i>fin</i> | The input stream |
|------------|------------------|

Definition at line 31 of file LogicPropertyDataReader.cpp.

References createNewLogicProperty(), and processLineFromInputFile().

Referenced by readLogicPropertiesFromValidFilepath().

7.91.2.6 `std::vector< std::string > LogicPropertyDataReader::readLogicPropertiesFromValidFilepath ( const std::string & fin ) [private]`

Read the logic properties from the given file.

Definition at line 17 of file LogicPropertyDataReader.cpp.

References ERR\_OPEN\_INPUT\_FILE, MS\_throw, and readLogicPropertiesFromOpenStream().

Referenced by readLogicPropertiesFromFile().

7.91.2.7 `void LogicPropertyDataReader::removeStringBuilderContents ( ) [private]`

Remove the contents of the string builder.

Definition at line 74 of file LogicPropertyDataReader.cpp.

References stringBuilder.

Referenced by createNewLogicProperty().

### 7.91.3 Member Data Documentation

7.91.3.1 `const char LogicPropertyDataReader::CHAR_START_COMMENT = '#' [static], [private]`

Definition at line 78 of file LogicPropertyDataReader.hpp.

Referenced by processLineFromInputFile().

7.91.3.2 `const char LogicPropertyDataReader::CHAR_START_LOGIC_PROPERTY = 'P' [static], [private]`

Definition at line 77 of file LogicPropertyDataReader.hpp.

Referenced by processLineFromInputFile().

7.91.3.3 `const std::string LogicPropertyDataReader::ERR_INVALID_INPUT_PATH = "The path to the file containing the logic queries is invalid. Please change." [static], [private]`

Definition at line 74 of file LogicPropertyDataReader.hpp.

Referenced by readLogicPropertiesFromFile().

7.91.3.4 `const std::string LogicPropertyDataReader::ERR_OPEN_INPUT_FILE = "The file containing the logic queries could not be opened. Please make sure it is not used by another process." [static], [private]`

Definition at line 75 of file LogicPropertyDataReader.hpp.

Referenced by readLogicPropertiesFromValidFilepath().

7.91.3.5 `std::stringstream multiscale::verification::LogicPropertyDataReader::stringBuilder [private]`

The string builder used to concatenate strings

Definition at line 19 of file LogicPropertyDataReader.hpp.

Referenced by appendLineUsingStringBuilder(), createNewLogicProperty(), and removeStringBuilderContents().

The documentation for this class was generated from the following files:

- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/[LogicPropertyDataReader.hpp](#)
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[Logic←  
PropertyDataReader.cpp](#)

## 7.92 multiscale::verification::LogicPropertyGrammar< Iterator > Class Template Reference

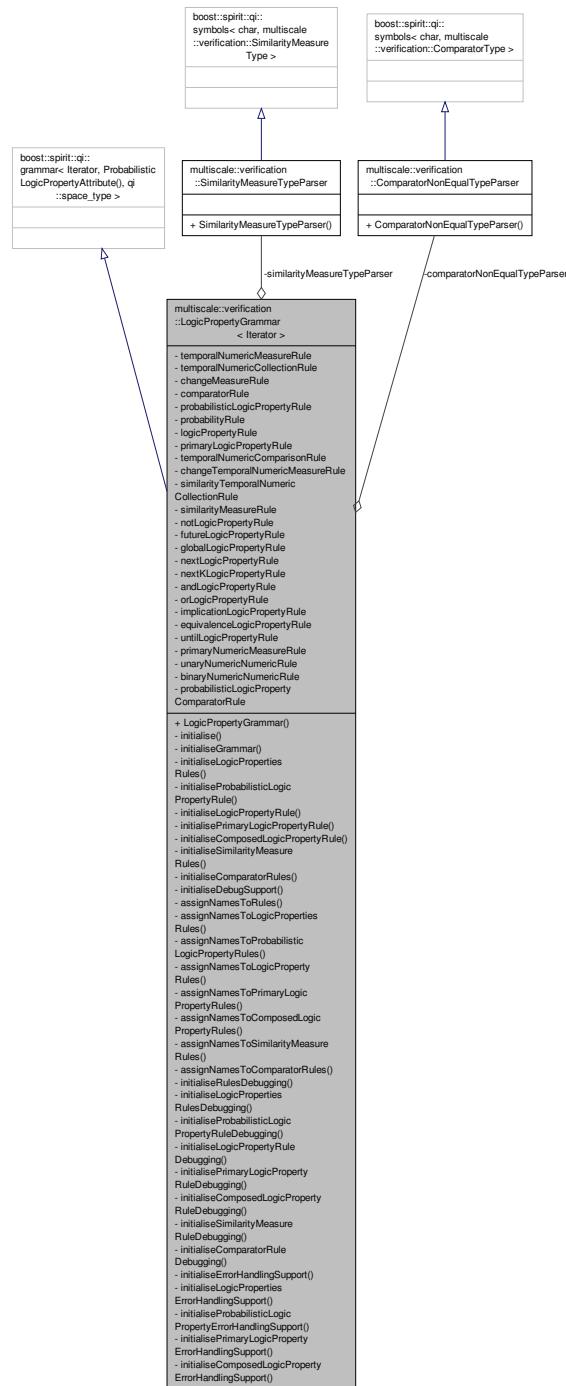
The grammar for parsing logic properties.

```
#include <LogicPropertyGrammar.hpp>
```

Inheritance diagram for multiscale::verification::LogicPropertyGrammar< Iterator >:



Collaboration diagram for multiscale::verification::LogicPropertyGrammar< Iterator >:



## Public Member Functions

- [LogicPropertyGrammar \(\)](#)

## Private Member Functions

- [void initialise \(\)](#)

- void **initialiseGrammar ()**  
*Initialise the grammar.*
- void **initialiseLogicPropertiesRules ()**  
*Initialise the logic properties rules.*
- void **initialiseProbabilisticLogicPropertyRule ()**  
*Initialise the probabilistic logic property rule.*
- void **initialiseLogicPropertyRule ()**  
*Initialise the logic property rule.*
- void **initialisePrimaryLogicPropertyRule ()**  
*Initialise the primary logic property rule.*
- void **initialiseComposedLogicPropertyRule ()**  
*Initialise the composed logic property rule.*
- void **initialiseSimilarityMeasureRules ()**  
*Initialise the similarity measure rules.*
- void **initialiseComparatorRules ()**  
*Initialise the comparator rules.*
- void **initialiseDebugSupport ()**  
*Initialise debug support.*
- void **assignNamesToRules ()**  
*Assign names to the rules.*
- void **assignNamesToLogicPropertiesRules ()**  
*Assign names to logic properties rules.*
- void **assignNamesToProbabilisticLogicPropertyRules ()**  
*Assign names to the probabilistic logic property rules.*
- void **assignNamesToLogicPropertyRules ()**  
*Assign names to the logic property rules.*
- void **assignNamesToPrimaryLogicPropertyRules ()**  
*Assign names to the primary logic property rules.*
- void **assignNamesToComposedLogicPropertyRules ()**  
*Assign names to the composed logic property rules.*
- void **assignNamesToSimilarityMeasureRules ()**  
*Assign names to the similarity measure rules.*
- void **assignNamesToComparatorRules ()**  
*Assign names to the comparator rules.*
- void **initialiseRulesDebugging ()**  
*Initialise the debugging of rules.*
- void **initialiseLogicPropertiesRulesDebugging ()**  
*Initialise the debugging of the logic properties rules.*
- void **initialiseProbabilisticLogicPropertyRuleDebugging ()**  
*Initialise debugging for the probabilistic logic property rule.*
- void **initialiseLogicPropertyRuleDebugging ()**  
*Initialise debugging for the logic property rule.*
- void **initialisePrimaryLogicPropertyRuleDebugging ()**  
*Initialise debugging for the primary logic property rule.*
- void **initialiseComposedLogicPropertyRuleDebugging ()**  
*Initialise debugging for the composed logic property rule.*
- void **initialiseSimilarityMeasureRuleDebugging ()**  
*Initialise debugging for the similarity measure rule.*
- void **initialiseComparatorRuleDebugging ()**  
*Initialise debugging for the comparator rule.*

- void [initialiseErrorHandlingSupport \(\)](#)  
*Initialise the error handling routines.*
- void [initialiseLogicPropertiesErrorHandlingSupport \(\)](#)  
*Initialise the logic properties error handling support.*
- void [initialiseProbabilisticLogicPropertyErrorHandlingSupport \(\)](#)  
*Initialise the probabilistic logic property error handling support.*
- void [initialisePrimaryLogicPropertyErrorHandlingSupport \(\)](#)  
*Initialise the primary logic property error handling support.*
- void [initialiseComposedLogicPropertyErrorHandlingSupport \(\)](#)  
*Initialise the compose logic property error handling support.*

## Private Attributes

- [TemporalNumericMeasureGrammar](#)  
< Iterator > [temporalNumericMeasureRule](#)
- [TemporalNumericCollectionGrammar](#)  
< Iterator > [temporalNumericCollectionRule](#)
- [ChangeMeasureGrammar](#)< Iterator > [changeMeasureRule](#)
- [ComparatorGrammar](#)< Iterator > [comparatorRule](#)
- qi::rule< Iterator,  
[ProbabilisticLogicPropertyAttribute\(\)](#),  
qi::space\_type > [probabilisticLogicPropertyRule](#)
- qi::rule< Iterator, double(),  
qi::space\_type > [probabilityRule](#)
- qi::rule< Iterator,  
[LogicPropertyAttribute\(\)](#),  
qi::space\_type > [logicPropertyRule](#)
- qi::rule< Iterator,  
[PrimaryLogicPropertyAttribute\(\)](#),  
qi::space\_type > [primaryLogicPropertyRule](#)
- qi::rule< Iterator,  
[TemporalNumericComparisonAttribute\(\)](#),  
qi::space\_type > [temporalNumericComparisonRule](#)
- qi::rule< Iterator,  
[ChangeTemporalNumericMeasureAttribute\(\)](#),  
qi::space\_type > [changeTemporalNumericMeasureRule](#)
- qi::rule< Iterator,  
[SimilarityTemporalNumericCollectionAttribute\(\)](#),  
qi::space\_type > [similarityTemporalNumericCollectionRule](#)
- qi::rule< Iterator,  
[SimilarityMeasureAttribute\(\)](#),  
qi::space\_type > [similarityMeasureRule](#)
- qi::rule< Iterator,  
[NotLogicPropertyAttribute\(\)](#),  
qi::space\_type > [notLogicPropertyRule](#)
- qi::rule< Iterator,  
[FutureLogicPropertyAttribute\(\)](#),  
qi::space\_type > [futureLogicPropertyRule](#)
- qi::rule< Iterator,  
[GlobalLogicPropertyAttribute\(\)](#),  
qi::space\_type > [globalLogicPropertyRule](#)
- qi::rule< Iterator,  
[NextLogicPropertyAttribute\(\)](#),  
qi::space\_type > [nextLogicPropertyRule](#)

- qi::rule< Iterator,  
  **NextKLogicPropertyAttribute()**,  
  qi::space\_type > **nextKLogicPropertyRule**
- qi::rule< Iterator,  
  **AndLogicPropertyAttribute()**,  
  qi::space\_type > **andLogicPropertyRule**
- qi::rule< Iterator,  
  **OrLogicPropertyAttribute()**,  
  qi::space\_type > **orLogicPropertyRule**
- qi::rule< Iterator,  
  **ImplicationLogicPropertyAttribute()**,  
  qi::space\_type > **implicationLogicPropertyRule**
- qi::rule< Iterator,  
  **EquivalenceLogicPropertyAttribute()**,  
  qi::space\_type > **equivalenceLogicPropertyRule**
- qi::rule< Iterator,  
  **UntilLogicPropertyAttribute()**,  
  qi::space\_type > **untilLogicPropertyRule**
- qi::rule< Iterator,  
  **PrimaryNumericMeasureAttribute()**,  
  qi::space\_type > **primaryNumericMeasureRule**
- qi::rule< Iterator,  
  **UnaryNumericNumericAttribute()**,  
  qi::space\_type > **unaryNumericNumericRule**
- qi::rule< Iterator,  
  **BinaryNumericNumericAttribute()**,  
  qi::space\_type > **binaryNumericNumericRule**
- qi::rule< Iterator,  
  **ComparatorAttribute()**,  
  qi::space\_type > **probabilisticLogicPropertyComparatorRule**
- **SimilarityMeasureTypeParser similarityMeasureTypeParser**
- **ComparatorNonEqualTypeParser comparatorNonEqualTypeParser**

### 7.92.1 Detailed Description

```
template<typename Iterator> class multiscale::verification::LogicPropertyGrammar< Iterator >
```

The grammar for parsing logic properties.

Definition at line 35 of file LogicPropertyGrammar.hpp.

### 7.92.2 Constructor & Destructor Documentation

7.92.2.1 `template<typename Iterator > multiscale::verification::LogicPropertyGrammar< Iterator >::LogicPropertyGrammar( )`

Definition at line 31 of file LogicPropertyGrammarDefinition.hpp.

References multiscale::verification::LogicPropertyGrammar< Iterator >::initialise().

### 7.92.3 Member Function Documentation

7.92.3.1 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToComparatorRules( ) [private]`

Assign names to the comparator rules.

Definition at line 292 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.2 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToComposedLogicPropertyRules( ) [private]**

Assign names to the composed logic property rules.

Definition at line 276 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.3 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToLogicPropertiesRules( ) [private]**

Assign names to logic properties rules.

Definition at line 241 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.4 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToLogicPropertyRules( ) [private]**

Assign names to the logic property rules.

Definition at line 257 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.5 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToPrimaryLogicPropertyRules( ) [private]**

Assign names to the primary logic property rules.

Definition at line 263 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.6 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToProbabilisticLogicPropertyRules( ) [private]**

Assign names to the probabilistic logic property rules.

Definition at line 250 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.7 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToRules( ) [private]**

Assign names to the rules.

Definition at line 233 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.8 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::assignNamesToSimilarityMeasureRules( ) [private]**

Assign names to the similarity measure rules.

Definition at line 286 of file LogicPropertyGrammarDefinition.hpp.

**7.92.3.9 template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialise( ) [private]**

Initialisation function.

Definition at line 41 of file LogicPropertyGrammarDefinition.hpp.

Referenced by multiscale::verification::LogicPropertyGrammar< Iterator >::LogicPropertyGrammar().

7.92.3.10 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseComparatorRuleDebugging( ) [private]

Initialise debugging for the comparator rule.

Definition at line 357 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.11 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseComparatorRules( ) [private]

Initialise the comparator rules.

Definition at line 217 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.12 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseComposedLogicPropertyErrorHandlingSupport( ) [private]

Initialise the compose logic property error handling support.

Definition at line 427 of file LogicPropertyGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.92.3.13 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseComposedLogicPropertyRule( ) [private]

Initialise the composed logic property rule.

Definition at line 181 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.14 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseComposedLogicPropertyRuleDebugging( ) [private]

Initialise debugging for the composed logic property rule.

Definition at line 341 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.15 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseDebugSupport( ) [private]

Initialise debug support.

Definition at line 224 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.16 template<typename Iterator> void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]

Initialise the error handling routines.

Definition at line 363 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.17 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseGrammar( ) [private]`

Initialise the grammar.

Definition at line 49 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.18 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseLogicPropertiesErrorHandlingSupport( ) [private]`

Initialise the logic properties error handling support.

Definition at line 369 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.19 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseLogicPropertiesRules( ) [private]`

Initialise the logic properties rules.

Definition at line 57 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.20 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseLogicPropertiesRulesDebugging( ) [private]`

Initialise the debugging of the logic properties rules.

Definition at line 306 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.21 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseLogicPropertyRule( ) [private]`

Initialise the logic property rule.

Definition at line 84 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.22 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseLogicPropertyRuleDebugging( ) [private]`

Initialise debugging for the logic property rule.

Definition at line 322 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.23 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport( ) [private]`

Initialise the primary logic property error handling support.

Definition at line 390 of file LogicPropertyGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.92.3.24 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialisePrimaryLogicPropertyRule( ) [private]`

Initialise the primary logic property rule.

Definition at line 98 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.25 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialisePrimaryLogicPropertyRuleDebugging( ) [private]`

Initialise debugging for the primary logic property rule.

Definition at line 328 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.26 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseProbabilisticLogicPropertyErrorHandlingSupport( ) [private]`

Initialise the probabilistic logic property error handling support.

Definition at line 377 of file LogicPropertyGrammarDefinition.hpp.

References multiscale::verification::handleProbabilityError, and multiscale::verification::handleUnexpectedToken← Error.

7.92.3.27 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseProbabilisticLogicPropertyRule( ) [private]`

Initialise the probabilistic logic property rule.

Definition at line 66 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.28 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseProbabilisticLogicPropertyRuleDebugging( ) [private]`

Initialise debugging for the probabilistic logic property rule.

Definition at line 315 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.29 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseRulesDebugging( ) [private]`

Initialise the debugging of rules.

Definition at line 298 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.30 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseSimilarityMeasureRuleDebugging( ) [private]`

Initialise debugging for the similarity measure rule.

Definition at line 351 of file LogicPropertyGrammarDefinition.hpp.

7.92.3.31 `template<typename Iterator > void multiscale::verification::LogicPropertyGrammar< Iterator >::initialiseSimilarityMeasureRules( ) [private]`

Initialise the similarity measure rules.

Definition at line 210 of file LogicPropertyGrammarDefinition.hpp.

## 7.92.4 Member Data Documentation

---

7.92.4.1 template<typename Iterator> qi::rule<Iterator, AndLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::andLogicPropertyRule [private]

The rule for parsing an "and" logic property

Definition at line 90 of file LogicPropertyGrammar.hpp.

7.92.4.2 template<typename Iterator> qi::rule<Iterator, BinaryNumericNumericAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::binaryNumericNumericRule [private]

The rule for parsing a binary numeric numeric attribute

Definition at line 109 of file LogicPropertyGrammar.hpp.

7.92.4.3 template<typename Iterator> ChangeMeasureGrammar<Iterator> multiscale::verification::Logic<→  
PropertyGrammar< Iterator >::changeMeasureRule [private]

The grammar for parsing a change measure

Definition at line 51 of file LogicPropertyGrammar.hpp.

7.92.4.4 template<typename Iterator> qi::rule<Iterator, ChangeTemporalNumericMeasureAttribute(), qi::space\_←  
type> multiscale::verification::LogicPropertyGrammar< Iterator >::changeTemporalNumericMeasureRule  
[private]

The rule for parsing a change temporal numeric measure

Definition at line 71 of file LogicPropertyGrammar.hpp.

7.92.4.5 template<typename Iterator> ComparatorNonEqualTypeParser multiscale<→  
::verification::LogicPropertyGrammar< Iterator >::comparatorNonEqualTypeParser  
[private]

The comparator type parser which does not accept the "=" symbol

Definition at line 122 of file LogicPropertyGrammar.hpp.

7.92.4.6 template<typename Iterator> ComparatorGrammar<Iterator> multiscale::verification::LogicProperty<→  
Grammar< Iterator >::comparatorRule [private]

The grammar for parsing a comparator

Definition at line 53 of file LogicPropertyGrammar.hpp.

7.92.4.7 template<typename Iterator> qi::rule<Iterator, EquivalenceLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::equivalenceLogicPropertyRule  
[private]

The rule for parsing an "equivalence" logic property

Definition at line 97 of file LogicPropertyGrammar.hpp.

7.92.4.8 template<typename Iterator> qi::rule<Iterator, FutureLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::futureLogicPropertyRule [private]

The rule for parsing a "future" logic property

Definition at line 81 of file LogicPropertyGrammar.hpp.

7.92.4.9 template<typename Iterator> qi::rule<Iterator, GlobalLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::globalLogicPropertyRule [private]

The rule for parsing a "global" logic property

Definition at line 83 of file LogicPropertyGrammar.hpp.

7.92.4.10 template<typename Iterator> qi::rule<Iterator, ImplicationLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::implicationLogicPropertyRule  
[private]

The rule for parsing an "implication" logic property

Definition at line 94 of file LogicPropertyGrammar.hpp.

7.92.4.11 template<typename Iterator> qi::rule<Iterator, LogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::logicPropertyRule [private]

The rule for parsing a logic property

Definition at line 63 of file LogicPropertyGrammar.hpp.

7.92.4.12 template<typename Iterator> qi::rule<Iterator, NextKLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::nextKLogicPropertyRule [private]

The rule for parsing a "next K" logic property

Definition at line 87 of file LogicPropertyGrammar.hpp.

7.92.4.13 template<typename Iterator> qi::rule<Iterator, NextLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::nextLogicPropertyRule [private]

The rule for parsing a "next" logic property

Definition at line 85 of file LogicPropertyGrammar.hpp.

7.92.4.14 template<typename Iterator> qi::rule<Iterator, NotLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::notLogicPropertyRule [private]

The rule for parsing a "not" logic property

Definition at line 79 of file LogicPropertyGrammar.hpp.

7.92.4.15 template<typename Iterator> qi::rule<Iterator, OrLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::orLogicPropertyRule [private]

The rule for parsing an "or" logic property

Definition at line 92 of file LogicPropertyGrammar.hpp.

7.92.4.16 template<typename Iterator> qi::rule<Iterator, PrimaryLogicPropertyAttribute(), qi::space\_type>  
multiscale::verification::LogicPropertyGrammar< Iterator >::primaryLogicPropertyRule [private]

The rule for parsing a primary logic property

Definition at line 66 of file LogicPropertyGrammar.hpp.

7.92.4.17 template<typename Iterator> qi::rule<Iterator, PrimaryNumericMeasureAttribute(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::primaryNumericMeasureRule [private]

The rule for parsing a primary numeric numeric attribute

Definition at line 103 of file LogicPropertyGrammar.hpp.

7.92.4.18 template<typename Iterator> qi::rule<Iterator, ComparatorAttribute(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::probabilisticLogicPropertyComparatorRule [private]

The rule for parsing a comparator for a probabilistic logic property

Definition at line 113 of file LogicPropertyGrammar.hpp.

7.92.4.19 template<typename Iterator> qi::rule<Iterator, ProbabilisticLogicPropertyAttribute(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::probabilisticLogicPropertyRule [private]

The rule for parsing a probabilistic logic property

Definition at line 58 of file LogicPropertyGrammar.hpp.

7.92.4.20 template<typename Iterator> qi::rule<Iterator, double(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::probabilityRule [private]

The rule for parsing a probability value

Definition at line 61 of file LogicPropertyGrammar.hpp.

7.92.4.21 template<typename Iterator> qi::rule<Iterator, SimilarityMeasureAttribute(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::similarityMeasureRule [private]

The rule for parsing a similarity measure

Definition at line 77 of file LogicPropertyGrammar.hpp.

7.92.4.22 template<typename Iterator> SimilarityMeasureTypeParser multiscale::verification::LogicPropertyGrammar< Iterator >::similarityMeasureTypeParser [private]

The similarity measure type parser

Definition at line 119 of file LogicPropertyGrammar.hpp.

7.92.4.23 template<typename Iterator> qi::rule<Iterator, SimilarityTemporalNumericCollectionAttribute(), qi::space\_type> multiscale::verification::LogicPropertyGrammar< Iterator >::similarityTemporalNumericCollectionRule [private]

The rule for parsing a similarity temporal numeric collection attribute

Definition at line 74 of file LogicPropertyGrammar.hpp.

---

7.92.4.24 template<typename Iterator> **TemporalNumericCollectionGrammar<Iterator>**  
**multiscale::verification::LogicPropertyGrammar< Iterator >::temporalNumericCollectionRule**  
[private]

The grammar for parsing a temporal numeric collection

Definition at line 47 of file LogicPropertyGrammar.hpp.

7.92.4.25 template<typename Iterator> qi::rule<Iterator, TemporalNumericComparisonAttribute(), qi::space\_type>  
**multiscale::verification::LogicPropertyGrammar< Iterator >::temporalNumericComparisonRule**  
[private]

The rule for parsing a temporal numeric comparison

Definition at line 68 of file LogicPropertyGrammar.hpp.

7.92.4.26 template<typename Iterator> **TemporalNumericMeasureGrammar<Iterator>**  
**multiscale::verification::LogicPropertyGrammar< Iterator >::temporalNumericMeasureRule**  
[private]

The grammar for parsing a temporal numeric measure

Definition at line 43 of file LogicPropertyGrammar.hpp.

7.92.4.27 template<typename Iterator> qi::rule<Iterator, UnaryNumericNumericAttribute(), qi::space\_type>  
**multiscale::verification::LogicPropertyGrammar< Iterator >::unaryNumericNumericRule** [private]

The rule for parsing a unary numeric numeric attribute

Definition at line 106 of file LogicPropertyGrammar.hpp.

7.92.4.28 template<typename Iterator> qi::rule<Iterator, UntilLogicPropertyAttribute(), qi::space\_type>  
**multiscale::verification::LogicPropertyGrammar< Iterator >::untilLogicPropertyRule** [private]

The rule for parsing an "until" logic property

Definition at line 100 of file LogicPropertyGrammar.hpp.

The documentation for this class was generated from the following files:

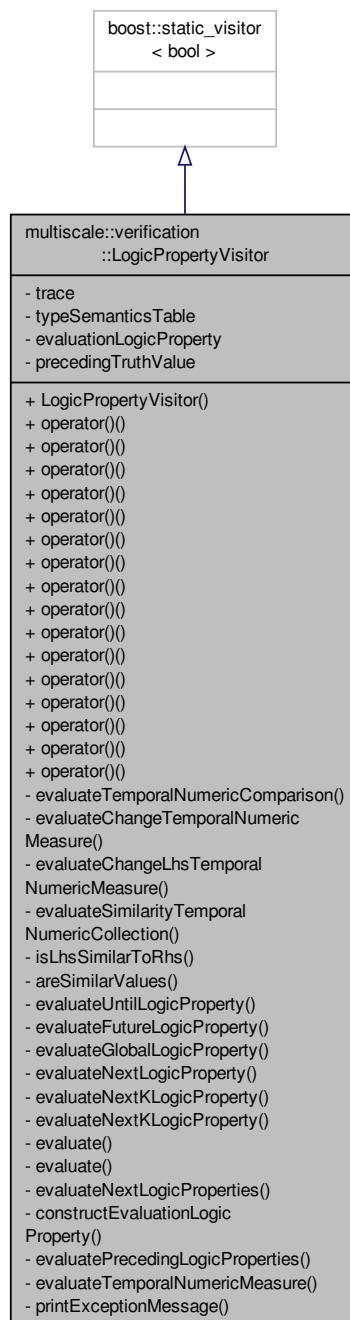
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[LogicPropertyGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[LogicPropertyGrammarDefinition.hpp](#)

## 7.93 multiscale::verification::LogicPropertyVisitor Class Reference

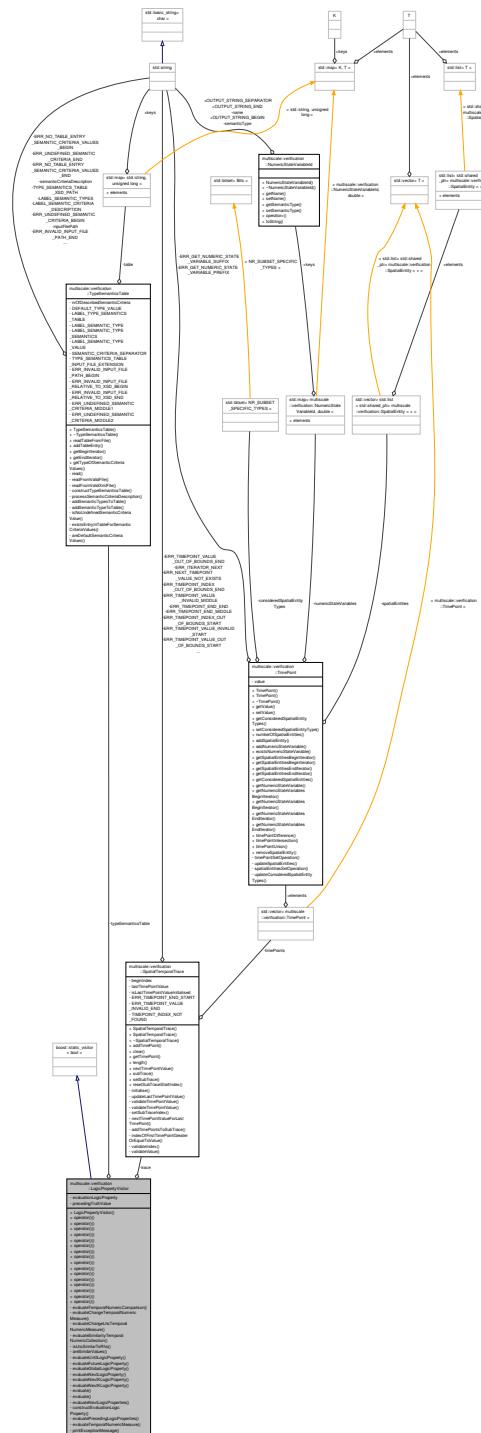
Class used to evaluate logic properties.

```
#include <LogicPropertyVisitor.hpp>
```

Inheritance diagram for multiscale::verification::LogicPropertyVisitor:



## Collaboration diagram for multiscale::verification::LogicPropertyVisitor:



## Public Member Functions

- `LogicPropertyVisitor` (const `SpatialTemporalTrace` &`trace`, const `TypeSemanticsTable` &`typeSemanticsTable`, bool `precedingTruthValue`=true)
  - template<typename T >  
bool `operator()` (const `Nil` &`logicProperty`, const T &`lhsLogicProperty`) const

*Overloading the "()" operator for the [Nil](#) alternative.*

- template<typename T>  
`bool operator() (const LogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [LogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const OrLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [OrLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const AndLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [AndLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const ImplicationLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [ImplicationLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const EquivalenceLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [EquivalenceLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const UntilLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the operator "(" for the [UntilLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const PrimaryLogicPropertyAttribute &logicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [PrimaryLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const TemporalNumericComparisonAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [TemporalNumericComparisonAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const ChangeTemporalNumericMeasureAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [ChangeTemporalNumericMeasureAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const SimilarityTemporalNumericCollectionAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [SimilarityTemporalNumericCollectionAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const NotLogicPropertyAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [NotLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const FutureLogicPropertyAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [FutureLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const GlobalLogicPropertyAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [GlobalLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const NextLogicPropertyAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [NextLogicPropertyAttribute](#) alternative.*
- template<typename T>  
`bool operator() (const NextKLogicPropertyAttribute &primaryLogicProperty, const T &lhsLogicProperty) const`  
*Overloading the "(" operator for the [NextKLogicPropertyAttribute](#) alternative.*

## Private Member Functions

- template<typename T >  
`bool evaluateTemporalNumericComparison (const TemporalNumericComparisonAttribute &comparisonAttribute, const T &lhsLogicProperty) const`  
*Evaluate the given TemporalNumericComparisonAttribute.*
- template<typename T >  
`bool evaluateChangeTemporalNumericMeasure (const ChangeTemporalNumericMeasureAttribute &changeAttribute, const T &lhsLogicProperty) const`  
*Evaluate the given ChangeTemporalNumericMeasureAttribute.*
- double `evaluateChangeLhsTemporalNumericMeasure (const ChangeTemporalNumericMeasureAttribute &changeAttribute) const`  
*Evaluate the left hand side temporal numeric measure of the given ChangeTemporalNumericMeasure.*
- bool `evaluateSimilarityTemporalNumericCollection (const SimilarityMeasureType &similarityMeasureType, const std::vector< double > &lhsTemporalNumericCollectionValues, const std::vector< double > &rhsTemporalNumericCollectionValues, double toleratedSimilarityDifference) const`  
*Evaluate the given SimilarityTemporalNumericCollectionAttribute.*
- bool `isLhsSimilarToRhs (const std::vector< double > &lhsTemporalNumericCollectionValues, const std::vector< double > &rhsTemporalNumericCollectionValues, double toleratedSimilarityDifference, const SimilarityMeasureType &similarityMeasureType) const`  
*Check if the left- and right-hand side collections of values are similar.*
- bool `areSimilarValues (double lhsValue, double rhsValue, double toleratedSimilarityDifference, const SimilarityMeasureType &similarityMeasureType) const`  
*Check if two values are similar considering the given similarity measure type.*
- template<typename T >  
`bool evaluateUntilLogicProperty (const UntilLogicPropertyAttribute &untilLogicProperty, const T &lhsLogicProperty) const`  
*Evaluate the given UntilLogicPropertyAttribute.*
- template<typename T >  
`bool evaluateFutureLogicProperty (const FutureLogicPropertyAttribute &futureLogicProperty, const T &lhsLogicProperty) const`  
*Evaluate the given FutureLogicPropertyAttribute.*
- template<typename T >  
`bool evaluateGlobalLogicProperty (const GlobalLogicPropertyAttribute &globalLogicProperty, const T &lhsLogicProperty) const`  
*Evaluate the given GlobalLogicPropertyAttribute.*
- template<typename T >  
`bool evaluateNextLogicProperty (const NextLogicPropertyAttribute &nextLogicProperty, const T &lhsLogicProperty) const`  
*Evaluate the given NextLogicPropertyAttribute.*
- template<typename T >  
`bool evaluateNextKLogicProperty (const NextKLogicPropertyAttribute &nextKLogicProperty, const T &lhsLogicProperty) const`  
*Evaluate the given NextKLogicPropertyAttribute.*
- template<typename T >  
`bool evaluateNextKLogicProperty (const LogicPropertyAttributeType &logicProperty, const T &lhsLogicProperty, unsigned long kValue) const`  
*Evaluate the given NextKLogicPropertyAttribute.*
- bool `evaluate (const LogicPropertyAttributeType &logicProperty, const SpatialTemporalTrace &trace) const`  
*Evaluate the logic property considering the given spatial temporal trace.*
- bool `evaluate (const PrimaryLogicPropertyAttributeType &primaryLogicProperty, const SpatialTemporalTrace &trace) const`  
*Evaluate the logic property considering the given spatial temporal trace.*
- bool `evaluateNextLogicProperties (const LogicPropertyAttribute &logicProperty, bool truthValue) const`

- Evaluate the next logic properties.
- LogicPropertyAttribute constructEvaluationLogicProperty (const LogicPropertyAttribute &logicProperty, const std::vector< LogicPropertyAttributeType > &evaluationLogicProperties) const
  - Construct a new logic property attribute using the evaluation logic properties.
- bool evaluatePrecedingLogicProperties (unsigned long startTime, unsigned long endTime, const LogicPropertyAttributeType &precedingLogicProperties) const
  - Evaluate the preceding logic properties considering the interval [startTime, endTime]
- double evaluateTemporalNumericMeasure (const TemporalNumericMeasureType &temporalNumericMeasure, const SpatialTemporalTrace &trace, unsigned int timePointIndex=0) const
  - Evaluate the temporal numeric measure considering the given spatial temporal trace.
- bool printExceptionMessage (const std::string &message) const
  - Print a warning message regarding the exception and return false.

## Private Attributes

- const SpatialTemporalTrace & trace
- const TypeSemanticsTable & typeSemanticsTable
- LogicPropertyAttributeType evaluationLogicProperty
- bool precedingTruthValue

### 7.93.1 Detailed Description

Class used to evaluate logic properties.

Definition at line 24 of file LogicPropertyVisitor.hpp.

### 7.93.2 Constructor & Destructor Documentation

7.93.2.1 multiscale::verification::LogicPropertyVisitor::LogicPropertyVisitor ( const SpatialTemporalTrace & trace, const TypeSemanticsTable & typeSemanticsTable, bool precedingTruthValue = true ) [inline]

Definition at line 38 of file LogicPropertyVisitor.hpp.

Referenced by evaluate(), and evaluateNextLogicProperties().

### 7.93.3 Member Function Documentation

7.93.3.1 bool multiscale::verification::LogicPropertyVisitor::areSimilarValues ( double lhsValue, double rhsValue, double toleratedSimilarityDifference, const SimilarityMeasureType & similarityMeasureType ) const [inline], [private]

Check if two values are similar considering the given similarity measure type.

#### Parameters

|                                      |                                                                |
|--------------------------------------|----------------------------------------------------------------|
| <i>similarityMeasureType</i>         | The specific similarity measure type                           |
| <i>lhsValue</i>                      | The left hand side value                                       |
| <i>rhsValue</i>                      | The right hand side value                                      |
| <i>toleratedSimilarityDifference</i> | The maximum tolerated similarity difference between two values |

Definition at line 432 of file LogicPropertyVisitor.hpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, multiscale::Numeric::lessOrEqual(), MS\_throw, multiscale::verification::Opposite, and multiscale::verification::Similar.

Referenced by isLhsSimilarToRhs().

**7.93.3.2 LogicPropertyAttribute multiscale::verification::LogicPropertyVisitor::constructEvaluationLogicProperty ( const LogicPropertyAttribute & *logicProperty*, const std::vector< LogicPropertyAttributeType > & *evaluationLogicProperties* ) const [inline], [private]**

Construct a new logic property attribute using the evaluation logic properties.

#### Parameters

|                                  |                                                                          |
|----------------------------------|--------------------------------------------------------------------------|
| <i>logicProperty</i>             | The logic property containing the currently evaluated logic subproperty  |
| <i>evaluationLogicProperties</i> | The logic properties preceding the currently evaluated logic subproperty |

Definition at line 630 of file LogicPropertyVisitor.hpp.

References multiscale::verification::LogicPropertyAttribute::firstLogicProperty.

Referenced by evaluateNextLogicProperties().

**7.93.3.3 bool multiscale::verification::LogicPropertyVisitor::evaluate ( const LogicPropertyAttributeType & *logicProperty*, const SpatialTemporalTrace & *trace* ) const [inline], [private]**

Evaluate the logic property considering the given spatial temporal trace.

#### Parameters

|                      |                                  |
|----------------------|----------------------------------|
| <i>logicProperty</i> | The logic property               |
| <i>trace</i>         | The given spatial temporal trace |

Definition at line 575 of file LogicPropertyVisitor.hpp.

References LogicPropertyVisitor().

Referenced by evaluateFutureLogicProperty(), evaluateGlobalLogicProperty(), evaluateNextKLogicProperty(), evaluatePrecedingLogicProperties(), evaluateUntilLogicProperty(), and operator()().

**7.93.3.4 bool multiscale::verification::LogicPropertyVisitor::evaluate ( const PrimaryLogicPropertyAttributeType & *primaryLogicProperty*, const SpatialTemporalTrace & *trace* ) const [inline], [private]**

Evaluate the logic property considering the given spatial temporal trace.

#### Parameters

|                             |                                  |
|-----------------------------|----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property       |
| <i>trace</i>                | The given spatial temporal trace |

Definition at line 586 of file LogicPropertyVisitor.hpp.

References LogicPropertyVisitor().

**7.93.3.5 double multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure ( const ChangeTemporalNumericMeasureAttribute & *changeAttribute* ) const [inline], [private]**

Evaluate the left hand side temporal numeric measure of the given ChangeTemporalNumericMeasure.

## Parameters

|                        |                                               |
|------------------------|-----------------------------------------------|
| <i>changeAttribute</i> | The change temporal numeric measure attribute |
|------------------------|-----------------------------------------------|

Definition at line 337 of file LogicPropertyVisitor.hpp.

References multiscale::verification::ChangeTemporalNumericMeasureAttribute::changeMeasure, multiscale::verification::ChangeMeasureAttribute::changeMeasureType, multiscale::verification::ChangeMeasureEvaluator::evaluate(), evaluateTemporalNumericMeasure(), multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::TimePoint::getValue(), and multiscale::verification::ChangeTemporalNumericMeasure::lhsTemporalNumericMeasure.

Referenced by evaluateChangeTemporalNumericMeasure().

**7.93.3.6 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateChangeTemporalNumericMeasure ( const ChangeTemporalNumericMeasureAttribute & *changeAttribute*, const T & *lhsLogicProperty* ) const [inline], [private]**

Evaluate the given [ChangeTemporalNumericMeasureAttribute](#).

## Parameters

|                         |                                               |
|-------------------------|-----------------------------------------------|
| <i>changeAttribute</i>  | The change temporal numeric measure attribute |
| <i>lhsLogicProperty</i> | The left hand side logic property             |

Definition at line 320 of file LogicPropertyVisitor.hpp.

References multiscale::verification::ChangeTemporalNumericMeasureAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, multiscale::verification::ComparatorEvaluator::evaluate(), evaluate::ChangeLhsTemporalNumericMeasure(), evaluateTemporalNumericMeasure(), and multiscale::verification::ChangeTemporalNumericMeasureAttribute::rhsTemporalNumericMeasure.

Referenced by operator()().

**7.93.3.7 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty ( const FutureLogicPropertyAttribute & *futureLogicProperty*, const T & *lhsLogicProperty* ) const [inline], [private]**

Evaluate the given [FutureLogicPropertyAttribute](#).

## Parameters

|                            |                                   |
|----------------------------|-----------------------------------|
| <i>futureLogicProperty</i> | The future logic property         |
| <i>lhsLogicProperty</i>    | The left hand side logic property |

Definition at line 491 of file LogicPropertyVisitor.hpp.

References multiscale::verification::FutureLogicPropertyAttribute::endTimepoint, evaluate(), multiscale::verification::FutureLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::nextTimePointValue(), multiscale::verification::SpatialTemporalTrace::setSubTrace(), and multiscale::verification::FutureLogicPropertyAttribute::startTimepoint.

Referenced by operator()().

**7.93.3.8 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty ( const GlobalLogicPropertyAttribute & *globalLogicProperty*, const T & *lhsLogicProperty* ) const [inline], [private]**

Evaluate the given [GlobalLogicPropertyAttribute](#).

**Parameters**

|                            |                                   |
|----------------------------|-----------------------------------|
| <i>globalLogicProperty</i> | The global logic property         |
| <i>lhsLogicProperty</i>    | The left hand side logic property |

Definition at line 515 of file LogicPropertyVisitor.hpp.

References multiscale::verification::GlobalLogicPropertyAttribute::endTimepoint, evaluate(), multiscale::verification::GlobalLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::nextTimePointValue(), multiscale::verification::SpatialTemporalTrace::setSubTrace(), and multiscale::verification::GlobalLogicPropertyAttribute::startTimepoint.

Referenced by operator()().

**7.93.3.9 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateNextKLogicProperty ( const NextKLogicPropertyAttribute & *nextKLogicProperty*, const T & *lhsLogicProperty* ) const [inline], [private]**

Evaluate the given [NextKLogicPropertyAttribute](#).

**Parameters**

|                           |                                   |
|---------------------------|-----------------------------------|
| <i>nextKLogicProperty</i> | The next "k" logic property       |
| <i>lhsLogicProperty</i>   | The left hand side logic property |

Definition at line 550 of file LogicPropertyVisitor.hpp.

References multiscale::verification::NextKLogicPropertyAttribute::logicProperty, and multiscale::verification::NextKLogicPropertyAttribute::nrOfTimepointsAhead.

Referenced by evaluateNextLogicProperty(), and operator()().

**7.93.3.10 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateNextKLogicProperty ( const LogicPropertyAttributeType & *logicProperty*, const T & *lhsLogicProperty*, unsigned long *kValue* ) const [inline], [private]**

Evaluate the given [NextKLogicPropertyAttribute](#).

**Parameters**

|                         |                                                            |
|-------------------------|------------------------------------------------------------|
| <i>logicProperty</i>    | The logic property enclosed by the next "k" logic property |
| <i>lhsLogicProperty</i> | The left hand side logic property                          |
| <i>kValue</i>           | The value of "k"                                           |

Definition at line 563 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::SpatialTemporalTrace::subTrace().

**7.93.3.11 bool multiscale::verification::LogicPropertyVisitor::evaluateNextLogicProperties ( const LogicPropertyAttribute & *logicProperty*, bool *truthValue* ) const [inline], [private]**

Evaluate the next logic properties.

Evaluate the next logic properties considering the given logic property, spatial temporal trace and truth value

**Parameters**

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
| <i>truthValue</i>    | The given truth value    |

Definition at line 603 of file LogicPropertyVisitor.hpp.

References `constructEvaluationLogicProperty()`, `LogicPropertyVisitor()`, and `multiscale::verification::LogicPropertyAttribute::nextLogicProperties`.

Referenced by `operator()()`.

**7.93.3.12 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateNextLogicProperty ( const NextLogicPropertyAttribute & *nextLogicProperty*, const T & *lhsLogicProperty* ) const [inline], [private]**

Evaluate the given `NextLogicPropertyAttribute`.

Parameters

|                          |                                   |
|--------------------------|-----------------------------------|
| <i>nextLogicProperty</i> | The next logic property           |
| <i>lhsLogicProperty</i>  | The left hand side logic property |

Definition at line 539 of file LogicPropertyVisitor.hpp.

References `evaluateNextKLogicProperty()`, and `multiscale::verification::NextLogicPropertyAttribute::logicProperty`.

Referenced by `operator()()`.

**7.93.3.13 bool multiscale::verification::LogicPropertyVisitor::evaluatePrecedingLogicProperties ( unsigned long *startTime*, unsigned long *endTime*, const LogicPropertyAttributeType & *precedingLogicProperties* ) const [inline], [private]**

Evaluate the preceding logic properties considering the interval [*startTime*, *endTime*)

Parameters

|                                 |                                           |
|---------------------------------|-------------------------------------------|
| <i>startTime</i>                | The considered start time value           |
| <i>endTime</i>                  | The considered end time value (exclusive) |
| <i>precedingLogicProperties</i> | The preceding logic properties            |

Definition at line 645 of file LogicPropertyVisitor.hpp.

References `evaluate()`, `multiscale::verification::SpatialTemporalTrace::nextTimePointValue()`, and `multiscale::verification::SpatialTemporalTrace::setSubTrace()`.

Referenced by `evaluateUntilLogicProperty()`.

**7.93.3.14 bool multiscale::verification::LogicPropertyVisitor::evaluateSimilarityTemporalNumericCollection ( const SimilarityMeasureType & *similarityMeasureType*, const std::vector< double > & *lhsTemporalNumericCollectionValues*, const std::vector< double > & *rhsTemporalNumericCollectionValues*, double *toleratedSimilarityDifference* ) const [inline], [private]**

Evaluate the given `SimilarityTemporalNumericCollectionAttribute`.

Parameters

|                              |                                      |
|------------------------------|--------------------------------------|
| <i>similarityMeasureType</i> | The specific similarity measure type |
|------------------------------|--------------------------------------|

|                                                       |                                                                |
|-------------------------------------------------------|----------------------------------------------------------------|
| <i>IhsTemporal←<br/>Numeric←<br/>CollectionValues</i> | The left hand side temporal numeric collection values          |
| <i>rhsTemporal←<br/>Numeric←<br/>CollectionValues</i> | The right hand side temporal numeric collection values         |
| <i>tolerated←<br/>Similarity←<br/>Difference</i>      | The maximum tolerated similarity difference between two values |

Definition at line 369 of file LogicPropertyVisitor.hpp.

References isLhsSimilarToRhs().

Referenced by operator()().

```
7.93.3.15 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericComparison (const TemporalNumericComparisonAttribute & comparisonAttribute, const T & IhsLogicProperty) const [inline], [private]
```

Evaluate the given [TemporalNumericComparisonAttribute](#).

#### Parameters

|                                  |                                          |
|----------------------------------|------------------------------------------|
| <i>comparison←<br/>Attribute</i> | The numeric numeric comparison attribute |
| <i>IhsLogicProperty</i>          | The left hand side logic property        |

Definition at line 298 of file LogicPropertyVisitor.hpp.

References multiscale::verification::TemporalNumericComparisonAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, multiscale::verification::ComparatorEvaluator::evaluate(), evaluate::TemporalNumericMeasure(), multiscale::verification::TemporalNumericComparisonAttribute::lhsTemporalNumericMeasure, and multiscale::verification::TemporalNumericComparisonAttribute::rhsTemporalNumericMeasure.

Referenced by operator()().

```
7.93.3.16 double multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericMeasure (const TemporalNumericMeasureType & temporalNumericMeasure, const SpatialTemporalTrace & trace, unsigned int timePointIndex = 0) const [inline], [private]
```

Evaluate the temporal numeric measure considering the given spatial temporal trace.

#### Parameters

|                                           |                                                               |
|-------------------------------------------|---------------------------------------------------------------|
| <i>temporal←<br/>Numeric←<br/>Measure</i> | The given temporal numeric measure                            |
| <i>trace</i>                              | The given spatial temporal trace                              |
| <i>timePointIndex</i>                     | The index of the considered starting timepoint from the trace |

Definition at line 667 of file LogicPropertyVisitor.hpp.

References multiscale::verification::SpatialTemporalTrace::subTrace().

Referenced by evaluateChangeLhsTemporalNumericMeasure(), evaluateChangeTemporalNumericMeasure(), and evaluateTemporalNumericComparison().

7.93.3.17 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty ( const  
UntilLogicPropertyAttribute & *untilLogicProperty*, const T & *lhsLogicProperty* ) const [inline],  
[private]

Evaluate the given UntilLogicPropertyAttribute.

## Parameters

|                           |                                   |
|---------------------------|-----------------------------------|
| <i>untilLogicProperty</i> | The until logic property          |
| <i>lhsLogicProperty</i>   | The left hand side logic property |

Definition at line 467 of file LogicPropertyVisitor.hpp.

References multiscale::verification::UntilLogicPropertyAttribute::endTimepoint, evaluate(), evaluatePrecedingLogicProperties(), multiscale::verification::UntilLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::nextTimePointValue(), multiscale::verification::SpatialTemporalTrace::setSubTrace(), and multiscale::verification::UntilLogicPropertyAttribute::startTimepoint.

Referenced by operator()().

```
7.93.3.18 bool multiscale::verification::LogicPropertyVisitor::isLhsSimilarToRhs (const std::vector< double > &
lhsTemporalNumericCollectionValues, const std::vector< double > & rhsTemporalNumericCollectionValues, double
toleratedSimilarityDifference, const SimilarityMeasureType & similarityMeasureType) const [inline],
[private]
```

Check if the left- and right-hand side collections of values are similar.

## Parameters

|                                           |                                                                |
|-------------------------------------------|----------------------------------------------------------------|
| <i>similarityMeasureType</i>              | The specific similarity measure type                           |
| <i>lhsTemporalNumericCollectionValues</i> | The left hand side temporal numeric collection values          |
| <i>rhsTemporalNumericCollectionValues</i> | The right hand side temporal numeric collection values         |
| <i>toleratedSimilarityDifference</i>      | The maximum tolerated similarity difference between two values |

Definition at line 395 of file LogicPropertyVisitor.hpp.

References areSimilarValues().

Referenced by evaluateSimilarTemporalNumericCollection().

```
7.93.3.19 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator() (const Nil & logicProperty,
const T & lhsLogicProperty) const [inline]
```

Overloading the "(") operator for the [Nil](#) alternative.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>lhsLogicProperty</i> | The left hand side logic property |

Definition at line 50 of file LogicPropertyVisitor.hpp.

```
7.93.3.20 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator() (const
LogicPropertyAttribute & logicProperty, const T & lhsLogicProperty) const [inline]
```

Overloading the "(") operator for the [LogicPropertyAttribute](#) alternative.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>lhsLogicProperty</i> | The left hand side logic property |

Definition at line 60 of file LogicPropertyVisitor.hpp.

References evaluate(), evaluateNextLogicProperties(), and multiscale::verification::LogicPropertyAttribute::first←LogicProperty.

7.93.3.21 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const  
OrLogicPropertyAttribute & *logicProperty*, const T & *lhsLogicProperty* ) const [inline]

Overloading the "()" operator for the [OrLogicPropertyAttribute](#) alternative.

Remark: Lazy evaluation is performed for efficiency purposes.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>lhsLogicProperty</i> | The left hand side logic property |

Definition at line 74 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::OrLogicPropertyAttribute::logicProperty.

7.93.3.22 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const  
AndLogicPropertyAttribute & *logicProperty*, const T & *lhsLogicProperty* ) const [inline]

Overloading the "()" operator for the [AndLogicPropertyAttribute](#) alternative.

Remark: Lazy evaluation is performed for efficiency purposes.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>lhsLogicProperty</i> | The left hand side logic property |

Definition at line 91 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::AndLogicPropertyAttribute::logicProperty.

7.93.3.23 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const  
ImplicationLogicPropertyAttribute & *logicProperty*, const T & *lhsLogicProperty* ) const [inline]

Overloading the "()" operator for the [ImplicationLogicPropertyAttribute](#) alternative.

Remark: Lazy evaluation is performed for efficiency purposes.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>lhsLogicProperty</i> | The left hand side logic property |

Definition at line 108 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::ImplicationLogicPropertyAttribute::logicProperty.

7.93.3.24 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const  
EquivalenceLogicPropertyAttribute & *logicProperty*, const T & *lhsLogicProperty* ) const [inline]

Overloading the "()" operator for the [EquivalenceLogicPropertyAttribute](#) alternative.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>IhsLogicProperty</i> | The left hand side logic property |

Definition at line 125 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::EquivalenceLogicPropertyAttribute::logicProperty.

7.93.3.25 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const UntilLogicPropertyAttribute & *logicProperty*, const T & *IhsLogicProperty* ) const [inline]

Overloading the operator "()" for the UntilLogicPropertyAttribute alternative.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>IhsLogicProperty</i> | The left hand side logic property |

Definition at line 140 of file LogicPropertyVisitor.hpp.

References evaluateUntilLogicProperty(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

7.93.3.26 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const PrimaryLogicPropertyAttribute & *logicProperty*, const T & *IhsLogicProperty* ) const [inline]

Overloading the "()" operator for the PrimaryLogicPropertyAttribute alternative.

## Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>logicProperty</i>    | The logic property                |
| <i>IhsLogicProperty</i> | The left hand side logic property |

Definition at line 154 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::PrimaryLogicPropertyAttribute::primaryLogicProperty.

7.93.3.27 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const TemporalNumericComparisonAttribute & *primaryLogicProperty*, const T & *IhsLogicProperty* ) const [inline]

Overloading the "()" operator for the TemporalNumericComparisonAttribute alternative.

## Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 165 of file LogicPropertyVisitor.hpp.

References evaluateTemporalNumericComparison(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

7.93.3.28 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const ChangeTemporalNumericMeasureAttribute & *primaryLogicProperty*, const T & *IhsLogicProperty* ) const [inline]

Overloading the "()" operator for the ChangeTemporalNumericMeasureAttribute alternative.

## Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 180 of file LogicPropertyVisitor.hpp.

References evaluateChangeTemporalNumericMeasure(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

```
7.93.3.29 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator() (const
 SimilarityTemporalNumericCollectionAttribute & primaryLogicProperty, const T & IhsLogicProperty) const
 [inline]
```

Overloading the "(" operator for the [SimilarityTemporalNumericCollectionAttribute](#) alternative.

## Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 195 of file LogicPropertyVisitor.hpp.

References evaluateSimilarityTemporalNumericCollection(), multiscale::verification::NumericMeasureCollectionEvaluator::evaluateTemporalNumericCollection(), multiscale::verification::SimilarityTemporalNumericCollectionAttribute::lhsTemporalNumericCollection, multiscale::verification::SimilarityTemporalNumericCollectionAttribute::rhsTemporalNumericCollection, multiscale::verification::SimilarityTemporalNumericCollectionAttribute::similarityMeasure, multiscale::verification::SimilarityMeasureAttribute::similarityMeasure, and multiscale::verification::SimilarityTemporalNumericCollectionAttribute::toleratedSimilarityDifference.

```
7.93.3.30 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator() (const
 NotLogicPropertyAttribute & primaryLogicProperty, const T & IhsLogicProperty) const [inline]
```

Overloading the "(" operator for the [NotLogicPropertyAttribute](#) alternative.

## Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 223 of file LogicPropertyVisitor.hpp.

References evaluate(), and multiscale::verification::NotLogicPropertyAttribute::logicProperty.

```
7.93.3.31 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator() (const
 FutureLogicPropertyAttribute & primaryLogicProperty, const T & IhsLogicProperty) const [inline]
```

Overloading the "(" operator for the [FutureLogicPropertyAttribute](#) alternative.

## Parameters

|                             |                            |
|-----------------------------|----------------------------|
| <i>primaryLogicProperty</i> | The primary logic property |
|-----------------------------|----------------------------|

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>IhsLogicProperty</i> | The left hand side logic property |
|-------------------------|-----------------------------------|

Definition at line 236 of file LogicPropertyVisitor.hpp.

References evaluateFutureLogicProperty(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

**7.93.3.32 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const GlobalLogicPropertyAttribute & primaryLogicProperty, const T & IhsLogicProperty ) const [inline]**

Overloading the "()" operator for the [GlobalLogicPropertyAttribute](#) alternative.

#### Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 251 of file LogicPropertyVisitor.hpp.

References evaluateGlobalLogicProperty(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

**7.93.3.33 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const NextLogicPropertyAttribute & primaryLogicProperty, const T & IhsLogicProperty ) const [inline]**

Overloading the "()" operator for the [NextLogicPropertyAttribute](#) alternative.

#### Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 266 of file LogicPropertyVisitor.hpp.

References evaluateNextLogicProperty(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

**7.93.3.34 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator() ( const NextKLogicPropertyAttribute & primaryLogicProperty, const T & IhsLogicProperty ) const [inline]**

Overloading the "()" operator for the [NextKLogicPropertyAttribute](#) alternative.

#### Parameters

|                             |                                   |
|-----------------------------|-----------------------------------|
| <i>primaryLogicProperty</i> | The primary logic property        |
| <i>IhsLogicProperty</i>     | The left hand side logic property |

Definition at line 281 of file LogicPropertyVisitor.hpp.

References evaluateNextKLogicProperty(), printExceptionMessage(), and multiscale::MultiscaleException::rawMessage().

**7.93.3.35 bool multiscale::verification::LogicPropertyVisitor::printExceptionMessage ( const std::string & message ) const [inline], [private]**

Print a warning message regarding the exception and return false.

**Parameters**

|                |                       |
|----------------|-----------------------|
| <i>message</i> | The exception message |
|----------------|-----------------------|

Definition at line 682 of file LogicPropertyVisitor.hpp.

References multiscale::ConsolePrinter::printWarningMessage(), and multiscale::verification::WRN\_LOGIC\_PROPERTY\_EVAL\_FALSE.

Referenced by operator()().

#### 7.93.4 Member Data Documentation

**7.93.4.1 LogicPropertyAttributeType multiscale::verification::LogicPropertyVisitor::evaluationLogicProperty [private]**

The logic property used only for evaluation purposes

Definition at line 31 of file LogicPropertyVisitor.hpp.

**7.93.4.2 bool multiscale::verification::LogicPropertyVisitor::precedingTruthValue [private]**

The truth value of the preceding logic property

Definition at line 34 of file LogicPropertyVisitor.hpp.

**7.93.4.3 const SpatialTemporalTrace& multiscale::verification::LogicPropertyVisitor::trace [private]**

The spatial temporal trace

Definition at line 28 of file LogicPropertyVisitor.hpp.

**7.93.4.4 const TypeSemanticsTable& multiscale::verification::LogicPropertyVisitor::typeSemanticsTable [private]**

The type semantics table

Definition at line 29 of file LogicPropertyVisitor.hpp.

The documentation for this class was generated from the following file:

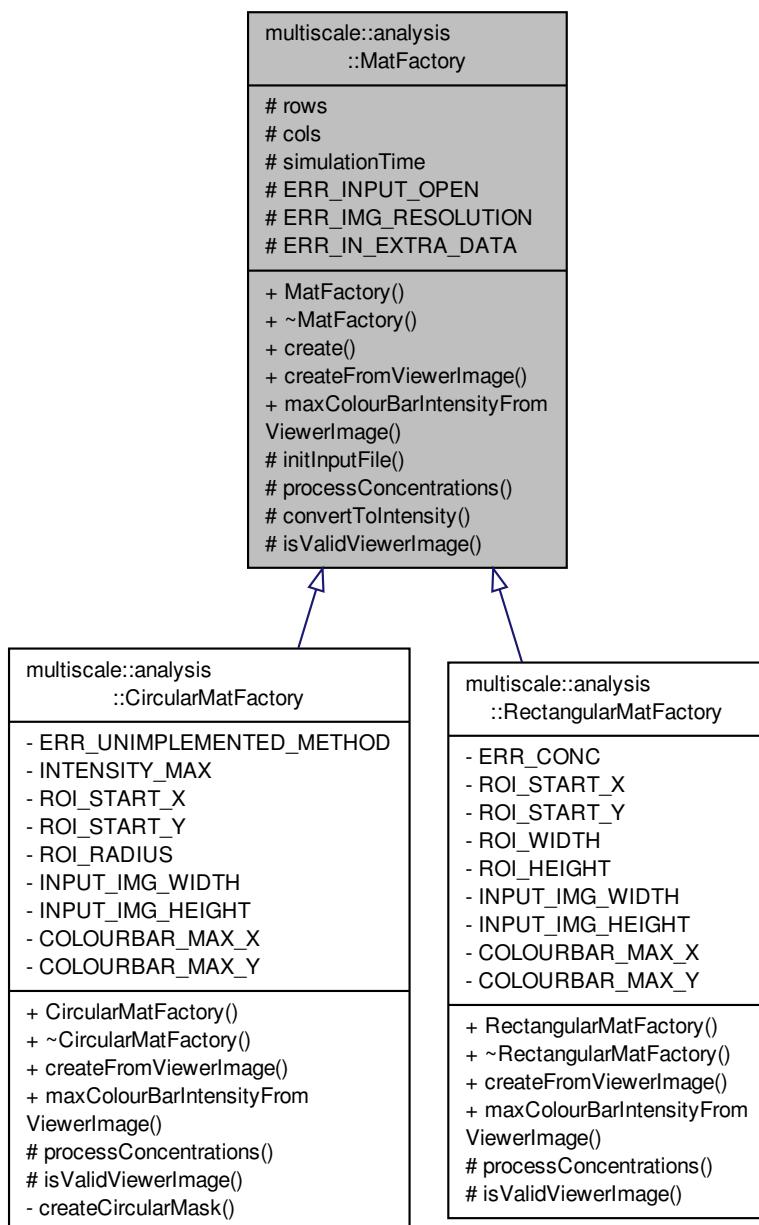
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[LogicPropertyVisitor.hpp](#)

## 7.94 multiscale::analysis::MatFactory Class Reference

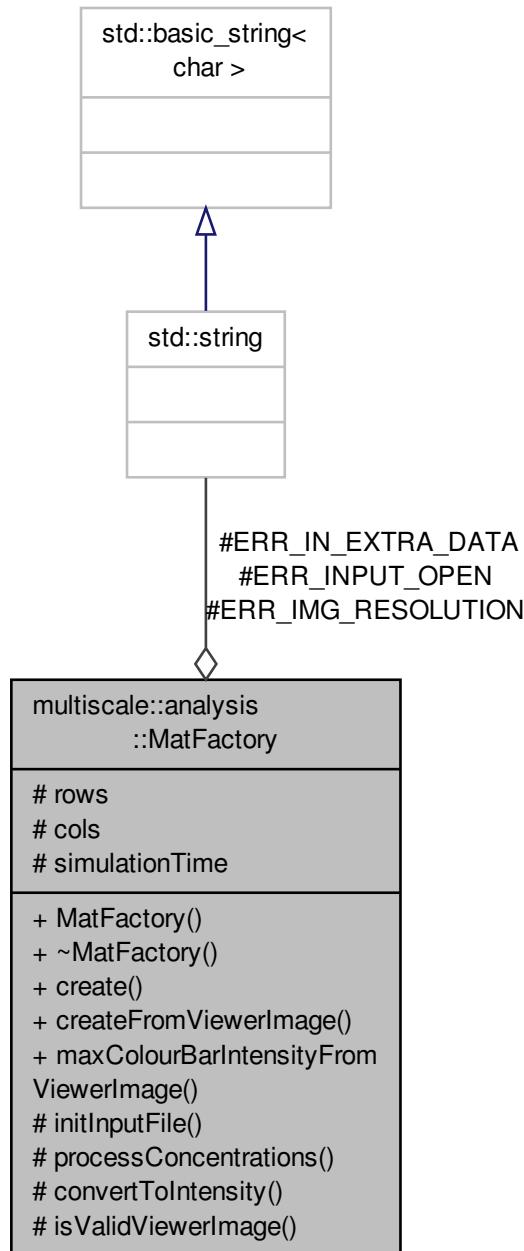
Class for creating a cv::Mat object.

```
#include <MatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::MatFactory:



Collaboration diagram for multiscale::analysis::MatFactory:



## Public Member Functions

- [MatFactory \(\)](#)
- virtual [~MatFactory \(\)](#)
- [cv::Mat create \(const std::string &inputFile\)](#)

*Create a cv::Mat object from the input file.*
- virtual [cv::Mat createFromViewerImage \(const std::string &inputFile\)=0](#)

*Create a Mat object from the image file obtained from Rectangular/CircularGeometryViewer.*

- virtual double `maxColourBarIntensityFromViewerImage` (const std::string &inputFile)=0  
*Get the maximum grayscale intensity of the colour bar in the image.*

## Protected Member Functions

- void `initInputFile` (std::ifstream &fin, const std::string &inputFile)  
*Initialise the input file.*
- virtual unsigned char \* `processConcentrations` (std::ifstream &fin)=0  
*Process concentrations from file.*
- unsigned char `convertToIntensity` (double concentration)  
*Convert concentration to intensity.*
- virtual bool `isValidViewerImage` (const cv::Mat &image)=0  
*Check if the image generated by the viewer has the required resolution.*

## Protected Attributes

- int `rows`
- int `cols`
- double `simulationTime`

## Static Protected Attributes

- static const std::string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const std::string `ERR_IMG_RESOLUTION` = "The resolution of the input image is not the expected one."
- static const std::string `ERR_IN_EXTRA_DATA` = "The input file contains more data than required."

### 7.94.1 Detailed Description

Class for creating a cv::Mat object.

Definition at line 14 of file MatFactory.hpp.

### 7.94.2 Constructor & Destructor Documentation

#### 7.94.2.1 MatFactory::MatFactory( )

Definition at line 10 of file MatFactory.cpp.

#### 7.94.2.2 MatFactory::~MatFactory( ) [virtual]

Definition at line 12 of file MatFactory.cpp.

### 7.94.3 Member Function Documentation

#### 7.94.3.1 unsigned char MatFactory::convertToIntensity( double concentration ) [protected]

Convert concentration to intensity.

Convert the concentration (real value between 0 and 1) to intensity (integer value between 0 and 255)

**Parameters**

|                      |                         |
|----------------------|-------------------------|
| <i>concentration</i> | A value between 0 and 1 |
|----------------------|-------------------------|

Definition at line 44 of file MatFactory.cpp.

Referenced by multiscale::analysis::RectangularMatFactory::processConcentrations().

**7.94.3.2 cv::Mat MatFactory::create ( const std::string & *inputFile* )**

Create a cv::Mat object from the input file.

Create the cv::Mat instance from the values given in the input file

**FORMAT OF INPUT FILE:**

- 1st line contains two positive integers and a real value: nr\_rows, nr\_cols and simulation\_time
- 2nd - (nr\_rows + 1)th lines contain the concentrations of the positions in the grid

**Parameters**

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the input file |
|------------------|----------------------------|

Definition at line 14 of file MatFactory.cpp.

References cols, ERR\_IN\_EXTRA\_DATA, initInputFile(), MS\_throw, processConcentrations(), and rows.

**7.94.3.3 virtual cv::Mat multiscale::analysis::MatFactory::createFromViewerImage ( const std::string & *inputFile* ) [pure virtual]**

Create a Mat object from the image file obtained from Rectangular/CircularGeometryViewer.

Create the Mat instance from the given image file

**Parameters**

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implemented in [multiscale::analysis::CircularMatFactory](#), and [multiscale::analysis::RectangularMatFactory](#).

**7.94.3.4 void MatFactory::initInputFile ( std::ifstream & *fin*, const std::string & *inputFile* ) [protected]**

Initialise the input file.

Initialise the input file. Open an input file stream to the given input file path.

**Parameters**

|                  |                                                      |
|------------------|------------------------------------------------------|
| <i>fin</i>       | An input stream for reading data from the input file |
| <i>inputFile</i> | The path to the input file                           |

Definition at line 34 of file MatFactory.cpp.

References cols, ERR\_INPUT\_OPEN, MS\_throw, rows, and simulationTime.

Referenced by create().

**7.94.3.5 virtual bool multiscale::analysis::MatFactory::isValidViewerImage ( const cv::Mat & *image* ) [protected], [pure virtual]**

Check if the image generated by the viewer has the required resolution.

**Parameters**

|              |                               |
|--------------|-------------------------------|
| <i>image</i> | Image generated by the viewer |
|--------------|-------------------------------|

Implemented in [multiscale::analysis::RectangularMatFactory](#), and [multiscale::analysis::CircularMatFactory](#).

**7.94.3.6 virtual double multiscale::analysis::MatFactory::maxColourBarIntensityFromViewerImage ( const std::string & *inputFile* ) [pure virtual]**

Get the maximum grayscale intensity of the colour bar in the image.

**Parameters**

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implemented in [multiscale::analysis::CircularMatFactory](#), and [multiscale::analysis::RectangularMatFactory](#).

**7.94.3.7 virtual unsigned char\* multiscale::analysis::MatFactory::processConcentrations ( std::ifstream & *fin* ) [protected], [pure virtual]**

Process concentrations from file.

Process the concentrations from the file. This method will be implemented only by subclasses of this abstract class

Implemented in [multiscale::analysis::RectangularMatFactory](#), and [multiscale::analysis::CircularMatFactory](#).

Referenced by [create\(\)](#).

## 7.94.4 Member Data Documentation

**7.94.4.1 int multiscale::analysis::MatFactory::cols [protected]**

Number of columns in the Mat object

Definition at line 19 of file [MatFactory.hpp](#).

Referenced by [create\(\)](#), [initInputFile\(\)](#), and [multiscale::analysis::RectangularMatFactory::processConcentrations\(\)](#).

**7.94.4.2 const std::string MatFactory::ERR\_IMG\_RESOLUTION = "The resolution of the input image is not the expected one." [static], [protected]**

Definition at line 89 of file [MatFactory.hpp](#).

Referenced by [multiscale::analysis::CircularMatFactory::isValidViewerImage\(\)](#), and [multiscale::analysis::RectangularMatFactory::isValidViewerImage\(\)](#).

**7.94.4.3 const std::string MatFactory::ERR\_IN\_EXTRA\_DATA = "The input file contains more data than required." [static], [protected]**

Definition at line 90 of file [MatFactory.hpp](#).

Referenced by [create\(\)](#).

**7.94.4.4 const std::string MatFactory::ERR\_INPUT\_OPEN = "The input file could not be opened." [static], [protected]**

Definition at line 88 of file [MatFactory.hpp](#).

Referenced by [initInputFile\(\)](#), [multiscale::analysis::CircularMatFactory::isValidViewerImage\(\)](#), and [multiscale::analysis::RectangularMatFactory::isValidViewerImage\(\)](#).

**7.94.4.5 int multiscale::analysis::MatFactory::rows [protected]**

Number of rows in the Mat object

Definition at line 18 of file MatFactory.hpp.

Referenced by create(), initInputFile(), and multiscale::analysis::RectangularMatFactory::processConcentrations().

**7.94.4.6 double multiscale::analysis::MatFactory::simulationTime [protected]**

Simulation time read from the input file

Definition at line 20 of file MatFactory.hpp.

Referenced by initInputFile().

The documentation for this class was generated from the following files:

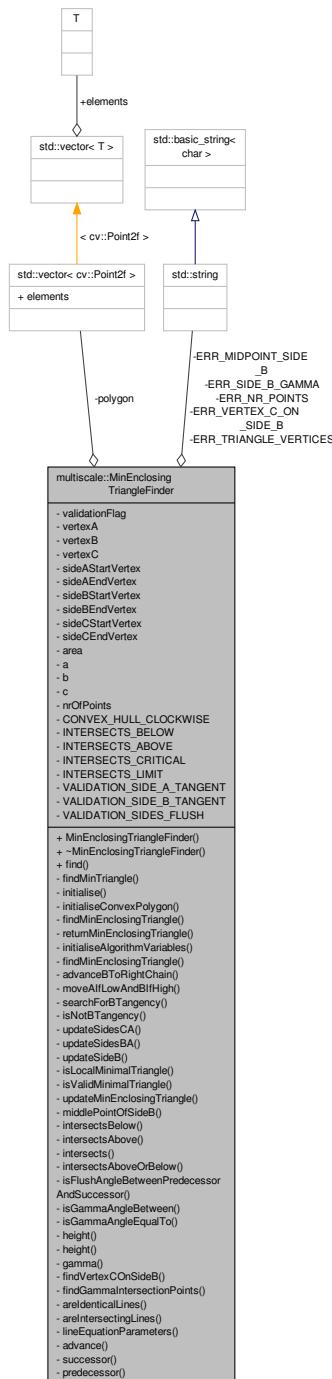
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[MatFactory.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[MatFactory.cpp](#)

## 7.95 multiscale::MinEnclosingTriangleFinder Class Reference

Class for computing the minimum area enclosing triangle for a given polygon.

```
#include <MinEnclosingTriangleFinder.hpp>
```

Collaboration diagram for multiscale::MinEnclosingTriangleFinder:



## Public Member Functions

- [MinEnclosingTriangleFinder \(\)](#)
- [~MinEnclosingTriangleFinder \(\)](#)
- [double find \(const std::vector< cv::Point2f > &points, std::vector< cv::Point2f > &minEnclosingTriangle\)](#)

*Find the minimum area enclosing triangle for the given 2D point set.*

## Private Member Functions

- double **findMinTriangle** (const std::vector< cv::Point2f > &points, std::vector< cv::Point2f > &minEnclosingTriangle)
 

*Find the minimum area enclosing triangle for the given 2D point set.*
- void **initialise** (const std::vector< cv::Point2f > &points, std::vector< cv::Point2f > &minEnclosingTriangle)
 

*Initialisation function for the class.*
- void **initialiseConvexPolygon** (const std::vector< cv::Point2f > &points)
 

*Initialise polygon as the convex hull of the given set of points.*
- double **findMinEnclosingTriangle** (const std::vector< cv::Point2f > &polygon, std::vector< cv::Point2f > &minEnclosingTriangle)
 

*Find the minimum area enclosing triangle for the given polygon.*
- double **returnMinEnclosingTriangle** (const std::vector< cv::Point2f > &polygon, std::vector< cv::Point2f > &minEnclosingTriangle)
 

*Return the minimum area enclosing triangle in case the given polygon has at most three points.*
- void **initialiseAlgorithmVariables** ()
 

*Initialisation of the algorithm variables.*
- void **findMinEnclosingTriangle** (std::vector< cv::Point2f > &minEnclosingTriangle, double &minEnclosingTriangleArea)
 

*Find the minimum area enclosing triangle for the given polygon.*
- void **advanceBToRightChain** ()
 

*Advance b to the right chain.*
- void **moveAIfLowAndBIfHigh** ()
 

*Move "a" if it is low and "b" if it is high.*
- void **searchForBTangency** ()
 

*Search for the tangency of side B.*
- bool **isNotBTangency** ()
 

*Check if tangency for side B was not obtained.*
- void **updateSidesCA** ()
 

*Update sides A and C.*
- void **updateSidesBA** ()
 

*Update sides B and possibly A if tangency for side B was not obtained.*
- void **updateSideB** ()
 

*Set side B if tangency for side B was obtained.*
- bool **isLocalMinimalTriangle** ()
 

*Update the triangle vertices after all sides were set and check if a local minimal triangle was found.*
- bool **isValidMinimalTriangle** ()
 

*Check if the found minimal triangle is valid.*
- void **updateMinEnclosingTriangle** (std::vector< cv::Point2f > &minEnclosingTriangle, double &minEnclosingTriangleArea)
 

*Update the current minimum area enclosing triangle if the newly obtained one has a smaller area.*
- bool **middlePointOfSideB** (cv::Point2f &middlePointOfSideB)
 

*Return the middle point of side B.*
- bool **intersectsBelow** (const cv::Point2f &gammaPoint, unsigned int polygonPointIndex)
 

*Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon below.*
- bool **intersectsAbove** (const cv::Point2f &gammaPoint, unsigned int polygonPointIndex)
 

*Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon above.*
- unsigned int **intersects** (double angleOfGammaAndPoint, unsigned int polygonPointIndex)
 

*Check if/where the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon.*
- unsigned int **intersectsAboveOrBelow** (unsigned int successorOrPredecessorIndex, unsigned int pointIndex)
 

*If (gamma(x) x) intersects P between successorOrPredecessorIndex and pointIndex is it above/below?*

- bool `isFlushAngleBetweenPredecessorAndSuccessor` (double &angleFlushEdge, double anglePredecessor, double angleSuccessor)
 

*Check if the flush edge (opposite) angle lies between the predecessor and successor angle.*
- bool `isGammaAngleBetween` (double &gammaAngle, double angle1, double angle2)
 

*Check if the angle of the line (gamma(p) p) or its opposite angle lie between angle1 and angle2.*
- bool `isGammaAngleEqualTo` (double &gammaAngle, double angle)
 

*Check if the angle of the line (gamma(p) p) or its opposite angle is equal to the given angle.*
- double `height` (unsigned int polygonPointIndex)
 

*Compute the height of the point specified by the given index.*
- double `height` (const cv::Point2f &polygonPoint)
 

*Compute the height of the point.*
- bool `gamma` (unsigned int polygonPointIndex, cv::Point2f &gammaPoint)
 

*Find gamma for a given point "p" specified by its index.*
- cv::Point2f `findVertexCOnSideB` ()
 

*Find vertex C which lies on side B at a distance = 2 \* height(a-1) from side C.*
- bool `findGammaIntersectionPoints` (unsigned int polygonPointIndex, const cv::Point2f &side1StartVertex, const cv::Point2f &side1EndVertex, const cv::Point2f &side2StartVertex, const cv::Point2f &side2EndVertex, cv::Point2f &intersectionPoint1, cv::Point2f &intersectionPoint2)
 

*Find the intersection points to compute gamma(point)*
- bool `areIdenticalLines` (const std::vector< double > &side1Params, const std::vector< double > &side2Params, double sideCEExtraParam)
 

*Check if the given lines are identical or not.*
- bool `areIntersectingLines` (const std::vector< double > &side1Params, const std::vector< double > &side2Params, double sideCEExtraParam, cv::Point2f &intersectionPoint1, cv::Point2f &intersectionPoint2)
 

*Check if the given lines intersect or not. If the lines intersect find their intersection points.*
- std::vector< double > `lineEquationParameters` (const cv::Point2f &p, const cv::Point2f &q)
 

*Get the line equation parameters "a", "b" and "c" for the line determined by points "p" and "q".*
- void `advance` (unsigned int &index)
 

*Advance the given index with one position.*
- unsigned int `successor` (unsigned int index)
 

*Return the succesor of the provided point index.*
- unsigned int `predecessor` (unsigned int index)
 

*Return the predecessor of the provided point index.*

## Private Attributes

- unsigned int `validationFlag`
- cv::Point2f `vertexA`
- cv::Point2f `vertexB`
- cv::Point2f `vertexC`
- cv::Point2f `sideAStartVertex`
- cv::Point2f `sideAEndVertex`
- cv::Point2f `sideBStartVertex`
- cv::Point2f `sideBEndVertex`
- cv::Point2f `sideCStartVertex`
- cv::Point2f `sideCEndVertex`
- double `area`
- unsigned int `a`
- unsigned int `b`
- unsigned int `c`
- unsigned int `nrOfPoints`
- std::vector< cv::Point2f > `polygon`

## Static Private Attributes

- static const bool `CONVEX_HULL_CLOCKWISE` = true
- static const unsigned int `INTERSECTS_BELOW` = 1
- static const unsigned int `INTERSECTS_ABOVE` = 2
- static const unsigned int `INTERSECTS_CRITICAL` = 3
- static const unsigned int `INTERSECTS_LIMIT` = 4
- static const std::string `ERR_NR_POINTS` = "The number of 2D points in the input std::vector should be greater than 0."
- static const std::string `ERR_MIDPOINT_SIDE_B` = "The position of the middle point of side B could not be determined."
- static const std::string `ERR_SIDE_B_GAMMA` = "The position of side B could not be determined, because `gamma(b)` could not be computed."
- static const std::string `ERR_VERTEX_C_ON_SIDE_B` = "The position of the vertex C on side B could not be determined, because the considered lines do not intersect."
- static const std::string `ERR_TRIANGLE_VERTICES` = "The position of the triangle vertices could not be determined, because the sides of the triangle do not intersect."
- static const unsigned int `VALIDATION_SIDE_A_TANGENT` = 0
- static const unsigned int `VALIDATION_SIDE_B_TANGENT` = 1
- static const unsigned int `VALIDATION_SIDES_FLUSH` = 2

### 7.95.1 Detailed Description

Class for computing the minimum area enclosing triangle for a given polygon.

This implementation has a linear complexity ( $\theta(n)$ ) with respect to the number of points defining the convex polygon and is based on the algorithm described in the following paper:

J. O'Rourke, A. Aggarwal, S. Maddila, and M. Baldwin, 'An optimal algorithm for finding minimal enclosing triangles', Journal of Algorithms, vol. 7, no. 2, pp. 258–269, Jun. 1986.

Definition at line 19 of file MinEnclosingTriangleFinder.hpp.

### 7.95.2 Constructor & Destructor Documentation

#### 7.95.2.1 MinEnclosingTriangleFinder::MinEnclosingTriangleFinder( )

Definition at line 13 of file MinEnclosingTriangleFinder.cpp.

References a, area, b, c, nrOfPoints, and validationFlag.

#### 7.95.2.2 MinEnclosingTriangleFinder::~MinEnclosingTriangleFinder( )

Definition at line 25 of file MinEnclosingTriangleFinder.cpp.

### 7.95.3 Member Function Documentation

#### 7.95.3.1 void MinEnclosingTriangleFinder::advance( unsigned int & index ) [private]

Advance the given index with one position.

**Parameters**

|              |                    |
|--------------|--------------------|
| <i>index</i> | Index of the point |
|--------------|--------------------|

Definition at line 440 of file MinEnclosingTriangleFinder.cpp.

References successor().

Referenced by advanceBToRightChain(), moveAlfLowAndBlfHigh(), and searchForBTangency().

#### 7.95.3.2 void MinEnclosingTriangleFinder::advanceBToRightChain ( ) [private]

Advance b to the right chain.

See paper for more details

Definition at line 115 of file MinEnclosingTriangleFinder.cpp.

References advance(), b, multiscale::Numeric::greaterOrEqual(), height(), and successor().

Referenced by findMinEnclosingTriangle().

#### 7.95.3.3 bool MinEnclosingTriangleFinder::areIdenticalLines ( const std::vector< double > & side1Params, const std::vector< double > & side2Params, double sideCExtraParam ) [private]

Check if the given lines are identical or not.

The lines are specified as:  $ax + by + c = 0$  OR  $ax + by + c (+/-) \text{sideCExtraParam} = 0$

Parameters

|                        |                                                       |
|------------------------|-------------------------------------------------------|
| <i>side1Params</i>     | Vector containing the values of a, b and c for side 1 |
| <i>side2Params</i>     | Vector containing the values of a, b and c for side 2 |
| <i>sideCExtraParam</i> | Extra parameter for the flush edge C                  |

Definition at line 400 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::areIdenticalLines().

Referenced by findGammaIntersectionPoints().

#### 7.95.3.4 bool MinEnclosingTriangleFinder::areIntersectingLines ( const std::vector< double > & side1Params, const std::vector< double > & side2Params, double sideCExtraParam, cv::Point2f & intersectionPoint1, cv::Point2f & intersectionPoint2 ) [private]

Check if the given lines intersect or not. If the lines intersect find their intersection points.

The lines are specified as:  $ax + by + c = 0$  OR  $ax + by + c (+/-) \text{sideCExtraParam} = 0$

Parameters

|                           |                                                       |
|---------------------------|-------------------------------------------------------|
| <i>side1Params</i>        | Vector containing the values of a, b and c for side 1 |
| <i>side2Params</i>        | Vector containing the values of a, b and c for side 2 |
| <i>sideCExtraParam</i>    | Extra parameter for the flush edge C                  |
| <i>intersectionPoint1</i> | The first intersection point, if it exists            |
| <i>intersectionPoint2</i> | The second intersection point, if it exists           |

Definition at line 411 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::lineIntersection().

Referenced by findGammaIntersectionPoints().

7.95.3.5 double MinEnclosingTriangleFinder::find ( const std::vector< cv::Point2f > & *points*, std::vector< cv::Point2f > & *minEnclosingTriangle* )

Find the minimum area enclosing triangle for the given 2D point set.

Precondition: Number of points in the set is at least 1.

#### Parameters

|                             |                                                   |
|-----------------------------|---------------------------------------------------|
| <i>points</i>               | Set of points                                     |
| <i>minEnclosingTriangle</i> | Minimum area triangle enclosing the given polygon |

Definition at line 27 of file MinEnclosingTriangleFinder.cpp.

References ERR\_NR\_POINTS, findMinTriangle(), and MS\_throw.

Referenced by multiscale::analysis::Cluster::isTriangularMeasure(), multiscale::analysis::Region::isTriangularMeasure(), runMinEnclosingTriangleFinder(), and multiscaletest::MinEnclosingTriangleFinderTest::RunTest().

7.95.3.6 bool MinEnclosingTriangleFinder::findGammaIntersectionPoints ( unsigned int *polygonPointIndex*, const cv::Point2f & *side1StartVertex*, const cv::Point2f & *side1EndVertex*, const cv::Point2f & *side2StartVertex*, const cv::Point2f & *side2EndVertex*, cv::Point2f & *intersectionPoint1*, cv::Point2f & *intersectionPoint2* ) [private]

Find the intersection points to compute gamma(point)

#### Parameters

|                           |                                                            |
|---------------------------|------------------------------------------------------------|
| <i>polygonPointIndex</i>  | Index of the polygon point for which the distance is known |
| <i>side1StartVertex</i>   | Start vertex for side 1                                    |
| <i>side1EndVertex</i>     | End vertex for side 1                                      |
| <i>side2StartVertex</i>   | Start vertex for side 2                                    |
| <i>side2EndVertex</i>     | End vertex for side 2                                      |
| <i>intersectionPoint1</i> | First intersection point between one pair of lines         |
| <i>intersectionPoint2</i> | Second intersection point between another pair of lines    |

Definition at line 374 of file MinEnclosingTriangleFinder.cpp.

References areIdenticalLines(), areIntersectingLines(), height(), and lineEquationParameters().

Referenced by findVertexCOnSideB(), and gamma().

7.95.3.7 double MinEnclosingTriangleFinder::findMinEnclosingTriangle ( const std::vector< cv::Point2f > & *polygon*, std::vector< cv::Point2f > & *minEnclosingTriangle* ) [private]

Find the minimum area enclosing triangle for the given polygon.

#### Parameters

|                             |                                                                               |
|-----------------------------|-------------------------------------------------------------------------------|
| <i>polygon</i>              | Polygon of points for which the minimum area enclosing triangle will be found |
| <i>minEnclosingTriangle</i> | Minimum area triangle enclosing the given polygon                             |

Definition at line 64 of file MinEnclosingTriangleFinder.cpp.

References initialiseAlgorithmVariables().

Referenced by findMinTriangle().

7.95.3.8 void MinEnclosingTriangleFinder::findMinEnclosingTriangle ( std::vector< cv::Point2f > & *minEnclosingTriangle*,  
double & *minEnclosingTriangleArea* ) [private]

Find the minimum area enclosing triangle for the given polygon.

## Parameters

|                                 |                                                   |
|---------------------------------|---------------------------------------------------|
| <i>minEnclosingTriangle</i>     | Minimum area triangle enclosing the given polygon |
| <i>minEnclosingTriangleArea</i> | Area of the minimum area enclosing triangle       |

Definition at line 94 of file MinEnclosingTriangleFinder.cpp.

References advanceBToRightChain(), c, isLocalMinimalTriangle(), isNotBTangency(), moveAIfLowAndBIfHigh(), nrOfPoints, searchForBTangency(), updateMinEnclosingTriangle(), updateSideB(), updateSidesBA(), and updateSidesCA().

#### 7.95.3.9 double MinEnclosingTriangleFinder::findMinTriangle ( const std::vector< cv::Point2f > & *points*, std::vector< cv::Point2f > & *minEnclosingTriangle* ) [private]

Find the minimum area enclosing triangle for the given 2D point set.

## Parameters

|                             |                                                   |
|-----------------------------|---------------------------------------------------|
| <i>points</i>               | Set of points                                     |
| <i>minEnclosingTriangle</i> | Minimum area triangle enclosing the given polygon |

Definition at line 39 of file MinEnclosingTriangleFinder.cpp.

References findMinEnclosingTriangle(), initialise(), polygon, and returnMinEnclosingTriangle().

Referenced by find().

#### 7.95.3.10 cv::Point2f MinEnclosingTriangleFinder::findVertexCOnSideB ( ) [private]

Find vertex C which lies on side B at a distance =  $2 * \text{height}(a-1)$  from side C.

Considering that line (x y) is a line parallel to (c c-1) and that the distance between the lines is equal to  $2 * \text{height}(a-1)$ , we can have two possible (x y) lines.

Therefore, we will compute two intersection points between the lines (x y) and (b b-1) and take the point which is closest to point polygon[b].

See paper and formula for distance from point to a line for more details

Definition at line 356 of file MinEnclosingTriangleFinder.cpp.

References a, multiscale::Geometry2D::areOnTheSameSideOfLine(), c, ERR\_VERTEX\_C\_ON\_SIDE\_B, findGammaIntersectionPoints(), MS\_throw, polygon, predecessor(), sideBEndVertex, sideBStartVertex, sideCEndVertex, sideCStartVertex, and successor().

Referenced by updateSidesBA().

#### 7.95.3.11 bool MinEnclosingTriangleFinder::gamma ( unsigned int *polygonPointIndex*, cv::Point2f & *gammaPoint* ) [private]

Find gamma for a given point "p" specified by its index.

The function returns true if gamma exists i.e. if lines (a a-1) and (x y) intersect and false otherwise. In case the two lines intersect in point intersectionPoint, gamma is computed.

Considering that line (x y) is a line parallel to (c c-1) and that the distance between the lines is equal to  $2 * \text{height}(p)$ , we can have two possible (x y) lines.

Therefore, we will compute two intersection points between the lines (x y) and (a a-1) and take the point which is closest to point polygon[a].

See paper and formula for distance from point to a line for more details

**Parameters**

|                          |                                               |
|--------------------------|-----------------------------------------------|
| <i>polygonPointIndex</i> | Index of the polygon point                    |
| <i>gammaPoint</i>        | cv::Point2f gamma(polygon[polygonPointIndex]) |

Definition at line 336 of file MinEnclosingTriangleFinder.cpp.

References a, multiscale::Geometry2D::areOnTheSameSideOfLine(), c, findGammaIntersectionPoints(), polygon, predecessor(), and successor().

Referenced by isNotBTangency(), moveAIfLowAndBIfHigh(), searchForBTangency(), and updateSideB().

#### 7.95.3.12 double MinEnclosingTriangleFinder::height ( unsigned int *polygonPointIndex* ) [private]

Compute the height of the point specified by the given index.

See paper for more details

**Parameters**

|                          |                            |
|--------------------------|----------------------------|
| <i>polygonPointIndex</i> | Index of the polygon point |
|--------------------------|----------------------------|

Definition at line 320 of file MinEnclosingTriangleFinder.cpp.

References c, multiscale::Geometry2D::distanceFromPointToLine(), polygon, and predecessor().

Referenced by advanceBToLeftChain(), findGammaIntersectionPoints(), intersectsAboveOrBelow(), isNotB←Tangency(), moveAIfLowAndBIfHigh(), searchForBTangency(), and updateSidesBA().

#### 7.95.3.13 double MinEnclosingTriangleFinder::height ( const cv::Point2f & *polygonPoint* ) [private]

Compute the height of the point.

See paper for more details

**Parameters**

|                     |               |
|---------------------|---------------|
| <i>polygonPoint</i> | Polygon point |
|---------------------|---------------|

Definition at line 329 of file MinEnclosingTriangleFinder.cpp.

References c, multiscale::Geometry2D::distanceFromPointToLine(), polygon, and predecessor().

#### 7.95.3.14 void MinEnclosingTriangleFinder::initialise ( const std::vector< cv::Point2f > & *points*, std::vector< cv::Point2f > & *minEnclosingTriangle* ) [private]

Initialisation function for the class.

Initialise the polygon and other class' fields.

**Parameters**

|                             |                                                   |
|-----------------------------|---------------------------------------------------|
| <i>points</i>               | Set of points                                     |
| <i>minEnclosingTriangle</i> | Minimum area triangle enclosing the given polygon |

Definition at line 50 of file MinEnclosingTriangleFinder.cpp.

References initialiseConvexPolygon().

Referenced by findMinTriangle().

## 7.95.3.15 void MinEnclosingTriangleFinder::initialiseAlgorithmVariables ( ) [private]

Initialisation of the algorithm variables.

Definition at line 86 of file MinEnclosingTriangleFinder.cpp.

References a, b, c, nrOfPoints, and polygon.

Referenced by findMinEnclosingTriangle().

## 7.95.3.16 void MinEnclosingTriangleFinder::initialiseConvexPolygon ( const std::vector&lt; cv::Point2f &gt; &amp; points ) [private]

Initialise polygon as the convex hull of the given set of points.

Parameters

|               |               |
|---------------|---------------|
| <i>points</i> | Set of points |
|---------------|---------------|

Definition at line 58 of file MinEnclosingTriangleFinder.cpp.

References CONVEX\_HULL\_CLOCKWISE, and polygon.

Referenced by initialise().

## 7.95.3.17 unsigned int MinEnclosingTriangleFinder::intersects ( double angleOfGammaAndPoint, unsigned int polygonPointIndex ) [private]

Check if/where the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon.

Parameters

|                             |                                                                          |
|-----------------------------|--------------------------------------------------------------------------|
| <i>angleOfGammaAndPoint</i> | Angle between gammaPoint and polygon[polygonPointIndex]                  |
| <i>polygonPointIndex</i>    | Index of the polygon point which is considered when determining the line |

Definition at line 256 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Numeric::almostEqual(), multiscale::Geometry2D::angleOfLineWrtOxAxis(), c, INTERSECTS\_BELOW, INTERSECTS\_CRITICAL, intersectsAboveOrBelow(), isFlushAngleBetweenPredecessorAndSuccessor(), isGammaAngleBetween(), isGammaAngleEqualTo(), polygon, predecessor(), and successor().

Referenced by intersectsAbove(), and intersectsBelow().

## 7.95.3.18 bool MinEnclosingTriangleFinder::intersectsAbove ( const cv::Point2f &amp; gammaPoint, unsigned int polygonPointIndex ) [private]

Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon above.

Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon above the point polygon[polygonPointIndex]

Parameters

|                          |                                                                          |
|--------------------------|--------------------------------------------------------------------------|
| <i>gammaPoint</i>        | Gamma(p)                                                                 |
| <i>polygonPointIndex</i> | Index of the polygon point which is considered when determining the line |

Definition at line 250 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::angleOfLineWrtOxAxis(), intersects(), INTERSECTS\_ABOVE, and polygon.

Referenced by isNotBTangency().

7.95.3.19 `unsigned int MinEnclosingTriangleFinder::intersectsAboveOrBelow ( unsigned int successorOrPredecessorIndex, unsigned int pointIndex ) [private]`

If ( $\text{gamma}(x)$  x) intersects P between successorOrPredecessorIndex and pointIndex is it above/below?

#### Parameters

|                                          |                                       |
|------------------------------------------|---------------------------------------|
| <code>successorOrPredecessorIndex</code> | Index of the successor or predecessor |
| <code>pointIndex</code>                  | Index of the point x in the polygon   |

Definition at line 289 of file MinEnclosingTriangleFinder.cpp.

References height(), INTERSECTS\_ABOVE, and INTERSECTS\_BELOW.

Referenced by intersects().

7.95.3.20 `bool MinEnclosingTriangleFinder::intersectsBelow ( const cv::Point2f & gammaPoint, unsigned int polygonPointIndex ) [private]`

Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon below.

Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon below the point polygon[polygonPointIndex]

#### Parameters

|                                |                                                                          |
|--------------------------------|--------------------------------------------------------------------------|
| <code>gammaPoint</code>        | Gamma(p)                                                                 |
| <code>polygonPointIndex</code> | Index of the polygon point which is considered when determining the line |

Definition at line 244 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::angleOfLineWrtOxAxis(), intersects(), INTERSECTS\_BELOW, and polygon.

Referenced by moveAlfLowAndBifHigh(), and searchForBTangency().

7.95.3.21 `bool MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor ( double & angleFlushEdge, double anglePredecessor, double angleSuccessor ) [private]`

Check if the flush edge (opposite) angle lies between the predecessor and successor angle.

#### Parameters

|                               |                          |
|-------------------------------|--------------------------|
| <code>angleFlushEdge</code>   | Angle of the flush edge  |
| <code>anglePredecessor</code> | Angle of the predecessor |
| <code>angleSuccessor</code>   | Angle of the successor   |

Definition at line 298 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::isAngleBetweenNonReflex(), multiscale::Geometry2D::isOppositeAngleBetweenNonReflex(), and multiscale::Geometry2D::oppositeAngle().

Referenced by intersects().

7.95.3.22 `bool MinEnclosingTriangleFinder::isGammaAngleBetween ( double & gammaAngle, double angle1, double angle2 ) [private]`

Check if the angle of the line ( $\text{gamma}(p)$  p) or its opposite angle lie between angle1 and angle2.

**Parameters**

|                   |                                             |
|-------------------|---------------------------------------------|
| <i>gammaAngle</i> | Angle of the line ( $\text{gamma}(p)$ $p$ ) |
| <i>angle1</i>     | One of the boundary angles                  |
| <i>angle2</i>     | Another boundary angle                      |

Definition at line 312 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::isAngleBetweenNonReflex().

Referenced by intersects().

**7.95.3.23 bool MinEnclosingTriangleFinder::isGammaAngleEqualTo ( double & *gammaAngle*, double *angle* ) [private]**

Check if the angle of the line ( $\text{gamma}(p)$   $p$ ) or its opposite angle is equal to the given angle.

**Parameters**

|                   |                                             |
|-------------------|---------------------------------------------|
| <i>gammaAngle</i> | Angle of the line ( $\text{gamma}(p)$ $p$ ) |
| <i>angle</i>      | Angle to compare against                    |

Definition at line 316 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Numeric::almostEqual().

Referenced by intersects().

**7.95.3.24 bool MinEnclosingTriangleFinder::isLocalMinimalTriangle ( ) [private]**

Update the triangle vertices after all sides were set and check if a local minimal triangle was found.

See paper for more details

Definition at line 188 of file MinEnclosingTriangleFinder.cpp.

References isValidMinimalTriangle(), multiscale::Geometry2D::lineIntersection(), sideAEndVertex, sideAStartVertex, sideBEndVertex, sideBStartVertex, sideCEndVertex, sideCStartVertex, vertexA, vertexB, and vertexC.

Referenced by findMinEnclosingTriangle().

**7.95.3.25 bool MinEnclosingTriangleFinder::isNotBTangency ( ) [private]**

Check if tangency for side B was not obtained.

See paper for more details

Definition at line 142 of file MinEnclosingTriangleFinder.cpp.

References a, b, gamma(), height(), intersectsAbove(), and predecessor().

Referenced by findMinEnclosingTriangle().

**7.95.3.26 bool MinEnclosingTriangleFinder::isValidMinimalTriangle ( ) [private]**

Check if the found minimal triangle is valid.

This means that all midpoints of the triangle should touch the polygon

See paper for more details

Definition at line 198 of file MinEnclosingTriangleFinder.cpp.

References a, multiscale::Geometry2D::areEqualPoints(), b, multiscale::Geometry2D::isPointOnLineSegment(), multiscale::Geometry2D::middlePoint(), polygon, predecessor(), sideAEndVertex, sideAStartVertex, sideBEndVertex, sideBStartVertex, sideCEndVertex, sideCStartVertex, VALIDATION\_SIDE\_A\_TANGENT, VALIDATION\_SIDE\_B\_TANGENT, validationFlag, vertexA, vertexB, and vertexC.

Referenced by `isLocalMinimalTriangle()`.

**7.95.3.27** `std::vector< double > MinEnclosingTriangleFinder::lineEquationParameters ( const cv::Point2f & p, const cv::Point2f & q ) [private]`

Get the line equation parameters "a", "b" and "c" for the line determined by points "p" and "q".

The equation of the line is considered in the general form:  $ax + by + c = 0$

#### Parameters

|          |                                                    |
|----------|----------------------------------------------------|
| <i>p</i> | One point for defining the equation of the line    |
| <i>q</i> | Second point for defining the equation of the line |

Definition at line 426 of file `MinEnclosingTriangleFinder.cpp`.

References `a`, `b`, `c`, and `multiscale::Geometry2D::lineEquationDeterminedByPoints()`.

Referenced by `findGammalIntersectionPoints()`.

**7.95.3.28** `bool MinEnclosingTriangleFinder::middlePointOfSideB ( cv::Point2f & middlePointOfSideB ) [private]`

Return the middle point of side B.

Definition at line 231 of file `MinEnclosingTriangleFinder.cpp`.

References `multiscale::Geometry2D::lineIntersection()`, `multiscale::Geometry2D::middlePoint()`, `sideAEndVertex`, `sideAStartVertex`, `sideBEndVertex`, `sideBStartVertex`, `sideCEndVertex`, `sideCStartVertex`, `vertexA`, and `vertexC`.

Referenced by `updateSidesBA()`.

**7.95.3.29** `void MinEnclosingTriangleFinder::moveAIfLowAndBIfHigh ( ) [private]`

Move "a" if it is low and "b" if it is high.

See paper for more details

Definition at line 121 of file `MinEnclosingTriangleFinder.cpp`.

References `a`, `advance()`, `b`, `gamma()`, `height()`, and `intersectsBelow()`.

Referenced by `findMinEnclosingTriangle()`.

**7.95.3.30** `unsigned int MinEnclosingTriangleFinder::predecessor ( unsigned int index ) [private]`

Return the predecessor of the provided point index.

The predecessor of the first polygon point is the last polygon point (circular referencing)

#### Parameters

|              |                    |
|--------------|--------------------|
| <i>index</i> | Index of the point |
|--------------|--------------------|

Definition at line 448 of file `MinEnclosingTriangleFinder.cpp`.

References `nrOfPoints`.

Referenced by `findVertexCOnSideB()`, `gamma()`, `height()`, `intersects()`, `isNotBTangency()`, `isValidMinimalTriangle()`, `searchForBTangency()`, `updateSidesBA()`, and `updateSidesCA()`.

**7.95.3.31** `double MinEnclosingTriangleFinder::returnMinEnclosingTriangle ( const std::vector< cv::Point2f > & polygon, std::vector< cv::Point2f > & minEnclosingTriangle ) [private]`

Return the minimum area enclosing triangle in case the given polygon has at most three points.

**Parameters**

|                             |                                                                               |
|-----------------------------|-------------------------------------------------------------------------------|
| <i>polygon</i>              | Polygon of points for which the minimum area enclosing triangle will be found |
| <i>minEnclosingTriangle</i> | Minimum area triangle enclosing the given polygon                             |

Definition at line 75 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::areaOfTriangle().

Referenced by findMinTriangle().

**7.95.3.32 void MinEnclosingTriangleFinder::searchForBTangency ( ) [private]**

Search for the tangency of side B.

See paper for more details

Definition at line 133 of file MinEnclosingTriangleFinder.cpp.

References a, advance(), b, gamma(), multiscale::Numeric::greaterOrEqual(), height(), intersectsBelow(), and predecessor().

Referenced by findMinEnclosingTriangle().

**7.95.3.33 unsigned int MinEnclosingTriangleFinder::successor ( unsigned int *index* ) [private]**

Return the successor of the provided point index.

The successor of the last polygon point is the first polygon point (circular referencing)

**Parameters**

|              |                    |
|--------------|--------------------|
| <i>index</i> | Index of the point |
|--------------|--------------------|

Definition at line 444 of file MinEnclosingTriangleFinder.cpp.

References nrOfPoints.

Referenced by advance(), advanceBToRightChain(), findVertexCOnSideB(), gamma(), and intersects().

**7.95.3.34 void MinEnclosingTriangleFinder::updateMinEnclosingTriangle ( std::vector< cv::Point2f > & *minEnclosingTriangle*, double & *minEnclosingTriangleArea* ) [private]**

Update the current minimum area enclosing triangle if the newly obtained one has a smaller area.

**Parameters**

|                                 |                                                               |
|---------------------------------|---------------------------------------------------------------|
| <i>minEnclosingTriangle</i>     | Minimum area triangle enclosing the given polygon             |
| <i>minEnclosingTriangleArea</i> | Area of the minimum area triangle enclosing the given polygon |

Definition at line 216 of file MinEnclosingTriangleFinder.cpp.

References area, multiscale::Geometry2D::areaOfTriangle(), vertexA, vertexB, and vertexC.

Referenced by findMinEnclosingTriangle().

**7.95.3.35 void MinEnclosingTriangleFinder::updateSideB ( ) [private]**

Set side B if tangency for side B was obtained.

See paper for more details

Definition at line 178 of file MinEnclosingTriangleFinder.cpp.

References b, ERR\_SIDE\_B\_GAMMA, gamma(), MS\_throw, polygon, sideBEndVertex, sideBStartVertex, VALIDATION\_SIDE\_B\_TANGENT, and validationFlag.

Referenced by findMinEnclosingTriangle().

#### 7.95.3.36 void MinEnclosingTriangleFinder::updateSidesBA( ) [private]

Update sides B and possibly A if tangency for side B was not obtained.

See paper for more details

Definition at line 160 of file MinEnclosingTriangleFinder.cpp.

References a, b, findVertexCOnSideB(), height(), middlePointOfSideB(), polygon, predecessor(), sideAEndVertex, sideAStartVertex, sideBEndVertex, sideBStartVertex, VALIDATION\_SIDE\_A\_TANGENT, VALIDATION\_SIDES\_FLUSH, and validationFlag.

Referenced by findMinEnclosingTriangle().

#### 7.95.3.37 void MinEnclosingTriangleFinder::updateSidesCA( ) [private]

Update sides A and C.

Side C will have as start and end vertices the polygon points "c" and "c-1" Side A will have as start and end vertices the polygon points "a" and "a-1"

Definition at line 152 of file MinEnclosingTriangleFinder.cpp.

References a, c, polygon, predecessor(), sideAEndVertex, sideAStartVertex, sideCEndVertex, and sideCStartVertex.

Referenced by findMinEnclosingTriangle().

### 7.95.4 Member Data Documentation

#### 7.95.4.1 unsigned int multiscale::MinEnclosingTriangleFinder::a [private]

Index of point "a"; see paper for more details

Definition at line 44 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), gamma(), initialiseAlgorithmVariables(), isNotBTangency(), isValidMinimalTriangle(), lineEquationParameters(), MinEnclosingTriangleFinder(), moveAIfLowAndBIfHigh(), searchForBTangency(), updateSidesBA(), and updateSidesCA().

#### 7.95.4.2 double multiscale::MinEnclosingTriangleFinder::area [private]

Area of the current considered enclosing triangle

Definition at line 42 of file MinEnclosingTriangleFinder.hpp.

Referenced by MinEnclosingTriangleFinder(), and updateMinEnclosingTriangle().

#### 7.95.4.3 unsigned int multiscale::MinEnclosingTriangleFinder::b [private]

Index of point "b"; see paper for more details

Definition at line 45 of file MinEnclosingTriangleFinder.hpp.

Referenced by advanceBToRightChain(), initialiseAlgorithmVariables(), isNotBTangency(), isValidMinimalTriangle(), lineEquationParameters(), MinEnclosingTriangleFinder(), moveAIfLowAndBIfHigh(), searchForBTangency(), updateSideB(), and updateSidesBA().

#### 7.95.4.4 `unsigned int multiscale::MinEnclosingTriangleFinder::c` [private]

Index of point "c"; see paper for more details

Definition at line 46 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinEnclosingTriangle(), findVertexCOnSideB(), gamma(), height(), initialiseAlgorithmVariables(), intersects(), lineEquationParameters(), MinEnclosingTriangleFinder(), and updateSidesCA().

#### 7.95.4.5 `const bool MinEnclosingTriangleFinder::CONVEX_HULL_CLOCKWISE = true` [static], [private]

Definition at line 372 of file MinEnclosingTriangleFinder.hpp.

Referenced by initialiseConvexPolygon().

#### 7.95.4.6 `const std::string MinEnclosingTriangleFinder::ERR_MIDPOINT_SIDE_B = "The position of the middle point of side B could not be determined."` [static], [private]

Definition at line 380 of file MinEnclosingTriangleFinder.hpp.

#### 7.95.4.7 `const std::string MinEnclosingTriangleFinder::ERR_NR_POINTS = "The number of 2D points in the input std::vector should be greater than 0."` [static], [private]

Definition at line 379 of file MinEnclosingTriangleFinder.hpp.

Referenced by find().

#### 7.95.4.8 `const std::string MinEnclosingTriangleFinder::ERR_SIDE_B_GAMMA = "The position of side B could not be determined, because gamma(b) could not be computed."` [static], [private]

Definition at line 381 of file MinEnclosingTriangleFinder.hpp.

Referenced by updateSideB().

#### 7.95.4.9 `const std::string MinEnclosingTriangleFinder::ERR_TRIANGLE_VERTICES = "The position of the triangle vertices could not be determined, because the sides of the triangle do not intersect."` [static], [private]

Definition at line 383 of file MinEnclosingTriangleFinder.hpp.

#### 7.95.4.10 `const std::string MinEnclosingTriangleFinder::ERR_VERTEX_C_ON_SIDE_B = "The position of the vertex C on side B could not be determined, because the considered lines do not intersect."` [static], [private]

Definition at line 382 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB().

#### 7.95.4.11 `const unsigned int MinEnclosingTriangleFinder::INTERSECTS_ABOVE = 2` [static], [private]

Definition at line 375 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersectsAbove(), and intersectsAboveOrBelow().

7.95.4.12 `const unsigned int MinEnclosingTriangleFinder::INTERSECTS_BELOW = 1` [static], [private]

Definition at line 374 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersects(), intersectsAboveOrBelow(), and intersectsBelow().

7.95.4.13 `const unsigned int MinEnclosingTriangleFinder::INTERSECTS_CRITICAL = 3` [static], [private]

Definition at line 376 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersects().

7.95.4.14 `const unsigned int MinEnclosingTriangleFinder::INTERSECTS_LIMIT = 4` [static], [private]

Definition at line 377 of file MinEnclosingTriangleFinder.hpp.

7.95.4.15 `unsigned int multiscale::MinEnclosingTriangleFinder::nrOfPoints` [private]

Number of points defining the polygon

Definition at line 48 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinEnclosingTriangle(), initialiseAlgorithmVariables(), MinEnclosingTriangleFinder(), predecessor(), and successor().

7.95.4.16 `std::vector<cv::Point2f> multiscale::MinEnclosingTriangleFinder::polygon` [private]

Polygon for which the minimum area enclosing triangle is computed

Definition at line 50 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinTriangle(), findVertexCOnSideB(), gamma(), height(), initialiseAlgorithmVariables(), initialiseConvexPolygon(), intersects(), intersectsAbove(), intersectsBelow(), isValidMinimalTriangle(), updateSideB(), updateSidesBA(), and updateSidesCA().

7.95.4.17 `cv::Point2f multiscale::MinEnclosingTriangleFinder::sideAEndVertex` [private]

Ending vertex for side A of triangle

Definition at line 34 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSidesBA(), and updateSidesCA().

7.95.4.18 `cv::Point2f multiscale::MinEnclosingTriangleFinder::sideAStartVertex` [private]

Starting vertex for side A of triangle

Definition at line 33 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSidesBA(), and updateSidesCA().

7.95.4.19 `cv::Point2f multiscale::MinEnclosingTriangleFinder::sideBEndVertex` [private]

Ending vertex for side B of triangle

Definition at line 37 of file MinEnclosingTriangleFinder.hpp.

Referenced by `findVertexCOnSideB()`, `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, `updateSideB()`, and `updateSidesBA()`.

#### 7.95.4.20 cv::Point2f multiscale::MinEnclosingTriangleFinder::sideBStartVertex [private]

Starting vertex for side B of triangle

Definition at line 36 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `findVertexCOnSideB()`, `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, `updateSideB()`, and `updateSidesBA()`.

#### 7.95.4.21 cv::Point2f multiscale::MinEnclosingTriangleFinder::sideCEndVertex [private]

Ending vertex for side C of triangle

Definition at line 40 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `findVertexCOnSideB()`, `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, and `updateSidesCA()`.

#### 7.95.4.22 cv::Point2f multiscale::MinEnclosingTriangleFinder::sideCStartVertex [private]

Starting vertex for side C of triangle

Definition at line 39 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `findVertexCOnSideB()`, `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, and `updateSidesCA()`.

#### 7.95.4.23 const unsigned int MinEnclosingTriangleFinder::VALIDATION\_SIDE\_A\_TANGENT = 0 [static], [private]

Definition at line 385 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `isValidMinimalTriangle()`, and `updateSidesBA()`.

#### 7.95.4.24 const unsigned int MinEnclosingTriangleFinder::VALIDATION\_SIDE\_B\_TANGENT = 1 [static], [private]

Definition at line 386 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `isValidMinimalTriangle()`, and `updateSideB()`.

#### 7.95.4.25 const unsigned int MinEnclosingTriangleFinder::VALIDATION\_SIDES\_FLUSH = 2 [static], [private]

Definition at line 387 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `updateSidesBA()`.

#### 7.95.4.26 unsigned int multiscale::MinEnclosingTriangleFinder::validationFlag [private]

Validation flag can take the following values:

- `VALIDATION_SIDE_A_TANGENT`;
- `VALIDATION_SIDE_B_TANGENT`;
- `VALIDATION_SIDES_FLUSH`.

Definition at line 23 of file MinEnclosingTriangleFinder.hpp.

Referenced by isValidMinimalTriangle(), MinEnclosingTriangleFinder(), updateSideB(), and updateSidesBA().

#### 7.95.4.27 cv::Point2f multiscale::MinEnclosingTriangleFinder::vertexA [private]

Vertex A of the current considered enclosing triangle

Definition at line 29 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), and updateMinEnclosingTriangle().

#### 7.95.4.28 cv::Point2f multiscale::MinEnclosingTriangleFinder::vertexB [private]

Vertex B of the current considered enclosing triangle

Definition at line 30 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), and updateMinEnclosingTriangle().

#### 7.95.4.29 cv::Point2f multiscale::MinEnclosingTriangleFinder::vertexC [private]

Vertex C of the current considered enclosing triangle

Definition at line 31 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), and updateMinEnclosingTriangle().

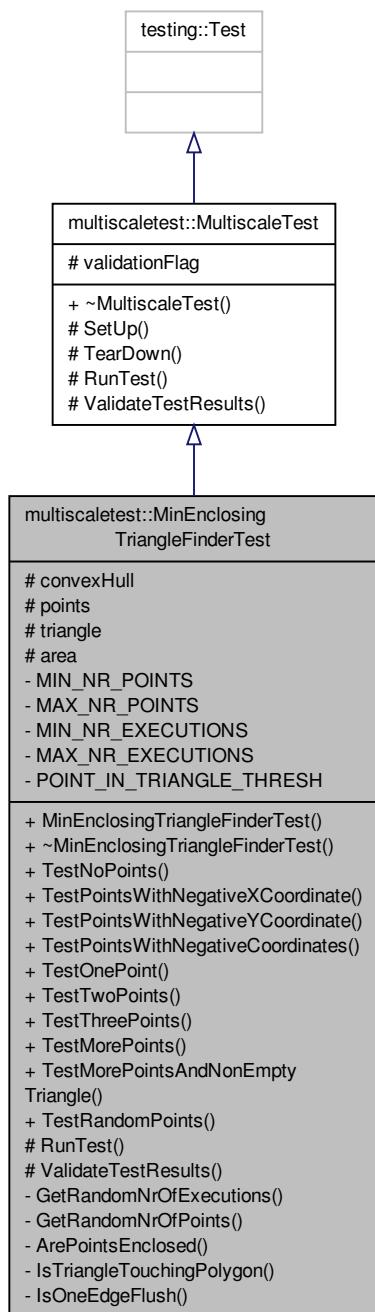
The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[MinEnclosingTriangleFinder.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[MinEnclosingTriangleFinder.cpp](#)

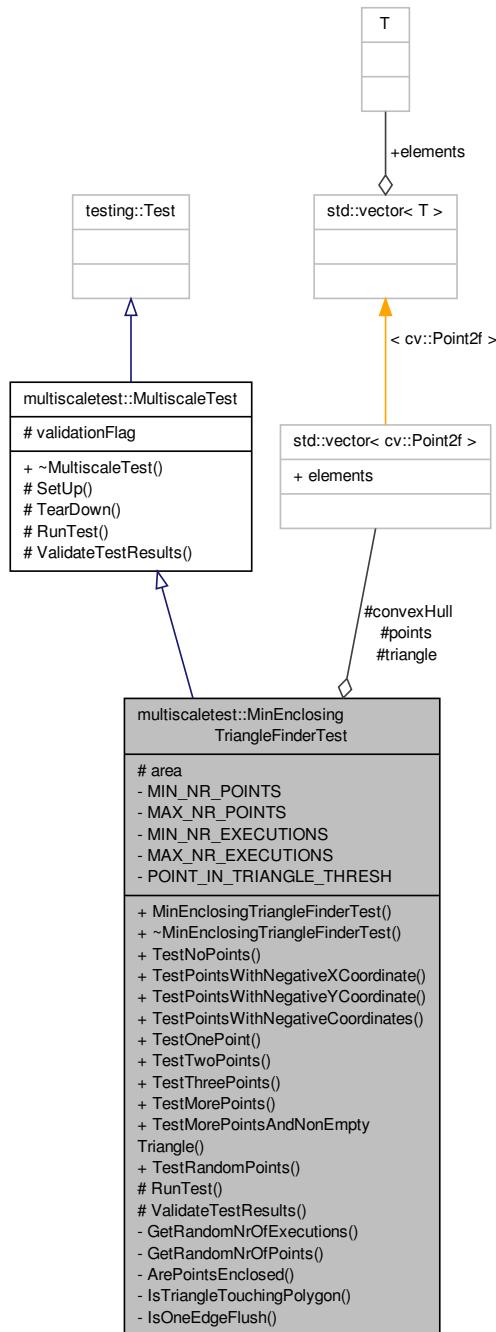
## 7.96 multiscaletest::MinEnclosingTriangleFinderTest Class Reference

Class for testing the minimum enclosing triangle algorithm.

Inheritance diagram for multiscaletest::MinEnclosingTriangleFinderTest:



Collaboration diagram for multiscaletest::MinEnclosingTriangleFinderTest:



## Public Member Functions

- [MinEnclosingTriangleFinderTest \(\)](#)
- [~MinEnclosingTriangleFinderTest \(\)](#)
- [bool TestNoPoints \(\)](#)

*Test the scenario when an empty std::vector of points is provided.*

- [bool TestPointsWithNegativeXCoordinate \(\)](#)

- Test the scenario when there exists at least one point with negative x coordinate.  
bool [TestPointsWithNegativeYCoordinate \(\)](#)  
Test the scenario when there exists at least one point with negative y coordinate.
- bool [TestPointsWithNegativeCoordinates \(\)](#)  
Test the scenario when there exists at least one point with negative coordinates.
- bool [TestOnePoint \(\)](#)  
Test the scenario when only one input point is provided.
- bool [TestTwoPoints \(\)](#)  
Test the scenario when only two input points are provided.
- bool [TestThreePoints \(\)](#)  
Test the scenario when only three input points are provided.
- bool [TestMorePoints \(\)](#)  
Test the scenario when more than three input points are provided.
- bool [TestMorePointsAndNonEmptyTriangle \(\)](#)  
Test the scenario when the output std::vector is not empty.
- bool [TestRandomPoints \(\)](#)  
Test the scenario when randomly initialised std::vectors of input points are provided.

## Protected Member Functions

- void [RunTest \(\) override](#)  
Run the test for the given set of points.
- void [ValidateTestResults \(\) override](#)  
Check if the obtained results are valid.

## Protected Attributes

- std::vector< cv::Point2f > [convexHull](#)
- std::vector< cv::Point2f > [points](#)
- std::vector< cv::Point2f > [triangle](#)
- double [area](#)

## Private Member Functions

- int [GetRandomNrOfExecutions \(\)](#)  
Get a random number of executions.
- int [GetRandomNrOfPoints \(\)](#)  
Get a random number of points.
- bool [ArePointsEnclosed \(\)](#)  
Check if all the points are enclosed by the polygon.
- bool [IsTriangleTouchingPolygon \(\)](#)  
Check if the triangle's middle points are touching the polygon.
- bool [IsOneEdgeFlush \(\)](#)  
Check if at least one of the triangle sides is flush with a polygon edge.

## Static Private Attributes

- static const int [MIN\\_NR\\_POINTS](#) = 1
- static const int [MAX\\_NR\\_POINTS](#) = 10000
- static const int [MIN\\_NR\\_EXECUTIONS](#) = 5000
- static const int [MAX\\_NR\\_EXECUTIONS](#) = 10000
- static const double [POINT\\_IN\\_TRIANGLE\\_THRESH](#) = 1E-4

### 7.96.1 Detailed Description

Class for testing the minimum enclosing triangle algorithm.

Definition at line 14 of file MinEnclosingTriangleFinderTest.cpp.

### 7.96.2 Constructor & Destructor Documentation

#### 7.96.2.1 multiscaletest::MinEnclosingTriangleFinderTest::MinEnclosingTriangleFinderTest ( )

Definition at line 96 of file MinEnclosingTriangleFinderTest.cpp.

#### 7.96.2.2 multiscaletest::MinEnclosingTriangleFinderTest::~MinEnclosingTriangleFinderTest ( )

Definition at line 104 of file MinEnclosingTriangleFinderTest.cpp.

### 7.96.3 Member Function Documentation

#### 7.96.3.1 bool multiscaletest::MinEnclosingTriangleFinderTest::ArePointsEnclosed ( ) [private]

Check if all the points are enclosed by the polygon.

Definition at line 244 of file MinEnclosingTriangleFinderTest.cpp.

References POINT\_IN\_TRIANGLE\_THRESH.

#### 7.96.3.2 int multiscaletest::MinEnclosingTriangleFinderTest::GetRandomNrOfExecutions ( ) [private]

Get a random number of executions.

Definition at line 234 of file MinEnclosingTriangleFinderTest.cpp.

#### 7.96.3.3 int multiscaletest::MinEnclosingTriangleFinderTest::GetRandomNrOfPoints ( ) [private]

Get a random number of points.

Definition at line 239 of file MinEnclosingTriangleFinderTest.cpp.

#### 7.96.3.4 bool multiscaletest::MinEnclosingTriangleFinderTest::IsOneEdgeFlush ( ) [private]

Check if at least one of the triangle sides is flush with a polygon edge.

Definition at line 280 of file MinEnclosingTriangleFinderTest.cpp.

References multiscale::Geometry2D::isPointOnLineSegment().

#### 7.96.3.5 bool multiscaletest::MinEnclosingTriangleFinderTest::IsTriangleTouchingPolygon ( ) [private]

Check if the triangle's middle points are touching the polygon.

Definition at line 258 of file MinEnclosingTriangleFinderTest.cpp.

References multiscale::Geometry2D::isPointOnLineSegment(), and multiscale::Geometry2D::middlePoint().

```
7.96.3.6 void multiscaletest::MinEnclosingTriangleFinderTest::RunTest() [override], [protected],
[virtual]
```

Run the test for the given set of points.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 220 of file MinEnclosingTriangleFinderTest.cpp.

References multiscale::MinEnclosingTriangleFinder::find().

```
7.96.3.7 bool multiscaletest::MinEnclosingTriangleFinderTest::TestMorePoints()
```

Test the scenario when more than three input points are provided.

Definition at line 173 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.8 bool multiscaletest::MinEnclosingTriangleFinderTest::TestMorePointsAndNonEmptyTriangle()
```

Test the scenario when the output std::vector is not empty.

Definition at line 183 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.9 bool multiscaletest::MinEnclosingTriangleFinderTest::TestNoPoints()
```

Test the scenario when an empty std::vector of points is provided.

Definition at line 112 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.10 bool multiscaletest::MinEnclosingTriangleFinderTest::TestOnePoint()
```

Test the scenario when only one input point is provided.

Definition at line 146 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.11 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWithNegativeCoordinates()
```

Test the scenario when there exists at least one point with negative coordinates.

Definition at line 137 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.12 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWithNegativeXCoordinate()
```

Test the scenario when there exists at least one point with negative x coordinate.

Definition at line 119 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.13 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWithNegativeYCoordinate()
```

Test the scenario when there exists at least one point with negative y coordinate.

Definition at line 128 of file MinEnclosingTriangleFinderTest.cpp.

```
7.96.3.14 bool multiscaletest::MinEnclosingTriangleFinderTest::TestRandomPoints()
```

Test the scenario when randomly initialised std::vectors of input points are provided.

Definition at line 195 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.3.15 bool multiscaletest::MinEnclosingTriangleFinderTest::TestThreePoints( )**

Test the scenario when only three input points are provided.

Definition at line 164 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.3.16 bool multiscaletest::MinEnclosingTriangleFinderTest::TestTwoPoints( )**

Test the scenario when only two input points are provided.

Definition at line 155 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.3.17 void multiscaletest::MinEnclosingTriangleFinderTest::ValidateTestResults( ) [override], [protected], [virtual]**

Check if the obtained results are valid.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 226 of file MinEnclosingTriangleFinderTest.cpp.

## 7.96.4 Member Data Documentation

**7.96.4.1 double multiscaletest::MinEnclosingTriangleFinderTest::area [protected]**

Area of the minimum enclosing triangle

Definition at line 22 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.4.2 std::vector<cv::Point2f> multiscaletest::MinEnclosingTriangleFinderTest::convexHull [protected]**

Convex hull of the 2D point set

Definition at line 18 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.4.3 const int multiscaletest::MinEnclosingTriangleFinderTest::MAX\_NR\_EXECUTIONS = 10000 [static], [private]**

Definition at line 90 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.4.4 const int multiscaletest::MinEnclosingTriangleFinderTest::MAX\_NR\_POINTS = 10000 [static], [private]**

Definition at line 88 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.4.5 const int multiscaletest::MinEnclosingTriangleFinderTest::MIN\_NR\_EXECUTIONS = 5000 [static], [private]**

Definition at line 89 of file MinEnclosingTriangleFinderTest.cpp.

**7.96.4.6 const int multiscaletest::MinEnclosingTriangleFinderTest::MIN\_NR\_POINTS = 1 [static], [private]**

Definition at line 87 of file MinEnclosingTriangleFinderTest.cpp.

7.96.4.7 `const double multiscaletest::MinEnclosingTriangleFinderTest::POINT_IN_TRIANGLE_THRESH = 1E-4 [static], [private]`

Definition at line 92 of file MinEnclosingTriangleFinderTest.cpp.

7.96.4.8 `std::vector<cv::Point2f> multiscaletest::MinEnclosingTriangleFinderTest::points [protected]`

Collection of 2D points

Definition at line 20 of file MinEnclosingTriangleFinderTest.cpp.

7.96.4.9 `std::vector<cv::Point2f> multiscaletest::MinEnclosingTriangleFinderTest::triangle [protected]`

Minimum enclosing triangle

Definition at line 21 of file MinEnclosingTriangleFinderTest.cpp.

The documentation for this class was generated from the following file:

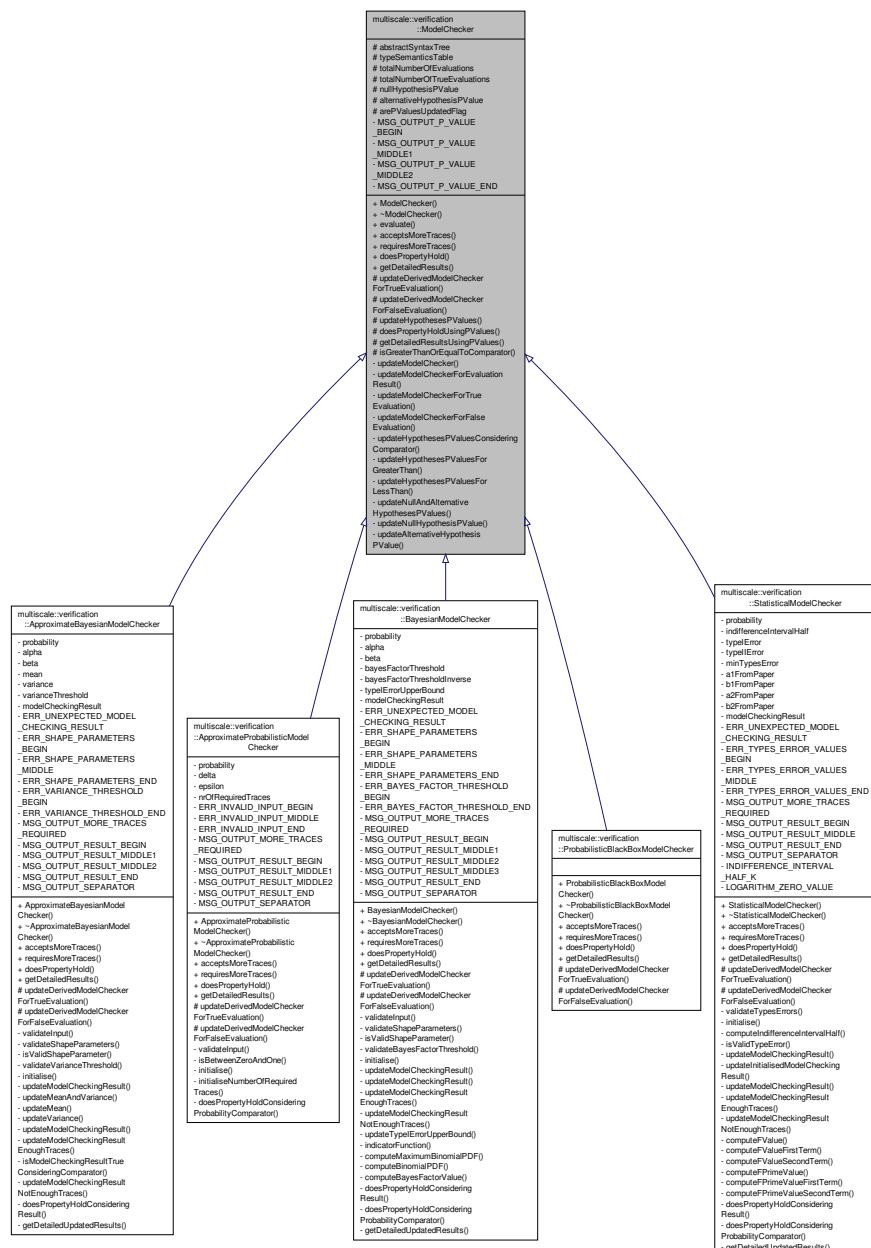
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/MinEnclosingTriangleFinderTest.cpp

## 7.97 multiscale::verification::ModelChecker Class Reference

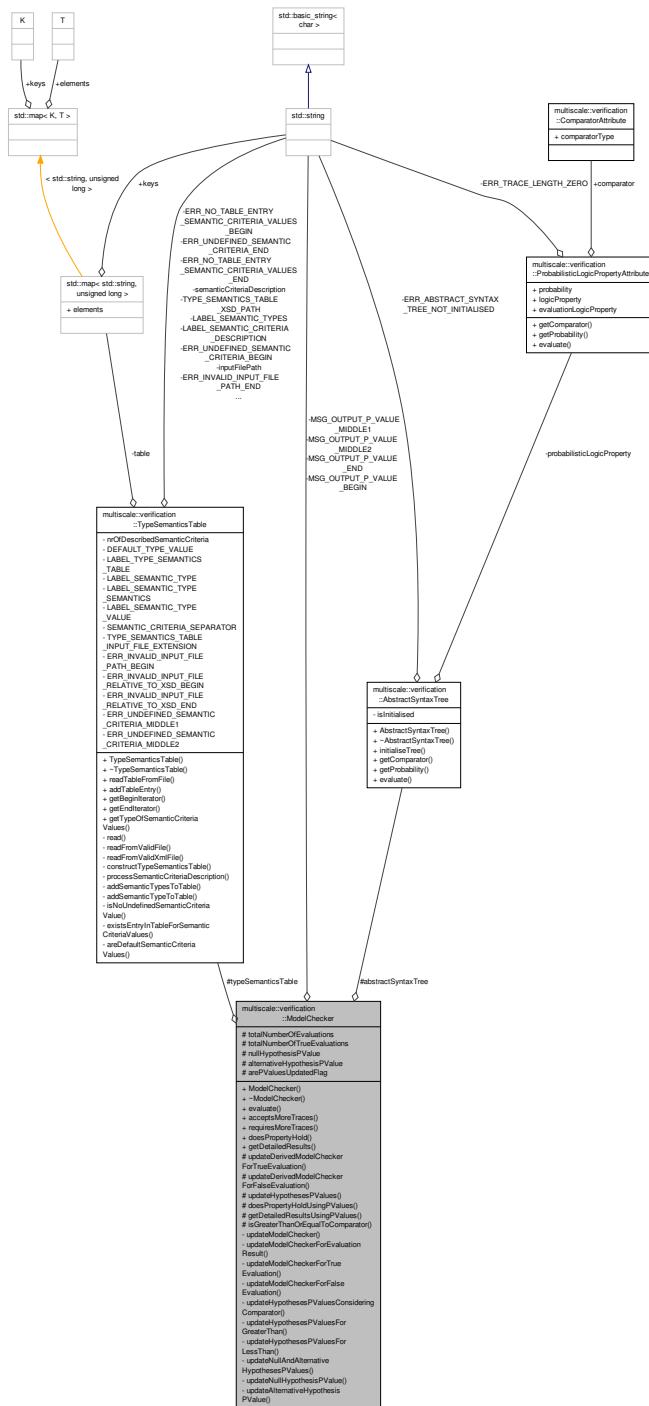
Abstract class representing a generic model checker.

```
#include <ModelChecker.hpp>
```

## Inheritance diagram for multiscale::verification::ModelChecker



## Collaboration diagram for multiscale::verification::ModelChecker:



## Public Member Functions

- `ModelChecker` (const `AbstractSyntaxTree` &`abstractSyntaxTree`, const `TypeSemanticsTable` &`typeSemanticsTable`)
  - virtual ~`ModelChecker` ()
  - bool `evaluate` (const `SpatialTemporalTrace` &`trace`)

Evaluate the abstract syntax tree for the given trace and return the result.

- virtual bool `acceptsMoreTraces ()=0`  
*Check if more traces are accepted for evaluating the logic property.*
- virtual bool `requiresMoreTraces ()=0`  
*Check if more traces are required for evaluating the logic property.*
- virtual bool `doesPropertyHold ()=0`  
*Check if the given property holds.*
- virtual std::string `getDetailedResults ()=0`  
*Get a detailed report of the results.*

## Protected Member Functions

- virtual void `updateDerivedModelCheckerForTrueEvaluation ()=0`  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- virtual void `updateDerivedModelCheckerForFalseEvaluation ()=0`  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*
- void `updateHypothesesPValues ()`  
*Update the p-values for the null and alternative hypothesis.*
- bool `doesPropertyHoldUsingPValues ()`  
*Check if the property holds considering the given p-values.*
- std::string `getDetailedResultsUsingPValues ()`  
*Get the detailed results when deciding if the property holds based on p-values.*
- bool `isGreaterThanOrEqualToComparator ()`  
*Check if the comparator used by the probabilistic logic property is greater than or equal to.*

## Protected Attributes

- `AbstractSyntaxTree abstractSyntaxTree`
- `TypeSemanticsTable typeSemanticsTable`
- unsigned int `totalNumberOfEvaluations`
- unsigned int `totalNumberOfTrueEvaluations`
- double `nullHypothesisPValue`
- double `alternativeHypothesisPValue`
- bool `arePValuesUpdatedFlag`

## Private Member Functions

- void `updateModelChecker (bool evaluationResult)`  
*Update the model checker results considering that the logic property was evaluated to evaluationResult for the last trace.*
- void `updateModelCheckerForEvaluationResult (bool evaluationResult)`  
*Update the model checker results considering that the logic property was evaluated to evaluationResult for the last trace.*
- void `updateModelCheckerForTrueEvaluation ()`  
*Update the results of the model checker considering that the logic property was evaluated to true for the last trace.*
- void `updateModelCheckerForFalseEvaluation ()`  
*Update the results of the model checker considering that the logic property was evaluated to false for the last trace.*
- void `updateHypothesesPValuesConsideringComparator ()`  
*Update the p-values for the null and alternative hypothesis considering the comparator contained by the probabilistic logic property.*

- void [updateHypothesesPValuesForGreater Than \(\)](#)  
*Update the p-values considering that the probabilistic logic property is of the form  $P > [=]\theta[\phi]$ .*
- void [updateHypothesesPValuesForLessThan \(\)](#)  
*Update the p-values considering that the probabilistic logic property is of the form  $P < [=]\theta[\phi]$ .*
- void [updateNullAndAlternativeHypothesesPValues \(unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability\)](#)  
*Update the null and alternative hypotheses p-values.*
- void [updateNullHypothesisPValue \(unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability\)](#)  
*Update the null hypothesis p-value.*
- void [updateAlternativeHypothesisPValue \(unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability\)](#)  
*Update the alternative hypothesis p-value.*

## Static Private Attributes

- static const std::string [MSG\\_OUTPUT\\_P\\_VALUE\\_BEGIN](#) = "The confidence level of the answer expressed as a p-value (lower is better): "
- static const std::string [MSG\\_OUTPUT\\_P\\_VALUE\\_MIDDLE1](#) = "(p-value H0: "
- static const std::string [MSG\\_OUTPUT\\_P\\_VALUE\\_MIDDLE2](#) = ", p-value H1: "
- static const std::string [MSG\\_OUTPUT\\_P\\_VALUE\\_END](#) = ")"

### 7.97.1 Detailed Description

Abstract class representing a generic model checker.

Definition at line 14 of file ModelChecker.hpp.

### 7.97.2 Constructor & Destructor Documentation

7.97.2.1 [multiscale::verification::ModelChecker::ModelChecker \( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable \) \[inline\]](#)

Definition at line 36 of file ModelChecker.hpp.

7.97.2.2 [virtual multiscale::verification::ModelChecker::~ModelChecker \( \) \[inline\], \[virtual\]](#)

Definition at line 45 of file ModelChecker.hpp.

### 7.97.3 Member Function Documentation

7.97.3.1 [virtual bool multiscale::verification::ModelChecker::acceptsMoreTraces \( \) \[pure virtual\]](#)

Check if more traces are accepted for evaluating the logic property.

Implemented in [multiscale::verification::StatisticalModelChecker](#), [multiscale::verification::ApproximateBayesianModelChecker](#), [multiscale::verification::BayesianModelChecker](#), [multiscale::verification::ApproximateProbabilisticModelChecker](#), and [multiscale::verification::ProbabilisticBlackBoxModelChecker](#).

### 7.97.3.2 virtual bool multiscale::verification::ModelChecker::doesPropertyHold( ) [pure virtual]

Check if the given property holds.

Implemented in [multiscale::verification::StatisticalModelChecker](#), [multiscale::verification::ApproximateBayesianModelChecker](#), [multiscale::verification::BayesianModelChecker](#), [multiscale::verification::ApproximateProbabilisticModelChecker](#), and [multiscale::verification::ProbabilisticBlackBoxModelChecker](#).

### 7.97.3.3 bool ModelChecker::doesPropertyHoldUsingPValues( ) [protected]

Check if the property holds considering the given p-values.

Definition at line 24 of file [ModelChecker.cpp](#).

References [alternativeHypothesisPValue](#), [nullHypothesisPValue](#), and [updateHypothesesPValues\(\)](#).

Referenced by [multiscale::verification::ProbabilisticBlackBoxModelChecker::doesPropertyHold\(\)](#), [multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHold\(\)](#), [multiscale::verification::ApproximateBayesianModelChecker::doesPropertyHoldConsideringResult\(\)](#), [multiscale::verification::BayesianModelChecker::doesPropertyHoldConsideringResult\(\)](#), and [multiscale::verification::StatisticalModelChecker::doesPropertyHoldConsideringResult\(\)](#).

### 7.97.3.4 bool ModelChecker::evaluate( const SpatialTemporalTrace & trace )

Evaluate the abstract syntax tree for the given trace and return the result.

Parameters

|              |                                  |
|--------------|----------------------------------|
| <i>trace</i> | The given spatial temporal trace |
|--------------|----------------------------------|

Definition at line 8 of file [ModelChecker.cpp](#).

References [abstractSyntaxTree](#), [multiscale::verification::AbstractSyntaxTree::evaluate\(\)](#), [typeSemanticsTable](#), and [updateModelChecker\(\)](#).

### 7.97.3.5 virtual std::string multiscale::verification::ModelChecker::getDetailedResults( ) [pure virtual]

Get a detailed report of the results.

Implemented in [multiscale::verification::StatisticalModelChecker](#), [multiscale::verification::ApproximateBayesianModelChecker](#), [multiscale::verification::BayesianModelChecker](#), [multiscale::verification::ApproximateProbabilisticModelChecker](#), and [multiscale::verification::ProbabilisticBlackBoxModelChecker](#).

### 7.97.3.6 std::string ModelChecker::getDetailedResultsUsingPValues( ) [protected]

Get the detailed results when deciding if the property holds based on p-values.

Definition at line 30 of file [ModelChecker.cpp](#).

References [alternativeHypothesisPValue](#), [MSG\\_OUTPUT\\_P\\_VALUE\\_BEGIN](#), [MSG\\_OUTPUT\\_P\\_VALUE\\_END](#), [MSG\\_OUTPUT\\_P\\_VALUE\\_MIDDLE1](#), [MSG\\_OUTPUT\\_P\\_VALUE\\_MIDDLE2](#), [nullHypothesisPValue](#), [multiscale::StringManipulator::toString\(\)](#), and [updateHypothesesPValues\(\)](#).

Referenced by [multiscale::verification::ProbabilisticBlackBoxModelChecker::getDetailedResults\(\)](#), [multiscale::verification::ApproximateProbabilisticModelChecker::getDetailedResults\(\)](#), [multiscale::verification::ApproximateBayesianModelChecker::getDetailedUpdatedResults\(\)](#), [multiscale::verification::BayesianModelChecker::getDetailedUpdatedResults\(\)](#), and [multiscale::verification::StatisticalModelChecker::getDetailedUpdatedResults\(\)](#).

### 7.97.3.7 bool ModelChecker::isGreaterThanOrEqualToComparator ( ) [protected]

Check if the comparator used by the probabilistic logic property is greater than or equal to.

Definition at line 42 of file ModelChecker.cpp.

References `abstractSyntaxTree`, `multiscale::verification::AbstractSyntaxTree::getComparator()`, `multiscale::verification::GreaterThan`, and `multiscale::verification::GreaterOrEqual`.

Referenced by `multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::BayesianModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::StatisticalModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator()`, and `updateHypothesesPValuesConsideringComparator()`.

### 7.97.3.8 virtual bool multiscale::verification::ModelChecker::requiresMoreTraces ( ) [pure virtual]

Check if more traces are required for evaluating the logic property.

Implemented in `multiscale::verification::StatisticalModelChecker`, `multiscale::verification::ApproximateBayesianModelChecker`, `multiscale::verification::BayesianModelChecker`, `multiscale::verification::ApproximateProbabilisticModelChecker`, and `multiscale::verification::ProbabilisticBlackBoxModelChecker`.

### 7.97.3.9 void ModelChecker::updateAlternativeHypothesisPValue ( unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability ) [private]

Update the alternative hypothesis p-value.

Parameters

|                              |                                                 |
|------------------------------|-------------------------------------------------|
| <code>nrOfEvaluations</code> | The number of evaluations                       |
| <code>nrOfSuccesses</code>   | The number of true evaluations                  |
| <code>probability</code>     | The probability specified in the logic property |

Definition at line 115 of file ModelChecker.cpp.

References `alternativeHypothesisPValue`, and `multiscale::BinomialDistribution::cdf()`.

Referenced by `updateNullAndAlternativeHypothesesPValues()`.

### 7.97.3.10 virtual void multiscale::verification::ModelChecker::updateDerivedModelCheckerForFalseEvaluation ( ) [protected], [pure virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Implemented in `multiscale::verification::StatisticalModelChecker`, `multiscale::verification::ApproximateBayesianModelChecker`, `multiscale::verification::BayesianModelChecker`, `multiscale::verification::ApproximateProbabilisticModelChecker`, and `multiscale::verification::ProbabilisticBlackBoxModelChecker`.

### 7.97.3.11 virtual void multiscale::verification::ModelChecker::updateDerivedModelCheckerForTrueEvaluation ( ) [protected], [pure virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Implemented in `multiscale::verification::StatisticalModelChecker`, `multiscale::verification::ApproximateBayesianModelChecker`, `multiscale::verification::BayesianModelChecker`, `multiscale::verification::ApproximateProbabilisticModelChecker`, and `multiscale::verification::ProbabilisticBlackBoxModelChecker`.

### 7.97.3.12 void ModelChecker::updateHypothesesPValues( ) [protected]

Update the p-values for the null and alternative hypothesis.

The method for updating the p-values is based on considering that each trace is represented by a Bernoulli variable which can be either true or false with respect to the given logic property.

The probability distribution of a sum of n Bernoulli variables (where n = number of traces) is a binomial distribution. Using the cumulative distribution function the p-values of the hypotheses can be computed.

More details are given in the following paper: H. L. S. Younes, ‘Probabilistic Verification for “Black-Box” Systems’, in Computer Aided Verification, K. Etessami and S. K. Rajamani, Eds. Springer Berlin Heidelberg, 2005, pp. 253–265.

Definition at line 16 of file ModelChecker.cpp.

References arePValuesUpdatedFlag, and updateHypothesesPValuesConsideringComparator().

Referenced by doesPropertyHoldUsingPValues(), and getDetailedResultsUsingPValues().

### 7.97.3.13 void ModelChecker::updateHypothesesPValuesConsideringComparator( ) [private]

Update the p-values for the null and alternative hypothesis considering the comparator contained by the probabilistic logic property.

Definition at line 74 of file ModelChecker.cpp.

References isGreaterThanOrEqualToComparator(), updateHypothesesPValuesForGreaterThan(), and updateHypothesesPValuesForLessThan().

Referenced by updateHypothesesPValues().

### 7.97.3.14 void ModelChecker::updateHypothesesPValuesForGreaterThan( ) [private]

Update the p-values considering that the probabilistic logic property is of the form  $P > [=]\theta[\phi]$ .

$p-value_{H_0} = 1 - F(d - 1; n, \theta)$   $p-value_{H_1} = F(d; n, \theta)$  where d = number of true evaluations, n = number of evaluations and  $\theta$  = probability specified in the logic property

Definition at line 82 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), totalNumberOfEvaluations, totalNumberOfTrueEvaluations, and updateNullAndAlternativeHypothesesPValues().

Referenced by updateHypothesesPValuesConsideringComparator().

### 7.97.3.15 void ModelChecker::updateHypothesesPValuesForLessThan( ) [private]

Update the p-values considering that the probabilistic logic property is of the form  $P < [=]\theta[\phi]$ .

$p-value_{H_0} = 1 - F(d' - 1; n, \theta)$   $p-value_{H_1} = F(d'; n, \theta)$  where  $d' = n - d$ , d = number of true evaluations, n = number of evaluations and  $\theta$  = probability specified in the logic property

Definition at line 90 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), totalNumberOfEvaluations, totalNumberOfTrueEvaluations, and updateNullAndAlternativeHypothesesPValues().

Referenced by updateHypothesesPValuesConsideringComparator().

### 7.97.3.16 void ModelChecker::updateModelChecker( bool evaluationResult ) [private]

Update the model checker results considering that the logic property was evaluated to evaluationResult for the last trace.

**Parameters**

|                         |                                                                        |
|-------------------------|------------------------------------------------------------------------|
| <i>evaluationResult</i> | The result of evaluating the logic property considering the last trace |
|-------------------------|------------------------------------------------------------------------|

Definition at line 51 of file ModelChecker.cpp.

References arePValuesUpdatedFlag, and updateModelCheckerForEvaluationResult().

Referenced by evaluate().

**7.97.3.17 void ModelChecker::updateModelCheckerForEvaluationResult ( bool *evaluationResult* ) [private]**

Update the model checker results considering that the logic property was evaluated to evaluationResult for the last trace.

**Parameters**

|                         |                                                                        |
|-------------------------|------------------------------------------------------------------------|
| <i>evaluationResult</i> | The result of evaluating the logic property considering the last trace |
|-------------------------|------------------------------------------------------------------------|

Definition at line 57 of file ModelChecker.cpp.

References updateModelCheckerForFalseEvaluation(), and updateModelCheckerForTrueEvaluation().

Referenced by updateModelChecker().

**7.97.3.18 void ModelChecker::updateModelCheckerForFalseEvaluation ( ) [private]**

Update the results of the model checker considering that the logic property was evaluated to false for the last trace.

Definition at line 70 of file ModelChecker.cpp.

References totalNumberOfEvaluations.

Referenced by updateModelCheckerForEvaluationResult().

**7.97.3.19 void ModelChecker::updateModelCheckerForTrueEvaluation ( ) [private]**

Update the results of the model checker considering that the logic property was evaluated to true for the last trace.

Definition at line 65 of file ModelChecker.cpp.

References totalNumberOfEvaluations, and totalNumberOfTrueEvaluations.

Referenced by updateModelCheckerForEvaluationResult().

**7.97.3.20 void ModelChecker::updateNullAndAlternativeHypothesesPValues ( unsigned int *nrOfEvaluations*, unsigned int *nrOfSuccesses*, double *probability* ) [private]**

Update the null and alternative hypotheses p-values.

**Parameters**

|                        |                                                 |
|------------------------|-------------------------------------------------|
| <i>nrOfEvaluations</i> | The number of evaluations                       |
| <i>nrOfSuccesses</i>   | The number of true evaluations                  |
| <i>probability</i>     | The probability specified in the logic property |

Definition at line 98 of file ModelChecker.cpp.

References updateAlternativeHypothesisPValue(), and updateNullHypothesisPValue().

Referenced by updateHypothesesPValuesForGreaterThan(), and updateHypothesesPValuesForLessThan().

7.97.3.21 void ModelChecker::updateNullHypothesisPValue ( *unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability* ) [private]

Update the null hypothesis p-value.

**Parameters**

|                        |                                                 |
|------------------------|-------------------------------------------------|
| <i>nrOfEvaluations</i> | The number of evaluations                       |
| <i>nrOfSuccesses</i>   | The number of true evaluations                  |
| <i>probability</i>     | The probability specified in the logic property |

Definition at line 105 of file ModelChecker.cpp.

References multiscale::BinomialDistribution::cdf(), and nullHypothesisPValue.

Referenced by updateNullAndAlternativeHypothesesPValues().

## 7.97.4 Member Data Documentation

### 7.97.4.1 AbstractSyntaxTree multiscale::verification::ModelChecker::abstractSyntaxTree [protected]

The abstract syntax tree representing the logic property which this model checker instance evaluates

Definition at line 18 of file ModelChecker.hpp.

Referenced by evaluate(), multiscale::verification::ApproximateProbabilisticModelChecker::initialise(), multiscale::verification::StatisticalModelChecker::initialise(), multiscale::verification::BayesianModelChecker::initialise(), multiscale::verification::ApproximateBayesianModelChecker::initialise(), isGreaterThanOrEqualToComparator(), updateHypothesesPValuesForGreaterThan(), and updateHypothesesPValuesForLessThan().

### 7.97.4.2 double multiscale::verification::ModelChecker::alternativeHypothesisPValue [protected]

The p-value for the alternative hypothesis to hold

Definition at line 30 of file ModelChecker.hpp.

Referenced by doesPropertyHoldUsingPValues(), getDetailedResultsUsingPValues(), and updateAlternativeHypothesisPValue().

### 7.97.4.3 bool multiscale::verification::ModelChecker::arePValuesUpdatedFlag [protected]

Flag indicating if the p-values were updated

Definition at line 32 of file ModelChecker.hpp.

Referenced by updateHypothesesPValues(), and updateModelChecker().

### 7.97.4.4 const std::string ModelChecker::MSG\_OUTPUT\_P\_VALUE\_BEGIN = "The confidence level of the answer expressed as a p-value (lower is better): " [static], [private]

Definition at line 169 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

### 7.97.4.5 const std::string ModelChecker::MSG\_OUTPUT\_P\_VALUE\_END = ")" [static], [private]

Definition at line 172 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

### 7.97.4.6 const std::string ModelChecker::MSG\_OUTPUT\_P\_VALUE\_MIDDLE1 = " (p-value H0: " [static], [private]

Definition at line 170 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

7.97.4.7 `const std::string ModelChecker::MSG_OUTPUT_P_VALUE_MIDDLE2 = "p-value H1: "` [static], [private]

Definition at line 171 of file ModelChecker.hpp.

Referenced by `getDetailedResultsUsingPValues()`.

7.97.4.8 `double multiscale::verification::ModelChecker::nullHypothesisPValue` [protected]

The p-value for the null hypothesis to hold

Definition at line 29 of file ModelChecker.hpp.

Referenced by `doesPropertyHoldUsingPValues()`, `getDetailedResultsUsingPValues()`, and `updateNullHypothesisPValue()`.

7.97.4.9 `unsigned int multiscale::verification::ModelChecker::totalNumberOfEvaluations` [protected]

The total number of evaluations

Definition at line 25 of file ModelChecker.hpp.

Referenced by `multiscale::verification::ApproximateProbabilisticModelChecker::acceptsMoreTraces()`, `multiscale::verification::BayesianModelChecker::computeBinomialPDF()`, `multiscale::verification::StatisticalModelChecker::computeFPrimeValue()`, `multiscale::verification::StatisticalModelChecker::computeFValue()`, `multiscale::verification::BayesianModelChecker::computeMaximumBinomialPDF()`, `multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::BayesianModelChecker::indicatorFunction()`, `updateHypothesesPValuesForGreaterThan()`, `updateHypothesesPValuesForLessThan()`, `multiscale::verification::ApproximateBayesianModelChecker::updateMean()`, `updateModelCheckerForFalseEvaluation()`, `updateModelCheckerForTrueEvaluation()`, `multiscale::verification::BayesianModelChecker::updateModelCheckingResult()`, `multiscale::verification::BayesianModelChecker::updateTypeIErrorUpperBound()`, and `multiscale::verification::ApproximateBayesianModelChecker::updateVariance()`.

7.97.4.10 `unsigned int multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations` [protected]

The total number of times the abstract syntax tree was evaluated to true

Definition at line 26 of file ModelChecker.hpp.

Referenced by `multiscale::verification::StatisticalModelChecker::computeFPrimeValue()`, `multiscale::verification::StatisticalModelChecker::computeFValue()`, `multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `updateHypothesesPValuesForGreaterThan()`, `updateHypothesesPValuesForLessThan()`, `multiscale::verification::ApproximateBayesianModelChecker::updateMean()`, `updateModelCheckerForTrueEvaluation()`, `multiscale::verification::BayesianModelChecker::updateModelCheckingResult()`, and `multiscale::verification::ApproximateBayesianModelChecker::updateVariance()`.

7.97.4.11 `TypeSemanticsTable multiscale::verification::ModelChecker::typeSemanticsTable` [protected]

The type semantics table mapping semantic criteria values to abstract natural numbers

Definition at line 22 of file ModelChecker.hpp.

Referenced by `evaluate()`.

The documentation for this class was generated from the following files:

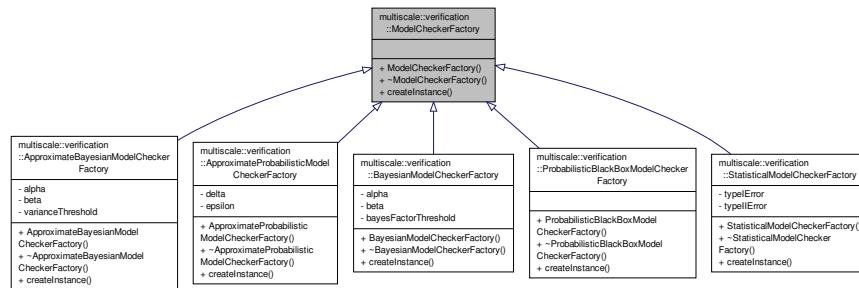
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ModelChecker.cpp](#)

## 7.98 multiscale::verification::ModelCheckerFactory Class Reference

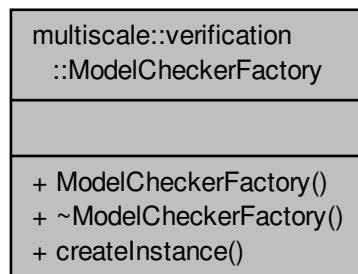
Interface for different model checker factories.

```
#include <ModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckerFactory:



Collaboration diagram for multiscale::verification::ModelCheckerFactory:



### Public Member Functions

- [ModelCheckerFactory \(\)](#)
- virtual [~ModelCheckerFactory \(\)](#)
- virtual std::shared\_ptr< [ModelChecker](#) > [createInstance](#) (const [AbstractSyntaxTree](#) &abstractSyntaxTree, const [TypeSemantics](#) &[Table](#) &[typeSemanticsTable](#))=0

*Create an instance of the model checker.*

#### 7.98.1 Detailed Description

Interface for different model checker factories.

Definition at line 15 of file ModelCheckerFactory.hpp.

## 7.98.2 Constructor & Destructor Documentation

7.98.2.1 `multiscale::verification::ModelCheckerFactory::ModelCheckerFactory( ) [inline]`

Definition at line 19 of file `ModelCheckerFactory.hpp`.

7.98.2.2 `virtual multiscale::verification::ModelCheckerFactory::~ModelCheckerFactory( ) [inline], [virtual]`

Definition at line 20 of file `ModelCheckerFactory.hpp`.

## 7.98.3 Member Function Documentation

7.98.3.1 `virtual std::shared_ptr<ModelChecker> multiscale::verification::ModelCheckerFactory::createInstance( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [pure virtual]`

Create an instance of the model checker.

### Parameters

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| <code>abstractSyntaxTree</code> | The abstract syntax tree representing the logic property to be checked                |
| <code>typeSemanticsTable</code> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implemented in `multiscale::verification::ApproximateBayesianModelCheckerFactory`, `multiscale::verification::BayesianModelCheckerFactory`, `multiscale::verification::ApproximateProbabilisticModelCheckerFactory`, `multiscale::verification::StatisticalModelCheckerFactory`, and `multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory`.

The documentation for this class was generated from the following file:

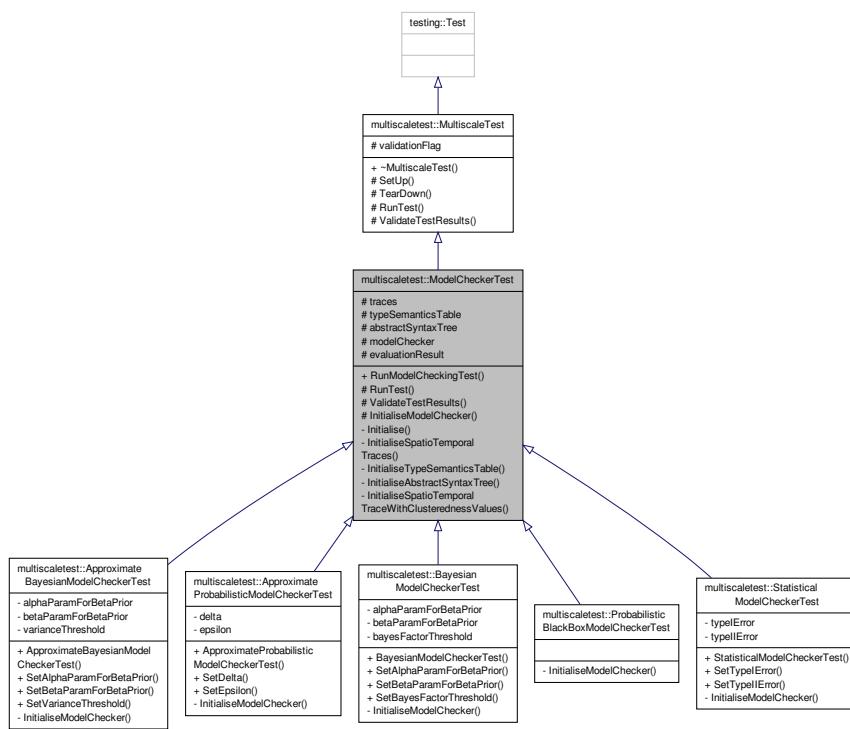
- `/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp`

## 7.99 multiscaletest::ModelCheckerTest Class Reference

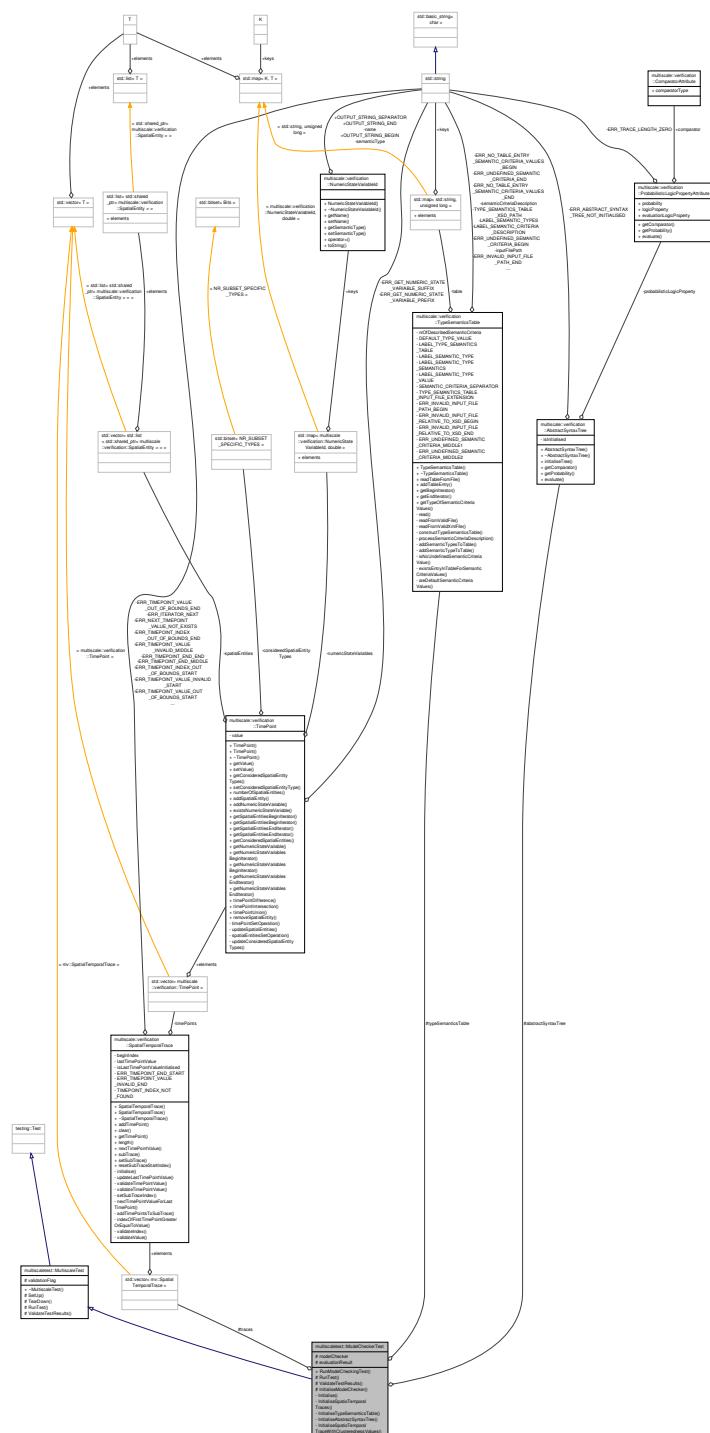
Class for testing model checkers.

```
#include <ModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::ModelCheckerTest:



## Collaboration diagram for multiscaletest::ModelCheckerTest:



## Public Member Functions

- bool RunModelCheckingTest()

*Run the test for the given logic property.*

## Protected Member Functions

- virtual void [RunTest \(\) override](#)  
*Run the test.*
- virtual void [ValidateTestResults \(\) override](#)  
*Validate the results of the test.*
- virtual void [InitialiseModelChecker \(\)=0](#)  
*Initialise the model checker.*

## Protected Attributes

- std::vector  
 $< \text{mv}::\text{SpatialTemporalTrace} >$  traces
- [TypeSemanticsTable typeSemanticsTable](#)
- [mv::AbstractSyntaxTree abstractSyntaxTree](#)
- std::shared\_ptr< [mv::ModelChecker](#) > modelChecker
- bool [evaluationResult](#)

## Private Member Functions

- void [Initialise \(\)](#)  
*Initialisation function.*
- void [InitialiseSpatioTemporalTraces \(\)](#)  
*Initialise the collection of spatio-temporal traces.*
- void [InitialiseTypeSemanticsTable \(\)](#)  
*Initialise the type semantics table.*
- void [InitialiseAbstractSyntaxTree \(\)](#)  
*Initialise the abstract syntax tree.*
- void [InitialiseSpatioTemporalTraceWithClusterednessValues \(const std::vector< double > clusterednessValues\)](#)  
*Initialise the collection of spatio-temporal traces with the given spatial entity clusteredness values.*

### 7.99.1 Detailed Description

Class for testing model checkers.

Definition at line 30 of file ModelCheckerTest.hpp.

### 7.99.2 Member Function Documentation

#### 7.99.2.1 void multiscaletest::ModelCheckerTest::Initialise ( ) [private]

Initialisation function.

Definition at line 111 of file ModelCheckerTest.hpp.

References [InitialiseAbstractSyntaxTree\(\)](#), [InitialiseModelChecker\(\)](#), [InitialiseSpatioTemporalTraces\(\)](#), and [InitialiseTypeSemanticsTable\(\)](#).

Referenced by [RunModelCheckingTest\(\)](#).

### 7.99.2.2 void multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree( ) [private]

Initialise the abstract syntax tree.

Definition at line 139 of file ModelCheckerTest.hpp.

References abstractSyntaxTree, INPUT\_LOGIC\_PROPERTY, and multiscale::verification::Parser::parse().

Referenced by Initialise().

### 7.99.2.3 virtual void multiscaletest::ModelCheckerTest::InitialiseModelChecker( ) [protected], [pure virtual]

Initialise the model checker.

Implemented in [multiscaletest::ApproximateBayesianModelCheckerTest](#), [multiscaletest::BayesianModelCheckerTest](#), [multiscaletest::ApproximateProbabilisticModelCheckerTest](#), [multiscaletest::StatisticalModelCheckerTest](#), and [multiscaletest::ProbabilisticBlackBoxModelCheckerTest](#).

Referenced by Initialise().

### 7.99.2.4 void multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraces( ) [private]

Initialise the collection of spatio-temporal traces.

Definition at line 118 of file ModelCheckerTest.hpp.

References InitialiseSpatioTemporalTraceWithClusterednessValues(), and traces.

Referenced by Initialise().

### 7.99.2.5 void multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues( const std::vector< double > clusterednessValues ) [private]

Initialise the collection of spatio-temporal traces with the given spatial entity clusteredness values.

The assumption is that each timepoint contains only one spatial entity of the same type. Therefore each clusteredness value corresponds to a different timepoint and spatial entity.

#### Parameters

|                            |                                        |
|----------------------------|----------------------------------------|
| <i>clusterednessValues</i> | The collection of clusteredness values |
|----------------------------|----------------------------------------|

Definition at line 145 of file ModelCheckerTest.hpp.

References multiscale::verification::TimePoint::addSpatialEntity(), multiscale::verification::SpatialTemporalTrace::addTimePoint(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::removeSpatialEntity(), multiscale::verification::TimePoint::setConsideredSpatialEntityType(), and traces.

Referenced by InitialiseSpatioTemporalTraces().

### 7.99.2.6 void multiscaletest::ModelCheckerTest::InitialiseTypeSemanticsTable( ) [private]

Initialise the type semantics table.

Definition at line 135 of file ModelCheckerTest.hpp.

Referenced by Initialise().

### 7.99.2.7 bool multiscaletest::ModelCheckerTest::RunModelCheckingTest( )

Run the test for the given logic property.

Definition at line 90 of file ModelCheckerTest.hpp.

References evaluationResult, Initialise(), RunTest(), and ValidateTestResults().

#### 7.99.2.8 void multiscaletest::ModelCheckerTest::RunTest( ) [override], [protected], [virtual]

Run the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 99 of file ModelCheckerTest.hpp.

References evaluationResult, modelChecker, and traces.

Referenced by RunModelCheckingTest().

#### 7.99.2.9 void multiscaletest::ModelCheckerTest::ValidateTestResults( ) [override], [protected], [virtual]

Validate the results of the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 109 of file ModelCheckerTest.hpp.

Referenced by RunModelCheckingTest().

### 7.99.3 Member Data Documentation

#### 7.99.3.1 [mv::AbstractSyntaxTree](#) multiscaletest::ModelCheckerTest::abstractSyntaxTree [protected]

The abstract syntax tree corresponding to the logic property

Definition at line 42 of file ModelCheckerTest.hpp.

Referenced by InitialiseAbstractSyntaxTree().

#### 7.99.3.2 bool multiscaletest::ModelCheckerTest::evaluationResult [protected]

The result of the model checking evaluation

Definition at line 47 of file ModelCheckerTest.hpp.

Referenced by RunModelCheckingTest(), and RunTest().

#### 7.99.3.3 std::shared\_ptr<[mv::ModelChecker](#)> multiscaletest::ModelCheckerTest::modelChecker [protected]

The specific type of model checker employed

Definition at line 44 of file ModelCheckerTest.hpp.

Referenced by RunTest().

#### 7.99.3.4 std::vector<[mv::SpatialTemporalTrace](#)> multiscaletest::ModelCheckerTest::traces [protected]

The collection of spatio-temporal traces

Definition at line 35 of file ModelCheckerTest.hpp.

Referenced by InitialiseSpatioTemporalTraces(), InitialiseSpatioTemporalTraceWithClusterednessValues(), and RunTest().

**7.99.3.5 TypeSemanticsTable multiscaletest::ModelCheckerTest::typeSemanticsTable [protected]**

The type semantics table mapping semantic criteria values to abstract natural numbers

Definition at line 38 of file ModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

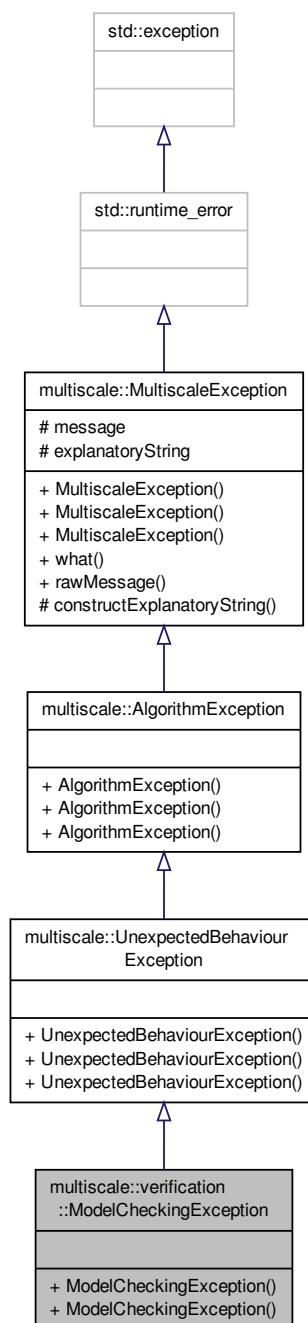
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckerTest.hpp

## 7.100 multiscale::verification::ModelCheckingException Class Reference

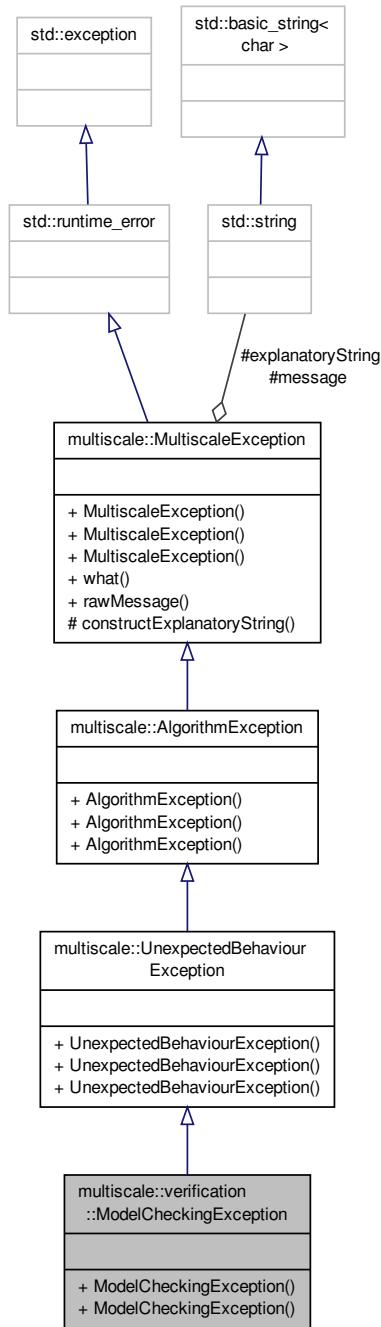
Class for representing a model checking exception.

```
#include <ModelCheckingException.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckingException:



Collaboration diagram for multiscale::verification::ModelCheckingException:



## Public Member Functions

- [ModelCheckingException](#) (const std::string &file, int line, const std::string &msg)
- [ModelCheckingException](#) (const std::string &file, int line, const char \*msg)

## Additional Inherited Members

### 7.100.1 Detailed Description

Class for representing a model checking exception.

Definition at line 12 of file ModelCheckingException.hpp.

### 7.100.2 Constructor & Destructor Documentation

7.100.2.1 multiscale::verification::ModelCheckingException::ModelCheckingException ( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 16 of file ModelCheckingException.hpp.

7.100.2.2 multiscale::verification::ModelCheckingException::ModelCheckingException ( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 21 of file ModelCheckingException.hpp.

The documentation for this class was generated from the following file:

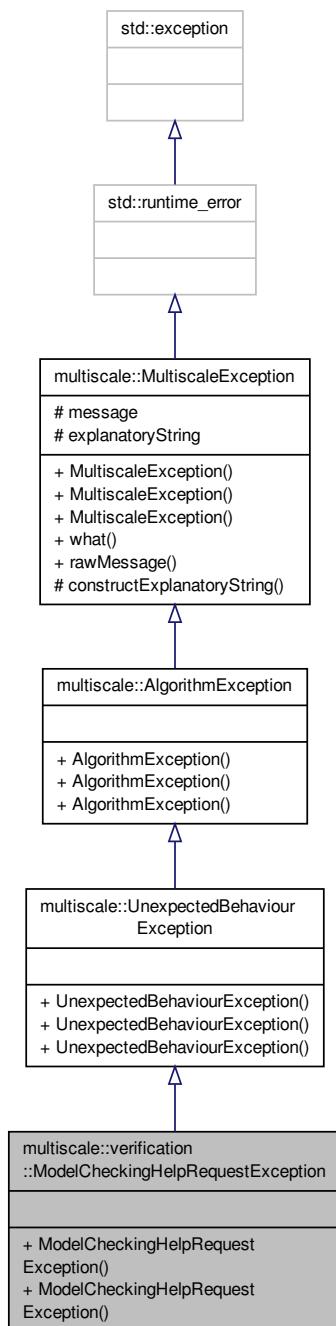
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingException.hpp

## 7.101 multiscale::verification::ModelCheckingHelpRequestException Class Reference

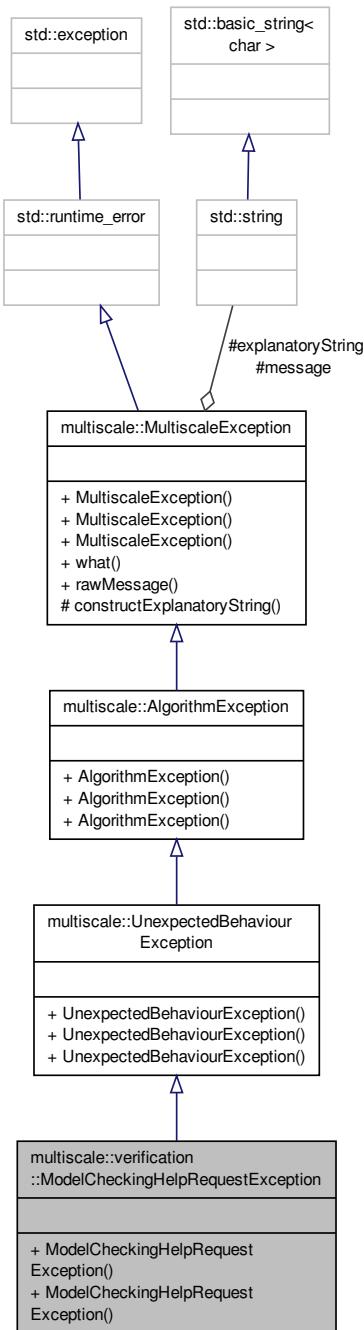
Class for representing a model checking help request exception.

```
#include <ModelCheckingHelpRequestException.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckingHelpRequestException:



Collaboration diagram for multiscale::verification::ModelCheckingHelpRequestException:



## Public Member Functions

- [ModelCheckingHelpRequestException \(const std::string &file, int line, const std::string &msg\)](#)
- [ModelCheckingHelpRequestException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.101.1 Detailed Description

Class for representing a model checking help request exception.

Definition at line 12 of file ModelCheckingHelpRequestException.hpp.

### 7.101.2 Constructor & Destructor Documentation

7.101.2.1 multiscale::verification::ModelCheckingHelpRequestException::ModelCheckingHelpRequestException ( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 16 of file ModelCheckingHelpRequestException.hpp.

7.101.2.2 multiscale::verification::ModelCheckingHelpRequestException::ModelCheckingHelpRequestException ( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 21 of file ModelCheckingHelpRequestException.hpp.

The documentation for this class was generated from the following file:

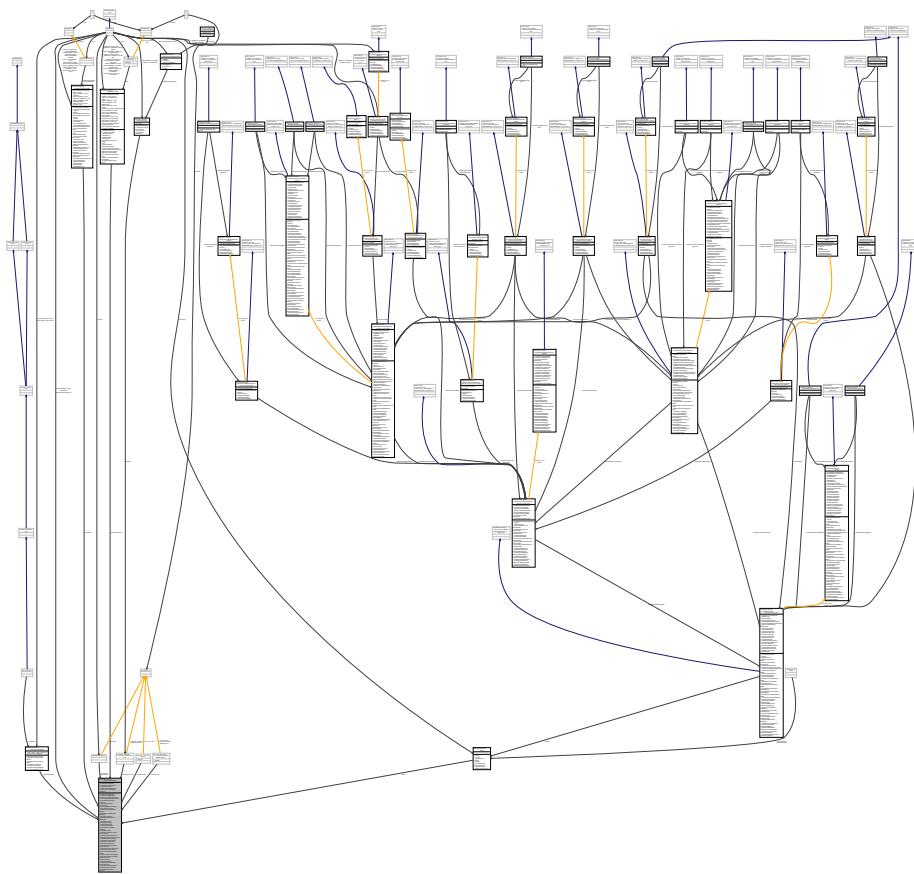
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingHelpRequestException.hpp

## 7.102 multiscale::verification::ModelCheckingManager Class Reference

Class for managing the model checking processes.

#include <ModelCheckingManager.hpp>

Collaboration diagram for multiscale::verification::ModelCheckingManager:



## Public Member Functions

- `ModelCheckingManager` (const std::string &logicPropertiesFilepath, const std::string &tracesFolderPath, unsigned long `extraEvaluationTime`, const std::string &typeSemanticsTableFilepath)
- `~ModelCheckingManager` ()
- void `setExtraEvaluationProgramPath` (const std::string &`extraEvaluationProgramPath`)  
*Set the path of the program which should be executed whenever extra evaluation is required.*
- void `setShouldPrintDetailedEvaluation` (bool `shouldPrintDetailedEvaluation`)  
*Set the flag indicating if the detailed evaluation should be printed.*
- void `runModelCheckingTasks` (const std::shared\_ptr< `ModelCheckerFactory` > &modelCheckerFactory)  
*Run the model checking tasks.*

## Private Member Functions

- void `initialise` (const std::string &logicPropertiesFilepath, unsigned long `extraEvaluationTime`, const std::string &typeSemanticsTableFilepath)  
*Initialise the model checking manager.*
- void `initialiseExtraEvaluationTimeCounters` (unsigned long `extraEvaluationTime`)  
*Initialise the extra evaluation time counters.*
- void `initialiseTypeSemanticsTable` (const std::string &typeSemanticsTableFilepath)  
*Initialise the type semantics table.*
- void `initialiseLogicProperties` (const std::string &logicPropertiesFilepath)  
*Initialise the logic properties using the provided input file.*

- void `runModelCheckingAndOutputResults` (const std::shared\_ptr< ModelCheckerFactory > &modelCheckerFactory)
 

*Run the model checking tasks and output the results.*
- void `parseLogicPropertiesAndPrintMessage` ()
 

*Parse the logic properties and print message informing the user about this.*
- void `parseLogicProperties` ()
 

*Parse the logic properties and create abstract syntax trees.*
- bool `parseLogicPropertyAndPrintMessages` (const std::string &logicProperty)
 

*Parse the logic property and inform the user if the logic property was syntactically correct.*
- bool `parseLogicProperty` (const std::string &logicProperty)
 

*Parse the given logic property and return true if parsing was successful and false otherwise.*
- bool `isValidLogicProperty` (const std::string &logicProperty)
 

*Parse the given logic property and return true if parsing was successful and false otherwise.*
- void `printParsingMessage` (bool isParsigSuccessful)
 

*Print a message stating if the logic property was parsed successfully.*
- void `createModelCheckers` (const std::shared\_ptr< ModelCheckerFactory > &modelCheckerFactory)
 

*Create the model checker instances using the provided model checker factory.*
- void `runModelCheckersAndPrintMessage` ()
 

*Run the model checkers and print a message informing the user about it.*
- void `runModelCheckers` ()
 

*Run the model checkers and verify the logic properties.*
- void `runModelCheckersForCurrentlyExistingTraces` ()
 

*Run the model checkers and verify the logic properties for the currently existing traces.*
- `SpatialTemporalTrace getNextSpatialTemporalTrace` ()
 

*Get the next spatial temporal trace and store its path.*
- void `storeNewSpatialTemporalTracePath` (const std::string &tracePath)
 

*Store new trace path if the shouldPrintDetailedEvaluation flag is set to true.*
- void `createNewEvaluationResults` ()
 

*Create a new vector for storing the evaluation results for the (logic property, new trace) pairs.*
- void `runModelCheckersForTrace` (const SpatialTemporalTrace &trace, bool &continueEvaluation)
 

*Run the model checkers and verify the logic properties considering the given trace.*
- void `runModelCheckerForTrace` (const std::size\_t &modelCheckerIndex, const SpatialTemporalTrace &trace)
 

*Run the model checker for the given trace.*
- void `updateEvaluationResults` (const std::size\_t &modelCheckerIndex, bool evaluationResult)
 

*Update the evaluation results for the given model checker index and result.*
- void `runModelCheckersAndRequestAdditionalTraces` ()
 

*Run the model checkers and request additional traces.*
- void `updateExtraEvaluationStartTime` ()
 

*Set the extra evaluation start time equal to current time.*
- bool `isEvaluationTimeRemaining` ()
 

*Check if there is evaluation time remaining.*
- bool `areUnfinishedModelCheckingTasks` ()
 

*Check if there exist model checkers which require extra traces.*
- void `executeExtraEvaluationProgram` ()
 

*Execute the extra evaluation program for generating potential new traces.*
- void `executeExtraEvaluationProgramAndPrintMessage` ()
 

*Execute the extra evaluation program for generating potential new traces.*
- void `waitForRetry` ()
 

*Wait TRACE\_INPUT\_REFRESH\_TIMEOUT minutes before updating the trace reader.*
- void `updateTraceReader` ()
 

*Update trace reader.*

- void `outputModelCheckersResultsAndPrintMessage ()`  
*Output the model checking results and print the message informing the user about this.*
- void `outputModelCheckersResults ()`  
*Output the model checking results.*
- void `outputModelCheckerResults (const std::shared_ptr< ModelChecker > &modelChecker, const std::string &logicProperty)`  
*Output the model checking results for the given model checker.*
- void `outputDetailedEvaluationResults ()`  
*Output the logic properties detailed evaluation results.*

## Private Attributes

- `Parser parser`
- `std::vector< std::string > logicProperties`
- `std::vector< AbstractSyntaxTree > abstractSyntaxTrees`
- `std::vector< std::string > tracesPaths`
- `LogicPropertyDataReader logicPropertyReader`
- `SpatialTemporalDataReader traceReader`
- `TypeSemanticsTable typeSemanticsTable`
- `std::vector< std::vector< bool > > evaluationResults`
- `std::vector< std::shared_ptr< ModelChecker > > modelCheckers`
- `std::chrono::time_point< std::chrono::system_clock > extraEvaluationStartTime`
- `double extraEvaluationElapsedTime`
- `unsigned long extraEvaluationTime`
- `std::string extraEvaluationProgramPath`
- `bool shouldPrintDetailedEvaluation`

## Static Private Attributes

- `static const unsigned long TRACE_INPUT_REFRESH_TIMEOUT = 30`
- `static const std::string PARSER_EMPTY_LOGIC_PROPERTY = ""`

### 7.102.1 Detailed Description

Class for managing the model checking processes.

Definition at line 25 of file ModelCheckingManager.hpp.

### 7.102.2 Constructor & Destructor Documentation

#### 7.102.2.1 ModelCheckingManager::ModelCheckingManager ( const std::string & logicPropertiesFilepath, const std::string & tracesFolderPath, unsigned long extraEvaluationTime, const std::string & typeSemanticsTableFilepath )

Definition at line 12 of file ModelCheckingManager.cpp.

References initialise().

#### 7.102.2.2 ModelCheckingManager::~ModelCheckingManager ( )

Definition at line 21 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, logicProperties, modelCheckers, and tracesPaths.

### 7.102.3 Member Function Documentation

#### 7.102.3.1 bool ModelCheckingManager::areUnfinishedModelCheckingTasks( ) [private]

Check if there exist model checkers which require extra traces.

Definition at line 254 of file ModelCheckingManager.cpp.

References modelCheckers.

Referenced by runModelCheckers(), and runModelCheckersAndRequestAdditionalTraces().

#### 7.102.3.2 void ModelCheckingManager::createModelCheckers( const std::shared\_ptr< ModelCheckerFactory > &modelCheckerFactory ) [private]

Create the model checker instances using the provided model checker factory.

Each model checker instance verifies one logic property

Parameters

|                            |                                           |
|----------------------------|-------------------------------------------|
| <i>modelCheckerFactory</i> | The factory used to create model checkers |
|----------------------------|-------------------------------------------|

Definition at line 137 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, modelCheckers, and typeSemanticsTable.

Referenced by runModelCheckingAndOutputResults().

#### 7.102.3.3 void ModelCheckingManager::createNewEvaluationResults( ) [private]

Create a new vector for storing the evaluation results for the (logic property, new trace) pairs.

The vector is created only if the shouldPrintDetailedEvaluation flag is set to true

Definition at line 191 of file ModelCheckingManager.cpp.

References evaluationResults, modelCheckers, and shouldPrintDetailedEvaluation.

Referenced by runModelCheckersForCurrentlyExistingTraces().

#### 7.102.3.4 void ModelCheckingManager::executeExtraEvaluationProgram( ) [private]

Execute the extra evaluation program for generating potential new traces.

Definition at line 264 of file ModelCheckingManager.cpp.

References executeExtraEvaluationProgramAndPrintMessage(), and extraEvaluationProgramPath.

Referenced by runModelCheckersAndRequestAdditionalTraces().

#### 7.102.3.5 void ModelCheckingManager::executeExtraEvaluationProgramAndPrintMessage( ) [private]

Execute the extra evaluation program for generating potential new traces.

Execute the extra evaluation program for generating potential new traces and print a message informing the user about this

Definition at line 270 of file ModelCheckingManager.cpp.

References multiscale::OperatingSystem::executeProgram(), extraEvaluationProgramPath, and multiscale::verification::ModelCheckingOutputWriter::printExecuteExtraEvaluationProgramMessage().

Referenced by executeExtraEvaluationProgram().

### 7.102.3.6 SpatialTemporalTrace ModelCheckingManager::getNextSpatialTemporalTrace( ) [private]

Get the next spatial temporal trace and store its path.

Definition at line 170 of file ModelCheckingManager.cpp.

References multiscale::verification::SpatialTemporalDataReader::getNextSpatialTemporalTrace(), multiscale::verification::ModelCheckingOutputWriter::printStartTraceEvaluationMessage(), storeNewSpatialTemporalTracePath(), and traceReader.

Referenced by runModelCheckersForCurrentlyExistingTraces().

### 7.102.3.7 void ModelCheckingManager::initialise( const std::string & logicPropertiesFilepath, unsigned long extraEvaluationTime, const std::string & typeSemanticsTableFilepath ) [private]

Initialise the model checking manager.

Initialise the model checking manager considering the given logic properties input file and extra evaluation time, and print the introduction message

#### Parameters

|                                   |                                                                       |
|-----------------------------------|-----------------------------------------------------------------------|
| <i>logicPropertiesFilepath</i>    | The path to the logic properties input file                           |
| <i>extraEvaluationTime</i>        | The number of extra minutes allocated for evaluating logic properties |
| <i>typeSemanticsTableFilepath</i> | The path to the type semantics table                                  |

Definition at line 40 of file ModelCheckingManager.cpp.

References initialiseExtraEvaluationTimeCounters(), initialiseLogicProperties(), initialiseTypeSemanticsTable(), and shouldPrintDetailedEvaluation.

Referenced by ModelCheckingManager().

### 7.102.3.8 void ModelCheckingManager::initialiseExtraEvaluationTimeCounters( unsigned long extraEvaluationTime ) [private]

Initialise the extra evaluation time counters.

#### Parameters

|                            |                                                                       |
|----------------------------|-----------------------------------------------------------------------|
| <i>extraEvaluationTime</i> | The number of extra minutes allocated for evaluating logic properties |
|----------------------------|-----------------------------------------------------------------------|

Definition at line 50 of file ModelCheckingManager.cpp.

References extraEvaluationElapsed, extraEvaluationStartTime, and extraEvaluationTime.

Referenced by initialise().

### 7.102.3.9 void ModelCheckingManager::initialiseLogicProperties( const std::string & logicPropertiesFilepath ) [private]

Initialise the logic properties using the provided input file.

#### Parameters

|                                    |                                             |
|------------------------------------|---------------------------------------------|
| <i>logicProperties</i><br>Filepath | The path to the logic properties input file |
|------------------------------------|---------------------------------------------|

Definition at line 62 of file ModelCheckingManager.cpp.

References logicProperties, logicPropertyReader, and multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile().

Referenced by initialise().

#### 7.102.3.10 void ModelCheckingManager::initialiseTypeSemanticsTable ( const std::string & *typeSemanticsTableFilepath* ) [private]

Initialise the type semantics table.

##### Parameters

|                                       |                                                 |
|---------------------------------------|-------------------------------------------------|
| <i>typeSemantics</i><br>TableFilepath | The path to the type semantics table input file |
|---------------------------------------|-------------------------------------------------|

Definition at line 56 of file ModelCheckingManager.cpp.

References multiscale::verification::TypeSemanticsTable::readTableFromFile(), and typeSemanticsTable.

Referenced by initialise().

#### 7.102.3.11 bool ModelCheckingManager::isEvaluationTimeRemaining ( ) [private]

Check if there is evaluation time remaining.

Definition at line 245 of file ModelCheckingManager.cpp.

References extraEvaluationElapsedTime, extraEvaluationStartTime, and extraEvaluationTime.

Referenced by runModelCheckersAndRequestAdditionalTraces().

#### 7.102.3.12 bool ModelCheckingManager::isValidLogicProperty ( const std::string & *logicProperty* ) [private]

Parse the given logic property and return true if parsing was successful and false otherwise.

Exceptions are not catched in this method

##### Parameters

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
|----------------------|--------------------------|

Definition at line 115 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, multiscale::verification::Parser::parse(), parser, and multiscale::verification::Parser::setLogicalQuery().

Referenced by parseLogicProperty().

#### 7.102.3.13 void ModelCheckingManager::outputDetailedEvaluationResults ( ) [private]

Output the logic properties detailed evaluation results.

Definition at line 309 of file ModelCheckingManager.cpp.

References evaluationResults, logicProperties, multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResults(), shouldPrintDetailedEvaluation, and tracesPaths.

Referenced by runModelCheckingAndOutputResults().

```
7.102.3.14 void ModelCheckingManager::outputModelCheckerResults (const std::shared_ptr< ModelChecker > &
modelChecker, const std::string & logicProperty) [private]
```

Output the model checking results for the given model checker.

**Parameters**

|                      |                                                        |
|----------------------|--------------------------------------------------------|
| <i>modelChecker</i>  | The given model checker                                |
| <i>logicProperty</i> | The logic property verified by the given model checker |

Definition at line 300 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultMessage().

Referenced by outputModelCheckersResults().

#### 7.102.3.15 void ModelCheckingManager::outputModelCheckersResults( ) [private]

Output the model checking results.

Definition at line 292 of file ModelCheckingManager.cpp.

References logicProperties, modelCheckers, and outputModelCheckerResults().

Referenced by outputModelCheckersResultsAndPrintMessage().

#### 7.102.3.16 void ModelCheckingManager::outputModelCheckersResultsAndPrintMessage( ) [private]

Output the model checking results and print the message informing the user about this.

Definition at line 286 of file ModelCheckingManager.cpp.

References outputModelCheckersResults(), and multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage().

Referenced by runModelCheckingAndOutputResults().

#### 7.102.3.17 void ModelCheckingManager::parseLogicProperties( ) [private]

Parse the logic properties and create abstract syntax trees.

Parse the logic properties and create abstract syntax trees whenever a logic property was successfully parsed

Definition at line 83 of file ModelCheckingManager.cpp.

References logicProperties, and parseLogicPropertyAndPrintMessages().

Referenced by parseLogicPropertiesAndPrintMessage().

#### 7.102.3.18 void ModelCheckingManager::parseLogicPropertiesAndPrintMessage( ) [private]

Parse the logic properties and print message informing the user about this.

Definition at line 75 of file ModelCheckingManager.cpp.

References parseLogicProperties(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage(), and multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesEndMessage().

Referenced by runModelCheckingAndOutputResults().

#### 7.102.3.19 bool ModelCheckingManager::parseLogicProperty( const std::string & *logicProperty* ) [private]

Parse the given logic property and return true if parsing was successful and false otherwise.

Exceptions are catched in this method

## Parameters

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
|----------------------|--------------------------|

Definition at line 105 of file ModelCheckingManager.cpp.

References isValidLogicProperty(), and multiscale::ExceptionHandler::printDetailedErrorMessage().

Referenced by parseLogicPropertyAndPrintMessages().

**7.102.3.20 bool ModelCheckingManager::parseLogicPropertyAndPrintMessages ( const std::string & *logicProperty* ) [private]**

Parse the logic property and inform the user if the logic property was syntactically correct.

## Parameters

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
|----------------------|--------------------------|

Definition at line 95 of file ModelCheckingManager.cpp.

References parseLogicProperty(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage(), and printParsingMessage().

Referenced by parseLogicProperties().

**7.102.3.21 void ModelCheckingManager::printParsingMessage ( bool *isParsingSuccessful* ) [private]**

Print a message stating if the logic property was parsed successfully.

## Parameters

|                            |                                               |
|----------------------------|-----------------------------------------------|
| <i>isParsingSuccessful</i> | Flag indicating if the parsing was successful |
|----------------------------|-----------------------------------------------|

Definition at line 129 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printFailedMessage(), and multiscale::verification::ModelCheckingOutputWriter::printSuccessMessage().

Referenced by parseLogicPropertyAndPrintMessages().

**7.102.3.22 void ModelCheckingManager::runModelCheckerForTrace ( const std::size\_t & *modelCheckerIndex*, const SpatialTemporalTrace & *trace* ) [private]**

Run the model checker for the given trace.

## Parameters

|                          |                                                                        |
|--------------------------|------------------------------------------------------------------------|
| <i>modelCheckerIndex</i> | The index of the model checker inside the collection of model checkers |
| <i>trace</i>             | The given spatial-temporal trace                                       |

Definition at line 211 of file ModelCheckingManager.cpp.

References modelCheckers, shouldPrintDetailedEvaluation, and updateEvaluationResults().

Referenced by runModelCheckersForTrace().

**7.102.3.23 void ModelCheckingManager::runModelCheckers ( ) [private]**

Run the model checkers and verify the logic properties.

Definition at line 151 of file ModelCheckingManager.cpp.

References `areUnfinishedModelCheckingTasks()`, `runModelCheckersAndRequestAdditionalTraces()`, and `runModelCheckersForCurrentlyExistingTraces()`.

Referenced by `runModelCheckersAndPrintMessage()`.

#### 7.102.3.24 void ModelCheckingManager::runModelCheckersAndPrintMessage( ) [private]

Run the model checkers and print a message informing the user about it.

Definition at line 145 of file `ModelCheckingManager.cpp`.

References `multiscale::verification::ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage()`, and `runModelCheckers()`.

Referenced by `runModelCheckingAndOutputResults()`.

#### 7.102.3.25 void ModelCheckingManager::runModelCheckersAndRequestAdditionalTraces( ) [private]

Run the model checkers and request additional traces.

Definition at line 229 of file `ModelCheckingManager.cpp`.

References `areUnfinishedModelCheckingTasks()`, `executeExtraEvaluationProgram()`, `isEvaluationTimeRemaining()`, `runModelCheckersForCurrentlyExistingTraces()`, `updateExtraEvaluationStartTime()`, `updateTraceReader()`, and `waitForRetry()`.

Referenced by `runModelCheckers()`.

#### 7.102.3.26 void ModelCheckingManager::runModelCheckersForCurrentlyExistingTraces( ) [private]

Run the model checkers and verify the logic properties for the currently existing traces.

Definition at line 159 of file `ModelCheckingManager.cpp`.

References `createNewEvaluationResults()`, `getNextSpatialTemporalTrace()`, `multiscale::verification::SpatialTemporalDataReader::hasNext()`, `runModelCheckersForTrace()`, and `traceReader`.

Referenced by `runModelCheckers()`, and `runModelCheckersAndRequestAdditionalTraces()`.

#### 7.102.3.27 void ModelCheckingManager::runModelCheckersForTrace( const SpatialTemporalTrace & trace, bool & continueEvaluation ) [private]

Run the model checkers and verify the logic properties considering the given trace.

If none of the model checkers need additional traces then the `continueEvaluation` flag will be set to false.

##### Parameters

|                                 |                                                                                                                                                                         |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>trace</code>              | The spatial temporal trace used for the logic properties evaluation                                                                                                     |
| <code>continueEvaluation</code> | The flag indicating if there is at least one logic property whose truth value was not determined yet and needs to be evaluated considering more spatial temporal traces |

Definition at line 197 of file `ModelCheckingManager.cpp`.

References `modelCheckers`, and `runModelCheckerForTrace()`.

Referenced by `runModelCheckersForCurrentlyExistingTraces()`.

#### 7.102.3.28 void ModelCheckingManager::runModelCheckingAndOutputResults( const std::shared\_ptr< ModelCheckerFactory > & modelCheckerFactory ) [private]

Run the model checking tasks and output the results.

## Parameters

|                            |                                           |
|----------------------------|-------------------------------------------|
| <i>modelCheckerFactory</i> | The factory used to create model checkers |
|----------------------------|-------------------------------------------|

Definition at line 66 of file ModelCheckingManager.cpp.

References `createModelCheckers()`, `outputDetailedEvaluationResults()`, `outputModelCheckersResultsAndPrintMessage()`, `parseLogicPropertiesAndPrintMessage()`, and `runModelCheckersAndPrintMessage()`.

Referenced by `runModelCheckingTasks()`.

**7.102.3.29 void ModelCheckingManager::runModelCheckingTasks ( const std::shared\_ptr< ModelCheckerFactory > & *modelCheckerFactory* )**

Run the model checking tasks.

## Parameters

|                            |                                           |
|----------------------------|-------------------------------------------|
| <i>modelCheckerFactory</i> | The factory used to create model checkers |
|----------------------------|-------------------------------------------|

Definition at line 36 of file ModelCheckingManager.cpp.

References `runModelCheckingAndOutputResults()`.

**7.102.3.30 void ModelCheckingManager::setExtraEvaluationProgramPath ( const std::string & *extraEvaluationProgramPath* )**

Set the path of the program which should be executed whenever extra evaluation is required.

## Parameters

|                                   |                                                                                  |
|-----------------------------------|----------------------------------------------------------------------------------|
| <i>extraEvaluationProgramPath</i> | The path to the program which will be executed when extra evaluation is required |
|-----------------------------------|----------------------------------------------------------------------------------|

Definition at line 28 of file ModelCheckingManager.cpp.

References `extraEvaluationProgramPath`.

**7.102.3.31 void ModelCheckingManager::setShouldPrintDetailedEvaluation ( bool *shouldPrintDetailedEvaluation* )**

Set the flag indicating if the detailed evaluation should be printed.

## Parameters

|                                      |          |
|--------------------------------------|----------|
| <i>shouldPrintDetailedEvaluation</i> | The flag |
|--------------------------------------|----------|

Definition at line 32 of file ModelCheckingManager.cpp.

References `shouldPrintDetailedEvaluation`.

**7.102.3.32 void ModelCheckingManager::storeNewSpatialTemporalTracePath ( const std::string & *tracePath* ) [private]**

Store new trace path if the `shouldPrintDetailedEvaluation` flag is set to true.

## Parameters

|                  |                       |
|------------------|-----------------------|
| <i>tracePath</i> | The path to the trace |
|------------------|-----------------------|

Definition at line 185 of file ModelCheckingManager.cpp.

References shouldPrintDetailedEvaluation, and tracesPaths.

Referenced by getNextSpatialTemporalTrace().

**7.102.3.33 void ModelCheckingManager::updateEvaluationResults ( const std::size\_t & *modelCheckerIndex*, bool *evaluationResult* ) [private]**

Update the evaluation results for the given model checker index and result.

## Parameters

|                          |                                                                        |
|--------------------------|------------------------------------------------------------------------|
| <i>modelCheckerIndex</i> | The index of the model checker inside the collection of model checkers |
| <i>evaluationResult</i>  | The result of evaluating the model checker for the last trace          |

Definition at line 221 of file ModelCheckingManager.cpp.

References evaluationResults.

Referenced by runModelCheckerForTrace().

**7.102.3.34 void ModelCheckingManager::updateExtraEvaluationStartTime ( ) [private]**

Set the extra evaluation start time equal to current time.

Definition at line 241 of file ModelCheckingManager.cpp.

References extraEvaluationStartTime.

Referenced by runModelCheckersAndRequestAdditionalTraces().

**7.102.3.35 void ModelCheckingManager::updateTraceReader ( ) [private]**

Update trace reader.

Definition at line 282 of file ModelCheckingManager.cpp.

References multiscale::verification::SpatialTemporalDataReader::refresh(), and traceReader.

Referenced by runModelCheckersAndRequestAdditionalTraces().

**7.102.3.36 void ModelCheckingManager::waitBeforeRetry ( ) [private]**

Wait TRACE\_INPUT\_REFRESH\_TIMEOUT minutes before updating the trace reader.

Definition at line 276 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printTimeoutMessage(), and TRACE\_INPUT\_REFRESH\_TIMEOUT.

Referenced by runModelCheckersAndRequestAdditionalTraces().

## 7.102.4 Member Data Documentation

7.102.4.1 `std::vector<AbstractSyntaxTree> multiscale::verification::ModelCheckingManager::abstractSyntaxTrees [private]`

The collection of abstract syntax tree obtained after parsing the logic properties

Definition at line 36 of file ModelCheckingManager.hpp.

Referenced by `createModelCheckers()`, `isValidLogicProperty()`, and `~ModelCheckingManager()`.

7.102.4.2 `std::vector<std::vector<bool>> multiscale::verification::ModelCheckingManager::evaluationResults [private]`

The two-dimensional array storing the evaluation result for each (logic property, trace) pair. A pair of boolean values (`isEvaluated`, `evaluationResult`) is associated to each (logic property, trace) pair

Definition at line 50 of file ModelCheckingManager.hpp.

Referenced by `createNewEvaluationResults()`, `outputDetailedEvaluationResults()`, and `updateEvaluationResults()`.

7.102.4.3 `double multiscale::verification::ModelCheckingManager::extraEvaluationElapsedTime [private]`

The elapsed time for the extra evaluation process expressed in seconds

Definition at line 61 of file ModelCheckingManager.hpp.

Referenced by `initialiseExtraEvaluationTimeCounters()`, and `isEvaluationTimeRemaining()`.

7.102.4.4 `std::string multiscale::verification::ModelCheckingManager::extraEvaluationProgramPath [private]`

The path to the program which should be executed when extra evaluation is required

Definition at line 67 of file ModelCheckingManager.hpp.

Referenced by `executeExtraEvaluationProgram()`, `executeExtraEvaluationProgramAndPrintMessage()`, and `setExtraEvaluationProgramPath()`.

7.102.4.5 `std::chrono::time_point<std::chrono::system_clock> multiscale::verification::ModelCheckingManager::extraEvaluationStartTime [private]`

The start time for the current evaluation process

Definition at line 59 of file ModelCheckingManager.hpp.

Referenced by `initialiseExtraEvaluationTimeCounters()`, `isEvaluationTimeRemaining()`, and `updateExtraEvaluationStartTime()`.

7.102.4.6 `unsigned long multiscale::verification::ModelCheckingManager::extraEvaluationTime [private]`

The number of minutes for which the program waits for new traces to be added to the trace folder

Definition at line 64 of file ModelCheckingManager.hpp.

Referenced by `initialiseExtraEvaluationTimeCounters()`, and `isEvaluationTimeRemaining()`.

7.102.4.7 `std::vector<std::string> multiscale::verification::ModelCheckingManager::logicProperties [private]`

The collection of logic properties

Definition at line 34 of file ModelCheckingManager.hpp.

Referenced by initialiseLogicProperties(), outputDetailedEvaluationResults(), outputModelCheckersResults(), parseLogicProperties(), and ~ModelCheckingManager().

#### 7.102.4.8 LogicPropertyDataReader multiscale::verification::ModelCheckingManager::logicPropertyReader [private]

The logic property reader

Definition at line 42 of file ModelCheckingManager.hpp.

Referenced by initialiseLogicProperties().

#### 7.102.4.9 std::vector<std::shared\_ptr<ModelChecker>> multiscale::verification::ModelCheckingManager::modelCheckers [private]

The collection of model checkers

Definition at line 56 of file ModelCheckingManager.hpp.

Referenced by areUnfinishedModelCheckingTasks(), createModelCheckers(), createNewEvaluationResults(), outputModelCheckersResults(), runModelCheckerForTrace(), runModelCheckersForTrace(), and ~ModelCheckingManager().

#### 7.102.4.10 Parser multiscale::verification::ModelCheckingManager::parser [private]

The parser used to verify if logical properties are syntactically correct

Definition at line 30 of file ModelCheckingManager.hpp.

Referenced by isValidLogicProperty().

#### 7.102.4.11 const std::string ModelCheckingManager::PARSER\_EMPTY\_LOGIC\_PROPERTY = "" [static], [private]

An empty logic property

Definition at line 284 of file ModelCheckingManager.hpp.

#### 7.102.4.12 bool multiscale::verification::ModelCheckingManager::shouldPrintDetailedEvaluation [private]

Flag indicating if detailed evaluation results should be printed

Definition at line 71 of file ModelCheckingManager.hpp.

Referenced by createNewEvaluationResults(), initialise(), outputDetailedEvaluationResults(), runModelCheckerForTrace(), setShouldPrintDetailedEvaluation(), and storeNewSpatialTemporalTracePath().

#### 7.102.4.13 const unsigned long ModelCheckingManager::TRACE\_INPUT\_REFRESH\_TIMEOUT = 30 [static], [private]

The number of seconds for which the manager waits before updating the trace reader

Definition at line 280 of file ModelCheckingManager.hpp.

Referenced by waitBeforeRetry().

#### 7.102.4.14 SpatialTemporalDataReader multiscale::verification::ModelCheckingManager::traceReader [private]

The behaviour/trace reader

Definition at line 44 of file ModelCheckingManager.hpp.

Referenced by getNextSpatialTemporalTrace(), runModelCheckersForCurrentlyExistingTraces(), and updateTraceReader().

#### 7.102.4.15 std::vector<std::string> multiscale::verification::ModelCheckingManager::tracesPaths [private]

The collection of traces paths

Definition at line 39 of file ModelCheckingManager.hpp.

Referenced by outputDetailedEvaluationResults(), storeNewSpatialTemporalTracePath(), and ~ModelCheckingManager().

#### 7.102.4.16 TypeSemanticsTable multiscale::verification::ModelCheckingManager::typeSemanticsTable [private]

The type semantics table

Definition at line 47 of file ModelCheckingManager.hpp.

Referenced by createModelCheckers(), and initialiseTypeSemanticsTable().

The documentation for this class was generated from the following files:

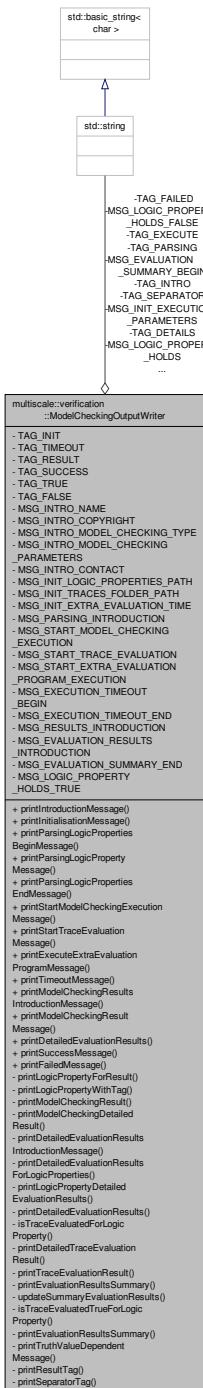
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ModelCheckingManager.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ModelCheckingManager.cpp](#)

## 7.103 multiscale::verification::ModelCheckingOutputWriter Class Reference

Class used to output the model checkers progress.

```
#include <ModelCheckingOutputWriter.hpp>
```

Collaboration diagram for multiscale::verification::ModelCheckingOutputWriter:



## Static Public Member Functions

- static void `printIntroductionMessage` (const `std::string` &modelCheckerType, const `std::string` &modelCheckerParameters)
 

*Print the model checker introduction message considering the given model checker details.*
- static void `printInitialisationMessage` (const `std::string` &logicProperty, const `std::string` &tracesFolderPath, unsigned long extraEvaluationTime)

- static void `printParsingLogicPropertiesBeginMessage ()`

*Print the model checker initialisation message.*
- static void `printParsingLogicPropertyMessage (const std::string &logicProperty)`

*Print an introduction message informing the user that the logic properties will be parsed.*
- static void `printParsingLogicPropertiesEndMessage ()`

*Print a message informing the user which logic property will be parsed.*
- static void `printStartModelCheckingExecutionMessage ()`

*Print a closing message after the logic properties were parsed.*
- static void `printStartTraceEvaluationMessage (const std::string &tracePath)`

*Print a message informing the user that the model checking execution has started.*
- static void `printExecuteExtraEvaluationProgramMessage (const std::string &programPath)`

*Print a message informing the user which trace will be evaluated next by the model checkers.*
- static void `printTimeoutMessage (unsigned long timeOut)`

*Print a message informing the user that the extra evaluation program located at the given path will be executed.*
- static void `printModelCheckingResultsIntroductionMessage ()`

*Print an introduction message informing the user that the model checking results will be displayed.*
- static void `printModelCheckingResultMessage (bool doesPropertyHold, const std::string &detailedResult, const std::string &logicProperty)`

*Print a message with the results of checking if the given property holds.*
- static void `printDetailedEvaluationResults (const std::vector< std::string > &logicProperties, const std::vector< std::string > &tracesPaths, const std::vector< std::vector< bool >> &evaluationResults)`

*Print for each logic property the traces for which the evaluation result was true/false.*
- static void `printSuccessMessage ()`

*Print a success message.*
- static void `printFailedMessage ()`

*Print a fail message.*

## Static Private Member Functions

- static void `printLogicPropertyForResult (const std::string &logicProperty)`

*Print the given logic property in the context of a result message.*
- static void `printLogicPropertyWithTag (const std::string &logicProperty, const std::string &tag)`

*Print the given logic property in the context of the provided tag.*
- static void `printModelCheckingResult (bool doesPropertyHold)`

*Print if the logic property verified by the model checker holds in the context of a result message.*
- static void `printModelCheckingDetailedResult (bool doesPropertyHold, const std::string &detailedResult)`

*Print the detailed result of the model checking procedure.*
- static void `printDetailedEvaluationResultsIntroductionMessage ()`

*Print an introduction message informing the user that the detailed evaluation results will be printed.*
- static void `printDetailedEvaluationResultsForLogicProperties (const std::vector< std::string > &logicProperties, const std::vector< std::string > &tracesPaths, const std::vector< std::vector< bool >> &evaluationResults)`

*Print the detailed evaluation results for the given logic properties and traces.*
- static void `printLogicPropertyDetailedEvaluationResults (const std::size_t &logicPropertyIndex, const std::vector< std::string > &tracesPaths, const std::vector< std::vector< bool >> &evaluationResults)`

*Print the detailed evaluation results for the given logic property.*
- static void `printDetailedEvaluationResults (const std::size_t &logicPropertyIndex, const std::vector< std::string > &tracesPaths, const std::vector< std::vector< bool >> &evaluationResults)`

*Print the detailed evaluation results for the given logic property.*

- static bool `isTraceEvaluatedForLogicProperty` (const std::size\_t &logicPropertyIndex, const std::size\_t &tracePathIndex, const std::vector< std::vector< bool >> &evaluationResults)
 

*Check if the trace was evaluated for the given logic property.*
- static void `printDetailedTraceEvaluationResult` (const std::size\_t &logicPropertyIndex, const std::string &tracePath, const std::size\_t &tracePathIndex, const std::vector< std::vector< bool >> &evaluationResults)
 

*Print the detailed evaluation result for the given logic property and trace.*
- static void `printTraceEvaluationResult` (const std::string &tracePath, bool evaluationResult)
 

*Print the trace path with the associated evaluation result.*
- static void `printEvaluationResultsSummary` (const std::size\_t &logicPropertyIndex, const std::vector< std::string > &tracesPaths, const std::vector< std::vector< bool >> &evaluationResults)
 

*Print the summary of the evaluation results for the given logic property.*
- static void `updateSummaryEvaluationResults` (const std::size\_t &logicPropertyIndex, const std::size\_t &tracePathIndex, const std::vector< std::vector< bool >> &evaluationResults, size\_t &nrOfEvaluatedTraces, size\_t &nrOfTracesEvaluatedTrue)
 

*Update the summary evaluation results considering the logic property, trace and evaluation results.*
- static bool `isTraceEvaluatedTrueForLogicProperty` (const std::size\_t &logicPropertyIndex, const std::size\_t &tracePathIndex, const std::vector< std::vector< bool >> &evaluationResults)
 

*Check if the trace was evaluated to true for the given logic property.*
- static void `printEvaluationResultsSummary` (std::size\_t nrOfTraces, std::size\_t nrOfCorrectTraces)
 

*Print the summary of the evaluation results for the given logic property.*
- static void `printTruthValueDependentMessage` (const std::string &message, const std::string &tag, bool truthValue)
 

*Print a message with the given tag and colour depending on the truth value.*
- static void `printResultTag` ()
 

*Print a line containing a result tag and no content.*
- static void `printSeparatorTag` ()
 

*Print a line containing a separator tag.*

## Static Private Attributes

- static const std::string `TAG_INTRO` = "[ INTRO ]"
- static const std::string `TAG_INIT` = "[ INIT ]"
- static const std::string `TAG_PARSING` = "[ PARSING ]"
- static const std::string `TAG_EXECUTE` = "[ EXECUTE ]"
- static const std::string `TAG_TIMEOUT` = "[ TIMEOUT ]"
- static const std::string `TAG_RESULT` = "[ RESULT ]"
- static const std::string `TAG_DETAILS` = "[ DETAILS ]"
- static const std::string `TAG_SUCCESS` = "[ SUCCESS ]"
- static const std::string `TAG_FAILED` = "[ FAILED ]"
- static const std::string `TAG_TRUE` = "[ TRUE ]"
- static const std::string `TAG_FALSE` = "[ FALSE ]"
- static const std::string `TAG_SEPARATOR` = "=====
- static const std::string `MSG_INTRO_NAME` = "Mule 1.0.337 (Multidimensional multiscale model checker)"
- static const std::string `MSG_INTRO_COPYRIGHT` = "Copyright Ovidiu Pârvu 2014"
- static const std::string `MSG_INTRO_MODEL_CHECKING_TYPE` = "Model checker type: "
- static const std::string `MSG_INTRO_MODEL_CHECKING_PARAMETERS` = "Parameters: "
- static const std::string `MSG_INTRO_CONTACT` = "For more details, recommendations or suggestions feel free to contact me at <ovidiu.parvu[AT]gmail.com>."
- static const std::string `MSG_INIT_EXECUTION_PARAMETERS` = "Multidimensional multiscale model checking input parameters"
- static const std::string `MSG_INIT_LOGIC_PROPERTIES_PATH` = "Logic properties input file: "
- static const std::string `MSG_INIT_TRACES_FOLDER_PATH` = "Spatio-temporal traces input folder: "
- static const std::string `MSG_INIT_EXTRA_EVALUATION_TIME` = "Extra evaluation time (minutes): "

- static const std::string **MSG\_PARSING\_INTRODUCTION** = "I am starting to parse logic properties..."
- static const std::string **MSG\_START\_MODEL\_CHECKING\_EXECUTION** = "I am starting the execution of the model checkers..."
- static const std::string **MSG\_START\_TRACE\_EVALUATION** = "Evaluating the spatio-temporal trace: "
- static const std::string **MSG\_START\_EXTRA\_EVALUATION\_PROGRAM\_EXECUTION** = "I am starting the execution of the extra evaluation program located at the following path: "
- static const std::string **MSG\_EXECUTION\_TIMEOUT\_BEGIN** = "The model checker execution was suspended for "
- static const std::string **MSG\_EXECUTION\_TIMEOUT\_END** = " seconds during which new traces can be provided in the traces input folder."
- static const std::string **MSG\_RESULTS\_INTRODUCTION** = "I have finished evaluating the logic properties and will display the results..."
- static const std::string **MSG\_EVALUATION\_RESULTS\_INTRODUCTION** = "I will display for each logic property which traces evaluated to **TRUE** and which evaluated to **FALSE**..."
- static const std::string **MSG\_EVALUATION\_SUMMARY\_BEGIN** = "/"
- static const std::string **MSG\_EVALUATION\_SUMMARY\_END** = " spatio-temporal traces evaluated to **TRUE**"
- static const std::string **MSG\_LOGIC\_PROPERTY HOLDS** = "The logic property holds: "
- static const std::string **MSG\_LOGIC\_PROPERTY HOLDS TRUE** = "**TRUE**"
- static const std::string **MSG\_LOGIC\_PROPERTY HOLDS FALSE** = "**FALSE**"

### 7.103.1 Detailed Description

Class used to output the model checkers progress.

Definition at line 12 of file ModelCheckingOutputWriter.hpp.

### 7.103.2 Member Function Documentation

**7.103.2.1 bool ModelCheckingOutputWriter::isTraceEvaluatedForLogicProperty ( const std::size\_t & logicPropertyIndex, const std::size\_t & tracePathIndex, const std::vector< std::vector< bool >> & evaluationResults ) [static], [private]**

Check if the trace was evaluated for the given logic property.

Parameters

|                           |                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                   |
| <i>tracePathIndex</i>     | The index of the trace path in the collection of trace paths                                                                                                                                                                                                                                            |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a (logicProperty, trace) pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 211 of file ModelCheckingOutputWriter.cpp.

Referenced by **isTraceEvaluatedTrueForLogicProperty()**, **printDetailedEvaluationResults()**, and **updateSummaryEvaluationResults()**.

**7.103.2.2 bool ModelCheckingOutputWriter::isTraceEvaluatedTrueForLogicProperty ( const std::size\_t & logicPropertyIndex, const std::size\_t & tracePathIndex, const std::vector< std::vector< bool >> & evaluationResults ) [static], [private]**

Check if the trace was evaluated to true for the given logic property.

## Parameters

|                           |                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                                   |
| <i>tracePathIndex</i>     | The index of the trace path in the collection of trace paths                                                                                                                                                                                                                                                            |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a $(\text{logicProperty}, \text{trace})$ pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 250 of file ModelCheckingOutputWriter.cpp.

References [isTraceEvaluatedForLogicProperty\(\)](#).

Referenced by [updateSummaryEvaluationResults\(\)](#).

```
7.103.2.3 void ModelCheckingOutputWriter::printDetailedEvaluationResults (const std::vector< std::string > &
 logicProperties, const std::vector< std::string > & tracesPaths, const std::vector< std::vector< bool >> &
 evaluationResults) [static]
```

Print for each logic property the traces for which the evaluation result was true/false.

## Parameters

|                          |                                                                                                                                                                                                                                                                                                                         |
|--------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicProperties</i>   | The collection of logic properties                                                                                                                                                                                                                                                                                      |
| <i>tracesPaths</i>       | The collection of trace paths                                                                                                                                                                                                                                                                                           |
| <i>evaluationResults</i> | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a $(\text{logicProperty}, \text{trace})$ pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 99 of file ModelCheckingOutputWriter.cpp.

References [printDetailedEvaluationResultsForLogicProperties\(\)](#), and [printDetailedEvaluationResultsIntroductionMessage\(\)](#).

Referenced by [multiscale::verification::ModelCheckingManager::outputDetailedEvaluationResults\(\)](#), and [printLogicPropertyDetailedEvaluationResults\(\)](#).

```
7.103.2.4 void ModelCheckingOutputWriter::printDetailedEvaluationResults (const std::size_t & logicPropertyIndex, const
 std::vector< std::string > & tracesPaths, const std::vector< std::vector< bool >> & evaluationResults)
 [static], [private]
```

Print the detailed evaluation results for the given logic property.

## Parameters

|                           |                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                                   |
| <i>tracesPaths</i>        | The collection of trace paths                                                                                                                                                                                                                                                                                           |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a $(\text{logicProperty}, \text{trace})$ pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 170 of file ModelCheckingOutputWriter.cpp.

References [isTraceEvaluatedForLogicProperty\(\)](#), and [printDetailedTraceEvaluationResult\(\)](#).

```
7.103.2.5 void ModelCheckingOutputWriter::printDetailedEvaluationResultsForLogicProperties (const std::vector< std::string > &
 logicProperties, const std::vector< std::string > & tracesPaths, const std::vector< std::vector< bool >> &
 evaluationResults) [static], [private]
```

Print the detailed evaluation results for the given logic properties and traces.

**Parameters**

|                          |                                                                                                                                                                                                                                                                                                         |
|--------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicProperties</i>   | The collection of logic properties                                                                                                                                                                                                                                                                      |
| <i>tracesPaths</i>       | The collection of trace paths                                                                                                                                                                                                                                                                           |
| <i>evaluationResults</i> | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a (logicProperty, trace) pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 149 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printColouredMessage(), printLogicPropertyDetailedEvaluationResults(), printLogicPropertyWithTag(), printSeparatorTag(), and TAG\_DETAILS.

Referenced by printDetailedEvaluationResults().

**7.103.2.6 void ModelCheckingOutputWriter::printDetailedEvaluationResultsIntroductionMessage( ) [static], [private]**

Print an introduction message informing the user that the detailed evaluation results will be printed.

Definition at line 140 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_EVALUATION\_RESULTS\_INTRODUCTION, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printEmptyLine(), multiscale::ConsolePrinter::printMessageWithColouredTag(), printSeparatorTag(), and TAG\_DETAILS.

Referenced by printDetailedEvaluationResults().

**7.103.2.7 void ModelCheckingOutputWriter::printDetailedTraceEvaluationResult( const std::size\_t & logicPropertyIndex, const std::string & tracePath, const std::size\_t & tracePathIndex, const std::vector< std::vector< bool >> & evaluationResults ) [static], [private]**

Print the detailed evaluation result for the given logic property and trace.

**Parameters**

|                           |                                                                                                                                                                                                                                                                                                         |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                   |
| <i>tracePath</i>          | The path to the spatial temporal trace                                                                                                                                                                                                                                                                  |
| <i>tracePathIndex</i>     | The index of the trace path in the collection of trace paths                                                                                                                                                                                                                                            |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a (logicProperty, trace) pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 231 of file ModelCheckingOutputWriter.cpp.

References printTraceEvaluationResult().

Referenced by printDetailedEvaluationResults().

**7.103.2.8 void ModelCheckingOutputWriter::printEvaluationResultsSummary( const std::size\_t & logicPropertyIndex, const std::vector< std::string > & tracesPaths, const std::vector< std::vector< bool >> & evaluationResults ) [static], [private]**

Print the summary of the evaluation results for the given logic property.

Print a message informing the user how many traces out of the total number of traces evaluated to true for the given logic property.

**Parameters**

|                           |                                                                                                                                                                                                                                                                                                                         |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                                   |
| <i>tracesPaths</i>        | The collection of trace paths                                                                                                                                                                                                                                                                                           |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a $(\text{logicProperty}, \text{trace})$ pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 182 of file ModelCheckingOutputWriter.cpp.

References updateSummaryEvaluationResults().

Referenced by printLogicPropertyDetailedEvaluationResults().

**7.103.2.9 void ModelCheckingOutputWriter::printEvaluationResultsSummary ( std::size\_t *nrOfTraces*, std::size\_t *nrOfCorrectTraces* ) [static], [private]**

Print the summary of the evaluation results for the given logic property.

Print a message informing the user how many traces out of the total number of traces evaluated to true for the given logic property.

**Parameters**

|                          |                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------|
| <i>nrOfTraces</i>        | The total number of traces                                                          |
| <i>nrOfCorrectTraces</i> | The number of traces out of the total number of traces which were evaluated to true |

Definition at line 217 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_EVALUATION\_SUMMARY\_BEGIN, MSG\_EVALUATION\_SUMMARY\_END, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printMessageWithColouredTag(), TAG\_DETAILS, and multiscale::StringManipulator::toString().

**7.103.2.10 void ModelCheckingOutputWriter::printExecuteExtraEvaluationProgramMessage ( const std::string & *programPath* ) [static]**

Print a message informing the user that the extra evaluation program located at the given path will be executed.

**Parameters**

|                    |                                          |
|--------------------|------------------------------------------|
| <i>programPath</i> | The path to the extra evaluation program |
|--------------------|------------------------------------------|

Definition at line 67 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_START\_EXTRA\_EVALUATION\_PROGRAM\_EXECUTION, multiscale::ConsolePrinter::printMessageWithColouredTag(), and TAG\_EXECUTE.

Referenced by multiscale::verification::ModelCheckingManager::executeExtraEvaluationProgramAndPrintMessage().

**7.103.2.11 void ModelCheckingOutputWriter::printFailedMessage ( ) [static]**

Print a fail message.

Definition at line 111 of file ModelCheckingOutputWriter.cpp.

References multiscale::ConsolePrinter::printColouredMessage(), printSeparatorTag(), multiscale::RED, and TAG\_FAILED.

Referenced by multiscale::verification::ModelCheckingManager::printParsingMessage().

7.103.2.12 void ModelCheckingOutputWriter::printInitialisationMessage ( const std::string & *logicProperty*, const std::string & *tracesFolderPath*, unsigned long *extraEvaluationTime* ) [static]

Print the model checker initialisation message.

**Parameters**

|                         |                                                                                                         |
|-------------------------|---------------------------------------------------------------------------------------------------------|
| <i>logicProperty</i>    | The path to the input file containing logic properties                                                  |
| <i>tracesFolderPath</i> | The path to the folder containing the traces                                                            |
| <i>extra</i>            | The number of extra minutes which the application will wait for new traces to be provided and evaluated |

Definition at line 24 of file ModelCheckingOutputWriter.cpp.

References multiscale::CYAN, MSG\_INIT\_EXECUTION\_PARAMETERS, MSG\_INIT\_EXTRA\_EVALUATION\_TIME, MSG\_INIT\_LOGIC\_PROPERTIES\_PATH, MSG\_INIT\_TRACES\_FOLDER\_PATH, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printEmptyLine(), multiscale::ConsolePrinter::printMessageWithColouredTag(), TAG\_INIT, and multiscale::StringManipulator::toString().

Referenced by multiscale::verification::CommandLineModelChecking::printModelCheckingInitialisationMessage().

**7.103.2.13 void ModelCheckingOutputWriter::printIntroductionMessage ( const std::string & *modelCheckerType*, const std::string & *modelCheckerParameters* ) [static]**

Print the model checker introduction message considering the given model checker details.

**Parameters**

|                               |                               |
|-------------------------------|-------------------------------|
| <i>modelCheckerType</i>       | The type of the model checker |
| <i>modelCheckerParameters</i> | The model checking parameters |

Definition at line 8 of file ModelCheckingOutputWriter.cpp.

References multiscale::CYAN, MSG\_INTRO\_CONTACT, MSG\_INTRO\_COPYRIGHT, MSG\_INTRO\_MODE\_L\_CHECKING\_PARAMETERS, MSG\_INTRO\_MODEL\_CHECKING\_TYPE, MSG\_INTRO\_NAME, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printEmptyLine(), multiscale::ConsolePrinter::printMessageWithColouredTag(), and TAG\_INTRO.

Referenced by multiscale::verification::CommandLineModelChecking::printModelCheckingInitialisationMessage().

**7.103.2.14 void ModelCheckingOutputWriter::printLogicPropertyDetailedEvaluationResults ( const std::size\_t & *logicPropertyIndex*, const std::vector< std::string > & *tracesPaths*, const std::vector< std::vector< bool >> & *evaluationResults* ) [static], [private]**

Print the detailed evaluation results for the given logic property.

**Parameters**

|                           |                                                                                                                                                                                                                                                                                    |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i> | The index of the logic property in the collection of logic properties                                                                                                                                                                                                              |
| <i>tracesPaths</i>        | The collection of trace paths                                                                                                                                                                                                                                                      |
| <i>evaluationResults</i>  | The evaluation results (i.e. a two-dimensional array of size  logicProperties  x  2 *  traces   where the first boolean value associated to a (logicProperty, trace) pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |

Definition at line 163 of file ModelCheckingOutputWriter.cpp.

References printDetailedEvaluationResults(), and printEvaluationResultsSummary().

Referenced by printDetailedEvaluationResultsForLogicProperties().

**7.103.2.15 void ModelCheckingOutputWriter::printLogicPropertyForResult ( const std::string & *logicProperty* ) [static], [private]**

Print the given logic property in the context of a result message.

## Parameters

|                      |                    |
|----------------------|--------------------|
| <i>logicProperty</i> | The logic property |
|----------------------|--------------------|

Definition at line 116 of file ModelCheckingOutputWriter.cpp.

References printLogicPropertyWithTag(), and TAG\_RESULT.

Referenced by printModelCheckingResultMessage().

**7.103.2.16 void ModelCheckingOutputWriter::printLogicPropertyWithTag ( const std::string & *logicProperty*, const std::string & *tag* ) [static], [private]**

Print the given logic property in the context of the provided tag.

## Parameters

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
| <i>tag</i>           | The given tag            |

Definition at line 120 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printMessageWithColouredTag(), and multiscale::StringManipulator::trimRight().

Referenced by printDetailedEvaluationResultsForLogicProperties(), and printLogicPropertyForResult().

**7.103.2.17 void ModelCheckingOutputWriter::printModelCheckingDetailedResult ( bool *doesPropertyHold*, const std::string & *detailedResult* ) [static], [private]**

Print the detailed result of the model checking procedure.

Definition at line 135 of file ModelCheckingOutputWriter.cpp.

References printTruthValueDependentMessage(), and TAG\_RESULT.

Referenced by printModelCheckingResultMessage().

**7.103.2.18 void ModelCheckingOutputWriter::printModelCheckingResult ( bool *doesPropertyHold* ) [static], [private]**

Print if the logic property verified by the model checker holds in the context of a result message.

## Parameters

|                         |                                             |
|-------------------------|---------------------------------------------|
| <i>doesPropertyHold</i> | Flag indicating if the logic property holds |
|-------------------------|---------------------------------------------|

Definition at line 126 of file ModelCheckingOutputWriter.cpp.

References MSG\_LOGIC\_PROPERTY HOLDS, MSG\_LOGIC\_PROPERTY HOLDS FALSE, MSG\_LOGIC\_PROPERTY HOLDS TRUE, printTruthValueDependentMessage(), and TAG\_RESULT.

Referenced by printModelCheckingResultMessage().

**7.103.2.19 void ModelCheckingOutputWriter::printModelCheckingResultMessage ( bool *doesPropertyHold*, const std::string & *detailedResult*, const std::string & *logicProperty* ) [static]**

Print a message with the results of checking if the given property holds.

**Parameters**

|                         |                                                                                                                |
|-------------------------|----------------------------------------------------------------------------------------------------------------|
| <i>doesPropertyHold</i> | The flag indicating if the logic property holds (with a given probability and/or confidence)                   |
| <i>detailedResult</i>   | The detailed result report indicating if the logic property holds (with a given probability and/or confidence) |
| <i>logicProperty</i>    | The logic property to be verified                                                                              |

Definition at line 88 of file ModelCheckingOutputWriter.cpp.

References printLogicPropertyForResult(), printModelCheckingDetailedResult(), printModelCheckingResult(), printResultTag(), and printSeparatorTag().

Referenced by multiscale::verification::ModelCheckingManager::outputModelCheckerResults().

**7.103.2.20 void ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage( ) [static]**

Print an introduction message informing the user that the model checking results will be displayed.

Definition at line 79 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_RESULTS\_INTRODUCTION, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printEmptyLine(), multiscale::ConsolePrinter::printMessageWithColouredTag(), printSeparatorTag(), and TAG\_RESULT.

Referenced by multiscale::verification::ModelCheckingManager::outputModelCheckersResultsAndPrintMessage().

**7.103.2.21 void ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage( ) [static]**

Print an introduction message informing the user that the logic properties will be parsed.

Definition at line 41 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_PARSING\_INTRODUCTION, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printMessageWithColouredTag(), printSeparatorTag(), and TAG\_PARSING.

Referenced by multiscale::verification::ModelCheckingManager::parseLogicPropertiesAndPrintMessage().

**7.103.2.22 void ModelCheckingOutputWriter::printParsingLogicPropertiesEndMessage( ) [static]**

Print a closing message after the logic properties were parsed.

Definition at line 52 of file ModelCheckingOutputWriter.cpp.

References multiscale::ConsolePrinter::printEmptyLine().

Referenced by multiscale::verification::ModelCheckingManager::parseLogicPropertiesAndPrintMessage().

**7.103.2.23 void ModelCheckingOutputWriter::printParsingLogicPropertyMessage( const std::string & logicProperty ) [static]**

Print a message informing the user which logic property will be parsed.

**Parameters**

|                      |                          |
|----------------------|--------------------------|
| <i>logicProperty</i> | The given logic property |
|----------------------|--------------------------|

Definition at line 48 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printMessageWithColouredTag(), TAG\_PARSING, and multiscale::StringManipulator::trimRight().

Referenced by multiscale::verification::ModelCheckingManager::parseLogicPropertyAndPrintMessages().

**7.103.2.24 void ModelCheckingOutputWriter::printResultTag( ) [static], [private]**

Print a line containing a result tag and no content.

Definition at line 270 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printColouredMessage(), and TAG\_RESULT.

Referenced by printModelCheckingResultMessage().

**7.103.2.25 void ModelCheckingOutputWriter::printSeparatorTag( ) [static], [private]**

Print a line containing a separator tag.

Definition at line 274 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printColouredMessage(), and TAG\_SEPARATOR.

Referenced by printDetailedEvaluationResultsForLogicProperties(), printDetailedEvaluationResultsIntroductionMessage(), printFailedMessage(), printModelCheckingResultMessage(), printModelCheckingResultsIntroductionMessage(), printParsingLogicPropertiesBeginMessage(), printStartModelCheckingExecutionMessage(), and printSuccessMessage().

**7.103.2.26 void ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage( ) [static]**

Print a message informing the user that the model checking execution has started.

Definition at line 57 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_START\_MODEL\_CHECKING\_EXECUTION, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printMessageWithColouredTag(), printSeparatorTag(), and TAG\_EXECUTE.

Referenced by multiscale::verification::ModelCheckingManager::runModelCheckersAndPrintMessage().

**7.103.2.27 void ModelCheckingOutputWriter::printStartTraceEvaluationMessage( const std::string & tracePath ) [static]**

Print a message informing the user which trace will be evaluated next by the model checkers.

**Parameters**

|                  |                                        |
|------------------|----------------------------------------|
| <i>tracePath</i> | The path to the spatial-temporal trace |
|------------------|----------------------------------------|

Definition at line 63 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_START\_TRACE\_EVALUATION, multiscale::ConsolePrinter::printMessageWithColouredTag(), and TAG\_EXECUTE.

Referenced by multiscale::verification::ModelCheckingManager::getNextSpatialTemporalTrace().

**7.103.2.28 void ModelCheckingOutputWriter::printSuccessMessage( ) [static]**

Print a success message.

Definition at line 106 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printColouredMessage(), printSeparatorTag(), and TAG\_SUCCESS.

Referenced by multiscale::verification::ModelCheckingManager::printParsingMessage().

**7.103.2.29 void ModelCheckingOutputWriter::printTimeoutMessage ( unsigned long *timeOut* ) [static]**

Print a message informing the user that the model checking execution is suspended for timeOut seconds.

Additionally let the user know that the list of traces is updated after the timeout

**Parameters**

|                |                   |
|----------------|-------------------|
| <i>timeOut</i> | The timeout value |
|----------------|-------------------|

Definition at line 72 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, MSG\_EXECUTION\_TIMEOUT\_BEGIN, MSG\_EXECUTION\_TIMEOUT\_END, multiscale::ConsolePrinter::printMessageWithColouredTag(), TAG\_TIMEOUT, and multiscale::StringManipulator::toString().

Referenced by multiscale::verification::ModelCheckingManager::waitBeforeRetry().

**7.103.2.30 void ModelCheckingOutputWriter::printTraceEvaluationResult ( const std::string & *tracePath*, bool *evaluationResult* ) [static], [private]**

Print the trace path with the associated evaluation result.

**Parameters**

|                         |                                        |
|-------------------------|----------------------------------------|
| <i>tracePath</i>        | The path to the spatial temporal trace |
| <i>evaluationResult</i> | The evaluation result                  |

Definition at line 242 of file ModelCheckingOutputWriter.cpp.

References printTruthValueDependentMessage(), TAG\_FALSE, and TAG\_TRUE.

Referenced by printDetailedTraceEvaluationResult().

**7.103.2.31 void ModelCheckingOutputWriter::printTruthValueDependentMessage ( const std::string & *message*, const std::string & *tag*, bool *truthValue* ) [static], [private]**

Print a message with the given tag and colour depending on the truth value.

If the truthValue is true then the tag colour is green, otherwise red

**Parameters**

|                   |                                                       |
|-------------------|-------------------------------------------------------|
| <i>message</i>    | The given message                                     |
| <i>tag</i>        | The given tag                                         |
| <i>truthValue</i> | Boolean flag depending on which the tag colour is set |

Definition at line 260 of file ModelCheckingOutputWriter.cpp.

References multiscale::GREEN, multiscale::ConsolePrinter::printMessageWithColouredTag(), and multiscale::RED.

Referenced by printModelCheckingDetailedResult(), printModelCheckingResult(), and printTraceEvaluationResult().

**7.103.2.32 void ModelCheckingOutputWriter::updateSummaryEvaluationResults ( const std::size\_t & *logicPropertyIndex*, const std::size\_t & *tracePathIndex*, const std::vector< std::vector< bool >> & *evaluationResults*, size\_t & *nrOfEvaluatedTraces*, size\_t & *nrOfTracesEvaluatedTrue* ) [static], [private]**

Update the summary evaluation results considering the logic property, trace and evaluation results.

**Parameters**

|                                |                                                                                                                                                                                                                                                                                                                         |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>logicPropertyIndex</i>      | The index of the logic property in the collection of logic properties                                                                                                                                                                                                                                                   |
| <i>tracePathIndex</i>          | The index of the trace path in the collection of trace paths                                                                                                                                                                                                                                                            |
| <i>evaluationResults</i>       | The evaluation results (i.e. a two-dimensional array of size $ \text{logicProperties}  \times  2 *  \text{traces}  $ where the first boolean value associated to a $(\text{logicProperty}, \text{trace})$ pair states if the logic property was evaluated for that trace and the second one stores the evaluation value |
| <i>nrOfEvaluatedTraces</i>     | The number of evaluated traces                                                                                                                                                                                                                                                                                          |
| <i>nrOfTracesEvaluatedTrue</i> | The number of traces evaluated true                                                                                                                                                                                                                                                                                     |

Definition at line 197 of file ModelCheckingOutputWriter.cpp.

References isTraceEvaluatedForLogicProperty(), and isTraceEvaluatedTrueForLogicProperty().

Referenced by printEvaluationResultsSummary().

### 7.103.3 Member Data Documentation

7.103.3.1 `const std::string ModelCheckingOutputWriter::MSG_EVALUATION_RESULTS_INTRODUCTION = "I will display for each logic property which traces evaluated to TRUE and which evaluated to FALSE..." [static], [private]`

Definition at line 290 of file ModelCheckingOutputWriter.hpp.

Referenced by printDetailedEvaluationResultsIntroductionMessage().

7.103.3.2 `const std::string ModelCheckingOutputWriter::MSG_EVALUATION_SUMMARY_BEGIN = "/" [static], [private]`

Definition at line 291 of file ModelCheckingOutputWriter.hpp.

Referenced by printEvaluationResultsSummary().

7.103.3.3 `const std::string ModelCheckingOutputWriter::MSG_EVALUATION_SUMMARY_END = " spatio-temporal traces evaluated to TRUE" [static], [private]`

Definition at line 292 of file ModelCheckingOutputWriter.hpp.

Referenced by printEvaluationResultsSummary().

7.103.3.4 `const std::string ModelCheckingOutputWriter::MSG_EXECUTION_TIMEOUT_BEGIN = "The model checker execution was suspended for" [static], [private]`

Definition at line 285 of file ModelCheckingOutputWriter.hpp.

Referenced by printTimeoutMessage().

7.103.3.5 `const std::string ModelCheckingOutputWriter::MSG_EXECUTION_TIMEOUT_END = " seconds during which new traces can be provided in the traces input folder." [static], [private]`

Definition at line 286 of file ModelCheckingOutputWriter.hpp.

Referenced by printTimeoutMessage().

```
7.103.3.6 const std::string ModelCheckingOutputWriter::MSG_INIT_EXECUTION_PARAMETERS = "Multidimensional multiscale
model checking input parameters" [static], [private]
```

Definition at line 275 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

```
7.103.3.7 const std::string ModelCheckingOutputWriter::MSG_INIT_EXTRA_EVALUATION_TIME = "Extra evaluation time
(minutes):" [static], [private]
```

Definition at line 278 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

```
7.103.3.8 const std::string ModelCheckingOutputWriter::MSG_INIT_LOGIC_PROPERTIES_PATH = "Logic properties input file:"
[static], [private]
```

Definition at line 276 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

```
7.103.3.9 const std::string ModelCheckingOutputWriter::MSG_INIT_TRACES_FOLDER_PATH = "Spatio-temporal traces input
folder:" [static], [private]
```

Definition at line 277 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

```
7.103.3.10 const std::string ModelCheckingOutputWriter::MSG_INTRO_CONTACT = "For more details, recommendations or
suggestions feel free to contact me at <ovidiu.parvu[AT]gmail.com>." [static], [private]
```

Definition at line 273 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.103.3.11 const std::string ModelCheckingOutputWriter::MSG_INTRO_COPYRIGHT = "Copyright Ovidiu Pârvu 2014"
[static], [private]
```

Definition at line 270 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.103.3.12 const std::string ModelCheckingOutputWriter::MSG_INTRO_MODEL_CHECKING_PARAMETERS = "Parameters:"
[static], [private]
```

Definition at line 272 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.103.3.13 const std::string ModelCheckingOutputWriter::MSG_INTRO_MODEL_CHECKING_TYPE = "Model checker type:"
[static], [private]
```

Definition at line 271 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.103.3.14 const std::string ModelCheckingOutputWriter::MSG_INTRO_NAME = "Mule 1.0.337 (Multidimensional multiscale
model checker)" [static], [private]
```

Definition at line 269 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.103.3.15 const std::string ModelCheckingOutputWriter::MSG_LOGIC_PROPERTY HOLDS = "The logic property holds:"
[static], [private]
```

Definition at line 294 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResult().

```
7.103.3.16 const std::string ModelCheckingOutputWriter::MSG_LOGIC_PROPERTY HOLDS FALSE = "FALSE" [static],
[private]
```

Definition at line 296 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResult().

```
7.103.3.17 const std::string ModelCheckingOutputWriter::MSG_LOGIC_PROPERTY HOLDS TRUE = "TRUE" [static],
[private]
```

Definition at line 295 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResult().

```
7.103.3.18 const std::string ModelCheckingOutputWriter::MSG_PARSING INTRODUCTION = "I am starting to parse logic
properties..." [static], [private]
```

Definition at line 280 of file ModelCheckingOutputWriter.hpp.

Referenced by printParsingLogicPropertiesBeginMessage().

```
7.103.3.19 const std::string ModelCheckingOutputWriter::MSG RESULTS INTRODUCTION = "I have finished evaluating the
logic properties and will display the results..." [static], [private]
```

Definition at line 288 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResultsIntroductionMessage().

```
7.103.3.20 const std::string ModelCheckingOutputWriter::MSG START EXTRA EVALUATION PROGRAM EXECUTION = "I am
starting the execution of the extra evaluation program located at the following path:" [static], [private]
```

Definition at line 284 of file ModelCheckingOutputWriter.hpp.

Referenced by printExecuteExtraEvaluationProgramMessage().

```
7.103.3.21 const std::string ModelCheckingOutputWriter::MSG START MODEL CHECKING EXECUTION = "I am starting the
execution of the model checkers..." [static], [private]
```

Definition at line 282 of file ModelCheckingOutputWriter.hpp.

Referenced by printStartModelCheckingExecutionMessage().

7.103.3.22 `const std::string ModelCheckingOutputWriter::MSG_START_TRACE_EVALUATION = "Evaluating the spatio-temporal trace: " [static], [private]`

Definition at line 283 of file ModelCheckingOutputWriter.hpp.

Referenced by printStartTraceEvaluationMessage().

7.103.3.23 `const std::string ModelCheckingOutputWriter::TAG_DETAILS = "[ DETAILS ]" [static], [private]`

Definition at line 262 of file ModelCheckingOutputWriter.hpp.

Referenced by printDetailedEvaluationResultsForLogicProperties(), printDetailedEvaluationResultsIntroductionMessage(), and printEvaluationResultsSummary().

7.103.3.24 `const std::string ModelCheckingOutputWriter::TAG_EXECUTE = "[ EXECUTE ]" [static], [private]`

Definition at line 259 of file ModelCheckingOutputWriter.hpp.

Referenced by printExecuteExtraEvaluationProgramMessage(), printStartModelCheckingExecutionMessage(), and printStartTraceEvaluationMessage().

7.103.3.25 `const std::string ModelCheckingOutputWriter::TAG_FAILED = "[ FAILED ]" [static], [private]`

Definition at line 264 of file ModelCheckingOutputWriter.hpp.

Referenced by printFailedMessage().

7.103.3.26 `const std::string ModelCheckingOutputWriter::TAG_FALSE = "[ FALSE ]" [static], [private]`

Definition at line 266 of file ModelCheckingOutputWriter.hpp.

Referenced by printTraceEvaluationResult().

7.103.3.27 `const std::string ModelCheckingOutputWriter::TAG_INIT = "[ INIT ]" [static], [private]`

Definition at line 257 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

7.103.3.28 `const std::string ModelCheckingOutputWriter::TAG_INTRO = "[ INTRO ]" [static], [private]`

Definition at line 256 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

7.103.3.29 `const std::string ModelCheckingOutputWriter::TAG_PARSING = "[ PARSING ]" [static], [private]`

Definition at line 258 of file ModelCheckingOutputWriter.hpp.

Referenced by printParsingLogicPropertiesBeginMessage(), and printParsingLogicPropertyMessage().

7.103.3.30 `const std::string ModelCheckingOutputWriter::TAG_RESULT = "[ RESULT ]" [static], [private]`

Definition at line 261 of file ModelCheckingOutputWriter.hpp.

Referenced by printLogicPropertyForResult(), printModelCheckingDetailedResult(), printModelCheckingResult(), printModelCheckingResultsIntroductionMessage(), and printResultTag().

7.103.3.31 const std::string ModelCheckingOutputWriter::TAG\_SEPARATOR = "[=====]" [static], [private]

Definition at line 267 of file ModelCheckingOutputWriter.hpp.

Referenced by printSeparatorTag().

7.103.3.32 const std::string ModelCheckingOutputWriter::TAG\_SUCCESS = "[ SUCCESS ]" [static], [private]

Definition at line 263 of file ModelCheckingOutputWriter.hpp.

Referenced by printSuccessMessage().

7.103.3.33 const std::string ModelCheckingOutputWriter::TAG\_TIMEOUT = "[ TIMEOUT ]" [static], [private]

Definition at line 260 of file ModelCheckingOutputWriter.hpp.

Referenced by printTimeoutMessage().

7.103.3.34 const std::string ModelCheckingOutputWriter::TAG\_TRUE = "[ TRUE ]" [static], [private]

Definition at line 265 of file ModelCheckingOutputWriter.hpp.

Referenced by printTraceEvaluationResult().

The documentation for this class was generated from the following files:

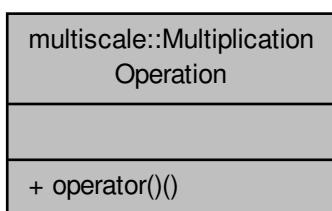
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.cpp

## 7.104 multiscale::MultiplicationOperation Class Reference

Functor representing a multiplication operation.

```
#include <Numeric.hpp>
```

Collaboration diagram for multiscale::MultiplicationOperation:



## Public Member Functions

- template<typename Operand >  
Operand **operator()** (Operand operand1, Operand operand2) const  
*Multiply the two operands.*

### 7.104.1 Detailed Description

Functor representing a multiplication operation.

Definition at line 50 of file Numeric.hpp.

### 7.104.2 Member Function Documentation

- template<typename Operand > Operand multiscale::MultiplicationOperation::operator() ( Operand *operand1*, Operand *operand2* ) const [inline]

Multiply the two operands.

#### Parameters

|                 |                    |
|-----------------|--------------------|
| <i>operand1</i> | The first operand  |
| <i>operand2</i> | The second operand |

Definition at line 60 of file Numeric.hpp.

The documentation for this class was generated from the following file:

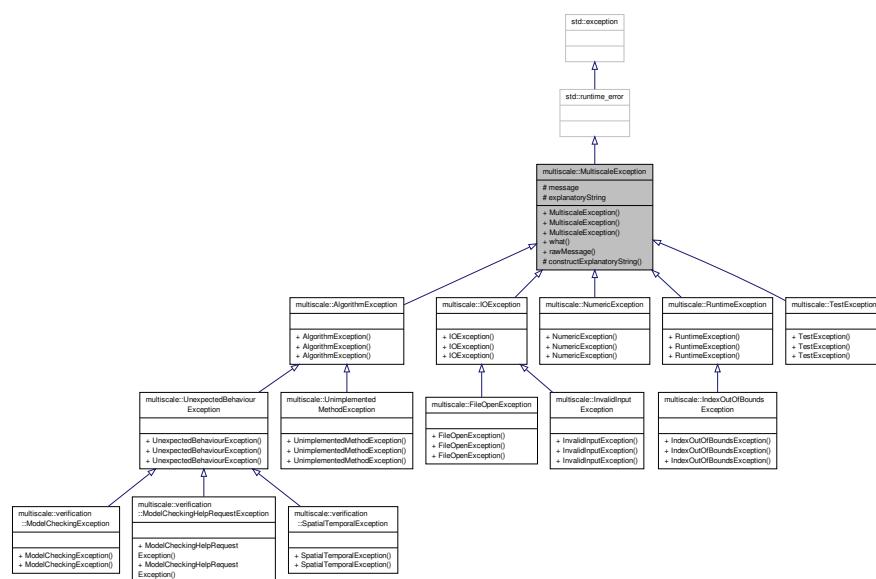
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Numeric.hpp

## 7.105 multiscale::MultiscaleException Class Reference

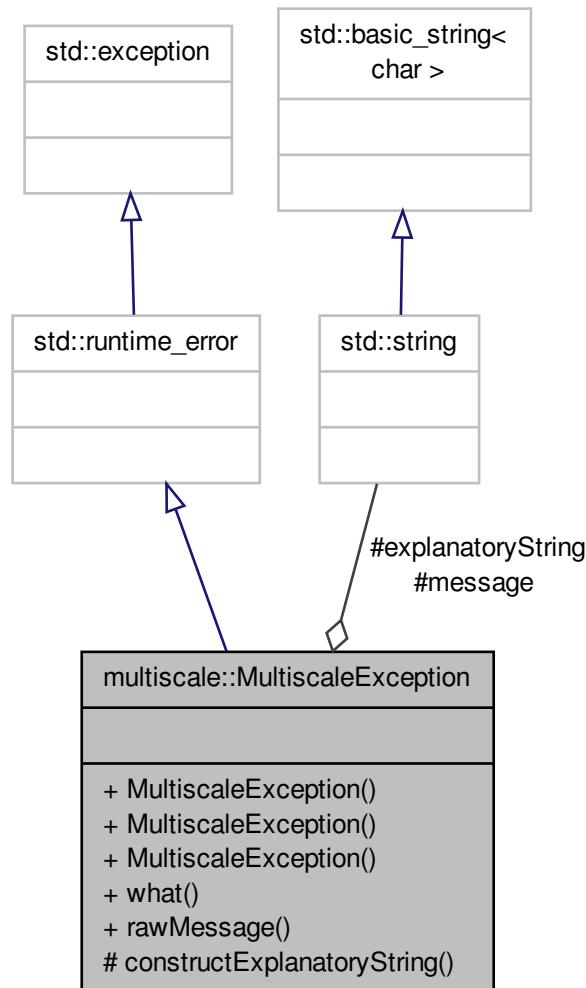
Parent exception class for the project.

```
#include <MultiscaleException.hpp>
```

Inheritance diagram for multiscale::MultiscaleException:



Collaboration diagram for multiscale::MultiscaleException:



## Public Member Functions

- `MultiscaleException ()`
- `MultiscaleException (const std::string &file, int line, const std::string &msg)`
- `MultiscaleException (const std::string &file, int line, const char *msg)`
- `const char * what () const noexceptoverride`
  - Returns an explanatory string.*
- `std::string rawMessage () const noexcept`
  - Return the raw message of the exception.*

## Protected Member Functions

- `template<typename T > void constructExplanatoryString (const std::string &file, int line, T msg)`
  - Construct the explanatory string.*

## Protected Attributes

- std::string [message](#)
- std::string [explanatoryString](#)

### 7.105.1 Detailed Description

Parent exception class for the project.

Definition at line 18 of file MultiscaleException.hpp.

### 7.105.2 Constructor & Destructor Documentation

#### 7.105.2.1 multiscale::MultiscaleException::MultiscaleException( ) [inline]

Definition at line 27 of file MultiscaleException.hpp.

#### 7.105.2.2 multiscale::MultiscaleException::MultiscaleException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 29 of file MultiscaleException.hpp.

#### 7.105.2.3 multiscale::MultiscaleException::MultiscaleException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 31 of file MultiscaleException.hpp.

### 7.105.3 Member Function Documentation

#### 7.105.3.1 template<typename T> void multiscale::MultiscaleException::constructExplanatoryString( const std::string & *file*, int *line*, T *msg* ) [inline], [protected]

Construct the explanatory string.

##### Parameters

|             |                                      |
|-------------|--------------------------------------|
| <i>file</i> | File where the error occurred        |
| <i>line</i> | Line number where the error occurred |
| <i>msg</i>  | Error message                        |

Definition at line 53 of file MultiscaleException.hpp.

#### 7.105.3.2 std::string multiscale::MultiscaleException::rawMessage( ) const [inline], [noexcept]

Return the raw message of the exception.

Definition at line 40 of file MultiscaleException.hpp.

References message.

Referenced by multiscale::OperatingSystem::executeProgram(), multiscale::verification::LogicPropertyVisitor<::operator()(), multiscale::ExceptionHandler::printRawErrorMessage(), and multiscale::XmlValidator::verifyIfValid<XmlAttribute().

**7.105.3.3** `const char* multiscale::MultiscaleException::what() const` [inline], [override], [noexcept]

Returns an explanatory string.

Definition at line 35 of file MultiscaleException.hpp.

#### **7.105.4 Member Data Documentation**

**7.105.4.1 std::string multiscale::MultiscaleException::explanatoryString [protected]**

## User friendly exception message

Definition at line 23 of file MultiscaleException.hpp.

#### 7.105.4.2 std::string multiscale::MultiscaleException::message [protected]

## The raw message of the exception

Definition at line 22 of file MultiscaleException.hpp.

Referenced by rawMessage().

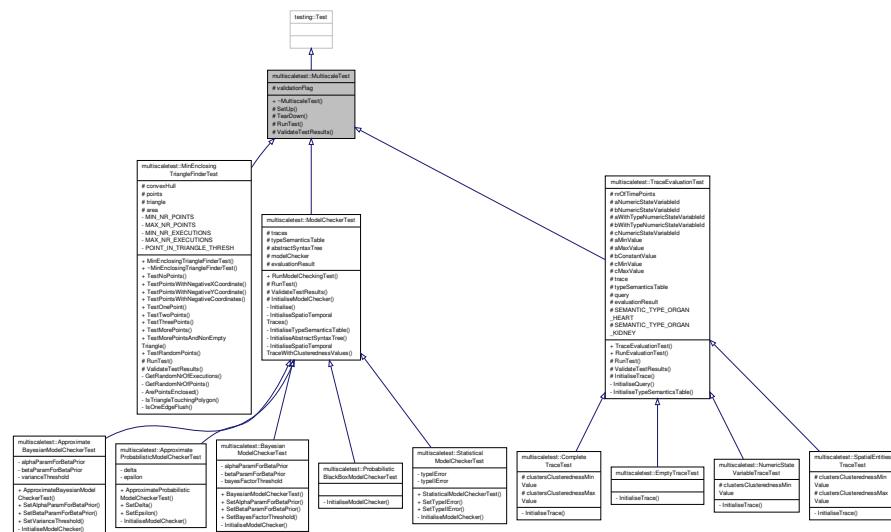
The documentation for this class was generated from the following file:

- `/home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/MultiscaleException.hpp`

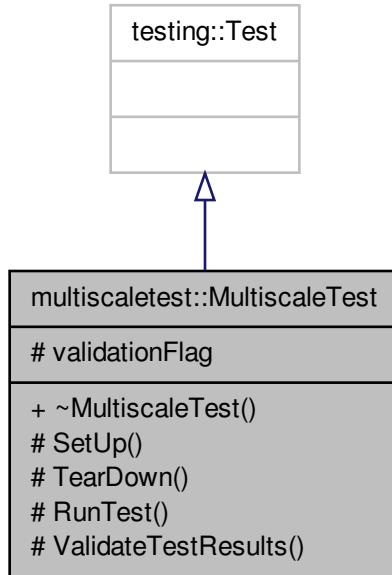
## 7.106 multiscaletest::MultiscaleTest Class Reference

```
#include <MultiscaleTest.hpp>
```

## Inheritance diagram for multiscaletest::MultiscaleTest:



Collaboration diagram for multiscaletest::MultiscaleTest:



## Public Member Functions

- virtual [~MultiscaleTest \(\)](#)

## Protected Member Functions

- virtual void [SetUp \(\)](#)
- virtual void [TearDown \(\)](#)
- virtual void [RunTest \(\)=0](#)  
*Run the test.*
- virtual void [ValidateTestResults \(\)=0](#)  
*Validate the results of the test.*

## Protected Attributes

- bool [validationFlag](#)

### 7.106.1 Detailed Description

Definition at line 9 of file MultiscaleTest.hpp.

### 7.106.2 Constructor & Destructor Documentation

7.106.2.1 virtual multiscaletest::MultiscaleTest::~MultiscaleTest( ) [inline], [virtual]

Definition at line 17 of file MultiscaleTest.hpp.

### 7.106.3 Member Function Documentation

7.106.3.1 virtual void multiscaletest::MultiscaleTest::RunTest( ) [protected], [pure virtual]

Run the test.

Implemented in [multiscaletest::TraceEvaluationTest](#), [multiscaletest::MinEnclosingTriangleFinderTest](#), and [multiscaletest::ModelCheckerTest](#).

7.106.3.2 virtual void multiscaletest::MultiscaleTest::SetUp( ) [inline], [protected], [virtual]

Definition at line 21 of file MultiscaleTest.hpp.

7.106.3.3 virtual void multiscaletest::MultiscaleTest::TearDown( ) [inline], [protected], [virtual]

Definition at line 22 of file MultiscaleTest.hpp.

7.106.3.4 virtual void multiscaletest::MultiscaleTest::ValidateTestResults( ) [protected], [pure virtual]

Validate the results of the test.

Implemented in [multiscaletest::TraceEvaluationTest](#), [multiscaletest::MinEnclosingTriangleFinderTest](#), and [multiscaletest::ModelCheckerTest](#).

### 7.106.4 Member Data Documentation

7.106.4.1 bool multiscaletest::MultiscaleTest::validationFlag [protected]

Flag indicating if the test results are valid

Definition at line 13 of file MultiscaleTest.hpp.

The documentation for this class was generated from the following file:

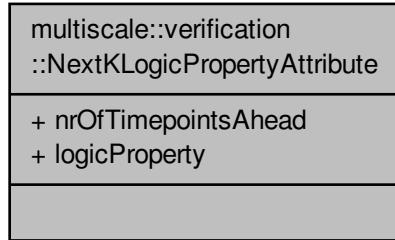
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/[MultiscaleTest.hpp](#)

## 7.107 multiscale::verification::NextKLogicPropertyAttribute Class Reference

Class for representing a "next K" logic property attribute.

```
#include <NextKLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NextKLogicPropertyAttribute:



## Public Attributes

- unsigned long [nrOfTimepointsAhead](#)
- [LogicPropertyAttributeType](#) [logicProperty](#)

### 7.107.1 Detailed Description

Class for representing a "next K" logic property attribute.

Definition at line 14 of file [NextKLogicPropertyAttribute.hpp](#).

### 7.107.2 Member Data Documentation

#### 7.107.2.1 [LogicPropertyAttributeType](#) [multiscale::verification::NextKLogicPropertyAttribute::logicProperty](#)

The logic property following the "next" operator

Definition at line 19 of file [NextKLogicPropertyAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::evaluateNextKLogicProperty\(\)](#).

#### 7.107.2.2 [unsigned long](#) [multiscale::verification::NextKLogicPropertyAttribute::nrOfTimepointsAhead](#)

The number of timepoints ahead "K"

Definition at line 18 of file [NextKLogicPropertyAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::evaluateNextKLogicProperty\(\)](#).

The documentation for this class was generated from the following file:

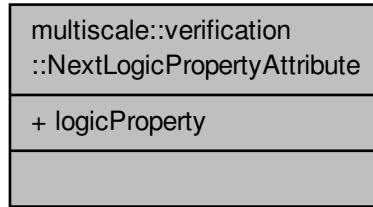
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[NextKLogicPropertyAttribute.hpp](#)

## 7.108 [multiscale::verification::NextLogicPropertyAttribute](#) Class Reference

Class for representing a "next" logic property attribute.

```
#include <NextLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NextLogicPropertyAttribute:



## Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.108.1 Detailed Description

Class for representing a "next" logic property attribute.

Definition at line 14 of file [NextLogicPropertyAttribute.hpp](#).

### 7.108.2 Member Data Documentation

#### 7.108.2.1 LogicPropertyAttributeType multiscale::verification::NextLogicPropertyAttribute::logicProperty

The logic property attribute following the "next" operator

Definition at line 18 of file [NextLogicPropertyAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::evaluateNextLogicProperty\(\)](#).

The documentation for this class was generated from the following file:

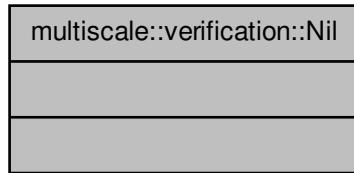
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextLogicPropertyAttribute.hpp](#)

## 7.109 multiscale::verification::Nil Class Reference

A class used to avoid run-time errors when defining a variant type.

```
#include <Nil.hpp>
```

Collaboration diagram for multiscale::verification::Nil:



### 7.109.1 Detailed Description

A class used to avoid run-time errors when defining a variant type.

When defining a variable of variant type "V" the default constructor of the first type within "V" is called. In order to avoid run-time errors this type needs to be different from the boost::recursive\_wrapper<T> type. In variants where all types are boost::recursive\_wrapper<T\_i> the [Nil](#) type can be added before them in order to avoid the potential run-time errors.

Definition at line 19 of file Nil.hpp.

The documentation for this class was generated from the following file:

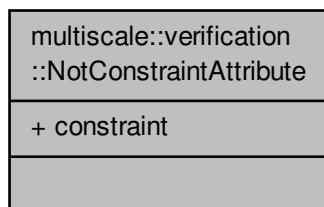
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Nil.hpp

## 7.110 multiscale::verification::NotConstraintAttribute Class Reference

Class for representing a "not" constraint attribute.

```
#include <NotConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NotConstraintAttribute:



### Public Attributes

- [ConstraintAttributeType](#) constraint

### 7.110.1 Detailed Description

Class for representing a "not" constraint attribute.

Definition at line 14 of file NotConstraintAttribute.hpp.

### 7.110.2 Member Data Documentation

#### 7.110.2.1 ConstraintAttributeType multiscale::verification::NotConstraintAttribute::constraint

The constraint which will be negated

Definition at line 18 of file NotConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

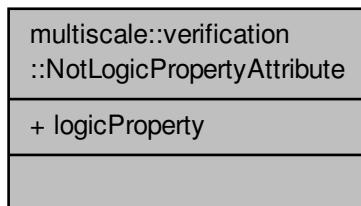
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotConstraintAttribute.hpp

## 7.111 multiscale::verification::NotLogicPropertyAttribute Class Reference

Class for representing a "not" logic property attribute.

```
#include <NotLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NotLogicPropertyAttribute:



### Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.111.1 Detailed Description

Class for representing a "not" logic property attribute.

Definition at line 14 of file NotLogicPropertyAttribute.hpp.

### 7.111.2 Member Data Documentation

### 7.111.2.1 LogicPropertyAttributeType multiscale::verification::NotLogicPropertyAttribute::logicProperty

The logic property following the "not" operator

Definition at line 18 of file [NotLogicPropertyAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

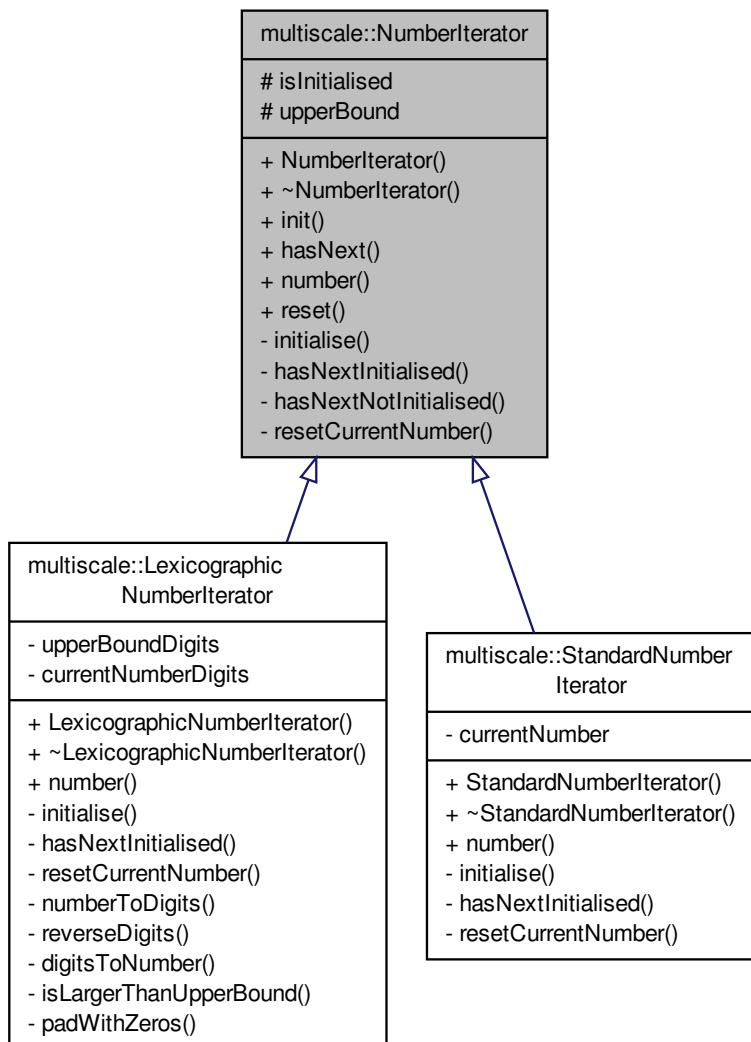
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotLogicPropertyAttribute.hpp](#)

## 7.112 multiscale::NumberIterator Class Reference

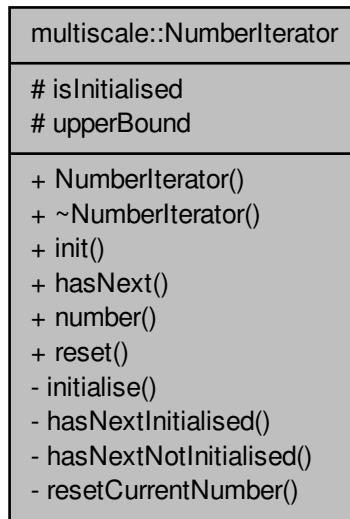
Abstract class representing a number iterator.

```
#include <NumberIterator.hpp>
```

Inheritance diagram for multiscale::NumberIterator:



Collaboration diagram for multiscale::NumberIterator:



## Public Member Functions

- `NumberIterator (unsigned int upperBound)`
- virtual `~NumberIterator ()`
- void `init (unsigned int upperBound)`

*Initialise the iterator considering the given upper bound.*
- bool `hasNext ()`

*Check if there is a next number.*
- virtual unsigned int `number ()=0`

*Get the number pointed by the iterator.*
- void `reset ()`

*Reset the iterator.*

## Protected Attributes

- bool `isInitialised`
- unsigned int `upperBound`

## Private Member Functions

- virtual void `initialise ()=0`

*Initialisation of the members of the class.*
- virtual bool `hasNextInitialised ()=0`

*Check if there is a next number when in initialised state.*
- bool `hasNextNotInitialised ()`

*Check if there is a next number when in not initialised state.*
- virtual void `resetCurrentNumber ()=0`

*Reset the current number to its initial value.*

### 7.112.1 Detailed Description

Abstract class representing a number iterator.

Definition at line 7 of file NumberIterator.hpp.

### 7.112.2 Constructor & Destructor Documentation

#### 7.112.2.1 NumberIterator::NumberIterator ( *unsigned int upperBound* )

Definition at line 6 of file NumberIterator.cpp.

References init().

#### 7.112.2.2 virtual multiscale::NumberIterator::~NumberIterator ( ) [inline], [virtual]

Definition at line 17 of file NumberIterator.hpp.

### 7.112.3 Member Function Documentation

#### 7.112.3.1 bool NumberIterator::hasNext ( )

Check if there is a next number.

Definition at line 14 of file NumberIterator.cpp.

References hasNextInitialised(), hasNextNotInitialised(), and isInitialised.

Referenced by main().

#### 7.112.3.2 virtual bool multiscale::NumberIterator::hasNextInitialised ( ) [private], [pure virtual]

Check if there is a next number when in initialised state.

Implemented in [multiscale::LexicographicNumberIterator](#), and [multiscale::StandardNumberIterator](#).

Referenced by hasNext().

#### 7.112.3.3 bool NumberIterator::hasNextNotInitialised ( ) [private]

Check if there is a next number when in not initialised state.

Definition at line 28 of file NumberIterator.cpp.

References isInitialised.

Referenced by hasNext().

#### 7.112.3.4 void NumberIterator::init ( *unsigned int upperBound* )

Initialise the iterator considering the given upper bound.

##### Parameters

|                   |                 |
|-------------------|-----------------|
| <i>upperBound</i> | The upper bound |
|-------------------|-----------------|

Definition at line 10 of file NumberIterator.cpp.

References upperBound.

Referenced by NumberIterator().

7.112.3.5 `virtual void multiscale::NumberIterator::initialise( ) [private], [pure virtual]`

Initialisation of the members of the class.

Implemented in [multiscale::LexicographicNumberIterator](#), and [multiscale::StandardNumberIterator](#).

7.112.3.6 `virtual unsigned int multiscale::NumberIterator::number( ) [pure virtual]`

Get the number pointed by the iterator.

Implemented in [multiscale::LexicographicNumberIterator](#), and [multiscale::StandardNumberIterator](#).

7.112.3.7 `void NumberIterator::reset( )`

Reset the iterator.

Reset the iterator such that it is not initialised and the value of the current number is reset to its initial value

Definition at line 22 of file [NumberIterator.cpp](#).

References [isInitialised](#), and [resetCurrentNumber\(\)](#).

Referenced by [multiscale::LexicographicNumberIterator::LexicographicNumberIterator\(\)](#), and [multiscale::StandardNumberIterator::StandardNumberIterator\(\)](#).

7.112.3.8 `virtual void multiscale::NumberIterator::resetCurrentNumber( ) [private], [pure virtual]`

Reset the current number to its initial value.

Implemented in [multiscale::LexicographicNumberIterator](#), and [multiscale::StandardNumberIterator](#).

Referenced by [reset\(\)](#).

## 7.112.4 Member Data Documentation

7.112.4.1 `bool multiscale::NumberIterator::isInitialised [protected]`

Flag for checking if the iterator was initialised

Definition at line 11 of file [NumberIterator.hpp](#).

Referenced by [hasNext\(\)](#), [hasNextNotInitialised\(\)](#), and [reset\(\)](#).

7.112.4.2 `unsigned int multiscale::NumberIterator::upperBound [protected]`

Upper bound of the iterator

Definition at line 12 of file [NumberIterator.hpp](#).

Referenced by [multiscale::StandardNumberIterator::hasNextInitialised\(\)](#), [init\(\)](#), [multiscale::LexicographicNumberIterator::initialise\(\)](#), and [multiscale::LexicographicNumberIterator::padWithZeros\(\)](#).

The documentation for this class was generated from the following files:

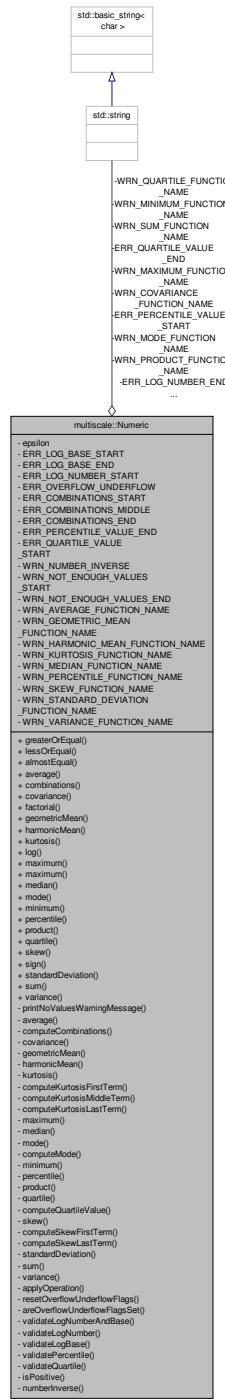
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[NumberIterator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[NumberIterator.cpp](#)

## 7.113 multiscale::Numeric Class Reference

Class for processing numeric (shorts, ints, floats, doubles etc.) expressions.

```
#include <Numeric.hpp>
```

Collaboration diagram for multiscale::Numeric:



## Static Public Member Functions

- static bool `greaterOrEqual` (double number1, double number2)

*Check if the first number is greater than or equal to the second number.*

- static bool `lessOrEqual` (double number1, double number2)

*Check if the first number is less than or equal to the second number.*

- static bool **almostEqual** (double number1, double number2)  
*Check if the two numbers are equal (almost)*
- static double **average** (const std::vector< double > &numbers)  
*Return the average (arithmetic mean) of the provided numbers.*
- static double **combinations** (unsigned int n, unsigned int k)  
*Return combinations of n taken as groups of k.*
- static double **covariance** (const std::vector< double > &values1, const std::vector< double > &values2)  
*Return the covariance for the provided collections of values.*
- static unsigned long **factorial** (unsigned int number)  
*Return the factorial of a number.*
- static double **geometricMean** (const std::vector< double > &numbers)  
*Return the geometric mean of the provided numbers.*
- static double **harmonicMean** (const std::vector< double > &numbers)  
*Return the harmonic mean of the provided numbers.*
- static double **kurtosis** (const std::vector< double > &numbers)  
*Return the kurtosis of the provided numbers.*
- static double **log** (double number, double base)  
*Return the logarithm of a number considering the given base.*
- static double **maximum** (double number1, double number2, double number3)  
*Return the maximum of the provided numbers.*
- static double **maximum** (const std::vector< double > &numbers)  
*Return the maximum of the provided numbers.*
- static double **median** (const std::vector< double > &numbers)  
*Return the median of the provided numbers.*
- static double **mode** (const std::vector< double > &numbers)  
*Return the mode of the provided numbers.*
- static double **minimum** (const std::vector< double > &numbers)  
*Return the minimum of the provided numbers.*
- static double **percentile** (const std::vector< double > &numbers, double percentile)  
*Return the p-th percentile of the provided set of values.*
- static double **product** (const std::vector< double > &numbers)  
*Return the product of the provided numbers.*
- static double **quartile** (const std::vector< double > &numbers, double quartile)  
*Return the q-th quartile of the provided set of values.*
- static double **skew** (const std::vector< double > &numbers)  
*Return the skew of the provided numbers.*
- static int **sign** (double number)  
*Return the sign of the number.*
- static double **standardDeviation** (const std::vector< double > &numbers)  
*Return the standard deviation of the provided set of values.*
- static double **sum** (const std::vector< double > &numbers)  
*Return the sum of the provided numbers.*
- static double **variance** (const std::vector< double > &numbers)  
*Return the variance of the provided set of values.*

## Static Private Member Functions

- static void [printNoValuesWarningMessage](#) (const std::string &functionName)
 

*Print the no values warning message for the given function name.*
- static double [average](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the average (arithmetic mean) of the provided numbers.*
- static double [computeCombinations](#) (unsigned int n, unsigned int k)
 

*Return combinations of n taken as groups of k.*
- static double [covariance](#) (const std::vector< double > &values1, const std::vector< double > &values2, unsigned int nrOfValues)
 

*Return the covariance for the provided collections of values.*
- static double [geometricMean](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the geometric mean of the provided numbers.*
- static double [harmonicMean](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the harmonic mean of the provided numbers.*
- static double [kurtosis](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the kurtosis of the provided numbers.*
- static double [computeKurtosisFirstTerm](#) (unsigned int nrOfValues)
 

*Compute the kurtosis first term considering the given number of values.*
- static double [computeKurtosisMiddleTerm](#) (const std::vector< double > &values, unsigned int nrOfValues)
 

*Compute the kurtosis middle term considering the given values.*
- static double [computeKurtosisLastTerm](#) (unsigned int nrOfValues)
 

*Compute the kurtosis last term considering the given number of values.*
- static double [maximum](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the maximum of the provided numbers.*
- static double [median](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the median of the provided numbers.*
- static double [mode](#) (const std::vector< double > &values, unsigned int nrOfValues)
 

*Compute the mode for the provided values.*
- static double [computeMode](#) (const std::vector< double > &values, unsigned int nrOfValues)
 

*Compute the mode for the provided values.*
- static double [minimum](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the minimum of the provided numbers.*
- static double [percentile](#) (const std::vector< double > &numbers, double percentile, unsigned int nrOfValues)
 

*Return the p-th percentile of the provided set of values.*
- static double [product](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the product of the provided numbers.*
- static double [quartile](#) (const std::vector< double > &numbers, double quartile, unsigned int nrOfValues)
 

*Return the q-th quartile of the provided set of values.*
- static double [computeQuartileValue](#) (double quartile, const std::vector< double > &values, unsigned int nrOfValues)
 

*Compute the quartile for the given collection of values.*
- static double [skew](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the skew of the provided numbers.*
- static double [computeSkewFirstTerm](#) (unsigned int nrOfValues)
 

*Return the skew first term considering the given values.*
- static double [computeSkewLastTerm](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the skew last term considering the given values.*
- static double [standardDeviation](#) (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the standard deviation of the provided set of values.*
- static double [sum](#) (const std::vector< double > &numbers, unsigned int nrOfValues)

- static double **variance** (const std::vector< double > &numbers, unsigned int nrOfValues)
 

*Return the sum of the provided numbers.*
- template<typename Operation , typename Operand >
 static Operand **applyOperation** (Operation operation, Operand operand1, Operand operand2)
 

*Apply the operation on the given operands and throw an exception in case of overflow.*
- static void **resetOverflowUnderflowFlags** ()
 

*Reset the overflow and underflow flags.*
- static bool **areOverflowUnderflowFlagsSet** ()
 

*Reset the overflow and underflow flags.*
- static void **validateLogNumberAndBase** (double number, double base)
 

*Check if the number and the base are positive real numbers, and if the base is different from 1.*
- static void **validateLogNumber** (double number)
 

*Check if the number is a positive real number.*
- static void **validateLogBase** (double base)
 

*Check if the base is a positive real number different from 1.*
- static void **validatePercentile** (double percentile)
 

*Check if the value of the percentile is between 0 and 100.*
- static void **validateQuartile** (double quartile)
 

*Check if the value of the quartile is either 25, 50 or 75.*
- template<typename T >
 static bool **isPositive** (T number)
 

*Check if the given number is positive.*
- template<typename T >
 static T **numberInverse** (T number)
 

*Return the inverse of a number.*

## Static Private Attributes

- static double **epsilon** = 1E-5
- static const std::string **ERR\_LOG\_BASE\_START** = "The base provided to the **log** function ("
- static const std::string **ERR\_LOG\_BASE\_END** = ") should be a positive real number different from 1. Please change."
- static const std::string **ERR\_LOG\_NUMBER\_START** = "The number provided to the **log** function ("
- static const std::string **ERR\_LOG\_NUMBER\_END** = ") should be a positive real number. Please change."
- static const std::string **ERR\_OVERFLOW\_UNDERFLOW** = "An underflow/overflow exception occurred."
- static const std::string **ERR\_COMBINATIONS\_START** = "The provided number of elements n ("
- static const std::string **ERR\_COMBINATIONS\_MIDDLE** = ") should be greater or equal to the number of elements in each group k ("
- static const std::string **ERR\_COMBINATIONS\_END** = ") when computing combinations."
- static const std::string **ERR\_PERCENTILE\_VALUE\_START** = "The provided **percentile** value ("
- static const std::string **ERR\_PERCENTILE\_VALUE\_END** = ") should be between 0 and 100. Please change."
- static const std::string **ERR\_QUARTILE\_VALUE\_START** = "The provided **quartile** value ("
- static const std::string **ERR\_QUARTILE\_VALUE\_END** = ") should be 25, 50 or 75. Please change."
- static const std::string **WRN\_NUMBER\_INVERSE** = "You provided the invalid value \"0\" to the Numeric::inverse(...) function. The default value \"0\" was returned."
- static const std::string **WRN\_NOT\_ENOUGH\_VALUES\_START** = "You provided less than the **minimum** required number of values to the Numeric::"
- static const std::string **WRN\_NOT\_ENOUGH\_VALUES\_END** = "(...) function. The default value \"0\" was returned."
- static const std::string **WRN\_AVERAGE\_FUNCTION\_NAME** = "average"
- static const std::string **WRN\_COVARIANCE\_FUNCTION\_NAME** = "covariance"
- static const std::string **WRN\_GEOMETRIC\_MEAN\_FUNCTION\_NAME** = "geometricMean"

- static const std::string `WRN_HARMONIC_MEAN_FUNCTION_NAME` = "harmonicMean"
- static const std::string `WRN_KURTOSIS_FUNCTION_NAME` = "kurtosis"
- static const std::string `WRN_MAXIMUM_FUNCTION_NAME` = "maximum"
- static const std::string `WRN_MEDIAN_FUNCTION_NAME` = "median"
- static const std::string `WRN_MODE_FUNCTION_NAME` = "mode"
- static const std::string `WRN_MINIMUM_FUNCTION_NAME` = "minimum"
- static const std::string `WRN_PERCENTILE_FUNCTION_NAME` = "percentile"
- static const std::string `WRN_PRODUCT_FUNCTION_NAME` = "product"
- static const std::string `WRN_QUARTILE_FUNCTION_NAME` = "quartile"
- static const std::string `WRN_SKEW_FUNCTION_NAME` = "skew"
- static const std::string `WRN_STANDARD_DEVIATION_FUNCTION_NAME` = "standardDeviation"
- static const std::string `WRN_SUM_FUNCTION_NAME` = "sum"
- static const std::string `WRN_VARIANCE_FUNCTION_NAME` = "variance"

### 7.113.1 Detailed Description

Class for processing numeric (shorts, ints, floats, doubles etc.) expressions.

Definition at line 85 of file Numeric.hpp.

### 7.113.2 Member Function Documentation

#### 7.113.2.1 bool Numeric::almostEqual ( double *number1*, double *number2* ) [static]

Check if the two numbers are equal (almost)

The expression for determining if two real numbers are equal is: if ( $\text{Abs}(x - y) \leq \text{EPSILON} * \text{Max}(1.0f, \text{Abs}(x), \text{Abs}(y))$ ).

Parameters

|                |               |
|----------------|---------------|
| <i>number1</i> | First number  |
| <i>number2</i> | Second number |

Definition at line 24 of file Numeric.cpp.

References epsilon, and maximum().

Referenced by multiscale::Geometry2D::areCollinear(), multiscale::Geometry2D::areEqualPoints(), multiscale::Geometry2D::areIdenticalLines(), multiscale::verification::BayesianModelChecker::computeBayesFactorValue(), multiscale::verification::StatisticalModelChecker::computeFPrimeValueFirstTerm(), multiscale::verification::StatisticalModelChecker::computeFPrimeValueSecondTerm(), multiscale::verification::StatisticalModelChecker::computeFValueFirstTerm(), multiscale::verification::StatisticalModelChecker::computeFValueSecondTerm(), multiscale::verification::ComparatorEvaluator::evaluate(), greaterOrEqual(), multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::hasValidSuccessors(), multiscale::MinEnclosingTriangleFinder::intersects(), multiscale::MinEnclosingTriangleFinder::isGammaAngleEqualTo(), multiscale::Geometry2D::isPointOnLineSegment(), lessOrEqual(), multiscale::Geometry2D::lineIntersection(), numberInverse(), TEST(), validateLogBase(), and validateQuartile().

#### 7.113.2.2 template<typename Operation , typename Operand > static Operand multiscale::Numeric::applyOperation ( Operation *operation*, Operand *operand1*, Operand *operand2* ) [inline], [static], [private]

Apply the operation on the given operands and throw an exception in case of overflow.

**Parameters**

|                  |                    |
|------------------|--------------------|
| <i>operation</i> | The operation      |
| <i>operand1</i>  | The first operand  |
| <i>operand2</i>  | The second operand |

Definition at line 460 of file Numeric.hpp.

References areOverflowUnderflowFlagsSet(), ERR\_OVERFLOW\_UNDERFLOW, MS\_throw, and resetOverflowUnderflowFlags().

Referenced by average(), computeKurtosisMiddleTerm(), computeSkewLastTerm(), covariance(), factorial(), geometricMean(), harmonicMean(), product(), standardDeviation(), sum(), and variance().

#### 7.113.2.3 bool Numeric::areOverflowUnderflowFlagsSet( ) [static], [private]

Reset the overflow and underflow flags.

Definition at line 549 of file Numeric.cpp.

Referenced by applyOperation().

#### 7.113.2.4 double Numeric::average( const std::vector< double > & numbers ) [static]

Return the average (arithmetic mean) of the provided numbers.

$$\text{average} = \frac{1}{n} \sum_{i=1}^n x_i$$

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 28 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_AVERAGE\_FUNCTION\_NAME.

Referenced by computeKurtosisMiddleTerm(), computeSkewLastTerm(), covariance(), multiscale::verification::NumericEvaluator::evaluate(), standardDeviation(), TEST(), and variance().

#### 7.113.2.5 double Numeric::average( const std::vector< double > & numbers, unsigned int nrOfValues ) [static], [private]

Return the average (arithmetic mean) of the provided numbers.

$$\text{average} = \frac{1}{n} \sum_{i=1}^n x_i$$

**Parameters**

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>numbers</i>    | The collection of numbers                         |
| <i>nrOfValues</i> | The number of values in the collection of numbers |

Definition at line 261 of file Numeric.cpp.

References applyOperation(), and sum().

#### 7.113.2.6 double Numeric::combinations( unsigned int n, unsigned int k ) [static]

Return combinations of n taken as groups of k.

**Parameters**

|          |                                            |
|----------|--------------------------------------------|
| <i>n</i> | The total number of elements               |
| <i>k</i> | The number of elements in each combination |

Definition at line 40 of file Numeric.cpp.

References computeCombinations(), ERR\_COMBINATIONS\_END, ERR\_COMBINATIONS\_MIDDLE, ERR\_CO←MBINATIONS\_START, MS\_throw, and multiscale::StringManipulator::toString().

Referenced by TEST().

**7.113.2.7 double Numeric::computeCombinations ( unsigned int *n*, unsigned int *k* ) [static], [private]**

Return combinations of *n* taken as groups of *k*.

**Parameters**

|          |                                            |
|----------|--------------------------------------------|
| <i>n</i> | The total number of elements               |
| <i>k</i> | The number of elements in each combination |

Definition at line 272 of file Numeric.cpp.

Referenced by combinations().

**7.113.2.8 double Numeric::computeKurtosisFirstTerm ( unsigned int *nrOfValues* ) [static], [private]**

Compute the kurtosis first term considering the given number of values.

**Parameters**

|                   |                      |
|-------------------|----------------------|
| <i>nrOfValues</i> | The number of values |
|-------------------|----------------------|

Definition at line 332 of file Numeric.cpp.

Referenced by kurtosis().

**7.113.2.9 double Numeric::computeKurtosisLastTerm ( unsigned int *nrOfValues* ) [static], [private]**

Compute the kurtosis last term considering the given number of values.

**Parameters**

|                   |                      |
|-------------------|----------------------|
| <i>nrOfValues</i> | The number of values |
|-------------------|----------------------|

Definition at line 355 of file Numeric.cpp.

Referenced by kurtosis().

**7.113.2.10 double Numeric::computeKurtosisMiddleTerm ( const std::vector< double > & *values*, unsigned int *nrOfValues* ) [static], [private]**

Compute the kurtosis middle term considering the given values.

**Parameters**

|                   |                      |
|-------------------|----------------------|
| <i>values</i>     | The values           |
| <i>nrOfValues</i> | The number of values |

Definition at line 340 of file Numeric.cpp.

References applyOperation(), average(), and standardDeviation().

Referenced by kurtosis().

7.113.2.11 double Numeric::computeMode ( const std::vector< double > & values, unsigned int nrOfValues ) [static], [private]

Compute the mode for the provided values.

Parameters

|                   |                      |
|-------------------|----------------------|
| <i>values</i>     | The values           |
| <i>nrOfValues</i> | The number of values |

Definition at line 408 of file Numeric.cpp.

Referenced by mode().

7.113.2.12 double Numeric::computeQuartileValue ( double quartile, const std::vector< double > & values, unsigned int nrOfValues ) [static], [private]

Compute the quartile for the given collection of values.

Parameters

|                   |                                        |
|-------------------|----------------------------------------|
| <i>quartile</i>   | The quartile                           |
| <i>values</i>     | The collection of values               |
| <i>nrOfValues</i> | The number of values in the collection |

Definition at line 466 of file Numeric.cpp.

References almostEqual().

Referenced by quartile().

7.113.2.13 double Numeric::computeSkewFirstTerm ( unsigned int nrOfValues ) [static], [private]

Return the skew first term considering the given values.

Parameters

|                   |                      |
|-------------------|----------------------|
| <i>nrOfValues</i> | The number of values |
|-------------------|----------------------|

Definition at line 487 of file Numeric.cpp.

Referenced by skew().

7.113.2.14 double Numeric::computeSkewLastTerm ( const std::vector< double > & numbers, unsigned int nrOfValues ) [static], [private]

Return the skew last term considering the given values.

Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>nrOfValues</i> | The number of values     |

Definition at line 494 of file Numeric.cpp.

References applyOperation(), average(), and standardDeviation().

Referenced by skew().

7.113.2.15 double Numeric::covariance ( const std::vector< double > & values1, const std::vector< double > & values2 ) [static]

Return the covariance for the provided collections of values.

**Parameters**

|                |                                 |
|----------------|---------------------------------|
| <i>values1</i> | The first collection of values  |
| <i>values2</i> | The second collection of values |

Definition at line 53 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_COVARIANCE\_FUNCTION\_NAME.

Referenced by covariance(), multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.16 double Numeric::covariance ( const std::vector< double > & *values1*, const std::vector< double > & *values2*, unsigned int *nrOfValues* ) [static], [private]**

Return the covariance for the provided collections of values.

**Parameters**

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>values1</i>    | The first collection of values                    |
| <i>values2</i>    | The second collection of values                   |
| <i>nrOfValues</i> | The number of values in the collection of numbers |

Definition at line 283 of file Numeric.cpp.

References applyOperation(), average(), and covariance().

**7.113.2.17 unsigned long Numeric::factorial ( unsigned int *number* ) [static]**

Return the factorial of a number.

**Parameters**

|               |                                                   |
|---------------|---------------------------------------------------|
| <i>number</i> | The number for which factorial should be computed |
|---------------|---------------------------------------------------|

Definition at line 65 of file Numeric.cpp.

References applyOperation().

Referenced by TEST().

**7.113.2.18 double Numeric::geometricMean ( const std::vector< double > & *numbers* ) [static]**

Return the geometric mean of the provided numbers.

$$\text{geometricMean} = e^{\frac{1}{n} \sum_{i=1}^n \log(x_i)}$$

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 75 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_GEOMETRIC\_MEAN\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.19 double Numeric::geometricMean ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the geometric mean of the provided numbers.

$$\text{geometricMean} = e^{\frac{1}{n} \sum_{i=1}^n \log(x_i)}$$

**Parameters**

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>numbers</i>    | The collection of numbers                         |
| <i>nrOfValues</i> | The number of values in the collection of numbers |

Definition at line 298 of file Numeric.cpp.

References applyOperation(), and validateLogNumber().

### 7.113.2.20 bool Numeric::greaterOrEqual ( double *number1*, double *number2* ) [static]

Check if the first number is greater than or equal to the second number.

**Parameters**

|                |                   |
|----------------|-------------------|
| <i>number1</i> | The first number  |
| <i>number2</i> | The second number |

Definition at line 16 of file Numeric.cpp.

References almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::advanceBToRightChain(), multiscale::analysis::Entity::isValid(), multiscale::analysis::Region::isValidInputValues(), multiscale::analysis::Cluster::isValidOriginDependentValues(), multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator(), multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), multiscale::MinEnclosingTriangleFinder::searchForBTangency(), and TEST().

### 7.113.2.21 double Numeric::harmonicMean ( const std::vector< double > & *numbers* ) [static]

Return the harmonic mean of the provided numbers.

$$\text{harmonicMean} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 87 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_HARMONIC\_MEAN\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

### 7.113.2.22 double Numeric::harmonicMean ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]

Return the harmonic mean of the provided numbers.

$$\text{harmonicMean} = \frac{n}{\sum_{i=1}^n \frac{1}{x_i}}$$

**Parameters**

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>numbers</i>    | The collection of numbers                         |
| <i>nrOfValues</i> | The number of values in the collection of numbers |

Definition at line 311 of file Numeric.cpp.

References applyOperation(), and numberInverse().

7.113.2.23 `template<typename T> static bool multiscale::Numeric::isPositive ( T number ) [inline], [static], [private]`

Check if the given number is positive.

**Parameters**

|               |                  |
|---------------|------------------|
| <i>number</i> | The given number |
|---------------|------------------|

Definition at line 514 of file Numeric.hpp.

Referenced by validateLogBase(), and validateLogNumber().

**7.113.2.24 double Numeric::kurtosis ( const std::vector< double > & *numbers* ) [static]**

Return the kurtosis of the provided numbers.

$$\text{kurtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \left( \sum_{i=1}^n \left( \frac{x_i - \text{mean}}{\text{stdev}} \right)^4 \right) - \frac{3(n-1)^2}{(n-2)(n-3)}$$

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 99 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_KURTOSIS\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.25 double Numeric::kurtosis ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the kurtosis of the provided numbers.

$$\text{kurtosis} = \frac{n(n+1)}{(n-1)(n-2)(n-3)} \left( \sum_{i=1}^n \left( \frac{x_i - \text{mean}}{\text{stdev}} \right)^4 \right) - \frac{3(n-1)^2}{(n-2)(n-3)}$$

**Parameters**

|                   |                                                   |
|-------------------|---------------------------------------------------|
| <i>numbers</i>    | The collection of numbers                         |
| <i>nrOfValues</i> | The number of values in the collection of numbers |

Definition at line 324 of file Numeric.cpp.

References computeKurtosisFirstTerm(), computeKurtosisLastTerm(), and computeKurtosisMiddleTerm().

**7.113.2.26 bool Numeric::lessOrEqual ( double *number1*, double *number2* ) [static]**

Check if the first number is less than or equal to the second number.

**Parameters**

|                |                   |
|----------------|-------------------|
| <i>number1</i> | The first number  |
| <i>number2</i> | The second number |

Definition at line 20 of file Numeric.cpp.

References almostEqual().

Referenced by multiscale::verification::LogicPropertyVisitor::areSimilarValues(), multiscale::analysis::Region::areValidInputValues(), multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::Geometry2D::isAngleBetweenNonReflex(), multiscale::verification::ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator(), multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), TEST(), multiscale::verification::BayesianModelChecker::validateBayesFactorThreshold(), and multiscale::verification::ApproximateBayesianModelChecker::validateVarianceThreshold().

7.113.2.27 double Numeric::log ( double *number*, double *base* ) [static]

Return the logarithm of a number considering the given base.

The conditions imposed on the number and base are:

- *number*: a positive real number
- *base*: a positive real number different from 1

#### Parameters

|               |                       |
|---------------|-----------------------|
| <i>number</i> | The considered number |
| <i>base</i>   | The considered base   |

Definition at line 111 of file Numeric.cpp.

References validateLogNumberAndBase().

Referenced by multiscale::verification::NumericEvaluator::evaluate().

7.113.2.28 double Numeric::maximum ( double *number1*, double *number2*, double *number3* ) [static]

Return the maximum of the provided numbers.

#### Parameters

|                |                   |
|----------------|-------------------|
| <i>number1</i> | The first number  |
| <i>number2</i> | The second number |
| <i>number3</i> | The third number  |

Definition at line 117 of file Numeric.cpp.

Referenced by almostEqual(), multiscale::verification::NumericEvaluator::evaluate(), maximum(), and TEST().

7.113.2.29 double Numeric::maximum ( const std::vector< double > & *numbers* ) [static]

Return the maximum of the provided numbers.

#### Parameters

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 121 of file Numeric.cpp.

References maximum(), printNoValuesWarningMessage(), and WRN\_MAXIMUM\_FUNCTION\_NAME.

7.113.2.30 double Numeric::maximum ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]

Return the maximum of the provided numbers.

#### Parameters

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 363 of file Numeric.cpp.

References maximum().

7.113.2.31 double Numeric::median ( const std::vector< double > & *numbers* ) [static]

Return the median of the provided numbers.

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 133 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_MEDIAN\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.32 double Numeric::median ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the median of the provided numbers.

**Parameters**

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 376 of file Numeric.cpp.

**7.113.2.33 double Numeric::minimum ( const std::vector< double > & *numbers* ) [static]**

Return the minimum of the provided numbers.

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 145 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_MINIMUM\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), minimum(), and TEST().

**7.113.2.34 double Numeric::minimum ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the minimum of the provided numbers.

**Parameters**

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 385 of file Numeric.cpp.

References minimum().

**7.113.2.35 double Numeric::mode ( const std::vector< double > & *numbers* ) [static]**

Return the mode of the provided numbers.

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 157 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_MODE\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

7.113.2.36 double Numeric::mode ( const std::vector< double > & *values*, unsigned int *nrOfValues* ) [static],  
[private]

Compute the mode for the provided values.

## Parameters

|                   |                      |
|-------------------|----------------------|
| <i>values</i>     | The values           |
| <i>nrOfValues</i> | The number of values |

Definition at line 398 of file Numeric.cpp.

References computeMode().

**7.113.2.37 template<typename T > static T multiscale::Numeric::numberInverse ( T *number* ) [inline], [static], [private]**

Return the inverse of a number.

If the number is equal to zero then a warning is displayed and the default value "0" is returned

## Parameters

|               |                  |
|---------------|------------------|
| <i>number</i> | The given number |
|---------------|------------------|

Definition at line 525 of file Numeric.hpp.

References almostEqual(), multiscale::ConsolePrinter::printWarningMessage(), and WRN\_NUMBER\_INVERSE.

Referenced by harmonicMean().

**7.113.2.38 double Numeric::percentile ( const std::vector< double > & *numbers*, double *percentile* ) [static]**

Return the p-th percentile of the provided set of values.

## Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>percentile</i> | The p-th percentile      |

Definition at line 169 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_PERCENTILE\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.39 double Numeric::percentile ( const std::vector< double > & *numbers*, double *percentile*, unsigned int *nrOfValues* ) [static], [private]**

Return the p-th percentile of the provided set of values.

## Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>percentile</i> | The p-th percentile      |
| <i>nrOfValues</i> | The number of values     |

Definition at line 433 of file Numeric.cpp.

References validatePercentile().

**7.113.2.40 void Numeric::printNoValuesWarningMessage ( const std::string & *functionName* ) [static], [private]**

Print the no values warning message for the given function name.

**Parameters**

|                     |                            |
|---------------------|----------------------------|
| <i>functionName</i> | The provided function name |
|---------------------|----------------------------|

Definition at line 257 of file Numeric.cpp.

References multiscale::ConsolePrinter::printWarningMessage(), WRN\_NOT\_ENOUGH\_VALUES\_END, and WRN\_NOT\_ENOUGH\_VALUES\_START.

Referenced by average(), covariance(), geometricMean(), harmonicMean(), kurtosis(), maximum(), median(), minimum(), mode(), percentile(), product(), quartile(), skew(), standardDeviation(), sum(), and variance().

#### 7.113.2.41 double Numeric::product ( const std::vector< double > & *numbers* ) [static]

Return the product of the provided numbers.

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 181 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_PRODUCT\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), product(), and TEST().

#### 7.113.2.42 double Numeric::product ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]

Return the product of the provided numbers.

**Parameters**

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 445 of file Numeric.cpp.

References applyOperation(), and product().

#### 7.113.2.43 double Numeric::quartile ( const std::vector< double > & *numbers*, double *quartile* ) [static]

Return the q-th quartile of the provided set of values.

**Parameters**

|                 |                          |
|-----------------|--------------------------|
| <i>numbers</i>  | The collection of values |
| <i>quartile</i> | The q-th quartile        |

Definition at line 193 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_QUARTILE\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

#### 7.113.2.44 double Numeric::quartile ( const std::vector< double > & *numbers*, double *quartile*, unsigned int *nrOfValues* ) [static], [private]

Return the q-th quartile of the provided set of values.

## Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>quartile</i>   | The q-th quartile        |
| <i>nrOfValues</i> | The number of values     |

Definition at line 455 of file Numeric.cpp.

References computeQuartileValue(), and validateQuartile().

## 7.113.2.45 void Numeric::resetOverflowUnderflowFlags( ) [static], [private]

Reset the overflow and underflow flags.

Definition at line 544 of file Numeric.cpp.

Referenced by applyOperation().

7.113.2.46 int Numeric::sign( double *number* ) [static]

Return the sign of the number.

The sign function returns: -1, if number < 0 +1, if number > 0 0, otherwise

## Parameters

|               |                       |
|---------------|-----------------------|
| <i>number</i> | The considered number |
|---------------|-----------------------|

Definition at line 217 of file Numeric.cpp.

Referenced by multiscale::Geometry2D::areOnTheSameSideOfLine(), and multiscale::verification::NumericEvaluator::evaluate().

7.113.2.47 double Numeric::skew( const std::vector< double > & *numbers* ) [static]

Return the skew of the provided numbers.

## Parameters

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 205 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_SKEW\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

7.113.2.48 double Numeric::skew( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]

Return the skew of the provided numbers.

## Parameters

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 480 of file Numeric.cpp.

References computeSkewFirstTerm(), and computeSkewLastTerm().

7.113.2.49 double Numeric::standardDeviation ( const std::vector< double > & *numbers* ) [static]

Return the standard deviation of the provided set of values.

**Parameters**

|                |                          |
|----------------|--------------------------|
| <i>numbers</i> | The collection of values |
|----------------|--------------------------|

Definition at line 221 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_STANDARD\_DEVIATION\_FUNCTION\_NAME.

Referenced by computeKurtosisMiddleTerm(), computeSkewLastTerm(), multiscale::verification::NumericEvaluator::evaluate(), and TEST().

**7.113.2.50 double Numeric::standardDeviation ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the standard deviation of the provided set of values.

**Parameters**

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>nrOfValues</i> | The number of values     |

Definition at line 510 of file Numeric.cpp.

References applyOperation(), and average().

**7.113.2.51 double Numeric::sum ( const std::vector< double > & *numbers* ) [static]**

Return the sum of the provided numbers.

**Parameters**

|                |                           |
|----------------|---------------------------|
| <i>numbers</i> | The collection of numbers |
|----------------|---------------------------|

Definition at line 233 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_SUM\_FUNCTION\_NAME.

Referenced by average(), multiscale::verification::NumericEvaluator::evaluate(), sum(), and TEST().

**7.113.2.52 double Numeric::sum ( const std::vector< double > & *numbers*, unsigned int *nrOfValues* ) [static], [private]**

Return the sum of the provided numbers.

**Parameters**

|                   |                           |
|-------------------|---------------------------|
| <i>numbers</i>    | The collection of numbers |
| <i>nrOfValues</i> | The number of values      |

Definition at line 522 of file Numeric.cpp.

References applyOperation(), and sum().

**7.113.2.53 void Numeric::validateLogBase ( double *base* ) [static], [private]**

Check if the base is a positive real number different from 1.

**Parameters**

|             |                     |
|-------------|---------------------|
| <i>base</i> | The considered base |
|-------------|---------------------|

Definition at line 568 of file Numeric.cpp.

References `almostEqual()`, `ERR_LOG_BASE_END`, `ERR_LOG_BASE_START`, `isPositive()`, `MS_throw_detailed`, and `multiscale::StringManipulator::toString()`.

Referenced by `validateLogNumberAndBase()`.

#### 7.113.2.54 void Numeric::validateLogNumber ( double *number* ) [static], [private]

Check if the number is a positive real number.

##### Parameters

|               |                       |
|---------------|-----------------------|
| <i>number</i> | The considered number |
|---------------|-----------------------|

Definition at line 561 of file Numeric.cpp.

References `ERR_LOG_NUMBER_END`, `ERR_LOG_NUMBER_START`, `isPositive()`, `MS_throw_detailed`, and `multiscale::StringManipulator::toString()`.

Referenced by `geometricMean()`, and `validateLogNumberAndBase()`.

#### 7.113.2.55 void Numeric::validateLogNumberAndBase ( double *number*, double *base* ) [static], [private]

Check if the number and the base are positive real numbers, and if the base is different from 1.

##### Parameters

|               |                       |
|---------------|-----------------------|
| <i>number</i> | The considered number |
| <i>base</i>   | The considered base   |

Definition at line 556 of file Numeric.cpp.

References `validateLogBase()`, and `validateLogNumber()`.

Referenced by `log()`.

#### 7.113.2.56 void Numeric::validatePercentile ( double *percentile* ) [static], [private]

Check if the value of the percentile is between 0 and 100.

##### Parameters

|                   |                      |
|-------------------|----------------------|
| <i>percentile</i> | The percentile value |
|-------------------|----------------------|

Definition at line 575 of file Numeric.cpp.

References `ERR_PERCENTILE_VALUE_END`, `ERR_PERCENTILE_VALUE_START`, and `MS_throw_detailed`.

Referenced by `percentile()`.

#### 7.113.2.57 void Numeric::validateQuartile ( double *quartile* ) [static], [private]

Check if the value of the quartile is either 25, 50 or 75.

##### Parameters

|                 |                    |
|-----------------|--------------------|
| <i>quartile</i> | The quartile value |
|-----------------|--------------------|

Definition at line 582 of file Numeric.cpp.

References `almostEqual()`, `ERR_QUARTILE_VALUE_END`, `ERR_QUARTILE_VALUE_START`, and `MS_throw_detailed`.

Referenced by quartile().

#### 7.113.2.58 double Numeric::variance ( const std::vector< double > & numbers ) [static]

Return the variance of the provided set of values.

##### Parameters

|                |                          |
|----------------|--------------------------|
| <i>numbers</i> | The collection of values |
|----------------|--------------------------|

Definition at line 245 of file Numeric.cpp.

References printNoValuesWarningMessage(), and WRN\_VARIANCE\_FUNCTION\_NAME.

Referenced by multiscale::verification::NumericEvaluator::evaluate(), and TEST().

#### 7.113.2.59 double Numeric::variance ( const std::vector< double > & numbers, unsigned int nrOfValues ) [static], [private]

Return the variance of the provided set of values.

##### Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>numbers</i>    | The collection of values |
| <i>nrOfValues</i> | The number of values     |

Definition at line 532 of file Numeric.cpp.

References applyOperation(), and average().

### 7.113.3 Member Data Documentation

#### 7.113.3.1 double Numeric::epsilon = 1E-5 [static], [private]

Value of epsilon used to compare two real numbers

Definition at line 89 of file Numeric.hpp.

Referenced by almostEqual().

#### 7.113.3.2 const std::string Numeric::ERR\_COMBINATIONS\_END = ") when computing combinations." [static], [private]

Definition at line 546 of file Numeric.hpp.

Referenced by combinations().

#### 7.113.3.3 const std::string Numeric::ERR\_COMBINATIONS\_MIDDLE = ") should be greater or equal to the number of elements in each group k (" [static], [private]

Definition at line 545 of file Numeric.hpp.

Referenced by combinations().

#### 7.113.3.4 const std::string Numeric::ERR\_COMBINATIONS\_START = "The provided number of elements n (" [static], [private]

Definition at line 544 of file Numeric.hpp.

Referenced by combinations().

7.113.3.5 `const std::string Numeric::ERR_LOG_BASE_END = ") should be a positive real number different from 1. Please change."` [static], [private]

Definition at line 538 of file Numeric.hpp.

Referenced by validateLogBase().

7.113.3.6 `const std::string Numeric::ERR_LOG_BASE_START = "The base provided to the log function ("` [static], [private]

Definition at line 537 of file Numeric.hpp.

Referenced by validateLogBase().

7.113.3.7 `const std::string Numeric::ERR_LOG_NUMBER_END = ") should be a positive real number. Please change."` [static], [private]

Definition at line 540 of file Numeric.hpp.

Referenced by validateLogNumber().

7.113.3.8 `const std::string Numeric::ERR_LOG_NUMBER_START = "The number provided to the log function ("` [static], [private]

Definition at line 539 of file Numeric.hpp.

Referenced by validateLogNumber().

7.113.3.9 `const std::string Numeric::ERR_OVERFLOW_UNDERFLOW = "An underflow/overflow exception occurred."` [static], [private]

Definition at line 542 of file Numeric.hpp.

Referenced by applyOperation().

7.113.3.10 `const std::string Numeric::ERR_PERCENTILE_VALUE_END = ") should be between 0 and 100. Please change."` [static], [private]

Definition at line 549 of file Numeric.hpp.

Referenced by validatePercentile().

7.113.3.11 `const std::string Numeric::ERR_PERCENTILE_VALUE_START = "The provided percentile value ("` [static], [private]

Definition at line 548 of file Numeric.hpp.

Referenced by validatePercentile().

7.113.3.12 `const std::string Numeric::ERR_QUARTILE_VALUE_END = ") should be 25, 50 or 75. Please change."` [static], [private]

Definition at line 552 of file Numeric.hpp.

Referenced by validateQuartile().

7.113.3.13 `const std::string Numeric::ERR_QUARTILE_VALUE_START = "The provided quartile value (" [static], [private]`

Definition at line 551 of file Numeric.hpp.

Referenced by validateQuartile().

7.113.3.14 `const std::string Numeric::WRN_AVERAGE_FUNCTION_NAME = "average" [static], [private]`

Definition at line 559 of file Numeric.hpp.

Referenced by average().

7.113.3.15 `const std::string Numeric::WRN_COVARIANCE_FUNCTION_NAME = "covariance" [static], [private]`

Definition at line 560 of file Numeric.hpp.

Referenced by covariance().

7.113.3.16 `const std::string Numeric::WRN_GEOMETRIC_MEAN_FUNCTION_NAME = "geometricMean" [static], [private]`

Definition at line 561 of file Numeric.hpp.

Referenced by geometricMean().

7.113.3.17 `const std::string Numeric::WRN_HARMONIC_MEAN_FUNCTION_NAME = "harmonicMean" [static], [private]`

Definition at line 562 of file Numeric.hpp.

Referenced by harmonicMean().

7.113.3.18 `const std::string Numeric::WRN_KURTOSIS_FUNCTION_NAME = "kurtosis" [static], [private]`

Definition at line 563 of file Numeric.hpp.

Referenced by kurtosis().

7.113.3.19 `const std::string Numeric::WRN_MAXIMUM_FUNCTION_NAME = "maximum" [static], [private]`

Definition at line 564 of file Numeric.hpp.

Referenced by maximum().

7.113.3.20 `const std::string Numeric::WRN_MEDIAN_FUNCTION_NAME = "median" [static], [private]`

Definition at line 565 of file Numeric.hpp.

Referenced by median().

7.113.3.21 `const std::string Numeric::WRN_MINIMUM_FUNCTION_NAME = "minimum" [static], [private]`

Definition at line 567 of file Numeric.hpp.

Referenced by minimum().

7.113.3.22 `const std::string Numeric::WRN_MODE_FUNCTION_NAME = "mode" [static], [private]`

Definition at line 566 of file Numeric.hpp.

Referenced by mode().

7.113.3.23 `const std::string Numeric::WRN_NOT_ENOUGH_VALUES_END = "(...) function. The default value \"0\" was returned." [static], [private]`

Definition at line 557 of file Numeric.hpp.

Referenced by printNoValuesWarningMessage().

7.113.3.24 `const std::string Numeric::WRN_NOT_ENOUGH_VALUES_START = "You provided less than the minimum required number of values to the Numeric:::" [static], [private]`

Definition at line 556 of file Numeric.hpp.

Referenced by printNoValuesWarningMessage().

7.113.3.25 `const std::string Numeric::WRN_NUMBER_INVERSE = "You provided the invalid value \"0\" to the Numeric::inverse(...) function. The default value \"0\" was returned." [static], [private]`

Definition at line 554 of file Numeric.hpp.

Referenced by numberInverse().

7.113.3.26 `const std::string Numeric::WRN_PERCENTILE_FUNCTION_NAME = "percentile" [static], [private]`

Definition at line 568 of file Numeric.hpp.

Referenced by percentile().

7.113.3.27 `const std::string Numeric::WRN_PRODUCT_FUNCTION_NAME = "product" [static], [private]`

Definition at line 569 of file Numeric.hpp.

Referenced by product().

7.113.3.28 `const std::string Numeric::WRN_QUARTILE_FUNCTION_NAME = "quartile" [static], [private]`

Definition at line 570 of file Numeric.hpp.

Referenced by quartile().

7.113.3.29 `const std::string Numeric::WRN_SKEW_FUNCTION_NAME = "skew" [static], [private]`

Definition at line 571 of file Numeric.hpp.

Referenced by skew().

7.113.3.30 `const std::string Numeric::WRN_STANDARD_DEVIATION_FUNCTION_NAME = "standardDeviation" [static], [private]`

Definition at line 572 of file Numeric.hpp.

Referenced by standardDeviation().

7.113.3.31 `const std::string Numeric::WRN_SUM_FUNCTION_NAME = "sum" [static], [private]`

Definition at line 573 of file Numeric.hpp.

Referenced by sum().

7.113.3.32 `const std::string Numeric::WRN_VARIANCE_FUNCTION_NAME = "variance" [static], [private]`

Definition at line 574 of file Numeric.hpp.

Referenced by variance().

The documentation for this class was generated from the following files:

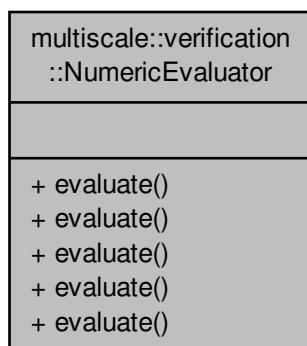
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[Numeric.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[Numeric.cpp](#)

## 7.114 multiscale::verification::NumericEvaluator Class Reference

Class for evaluating numeric expressions.

```
#include <NumericEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::NumericEvaluator:



### Static Public Member Functions

- template<typename T >  
`static double evaluate (const UnaryNumericMeasureType &unaryNumericMeasure, T value)`  
*Evaluate the given unary numeric expression.*
- template<typename T >  
`static double evaluate (const BinaryNumericMeasureType &binaryNumericMeasure, T firstValue, T secondValue)`  
*Evaluate the given binary numeric expression.*
- static double evaluate (const UnaryStatisticalMeasureType &unaryStatisticalMeasure, const std::vector<double > &values)  
*Evaluate the given unary statistical measure expression.*

- static double `evaluate` (const `BinaryStatisticalMeasureType` &binaryStatisticalMeasure, const std::vector<double> &values1, const std::vector<double> &values2)  
*Evaluate the given binary statistical measure expression.*
- static double `evaluate` (const `BinaryStatisticalQuantileMeasureType` &binaryStatisticalQuantileMeasure, const std::vector<double> &values, double parameter)  
*Evaluate the given binary statistical quantile measure expression.*

### 7.114.1 Detailed Description

Class for evaluating numeric expressions.

Definition at line 14 of file NumericEvaluator.hpp.

### 7.114.2 Member Function Documentation

#### 7.114.2.1 template<typename T> static double multiscale::verification::NumericEvaluator::evaluate ( const UnaryNumericMeasureType & unaryNumericMeasure, T value ) [inline], [static]

Evaluate the given unary numeric expression.

Parameters

|                                  |                                                          |
|----------------------------------|----------------------------------------------------------|
| <code>unaryNumericMeasure</code> | The unary numeric measure type                           |
| <code>value</code>               | The value for which the unary numeric measure is applied |

Definition at line 24 of file NumericEvaluator.hpp.

References multiscale::verification::Abs, multiscale::verification::Ceil, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, multiscale::verification::Floor, MS\_throw, multiscale::verification::Round, multiscale::verification::Sign, multiscale::Numeric::sign(), multiscale::verification::Sqrt, and multiscale::verification::Trunc.

Referenced by multiscale::verification::FilterNumericVisitor::operator()(), multiscale::verification::TemporalNumericVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

#### 7.114.2.2 template<typename T> static double multiscale::verification::NumericEvaluator::evaluate ( const BinaryNumericMeasureType & binaryNumericMeasure, T firstValue, T secondValue ) [inline], [static]

Evaluate the given binary numeric expression.

Parameters

|                                   |                                                                  |
|-----------------------------------|------------------------------------------------------------------|
| <code>binaryNumericMeasure</code> | The binary numeric measure type                                  |
| <code>firstValue</code>           | The first value for which the binary numeric measure is applied  |
| <code>secondValue</code>          | The second value for which the binary numeric measure is applied |

Definition at line 62 of file NumericEvaluator.hpp.

References multiscale::verification::Add, multiscale::verification::Div, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, multiscale::verification::Log, multiscale::Numeric::log(), multiscale::verification::Mod, MS\_throw, multiscale::verification::Multiply, multiscale::verification::Power, and multiscale::verification::Subtract.

#### 7.114.2.3 static double multiscale::verification::NumericEvaluator::evaluate ( const UnaryStatisticalMeasureType & unaryStatisticalMeasure, const std::vector<double> & values ) [inline], [static]

Evaluate the given unary statistical measure expression.

## Parameters

|                                                      |                                     |
|------------------------------------------------------|-------------------------------------|
| <i>unary</i><br><i>Statistical</i><br><i>Measure</i> | The unary statistical measure type  |
| <i>values</i>                                        | The considered collection of values |

Definition at line 99 of file NumericEvaluator.hpp.

References multiscale::Numeric::average(), multiscale::verification::Avg, multiscale::verification::Count, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, multiscale::verification::Geomean, multiscale::Numeric::geometricMean(), multiscale::verification::Harmean, multiscale::Numeric::harmonicMean(), multiscale::verification::Kurt, multiscale::Numeric::kurtosis(), multiscale::verification::Max, multiscale::Numeric::maximum(), multiscale::verification::Median, multiscale::Numeric::median(), multiscale::verification::Min, multiscale::Numeric::minimum(), multiscale::verification::Mode, multiscale::Numeric::mode(), MS\_throw, multiscale::verification::Product, multiscale::Numeric::product(), multiscale::verification::Skew, multiscale::Numeric::skew(), multiscale::Numeric::standardDeviation(), multiscale::verification::Stdev, multiscale::verification::Sum, multiscale::Numeric::sum(), multiscale::verification::Var, and multiscale::Numeric::variance().

**7.114.2.4 static double multiscale::verification::NumericEvaluator::evaluate ( const BinaryStatisticalMeasureType & *binaryStatisticalMeasure*, const std::vector< double > & *values1*, const std::vector< double > & *values2* ) [inline], [static]**

Evaluate the given binary statistical measure expression.

## Parameters

|                                                       |                                            |
|-------------------------------------------------------|--------------------------------------------|
| <i>binary</i><br><i>Statistical</i><br><i>Measure</i> | The binary statistical measure type        |
| <i>values1</i>                                        | The first collection of considered values  |
| <i>values2</i>                                        | The second collection of considered values |

Definition at line 172 of file NumericEvaluator.hpp.

References multiscale::verification::Covar, multiscale::Numeric::covariance(), multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, and MS\_throw.

**7.114.2.5 static double multiscale::verification::NumericEvaluator::evaluate ( const BinaryStatisticalQuantileMeasureType & *binaryStatisticalQuantileMeasure*, const std::vector< double > & *values*, double *parameter* ) [inline], [static]**

Evaluate the given binary statistical quantile measure expression.

## Parameters

|                                                                          |                                                  |
|--------------------------------------------------------------------------|--------------------------------------------------|
| <i>binary</i><br><i>Statistical</i><br><i>Quantile</i><br><i>Measure</i> | The binary statistical quantile measure type     |
| <i>values</i>                                                            | The considered values                            |
| <i>parameter</i>                                                         | The parameter used by the ternary subset measure |

Definition at line 192 of file NumericEvaluator.hpp.

References multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, multiscale::verification::Percentile, multiscale::Numeric::percentile(), multiscale::verification::Quartile, and multiscale::Numeric::quartile().

The documentation for this class was generated from the following file:

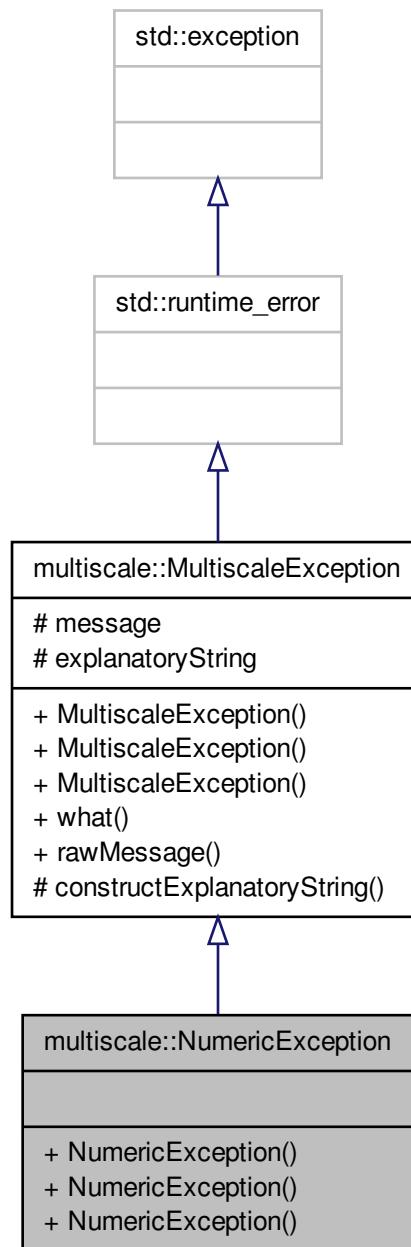
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericEvaluator.hpp](#)

## 7.115 multiscale::NumericException Class Reference

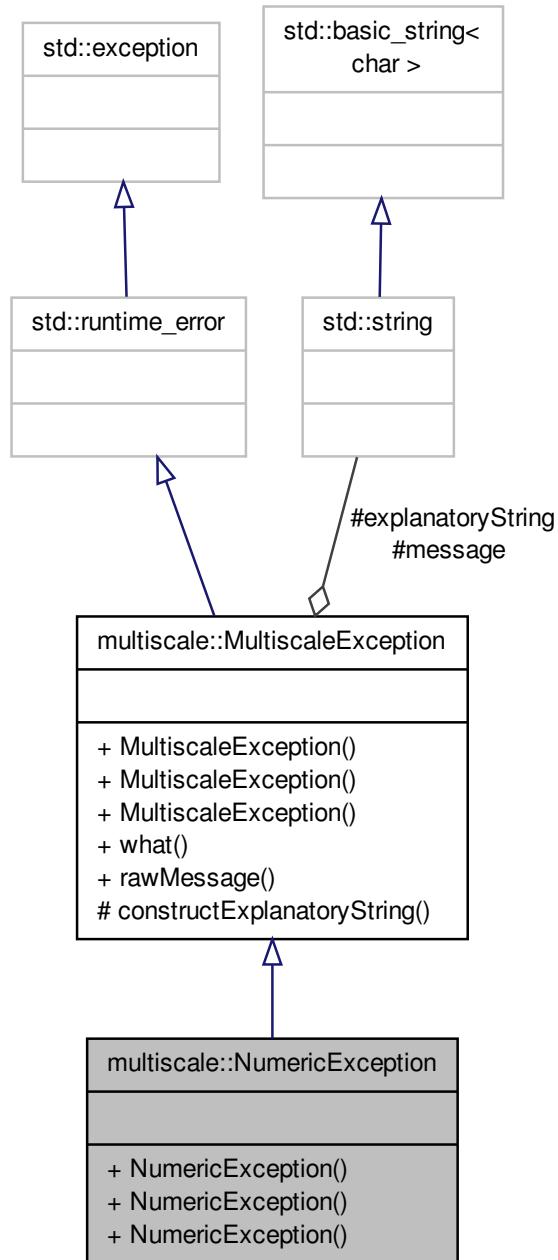
Class for representing algorithm exceptions.

```
#include <NumericException.hpp>
```

Inheritance diagram for multiscale::NumericException:



Collaboration diagram for multiscale::NumericException:



## Public Member Functions

- [NumericException \(\)](#)
- [NumericException \(const std::string &file, int line, const std::string &msg\)](#)
- [NumericException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.115.1 Detailed Description

Class for representing algorithm exceptions.

Definition at line 12 of file NumericException.hpp.

### 7.115.2 Constructor & Destructor Documentation

#### 7.115.2.1 multiscale::NumericException::NumericException( ) [inline]

Definition at line 16 of file NumericException.hpp.

#### 7.115.2.2 multiscale::NumericException::NumericException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 18 of file NumericException.hpp.

#### 7.115.2.3 multiscale::NumericException::NumericException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 23 of file NumericException.hpp.

The documentation for this class was generated from the following file:

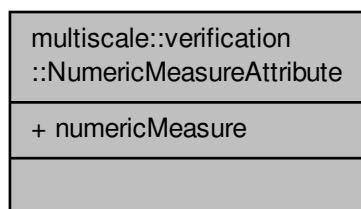
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/NumericException.hpp

## 7.116 multiscale::verification::NumericMeasureAttribute Class Reference

Class for representing a numeric measure attribute.

```
#include <NumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericMeasureAttribute:



### Public Attributes

- [NumericMeasureType numericMeasure](#)

### 7.116.1 Detailed Description

Class for representing a numeric measure attribute.

Definition at line 35 of file NumericMeasureAttribute.hpp.

### 7.116.2 Member Data Documentation

#### 7.116.2.1 NumericMeasureType multiscale::verification::NumericMeasureAttribute::numericMeasure

The numeric measure

Definition at line 39 of file NumericMeasureAttribute.hpp.

Referenced by multiscale::verification::NumericVisitor::operator()().

The documentation for this class was generated from the following file:

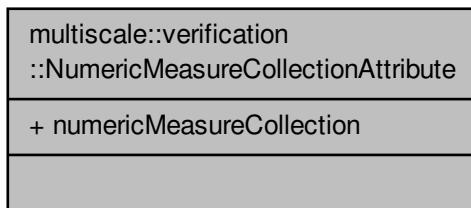
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp

## 7.117 multiscale::verification::NumericMeasureCollectionAttribute Class Reference

Class for representing a numeric measure collection attribute.

```
#include <NumericMeasureCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericMeasureCollectionAttribute:



### Public Attributes

- [NumericMeasureCollectionType numericMeasureCollection](#)

### 7.117.1 Detailed Description

Class for representing a numeric measure collection attribute.

Definition at line 22 of file NumericMeasureCollectionAttribute.hpp.

### 7.117.2 Member Data Documentation

### 7.117.2.1 NumericMeasureCollectionType multiscale::verification::NumericMeasureCollectionAttribute::numericMeasureCollection

The numeric measure collection

Definition at line 26 of file `NumericMeasureCollectionAttribute.hpp`.

Referenced by `multiscale::verification::TemporalNumericVisitor::evaluateNumericMeasureCollection()`.

The documentation for this class was generated from the following file:

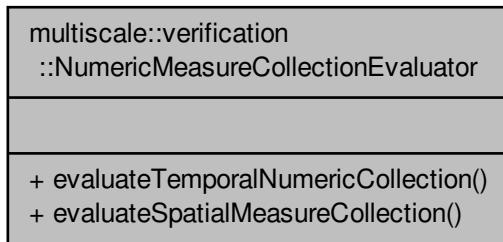
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp`

## 7.118 multiscale::verification::NumericMeasureCollectionEvaluator Class Reference

Class used to evaluate numeric measure collections.

```
#include <NumericMeasureCollectionEvaluator.hpp>
```

Collaboration diagram for `multiscale::verification::NumericMeasureCollectionEvaluator`:



### Static Public Member Functions

- static `std::vector< double > evaluateTemporalNumericCollection (const SpatialTemporalTrace &trace, const TypeSemanticsTable &typeSemanticsTable, const TemporalNumericCollectionAttribute &temporalNumericCollection)`

*Evaluate the given temporal numeric collection.*
- static `std::vector< double > evaluateSpatialMeasureCollection (const TimePoint &timePoint, const TypeSemanticsTable &typeSemanticsTable, const SpatialMeasureCollectionAttribute &spatialMeasureCollection)`

*Evaluate the spatial measure collection considering the given timepoint.*

### 7.118.1 Detailed Description

Class used to evaluate numeric measure collections.

Definition at line 14 of file `NumericMeasureCollectionEvaluator.hpp`.

### 7.118.2 Member Function Documentation

```
7.118.2.1 static std::vector<double> multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatial←
MeasureCollection (const TimePoint & timePoint, const TypeSemanticsTable & typeSemanticsTable, const
SpatialMeasureCollectionAttribute & spatialMeasureCollection) [inline], [static]
```

Evaluate the spatial measure collection considering the given timepoint.

## Parameters

|                                 |                                           |
|---------------------------------|-------------------------------------------|
| <i>timePoint</i>                | The given timepoint                       |
| <i>typeSemanticsTable</i>       | The given type semantics table            |
| <i>spatialMeasureCollection</i> | The considered spatial measure collection |

Definition at line 42 of file NumericMeasureCollectionEvaluator.hpp.

References multiscale::verification::TimePointEvaluator::getSpatialMeasureValues(), multiscale::verification::SpatialMeasureCollectionAttribute::spatialMeasure, multiscale::verification::SpatialMeasureAttribute::spatialMeasureType, multiscale::verification::SpatialMeasureCollectionAttribute::subset, and multiscale::verification::SubsetAttribute::subset.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

**7.118.2.2 static std::vector<double> multiscale::verification::NumericMeasureCollectionEvaluator::evaluateTemporal<br>NumericCollection ( const SpatialTemporalTrace & trace, const TypeSemanticsTable & typeSemanticsTable, const TemporalNumericCollectionAttribute & temporalNumericCollection ) [inline], [static]**

Evaluate the given temporal numeric collection.

## Parameters

|                                  |                                       |
|----------------------------------|---------------------------------------|
| <i>trace</i>                     | The given spatial temporal trace      |
| <i>typeSemanticsTable</i>        | The given type semantics table        |
| <i>temporalNumericCollection</i> | The given temporal numeric collection |

Definition at line 25 of file NumericMeasureCollectionEvaluator.hpp.

References multiscale::verification::TemporalNumericCollectionAttribute::temporalNumericCollection.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()(), and multiscale::verification::LogicPropertyVisitor::operator()().

The documentation for this class was generated from the following file:

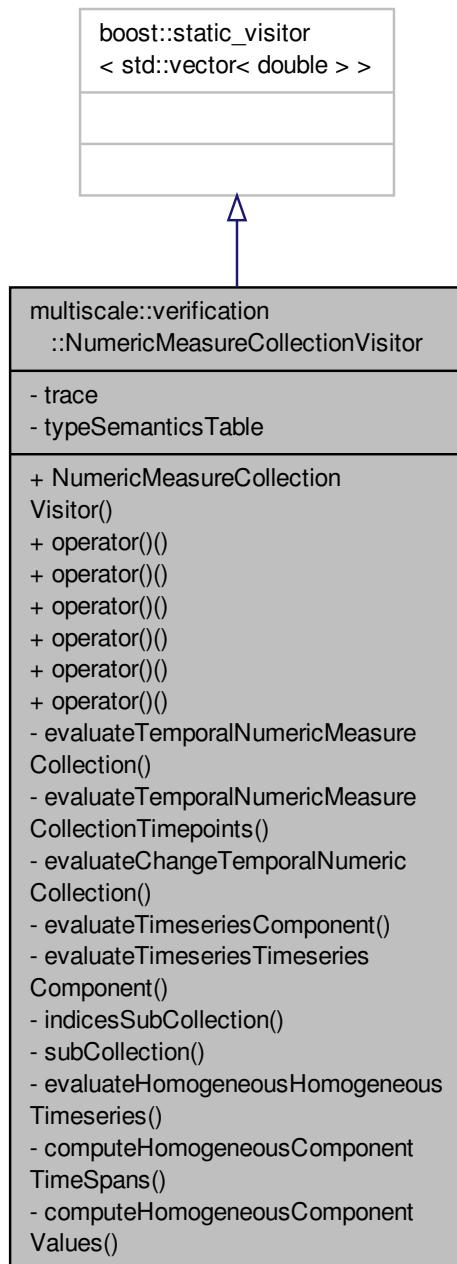
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericMeasureCollectionEvaluator.hpp](#)

## 7.119 multiscale::verification::NumericMeasureCollectionVisitor Class Reference

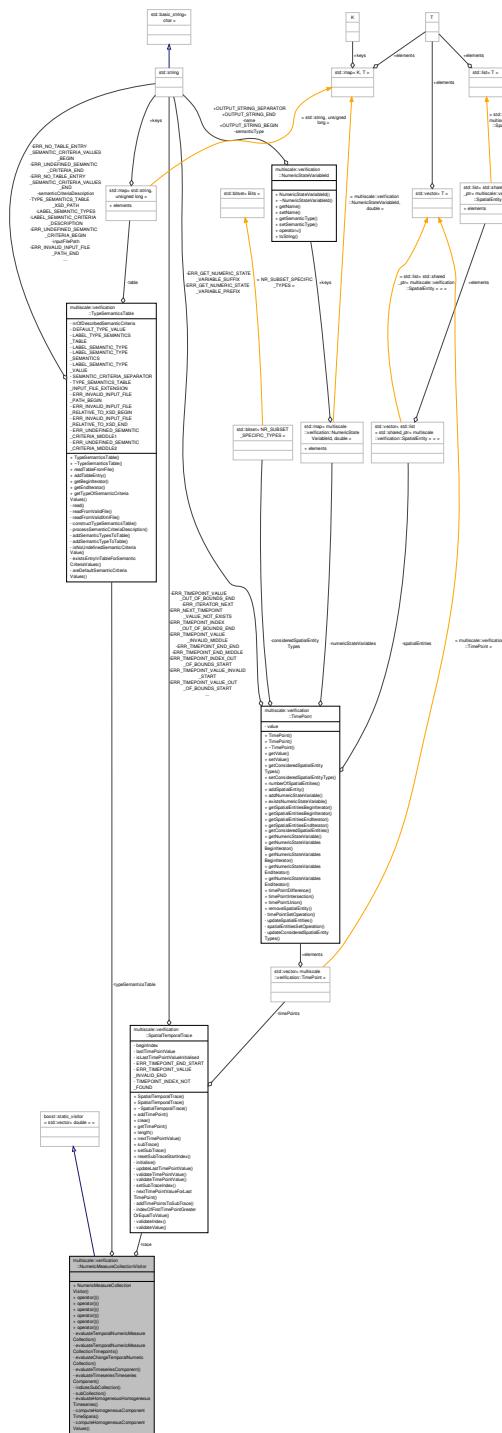
Class for evaluating numeric measure collections.

```
#include <NumericMeasureCollectionVisitor.hpp>
```

Inheritance diagram for multiscale::verification::NumericMeasureCollectionVisitor:



## Collaboration diagram for multiscale::verification::NumericMeasureCollectionVisitor:



## Public Member Functions

- `NumericMeasureCollectionVisitor` (const `SpatialTemporalTrace` &`trace`, const `TypeSemanticsTable` &`typeSemanticsTable`)
  - `std::vector< double > operator()` (const `TemporalNumericCollectionAttribute` &`temporalNumericCollection`)  
const

Overloading the "`()`" operator for the [TemporalNumericCollectionAttribute](#) alternative.

- std::vector< double > **operator()** (const [SpatialMeasureCollectionAttribute](#) &spatialMeasureCollection) const  
*Overloading the "()" operator for the [SpatialMeasureCollectionAttribute](#) alternative.*
- std::vector< double > **operator()** (const [TemporalNumericMeasureCollectionAttribute](#) &temporalNumericMeasureCollection) const  
*Overloading the "()" operator for the [TemporalNumericMeasureCollectionAttribute](#) alternative.*
- std::vector< double > **operator()** (const [ChangeTemporalNumericCollectionAttribute](#) &changeTemporalNumericCollection) const  
*Overloading the "()" operator for the [ChangeTemporalNumericCollectionAttribute](#) alternative.*
- std::vector< double > **operator()** (const [TimeseriesTimeseriesComponentAttribute](#) &timeseriesTimeseriesComponent) const  
*Overloading the "()" operator for the [TimeseriesTimeseriesComponent](#) alternative.*
- std::vector< double > **operator()** (const [HomogeneousHomogeneousTimeseriesAttribute](#) &homogeneousHomogeneousTimeseries) const  
*Overloading the "()" operator for the [HomogeneousHomogeneousTimeseriesAttribute](#) alternative.*

## Private Member Functions

- std::vector< double > **evaluateTemporalNumericMeasureCollection** (const [SpatialTemporalTrace](#) &trace, unsigned long startTimepoint, unsigned long endTimepoint, const [NumericMeasureType](#) &numericMeasure) const  
*Evaluate the temporal numeric measure collection considering the given spatio-temporal trace.*
- std::vector< double > **evaluateTemporalNumericMeasureCollectionTimepoints** (const [SpatialTemporalTrace](#) &trace, unsigned long startTimepoint, unsigned long endTimepoint) const  
*Compute the collection of timepoints considering the given trace, and start and end timepoints.*
- std::vector< double > **evaluateChangeTemporalNumericCollection** (const [ChangeMeasureAttribute](#) &changeMeasure, const std::vector< double > &temporalNumericCollectionValues) const  
*Evaluate the temporal numeric collection values considering the given change measure.*
- std::vector< std::size\_t > **evaluateTimeseriesComponent** (const std::vector< double > &values, const [TimeseriesComponentAttribute](#) &timeseriesComponent) const  
*Evaluate the timeseries component considering the given collection of values.*
- std::vector< double > **evaluateTimeseriesTimeseriesComponent** (const [TimeseriesMeasureType](#) &timeseriesMeasureType, const std::vector< double > &values, const std::vector< double > &timePoints, const std::vector< std::size\_t > &indices) const  
*Evaluate the given timeseries timeseries component.*
- std::vector< std::size\_t > **indicesSubCollection** (const std::vector< std::size\_t > &indices, std::size\_t startPosition, std::size\_t step) const  
*Compute the collection of sub-indices from the given collection of indices.*
- template<typename T>  
std::vector< T > **subCollection** (const std::vector< T > &initialCollection, const std::vector< std::size\_t > &indices) const  
*Construct sub-collection considering the given indices.*
- std::vector< double > **evaluateHomogeneousHomogeneousTimeseries** (const [HomogeneousTimeseriesMeasureType](#) &homogeneousTimeseriesMeasureType, const std::vector< double > &values, const std::vector< double > &timePoints, const std::vector< std::size\_t > &indices) const  
*Evaluate the given homogenous homogeneous timeseries.*
- std::vector< double > **computeHomogeneousComponentTimeSpans** (const std::vector< double > &timePoints, const std::vector< std::size\_t > &indices) const  
*Compute the time spans from a timepoints collection using the given start and end timepoint indices.*
- std::vector< double > **computeHomogeneousComponentValues** (const std::vector< double > &values, const std::vector< std::size\_t > &indices) const  
*Compute the values subsequence from a collection considering the given start and end indices.*

## Private Attributes

- const [SpatialTemporalTrace](#) & `trace`
- const [TypeSemanticsTable](#) & `typeSemanticsTable`

### 7.119.1 Detailed Description

Class for evaluating numeric measure collections.

Definition at line 15 of file `NumericMeasureCollectionVisitor.hpp`.

### 7.119.2 Constructor & Destructor Documentation

7.119.2.1 `multiscale::verification::NumericMeasureCollectionVisitor::NumericMeasureCollectionVisitor ( const SpatialTemporalTrace & trace, const TypeSemanticsTable & typeSemanticsTable ) [inline]`

Definition at line 25 of file `NumericMeasureCollectionVisitor.hpp`.

### 7.119.3 Member Function Documentation

7.119.3.1 `std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::computeHomogeneousComponentTimeSpans ( const std::vector< double > & timePoints, const std::vector< std::size_t > & indices ) const [inline], [private]`

Compute the time spans from a timepoints collection using the given start and end timepoint indices.

The provided indices collection contains (start, end) pairs where start/end denotes the starting/ending position of a time span in the initial collection which should be included in the resulting collection.

#### Parameters

|                         |                                                                                                               |
|-------------------------|---------------------------------------------------------------------------------------------------------------|
| <code>timePoints</code> | The given collection of timePoints                                                                            |
| <code>indices</code>    | The collection of both start and end indices pointing to the start and end positions in the values collection |

Definition at line 350 of file `NumericMeasureCollectionVisitor.hpp`.

Referenced by `evaluateHomogeneousHomogeneousTimeseries()`.

7.119.3.2 `std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::computeHomogeneousComponentValues ( const std::vector< double > & values, const std::vector< std::size_t > & indices ) const [inline], [private]`

Compute the values subsequence from a collection considering the given start and end indices.

The provided indices collection contains (start, end) pairs where start/end denotes the starting/ending position of a sequence of values in the initial collection which should be included in the resulting collection.

#### Parameters

|                      |                                                                                                               |
|----------------------|---------------------------------------------------------------------------------------------------------------|
| <code>values</code>  | The given collection of values                                                                                |
| <code>indices</code> | The collection of both start and end indices pointing to the start and end positions in the values collection |

Definition at line 378 of file `NumericMeasureCollectionVisitor.hpp`.

Referenced by `evaluateHomogeneousHomogeneousTimeseries()`.

```
7.119.3.3 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::evaluateChangeTemporal<-
 NumericCollection (const ChangeMeasureAttribute & changeMeasure, const std::vector< double > &
 temporalNumericCollectionValues) const [inline], [private]
```

Evaluate the temporal numeric collection values considering the given change measure.

**Parameters**

|                                                              |                                               |
|--------------------------------------------------------------|-----------------------------------------------|
| <i>changeMeasure</i>                                         | The given change measure                      |
| <i>temporal</i><br><i>Numeric</i><br><i>CollectionValues</i> | The values in the temporal numeric collection |

Definition at line 172 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::ChangeMeasureAttribute::changeMeasureType, and multiscale::verification::ChangeMeasureEvaluator::evaluate().

**7.119.3.4** `std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::evaluateHomogeneousHomogeneousTimeseries ( const HomogeneousTimeseriesMeasureType & homogenousTimeseriesMeasureType, const std::vector< double > & values, const std::vector< double > & timePoints, const std::vector< std::size_t > & indices ) const [inline], [private]`

Evaluate the given homogenous homogeneous timeseries.

Provided are the temporal numeric measure collection values, the corresponding timepoints and both start and end timepoints for timeseries components.

**Parameters**

|                                                               |                                                                                                                           |
|---------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>homogeneous</i><br><i>Timeseries</i><br><i>MeasureType</i> | The considered specific homogeneous timeseries measure type                                                               |
| <i>values</i>                                                 | The temporal numeric measure collection values                                                                            |
| <i>timePoints</i>                                             | The collection of timepoints corresponding to the temporal numeric collection values                                      |
| <i>indices</i>                                                | The indices pointing to the starting and ending positions of timeseries components in the collection of values/timepoints |

Definition at line 312 of file NumericMeasureCollectionVisitor.hpp.

References computeHomogeneousComponentTimeSpans(), computeHomogeneousComponentValues(), multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, MS\_throw, multiscale::verification::TimeSpan, and multiscale::verification::Values.

Referenced by operator()().

**7.119.3.5** `std::vector< double > multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollection ( const SpatialTemporalTrace & trace, unsigned long startTimepoint, unsigned long endTimepoint, const NumericMeasureType & numericMeasure ) const [inline], [private]`

Evaluate the temporal numeric measure collection considering the given spatio-temporal trace.

**Parameters**

|                       |                                      |
|-----------------------|--------------------------------------|
| <i>trace</i>          | The considered spatio-temporal trace |
| <i>startTimepoint</i> | The considered start timepoint value |
| <i>endTimepoint</i>   | The considered end timepoint value   |
| <i>numericMeasure</i> | The numeric measure to be evaluated  |

Definition at line 448 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::SpatialTemporalTrace::nextTimePointValue(), and multiscale::verification::SpatialTemporalTrace::setSubTrace().

Referenced by operator()().

```
7.119.3.6 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumeric←
MeasureCollectionTimepoints (const SpatialTemporalTrace & trace, unsigned long startTimepoint, unsigned
long endTimepoint) const [inline], [private]
```

Compute the collection of timepoints considering the given trace, and start and end timepoints.

**Parameters**

|                      |                                      |
|----------------------|--------------------------------------|
| <i>trace</i>         | The considered spatio-temporal trace |
| <i>startTimestep</i> | The considered start timepoint value |
| <i>endTimestep</i>   | The considered end timepoint value   |

Definition at line 146 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::SpatialTemporalTrace::nextTimePointValue(), and multiscale::verification::SpatialTemporalTrace::setSubTrace().

Referenced by operator()().

**7.119.3.7 std::vector<std::size\_t> multiscale::verification::NumericMeasureCollectionVisitor::evaluateTimeseriesComponent ( const std::vector< double > & *values*, const TimeseriesComponentAttribute & *timeseriesComponent* ) const [inline], [private]**

Evaluate the timeseries component considering the given collection of values.

**Parameters**

|                            |                                                                            |
|----------------------------|----------------------------------------------------------------------------|
| <i>values</i>              | The collection of real values for which timeseries components are computed |
| <i>timeseriesComponent</i> | The considered specific timeseries component type                          |

Definition at line 205 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::TimeseriesComponentAttribute::timeseriesComponent.

Referenced by operator()().

**7.119.3.8 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::evaluateTimeseriesTimeseriesComponent ( const TimeseriesMeasureType & *timeseriesMeasureType*, const std::vector< double > & *values*, const std::vector< double > & *timePoints*, const std::vector< std::size\_t > & *indices* ) const [inline], [private]**

Evaluate the given timeseries timeseries component.

Provided are the temporal numeric measure collection values, the corresponding timepoints and both start and end timepoints for timeseries components.

**Parameters**

|                              |                                                                                                                           |
|------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| <i>timeseriesMeasureType</i> | The considered specific timeseries measure type                                                                           |
| <i>values</i>                | The temporal numeric measure collection values                                                                            |
| <i>timePoints</i>            | The collection of timepoints corresponding to the temporal numeric collection values                                      |
| <i>indices</i>               | The indices pointing to the starting and ending positions of timeseries components in the collection of values/timepoints |

Definition at line 225 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::EnteringTime, multiscale::verification::EnteringValue, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, indicesSubCollection(), MS\_throw, and subCollection().

Referenced by operator()().

**7.119.3.9 std::vector<std::size\_t> multiscale::verification::NumericMeasureCollectionVisitor::indicesSubCollection ( const std::vector< std::size\_t > & *indices*, std::size\_t *StartPosition*, std::size\_t *Step* ) const [inline], [private]**

Compute the collection of sub-indices from the given collection of indices.

## Parameters

|                      |                                                                                                                                               |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| <i>indices</i>       | The collection of start and end indices                                                                                                       |
| <i>startPosition</i> | The first considered position from the indices collection                                                                                     |
| <i>step</i>          | The ("number of positions" + 1) between two consecutive elements from the initial indices collection which are included in the sub-collection |

Definition at line 261 of file NumericMeasureCollectionVisitor.hpp.

Referenced by `evaluateTimeseriesTimeseriesComponent()`.

**7.119.3.10 std::vector< double > multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const TemporalNumericCollectionAttribute & *temporalNumericCollection* ) const [inline]**

Overloading the "()" operator for the [TemporalNumericCollectionAttribute](#) alternative.

## Parameters

|                                  |                                 |
|----------------------------------|---------------------------------|
| <i>temporalNumericCollection</i> | The temporal numeric collection |
|----------------------------------|---------------------------------|

Definition at line 412 of file NumericMeasureCollectionVisitor.hpp.

References `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateTemporalNumericCollection()`, `trace`, and `typeSemanticsTable`.

**7.119.3.11 std::vector< double > multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const SpatialMeasureCollectionAttribute & *spatialMeasureCollection* ) const [inline]**

Overloading the "()" operator for the [SpatialMeasureCollectionAttribute](#) alternative.

## Parameters

|                                 |                                |
|---------------------------------|--------------------------------|
| <i>spatialMeasureCollection</i> | The spatial measure collection |
|---------------------------------|--------------------------------|

Definition at line 422 of file NumericMeasureCollectionVisitor.hpp.

References `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection()`.

**7.119.3.12 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const TemporalNumericMeasureCollectionAttribute & *temporalNumericMeasureCollection* ) const [inline]**

Overloading the "()" operator for the [TemporalNumericMeasureCollectionAttribute](#) alternative.

## Parameters

|                                         |                                         |
|-----------------------------------------|-----------------------------------------|
| <i>temporalNumericMeasureCollection</i> | The temporal numeric measure collection |
|-----------------------------------------|-----------------------------------------|

Definition at line 48 of file NumericMeasureCollectionVisitor.hpp.

References `multiscale::verification::TemporalNumericMeasureCollectionAttribute::endTimepoint`, `evaluateTemporalNumericMeasureCollection()`, `multiscale::verification::TemporalNumericMeasureCollectionAttribute::numericMeasure`, and `multiscale::verification::TemporalNumericMeasureCollectionAttribute::startTimepoint`.

7.119.3.13 `std::vector< double > multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const ChangeTemporalNumericCollectionAttribute & changeTemporalNumericCollection ) const [inline]`

Overloading the "()" operator for the [ChangeTemporalNumericCollectionAttribute](#) alternative.

## Parameters

|                                                                         |                                        |
|-------------------------------------------------------------------------|----------------------------------------|
| <i>change</i><br><i>Temporal</i><br><i>Numeric</i><br><i>Collection</i> | The change temporal numeric collection |
|-------------------------------------------------------------------------|----------------------------------------|

Definition at line 432 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::ChangeTemporalNumericCollectionAttribute::changeMeasure, multiscale::verification::NumericMeasureCollectionEvaluator::evaluateTemporalNumericCollection(), and multiscale::verification::ChangeTemporalNumericCollectionAttribute::temporalNumericCollection.

**7.119.3.14 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const TimeseriesTimeseriesComponentAttribute & *timeseriesTimeseriesComponent* ) const [inline]**

Overloading the "(") operator for the TimeseriesTimeseriesComponent alternative.

## Parameters

|                                                            |                                               |
|------------------------------------------------------------|-----------------------------------------------|
| <i>timeseries</i><br><i>Timeseries</i><br><i>Component</i> | The timeseries measure - timeseries component |
|------------------------------------------------------------|-----------------------------------------------|

Definition at line 69 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::TemporalNumericMeasureCollectionAttribute::endTimepoint, evaluate(TemporalNumericMeasureCollection()), evaluateTemporalNumericMeasureCollectionTimepoints(), evaluate(TimeseriesComponent()), evaluateTimeseriesTimeseriesComponent(), multiscale::verification::TemporalNumericMeasureCollectionAttribute::numericMeasure, multiscale::verification::TemporalNumericMeasureCollectionAttribute::startTimepoint, multiscale::verification::TimeseriesTimeseriesComponentAttribute::temporalNumericMeasureCollection, multiscale::verification::TimeseriesTimeseriesComponentAttribute::timeseriesComponent, multiscale::verification::TimeseriesTimeseriesComponentAttribute::timeseriesMeasure, and multiscale::verification::TimeseriesMeasureAttribute::timeseriesMeasure.

**7.119.3.15 std::vector<double> multiscale::verification::NumericMeasureCollectionVisitor::operator() ( const HomogeneousHomogeneousTimeseriesAttribute & *homogeneousHomogeneousTimeseries* ) const [inline]**

Overloading the "(") operator for the [HomogeneousHomogeneousTimeseriesAttribute](#) alternative.

## Parameters

|                                                               |                                                  |
|---------------------------------------------------------------|--------------------------------------------------|
| <i>homogeneous</i><br><i>Homogeneous</i><br><i>Timeseries</i> | The homogeneous homogeneous timeseries component |
|---------------------------------------------------------------|--------------------------------------------------|

Definition at line 99 of file NumericMeasureCollectionVisitor.hpp.

References multiscale::verification::TemporalNumericMeasureCollectionAttribute::endTimepoint, multiscale::verification::TimeseriesComponentEvaluator::evaluate(), evaluateHomogeneousHomogeneousTimeseries(), evaluateTemporalNumericMeasureCollection(), evaluateTemporalNumericMeasureCollectionTimepoints(), multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::homogeneousTimeseriesComponent, multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::homogeneousTimeseriesMeasure, multiscale::verification::HomogeneousTimeseriesMeasureAttribute::homogeneousTimeseriesMeasure, multiscale::verification::TemporalNumericMeasureCollectionAttribute::numericMeasure, multiscale::verification::TemporalNumericMeasureCollectionAttribute::startTimepoint, and multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute::temporalNumericMeasureCollection.

---

7.119.3.16 `template<typename T> std::vector<T> multiscale::verification::NumericMeasureCollectionVisitor::subCollection( const std::vector<T> & initialCollection, const std::vector<std::size_t> & indices ) const [inline], [private]`

Construct sub-collection considering the given indices.

Only valid indices will be considered. If an invalid index is encountered an exception is thrown.

#### Parameters

|                          |                                                                                                            |
|--------------------------|------------------------------------------------------------------------------------------------------------|
| <i>initialCollection</i> | The initial collection                                                                                     |
| <i>indices</i>           | The indices pointing to element positions from initial collection which will be included in sub-collection |

Definition at line 285 of file NumericMeasureCollectionVisitor.hpp.

Referenced by `evaluateTimeseriesTimeseriesComponent()`.

## 7.119.4 Member Data Documentation

7.119.4.1 `const SpatialTemporalTrace& multiscale::verification::NumericMeasureCollectionVisitor::trace [private]`

The considered spatial temporal trace

Definition at line 19 of file NumericMeasureCollectionVisitor.hpp.

Referenced by `operator()()`.

7.119.4.2 `const TypeSemanticsTable& multiscale::verification::NumericMeasureCollectionVisitor::typeSemanticsTable [private]`

The type semantics table

Definition at line 20 of file NumericMeasureCollectionVisitor.hpp.

Referenced by `operator()()`.

The documentation for this class was generated from the following file:

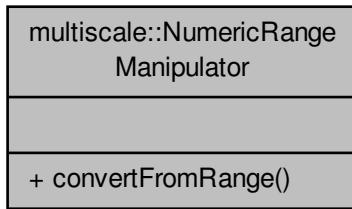
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericMeasureCollectionVisitor.hpp](#)

## 7.120 multiscale::NumericRangeManipulator Class Reference

Operations for ranges of numeric values.

```
#include <NumericRangeManipulator.hpp>
```

Collaboration diagram for multiscale::NumericRangeManipulator:



## Static Public Member Functions

- template<class T , class U >  
static U [convertFromRange](#) ( T oldRangeMin, T oldRangeMax, U newRangeMin, U newRangeMax, T oldValue )  
*Convert a value from an old range to a new one.*

### 7.120.1 Detailed Description

Operations for ranges of numeric values.

Definition at line 7 of file NumericRangeManipulator.hpp.

### 7.120.2 Member Function Documentation

7.120.2.1 template<class T , class U > static U multiscale::NumericRangeManipulator::convertFromRange ( T oldRangeMin, T oldRangeMax, U newRangeMin, U newRangeMax, T oldValue ) [inline], [static]

Convert a value from an old range to a new one.

#### Parameters

|                    |                              |
|--------------------|------------------------------|
| <i>oldRangeMin</i> | The minimum of the old range |
| <i>oldRangeMax</i> | The maximum of the old range |
| <i>newRangeMin</i> | The minimum of the new range |
| <i>newRangeMax</i> | The maximum of the new range |
| <i>oldValue</i>    | The old value                |

Definition at line 20 of file NumericRangeManipulator.hpp.

The documentation for this class was generated from the following file:

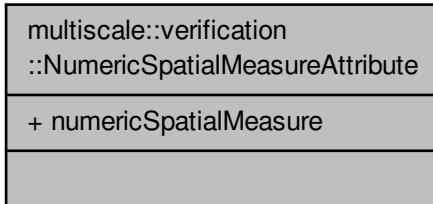
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[NumericRangeManipulator.hpp](#)

## 7.121 multiscale::verification::NumericSpatialMeasureAttribute Class Reference

Class for representing a numeric spatial measure attribute.

```
#include <NumericSpatialMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericSpatialMeasureAttribute:



## Public Attributes

- [NumericSpatialMeasureType numericSpatialMeasure](#)

### 7.121.1 Detailed Description

Class for representing a numeric spatial measure attribute.

Definition at line 29 of file [NumericSpatialMeasureAttribute.hpp](#).

### 7.121.2 Member Data Documentation

#### 7.121.2.1 [NumericSpatialMeasureType multiscale::verification::NumericSpatialMeasureAttribute::numericSpatialMeasure](#)

The numeric spatial measure

Definition at line 33 of file [NumericSpatialMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::NumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

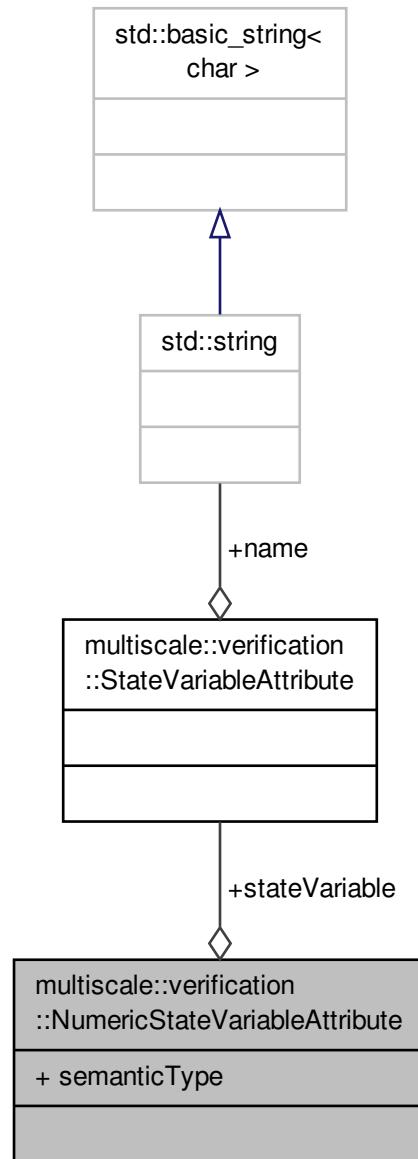
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericSpatialMeasureAttribute.hpp](#)

## 7.122 [multiscale::verification::NumericStateVariableAttribute Class Reference](#)

Class for representing a numeric state variable attribute.

```
#include <NumericStateVariableAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericStateVariableAttribute:



## Public Attributes

- `StateVariableAttribute stateVariable`
- `boost::optional< SemanticTypeAttribute > semanticType`

### 7.122.1 Detailed Description

Class for representing a numeric state variable attribute.

Definition at line 16 of file NumericStateVariableAttribute.hpp.

## 7.122.2 Member Data Documentation

### 7.122.2.1 boost::optional<SemanticTypeAttribute> multiscale::verification::NumericStateVariableAttribute::semanticType

The semantic type

Definition at line 23 of file NumericStateVariableAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

### 7.122.2.2 StateVariableAttribute multiscale::verification::NumericStateVariableAttribute::stateVariable

The state variable

Definition at line 21 of file NumericStateVariableAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

The documentation for this class was generated from the following file:

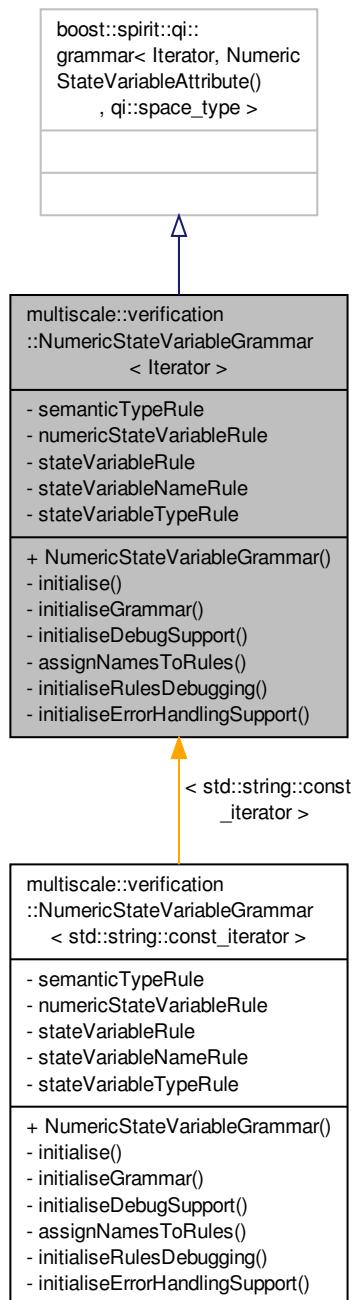
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp

## 7.123 multiscale::verification::NumericStateVariableGrammar< Iterator > Class Template Reference

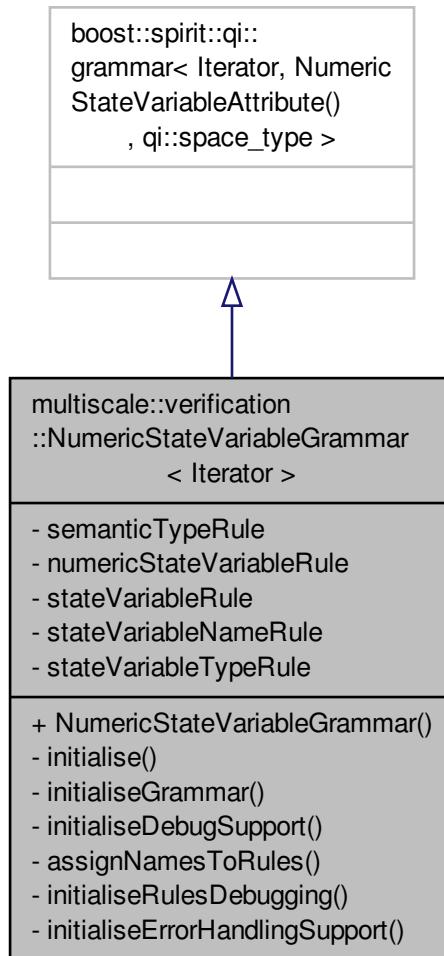
The grammar for parsing numeric state variable statements.

```
#include <NumericStateVariableGrammar.hpp>
```

Inheritance diagram for multiscale::verification::NumericStateVariableGrammar< Iterator >:



Collaboration diagram for multiscale::verification::NumericStateVariableGrammar< Iterator >:



## Public Member Functions

- [NumericStateVariableGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)

- *Initialise the debugging of rules.*
- void [initialiseErrorHandlingSupport \(\)](#)  
*Initialise the error handling routines.*

## Private Attributes

- [SemanticTypeStringGrammar< Iterator >::semanticTypeRule](#)
- [qi::rule< Iterator, NumericStateVariableAttribute\(\), qi::space\\_type >::numericStateVariableRule](#)
- [qi::rule< Iterator, StateVariableAttribute\(\), qi::space\\_type >::stateVariableRule](#)
- [qi::rule< Iterator, std::string\(\), qi::space\\_type >::stateVariableNameRule](#)
- [qi::rule< Iterator, SemanticTypeAttribute\(\), qi::space\\_type >::stateVariableTypeRule](#)

### 7.123.1 Detailed Description

```
template<typename Iterator> class multiscale::verification::NumericStateVariableGrammar< Iterator >
```

The grammar for parsing numeric state variable statements.

Definition at line 29 of file NumericStateVariableGrammar.hpp.

### 7.123.2 Constructor & Destructor Documentation

7.123.2.1 template<typename Iterator > multiscale::verification::NumericStateVariableGrammar< Iterator >::NumericStateVariableGrammar( )

Definition at line 28 of file NumericStateVariableGrammarDefinition.hpp.

References multiscale::verification::NumericStateVariableGrammar< Iterator >::initialise().

### 7.123.3 Member Function Documentation

7.123.3.1 template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::assignNamesToRules( ) [private]

Assign names to the rules.

Definition at line 75 of file NumericStateVariableGrammarDefinition.hpp.

7.123.3.2 template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::initialise( ) [private]

Initialisation function.

Definition at line 35 of file NumericStateVariableGrammarDefinition.hpp.

Referenced by multiscale::verification::NumericStateVariableGrammar< Iterator >::NumericStateVariableGrammar().

---

7.123.3.3 `template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::initialiseDebugSupport( ) [private]`

Initialise debug support.

Definition at line 66 of file NumericStateVariableGrammarDefinition.hpp.

7.123.3.4 `template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]`

Initialise the error handling routines.

Definition at line 93 of file NumericStateVariableGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.123.3.5 `template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::initialiseGrammar( ) [private]`

Initialise the grammar.

Definition at line 43 of file NumericStateVariableGrammarDefinition.hpp.

7.123.3.6 `template<typename Iterator > void multiscale::verification::NumericStateVariableGrammar< Iterator >::initialiseRulesDebugging( ) [private]`

Initialise the debugging of rules.

Definition at line 84 of file NumericStateVariableGrammarDefinition.hpp.

## 7.123.4 Member Data Documentation

7.123.4.1 `template<typename Iterator> qi::rule<Iterator, NumericStateVariableAttribute(), qi::space_type> multiscale::verification::NumericStateVariableGrammar< Iterator >::numericStateVariableRule [private]`

The rule for parsing a numeric state variable

Definition at line 42 of file NumericStateVariableGrammar.hpp.

7.123.4.2 `template<typename Iterator> SemanticTypeStringGrammar<Iterator> multiscale::verification::NumericStateVariableGrammar< Iterator >::semanticTypeRule [private]`

The grammar for parsing semantic types

Definition at line 37 of file NumericStateVariableGrammar.hpp.

7.123.4.3 `template<typename Iterator> qi::rule<Iterator, std::string(), qi::space_type> multiscale::verification::NumericStateVariableGrammar< Iterator >::stateVariableNameRule [private]`

The rule for parsing the name of a state variable without escaping white space

Definition at line 46 of file NumericStateVariableGrammar.hpp.

```
7.123.4.4 template<typename Iterator> qi::rule<Iterator, StateVariableAttribute(), qi::space_type>
multiscale::verification::NumericStateVariableGrammar< Iterator >::stateVariableRule [private]
```

The rule for parsing a state variable

Definition at line 44 of file NumericStateVariableGrammar.hpp.

```
7.123.4.5 template<typename Iterator> qi::rule<Iterator, SemanticTypeAttribute(), qi::space_type>
multiscale::verification::NumericStateVariableGrammar< Iterator >::stateVariableTypeRule
[private]
```

The rule for parsing a state variable type

Definition at line 49 of file NumericStateVariableGrammar.hpp.

The documentation for this class was generated from the following files:

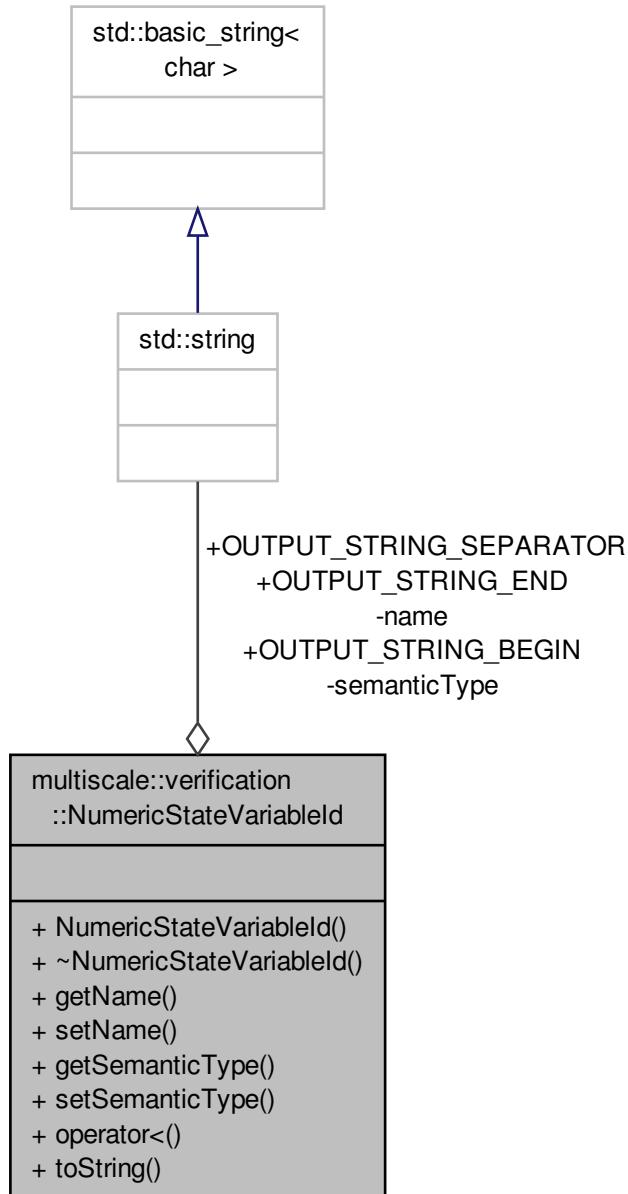
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[NumericStateVariableGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[NumericStateVariableGrammarDefinition.hpp](#)

## 7.124 multiscale::verification::NumericStateVariableId Class Reference

Class for representing the identity (name, type) of a numeric state variable.

```
#include <NumericStateVariableId.hpp>
```

Collaboration diagram for multiscale::verification::NumericStateVariableId:



## Public Member Functions

- `NumericStateVariableId (const std::string &name, const std::string &semanticType=SemanticType::DEFAUL_LT_VALUE)`
- `~NumericStateVariableId ()`
- `const std::string & getName () const`

*Get the name of the numeric state variable.*
- `void setName (const std::string &name)`

*Set the name of the numeric state variable.*

- std::string [getSemanticType \(\) const](#)  
*Get the semantic type of the numeric state variable.*
- void [setSemanticType \(const std::string &semanticType\)](#)  
*Set the semantic type of the numeric state variable.*
- bool [operator< \(const NumericStateVariableId &rhs\) const](#)  
*Overload the < operator.*
- std::string [toString \(\) const](#)  
*Return the string representation of the numeric state variable identity.*

## Static Public Attributes

- static const std::string [OUTPUT\\_STRING\\_BEGIN = "\("](#)
- static const std::string [OUTPUT\\_STRING\\_SEPARATOR = ", "](#)
- static const std::string [OUTPUT\\_STRING\\_END = "\)"](#)

## Private Attributes

- std::string [name](#)
- std::string [semanticType](#)

### 7.124.1 Detailed Description

Class for representing the identity (name, type) of a numeric state variable.

Definition at line 14 of file NumericStateVariableId.hpp.

### 7.124.2 Constructor & Destructor Documentation

#### 7.124.2.1 NumericStateVariableId::NumericStateVariableId ( const std::string & name, const std::string & semanticType = SemanticType::DEFAULT\_VALUE )

Definition at line 8 of file NumericStateVariableId.cpp.

#### 7.124.2.2 NumericStateVariableId::~NumericStateVariableId ( )

Definition at line 12 of file NumericStateVariableId.cpp.

### 7.124.3 Member Function Documentation

#### 7.124.3.1 const std::string & NumericStateVariableId::getName ( ) const

Get the name of the numeric state variable.

Definition at line 14 of file NumericStateVariableId.cpp.

References name.

#### 7.124.3.2 std::string NumericStateVariableId::getSemanticType ( ) const

Get the semantic type of the numeric state variable.

Definition at line 22 of file NumericStateVariableId.cpp.

References semanticType.

### 7.124.3.3 bool NumericStateVariableId::operator< ( const NumericStateVariableId & rhs ) const

Overload the < operator.

#### Parameters

|            |                                                                       |
|------------|-----------------------------------------------------------------------|
| <i>rhs</i> | The right hand side numeric state variable identity i.e. (lhs < rhs)? |
|------------|-----------------------------------------------------------------------|

Definition at line 30 of file NumericStateVariableId.cpp.

References name, and semanticType.

### 7.124.3.4 void NumericStateVariableId::setName ( const std::string & name )

Set the name of the numeric state variable.

#### Parameters

|             |                                        |
|-------------|----------------------------------------|
| <i>name</i> | The name of the numeric state variable |
|-------------|----------------------------------------|

Definition at line 18 of file NumericStateVariableId.cpp.

References name.

### 7.124.3.5 void NumericStateVariableId::setSemanticType ( const std::string & semanticType )

Set the semantic type of the numeric state variable.

#### Parameters

|                     |                                                 |
|---------------------|-------------------------------------------------|
| <i>semanticType</i> | The semantic type of the numeric state variable |
|---------------------|-------------------------------------------------|

Definition at line 26 of file NumericStateVariableId.cpp.

References semanticType.

### 7.124.3.6 std::string NumericStateVariableId::toString ( ) const

Return the string representation of the numeric state variable identity.

Definition at line 42 of file NumericStateVariableId.cpp.

References name, OUTPUT\_STRING\_BEGIN, OUTPUT\_STRING\_END, OUTPUT\_STRING\_SEPARATOR, and semanticType.

## 7.124.4 Member Data Documentation

### 7.124.4.1 std::string multiscale::verification::NumericStateVariableId::name [private]

The name of the numeric state variable

Definition at line 18 of file NumericStateVariableId.hpp.

Referenced by getName(), operator<(), setName(), and toString().

### 7.124.4.2 const std::string NumericStateVariableId::OUTPUT\_STRING\_BEGIN = "(" [static]

Definition at line 58 of file NumericStateVariableId.hpp.

Referenced by toString().

7.124.4.3 const std::string NumericStateVariableId::OUTPUT\_STRING\_END = ")" [static]

Definition at line 60 of file NumericStateVariableId.hpp.

Referenced by `toString()`.

7.124.4.4 const std::string NumericStateVariableId::OUTPUT\_STRING\_SEPARATOR = "," [static]

Definition at line 59 of file NumericStateVariableId.hpp.

Referenced by `toString()`.

7.124.4.5 std::string multiscale::verification::NumericStateVariableId::semanticType [private]

The semantic type of the numeric state variable

Definition at line 19 of file NumericStateVariableId.hpp.

Referenced by `getSemanticType()`, `operator<()`, `setSemanticType()`, and `toString()`.

The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[NumericStateVariableId.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[NumericStateVariableId.cpp](#)

## 7.125 multiscaletest::NumericStateVariableTraceTest Class Reference

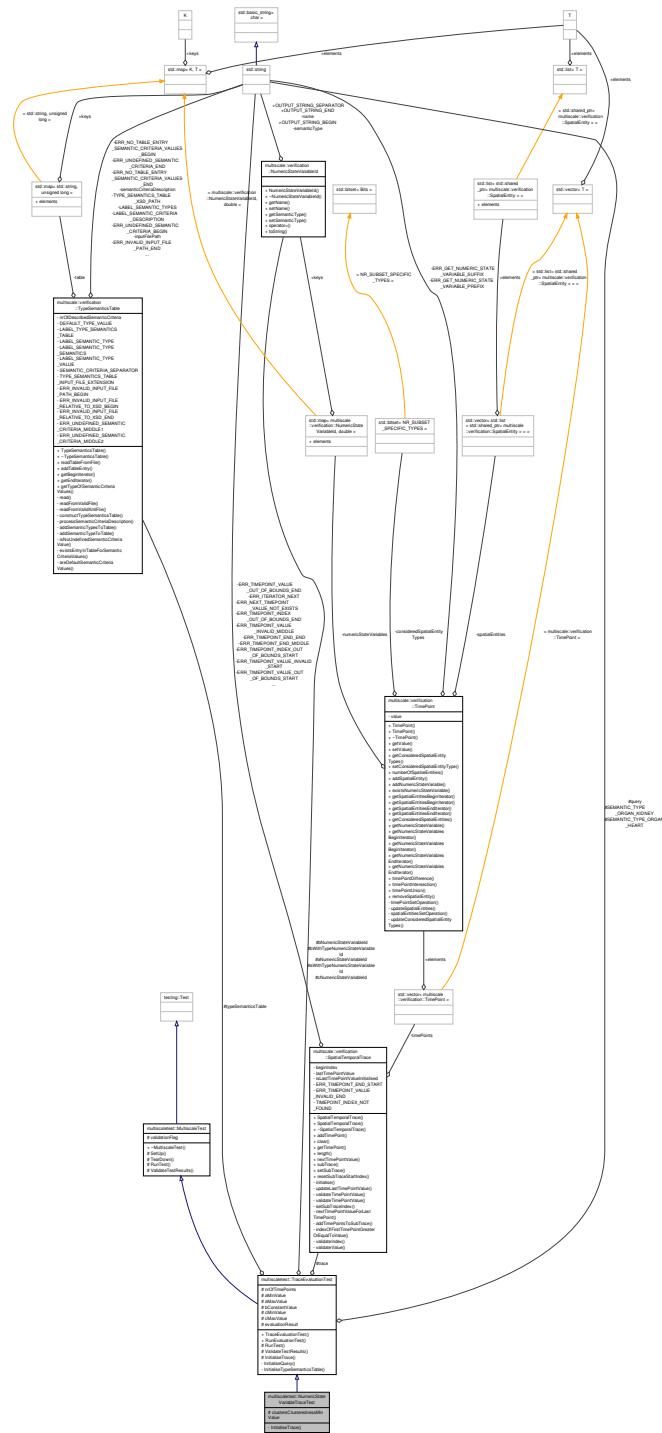
Class for testing evaluation of numeric state variable-only traces.

```
#include <NumericStateVariableTraceTest.hpp>
```

Inheritance diagram for multiscaletest::NumericStateVariableTraceTest:



## Collaboration diagram for multiscaletest::NumericStateVariableTraceTest:



## Protected Attributes

- double `clustersClusterednessMinValue`

## Private Member Functions

- virtual void InitialiseTrace () override

*Initialise the trace.*

## Additional Inherited Members

### 7.125.1 Detailed Description

Class for testing evaluation of numeric state variable-only traces.

Definition at line 22 of file NumericStateVariableTraceTest.hpp.

### 7.125.2 Member Function Documentation

#### 7.125.2.1 void multiscaletest::NumericStateVariableTraceTest::InitialiseTrace( ) [override], [private], [virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 35 of file NumericStateVariableTraceTest.hpp.

### 7.125.3 Member Data Documentation

#### 7.125.3.1 double multiscaletest::NumericStateVariableTraceTest::clustersClusterednessMinValue [protected]

The minimum clusteredness value for the cluster spatial entity type

Definition at line 26 of file NumericStateVariableTraceTest.hpp.

The documentation for this class was generated from the following file:

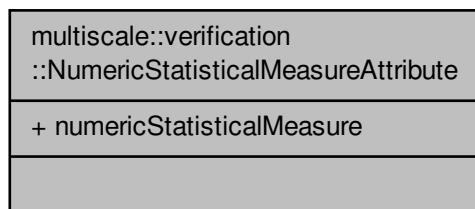
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[NumericStateVariableTraceTest.hpp](#)

## 7.126 multiscale::verification::NumericStatisticalMeasureAttribute Class Reference

Class for representing a numeric statistical measure attribute.

```
#include <NumericStatisticalMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericStatisticalMeasureAttribute:



## Public Attributes

- [NumericStatisticalMeasureType numericStatisticalMeasure](#)

### 7.126.1 Detailed Description

Class for representing a numeric statistical measure attribute.

Definition at line 24 of file [NumericStatisticalMeasureAttribute.hpp](#).

### 7.126.2 Member Data Documentation

#### 7.126.2.1 [NumericStatisticalMeasureType multiscale::verification::NumericStatisticalMeasureAttribute::numericStatisticalMeasure](#)

The numeric statistical measure

Definition at line 28 of file [NumericStatisticalMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

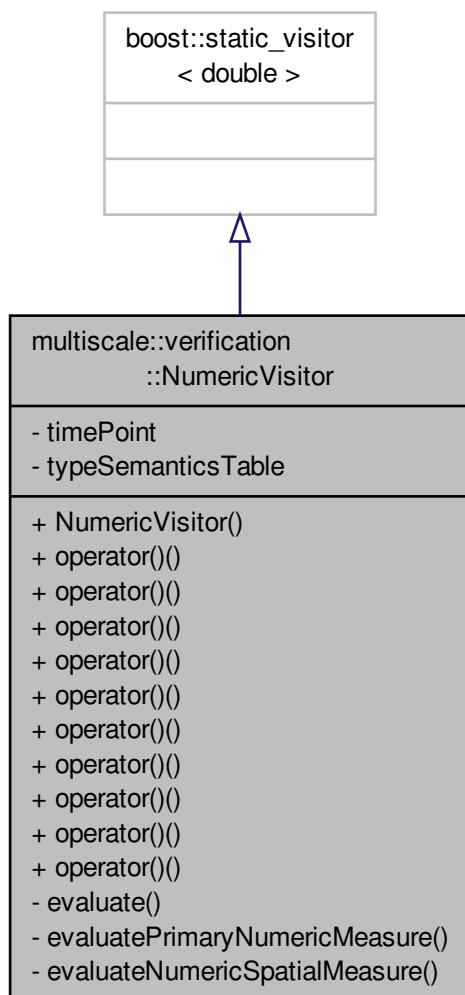
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStatisticalMeasureAttribute.hpp](#)

## 7.127 multiscale::verification::NumericVisitor Class Reference

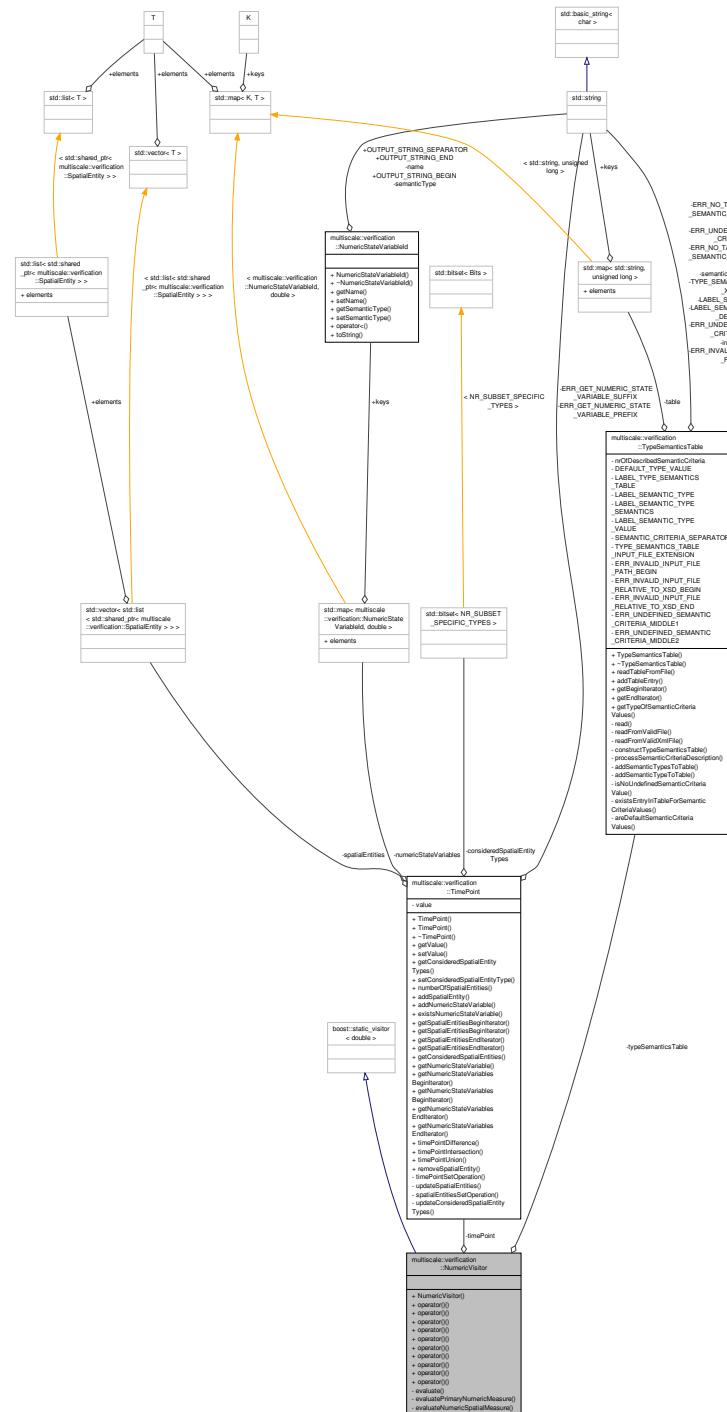
Class for evaluating numeric measures.

```
#include <NumericVisitor.hpp>
```

Inheritance diagram for multiscale::verification::NumericVisitor:



## Collaboration diagram for multiscale::verification::NumericVisitor:



## Public Member Functions

- `NumericVisitor` (const `TimePoint` &`timePoint`, const `TypeSemanticsTable` &`typeSemanticsTable`)
  - `double operator()` (const `NumericMeasureAttribute` &`numericMeasure`) const

Overloading the "/" operator for the [NumericMeasureAttribute](#) alternative

- double **operator()** (const PrimaryNumericMeasureAttribute &primaryNumericMeasure) const

Overloading the “()” operator for the *PrimaryNumericMeasureAttribute* alternative

- double operator() (double realNumber) const  
*Overloading the "()" operator for the real number alternative.*
- double operator() (const NumericStateVariableAttribute &numericStateVariable) const  
*Overloading the "()" operator for the [NumericStateVariableAttribute](#) alternative.*
- double operator() (const NumericSpatialMeasureAttribute &numericSpatialMeasure) const  
*Overloading the "()" operator for the [NumericSpatialMeasureAttribute](#) alternative.*
- double operator() (const UnaryNumericNumericAttribute &unaryNumericNumericMeasure) const  
*Overloading the "()" operator for the [UnaryNumericNumericAttribute](#) alternative.*
- double operator() (const BinaryNumericNumericAttribute &binaryNumericNumericMeasure) const  
*Overloading the "()" operator for the [BinaryNumericNumericAttribute](#) alternative.*
- double operator() (const UnaryStatisticalSpatialAttribute &unaryStatisticalSpatialAttribute) const  
*Overloading the "()" operator for the [UnaryStatisticalSpatialAttribute](#) alternative.*
- double operator() (const BinaryStatisticalSpatialAttribute &binaryStatisticalSpatialAttribute) const  
*Overloading the "()" operator for the [BinaryStatisticalSpatialAttribute](#) alternative.*
- double operator() (const BinaryStatisticalQuantileSpatialAttribute &binaryStatisticalQuantileSpatialAttribute) const  
*Overloading the "()" operator for the [BinaryStatisticalQuantileSpatialAttribute](#) alternative.*

## Private Member Functions

- double evaluate (const NumericMeasureType &numericMeasure) const  
*Evaluate the given numeric measure considering the timePoint field.*
- double evaluatePrimaryNumericMeasure (const PrimaryNumericMeasureAttributeType &primaryNumericMeasure) const  
*Evaluate the given primary numeric measure considering the timePoint field.*
- double evaluateNumericSpatialMeasure (const NumericSpatialMeasureType &numericSpatialMeasure) const  
*Evaluate the given numeric spatial measure considering the timePoint field.*

## Private Attributes

- const TimePoint & timePoint
- const TypeSemanticsTable & typeSemanticsTable

### 7.127.1 Detailed Description

Class for evaluating numeric measures.

Definition at line 19 of file NumericVisitor.hpp.

### 7.127.2 Constructor & Destructor Documentation

#### 7.127.2.1 multiscale::verification::NumericVisitor::NumericVisitor ( const TimePoint & timePoint, const TypeSemanticsTable & typeSemanticsTable ) [inline]

Definition at line 28 of file NumericVisitor.hpp.

Referenced by evaluate(), evaluateNumericSpatialMeasure(), and evaluatePrimaryNumericMeasure().

### 7.127.3 Member Function Documentation

7.127.3.1 double multiscale::verification::NumericVisitor::evaluate ( const NumericMeasureType & *numericMeasure* )  
const [inline], [private]

Evaluate the given numeric measure considering the timePoint field.

**Parameters**

|                       |                           |
|-----------------------|---------------------------|
| <i>numericMeasure</i> | The given numeric measure |
|-----------------------|---------------------------|

Definition at line 148 of file NumericVisitor.hpp.

References NumericVisitor().

Referenced by operator()().

**7.127.3.2 double multiscale::verification::NumericVisitor::evaluateNumericSpatialMeasure ( const NumericSpatialMeasureType & *numericSpatialMeasure* ) const [inline], [private]**

Evaluate the given numeric spatial measure considering the timePoint field.

**Parameters**

|                              |                                   |
|------------------------------|-----------------------------------|
| <i>numericSpatialMeasure</i> | The given numeric spatial measure |
|------------------------------|-----------------------------------|

Definition at line 167 of file NumericVisitor.hpp.

References NumericVisitor().

Referenced by operator()().

**7.127.3.3 double multiscale::verification::NumericVisitor::evaluatePrimaryNumericMeasure ( const PrimaryNumericMeasureAttributeType & *primaryNumericMeasure* ) const [inline], [private]**

Evaluate the given primary numeric measure considering the timePoint field.

**Parameters**

|                              |                                   |
|------------------------------|-----------------------------------|
| <i>primaryNumericMeasure</i> | The given primary numeric measure |
|------------------------------|-----------------------------------|

Definition at line 157 of file NumericVisitor.hpp.

References NumericVisitor().

Referenced by operator()().

**7.127.3.4 double multiscale::verification::NumericVisitor::operator() ( const NumericMeasureAttribute & *numericMeasure* ) const [inline]**

Overloading the "(" operator for the [NumericMeasureAttribute](#) alternative.

**Parameters**

|                       |                     |
|-----------------------|---------------------|
| <i>numericMeasure</i> | The numeric measure |
|-----------------------|---------------------|

Definition at line 37 of file NumericVisitor.hpp.

References evaluate(), and multiscale::verification::NumericMeasureAttribute::numericMeasure.

**7.127.3.5 double multiscale::verification::NumericVisitor::operator() ( const PrimaryNumericMeasureAttribute & *primaryNumericMeasure* ) const [inline]**

Overloading the "(" operator for the [PrimaryNumericMeasureAttribute](#) alternative.

**Parameters**

|                                                    |                             |
|----------------------------------------------------|-----------------------------|
| <i>primary</i><br><i>Numeric</i><br><i>Measure</i> | The primary numeric measure |
|----------------------------------------------------|-----------------------------|

Definition at line 46 of file NumericVisitor.hpp.

References `evaluatePrimaryNumericMeasure()`, and `multiscale::verification::PrimaryNumericMeasureAttribute`  
::`primaryNumericMeasure`.

**7.127.3.6 double multiscale::verification::NumericVisitor::operator() ( double *realNumber* ) const [inline]**

Overloading the "(" operator for the real number alternative.

**Parameters**

|                   |                 |
|-------------------|-----------------|
| <i>realNumber</i> | The real number |
|-------------------|-----------------|

Definition at line 55 of file NumericVisitor.hpp.

**7.127.3.7 double multiscale::verification::NumericVisitor::operator() ( const NumericStateVariableAttribute & *numericStateVariable* ) const [inline]**

Overloading the "(" operator for the `NumericStateVariableAttribute` alternative.

**Parameters**

|                                        |                            |
|----------------------------------------|----------------------------|
| <i>numericState</i><br><i>Variable</i> | The numeric state variable |
|----------------------------------------|----------------------------|

Definition at line 64 of file NumericVisitor.hpp.

References `multiscale::verification::TimePoint::getNumericStateVariable()`, `multiscale::verification::StateVariable`  
Attribute::`name`, `multiscale::verification::NumericStateVariableAttribute::semanticType`, and `multiscale::verification`  
::`NumericStateVariableAttribute::stateVariable`.

**7.127.3.8 double multiscale::verification::NumericVisitor::operator() ( const NumericSpatialMeasureAttribute & *numericSpatialMeasure* ) const [inline]**

Overloading the "(" operator for the `NumericSpatialAttribute` alternative.

**Parameters**

|                                         |                                       |
|-----------------------------------------|---------------------------------------|
| <i>numericSpatial</i><br><i>Measure</i> | The numeric spatial measure attribute |
|-----------------------------------------|---------------------------------------|

Definition at line 80 of file NumericVisitor.hpp.

References `evaluateNumericSpatialMeasure()`, and `multiscale::verification::NumericSpatialMeasureAttribute`  
::`numericSpatialMeasure`.

**7.127.3.9 double multiscale::verification::NumericVisitor::operator() ( const UnaryNumericNumericAttribute & *unaryNumericNumericMeasure* ) const [inline]**

Overloading the "(" operator for the `UnaryNumericNumericAttribute` alternative.

**Parameters**

|                                                         |                                   |
|---------------------------------------------------------|-----------------------------------|
| <i>unaryNumeric</i><br><i>Numeric</i><br><i>Measure</i> | The unary numeric numeric measure |
|---------------------------------------------------------|-----------------------------------|

Definition at line 89 of file NumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluate(), and multiscale::verification::UnaryNumericNumericAttribute::numericMeasure.

**7.127.3.10 double multiscale::verification::NumericVisitor::operator() ( const BinaryNumericNumericAttribute & *binaryNumericNumericMeasure* ) const [inline]**

Overloading the "(" operator for the [BinaryNumericNumericAttribute](#) alternative.

**Parameters**

|                                                          |                                    |
|----------------------------------------------------------|------------------------------------|
| <i>binaryNumeric</i><br><i>Numeric</i><br><i>Measure</i> | The binary numeric numeric measure |
|----------------------------------------------------------|------------------------------------|

Definition at line 104 of file NumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluate(), multiscale::verification::BinaryNumericNumericAttribute::firstNumericMeasure, and multiscale::verification::BinaryNumericNumericAttribute::secondNumericMeasure.

**7.127.3.11 double multiscale::verification::NumericVisitor::operator() ( const UnaryStatisticalSpatialAttribute & *unaryStatisticalSpatialAttribute* ) const [inline]**

Overloading the "(" operator for the [UnaryStatisticalSpatialAttribute](#) alternative.

**Parameters**

|                                                               |                                         |
|---------------------------------------------------------------|-----------------------------------------|
| <i>unary</i><br><i>Statistical</i><br><i>SpatialAttribute</i> | The unary statistical spatial attribute |
|---------------------------------------------------------------|-----------------------------------------|

Definition at line 188 of file NumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::NumericMeasure::CollectionEvaluator::evaluateSpatialMeasureCollection(), multiscale::verification::UnaryStatisticalSpatialAttribute::spatialMeasureCollection, timePoint, typeSemanticsTable, multiscale::verification::UnaryStatisticalSpatialAttribute::unaryStatisticalMeasure, and multiscale::verification::UnaryStatisticalMeasureAttribute::unaryStatisticalMeasureType.

**7.127.3.12 double multiscale::verification::NumericVisitor::operator() ( const BinaryStatisticalSpatialAttribute & *binaryStatisticalSpatialAttribute* ) const [inline]**

Overloading the "(" operator for the [BinaryStatisticalSpatialAttribute](#) alternative.

**Parameters**

|                                                                |                                          |
|----------------------------------------------------------------|------------------------------------------|
| <i>binary</i><br><i>Statistical</i><br><i>SpatialAttribute</i> | The binary statistical spatial attribute |
|----------------------------------------------------------------|------------------------------------------|

Definition at line 203 of file NumericVisitor.hpp.

References multiscale::verification::BinaryStatisticalSpatialAttribute::binaryStatisticalMeasure, multiscale::verification::BinaryStatisticalMeasureAttribute::binaryStatisticalMeasureType, multiscale::verification::Numeric

Evaluator::evaluate(), multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection(), multiscale::verification::BinaryStatisticalSpatialAttribute::firstSpatialMeasureCollection, and multiscale::verification::BinaryStatisticalSpatialAttribute::secondSpatialMeasureCollection.

### 7.127.3.13 double multiscale::verification::NumericVisitor::operator() ( const BinaryStatisticalQuantileSpatialAttribute & *binaryStatisticalQuantileSpatialAttribute* ) const [inline]

Overloading the "(") operator for the [BinaryStatisticalQuantileSpatialAttribute](#) alternative.

#### Parameters

|                                                                                   |                                                   |
|-----------------------------------------------------------------------------------|---------------------------------------------------|
| <i>binary</i><br><i>Statistical</i><br><i>Quantile</i><br><i>SpatialAttribute</i> | The binary statistical quantile spatial attribute |
|-----------------------------------------------------------------------------------|---------------------------------------------------|

Definition at line 224 of file NumericVisitor.hpp.

References multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::binaryStatisticalQuantileMeasure, multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::NumericMeasureCollection::Evaluator::evaluateSpatialMeasureCollection(), multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::parameter, and multiscale::verification::BinaryStatisticalQuantileSpatialAttribute::spatialMeasureCollection.

## 7.127.4 Member Data Documentation

### 7.127.4.1 const TimePoint& multiscale::verification::NumericVisitor::timePoint [private]

The considered timepoint

Definition at line 23 of file NumericVisitor.hpp.

Referenced by operator()().

### 7.127.4.2 const TypeSemanticsTable& multiscale::verification::NumericVisitor::typeSemanticsTable [private]

The type semantics table

Definition at line 24 of file NumericVisitor.hpp.

Referenced by operator()().

The documentation for this class was generated from the following file:

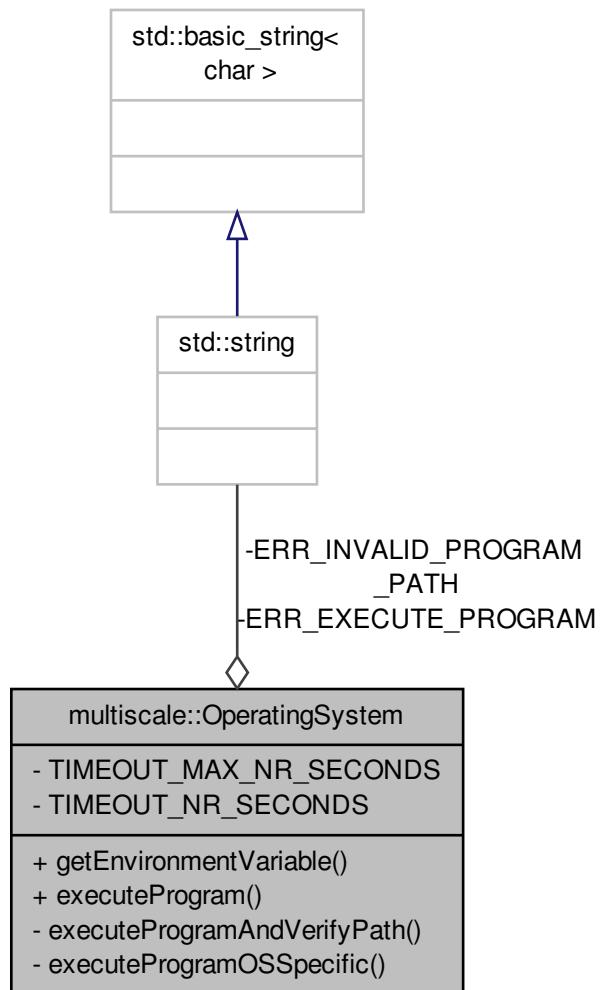
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericVisitor.hpp](#)

## 7.128 multiscale::OperatingSystem Class Reference

Class for executing operating system related functions.

```
#include <OperatingSystem.hpp>
```

## Collaboration diagram for multiscale::OperatingSystem:



## Static Public Member Functions

- static std::string [getEnvironmentVariable](#) (const std::string &name)  
*Get the value of the environment variable having the given name.*
  - static void [executeProgram](#) (const std::string &path)  
*Create a child process and execute the program with the given path.*

## Static Private Member Functions

- static void `executeProgramAndVerifyPath` (const std::string &path)  
*Create a child process and execute the program with the given path if the provided path is valid.*
  - static void `executeProgramOSSpecific` (const std::string &path)  
*Create a child process and execute the program with the given path considering the specific Operating system.*

## Static Private Attributes

- static const std::string **ERR\_EXECUTE\_PROGRAM** = "The process executing the program located at the following path could not be created: "
- static const std::string **ERR\_INVALID\_PROGRAM\_PATH** = "The process was not created because the provided program path is invalid: "
- static const unsigned int **TIMEOUT\_MAX\_NR\_SECONDS** = 100
- static const unsigned int **TIMEOUT\_NR\_SECONDS** = 1

### 7.128.1 Detailed Description

Class for executing operating system related functions.

Definition at line 23 of file OperatingSystem.hpp.

### 7.128.2 Member Function Documentation

#### 7.128.2.1 void OperatingSystem::executeProgram ( const std::string & *path* ) [static]

Create a child process and execute the program with the given path.

##### Parameters

|             |                                                |
|-------------|------------------------------------------------|
| <i>path</i> | The path to the program which will be executed |
|-------------|------------------------------------------------|

Definition at line 24 of file OperatingSystem.cpp.

References executeProgramAndVerifyPath(), multiscale::ExceptionHandler::printDetailedErrorMessage(), multiscale::ConsolePrinter::printWarningMessage(), and multiscale::MultiscaleException::rawMessage().

Referenced by multiscale::verification::ModelCheckingManager::executeExtraEvaluationProgramAndPrintMessage(), and main().

#### 7.128.2.2 void OperatingSystem::executeProgramAndVerifyPath ( const std::string & *path* ) [static], [private]

Create a child process and execute the program with the given path if the provided path is valid.

##### Parameters

|             |                                                |
|-------------|------------------------------------------------|
| <i>path</i> | The path to the program which will be executed |
|-------------|------------------------------------------------|

Definition at line 34 of file OperatingSystem.cpp.

References ERR\_INVALID\_PROGRAM\_PATH, executeProgramOSSpecific(), multiscale::Filesystem::isValidFilePath(), and MS\_throw.

Referenced by executeProgram().

#### 7.128.2.3 static void multiscale::OperatingSystem::executeProgramOSSpecific ( const std::string & *path* ) [static], [private]

Create a child process and execute the program with the given path considering the specific Operating system.

##### Parameters

|             |                                                |
|-------------|------------------------------------------------|
| <i>path</i> | The path to the program which will be executed |
|-------------|------------------------------------------------|

Referenced by executeProgramAndVerifyPath().

7.128.2.4 std::string OperatingSystem::getEnvironmentVariable ( const std::string & *name* ) [static]

Get the value of the environment variable having the given name.

**Parameters**

|             |                                      |
|-------------|--------------------------------------|
| <i>name</i> | The name of the environment variable |
|-------------|--------------------------------------|

Definition at line 14 of file OperatingSystem.cpp.

Referenced by multiscale::ConsolePrinter::terminalSupportsColour().

### 7.128.3 Member Data Documentation

7.128.3.1 `const std::string OperatingSystem::ERR_EXECUTE_PROGRAM = "The process executing the program located at the following path could not be created: " [static], [private]`

Definition at line 114 of file OperatingSystem.hpp.

7.128.3.2 `const std::string OperatingSystem::ERR_INVALID_PROGRAM_PATH = "The process was not created because the provided program path is invalid: " [static], [private]`

Definition at line 115 of file OperatingSystem.hpp.

Referenced by executeProgramAndVerifyPath().

7.128.3.3 `const unsigned int OperatingSystem::TIMEOUT_MAX_NR_SECONDS = 100 [static], [private]`

Definition at line 117 of file OperatingSystem.hpp.

7.128.3.4 `const unsigned int OperatingSystem::TIMEOUT_NR_SECONDS = 1 [static], [private]`

Definition at line 118 of file OperatingSystem.hpp.

The documentation for this class was generated from the following files:

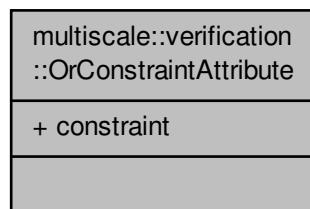
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[OperatingSystem.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[OperatingSystem.cpp](#)

## 7.129 multiscale::verification::OrConstraintAttribute Class Reference

Class for representing an "or" constraint attribute.

```
#include <OrConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::OrConstraintAttribute:



## Public Attributes

- [ConstraintAttributeType constraint](#)

### 7.129.1 Detailed Description

Class for representing an "or" constraint attribute.

Definition at line 14 of file [OrConstraintAttribute.hpp](#).

### 7.129.2 Member Data Documentation

#### 7.129.2.1 ConstraintAttributeType multiscale::verification::OrConstraintAttribute::constraint

The constraint following the "or" operator

Definition at line 18 of file [OrConstraintAttribute.hpp](#).

Referenced by [multiscale::verification::ConstraintVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

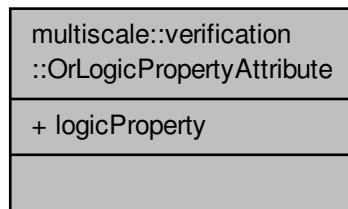
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrConstraintAttribute.hpp](#)

## 7.130 multiscale::verification::OrLogicPropertyAttribute Class Reference

Class for representing an "or" logic property attribute.

```
#include <OrLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::OrLogicPropertyAttribute:



## Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

### 7.130.1 Detailed Description

Class for representing an "or" logic property attribute.

Definition at line 14 of file [OrLogicPropertyAttribute.hpp](#).

## 7.130.2 Member Data Documentation

### 7.130.2.1 LogicPropertyAttributeType multiscale::verification::OrLogicPropertyAttribute::logicProperty

The logical property following the "or" operator

Definition at line 18 of file OrLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::operator()().

The documentation for this class was generated from the following file:

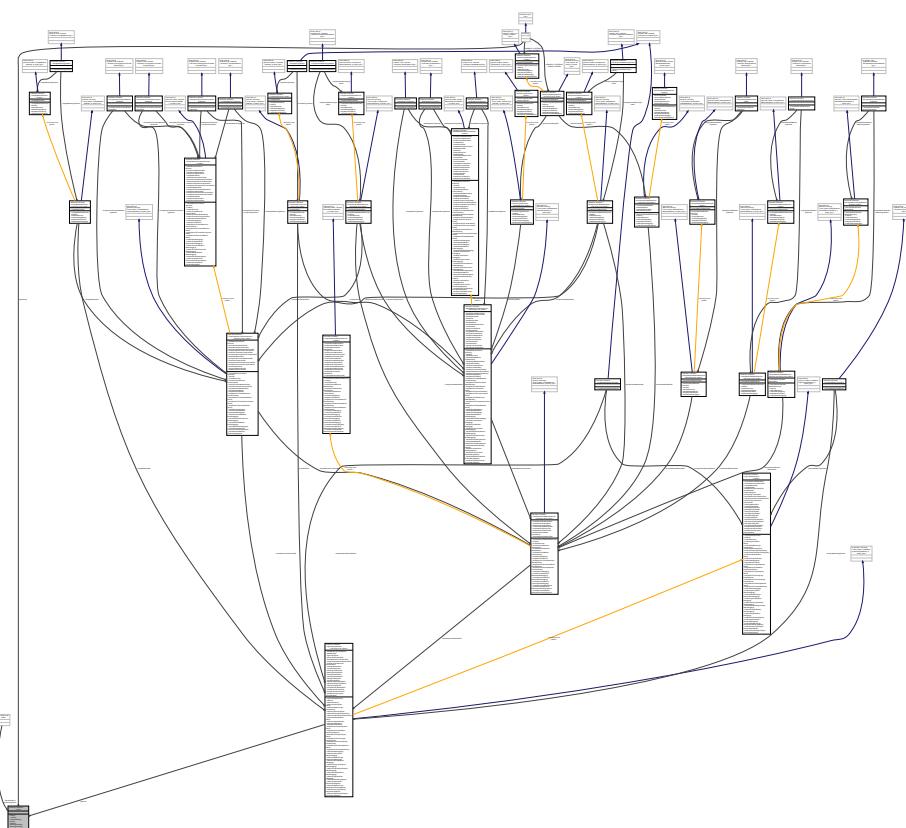
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrLogicPropertyAttribute.hpp

## 7.131 multiscale::verification::Parser Class Reference

Class used for parsing (P)BLSTL logical queries.

```
#include <Parser.hpp>
```

Collaboration diagram for multiscale::verification::Parser:



## Public Member Functions

- `Parser (const std::string &logicalQuery)`
- `~Parser ()`
- `void setLogicalQuery (const std::string &logicalQuery)`  
*Set the value of the logical query.*
- `bool parse (AbstractSyntaxTree &parseResult)`

*Parse the logical query.*

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- bool [parseLogicalQuery \(AbstractSyntaxTree &parseResult\)](#)  
*Parse the logical query and wrap the ProbabilisticLogicProperty into an [AbstractSyntaxTree](#) instance.*
- bool [parseLogicalQuery \(ProbabilisticLogicPropertyAttribute &parseResult\)](#)  
*Parse the logical query and construct the abstract syntax tree.*
- void [checkIfErrorCase \(bool isSuccessfulParse\)](#)  
*Check if an error case was encountered.*
- bool [isStringParsedCompletely \(\)](#)  
*Check if the string was parsed completely.*

## Private Attributes

- std::string [logicalQuery](#)
- std::string::const\_iterator [logicalQueryIterator](#)
- std::string::const\_iterator [logicalQueryEnd](#)
- [LogicPropertyGrammar](#)  
 $<$  std::string::const\_iterator  $>$  grammar

### 7.131.1 Detailed Description

Class used for parsing (P)BLSTL logical queries.

Definition at line 17 of file Parser.hpp.

### 7.131.2 Constructor & Destructor Documentation

#### 7.131.2.1 Parser::Parser ( const std::string & logicalQuery )

Definition at line 11 of file Parser.cpp.

References initialise(), and logicalQuery.

#### 7.131.2.2 Parser::~Parser ( )

Definition at line 17 of file Parser.cpp.

### 7.131.3 Member Function Documentation

#### 7.131.3.1 void Parser::checkIfErrorCase ( bool isSuccessfulParse ) [private]

Check if an error case was encountered.

Parameters

---

|                                    |                                 |
|------------------------------------|---------------------------------|
| <code>isSuccessful</code><br>Parse | The parse was successful or not |
|------------------------------------|---------------------------------|

Definition at line 82 of file Parser.cpp.

References `isStringParsedCompletely()`, `logicalQueryEnd`, and `logicalQueryIterator`.

Referenced by `parseLogicalQuery()`.

#### 7.131.3.2 void Parser::initialise( ) [private]

Initialisation function.

Definition at line 57 of file Parser.cpp.

References `logicalQuery`, `logicalQueryEnd`, and `logicalQueryIterator`.

Referenced by `Parser()`, and `setLogicalQuery()`.

#### 7.131.3.3 bool Parser::isStringParsedCompletely( ) [private]

Check if the string was parsed completely.

Definition at line 92 of file Parser.cpp.

References `logicalQueryEnd`, and `logicalQueryIterator`.

Referenced by `checkIfErrorCase()`.

#### 7.131.3.4 bool Parser::parse( AbstractSyntaxTree & parseResult )

Parse the logical query.

##### Parameters

|                          |                                     |
|--------------------------|-------------------------------------|
| <code>parseResult</code> | The result of the parsing procedure |
|--------------------------|-------------------------------------|

Definition at line 25 of file Parser.cpp.

References `multiscale::verification::ParserGrammarExtraInputException::getErrorMessage()`, `multiscale::verification::ParserGrammarUnparseableInputException::getErrorMessage()`, `multiscale::verification::ParserGrammarProbabilityException::getErrorMessage()`, `multiscale::verification::ParserGrammarUnexpectedTokenException::getErrorMessage()`, `multiscale::verification::ParserGrammarProbabilityException::getExpectedToken()`, `multiscale::verification::ParserGrammarUnexpectedTokenException::getExpectedToken()`, `multiscale::verification::ParserGrammarExceptionHandler::handleExtraInputException()`, `multiscale::verification::ParserGrammarExceptionHandler::handleProbabilityException()`, `multiscale::verification::ParserGrammarExceptionHandler::handleUnexpectedTokenException()`, `multiscale::verification::ParserGrammarExceptionHandler::handleUnparseableInputException()`, `logicalQuery`, `logicalQueryEnd`, `logicalQueryIterator`, and `parseLogicalQuery()`.

Referenced by `analysePatterns()`, `multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree()`, `multiscale::verification::ModelCheckingManager::isValidLogicProperty()`, `main()`, `multiscaletest::verification::parseInputString()`, `printParsingResult()`, and `multiscaletest::TraceEvaluationTest::RunTest()`.

#### 7.131.3.5 bool Parser::parseLogicalQuery( AbstractSyntaxTree & parseResult ) [private]

Parse the logical query and wrap the ProbabilisticLogicProperty into an `AbstractSyntaxTree` instance.

##### Parameters

|                          |                                     |
|--------------------------|-------------------------------------|
| <code>parseResult</code> | The result of the parsing procedure |
|--------------------------|-------------------------------------|

Definition at line 62 of file Parser.cpp.

References multiscale::verification::AbstractSyntaxTree::initialiseTree().

Referenced by parse().

#### 7.131.3.6 `bool Parser::parseLogicalQuery ( ProbabilisticLogicPropertyAttribute & parseResult ) [private]`

Parse the logical query and construct the abstract syntax tree.

##### Parameters

|                          |                                     |
|--------------------------|-------------------------------------|
| <code>parseResult</code> | The result of the parsing procedure |
|--------------------------|-------------------------------------|

Definition at line 72 of file Parser.cpp.

References checkIfErrorCase(), grammar, logicalQueryEnd, and logicalQueryIterator.

#### 7.131.3.7 `void Parser::setLogicalQuery ( const std::string & logicalQuery )`

Set the value of the logical query.

Definition at line 19 of file Parser.cpp.

References initialise(), and logicalQuery.

Referenced by multiscale::verification::ModelCheckingManager::isValidLogicProperty(), and printQueries().

### 7.131.4 Member Data Documentation

#### 7.131.4.1 `LogicPropertyGrammar<std::string::const_iterator> multiscale::verification::Parser::grammar [private]`

The grammar used for parsing logic properties

Definition at line 27 of file Parser.hpp.

Referenced by parseLogicalQuery().

#### 7.131.4.2 `std::string multiscale::verification::Parser::logicalQuery [private]`

The logical query to be parsed

Definition at line 21 of file Parser.hpp.

Referenced by initialise(), parse(), Parser(), and setLogicalQuery().

#### 7.131.4.3 `std::string::const_iterator multiscale::verification::Parser::logicalQueryEnd [private]`

Iterator pointing at the end of the logical query

Definition at line 24 of file Parser.hpp.

Referenced by checkIfErrorCase(), initialise(), isStringParsedCompletely(), parse(), and parseLogicalQuery().

#### 7.131.4.4 `std::string::const_iterator multiscale::verification::Parser::logicalQueryIterator [private]`

Iterator of the logical query

Definition at line 23 of file Parser.hpp.

Referenced by checkIfErrorCase(), initialise(), isStringParsedCompletely(), parse(), and parseLogicalQuery().

The documentation for this class was generated from the following files:

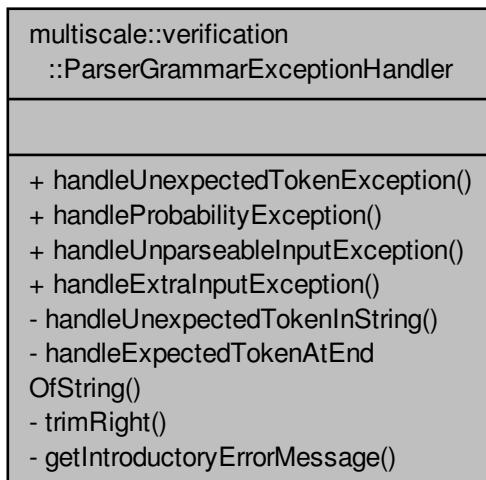
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[Parser.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/[Parser.cpp](#)

## 7.132 multiscale::verification::ParserGrammarExceptionHandler Class Reference

Class for handling parser grammar exceptions.

```
#include <ParserGrammarExceptionHandler.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarExceptionHandler:



### Static Public Member Functions

- static void [handleUnexpectedTokenException](#) (const std::string &initialString, const std::string &errorString, const std::string &expectedToken)
 

*Handle the exception when an unexpected token was encountered.*
- static void [handleProbabilityException](#) (const std::string &initialString, const std::string &errorString, const std::string &expectedToken)
 

*Handle the exception when an invalid probability was encountered.*
- static void [handleUnparseableInputException](#) (const std::string &initialString, const std::string &errorString)
 

*Handle the exception when wrong input is provided.*
- static void [handleExtraInputException](#) (const std::string &initialString, const std::string &extraInput)
 

*Handle the exception when extra input is provided.*

## Static Private Member Functions

- static std::string `handleUnexpectedTokenInString` (const std::string &initialString, const std::string &errorString, const std::string &expectedToken)
 

*Handle the case where an unexpected token was found in the std::string.*
- static std::string `handleExpectedTokenAtEndOfString` (const std::string &initialString, const std::string &expectedToken)
 

*Handle the case where an expected token was not encountered at the end of the std::string.*
- static std::string `trimRight` (const std::string &inputString)
 

*Remove the trailing "new line" characters from the end of the string.*
- static std::string `getIntroductoryErrorMessage` ()
 

*Return the generic introductory error message.*

### 7.132.1 Detailed Description

Class for handling parser grammar exceptions.

Definition at line 16 of file ParserGrammarExceptionHandler.hpp.

### 7.132.2 Member Function Documentation

#### 7.132.2.1 std::string ParserGrammarExceptionHandler::getIntroductoryErrorMessage( ) [static], [private]

Return the generic introductory error message.

Definition at line 125 of file ParserGrammarExceptionHandler.cpp.

Referenced by `handleExpectedTokenAtEndOfString()`, `handleExtraInputException()`, `handleProbabilityException()`, `handleUnexpectedTokenInString()`, and `handleUnparseableInputException()`.

#### 7.132.2.2 std::string ParserGrammarExceptionHandler::handleExpectedTokenAtEndOfString( const std::string & initialString, const std::string & expectedToken ) [static], [private]

Handle the case where an expected token was not encountered at the end of the std::string.

##### Parameters

|                            |                                                |
|----------------------------|------------------------------------------------|
| <code>initialString</code> | The initial std::string                        |
| <code>expectedToken</code> | The token which should replace the error token |

Definition at line 107 of file ParserGrammarExceptionHandler.cpp.

References `getIntroductoryErrorMessage()`.

Referenced by `handleUnexpectedTokenException()`.

#### 7.132.2.3 void ParserGrammarExceptionHandler::handleExtraInputException( const std::string & initialString, const std::string & extraInput ) [static]

Handle the exception when extra input is provided.

##### Parameters

|                            |                         |
|----------------------------|-------------------------|
| <code>initialString</code> | The initial std::string |
|----------------------------|-------------------------|

|                   |             |
|-------------------|-------------|
| <i>extraInput</i> | Extra input |
|-------------------|-------------|

Definition at line 65 of file ParserGrammarExceptionHandler.cpp.

References getIntroductoryErrorMessage(), MS\_throw, and trimRight().

Referenced by multiscale::verification::Parser::parse().

#### 7.132.2.4 void ParserGrammarExceptionHandler::handleProbabilityException ( const std::string & *initialString*, const std::string & *errorString*, const std::string & *expectedToken* ) [static]

Handle the exception when an invalid probability was encountered.

##### Parameters

|                      |                                                                                  |
|----------------------|----------------------------------------------------------------------------------|
| <i>initialString</i> | The initial std::string                                                          |
| <i>errorString</i>   | A substd::string of the initial std::string which starts from the error position |
| <i>expectedToken</i> | The token which should replace the error token                                   |

Definition at line 27 of file ParserGrammarExceptionHandler.cpp.

References getIntroductoryErrorMessage(), and MS\_throw.

Referenced by multiscale::verification::Parser::parse().

#### 7.132.2.5 void ParserGrammarExceptionHandler::handleUnexpectedTokenException ( const std::string & *initialString*, const std::string & *errorString*, const std::string & *expectedToken* ) [static]

Handle the exception when an unexpected token was encountered.

##### Parameters

|                      |                                                                                  |
|----------------------|----------------------------------------------------------------------------------|
| <i>initialString</i> | The initial std::string                                                          |
| <i>errorString</i>   | A substd::string of the initial std::string which starts from the error position |
| <i>expectedToken</i> | The token which should replace the error token                                   |

Definition at line 13 of file ParserGrammarExceptionHandler.cpp.

References handleExpectedTokenAtEndOfString(), handleUnexpectedTokenInString(), and MS\_throw.

Referenced by multiscale::verification::Parser::parse().

#### 7.132.2.6 std::string ParserGrammarExceptionHandler::handleUnexpectedTokenInString ( const std::string & *initialString*, const std::string & *errorString*, const std::string & *expectedToken* ) [static], [private]

Handle the case where an unexpected token was found in the std::string.

##### Parameters

|                      |                                                                                  |
|----------------------|----------------------------------------------------------------------------------|
| <i>initialString</i> | The initial std::string                                                          |
| <i>errorString</i>   | A substd::string of the initial std::string which starts from the error position |
| <i>expectedToken</i> | The token which should replace the error token                                   |

Definition at line 84 of file ParserGrammarExceptionHandler.cpp.

References getIntroductoryErrorMessage().

Referenced by handleUnexpectedTokenException().

#### 7.132.2.7 void ParserGrammarExceptionHandler::handleUnparseableInputException ( const std::string & *initialString*, const std::string & *errorString* ) [static]

Handle the exception when wrong input is provided.

## Parameters

|                      |                         |
|----------------------|-------------------------|
| <i>initialString</i> | The initial std::string |
| <i>errorString</i>   | Error std::string       |

Definition at line 47 of file ParserGrammarExceptionHandler.cpp.

References getIntroductoryErrorMessage(), MS\_throw, and trimRight().

Referenced by multiscale::verification::Parser::parse().

**7.132.2.8 std::string ParserGrammarExceptionHandler::trimRight ( const std::string & *inputString* ) [static], [private]**

Remove the trailing "new line" characters from the end of the string.

## Parameters

|                    |                        |
|--------------------|------------------------|
| <i>inputString</i> | The given input string |
|--------------------|------------------------|

Definition at line 121 of file ParserGrammarExceptionHandler.cpp.

References multiscale::StringManipulator::trimRight().

Referenced by handleExtraInputException(), and handleUnparseableInputException().

The documentation for this class was generated from the following files:

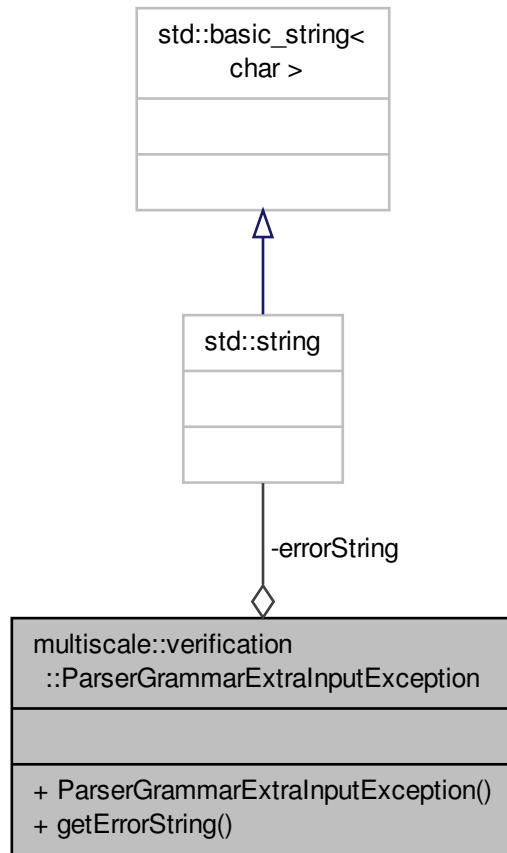
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ParserGrammarExceptionHandler.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/[ParserGrammarExceptionHandler.cpp](#)

## 7.133 multiscale::verification::ParserGrammarExtraInputException Class Reference

Class for representing "extra input" exceptions in the parsing process.

```
#include <ParserGrammarExtraInputException.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarExtraInputException:



## Public Member Functions

- `ParserGrammarExtraInputException (const std::string &errorString)`
- `std::string getErrorString () const`  
*Get the error std::string.*

## Private Attributes

- `std::string errorString`

### 7.133.1 Detailed Description

Class for representing "extra input" exceptions in the parsing process.

Definition at line 14 of file `ParserGrammarExtraInputException.hpp`.

### 7.133.2 Constructor & Destructor Documentation

7.133.2.1 `multiscale::verification::ParserGrammarExtraInputException::ParserGrammarExtraInputException ( const std::string & errorString ) [inline]`

Definition at line 23 of file ParserGrammarExtraInputException.hpp.

References errorString.

### 7.133.3 Member Function Documentation

7.133.3.1 `std::string multiscale::verification::ParserGrammarExtraInputException::getErrorResponse( ) const [inline]`

Get the error std::string.

Definition at line 28 of file ParserGrammarExtraInputException.hpp.

References errorString.

Referenced by multiscale::verification::Parser::parse().

### 7.133.4 Member Data Documentation

7.133.4.1 `std::string multiscale::verification::ParserGrammarExtraInputException::errorString [private]`

The substring from the original std::string starting with the index of the error token

Definition at line 18 of file ParserGrammarExtraInputException.hpp.

Referenced by getErrorResponse(), and ParserGrammarExtraInputException().

The documentation for this class was generated from the following file:

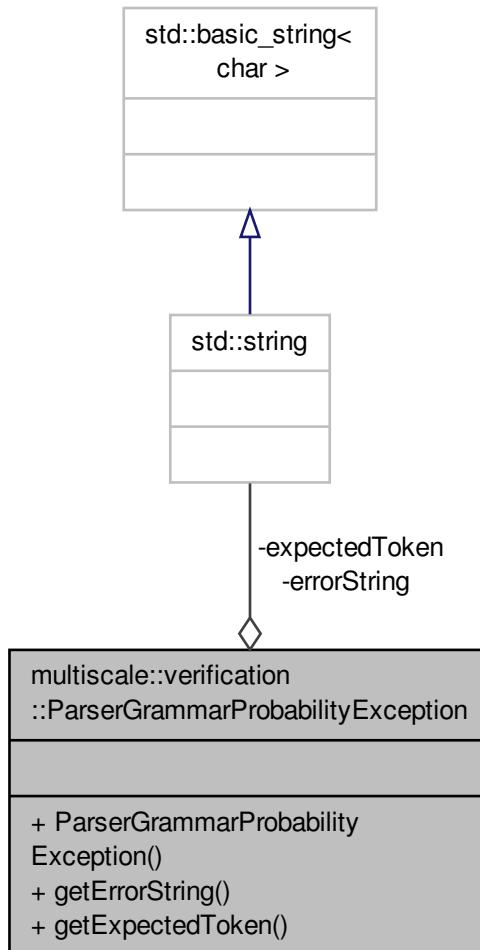
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarExtraInputException.hpp

## 7.134 multiscale::verification::ParserGrammarProbabilityException Class Reference

Class for representing "probability" exceptions in the parsing process.

```
#include <ParserGrammarProbabilityException.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarProbabilityException:



## Public Member Functions

- `ParserGrammarProbabilityException (const std::string &expectedToken, const std::string &errorString)`
- `std::string getErrorString () const`  
*Get the error string.*
- `std::string getExpectedToken () const`  
*Get the expected token.*

## Private Attributes

- `std::string expectedToken`
- `std::string errorString`

### 7.134.1 Detailed Description

Class for representing "probability" exceptions in the parsing process.

Definition at line 12 of file ParserGrammarProbabilityException.hpp.

### 7.134.2 Constructor & Destructor Documentation

7.134.2.1 `multiscale::verification::ParserGrammarProbabilityException::ParserGrammarProbabilityException ( const std::string & expectedToken, const std::string & errorString ) [inline]`

Definition at line 22 of file ParserGrammarProbabilityException.hpp.

References errorString, and expectedToken.

### 7.134.3 Member Function Documentation

7.134.3.1 `std::string multiscale::verification::ParserGrammarProbabilityException::getErrorString ( ) const [inline]`

Get the error string.

Definition at line 29 of file ParserGrammarProbabilityException.hpp.

References errorString.

Referenced by multiscale::verification::Parser::parse().

7.134.3.2 `std::string multiscale::verification::ParserGrammarProbabilityException::getExpectedToken ( ) const [inline]`

Get the expected token.

Definition at line 34 of file ParserGrammarProbabilityException.hpp.

References expectedToken.

Referenced by multiscale::verification::Parser::parse().

### 7.134.4 Member Data Documentation

7.134.4.1 `std::string multiscale::verification::ParserGrammarProbabilityException::errorString [private]`

The substring from the original string starting with the index of the error token

Definition at line 17 of file ParserGrammarProbabilityException.hpp.

Referenced by getErrorString(), and ParserGrammarProbabilityException().

7.134.4.2 `std::string multiscale::verification::ParserGrammarProbabilityException::expectedToken [private]`

The token which was expected and was not found during parsing

Definition at line 16 of file ParserGrammarProbabilityException.hpp.

Referenced by getExpectedToken(), and ParserGrammarProbabilityException().

The documentation for this class was generated from the following file:

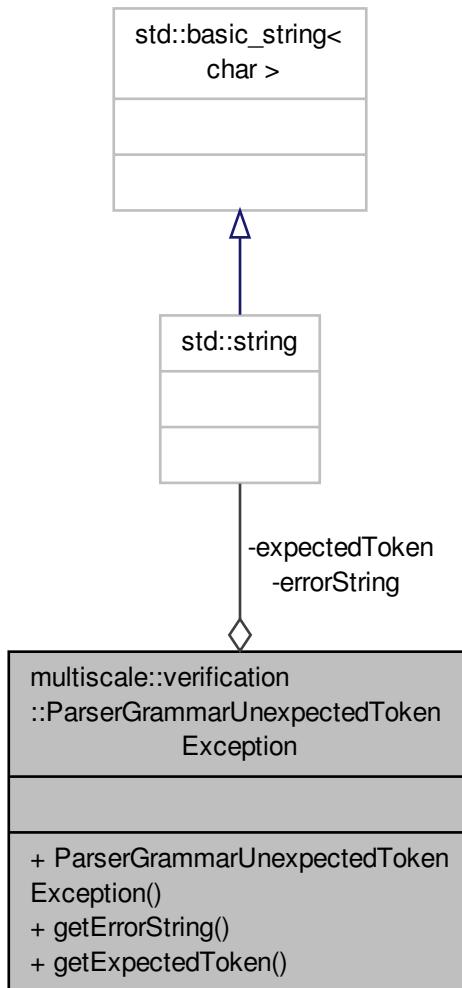
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarProbabilityException.hpp

## 7.135 multiscale::verification::ParserGrammarUnexpectedTokenException Class Reference

Class for representing "unexpected token" exceptions in the parsing process.

```
#include <ParserGrammarUnexpectedTokenException.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarUnexpectedTokenException:



### Public Member Functions

- `ParserGrammarUnexpectedTokenException` (const std::string &`expectedToken`, const std::string &`errorString`)
- std::string `getErrorResponse()` const
  - Get the error string.*
- std::string `getExpectedToken()` const
  - Get the expected token.*

## Private Attributes

- std::string `expectedToken`
- std::string `errorString`

### 7.135.1 Detailed Description

Class for representing "unexpected token" exceptions in the parsing process.

Definition at line 12 of file ParserGrammarUnexpectedTokenException.hpp.

### 7.135.2 Constructor & Destructor Documentation

7.135.2.1 `multiscale::verification::ParserGrammarUnexpectedTokenException::ParserGrammarUnexpectedTokenException ( const std::string & expectedToken, const std::string & errorString ) [inline]`

Definition at line 22 of file ParserGrammarUnexpectedTokenException.hpp.

References `errorString`, and `expectedToken`.

### 7.135.3 Member Function Documentation

7.135.3.1 `std::string multiscale::verification::ParserGrammarUnexpectedTokenException::getErrorString ( ) const [inline]`

Get the error string.

Definition at line 29 of file ParserGrammarUnexpectedTokenException.hpp.

References `errorString`.

Referenced by `multiscale::verification::Parser::parse()`.

7.135.3.2 `std::string multiscale::verification::ParserGrammarUnexpectedTokenException::getExpectedToken ( ) const [inline]`

Get the expected token.

Definition at line 34 of file ParserGrammarUnexpectedTokenException.hpp.

References `expectedToken`.

Referenced by `multiscale::verification::Parser::parse()`.

### 7.135.4 Member Data Documentation

7.135.4.1 `std::string multiscale::verification::ParserGrammarUnexpectedTokenException::errorString [private]`

The substring from the original string starting with the index of the error token

Definition at line 17 of file ParserGrammarUnexpectedTokenException.hpp.

Referenced by `getErrorString()`, and `ParserGrammarUnexpectedTokenException()`.

7.135.4.2 `std::string multiscale::verification::ParserGrammarUnexpectedTokenException::expectedToken [private]`

The token which was expected and was not found during parsing

Definition at line 16 of file ParserGrammarUnexpectedTokenException.hpp.

Referenced by `getExpectedToken()`, and `ParserGrammarUnexpectedTokenException()`.

The documentation for this class was generated from the following file:

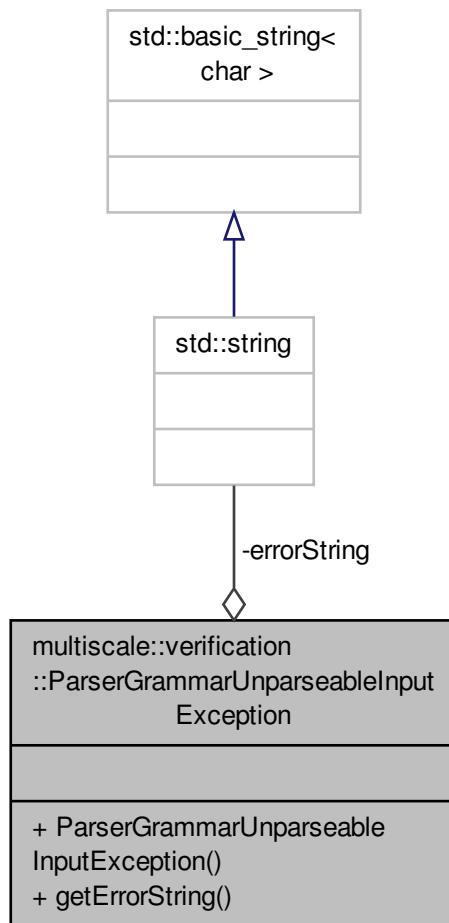
- `/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarUnexpectedTokenException.hpp`

## 7.136 multiscale::verification::ParserGrammarUnparseableInputException Class Reference

Class for representing "unparseable input" exceptions in the parsing process.

```
#include <ParserGrammarUnparseableInputException.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarUnparseableInputException:



### Public Member Functions

- `ParserGrammarUnparseableInputException (const std::string &errorString)`

- std::string [getErrorResponse \(\) const](#)  
*Get the error string.*

## Private Attributes

- std::string [errorString](#)

### 7.136.1 Detailed Description

Class for representing "unparseable input" exceptions in the parsing process.

Definition at line 14 of file ParserGrammarUnparseableInputException.hpp.

### 7.136.2 Constructor & Destructor Documentation

- 7.136.2.1 multiscale::verification::ParserGrammarUnparseableInputException::ParserGrammarUnparseableInputException (const std::string & [errorString](#)) [inline]

Definition at line 23 of file ParserGrammarUnparseableInputException.hpp.

References [errorString](#).

### 7.136.3 Member Function Documentation

- 7.136.3.1 std::string multiscale::verification::ParserGrammarUnparseableInputException::getErrorString ( ) const [inline]

Get the error string.

Definition at line 28 of file ParserGrammarUnparseableInputException.hpp.

References [errorString](#).

Referenced by multiscale::verification::Parser::parse().

### 7.136.4 Member Data Documentation

- 7.136.4.1 std::string multiscale::verification::ParserGrammarUnparseableInputException::errorString [private]

The substring from the original std::string starting with the index of the error token

Definition at line 18 of file ParserGrammarUnparseableInputException.hpp.

Referenced by [getErrorString\(\)](#), and [ParserGrammarUnparseableInputException\(\)](#).

The documentation for this class was generated from the following file:

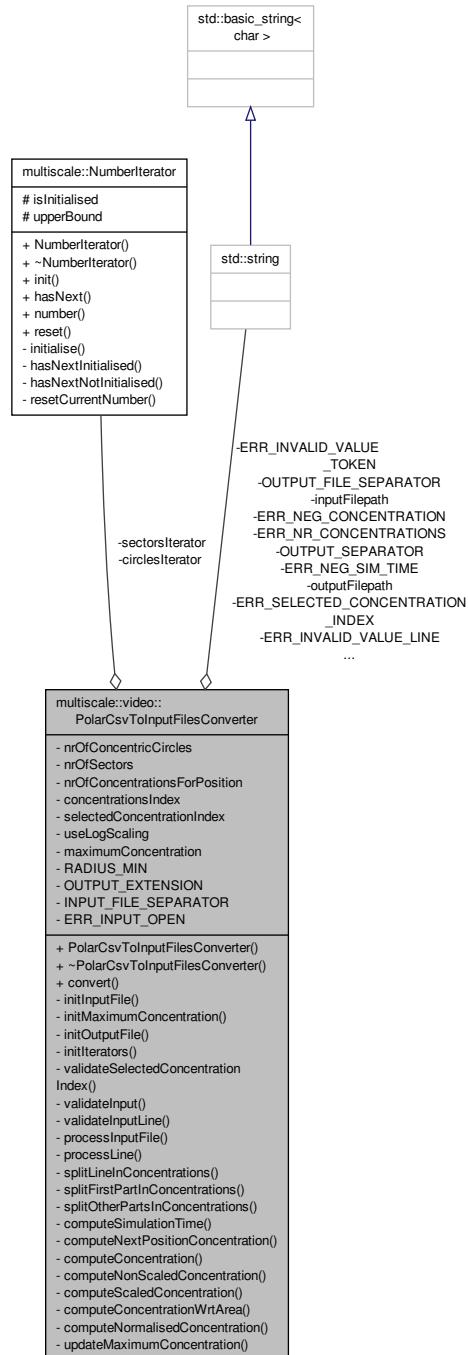
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ParserGrammarUnparseableInputException.hpp](#)

## 7.137 multiscale::video::PolarCsvToInputFilesConverter Class Reference

Csv file to input file converter considering polar coordinates.

```
#include <PolarCsvToInputFilesConverter.hpp>
```

Collaboration diagram for multiscale::video::PolarCsvToInputFilesConverter:



## Public Member Functions

- `PolarCsvToInputFilesConverter` (const `std::string &inputFilepath`, const `std::string &outputFilepath`, unsigned int `nrOfConcentricCircles`, unsigned int `nrOfSectors`, unsigned int `nrOfConcentrationsForPosition`, unsigned int `selectedConcentrationIndex`, bool `useLogScaling`, `NumberIteratorType` `numberIteratorType`)
- `~PolarCsvToInputFilesConverter ()`
- `void convert ()`

*Start the conversion.*

## Private Member Functions

- void `initInputFile` (std::ifstream &fin)
 

*Initialise the input file stream over the given input file.*
- void `initMaximumConcentration` (std::ifstream &fin)
 

*Compute the value of member maximum concentration.*
- void `initOutputFile` (std::ofstream &fout, unsigned int index, double &simulationTime)
 

*Initialise the output file with the given index and simulation time.*
- void `initIterators` (const NumberIteratorType &numberIteratorType)
 

*Initialise the iterators considering the given number iterator type.*
- void `validateSelectedConcentrationIndex` ()
 

*Validate the selected concentration index in case of more than one concentration for each position.*
- void `validateInput` (std::ifstream &fin)
 

*Validate the input.*
- void `validateInputLine` (const std::string &line, unsigned int lineNumber)
 

*Validate the provided line identified by a line number.*
- void `processInputFile` (std::ifstream &fin)
 

*Process the input file.*
- void `processLine` (const std::string &line, unsigned int outputIndex)
 

*Process the provided line.*
- std::vector< double > `splitLineInConcentrations` (const std::string &line, double &simulationTime)
 

*Split the line in concentrations.*
- void `splitFirstPartInConcentrations` (std::vector< double > &concentrations, const std::vector< std::string > &tokens, unsigned int circleIndex)
 

*Split first part of the line (i.e. part representing the origin) into concentrations.*
- void `splitOtherPartsInConcentrations` (std::vector< double > &concentrations, const std::vector< std::string > &tokens, unsigned int circleIndex)
 

*Split other parts of the line (i.e. non-first part) into concentrations.*
- double `computeSimulationTime` (const std::string &token)
 

*Compute the simulation time from the given token and check if it is valid.*
- double `computeNextPositionConcentration` (unsigned int circleIndex, int concentrationIndex, const std::vector< std::string > &tokens)
 

*Compute the concentration for the next position.*
- double `computeConcentration` (const std::string &concentration, int circleIndex)
 

*Compute the concentration from the given string.*
- double `computeNonScaledConcentration` (const std::string &concentration, int circleIndex)
 

*Compute the non-scaled concentration from the given string.*
- double `computeScaledConcentration` (const std::string &concentration, int circleIndex)
 

*Compute the scaled concentration from the given string.*
- double `computeConcentrationWrtArea` (double amount, int circleIndex)
 

*Compute the concentration wrt. the area of the annular sector.*
- double `computeNormalisedConcentration` (double concentration, int circleIndex)
 

*Normalise the concentration considering the index of the current concentric circle.*
- void `updateMaximumConcentration` (const std::string &line, double &maximumConcentration)
 

*Update the maximum concentration if the values from the given line are greater than it.*

## Private Attributes

- std::string `inputfilepath`
- std::string `outputfilepath`
- unsigned int `nrOfConcentricCircles`
- unsigned int `nrOfSectors`
- unsigned int `nrOfConcentrationsForPosition`
- unsigned int `concentrationsIndex`
- unsigned int `selectedConcentrationIndex`
- bool `useLogScaling`
- double `maximumConcentration`
- NumberIterator \* `circlesIterator`
- NumberIterator \* `sectorsIterator`

## Static Private Attributes

- static const int `RADIUS_MIN` = 1
- static const std::string `OUTPUT_EXTENSION` = ".in"
- static const std::string `OUTPUT_SEPARATOR` = " "
- static const std::string `OUTPUT_FILE_SEPARATOR` = "\_"
- static const std::string `INPUT_FILE_SEPARATOR` = ","
- static const std::string `ERR_NEG_CONCENTRATION` = "All concentrations must be non-negative."
- static const std::string `ERR_SELECTED_CONCENTRATION_INDEX` = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations."
- static const std::string `ERR_NR_CONCENTRATIONS` = "The number of concentrations in the input file does not match the values of the input parameters height and width."
- static const std::string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const std::string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const std::string `ERR_INVALID_VALUE_LINE` = "Invalid value on line: "
- static const std::string `ERR_INVALID_VALUE_TOKEN` = ", value: "

### 7.137.1 Detailed Description

Csv file to input file converter considering polar coordinates.

Definition at line 16 of file PolarCsvToInputFilesConverter.hpp.

### 7.137.2 Constructor & Destructor Documentation

7.137.2.1 `PolarCsvToInputFilesConverter::PolarCsvToInputFilesConverter ( const std::string & inputfilepath, const std::string & outputfilepath, unsigned int nrOfConcentricCircles, unsigned int nrOfSectors, unsigned int nrOfConcentrationsForPosition, unsigned int selectedConcentrationIndex, bool useLogScaling, NumberIteratorType numberIteratorType )`

Definition at line 20 of file PolarCsvToInputFilesConverter.cpp.

7.137.2.2 `PolarCsvToInputFilesConverter::~PolarCsvToInputFilesConverter ( )`

Definition at line 44 of file PolarCsvToInputFilesConverter.cpp.

### 7.137.3 Member Function Documentation

7.137.3.1 double PolarCsvToInputFilesConverter::computeConcentration ( const std::string & *concentration*, int *circleIndex* ) [private]

Compute the concentration from the given string.

Compute the concentration from the given string considering the index of the current concentric circle

#### Parameters

|                      |                                       |
|----------------------|---------------------------------------|
| <i>concentration</i> | String representing the concentration |
| <i>circleIndex</i>   | Index of the concentric circle        |

Definition at line 309 of file PolarCsvToInputFilesConverter.cpp.

7.137.3.2 double PolarCsvToInputFilesConverter::computeConcentrationWrtArea ( double *amount*, int *circleIndex* ) [private]

Compute the concentration wrt. the area of the annular sector.

#### Parameters

|                    |                                                                         |
|--------------------|-------------------------------------------------------------------------|
| <i>amount</i>      | Amount in annular sector                                                |
| <i>circleIndex</i> | Index of the concentric circle which will be used to determine the area |

Definition at line 338 of file PolarCsvToInputFilesConverter.cpp.

7.137.3.3 double PolarCsvToInputFilesConverter::computeNextPositionConcentration ( unsigned int *circleIndex*, int *concentrationIndex*, const std::vector< std::string > & *tokens* ) [private]

Compute the concentration for the next position.

#### Parameters

|                           |                                                              |
|---------------------------|--------------------------------------------------------------|
| <i>circleIndex</i>        | Index of the current concentric circle                       |
| <i>concentrationIndex</i> | Index of the current concentration from the vector of tokens |
| <i>tokens</i>             | Vector of tokens                                             |

Definition at line 280 of file PolarCsvToInputFilesConverter.cpp.

7.137.3.4 double PolarCsvToInputFilesConverter::computeNonScaledConcentration ( const std::string & *concentration*, int *circleIndex* ) [private]

Compute the non-scaled concentration from the given string.

Compute the non-scaled concentration from the given string considering the index of the current concentric circle

#### Parameters

|                      |                                       |
|----------------------|---------------------------------------|
| <i>concentration</i> | String representing the concentration |
| <i>circleIndex</i>   | Index of the concentric circle        |

Definition at line 316 of file PolarCsvToInputFilesConverter.cpp.

7.137.3.5 double PolarCsvToInputFilesConverter::computeNormalisedConcentration ( double *concentration*, int *circleIndex* ) [private]

Normalise the concentration considering the index of the current concentric circle.

Normalise the concentration considering the index of the current concentric circle by dividing it to the maximum concentration

#### Parameters

|                      |                                |
|----------------------|--------------------------------|
| <i>concentration</i> | The concentration              |
| <i>circleIndex</i>   | Index of the concentric circle |

Definition at line 342 of file PolarCsvToInputFilesConverter.cpp.

**7.137.3.6 double PolarCsvToInputFilesConverter::computeScaledConcentration ( const std::string & *concentration*, int *circleIndex* ) [private]**

Compute the scaled concentration from the given string.

Compute the scaled concentration from the given string considering the index of the current concentric circle by applying a logit transformation to it

#### Parameters

|                      |                                       |
|----------------------|---------------------------------------|
| <i>concentration</i> | String representing the concentration |
| <i>circleIndex</i>   | Index of the concentric circle        |

Definition at line 323 of file PolarCsvToInputFilesConverter.cpp.

**7.137.3.7 double PolarCsvToInputFilesConverter::computeSimulationTime ( const std::string & *token* ) [private]**

Compute the simulation time from the given token and check if it is valid.

#### Parameters

|              |                     |
|--------------|---------------------|
| <i>token</i> | Token (std::string) |
|--------------|---------------------|

Definition at line 270 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.137.3.8 void PolarCsvToInputFilesConverter::convert ( )**

Start the conversion.

Definition at line 49 of file PolarCsvToInputFilesConverter.cpp.

Referenced by main().

**7.137.3.9 void PolarCsvToInputFilesConverter::initInputModule ( std::ifstream & *fin* ) [private]**

Initialise the input file stream over the given input file.

#### Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 62 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.137.3.10 void PolarCsvToInputFilesConverter::initIterators ( const NumberIteratorType & *numberIteratorType* ) [private]**

Initialise the iterators considering the given number iterator type.

## Parameters

|                               |                                 |
|-------------------------------|---------------------------------|
| <i>numberIterator</i><br>Type | The type of the number iterator |
|-------------------------------|---------------------------------|

Definition at line 112 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

### 7.137.3.11 void PolarCsvToInputFilesConverter::initMaximumConcentration ( std::ifstream & *fin* ) [private]

Compute the value of member maximum concentration.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 70 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw.

### 7.137.3.12 void PolarCsvToInputFilesConverter::initOutputFile ( std::ofstream & *fout*, unsigned int *index*, double & *simulationTime* ) [private]

Initialise the output file with the given index and simulation time.

## Parameters

|                       |                          |
|-----------------------|--------------------------|
| <i>fout</i>           | Output file stream       |
| <i>index</i>          | Index of the output file |
| <i>simulationTime</i> | Simulation time          |

Definition at line 94 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

### 7.137.3.13 void PolarCsvToInputFilesConverter::processInputFile ( std::ifstream & *fin* ) [private]

Process the input file.

Read the concentrations and normalise them if it is the case.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 180 of file PolarCsvToInputFilesConverter.cpp.

### 7.137.3.14 void PolarCsvToInputFilesConverter::processLine ( const std::string & *line*, unsigned int *outputIndex* ) [private]

Process the provided line.

## Parameters

|                    |                                                 |
|--------------------|-------------------------------------------------|
| <i>line</i>        | Line                                            |
| <i>outputIndex</i> | Index integrated in the name of the output file |

Definition at line 195 of file PolarCsvToInputFilesConverter.cpp.

```
7.137.3.15 void PolarCsvToInputFilesConverter::splitFirstPartInConcentrations (std::vector< double > & concentrations,
 const std::vector< std::string > & tokens, unsigned int circleIndex) [private]
```

Split first part of the line (i.e. part representing the origin) into concentrations.

**Parameters**

|                       |                                        |
|-----------------------|----------------------------------------|
| <i>concentrations</i> | Concentrations extracted from tokens   |
| <i>tokens</i>         | Tokens representing the line           |
| <i>circleIndex</i>    | Index of the current concentric circle |

Definition at line 240 of file PolarCsvToInputFilesConverter.cpp.

**7.137.3.16** `std::vector< double > PolarCsvToInputFilesConverter::splitLineInConcentrations ( const std::string & line, double & simulationTime ) [private]`

Split the line in concentrations.

**Parameters**

|                       |                                          |
|-----------------------|------------------------------------------|
| <i>line</i>           | Line                                     |
| <i>simulationTime</i> | Simulation time associated with the line |

Definition at line 214 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

**7.137.3.17** `void PolarCsvToInputFilesConverter::splitOtherPartsInConcentrations ( std::vector< double > & concentrations, const std::vector< std::string > & tokens, unsigned int circleIndex ) [private]`

Split other parts of the line (i.e. non-first part) into concentrations.

**Parameters**

|                       |                                        |
|-----------------------|----------------------------------------|
| <i>concentrations</i> | Concentrations extracted from tokens   |
| <i>tokens</i>         | Tokens representing the line           |
| <i>circleIndex</i>    | Index of the current concentric circle |

Definition at line 254 of file PolarCsvToInputFilesConverter.cpp.

**7.137.3.18** `void PolarCsvToInputFilesConverter::updateMaximumConcentration ( const std::string & line, double & maximumConcentration ) [private]`

Update the maximum concentration if the values from the given line are greater than it.

**Parameters**

|                             |                           |
|-----------------------------|---------------------------|
| <i>line</i>                 | Line from input file      |
| <i>maximumConcentration</i> | The maximum concentration |

Definition at line 346 of file PolarCsvToInputFilesConverter.cpp.

**7.137.3.19** `void PolarCsvToInputFilesConverter::validateInput ( std::ifstream & fin ) [private]`

Validate the input.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 135 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw.

```
7.137.3.20 void PolarCsvToInputFilesConverter::validateInputLine (const std::string & line, unsigned int lineNumber)
[private]
```

Validate the provided line identified by a line number.

**Parameters**

|                   |                      |
|-------------------|----------------------|
| <i>line</i>       | Line from input file |
| <i>lineNumber</i> | Number of the line   |

Definition at line 159 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw, and multiscale::StringManipulator::split().

#### 7.137.3.21 void PolarCsvToInputFilesConverter::validateSelectedConcentrationIndex( ) [private]

Validate the selected concentration index in case of more than one concentration for each position.

Definition at line 129 of file PolarCsvToInputFilesConverter.cpp.

References MS\_throw.

### 7.137.4 Member Data Documentation

#### 7.137.4.1 NumberIterator\* multiscale::video::PolarCsvToInputFilesConverter::circlesIterator [private]

Iterator over the number of concentric circles

Definition at line 41 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.2 unsigned int multiscale::video::PolarCsvToInputFilesConverter::concentrationsIndex [private]

Index of the current concentration

Definition at line 27 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.3 const std::string PolarCsvToInputFilesConverter::ERR\_INPUT\_OPEN = "The input file could not be opened." [static], [private]

Definition at line 225 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.4 const std::string PolarCsvToInputFilesConverter::ERR\_INVALID\_VALUE\_LINE = "Invalid value on line: " [static], [private]

Definition at line 226 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.5 const std::string PolarCsvToInputFilesConverter::ERR\_INVALID\_VALUE\_TOKEN = ", value: " [static], [private]

Definition at line 227 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.6 const std::string PolarCsvToInputFilesConverter::ERR\_NEG\_CONCENTRATION = "All concentrations must be non-negative." [static], [private]

Definition at line 221 of file PolarCsvToInputFilesConverter.hpp.

#### 7.137.4.7 const std::string PolarCsvToInputFilesConverter::ERR\_NEG\_SIM\_TIME = "The simulation time must be non-negative." [static], [private]

Definition at line 224 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.8 `const std::string PolarCsvToInputFilesConverter::ERR_NR_CONCENTRATIONS = "The number of concentrations in the input file does not match the values of the input parameters height and width." [static], [private]`

Definition at line 223 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.9 `const std::string PolarCsvToInputFilesConverter::ERR_SELECTED_CONCENTRATION_INDEX = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations." [static], [private]`

Definition at line 222 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.10 `const std::string PolarCsvToInputFilesConverter::INPUT_FILE_SEPARATOR = "," [static], [private]`

Definition at line 219 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.11 `std::string multiscale::video::PolarCsvToInputFilesConverter::inputFilepath [private]`

Path to the input file

Definition at line 20 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.12 `double multiscale::video::PolarCsvToInputFilesConverter::maximumConcentration [private]`

The maximum concentration in the input file

Definition at line 39 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.13 `unsigned int multiscale::video::PolarCsvToInputFilesConverter::nrOfConcentrationsForPosition [private]`

Number of concentrations for each position

Definition at line 25 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.14 `unsigned int multiscale::video::PolarCsvToInputFilesConverter::nrOfConcentricCircles [private]`

Number of concentric circles

Definition at line 23 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.15 `unsigned int multiscale::video::PolarCsvToInputFilesConverter::nrOfSectors [private]`

Number of sectors

Definition at line 24 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.16 `const std::string PolarCsvToInputFilesConverter::OUTPUT_EXTENSION = ".in" [static], [private]`

Definition at line 216 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.17 `const std::string PolarCsvToInputFilesConverter::OUTPUT_FILE_SEPARATOR = "_" [static], [private]`

Definition at line 218 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.18 const std::string PolarCsvToInputFilesConverter::OUTPUT\_SEPARATOR = " " [static], [private]

Definition at line 217 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.19 std::string multiscale::video::PolarCsvToInputFilesConverter::outputFilepath [private]

Path to the output file

Definition at line 21 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.20 const int PolarCsvToInputFilesConverter::RADIUS\_MIN = 1 [static], [private]

Definition at line 214 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.21 NumberIterator\* multiscale::video::PolarCsvToInputFilesConverter::sectorsIterator [private]

Iterator over the number of sectors

Definition at line 42 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.22 unsigned int multiscale::video::PolarCsvToInputFilesConverter::selectedConcentrationIndex [private]

Index of the concentration A in case the number of concentrations for each position is greater than 1

finalConcentration = A / (A1 + A2 + ... + AN), where N is the number of concentrations for each position

Definition at line 29 of file PolarCsvToInputFilesConverter.hpp.

7.137.4.23 bool multiscale::video::PolarCsvToInputFilesConverter::useLogScaling [private]

Flag for using logarithmic scaling for concentrations

Definition at line 36 of file PolarCsvToInputFilesConverter.hpp.

The documentation for this class was generated from the following files:

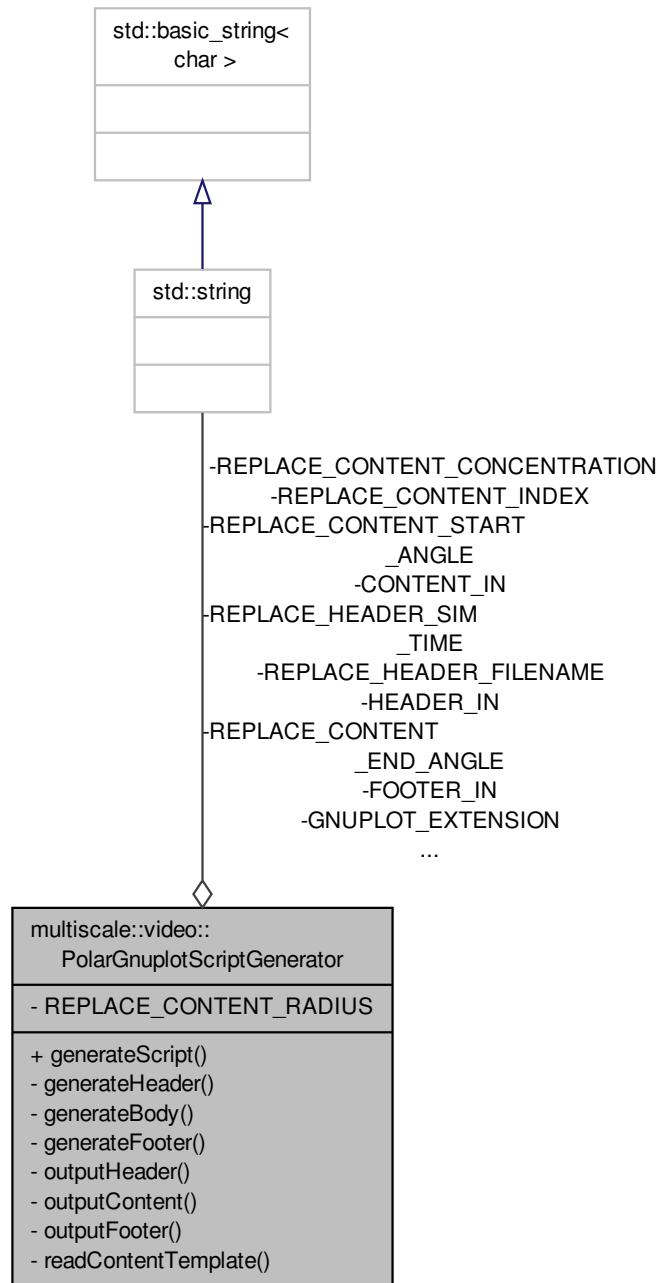
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarCsvToInputFilesConverter.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarCsvToInputFilesConverter.cpp

## 7.138 multiscale::video::PolarGnuplotScriptGenerator Class Reference

Gnuplot script generator from the provided annular sectors.

```
#include <PolarGnuplotScriptGenerator.hpp>
```

Collaboration diagram for multiscale::video::PolarGnuplotScriptGenerator:



## Static Public Member Functions

- static void [generateScript](#) (const std::vector< [AnnularSector](#) > &annularSectors, double simulationTime, const std::string &outputFilepath)

*Generate the script.*

## Static Private Member Functions

- static void [generateHeader](#) (std::ofstream &fout, const std::string &outputFilepath, double simulationTime)
   
*Generate the header of the script.*
- static void [generateBody](#) (const std::vector< [AnnularSector](#) > &annularSectors, std::ofstream &fout)
   
*Generate the body/content of the script.*
- static void [generateFooter](#) (std::ofstream &fout)
   
*Generate the footer of the script.*
- static void [outputHeader](#) (std::ifstream &fin, const std::string &outputFilename, double simulationTime, std::ofstream &fout)
   
*Output the header of the script.*
- static void [outputContent](#) (const std::vector< [AnnularSector](#) > &annularSectors, const std::string &contentTemplate, std::ofstream &fout)
   
*Output the content of the script.*
- static void [outputFooter](#) (std::ifstream &fin, std::ofstream &fout)
   
*Output the footer of the script.*
- static std::string [readContentTemplate](#) (std::ifstream &fin)
   
*Read content template.*

## Static Private Attributes

- static const std::string [HEADER\\_IN](#) = "/usr/local/share/mule/config/video/circular/header.in"
- static const std::string [CONTENT\\_IN](#) = "/usr/local/share/mule/config/video/circular/content.in"
- static const std::string [FOOTER\\_IN](#) = "/usr/local/share/mule/config/video/circular/footer.in"
- static const std::string [REPLACE\\_HEADER\\_FILENAME](#) = "OUTPUT\_FILENAME"
- static const std::string [REPLACE\\_HEADER\\_SIM\\_TIME](#) = "OUTPUT\_SIM\_TIME"
- static const std::string [REPLACE\\_CONTENT\\_INDEX](#) = "OBJ\_INDEX"
- static const std::string [REPLACE\\_CONTENT\\_RADIUS](#) = "OBJ\_END\_RADIUS"
- static const std::string [REPLACE\\_CONTENT\\_START\\_ANGLE](#) = "OBJ\_START\_ANGLE"
- static const std::string [REPLACE\\_CONTENT\\_END\\_ANGLE](#) = "OBJ\_END\_ANGLE"
- static const std::string [REPLACE\\_CONTENT\\_CONCENTRATION](#) = "OBJ\_CONCENTRATION"
- static const std::string [GNUPLOT\\_EXTENSION](#) = ".plt"

### 7.138.1 Detailed Description

Gnuplot script generator from the provided annular sectors.

Definition at line 14 of file PolarGnuplotScriptGenerator.hpp.

### 7.138.2 Member Function Documentation

#### 7.138.2.1 void PolarGnuplotScriptGenerator::generateBody ( const std::vector< [AnnularSector](#) > & *annularSectors*, std::ofstream & *fout* ) [static], [private]

Generate the body/content of the script.

##### Parameters

|                       |                    |
|-----------------------|--------------------|
| <i>annularSectors</i> | Annular sectors    |
| <i>fout</i>           | Output file stream |

Definition at line 40 of file PolarGnuplotScriptGenerator.cpp.

References [CONTENT\\_IN](#), [outputContent\(\)](#), and [readContentTemplate\(\)](#).

Referenced by [generateScript\(\)](#).

7.138.2.2 void PolarGnuplotScriptGenerator::generateFooter ( std::ofstream & *fout* ) [static], [private]

Generate the footer of the script.

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>fout</i> | Output file stream |
|-------------|--------------------|

Definition at line 53 of file PolarGnuplotScriptGenerator.cpp.

References FOOTER\_IN, and outputFooter().

Referenced by generateScript().

**7.138.2.3 void PolarGnuplotScriptGenerator::generateHeader ( std::ofstream & *fout*, const std::string & *outputfilepath*, double *simulationTime* ) [static], [private]**

Generate the header of the script.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>fout</i>           | Output file stream      |
| <i>outputfilepath</i> | Path to the output file |
| <i>simulationTime</i> | Simulation time         |

Definition at line 27 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::filenameFromPath(), HEADER\_IN, and outputHeader().

Referenced by generateScript().

**7.138.2.4 void PolarGnuplotScriptGenerator::generateScript ( const std::vector< AnnularSector > & *annularSectors*, double *simulationTime*, const std::string & *outputfilepath* ) [static]**

Generate the script.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>annularSectors</i> | Annular sectors         |
| <i>simulationTime</i> | Simulation time         |
| <i>outputfilepath</i> | Path of the output file |

Definition at line 13 of file PolarGnuplotScriptGenerator.cpp.

References generateBody(), generateFooter(), generateHeader(), and GNUPLOT\_EXTENSION.

Referenced by multiscale::video::CartesianToPolarConverter::outputResultsAsScript().

**7.138.2.5 void PolarGnuplotScriptGenerator::outputContent ( const std::vector< AnnularSector > & *annularSectors*, const std::string & *contentTemplate*, std::ofstream & *fout* ) [static], [private]**

Output the content of the script.

**Parameters**

|                        |                                                             |
|------------------------|-------------------------------------------------------------|
| <i>annularSectors</i>  | Annular sectors                                             |
| <i>contentTemplate</i> | Template used for generating output for each annular sector |
| <i>fout</i>            | Output file stream                                          |

Definition at line 85 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::replace(), REPLACE\_CONTENT\_CONCENTRATION, REPLACE\_CO\_NTENT\_END\_ANGLE, REPLACE\_CONTENT\_INDEX, REPLACE\_CONTENT\_RADIUS, and REPLACE\_CONT ENT\_START\_ANGLE.

Referenced by generateBody().

```
7.138.2.6 void PolarGnuplotScriptGenerator::outputFooter (std::ifstream & fin, std::ofstream & fout) [static],
[private]
```

Output the footer of the script.

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>fin</i>  | Input file stream  |
| <i>fout</i> | Output file stream |

Definition at line 124 of file PolarGnuplotScriptGenerator.cpp.

Referenced by generateFooter().

**7.138.2.7 void PolarGnuplotScriptGenerator::outputHeader ( std::ifstream & *fin*, const std::string & *outputFilename*, double *simulationTime*, std::ofstream & *fout* ) [static], [private]**

Output the header of the script.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>fin</i>            | Input file stream       |
| <i>outputFilename</i> | Name of the output file |
| <i>simulationTime</i> | Simulation time         |
| <i>fout</i>           | Output file stream      |

Definition at line 63 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::replace(), REPLACE\_HEADER\_FILENAME, and REPLACE\_HEADE←R\_SIM\_TIME.

Referenced by generateHeader().

**7.138.2.8 std::string PolarGnuplotScriptGenerator::readContentTemplate ( std::ifstream & *fin* ) [static], [private]**

Read content template.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 134 of file PolarGnuplotScriptGenerator.cpp.

Referenced by generateBody().

### 7.138.3 Member Data Documentation

**7.138.3.1 const std::string PolarGnuplotScriptGenerator::CONTENT\_IN = "/usr/local/share/mule/config/video/circular/content.in" [static], [private]**

Definition at line 90 of file PolarGnuplotScriptGenerator.hpp.

Referenced by generateBody().

**7.138.3.2 const std::string PolarGnuplotScriptGenerator::FOOTER\_IN = "/usr/local/share/mule/config/video/circular/footer.in" [static], [private]**

Definition at line 91 of file PolarGnuplotScriptGenerator.hpp.

Referenced by generateFooter().

**7.138.3.3 const std::string PolarGnuplotScriptGenerator::GNUPLOT\_EXTENSION = ".plt" [static], [private]**

Definition at line 102 of file PolarGnuplotScriptGenerator.hpp.

Referenced by generateScript().

7.138.3.4 const std::string PolarGnuplotScriptGenerator::HEADER\_IN = "/usr/local/share/mule/config/video/circular/header.in"  
[static], [private]

Definition at line 89 of file PolarGnuplotScriptGenerator.hpp.

Referenced by generateHeader().

7.138.3.5 const std::string PolarGnuplotScriptGenerator::REPLACE\_CONTENT\_CONCENTRATION = "OBJ\_CONCENTRATION"  
[static], [private]

Definition at line 100 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputContent().

7.138.3.6 const std::string PolarGnuplotScriptGenerator::REPLACE\_CONTENT\_END\_ANGLE = "OBJ\_END\_ANGLE"  
[static], [private]

Definition at line 99 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputContent().

7.138.3.7 const std::string PolarGnuplotScriptGenerator::REPLACE\_CONTENT\_INDEX = "OBJ\_INDEX" [static],  
[private]

Definition at line 96 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputContent().

7.138.3.8 const std::string PolarGnuplotScriptGenerator::REPLACE\_CONTENT\_RADIUS = "OBJ\_END\_RADIUS"  
[static], [private]

Definition at line 97 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputContent().

7.138.3.9 const std::string PolarGnuplotScriptGenerator::REPLACE\_CONTENT\_START\_ANGLE = "OBJ\_START\_ANGLE"  
[static], [private]

Definition at line 98 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputContent().

7.138.3.10 const std::string PolarGnuplotScriptGenerator::REPLACE\_HEADER\_FILENAME = "OUTPUT\_FILENAME"  
[static], [private]

Definition at line 93 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

7.138.3.11 const std::string PolarGnuplotScriptGenerator::REPLACE\_HEADER\_SIM\_TIME = "OUTPUT\_SIM\_TIME"  
[static], [private]

Definition at line 94 of file PolarGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

The documentation for this class was generated from the following files:

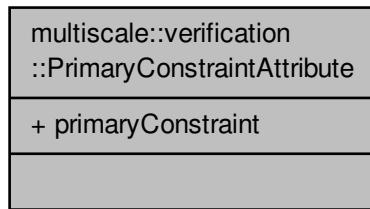
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/[PolarGnuplotScriptGenerator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/[PolarGnuplotScriptGenerator.cpp](#)

## 7.139 multiscale::verification::PrimaryConstraintAttribute Class Reference

Class for representing a primary constraint attribute.

```
#include <PrimaryConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::PrimaryConstraintAttribute:



### Public Attributes

- [PrimaryConstraintAttributeType primaryConstraint](#)

#### 7.139.1 Detailed Description

Class for representing a primary constraint attribute.

Definition at line 32 of file PrimaryConstraintAttribute.hpp.

#### 7.139.2 Member Data Documentation

##### 7.139.2.1 PrimaryConstraintAttributeType multiscale::verification::PrimaryConstraintAttribute::primaryConstraint

The primary constraint

Definition at line 36 of file PrimaryConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

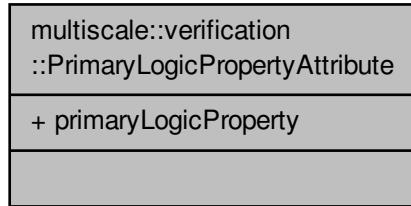
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[PrimaryConstraintAttribute.hpp](#)

## 7.140 multiscale::verification::PrimaryLogicPropertyAttribute Class Reference

Class for representing a primary logic property attribute.

```
#include <PrimaryLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::PrimaryLogicPropertyAttribute:



## Public Attributes

- [PrimaryLogicPropertyAttributeType primaryLogicProperty](#)

### 7.140.1 Detailed Description

Class for representing a primary logic property attribute.

Definition at line 40 of file PrimaryLogicPropertyAttribute.hpp.

### 7.140.2 Member Data Documentation

#### 7.140.2.1 PrimaryLogicPropertyAttributeType multiscale::verification::PrimaryLogicAttribute::primaryLogicProperty

The primary logic property

Definition at line 44 of file PrimaryLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::operator()().

The documentation for this class was generated from the following file:

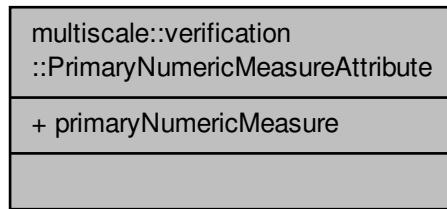
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryLogicPropertyAttribute.hpp](#)

## 7.141 multiscale::verification::PrimaryNumericMeasureAttribute Class Reference

Class for representing a primary numeric measure attribute.

```
#include <PrimaryNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::PrimaryNumericMeasureAttribute:



## Public Attributes

- [PrimaryNumericMeasureAttributeType primaryNumericMeasure](#)

### 7.141.1 Detailed Description

Class for representing a primary numeric measure attribute.

Definition at line 28 of file PrimaryNumericMeasureAttribute.hpp.

### 7.141.2 Member Data Documentation

#### 7.141.2.1 PrimaryNumericMeasureAttributeType multiscale::verification::PrimaryNumericMeasureAttribute::primary< NumericMeasure >

The primary numeric measure

Definition at line 32 of file PrimaryNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::FilterNumericVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

The documentation for this class was generated from the following file:

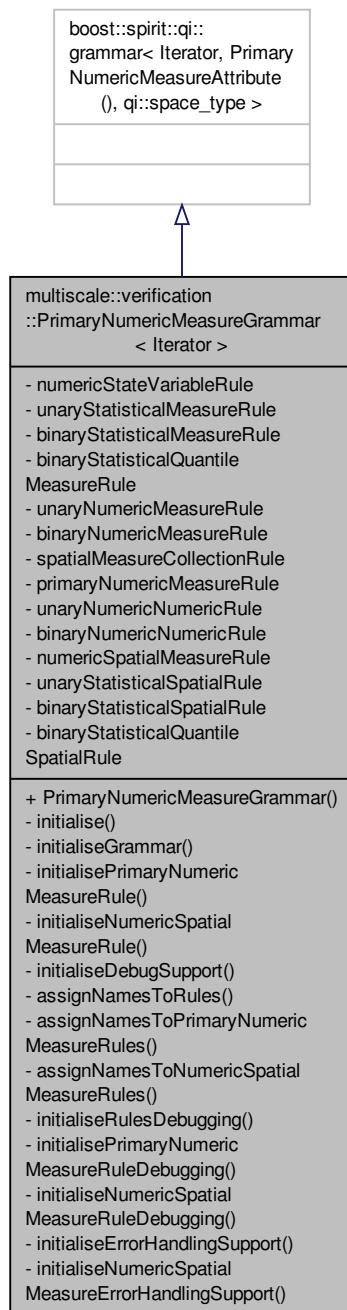
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[PrimaryNumericMeasureAttribute.hpp](#)

## 7.142 multiscale::verification::PrimaryNumericMeasureGrammar< Iterator > Singleton Reference

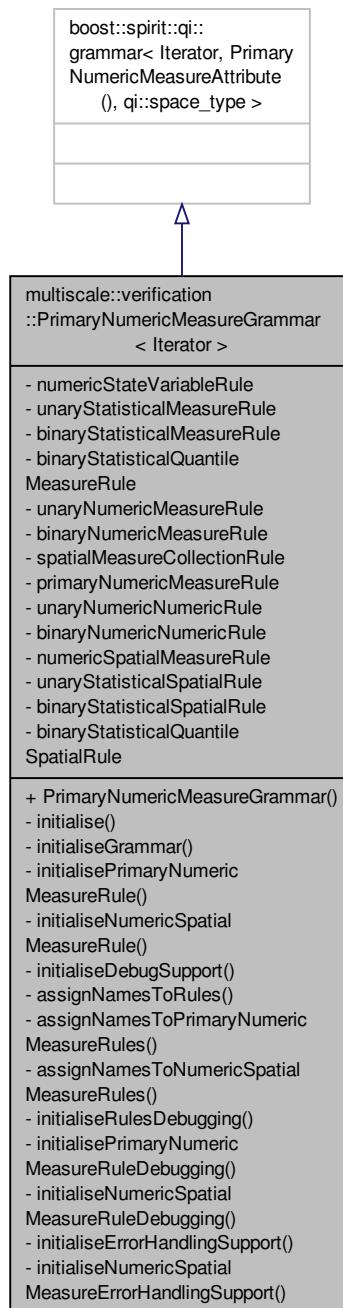
The grammar for parsing primary numeric measure statements.

```
#include <PrimaryNumericMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >:



## Public Member Functions

- `PrimaryNumericMeasureGrammar (SpatialMeasureCollectionGrammar< Iterator > *spatialMeasure← CollectionGrammar)`

## Private Member Functions

- void `initialise ()`  
*Initialisation function.*
- void `initialiseGrammar ()`  
*Initialise the grammar.*
- void `initialisePrimaryNumericMeasureRule ()`  
*Initialise the primary numeric measure rule.*
- void `initialiseNumericSpatialMeasureRule ()`  
*Initialise the numeric spatial measure rule.*
- void `initialiseDebugSupport ()`  
*Initialise debug support.*
- void `assignNamesToRules ()`  
*Assign names to the rules.*
- void `assignNamesToPrimaryNumericMeasureRules ()`  
*Assign names to the primary numeric measure rules.*
- void `assignNamesToNumericSpatialMeasureRules ()`  
*Assign names to the numeric spatial measure rules.*
- void `initialiseRulesDebugging ()`  
*Initialise the debugging of rules.*
- void `initialisePrimaryNumericMeasureRuleDebugging ()`  
*Initialise debugging for the primary numeric measure rule.*
- void `initialiseNumericSpatialMeasureRuleDebugging ()`  
*Initialise debugging for the numeric spatial measure rule.*
- void `initialiseErrorHandlingSupport ()`  
*Initialise the error handling routines.*
- void `initialiseNumericSpatialMeasureErrorHandlingSupport ()`  
*Initialise the numeric spatial measure error handling support.*

## Private Attributes

- `NumericStateVariableGrammar < Iterator > numericStateVariableRule`
- `UnaryStatisticalMeasureGrammar < Iterator > unaryStatisticalMeasureRule`
- `BinaryStatisticalMeasureGrammar < Iterator > binaryStatisticalMeasureRule`
- `BinaryStatisticalQuantileMeasureGrammar < Iterator > binaryStatisticalQuantileMeasureRule`
- `UnaryNumericMeasureGrammar < Iterator > unaryNumericMeasureRule`
- `BinaryNumericMeasureGrammar < Iterator > binaryNumericMeasureRule`
- `SpatialMeasureCollectionGrammar < Iterator > * spatialMeasureCollectionRule`
- `qi::rule< Iterator, PrimaryNumericMeasureAttribute(), qi::space_type > primaryNumericMeasureRule`
- `qi::rule< Iterator, UnaryNumericNumericAttribute(), qi::space_type > unaryNumericNumericRule`

- qi::rule< Iterator,  
  BinaryNumericNumericAttribute(),  
  qi::space\_type > binaryNumericNumericRule
- qi::rule< Iterator,  
  NumericSpatialMeasureAttribute(),  
  qi::space\_type > numericSpatialMeasureRule
- qi::rule< Iterator,  
  UnaryStatisticalSpatialAttribute(),  
  qi::space\_type > unaryStatisticalSpatialRule
- qi::rule< Iterator,  
  BinaryStatisticalSpatialAttribute(),  
  qi::space\_type > binaryStatisticalSpatialRule
- qi::rule< Iterator,  
  BinaryStatisticalQuantileSpatialAttribute(),  
  qi::space\_type > binaryStatisticalQuantileSpatialRule

### 7.142.1 Detailed Description

```
template<typename Iterator>singleton multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >
```

The grammar for parsing primary numeric measure statements.

Definition at line 43 of file PrimaryNumericMeasureGrammar.hpp.

### 7.142.2 Constructor & Destructor Documentation

```
7.142.2.1 template<typename Iterator > multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >
 ::PrimaryNumericMeasureGrammar (SpatialMeasureCollectionGrammar< Iterator > *
 spatialMeasureCollectionGrammar)
```

Definition at line 28 of file PrimaryNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialise().

### 7.142.3 Member Function Documentation

```
7.142.3.1 template<typename Iterator > void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >
 ::assignNamesToNumericSpatialMeasureRules () [private]
```

Assign names to the numeric spatial measure rules.

Definition at line 120 of file PrimaryNumericMeasureGrammarDefinition.hpp.

```
7.142.3.2 template<typename Iterator > void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >
 ::assignNamesToPrimaryNumericMeasureRules () [private]
```

Assign names to the primary numeric measure rules.

Definition at line 114 of file PrimaryNumericMeasureGrammarDefinition.hpp.

```
7.142.3.3 template<typename Iterator > void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >
 ::assignNamesToRules () [private]
```

Assign names to the rules.

Definition at line 107 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.4 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialise( ) [private]

Initialisation function.

Definition at line 37 of file PrimaryNumericMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::PrimaryNumericMeasureGrammar().

7.142.3.5 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseDebugSupport( ) [private]

Initialise debug support.

Definition at line 98 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.6 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]

Initialise the error handling routines.

Definition at line 151 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.7 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 45 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.8 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseNumericSpatialMeasureErrorHandlingSupport( ) [private]

Initialise the numeric spatial measure error handling support.

Definition at line 157 of file PrimaryNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.142.3.9 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseNumericSpatialMeasureRule( ) [private]

Initialise the numeric spatial measure rule.

Definition at line 61 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.10 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseNumericSpatialMeasureRuleDebugging( ) [private]

Initialise debugging for the numeric spatial measure rule.

Definition at line 142 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.11 template<typename Iterator> void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialisePrimaryNumericMeasureRule( ) [private]

Initialise the primary numeric measure rule.

Definition at line 52 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.12 template<typename Iterator > void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialisePrimaryNumericMeasureRuleDebugging ( ) [private]

Initialise debugging for the primary numeric measure rule.

Definition at line 136 of file PrimaryNumericMeasureGrammarDefinition.hpp.

7.142.3.13 template<typename Iterator > void multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::initialiseRulesDebugging ( ) [private]

Initialise the debugging of rules.

Definition at line 129 of file PrimaryNumericMeasureGrammarDefinition.hpp.

#### 7.142.4 Member Data Documentation

7.142.4.1 template<typename Iterator > BinaryNumericMeasureGrammar<Iterator> multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryNumericMeasureRule [private]

The grammar for parsing binary numeric measures

Definition at line 68 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.2 template<typename Iterator > qi::rule<Iterator, BinaryNumericNumericAttribute(), qi::space\_type> multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryNumericNumericRule [private]

The rule for parsing a binary numeric numeric attribute

Definition at line 84 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.3 template<typename Iterator > BinaryStatisticalMeasureGrammar<Iterator> multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryStatisticalMeasureRule [private]

The grammar for parsing binary statistical measures

Definition at line 58 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.4 template<typename Iterator > BinaryStatisticalQuantileMeasureGrammar<Iterator> multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryStatisticalQuantileMeasureRule [private]

The grammar for parsing binary statistical quantile measures

Definition at line 61 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.5 template<typename Iterator > qi::rule<Iterator, BinaryStatisticalQuantileSpatialAttribute(), qi::space\_type> multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryStatisticalQuantileSpatialRule [private]

The rule for parsing a binary statistical quantile spatial attribute

Definition at line 96 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.6 template<typename Iterator> qi::rule<Iterator, BinaryStatisticalSpatialAttribute(), qi::space_type>
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::binaryStatisticalSpatialRule
[private]
```

The rule for parsing a binary statistical spatial attribute

Definition at line 93 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.7 template<typename Iterator> qi::rule<Iterator, NumericSpatialMeasureAttribute(), qi::space_type>
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::numericSpatialMeasureRule
[private]
```

The rule for parsing a numeric spatial measure

Definition at line 88 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.8 template<typename Iterator> NumericStateVariableGrammar<Iterator> multiscale<-
::verification::PrimaryNumericMeasureGrammar< Iterator >::numericStateVariableRule
[private]
```

The grammar for parsing numeric state variables

Definition at line 51 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.9 template<typename Iterator> qi::rule<Iterator, PrimaryNumericMeasureAttribute(), qi::space_type>
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::primaryNumericMeasureRule
[private]
```

The rule for parsing a primary numeric numeric attribute

Definition at line 78 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.10 template<typename Iterator> SpatialMeasureCollectionGrammar<Iterator>*
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::spatialMeasureCollectionRule
[private]
```

The grammar for parsing spatial measure collections

Definition at line 72 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.11 template<typename Iterator> UnaryNumericMeasureGrammar<Iterator> multiscale<-
::verification::PrimaryNumericMeasureGrammar< Iterator >::unaryNumericMeasureRule
[private]
```

The grammar for parsing unary numeric measures

Definition at line 65 of file PrimaryNumericMeasureGrammar.hpp.

```
7.142.4.12 template<typename Iterator> qi::rule<Iterator, UnaryNumericNumericAttribute(), qi::space_type>
multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >::unaryNumericNumericRule
[private]
```

The rule for parsing a unary numeric numeric attribute

Definition at line 81 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.13 template<typename Iterator > **UnaryStatisticalMeasureGrammar**<Iterator> **multiscale**←  
::verification::PrimaryNumericMeasureGrammar< Iterator >::unaryStatisticalMeasureRule  
[private]

The grammar for parsing unary statistical measures

Definition at line 55 of file PrimaryNumericMeasureGrammar.hpp.

7.142.4.14 template<typename Iterator > qi::rule<Iterator, UnaryStatisticalSpatialAttribute(), qi::space\_type>  
**multiscale**::verification::PrimaryNumericMeasureGrammar< Iterator >::unaryStatisticalSpatialRule  
[private]

The rule for parsing a unary statistical spatial attribute

Definition at line 90 of file PrimaryNumericMeasureGrammar.hpp.

The documentation for this singleton was generated from the following files:

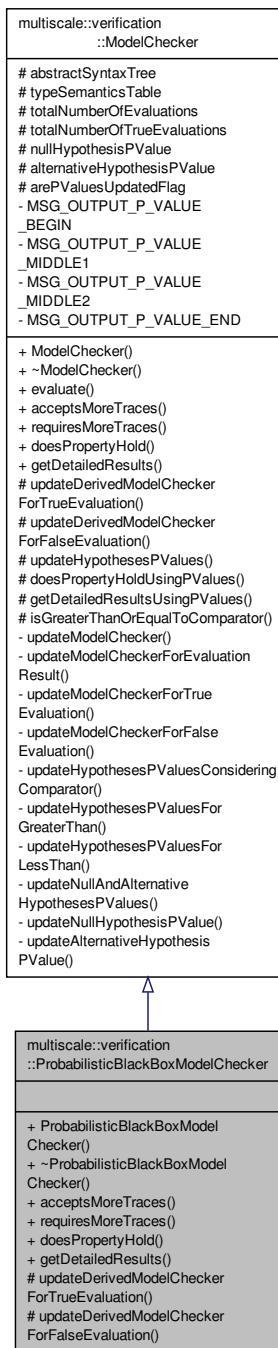
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[PrimaryNumericMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[PrimaryNumericMeasureGrammarDefinition.hpp](#)

## 7.143 multiscale::verification::ProbabilisticBlackBoxModelChecker Class Reference

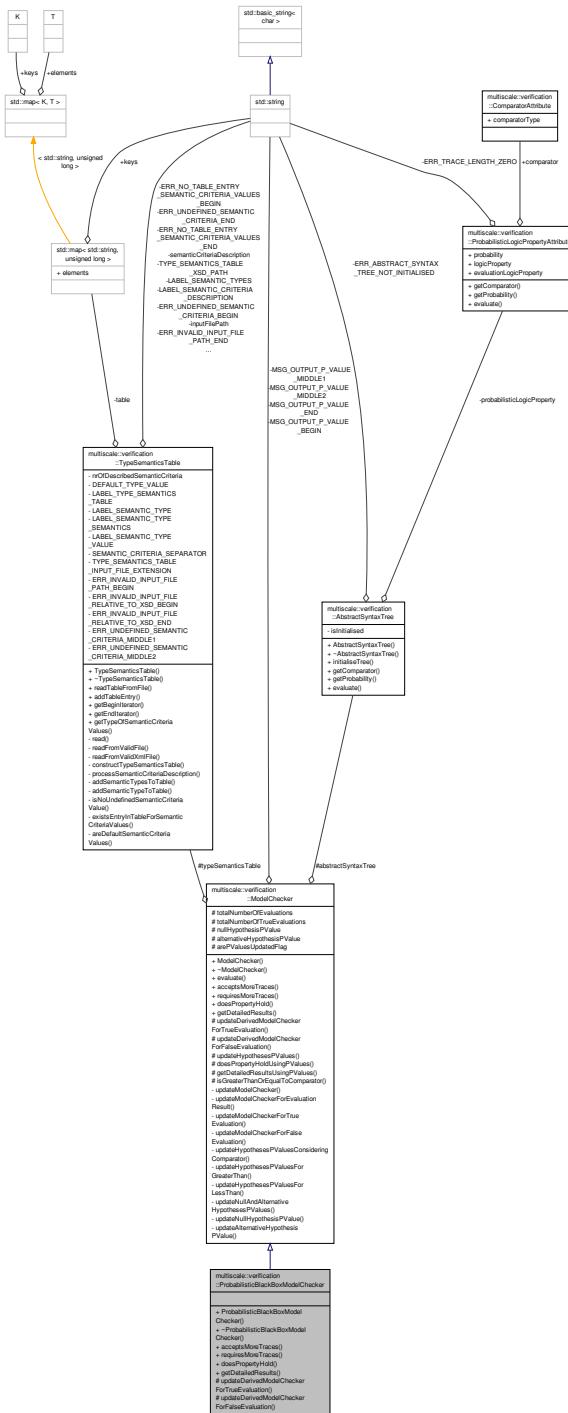
Class used to run probabilistic black-box model checking tasks.

```
#include <ProbabilisticBlackBoxModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker:



## Collaboration diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker:



## Public Member Functions

- ProbabilisticBlackBoxModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, const TypeSemanticsTable &typeSemanticsTable)
  - ~ProbabilisticBlackBoxModelChecker ()
  - bool acceptsMoreTraces () override

*Check if more traces are accepted for evaluating the logic property.*

- bool [requiresMoreTraces \(\)](#) override  
*Check if more traces are required for evaluating the logic property.*
- bool [doesPropertyHold \(\)](#) override  
*Check if the given property holds.*
- std::string [getDetailedResults \(\)](#) override  
*Get the detailed description of the results.*

## Protected Member Functions

- void [updateDerivedModelCheckerForTrueEvaluation \(\)](#) override  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- void [updateDerivedModelCheckerForFalseEvaluation \(\)](#) override  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*

## Additional Inherited Members

### 7.143.1 Detailed Description

Class used to run probabilistic black-box model checking tasks.

The implementation of this class is (partially) based on the algorithms described in the following paper:

H. L. S. Younes, ‘Probabilistic Verification for “Black-Box” Systems’, in Computer Aided Verification, K. Etessami and S. K. Rajamani, Eds. Springer Berlin Heidelberg, 2005, pp. 253–265.

Definition at line 21 of file ProbabilisticBlackBoxModelChecker.hpp.

### 7.143.2 Constructor & Destructor Documentation

#### 7.143.2.1 ProbabilisticBlackBoxModelChecker::ProbabilisticBlackBoxModelChecker ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable )

Definition at line 12 of file ProbabilisticBlackBoxModelChecker.cpp.

#### 7.143.2.2 ProbabilisticBlackBoxModelChecker::~ProbabilisticBlackBoxModelChecker ( )

Definition at line 19 of file ProbabilisticBlackBoxModelChecker.cpp.

### 7.143.3 Member Function Documentation

#### 7.143.3.1 bool ProbabilisticBlackBoxModelChecker::acceptsMoreTraces ( ) [override], [virtual]

Check if more traces are accepted for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 21 of file ProbabilisticBlackBoxModelChecker.cpp.

#### 7.143.3.2 bool ProbabilisticBlackBoxModelChecker::doesPropertyHold ( ) [override], [virtual]

Check if the given property holds.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 29 of file ProbabilisticBlackBoxModelChecker.cpp.

References multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues().

#### 7.143.3.3 std::string ProbabilisticBlackBoxModelChecker::getDetailedResults( ) [override], [virtual]

Get the detailed description of the results.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 33 of file ProbabilisticBlackBoxModelChecker.cpp.

References multiscale::verification::ModelChecker::getDetailedResultsUsingPValues().

#### 7.143.3.4 bool ProbabilisticBlackBoxModelChecker::requiresMoreTraces( ) [override], [virtual]

Check if more traces are required for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 25 of file ProbabilisticBlackBoxModelChecker.cpp.

#### 7.143.3.5 void ProbabilisticBlackBoxModelChecker::updateDerivedModelCheckerForFalseEvaluation( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 39 of file ProbabilisticBlackBoxModelChecker.cpp.

#### 7.143.3.6 void ProbabilisticBlackBoxModelChecker::updateDerivedModelCheckerForTrueEvaluation( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 37 of file ProbabilisticBlackBoxModelChecker.cpp.

The documentation for this class was generated from the following files:

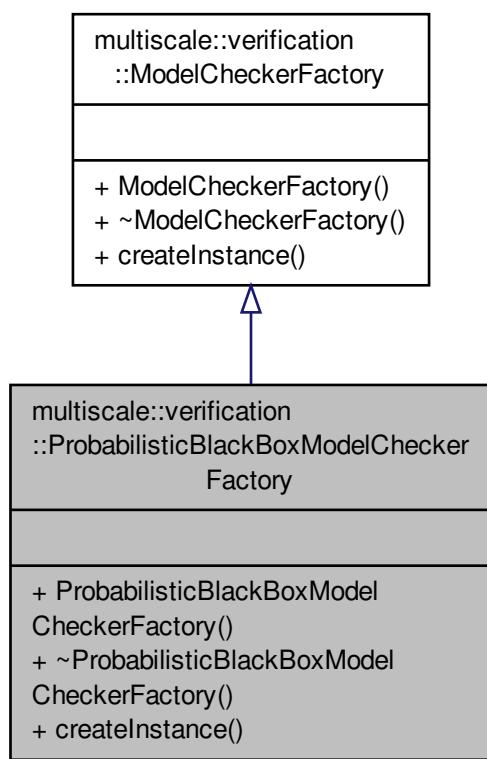
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ProbabilisticBlackBoxModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ProbabilisticBlackBoxModelChecker.cpp](#)

## 7.144 multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory Class Reference

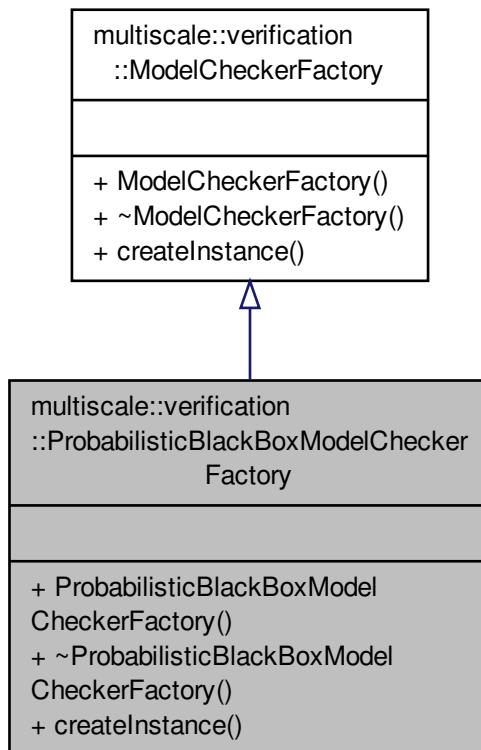
Class for creating [ProbabilisticBlackBoxModelChecker](#) instances.

```
#include <ProbabilisticBlackBoxModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory:



Collaboration diagram for multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory:



## Public Member Functions

- [ProbabilisticBlackBoxModelCheckerFactory \(\)](#)
- [~ProbabilisticBlackBoxModelCheckerFactory \(\)](#)
- [std::shared\\_ptr< ModelChecker > createInstance \(const AbstractSyntaxTree &abstractSyntaxTree, const TypeSemanticsTable &typeSemanticsTable\) override](#)

*Create an instance of [ProbabilisticBlackBoxModelChecker](#).*

### 7.144.1 Detailed Description

Class for creating [ProbabilisticBlackBoxModelChecker](#) instances.

Definition at line 12 of file ProbabilisticBlackBoxModelCheckerFactory.hpp.

### 7.144.2 Constructor & Destructor Documentation

#### 7.144.2.1 ProbabilisticBlackBoxModelCheckerFactory::ProbabilisticBlackBoxModelCheckerFactory ( )

Definition at line 7 of file ProbabilisticBlackBoxModelCheckerFactory.cpp.

## 7.144.2.2 ProbabilisticBlackBoxModelCheckerFactory::~ProbabilisticBlackBoxModelCheckerFactory( )

Definition at line 9 of file ProbabilisticBlackBoxModelCheckerFactory.cpp.

## 7.144.3 Member Function Documentation

7.144.3.1 `std::shared_ptr< ModelChecker > ProbabilisticBlackBoxModelCheckerFactory::createInstance ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [override], [virtual]`

Create an instance of [ProbabilisticBlackBoxModelChecker](#).

## Parameters

|                                 |                                                                                       |
|---------------------------------|---------------------------------------------------------------------------------------|
| <code>abstractSyntaxTree</code> | The abstract syntax tree representing the logic property to be checked                |
| <code>typeSemanticsTable</code> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implements [multiscale::verification::ModelCheckerFactory](#).

Definition at line 12 of file ProbabilisticBlackBoxModelCheckerFactory.cpp.

The documentation for this class was generated from the following files:

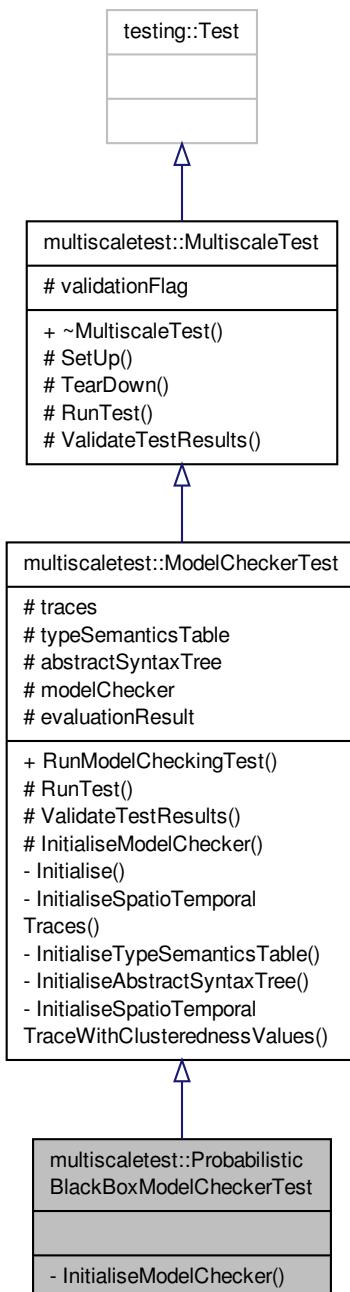
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ProbabilisticBlackBoxModelCheckerFactory.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ProbabilisticBlackBoxModelCheckerFactory.cpp](#)

## 7.145 multiscaletest::ProbabilisticBlackBoxModelCheckerTest Class Reference

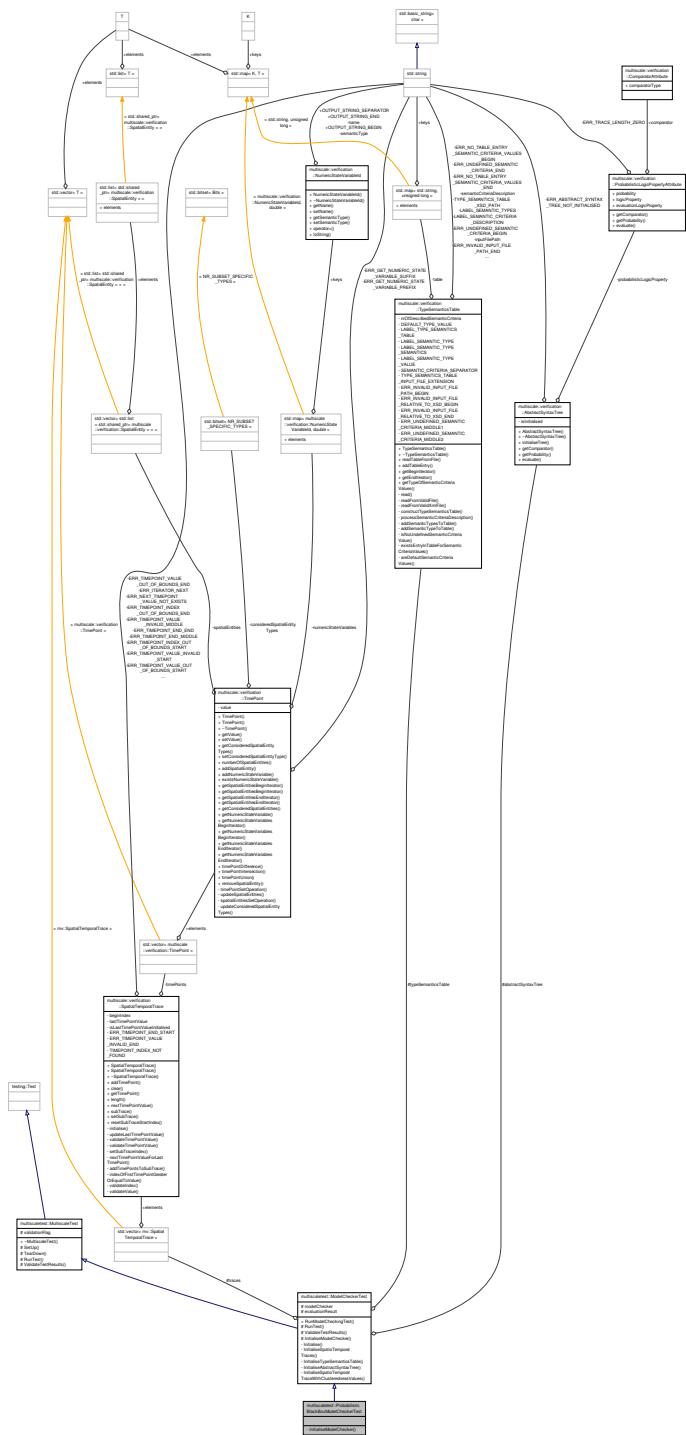
Class for testing the probabilistic black-box model checker.

```
#include <ProbabilisticBlackBoxModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::ProbabilisticBlackBoxModelCheckerTest:



## Collaboration diagram for multiscaletest::ProbabilisticBlackBoxModelCheckerTest:



## Private Member Functions

- void **InitialiseModelChecker** () override  
*Initialise the model checker.*

### Additional Inherited Members

### 7.145.1 Detailed Description

Class for testing the probabilistic black-box model checker.

Definition at line 15 of file ProbabilisticBlackBoxModelCheckerTest.hpp.

### 7.145.2 Member Function Documentation

7.145.2.1 `void multiscaletest::ProbabilisticBlackBoxModelCheckerTest::InitialiseModelChecker( ) [override], [private], [virtual]`

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 25 of file ProbabilisticBlackBoxModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

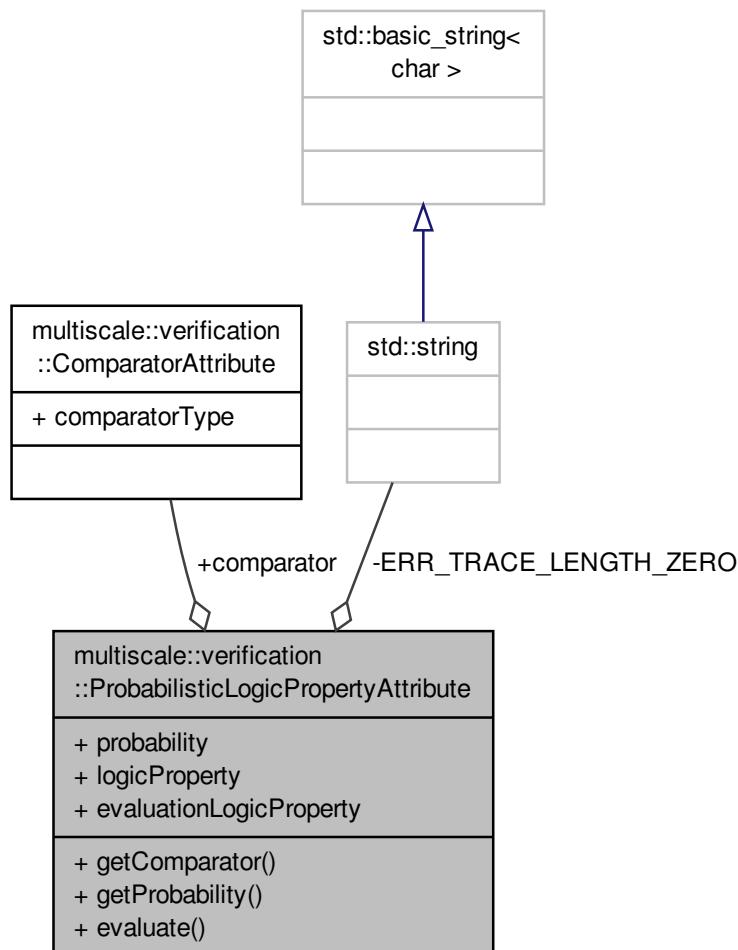
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/[ProbabilisticBlackBoxModelCheckerTest.hpp](#)

## 7.146 multiscale::verification::ProbabilisticLogicPropertyAttribute Class Reference

Class for representing a probabilistic logic property attribute.

```
#include <ProbabilisticLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::ProbabilisticLogicPropertyAttribute:



## Public Member Functions

- [ComparatorType getComparator \(\)](#)  
*Get the type of the comparator.*
- [double getProbability \(\)](#)  
*Get the probability.*
- [bool evaluate \(const SpatialTemporalTrace &trace, const TypeSemanticsTable &typeSemanticsTable\)](#)  
*Evaluate the truth value of the logic property considering the given trace and type semantics table.*

## Public Attributes

- [ComparatorAttributeType comparator](#)
- [double probability](#)
- [LogicPropertyAttributeType logicProperty](#)
- [LogicPropertyAttributeType evaluationLogicProperty](#)

## Static Private Attributes

- static const std::string **ERR\_TRACE\_LENGTH\_ZERO** = "The length of the trace provided for evaluating the probabilistic logic property is zero. Please provide a trace which contains at least one timepoint."

### 7.146.1 Detailed Description

Class for representing a probabilistic logic property attribute.

Definition at line 19 of file ProbabilisticLogicPropertyAttribute.hpp.

### 7.146.2 Member Function Documentation

#### 7.146.2.1 bool ProbabilisticLogicPropertyAttribute::evaluate ( const SpatialTemporalTrace & trace, const TypeSemanticsTable & typeSemanticsTable )

Evaluate the truth value of the logic property considering the given trace and type semantics table.

##### Parameters

|                           |                                             |
|---------------------------|---------------------------------------------|
| <i>trace</i>              | The considered trace (i.e. timeseries data) |
| <i>typeSemanticsTable</i> | The considered type semantics table         |

Definition at line 15 of file ProbabilisticLogicPropertyAttribute.cpp.

References **ERR\_TRACE\_LENGTH\_ZERO**, **evaluationLogicProperty**, **multiscale::verification::SpatialTemporalTrace::length()**, **logicProperty**, and **MS\_throw**.

Referenced by **multiscale::verification::AbstractSyntaxTree::evaluate()**.

#### 7.146.2.2 ComparatorType ProbabilisticLogicPropertyAttribute::getComparator ( )

Get the type of the comparator.

Definition at line 7 of file ProbabilisticLogicPropertyAttribute.cpp.

References **comparator**, and **multiscale::verification::ComparatorAttribute::comparatorType**.

Referenced by **multiscale::verification::AbstractSyntaxTree::getComparator()**.

#### 7.146.2.3 double ProbabilisticLogicPropertyAttribute::getProbability ( )

Get the probability.

Definition at line 11 of file ProbabilisticLogicPropertyAttribute.cpp.

References **probability**.

Referenced by **multiscale::verification::AbstractSyntaxTree::getProbability()**.

### 7.146.3 Member Data Documentation

#### 7.146.3.1 ComparatorAttribute multiscale::verification::ProbabilisticLogicPropertyAttribute::comparator

The comparator

Definition at line 23 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by **getComparator()**.

7.146.3.2 const std::string ProbabilisticLogicPropertyAttribute::ERR\_TRACE\_LENGTH\_ZERO = "The length of the trace provided for evaluating the probabilistic logic property is zero. Please provide a trace which contains at least one timepoint." [static], [private]

Definition at line 49 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.146.3.3 LogicPropertyAttributeType multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluationLogicProperty

The logic property used only for evaluation purposes

Definition at line 27 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.146.3.4 LogicPropertyAttributeType multiscale::verification::ProbabilisticLogicPropertyAttribute::logicProperty

The logic property

Definition at line 25 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.146.3.5 double multiscale::verification::ProbabilisticLogicPropertyAttribute::probability

The probability

Definition at line 24 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by getProbability().

The documentation for this class was generated from the following files:

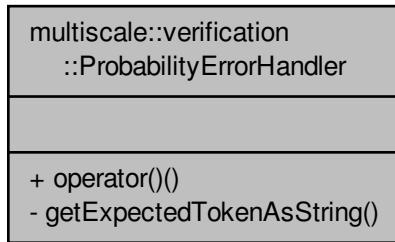
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ProbabilisticLogicPropertyAttribute.cpp

## 7.147 multiscale::verification::ProbabilityErrorHandler Struct Reference

Structure for defining the error handler for invalid probability errors.

```
#include <ProbabilityErrorHandler.hpp>
```

Collaboration diagram for multiscale::verification::ProbabilityErrorHandler:



## Classes

- struct **result**

*Structure for specifying the type of the result.*

## Public Member Functions

- template<typename Iterator >  
void **operator()** (qi::info const &expectedToken, Iterator errorPosition, Iterator last) const  
*Overloaded operator.*

## Private Member Functions

- std::string **getExpectedTokenAsString** (qi::info const &expectedToken) const  
*Convert the expected token to a string.*

### 7.147.1 Detailed Description

Structure for defining the error handler for invalid probability errors.

Definition at line 17 of file ProbabilityErrorHandler.hpp.

### 7.147.2 Member Function Documentation

7.147.2.1 std::string multiscale::verification::ProbabilityErrorHandler::getExpectedTokenAsString ( qi::info const & *expectedToken* ) const [inline], [private]

Convert the expected token to a string.

Convert the expected token to a string and remove enclosing quotes

#### Parameters

---

|                      |                                   |
|----------------------|-----------------------------------|
| <i>expectedToken</i> | The expected token (not a string) |
|----------------------|-----------------------------------|

Definition at line 46 of file ProbabilityErrorHandler.hpp.

Referenced by operator()().

7.147.2.2 template<typename Iterator > void multiscale::verification::ProbabilityErrorHandler::operator() ( qi::info const & *expectedToken*, Iterator *errorPosition*, Iterator *last* ) const [inline]

Overloaded operator.

#### Parameters

|                      |                                           |
|----------------------|-------------------------------------------|
| <i>expectedToken</i> | The expected token                        |
| <i>errorPosition</i> | Iterator pointing to the error position   |
| <i>last</i>          | Iterator pointing to the end of the query |

Definition at line 32 of file ProbabilityErrorHandler.hpp.

References getExpectedTokenAsString().

The documentation for this struct was generated from the following file:

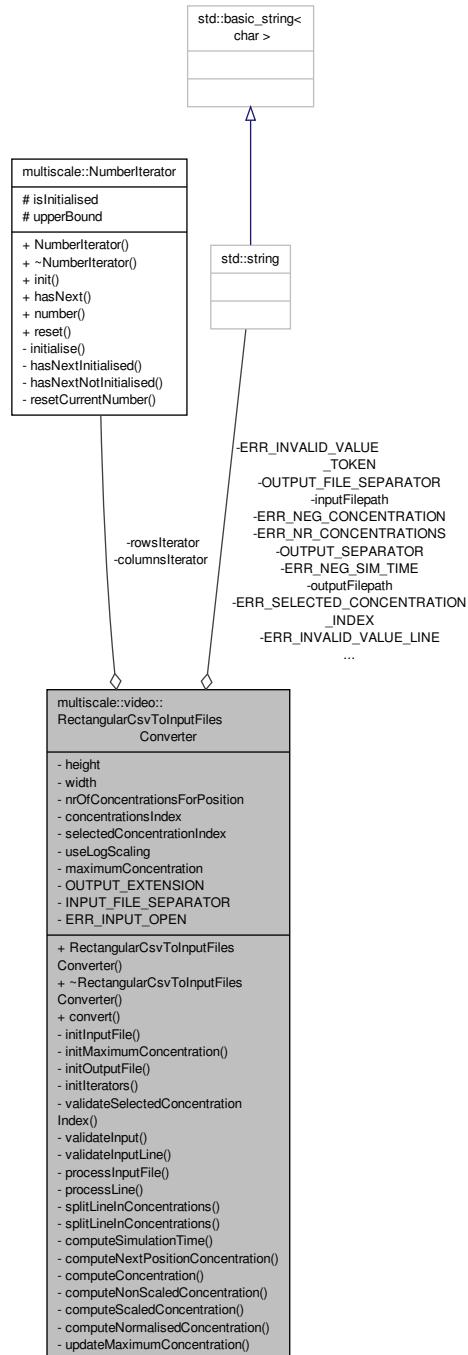
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/[ProbabilityErrorHandler.hpp](#)

## 7.148 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

Csv file to input file converter considering cartesian coordinates.

```
#include <RectangularCsvToInputFilesConverter.hpp>
```

Collaboration diagram for multiscale::video::RectangularCsvToInputFilesConverter:



## Public Member Functions

- **RectangularCsvToInputFilesConverter** (const std::string &**inputFilepath**, const std::string &**outputFilepath**, unsigned int **height**, unsigned int **width**, unsigned int **nrOfConcentrationsForPosition**, unsigned int **selectedConcentrationIndex**, bool **useLogScaling**, **NumberIteratorType** **numberIteratorType**)
- **~RectangularCsvToInputFilesConverter ()**
- **void convert ()**

*Start the conversion.*

## Private Member Functions

- void `initInputFile` (std::ifstream &fin)  
*Initialise the input file stream over the given input file.*
- void `initMaximumConcentration` (std::ifstream &fin)  
*Compute the value of member maximum concentration.*
- void `initOutputFile` (std::ofstream &fout, unsigned int index, double &simulationTime)  
*Initialise the output file with the given index and simulation time.*
- void `initIterators` (const NumberIteratorType &numberIteratorType)  
*Initialise the iterators considering the given number iterator type.*
- void `validateSelectedConcentrationIndex` ()  
*Validate the selected concentration index in case of more than one concentration for each position.*
- void `validateInput` (std::ifstream &fin)  
*Validate the input.*
- void `validateInputLine` (const std::string &line, unsigned int lineNumber)  
*Validate the provided line identified by a line number.*
- void `processInputFile` (std::ifstream &fin)  
*Process the input file.*
- void `processLine` (const std::string &line, unsigned int outputIndex)  
*Process the provided line.*
- std::vector< double > `splitLineInConcentrations` (const std::string &line, double &simulationTime)  
*Split the line in concentrations.*
- void `splitLineInConcentrations` (std::vector< double > &concentrations, std::vector< std::string > &tokens, unsigned int rowIndex)  
*Split line into concentrations.*
- double `computeSimulationTime` (const std::string &token)  
*Compute the simulation time from the given token and check if it is valid.*
- double `computeNextPositionConcentration` (int concentrationIndex, std::vector< std::string > &tokens)  
*Compute the concentration for the next position.*
- double `computeConcentration` (const std::string &concentration)  
*Compute the concentration from the given std::string.*
- double `computeNonScaledConcentration` (const std::string &concentration)  
*Compute the non-scaled concentration from the given std::string.*
- double `computeScaledConcentration` (const std::string &concentration)  
*Compute the scaled concentration from the given std::string.*
- double `computeNormalisedConcentration` (double concentration)  
*Normalise the given concentration by dividing it to the maximum concentration.*
- void `updateMaximumConcentration` (const std::string &line, double &maximumConcentration)  
*Update the maximum concentration if the values from the given line are greater than it.*

## Private Attributes

- std::string `inputfilepath`
- std::string `outputfilepath`
- unsigned int `height`
- unsigned int `width`
- unsigned int `nrOfConcentrationsForPosition`
- unsigned int `concentrationsIndex`
- unsigned int `selectedConcentrationIndex`
- bool `useLogScaling`
- double `maximumConcentration`
- NumberIterator \* `rowsIterator`
- NumberIterator \* `columnsIterator`

## Static Private Attributes

- static const std::string `OUTPUT_EXTENSION` = ".in"
- static const std::string `OUTPUT_SEPARATOR` = " "
- static const std::string `OUTPUT_FILE_SEPARATOR` = "\_"
- static const std::string `INPUT_FILE_SEPARATOR` = ","
- static const std::string `ERR_NEG_CONCENTRATION` = "All concentrations must be non-negative."
- static const std::string `ERR_SELECTED_CONCENTRATION_INDEX` = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations."
- static const std::string `ERR_NR_CONCENTRATIONS` = "The number of concentrations in the input file does not match the values of the input parameters `height` and `width`."
- static const std::string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const std::string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const std::string `ERR_INVALID_VALUE_LINE` = "Invalid value on line: "
- static const std::string `ERR_INVALID_VALUE_TOKEN` = ", value: "

### 7.148.1 Detailed Description

Csv file to input file converter considering cartesian coordinates.

Definition at line 16 of file `RectangularCsvToInputFilesConverter.hpp`.

### 7.148.2 Constructor & Destructor Documentation

7.148.2.1 `RectangularCsvToInputFilesConverter::RectangularCsvToInputFilesConverter ( const std::string & inputFilepath, const std::string & outputFilepath, unsigned int height, unsigned int width, unsigned int nrOfConcentrationsForPosition, unsigned int selectedConcentrationIndex, bool useLogScaling, NumberIteratorType numberIteratorType )`

Definition at line 19 of file `RectangularCsvToInputFilesConverter.cpp`.

7.148.2.2 `RectangularCsvToInputFilesConverter::~RectangularCsvToInputFilesConverter ( )`

Definition at line 43 of file `RectangularCsvToInputFilesConverter.cpp`.

### 7.148.3 Member Function Documentation

7.148.3.1 `double RectangularCsvToInputFilesConverter::computeConcentration ( const std::string & concentration ) [private]`

Compute the concentration from the given std::string.

Parameters

|                            |                                       |
|----------------------------|---------------------------------------|
| <code>concentration</code> | String representing the concentration |
|----------------------------|---------------------------------------|

Definition at line 285 of file `RectangularCsvToInputFilesConverter.cpp`.

7.148.3.2 `double RectangularCsvToInputFilesConverter::computeNextPositionConcentration ( int concentrationIndex, std::vector< std::string > & tokens ) [private]`

Compute the concentration for the next position.

**Parameters**

|                                      |                                                                   |
|--------------------------------------|-------------------------------------------------------------------|
| <i>concentration</i><br><i>Index</i> | Index of the current concentration from the std::vector of tokens |
| <i>tokens</i>                        | Vector of tokens                                                  |

Definition at line 258 of file RectangularCsvToInputFilesConverter.cpp.

**7.148.3.3 double RectangularCsvToInputFilesConverter::computeNonScaledConcentration ( const std::string & *concentration* ) [private]**

Compute the non-scaled concentration from the given std::string.

**Parameters**

|                      |                                       |
|----------------------|---------------------------------------|
| <i>concentration</i> | String representing the concentration |
|----------------------|---------------------------------------|

Definition at line 291 of file RectangularCsvToInputFilesConverter.cpp.

**7.148.3.4 double RectangularCsvToInputFilesConverter::computeNormalisedConcentration ( double *concentration* ) [private]**

Normalise the given concentration by dividing it to the maximum concentration.

**Parameters**

|                      |                   |
|----------------------|-------------------|
| <i>concentration</i> | The concentration |
|----------------------|-------------------|

Definition at line 307 of file RectangularCsvToInputFilesConverter.cpp.

**7.148.3.5 double RectangularCsvToInputFilesConverter::computeScaledConcentration ( const std::string & *concentration* ) [private]**

Compute the scaled concentration from the given std::string.

Compute the scaled concentration from the given std::string by applying a logit transformation to it

**Parameters**

|                      |                                       |
|----------------------|---------------------------------------|
| <i>concentration</i> | String representing the concentration |
|----------------------|---------------------------------------|

Definition at line 295 of file RectangularCsvToInputFilesConverter.cpp.

**7.148.3.6 double RectangularCsvToInputFilesConverter::computeSimulationTime ( const std::string & *token* ) [private]**

Compute the simulation time from the given token and check if it is valid.

**Parameters**

|              |                     |
|--------------|---------------------|
| <i>token</i> | Token (std::string) |
|--------------|---------------------|

Definition at line 248 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.148.3.7 void RectangularCsvToInputFilesConverter::convert ( )**

Start the conversion.

Definition at line 48 of file RectangularCsvToInputFilesConverter.cpp.

Referenced by main().

**7.148.3.8 void RectangularCsvToInputFilesConverter::initInputModule ( std::ifstream & *fin* ) [private]**

Initialise the input file stream over the given input file.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 61 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.148.3.9 void RectangularCsvToInputFilesConverter::initIterators ( const NumberIteratorType & *numberIteratorType* ) [private]**

Initialise the iterators considering the given number iterator type.

**Parameters**

|                           |                                 |
|---------------------------|---------------------------------|
| <i>numberIteratorType</i> | The type of the number iterator |
|---------------------------|---------------------------------|

Definition at line 111 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

**7.148.3.10 void RectangularCsvToInputFilesConverter::initMaximumConcentration ( std::ifstream & *fin* ) [private]**

Compute the value of member maximum concentration.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 69 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.148.3.11 void RectangularCsvToInputFilesConverter::initOutputFile ( std::ofstream & *fout*, unsigned int *index*, double & *simulationTime* ) [private]**

Initialise the output file with the given index and simulation time.

**Parameters**

|                       |                          |
|-----------------------|--------------------------|
| <i>fout</i>           | Output file stream       |
| <i>index</i>          | Index of the output file |
| <i>simulationTime</i> | Simulation time          |

Definition at line 93 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

**7.148.3.12 void RectangularCsvToInputFilesConverter::processInputModule ( std::ifstream & *fin* ) [private]**

Process the input file.

Read the concentrations and normalise them if it is the case.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 179 of file RectangularCsvToInputFilesConverter.cpp.

7.148.3.13 void RectangularCsvToInputFilesConverter::processLine ( const std::string & *line*, unsigned int *outputIndex* ) [private]

Process the provided line.

## Parameters

|                    |                                                 |
|--------------------|-------------------------------------------------|
| <i>line</i>        | Line                                            |
| <i>outputIndex</i> | Index integrated in the name of the output file |

Definition at line 194 of file RectangularCsvToInputFilesConverter.cpp.

7.148.3.14 std::vector< double > RectangularCsvToInputFilesConverter::splitLineInConcentrations ( const std::string & *line*, double & *simulationTime* ) [private]

Split the line in concentrations.

## Parameters

|                       |                                          |
|-----------------------|------------------------------------------|
| <i>line</i>           | Line                                     |
| <i>simulationTime</i> | Simulation time associated with the line |

Definition at line 211 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

7.148.3.15 void RectangularCsvToInputFilesConverter::splitLineInConcentrations ( std::vector< double > & *concentrations*, std::vector< std::string > & *tokens*, unsigned int *rowIndex* ) [private]

Split line into concentrations.

## Parameters

|                       |                                      |
|-----------------------|--------------------------------------|
| <i>concentrations</i> | Concentrations extracted from tokens |
| <i>tokens</i>         | Tokens representing the line         |
| <i>rowIndex</i>       | Index of the current row             |

Definition at line 233 of file RectangularCsvToInputFilesConverter.cpp.

7.148.3.16 void RectangularCsvToInputFilesConverter::updateMaximumConcentration ( const std::string & *line*, double & *maximumConcentration* ) [private]

Update the maximum concentration if the values from the given line are greater than it.

## Parameters

|                             |                           |
|-----------------------------|---------------------------|
| <i>line</i>                 | Line from input file      |
| <i>maximumConcentration</i> | The maximum concentration |

Definition at line 311 of file RectangularCsvToInputFilesConverter.cpp.

7.148.3.17 void RectangularCsvToInputFilesConverter::validateInput( std::ifstream & *fin* ) [private]

Validate the input.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 134 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.148.3.18 void RectangularCsvToInputFilesConverter::validateInputLine ( const std::string & *line*, unsigned int *lineNumber* ) [private]**

Validate the provided line identified by a line number.

## Parameters

|                   |                      |
|-------------------|----------------------|
| <i>line</i>       | Line from input file |
| <i>lineNumber</i> | Number of the line   |

Definition at line 158 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw, and multiscale::StringManipulator::split().

**7.148.3.19 void RectangularCsvToInputFilesConverter::validateSelectedConcentrationIndex ( ) [private]**

Validate the selected concentration index in case of more than one concentration for each position.

Definition at line 128 of file RectangularCsvToInputFilesConverter.cpp.

References MS\_throw.

## 7.148.4 Member Data Documentation

**7.148.4.1 NumberIterator\* multiscale::video::RectangularCsvToInputFilesConverter::columnsIterator [private]**

Iterator over the number of columns

Definition at line 42 of file RectangularCsvToInputFilesConverter.hpp.

**7.148.4.2 unsigned int multiscale::video::RectangularCsvToInputFilesConverter::concentrationsIndex [private]**

Index of the current concentration

Definition at line 27 of file RectangularCsvToInputFilesConverter.hpp.

**7.148.4.3 const std::string RectangularCsvToInputFilesConverter::ERR\_INPUT\_OPEN = "The input file could not be opened." [static], [private]**

Definition at line 196 of file RectangularCsvToInputFilesConverter.hpp.

**7.148.4.4 const std::string RectangularCsvToInputFilesConverter::ERR\_INVALID\_VALUE\_LINE = "Invalid value on line: " [static], [private]**

Definition at line 197 of file RectangularCsvToInputFilesConverter.hpp.

**7.148.4.5 const std::string RectangularCsvToInputFilesConverter::ERR\_INVALID\_VALUE\_TOKEN = ", value: " [static], [private]**

Definition at line 198 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.6 `const std::string RectangularCsvToInputFilesConverter::ERR_NEG_CONCENTRATION = "All concentrations must be non-negative." [static], [private]`

Definition at line 192 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.7 `const std::string RectangularCsvToInputFilesConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative." [static], [private]`

Definition at line 195 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.8 `const std::string RectangularCsvToInputFilesConverter::ERR_NR_CONCENTRATIONS = "The number of concentrations in the input file does not match the values of the input parameters height and width." [static], [private]`

Definition at line 194 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.9 `const std::string RectangularCsvToInputFilesConverter::ERR_SELECTED_CONCENTRATION_INDEX = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations." [static], [private]`

Definition at line 193 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.10 `unsigned int multiscale::video::RectangularCsvToInputFilesConverter::height [private]`

Height of the grid

Definition at line 23 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.11 `const std::string RectangularCsvToInputFilesConverter::INPUT_FILE_SEPARATOR = "," [static], [private]`

Definition at line 190 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.12 `std::string multiscale::video::RectangularCsvToInputFilesConverter::inputFilepath [private]`

Path to the input file

Definition at line 20 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.13 `double multiscale::video::RectangularCsvToInputFilesConverter::maximumConcentration [private]`

The maximum concentration in the input file

Definition at line 39 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.14 `unsigned int multiscale::video::RectangularCsvToInputFilesConverter::nrOfConcentrationsForPosition [private]`

Number of concentrations for each position

Definition at line 25 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.15 `const std::string RectangularCsvToInputFilesConverter::OUTPUT_EXTENSION = ".in" [static], [private]`

Definition at line 187 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.16 `const std::string RectangularCsvToInputFilesConverter::OUTPUT_FILE_SEPARATOR = "_" [static], [private]`

Definition at line 189 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.17 `const std::string RectangularCsvToInputFilesConverter::OUTPUT_SEPARATOR = " " [static], [private]`

Definition at line 188 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.18 `std::string multiscale::video::RectangularCsvToInputFilesConverter::outputFilepath [private]`

Path to the output file

Definition at line 21 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.19 `NumberIterator* multiscale::video::RectangularCsvToInputFilesConverter::rowsIterator [private]`

Iterator over the number of rows

Definition at line 41 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.20 `unsigned int multiscale::video::RectangularCsvToInputFilesConverter::selectedConcentrationIndex [private]`

Index of the concentration A in case the number of concentrations for each position is greater than 1

finalConcentration = A / (A1 + A2 + ... + AN), where N is the number of concentrations for each position

Definition at line 29 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.21 `bool multiscale::video::RectangularCsvToInputFilesConverter::useLogScaling [private]`

Flag for using logarithmic scaling for concentrations

Definition at line 36 of file RectangularCsvToInputFilesConverter.hpp.

7.148.4.22 `unsigned int multiscale::video::RectangularCsvToInputFilesConverter::width [private]`

Width of the grid

Definition at line 24 of file RectangularCsvToInputFilesConverter.hpp.

The documentation for this class was generated from the following files:

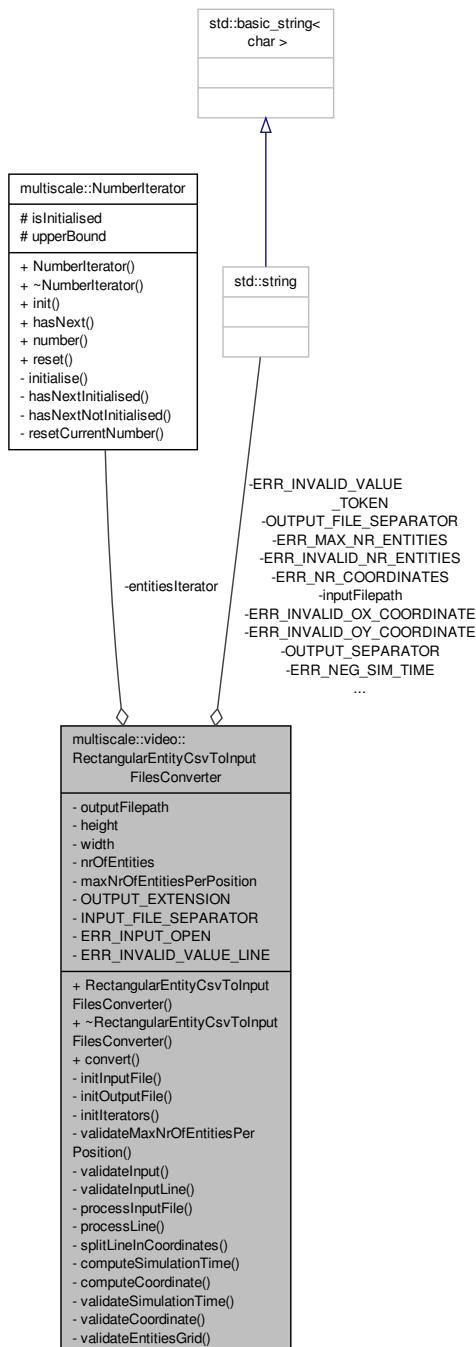
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularCsvToInputFilesConverter.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularCsvToInputFilesConverter.cpp](#)

## 7.149 multiscale::video::RectangularEntityCsvToInputFilesConverter Class Reference

Csv entity file to input file converter considering cartesian coordinates.

```
#include <RectangularEntityCsvToInputFilesConverter.hpp>
```

Collaboration diagram for multiscale::video::RectangularEntityCsvToInputFilesConverter:



## Public Member Functions

- `RectangularEntityCsvToInputFilesConverter` (const std::string &`inputfilepath`, const std::string &`outputfilepath`, unsigned int `height`, unsigned int `width`, unsigned int `nrOfEntities`, unsigned int `maxNrOfEntitiesPerPosition`, NumberIteratorType `numberIteratorType`)
- `~RectangularEntityCsvToInputFilesConverter` ()
- void `convert` ()
 

*Start the conversion.*

## Private Member Functions

- void `initInputFile` (std::ifstream &`fin`)
 

*Initialise the input file stream over the given input file.*
- void `initOutputFile` (std::ofstream &`fout`, unsigned int `index`, double &`simulationTime`)
 

*Initialise the output file with the given index and simulation time.*
- void `initIterators` (const NumberIteratorType &`numberIteratorType`)
 

*Initialise the iterators considering the given number iterator type.*
- void `validateMaxNrOfEntitiesPerPosition` ()
 

*Check if the maximum number of entities per position is a non-zero natural number.*
- void `validateInput` (std::ifstream &`fin`)
 

*Validate the input.*
- void `validateInputLine` (const std::string &`line`, unsigned int `lineNumber`)
 

*Validate the provided line identified by a line number.*
- void `processInputFile` (std::ifstream &`fin`)
 

*Process the input file.*
- void `processLine` (const std::string &`line`, unsigned int `outputIndex`)
 

*Process the provided line.*
- std::vector< double > `splitLineInCoordinates` (const std::string &`line`, double &`simulationTime`)
 

*Split the line in coordinates.*
- double `computeSimulationTime` (const std::string &`token`)
 

*Compute the simulation time from the given token and check if it is valid.*
- unsigned int `computeCoordinate` (const std::string &`token`, bool `isOxCoordinate`)
 

*Compute the coordinate from the given std::string and check if it is valid.*
- void `validateSimulationTime` (const std::string &`token`, unsigned int `lineNumber`)
 

*Check if the simulation time is valid.*
- void `validateCoordinate` (const std::string &`token`, unsigned int `lineNumber`, bool `isOxCoordinate`)
 

*Check if the coordinate is valid.*
- void `validateEntitiesGrid` (const std::vector< double > &`entitiesGrid`)
 

*Check if the entities grid contains only values between zero and one.*

## Private Attributes

- std::string `inputfilepath`
- std::string `outputfilepath`
- unsigned int `height`
- unsigned int `width`
- unsigned int `nrOfEntities`
- unsigned int `maxNrOfEntitiesPerPosition`
- NumberIterator \* `entitiesIterator`

## Static Private Attributes

- static const std::string **OUTPUT\_EXTENSION** = ".in"
- static const std::string **OUTPUT\_SEPARATOR** = " "
- static const std::string **OUTPUT\_FILE\_SEPARATOR** = "\_"
- static const std::string **INPUT\_FILE\_SEPARATOR** = ","
- static const std::string **ERR\_INVALID\_NR\_ENTITIES** = "The number of entities at the given position is invalid."
- static const std::string **ERR\_INVALID\_OX\_COORDINATE** = "The value of the Ox coordinate is invalid."
- static const std::string **ERR\_INVALID\_OY\_COORDINATE** = "The value of the Oy coordinate is invalid."
- static const std::string **ERR\_MAX\_NR\_ENTITIES** = "The maximum number of entities per grid position is equal to zero."
- static const std::string **ERR\_NR\_COORDINATES** = "The number of coordinates in the input file does not match the values of the input parameters `height`, `width` and `nrOfEntities`."
- static const std::string **ERR\_NEG\_SIM\_TIME** = "The simulation time must be non-negative."
- static const std::string **ERR\_INPUT\_OPEN** = "The input file could not be opened."
- static const std::string **ERR\_INVALID\_VALUE\_LINE** = "Invalid value on line: "
- static const std::string **ERR\_INVALID\_VALUE\_TOKEN** = ", value: "

### 7.149.1 Detailed Description

Csv entity file to input file converter considering cartesian coordinates.

Definition at line 16 of file `RectangularEntityCsvToInputFilesConverter.hpp`.

### 7.149.2 Constructor & Destructor Documentation

7.149.2.1 `RectangularEntityCsvToInputFilesConverter::RectangularEntityCsvToInputFilesConverter ( const std::string & inputFilepath, const std::string & outputFilepath, unsigned int height, unsigned int width, unsigned int nrOfEntities, unsigned int maxNrOfEntitiesPerPosition, NumberIteratorType numberIteratorType )`

Definition at line 19 of file `RectangularEntityCsvToInputFilesConverter.cpp`.

7.149.2.2 `RectangularEntityCsvToInputFilesConverter::~RectangularEntityCsvToInputFilesConverter ( )`

Definition at line 36 of file `RectangularEntityCsvToInputFilesConverter.cpp`.

### 7.149.3 Member Function Documentation

7.149.3.1 `unsigned int RectangularEntityCsvToInputFilesConverter::computeCoordinate ( const std::string & token, bool isOxCoordinate ) [private]`

Compute the coordinate from the given std::string and check if it is valid.

#### Parameters

|                       |                                                                      |
|-----------------------|----------------------------------------------------------------------|
| <i>token</i>          | Token (std::string)                                                  |
| <i>isOxCoordinate</i> | Flag which indicates if the coordinate corresponds to Ox axis or not |

Definition at line 210 of file `RectangularEntityCsvToInputFilesConverter.cpp`.

References `MS_throw`.

7.149.3.2 double RectangularEntityCsvToInputFilesConverter::computeSimulationTime ( const std::string & *token* )  
[private]

Compute the simulation time from the given token and check if it is valid.

**Parameters**

|              |                     |
|--------------|---------------------|
| <i>token</i> | Token (std::string) |
|--------------|---------------------|

Definition at line 200 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.149.3.3 void RectangularEntityCsvToInputFilesConverter::convert ( )**

Start the conversion.

Definition at line 40 of file RectangularEntityCsvToInputFilesConverter.cpp.

Referenced by main().

**7.149.3.4 void RectangularEntityCsvToInputFilesConverter::initInputFile ( std::ifstream & *fin* ) [private]**

Initialise the input file stream over the given input file.

**Parameters**

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 52 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.149.3.5 void RectangularEntityCsvToInputFilesConverter::initIterators ( const NumberIteratorType & *numberIteratorType* ) [private]**

Initialise the iterators considering the given number iterator type.

**Parameters**

|                           |                                 |
|---------------------------|---------------------------------|
| <i>numberIteratorType</i> | The type of the number iterator |
|---------------------------|---------------------------------|

Definition at line 78 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

**7.149.3.6 void RectangularEntityCsvToInputFilesConverter::initOutputFile ( std::ofstream & *fout*, unsigned int *index*, double & *simulationTime* ) [private]**

Initialise the output file with the given index and simulation time.

**Parameters**

|                       |                          |
|-----------------------|--------------------------|
| <i>fout</i>           | Output file stream       |
| <i>index</i>          | Index of the output file |
| <i>simulationTime</i> | Simulation time          |

Definition at line 60 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

**7.149.3.7 void RectangularEntityCsvToInputFilesConverter::processInputFile ( std::ifstream & *fin* ) [private]**

Process the input file.

Read the concentrations and normalise them if it is the case.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 142 of file RectangularEntityCsvToInputFilesConverter.cpp.

**7.149.3.8 void RectangularEntityCsvToInputFilesConverter::processLine ( const std::string & *line*, unsigned int *outputIndex* ) [private]**

Process the provided line.

## Parameters

|                    |                                                 |
|--------------------|-------------------------------------------------|
| <i>line</i>        | Line                                            |
| <i>outputIndex</i> | Index integrated in the name of the output file |

Definition at line 157 of file RectangularEntityCsvToInputFilesConverter.cpp.

**7.149.3.9 std::vector< double > RectangularEntityCsvToInputFilesConverter::splitLineInCoordinates ( const std::string & *line*, double & *simulationTime* ) [private]**

Split the line in coordinates.

Split the line in coordinates and return the grid of size height \* width recording the position of the entities. The number of entities per grid position is normalised to the range [0, 1]

## Parameters

|                       |                                          |
|-----------------------|------------------------------------------|
| <i>line</i>           | Line                                     |
| <i>simulationTime</i> | Simulation time associated with the line |

Definition at line 176 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

**7.149.3.10 void RectangularEntityCsvToInputFilesConverter::validateCoordinate ( const std::string & *token*, unsigned int *lineNumber*, bool *isOxCoordinate* ) [private]**

Check if the coordinate is valid.

## Parameters

|                       |                                                                      |
|-----------------------|----------------------------------------------------------------------|
| <i>token</i>          | Token (std::string)                                                  |
| <i>lineNumber</i>     | Number of the line                                                   |
| <i>isOxCoordinate</i> | Flag which indicates if the coordinate corresponds to Ox axis or not |

Definition at line 241 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.149.3.11 void RectangularEntityCsvToInputFilesConverter::validateEntitiesGrid ( const std::vector< double > & *entitiesGrid* ) [private]**

Check if the entities grid contains only values between zero and one.

## Parameters

|                     |                      |
|---------------------|----------------------|
| <i>entitiesGrid</i> | The grid of entities |
|---------------------|----------------------|

Definition at line 257 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

7.149.3.12 void RectangularEntityCsvToInputFilesConverter::validateInput ( std::ifstream & *fin* ) [private]

Validate the input.

## Parameters

|            |                   |
|------------|-------------------|
| <i>fin</i> | Input file stream |
|------------|-------------------|

Definition at line 100 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.149.3.13 void RectangularEntityCsvToInputFilesConverter::validateInputLine ( const std::string & *line*, unsigned int *lineNumber* ) [private]**

Validate the provided line identified by a line number.

## Parameters

|                   |                      |
|-------------------|----------------------|
| <i>line</i>       | Line from input file |
| <i>lineNumber</i> | Number of the line   |

Definition at line 124 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw, and multiscale::StringManipulator::split().

**7.149.3.14 void RectangularEntityCsvToInputFilesConverter::validateMaxNrOfEntitiesPerPosition ( ) [private]**

Check if the maximum number of entities per position is a non-zero natural number.

Definition at line 94 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

**7.149.3.15 void RectangularEntityCsvToInputFilesConverter::validateSimulationTime ( const std::string & *token*, unsigned int *lineNumber* ) [private]**

Check if the simulation time is valid.

## Parameters

|                   |                     |
|-------------------|---------------------|
| <i>token</i>      | Token (std::string) |
| <i>lineNumber</i> | Number of the line  |

Definition at line 227 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS\_throw.

## 7.149.4 Member Data Documentation

**7.149.4.1 NumberIterator\* multiscale::video::RectangularEntityCsvToInputFilesConverter::entitiesIterator [private]**

Iterator over the number of rows

Definition at line 29 of file RectangularEntityCsvToInputFilesConverter.hpp.

**7.149.4.2 const std::string RectangularEntityCsvToInputFilesConverter::ERR\_INPUT\_OPEN = "The input file could not be opened." [static], [private]**

Definition at line 156 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.3 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_NR_ENTITIES = "The number of entities at the given position is invalid."` [static], [private]

Definition at line 150 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.4 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_OX_COORDINATE = "The value of the Ox coordinate is invalid."` [static], [private]

Definition at line 151 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.5 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_OY_COORDINATE = "The value of the Oy coordinate is invalid."` [static], [private]

Definition at line 152 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.6 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_VALUE_LINE = "Invalid value on line: "` [static], [private]

Definition at line 157 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.7 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_VALUE_TOKEN = ", value: "` [static], [private]

Definition at line 158 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.8 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_MAX_NR_ENTITIES = "The maximum number of entities per grid position is equal to zero."` [static], [private]

Definition at line 153 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.9 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative."` [static], [private]

Definition at line 155 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.10 `const std::string RectangularEntityCsvToInputFilesConverter::ERR_NR_COORDINATES = "The number of coordinates in the input file does not match the values of the input parameters height, width and nrOfEntities."` [static], [private]

Definition at line 154 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.11 `unsigned int multiscale::video::RectangularEntityCsvToInputFilesConverter::height` [private]

Height of the grid

Definition at line 23 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.12 `const std::string RectangularEntityCsvToInputFilesConverter::INPUT_FILE_SEPARATOR = ","` [static], [private]

Definition at line 148 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.13 `std::string multiscale::video::RectangularEntityCsvToInputFilesConverter::inputFilepath` [private]

Path to the input file

Definition at line 20 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.14 `unsigned int multiscale::video::RectangularEntityCsvToInputFilesConverter::maxNrOfEntitiesPerPosition` [private]

The maximum number of entities per position

Definition at line 27 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.15 `unsigned int multiscale::video::RectangularEntityCsvToInputFilesConverter::nrOfEntities` [private]

Number of entities

Definition at line 25 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.16 `const std::string RectangularEntityCsvToInputFilesConverter::OUTPUT_EXTENSION = ".in"` [static], [private]

Definition at line 145 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.17 `const std::string RectangularEntityCsvToInputFilesConverter::OUTPUT_FILE_SEPARATOR = "_"` [static], [private]

Definition at line 147 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.18 `const std::string RectangularEntityCsvToInputFilesConverter::OUTPUT_SEPARATOR = " "` [static], [private]

Definition at line 146 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.19 `std::string multiscale::video::RectangularEntityCsvToInputFilesConverter::outputFilepath` [private]

Path to the output file

Definition at line 21 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.149.4.20 `unsigned int multiscale::video::RectangularEntityCsvToInputFilesConverter::width` [private]

Width of the grid

Definition at line 24 of file RectangularEntityCsvToInputFilesConverter.hpp.

The documentation for this class was generated from the following files:

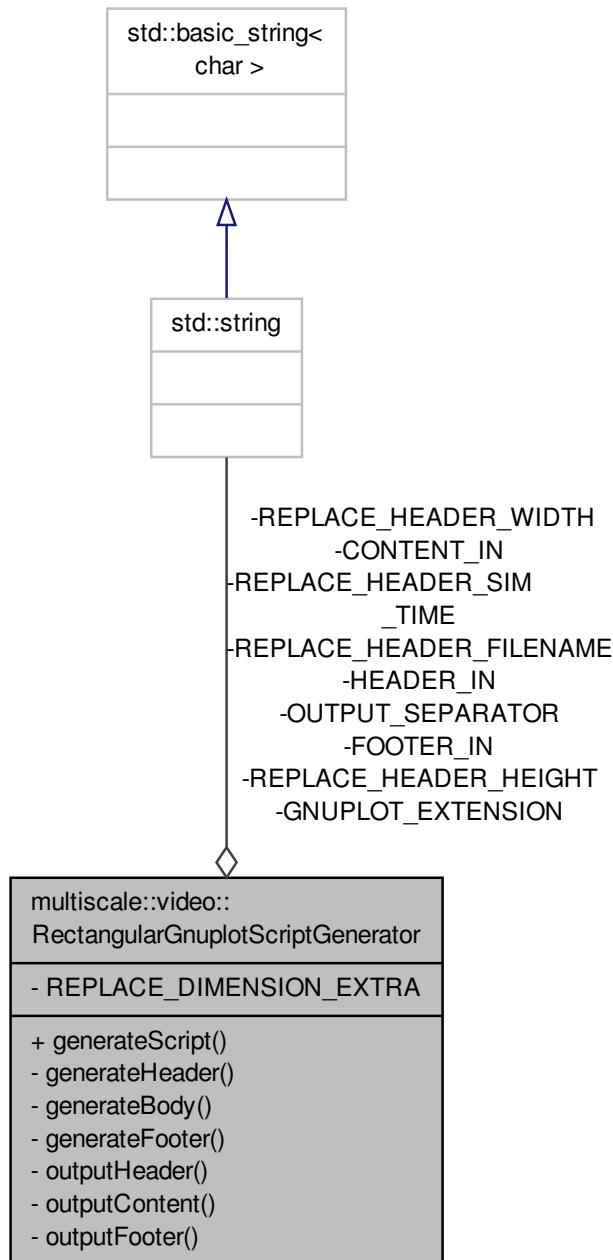
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/[RectangularEntityCsvToInputFilesConverter.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/[RectangularEntityCsvToInputFilesConverter.cpp](#)

## 7.150 multiscale::video::RectangularGnuplotScriptGenerator Class Reference

Gnuplot script generator from the provided concentrations considering a rectangular geometry.

```
#include <RectangularGnuplotScriptGenerator.hpp>
```

Collaboration diagram for multiscale::video::RectangularGnuplotScriptGenerator:



## Static Public Member Functions

- static void `generateScript` (const std::vector< double > &concentrations, double simulationTime, unsigned long height, unsigned long width, const std::string &outputfilepath)
   
*Generate the script.*

## Static Private Member Functions

- static void `generateHeader` (std::ofstream &fout, const std::string &outputfilepath, double simulationTime, unsigned long height, unsigned long width)
   
*Generate the header of the script.*
- static void `generateBody` (const std::vector< double > &concentrations, unsigned long height, unsigned long width, std::ofstream &fout)
   
*Generate the body/content of the script.*
- static void `generateFooter` (std::ofstream &fout)
   
*Generate the footer of the script.*
- static void `outputHeader` (std::ifstream &fin, const std::string &outputfilename, double simulationTime, unsigned long height, unsigned long width, std::ofstream &fout)
   
*Output the header of the script.*
- static void `outputContent` (const std::vector< double > &concentrations, unsigned long height, unsigned long width, std::ofstream &fout)
   
*Output the content of the script.*
- static void `outputFooter` (std::ifstream &fin, std::ofstream &fout)
   
*Output the footer of the script.*

## Static Private Attributes

- static const std::string `HEADER_IN` = "/usr/local/share/mule/config/video/rectangular/header.in"
- static const std::string `CONTENT_IN` = "/usr/local/share/mule/config/video/rectangular/content.in"
- static const std::string `FOOTER_IN` = "/usr/local/share/mule/config/video/rectangular/footer.in"
- static const std::string `REPLACE_HEADER_FILENAME` = "OUTPUT\_FILENAME"
- static const std::string `REPLACE_HEADER_HEIGHT` = "OUTPUT\_DIMENSION1"
- static const std::string `REPLACE_HEADER_WIDTH` = "OUTPUT\_DIMENSION2"
- static const std::string `REPLACE_HEADER_SIM_TIME` = "OUTPUT\_SIM\_TIME"
- static const double `REPLACE_DIMENSION_EXTRA` = 0.5
- static const std::string `OUTPUT_SEPARATOR` = " "
- static const std::string `GNUPLOT_EXTENSION` = ".plt"

### 7.150.1 Detailed Description

Gnuplot script generator from the provided concentrations considering a rectangular geometry.

Definition at line 13 of file RectangularGnuplotScriptGenerator.hpp.

### 7.150.2 Member Function Documentation

#### 7.150.2.1 void RectangularGnuplotScriptGenerator::generateBody ( const std::vector< double > & `concentrations`, unsigned long `height`, unsigned long `width`, std::ofstream & `fout` ) [static], [private]

Generate the body/content of the script.

**Parameters**

|                       |                        |
|-----------------------|------------------------|
| <i>concentrations</i> | The concentrations     |
| <i>height</i>         | The height of the grid |
| <i>width</i>          | The width of the grid  |
| <i>fout</i>           | Output file stream     |

Definition at line 43 of file RectangularGnuplotScriptGenerator.cpp.

References CONTENT\_IN, and outputContent().

Referenced by generateScript().

#### 7.150.2.2 void RectangularGnuplotScriptGenerator::generateFooter ( std::ofstream & *fout* ) [static], [private]

Generate the footer of the script.

**Parameters**

|             |                    |
|-------------|--------------------|
| <i>fout</i> | Output file stream |
|-------------|--------------------|

Definition at line 55 of file RectangularGnuplotScriptGenerator.cpp.

References FOOTER\_IN, and outputFooter().

Referenced by generateScript().

#### 7.150.2.3 void RectangularGnuplotScriptGenerator::generateHeader ( std::ofstream & *fout*, const std::string & *outputfilepath*, double *simulationTime*, unsigned long *height*, unsigned long *width* ) [static], [private]

Generate the header of the script.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>fout</i>           | Output file stream      |
| <i>outputfilepath</i> | Path to the output file |
| <i>simulationTime</i> | Simulation time         |
| <i>height</i>         | Height of the grid      |
| <i>width</i>          | Width of the grid       |

Definition at line 29 of file RectangularGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::filenameFromPath(), HEADER\_IN, and outputHeader().

Referenced by generateScript().

#### 7.150.2.4 void RectangularGnuplotScriptGenerator::generateScript ( const std::vector< double > & *concentrations*, double *simulationTime*, unsigned long *height*, unsigned long *width*, const std::string & *outputfilepath* ) [static]

Generate the script.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>concentrations</i> | Concentrations          |
| <i>simulationTime</i> | Simulation time         |
| <i>height</i>         | Height of the grid      |
| <i>width</i>          | Width of the grid       |
| <i>outputfilepath</i> | Path of the output file |

Definition at line 13 of file RectangularGnuplotScriptGenerator.cpp.

References generateBody(), generateFooter(), generateHeader(), and GNUPLOT\_EXTENSION.

Referenced by multiscale::video::CartesianToConcentrationsConverter::outputResults().

7.150.2.5 void RectangularGnuplotScriptGenerator::outputContent ( const std::vector< double > & *concentrations*, unsigned long *height*, unsigned long *width*, std::ofstream & *fout* ) [static], [private]

Output the content of the script.

#### Parameters

|                       |                        |
|-----------------------|------------------------|
| <i>concentrations</i> | The concentrations     |
| <i>height</i>         | The height of the grid |
| <i>width</i>          | The width of the grid  |
| <i>fout</i>           | Output file stream     |

Definition at line 100 of file RectangularGnuplotScriptGenerator.cpp.

References OUTPUT\_SEPARATOR.

Referenced by generateBody().

7.150.2.6 void RectangularGnuplotScriptGenerator::outputFooter ( std::ifstream & *fin*, std::ofstream & *fout* ) [static], [private]

Output the footer of the script.

#### Parameters

|             |                    |
|-------------|--------------------|
| <i>fin</i>  | Input file stream  |
| <i>fout</i> | Output file stream |

Definition at line 114 of file RectangularGnuplotScriptGenerator.cpp.

Referenced by generateFooter().

7.150.2.7 void RectangularGnuplotScriptGenerator::outputHeader ( std::ifstream & *fin*, const std::string & *outputFilename*, double *simulationTime*, unsigned long *height*, unsigned long *width*, std::ofstream & *fout* ) [static], [private]

Output the header of the script.

#### Parameters

|                       |                         |
|-----------------------|-------------------------|
| <i>fin</i>            | Input file stream       |
| <i>outputFilename</i> | Name of the output file |
| <i>simulationTime</i> | Simulation time         |
| <i>height</i>         | The height of the grid  |
| <i>width</i>          | The width of the grid   |
| <i>fout</i>           | Output file stream      |

Definition at line 65 of file RectangularGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::replace(), REPLACE\_DIMENSION\_EXTRA, REPLACE\_HEADER\_FI←LENAME, REPLACE\_HEADER\_HEIGHT, REPLACE\_HEADER\_SIM\_TIME, and REPLACE\_HEADER\_WIDTH.

Referenced by generateHeader().

## 7.150.3 Member Data Documentation

7.150.3.1 const std::string RectangularGnuplotScriptGenerator::CONTENT\_IN =  
 "/usr/local/share/mule/config/video/rectangular/content.in" [static], [private]

Definition at line 102 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by generateBody().

```
7.150.3.2 const std::string RectangularGnuplotScriptGenerator::FOOTER_IN =
 "/usr/local/share/mule/config/video/rectangular/footer.in" [static], [private]
```

Definition at line 103 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by generateFooter().

```
7.150.3.3 const std::string RectangularGnuplotScriptGenerator::GNUPLOT_EXTENSION = ".plt" [static], [private]
```

Definition at line 114 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by generateScript().

```
7.150.3.4 const std::string RectangularGnuplotScriptGenerator::HEADER_IN =
 "/usr/local/share/mule/config/video/rectangular/header.in" [static], [private]
```

Definition at line 101 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by generateHeader().

```
7.150.3.5 const std::string RectangularGnuplotScriptGenerator::OUTPUT_SEPARATOR = " " [static], [private]
```

Definition at line 112 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputContent().

```
7.150.3.6 const double RectangularGnuplotScriptGenerator::REPLACE_DIMENSION_EXTRA = 0.5 [static],
 [private]
```

Definition at line 110 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

```
7.150.3.7 const std::string RectangularGnuplotScriptGenerator::REPLACE_HEADER_FILENAME = "OUTPUT_FILENAME"
 [static], [private]
```

Definition at line 105 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

```
7.150.3.8 const std::string RectangularGnuplotScriptGenerator::REPLACE_HEADER_HEIGHT = "OUTPUT_DIMENSION1"
 [static], [private]
```

Definition at line 106 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

```
7.150.3.9 const std::string RectangularGnuplotScriptGenerator::REPLACE_HEADER_SIM_TIME = "OUTPUT_SIM_TIME"
 [static], [private]
```

Definition at line 108 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

```
7.150.3.10 const std::string RectangularGnuplotScriptGenerator::REPLACE_HEADER_WIDTH = "OUTPUT_DIMENSION2"
[static], [private]
```

Definition at line 107 of file RectangularGnuplotScriptGenerator.hpp.

Referenced by outputHeader().

The documentation for this class was generated from the following files:

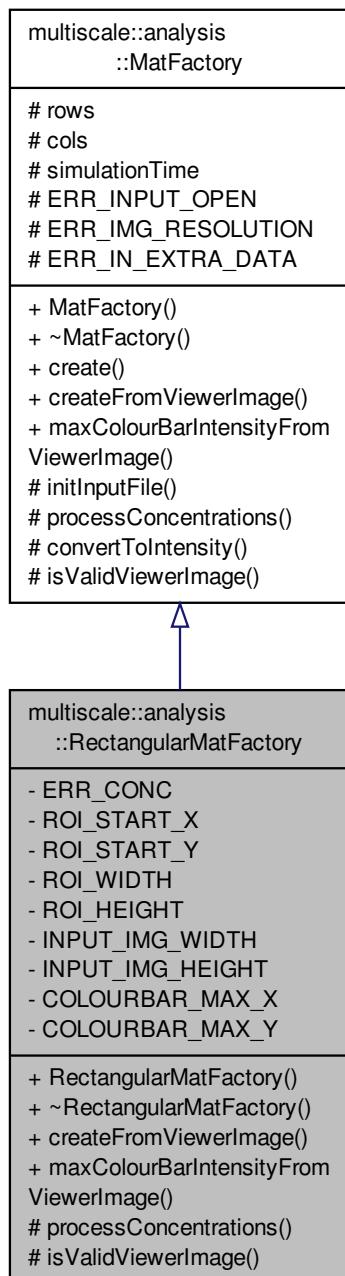
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/[RectangularGnuplotScriptGenerator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/[RectangularGnuplotScriptGenerator.cpp](#)

## 7.151 multiscale::analysis::RectangularMatFactory Class Reference

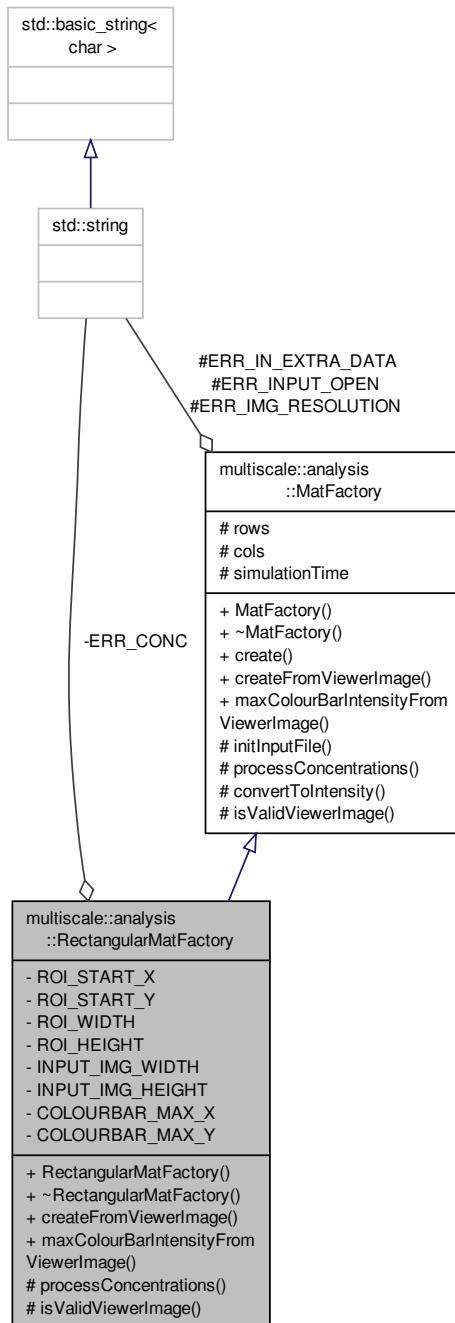
Class for creating a cv::Mat object considering a rectangular grid.

```
#include <RectangularMatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::RectangularMatFactory:



Collaboration diagram for multiscale::analysis::RectangularMatFactory:



## Public Member Functions

- `RectangularMatFactory ()`
- `~RectangularMatFactory ()`
- `cv::Mat createFromViewerImage (const std::string &inputFile) override`

*Create a cv::Mat object from the image file obtained from the RectangularGeometryViewer.*

- `double maxColourBarIntensityFromViewerImage (const std::string &inputFile) override`

*Get the maximum grayscale intensity of the colour bar in the image.*

## Protected Member Functions

- `unsigned char * processConcentrations (std::ifstream &fin) override`  
*Process the concentrations from the input file.*
- `bool isValidViewerImage (const cv::Mat &image) override`  
*Check if the image generated by the viewer has the required resolution.*

## Static Private Attributes

- `static const std::string ERR_CONC = "All concentrations have to be between 0 and 1."`
- `static const int ROI_START_X = 321`
- `static const int ROI_START_Y = 318`
- `static const int ROI_WIDTH = 1407`
- `static const int ROI_HEIGHT = 1358`
- `static const int INPUT_IMG_WIDTH = 2048`
- `static const int INPUT_IMG_HEIGHT = 2048`
- `static const int COLOURBAR_MAX_X = 1799`
- `static const int COLOURBAR_MAX_Y = 320`

## Additional Inherited Members

### 7.151.1 Detailed Description

Class for creating a cv::Mat object considering a rectangular grid.

Definition at line 12 of file RectangularMatFactory.hpp.

### 7.151.2 Constructor & Destructor Documentation

#### 7.151.2.1 RectangularMatFactory::RectangularMatFactory ( )

Definition at line 9 of file RectangularMatFactory.cpp.

#### 7.151.2.2 RectangularMatFactory::~RectangularMatFactory ( )

Definition at line 11 of file RectangularMatFactory.cpp.

### 7.151.3 Member Function Documentation

#### 7.151.3.1 cv::Mat RectangularMatFactory::createFromViewerImage ( const std::string & inputFile ) [override], [virtual]

Create a cv::Mat object from the image file obtained from the RectangularGeometryViewer.

Create the cv::Mat instance from the given image file

#### Parameters

---

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 13 of file RectangularMatFactory.cpp.

References [isValidViewerImage\(\)](#), ROI\_HEIGHT, ROI\_START\_X, ROI\_START\_Y, and ROI\_WIDTH.

Referenced by [main\(\)](#).

### 7.151.3.2 bool RectangularMatFactory::isValidViewerImage ( const cv::Mat & *image* ) [override], [protected], [virtual]

Check if the image generated by the viewer has the required resolution.

Parameters

|              |                               |
|--------------|-------------------------------|
| <i>image</i> | Image generated by the viewer |
|--------------|-------------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 47 of file RectangularMatFactory.cpp.

References [multiscale::analysis::MatFactory::ERR\\_IMG\\_RESOLUTION](#), [multiscale::analysis::MatFactory::ERR\\_INPUT\\_OPEN](#), [INPUT\\_IMG\\_HEIGHT](#), [INPUT\\_IMG\\_WIDTH](#), and [MS\\_throw](#).

Referenced by [createFromViewerImage\(\)](#), and [maxColourBarIntensityFromViewerImage\(\)](#).

### 7.151.3.3 double RectangularMatFactory::maxColourBarIntensityFromViewerImage ( const std::string & *inputFile* ) [override], [virtual]

Get the maximum grayscale intensity of the colour bar in the image.

Parameters

|                  |                            |
|------------------|----------------------------|
| <i>inputFile</i> | The path to the image file |
|------------------|----------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 21 of file RectangularMatFactory.cpp.

References [COLOURBAR\\_MAX\\_X](#), [COLOURBAR\\_MAX\\_Y](#), and [isValidViewerImage\(\)](#).

Referenced by [main\(\)](#).

### 7.151.3.4 unsigned char \* RectangularMatFactory::processConcentrations ( std::ifstream & *fin* ) [override], [protected], [virtual]

Process the concentrations from the input file.

Read the concentrations from the input file and return them as an array which can be used afterwards to create a cv::Mat object from them

REMARK: The constructor of cv::Mat does not copy the data. Therefore, DO NOT deallocate it in this class.

Parameters

|            |                                                          |
|------------|----------------------------------------------------------|
| <i>fin</i> | Input file stream from which the concentrations are read |
|------------|----------------------------------------------------------|

Implements [multiscale::analysis::MatFactory](#).

Definition at line 29 of file RectangularMatFactory.cpp.

References [multiscale::analysis::MatFactory::cols](#), [multiscale::analysis::MatFactory::convertToIntensity\(\)](#), [ERR\\_C\\_ONC](#), [MS\\_throw](#), and [multiscale::analysis::MatFactory::rows](#).

#### 7.151.4 Member Data Documentation

7.151.4.1 `const int RectangularMatFactory::COLOURBAR_MAX_X = 1799 [static], [private]`

Definition at line 66 of file RectangularMatFactory.hpp.

Referenced by `maxColourBarIntensityFromViewerImage()`.

7.151.4.2 `const int RectangularMatFactory::COLOURBAR_MAX_Y = 320 [static], [private]`

Definition at line 67 of file RectangularMatFactory.hpp.

Referenced by `maxColourBarIntensityFromViewerImage()`.

7.151.4.3 `const std::string RectangularMatFactory::ERR_CONC = "All concentrations have to be between 0 and 1." [static], [private]`

Definition at line 56 of file RectangularMatFactory.hpp.

Referenced by `processConcentrations()`.

7.151.4.4 `const int RectangularMatFactory::INPUT_IMG_HEIGHT = 2048 [static], [private]`

Definition at line 64 of file RectangularMatFactory.hpp.

Referenced by `isValidViewerImage()`.

7.151.4.5 `const int RectangularMatFactory::INPUT_IMG_WIDTH = 2048 [static], [private]`

Definition at line 63 of file RectangularMatFactory.hpp.

Referenced by `isValidViewerImage()`.

7.151.4.6 `const int RectangularMatFactory::ROI_HEIGHT = 1358 [static], [private]`

Definition at line 61 of file RectangularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.151.4.7 `const int RectangularMatFactory::ROI_START_X = 321 [static], [private]`

Definition at line 58 of file RectangularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.151.4.8 `const int RectangularMatFactory::ROI_START_Y = 318 [static], [private]`

Definition at line 59 of file RectangularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.151.4.9 `const int RectangularMatFactory::ROI_WIDTH = 1407 [static], [private]`

Definition at line 60 of file RectangularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

The documentation for this class was generated from the following files:

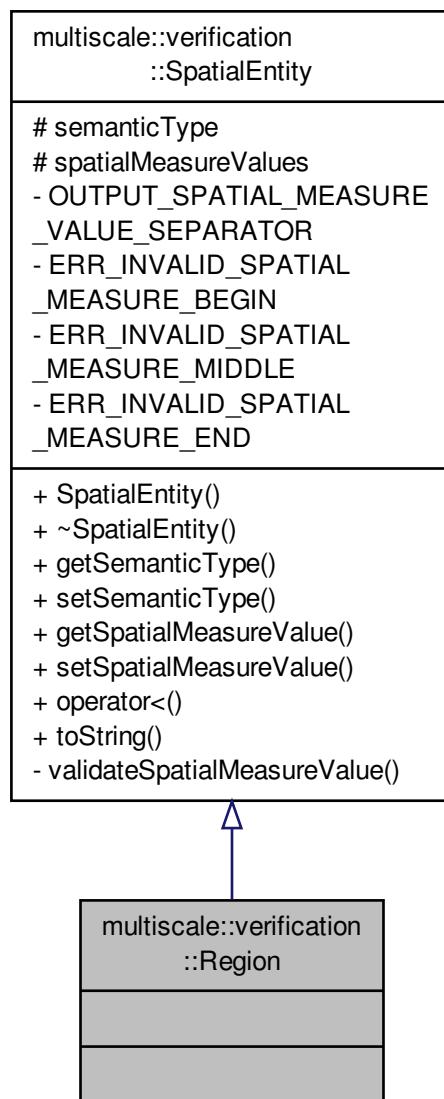
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/[RectangularMatFactory.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/[RectangularMatFactory.cpp](#)

## 7.152 multiscale::verification::Region Class Reference

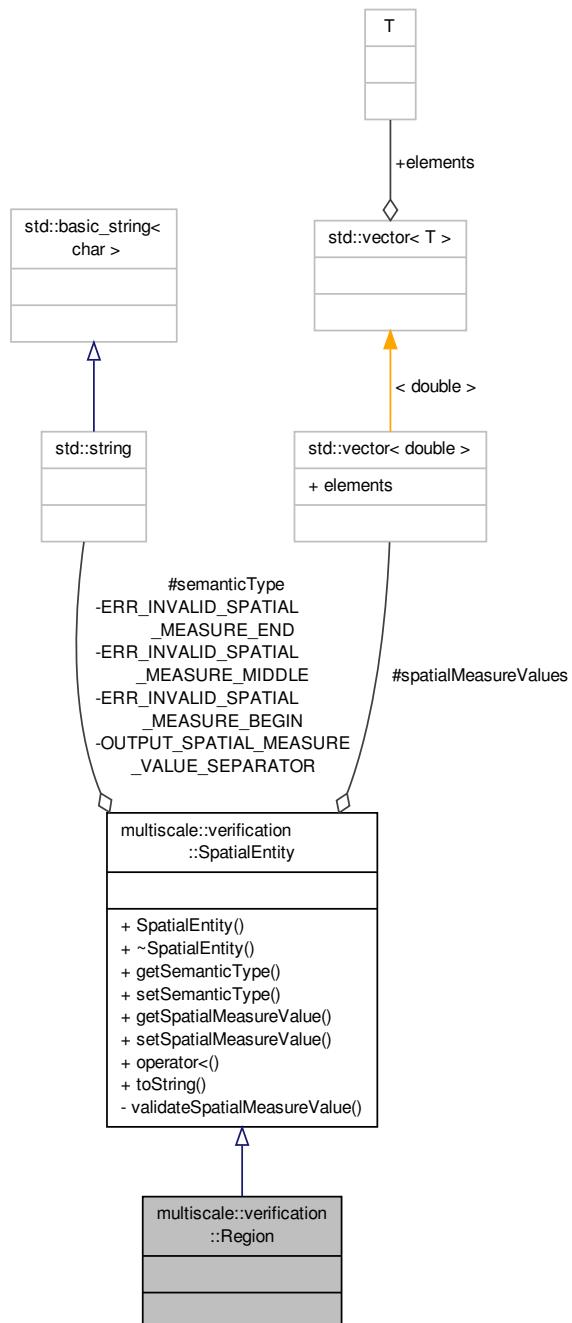
Class for representing a region.

```
#include <Region.hpp>
```

Inheritance diagram for multiscale::verification::Region:



Collaboration diagram for multiscale::verification::Region:



## Additional Inherited Members

### 7.152.1 Detailed Description

Class for representing a region.

Definition at line 21 of file Region.hpp.

The documentation for this class was generated from the following file:

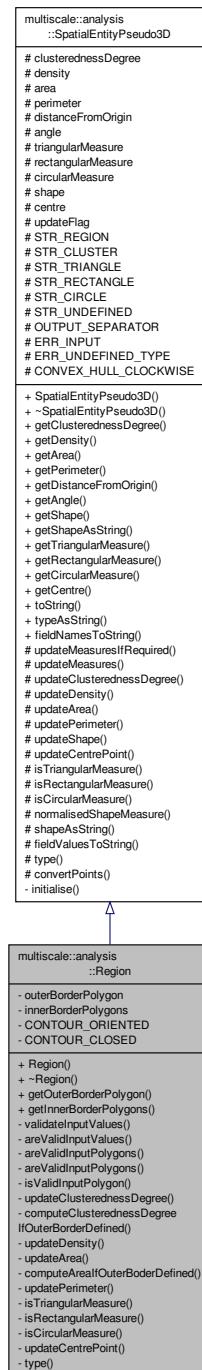
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Region.hpp

## 7.153 multiscale::analysis::Region Class Reference

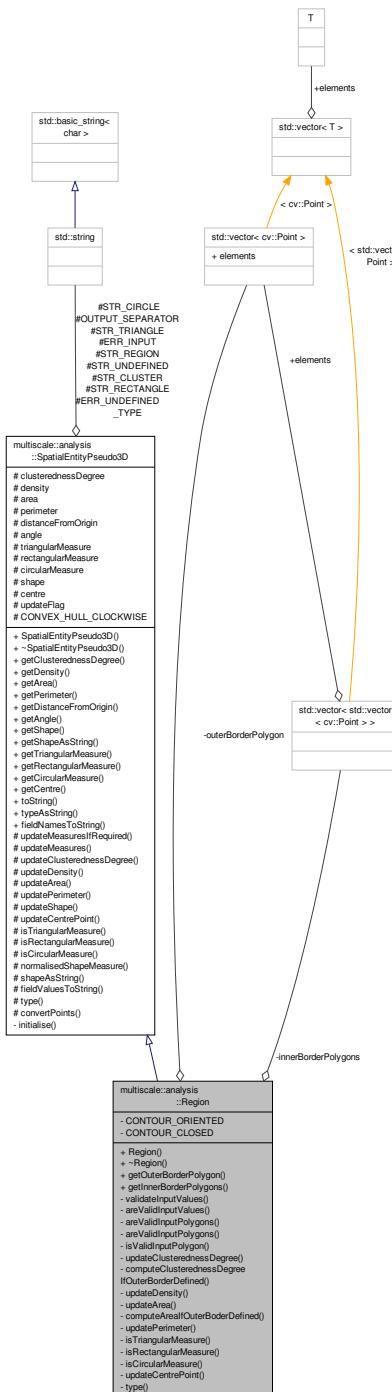
Class for representing a region.

```
#include <Region.hpp>
```

Inheritance diagram for multiscale::analysis::Region:



Collaboration diagram for multiscale::analysis::Region:



## Public Member Functions

- `Region (double density, double distanceFromOrigin, double angleWrtOrigin, const std::vector<cv::Point> &outerBorderPolygon, const std::vector<std::vector<cv::Point>> &innerBorderPolygons)`
- `~Region ()`
- `const std::vector<cv::Point> & getOuterBorderPolygon () const`

*Get the polygon defining the outer border of the region.*

- const std::vector< std::vector< cv::Point > > & **getInnerBorderPolygons** () const  
*Get the polygons defining the inner borders of the region.*

## Private Member Functions

- void **validateInputValues** (double **density**, double **distanceFromOrigin**, double **angleWrtOrigin**, const std::vector< cv::Point > &**outerBorderPolygon**, const std::vector< std::vector< cv::Point > > &**innerBorderPolygons**)  
*Validate the input values.*
- bool **areValidInputValues** (double **density**, double **distanceFromOrigin**, double **angleWrtOrigin**, const std::vector< cv::Point > &**outerBorderPolygon**, const std::vector< std::vector< cv::Point > > &**innerBorderPolygons**)  
*Check if the input values are valid or not.*
- bool **areValidInputPolygons** (const std::vector< cv::Point > &**outerBorderPolygon**, const std::vector< std::vector< cv::Point > > &**innerBorderPolygons**)  
*Check if the given input outer/inner border polygons are valid.*
- bool **areValidInputPolygons** (const std::vector< std::vector< cv::Point > > &**polygons**)  
*Check if the given input polygons are valid.*
- bool **isValidInputPolygon** (const std::vector< cv::Point > &**polygon**)  
*Check if the given input polygons are valid.*
- void **updateClusterednessDegree** () override  
*Update the value of the clusteredness degree.*
- double **computeClusterednessDegreeIfOuterBorderDefined** ()  
*Compute the value of the clusteredness degree if the outer border of the region is defined.*
- void **updateDensity** () override  
*Update the value of the density.*
- void **updateArea** () override  
*Update the area.*
- double **computeAreaIfOuterBoderDefined** ()  
*Compute the value of the area if the outer border of the region is defined.*
- void **updatePerimeter** () override  
*Update the perimeter.*
- double **isTriangularMeasure** () override  
*Get the measure that the cluster has a triangular shape.*
- double **isRectangularMeasure** () override  
*Get the measure that the cluster has a rectangular shape.*
- double **isCircularMeasure** () override  
*Get the measure that the cluster has a circular shape.*
- void **updateCentrePoint** () override  
*Update the centre of the region.*
- **SpatialEntityPseudo3DType type** () override  
*Return the type of the pseudo 3D spatial entity.*

## Private Attributes

- std::vector< cv::Point > **outerBorderPolygon**
- std::vector< std::vector< cv::Point > > **innerBorderPolygons**

## Static Private Attributes

- static const bool **CONTOUR\_ORIENTED** = false
- static const bool **CONTOUR\_CLOSED** = true

## Additional Inherited Members

### 7.153.1 Detailed Description

Class for representing a region.

Definition at line 16 of file Region.hpp.

### 7.153.2 Constructor & Destructor Documentation

7.153.2.1 **Region::Region ( double *density*, double *distanceFromOrigin*, double *angleWrtOrigin*, const std::vector< cv::Point > & *outerBorderPolygon*, const std::vector< std::vector< cv::Point > > & *innerBorderPolygons* )**

Definition at line 11 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::angle, multiscale::analysis::SpatialEntityPseudo3D::density, multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin, innerBorderPolygons, outerBorderPolygon, and validateInputValues().

### 7.153.2.2 Region::~Region ( )

Definition at line 25 of file Region.cpp.

### 7.153.3 Member Function Documentation

7.153.3.1 **bool Region::isValidInputPolygons ( const std::vector< cv::Point > & *outerBorderPolygon*, const std::vector< std::vector< cv::Point > > & *innerBorderPolygons* ) [private]**

Check if the given input outer/inner border polygons are valid.

For each polygon p and each point a:  $0 \leq p.a.x \leq p.a.y$

#### Parameters

|                            |                                                      |
|----------------------------|------------------------------------------------------|
| <i>outerBorderPolygon</i>  | The polygon defining the outer border of the region  |
| <i>innerBorderPolygons</i> | The polygon defining the inner borders of the region |

Definition at line 61 of file Region.cpp.

References innerBorderPolygons, and isValidInputPolygon().

Referenced by areValidInputValues().

7.153.3.2 **bool Region::isValidInputPolygons ( const std::vector< std::vector< cv::Point > > & *polygons* ) [private]**

Check if the given input polygons are valid.

For each polygon p and each point a:  $0 \leq p.a.x \leq p.a.y$

**Parameters**

|                 |                                  |
|-----------------|----------------------------------|
| <i>polygons</i> | The given collection of polygons |
|-----------------|----------------------------------|

Definition at line 69 of file Region.cpp.

References isValidInputPolygon().

```
7.153.3.3 bool Region::isValidInputValues (double density, double distanceFromOrigin, double angleWrtOrigin,
const std::vector< cv::Point > & outerBorderPolygon, const std::vector< std::vector< cv::Point > > &
innerBorderPolygons) [private]
```

Check if the input values are valid or not.

Validation rules:  $0 < \text{density}$   $0 < \text{distanceFromOrigin}$   $0 \leq \text{angleWrtOrigin} \leq 360$

For each polygon point p:  $0 \leq p.x \leq p.y$

**Parameters**

|                            |                                                      |
|----------------------------|------------------------------------------------------|
| <i>density</i>             | The density of the region                            |
| <i>distanceFromOrigin</i>  | The distance from the origin                         |
| <i>angleWrtOrigin</i>      | The angle computed wrt to the origin                 |
| <i>outerBorderPolygon</i>  | The polygon defining the outer border of the region  |
| <i>innerBorderPolygons</i> | The polygon defining the inner borders of the region |

Definition at line 44 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::area, isValidInputPolygons(), multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree, multiscale::Numeric::greaterOrEqual(), innerBorderPolygons, and multiscale::Numeric::lessOrEqual().

Referenced by validateInputValues().

```
7.153.3.4 double Region::computeAreaIfOuterBoderDefined () [private]
```

Compute the value of the area if the outer border of the region is defined.

Definition at line 116 of file Region.cpp.

References CONTOUR\_ORIENTED, innerBorderPolygons, and outerBorderPolygon.

Referenced by updateArea().

```
7.153.3.5 double Region::computeClusterednessDegreeIfOuterBorderDefined () [private]
```

Compute the value of the clusteredness degree if the outer border of the region is defined.

Definition at line 95 of file Region.cpp.

References CONTOUR\_ORIENTED, innerBorderPolygons, and outerBorderPolygon.

Referenced by updateClusterednessDegree().

```
7.153.3.6 const std::vector< std::vector< cv::Point > > & Region::getInnerBorderPolygons () const
```

Get the polygons defining the inner borders of the region.

Definition at line 31 of file Region.cpp.

References innerBorderPolygons.

Referenced by multiscale::analysis::RegionDetector::outputRegionToImage().

#### 7.153.3.7 const std::vector< cv::Point > & Region::getOuterBorderPolygon( ) const

Get the polygon defining the outer border of the region.

Definition at line 27 of file Region.cpp.

References outerBorderPolygon.

Referenced by multiscale::analysis::RegionDetector::outputRegionToImage().

#### 7.153.3.8 double Region::isCircularMeasure( ) [override], [private], [virtual]

Get the measure that the cluster has a circular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 153 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), outerBorderPolygon, and multiscale::Geometry2D::PI.

#### 7.153.3.9 double Region::isRectangularMeasure( ) [override], [private], [virtual]

Get the measure that the cluster has a rectangular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 144 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), and outerBorderPolygon.

#### 7.153.3.10 double Region::isTriangularMeasure( ) [override], [private], [virtual]

Get the measure that the cluster has a triangular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 131 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::convertPoints(), multiscale::analysis::SpatialEntityPseudo3D::CONVEX\_HULL\_CLOCKWISE, multiscale::MinEnclosingTriangleFinder::find(), multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), and outerBorderPolygon.

#### 7.153.3.11 bool Region::isValidInputPolygon( const std::vector< cv::Point > & polygon ) [private]

Check if the given input polygons are valid.

For each polygon point p:  $0 \leq p.x \leq p.y$

Parameters

|                |                   |
|----------------|-------------------|
| <i>polygon</i> | The given polygon |
|----------------|-------------------|

Definition at line 79 of file Region.cpp.

Referenced by [areValidInputPolygons\(\)](#).

#### 7.153.3.12 SpatialEntityPseudo3DType Region::type( ) [override], [private], [virtual]

Return the type of the pseudo 3D spatial entity.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 172 of file Region.cpp.

References [multiscale::analysis::Region](#).

#### 7.153.3.13 void Region::updateArea( ) [override], [private], [virtual]

Update the area.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 110 of file Region.cpp.

References [multiscale::analysis::SpatialEntityPseudo3D::area](#), [computeArealfOuterBoderDefined\(\)](#), and [outerBorderPolygon](#).

#### 7.153.3.14 void Region::updateCentrePoint( ) [override], [private], [virtual]

Update the centre of the region.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 165 of file Region.cpp.

References [multiscale::analysis::SpatialEntityPseudo3D::centre](#), and [outerBorderPolygon](#).

#### 7.153.3.15 void Region::updateClusterednessDegree( ) [override], [private], [virtual]

Update the value of the clusteredness degree.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 89 of file Region.cpp.

References [multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree](#), [computeClusterednessDegreeifOuterBorderDefined\(\)](#), and [outerBorderPolygon](#).

#### 7.153.3.16 void Region::updateDensity( ) [override], [private], [virtual]

Update the value of the density.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 108 of file Region.cpp.

#### 7.153.3.17 void Region::updatePerimeter( ) [override], [private], [virtual]

Update the perimeter.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 127 of file Region.cpp.

References [CONTOUR\\_CLOSED](#), [outerBorderPolygon](#), and [multiscale::analysis::SpatialEntityPseudo3D::perimeter](#).

#### 7.153.3.18 void Region::validateInputValues( double density, double distanceFromOrigin, double angleWrtOrigin, const std::vector< cv::Point > & outerBorderPolygon, const std::vector< std::vector< cv::Point > > & innerBorderPolygons ) [private]

Validate the input values.

Validation rules:  $0 < \text{density}$   $0 < \text{distanceFromOrigin}$   $0 \leq \text{angleWrtOrigin} \leq 360$

For each polygon point p:  $0 \leq p.x \leq p.y$

#### Parameters

|                            |                                                      |
|----------------------------|------------------------------------------------------|
| <i>density</i>             | The density of the region                            |
| <i>distanceFromOrigin</i>  | The distance from the origin                         |
| <i>angleWrtOrigin</i>      | The angle computed wrt to the origin                 |
| <i>outerBorderPolygon</i>  | The polygon defining the outer border of the region  |
| <i>innerBorderPolygons</i> | The polygon defining the inner borders of the region |

Definition at line 35 of file Region.cpp.

References `isValidInputValues()`, `multiscale::analysis::SpatialEntityPseudo3D::ERR_INPUT`, `innerBorderPolygons`, and `MS_throw`.

Referenced by `Region()`.

### 7.153.4 Member Data Documentation

#### 7.153.4.1 const bool Region::CONTOUR\_CLOSED = true [static], [private]

Definition at line 151 of file Region.hpp.

Referenced by `updatePerimeter()`.

#### 7.153.4.2 const bool Region::CONTOUR\_ORIENTED = false [static], [private]

Definition at line 150 of file Region.hpp.

Referenced by `computeArealfOuterBoderDefined()`, and `computeClusterednessDegreeIfOuterBorderDefined()`.

#### 7.153.4.3 std::vector<std::vector<cv::Point>> multiscale::analysis::Region::innerBorderPolygons [private]

Polygon defining the inner borders of the region

Definition at line 23 of file Region.hpp.

Referenced by `isValidInputPolygons()`, `isValidInputValues()`, `computeArealfOuterBoderDefined()`, `computeClusterednessDegreeIfOuterBorderDefined()`, `getInnerBorderPolygons()`, `Region()`, and `validateInputValues()`.

#### 7.153.4.4 std::vector<cv::Point> multiscale::analysis::Region::outerBorderPolygon [private]

Polygon defining the outer border of the region

Definition at line 21 of file Region.hpp.

Referenced by `computeArealfOuterBoderDefined()`, `computeClusterednessDegreeIfOuterBorderDefined()`, `getOuterBorderPolygon()`, `isCircularMeasure()`, `isRectangularMeasure()`, `isTriangularMeasure()`, `Region()`, `updateArea()`, `updateCentrePoint()`, `updateClusterednessDegree()`, and `updatePerimeter()`.

The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Region.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Region.cpp

## 7.154 multiscale::analysis::RegionDetector Class Reference

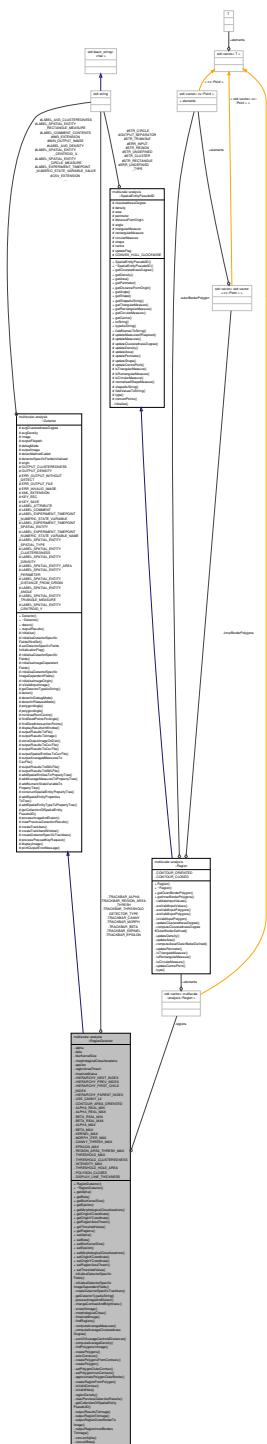
Class for detecting regions of high intensity in grayscale images.

```
#include <RegionDetector.hpp>
```

Inheritance diagram for multiscale::analysis::RegionDetector:



Collaboration diagram for multiscale::analysis::RegionDetector:



## Public Member Functions

- `RegionDetector (bool debugMode=false)`
- `~RegionDetector ()`
- `int getAlpha ()`

*Get the value of field alpha.*

- `int getBeta ()`

- `int getBlurKernelSize ()`

*Get the value of field blurKernelSize.*
- `int getEpsilon ()`

*Get the value of field epsilon.*
- `int getMorphologicalCloselterations ()`

*Get the value of field morphologicalCloselterations.*
- `int getOriginXCoordinate ()`

*Get the value of field originXCoordinate.*
- `int getOriginYCoordinate ()`

*Get the value of field originYCoordinate.*
- `int getRegionAreaThresh ()`

*Get the value of field regionAreaThresh.*
- `int getThresholdValue ()`

*Get the value of field thresholdValue.*
- `std::vector< Region > const & getRegions ()`

*Get a const reference to the std::vector of detected regions.*
- `void setAlpha (int alpha)`

*Set the value of field alpha.*
- `void setBeta (int beta)`

*Set the value of field beta.*
- `void setBlurKernelSize (int blurKernelSize)`

*Set the value of field blurKernelSize.*
- `void setEpsilon (int epsilon)`

*Set the value of field epsilon.*
- `void setMorphologicalCloselterations (int morphologicalCloselterations)`

*Set the value of field morphologicalCloselterations.*
- `void setOriginXCoordinate (int originXCoordinate)`

*Set the value of field originXCoordinate.*
- `void setOriginYCoordinate (int originYCoordinate)`

*Set the value of field originYCoordinate.*
- `void setRegionAreaThresh (int regionAreaThresh)`

*Set the value of field regionAreaThresh.*
- `void setThresholdValue (int thresholdValue)`

*Set the value of field thresholdValue.*

## Private Member Functions

- `void initialiseDetectorSpecificFields () override`

*Initialise the vision members.*
- `void initialiseDetectorSpecificImageDependentFields () override`

*Initialisation of the detector specific image dependent values.*
- `void createDetectorSpecificTrackbars () override`

*Create the trackbars.*
- `std::string getDetectorTypeAsString () override`

*Get the type of the detector as a std::string.*
- `void processImageAndDetect () override`

*Process the given image.*
- `void changeContrastAndBrightness (cv::Mat &processedImage)`

*Change the contrast and brightness of the image.*

- void **smoothImage** (cv::Mat &image)
 

*Smooth out differences in the image.*
- void **morphologicalClose** (cv::Mat &image)
 

*Apply the morphological close operator on the image.*
- void **thresholdImage** (const cv::Mat &image, cv::Mat &thresholdedImage)
 

*Apply the threshold filter on the image.*
- void **findRegions** (const cv::Mat &image, std::vector< Region > &regions)
 

*Find the regions in the image.*
- void **computeAverageMeasures** (std::vector< Region > &regions)
 

*Compute the average clusteredness degree and average density.*
- void **computeAverageClusterednessDegree** (std::vector< Region > &regions)
 

*Compute the average clusteredness degree.*
- double **sumOfAverageCentroidDistances** (std::vector< Region > &regions)
 

*Compute the average distances sum between regions' centroids.*
- void **computeAverageDensity** (std::vector< Region > &regions)
 

*Compute the average density.*
- std::vector< Polygon > **findPolygonsInImage** (const cv::Mat &image)
 

*Find polygons in image.*
- std::vector< Polygon > **createPolygons** (const std::vector< std::vector< cv::Point > > &contours, const std::vector< cv::Vec4i > &hierarchy)
 

*Create polygons from the given contours and hierarchy information.*
- bool **existContours** (const std::vector< std::vector< cv::Point > > &contours)
 

*Check if the number of contours is greater than 0.*
- void **createPolygonsFromContours** (const std::vector< std::vector< cv::Point > > &contours, const std::vector< cv::Vec4i > &hierarchy, std::vector< Polygon > &polygons)
 

*Create polygons from the given contours and hierarchy information.*
- **Polygon createPolygon** (int contourIndex, const std::vector< std::vector< cv::Point > > &contours, const std::vector< cv::Vec4i > &hierarchy)
 

*Create a new polygon considering the given contour index, contours and hierarchy information.*
- void **setPolygonOuterContour** (int contourIndex, const std::vector< std::vector< cv::Point > > &contours, const std::vector< cv::Vec4i > &hierarchy, Polygon &polygon)
 

*Set the outer contour of the polygon.*
- void **setPolygonInnerContours** (int contourIndex, const std::vector< std::vector< cv::Point > > &contours, const std::vector< cv::Vec4i > &hierarchy, Polygon &polygon)
 

*Set the inner contours of the polygon.*
- void **approximatePolygonOuterBorder** (Polygon &polygon)
 

*Approximate the outer contour of the given polygon.*
- **Region createRegionFromPolygon** (const Polygon &polygon)
 

*Create a new region from the given polygon.*
- bool **isValidContour** (const std::vector< cv::Point > &contour)
 

*Check if the contour is valid.*
- bool **isValidHole** (const std::vector< cv::Point > &hole)
 

*Check if the hole is valid.*
- double **regionDensity** (const Polygon &polygon)
 

*Compute the density of the area delimited by the given polygon.*
- void **clearPreviousDetectionResults** () override
 

*Clear the element present in the regions std::vector.*
- std::vector< std::shared\_ptr< SpatialEntityPseudo3D > > **getCollectionOfSpatialEntityPseudo3D** () override
 

*Get the collection of clusters detected in the image.*
- void **outputResultsToImage** () override

- **Output the results to the outputImage instance.**
- void **outputRegionToImage** (const Region &region, cv::Mat &**outputImage**)
  - Output the region to the outputImage instance.*
- void **outputRegionOuterBorderToImage** (const std::vector< cv::Point > &outerBorder, cv::Mat &**outputImage**)
  - Output the outer border polygon of a region to the outputImage instance.*
- void **outputRegionInnerBordersToImage** (const std::vector< std::vector< cv::Point > > &innerBorders, cv::Mat &**outputImage**)
  - Output the inner border polygons of a region to the outputImage instance.*
- double **convertAlpha** (int alpha)
  - Convert alpha from the range [0, ALPHA\_MAX] to [ALPHA\_REAL\_MIN, ALPHA\_REAL\_MAX].*
- int **convertBeta** (int beta)
  - Convert beta from the range [0, BETA\_MAX] to [BETA\_REAL\_MIN, BETA\_REAL\_MAX].*

## Private Attributes

- int **alpha**
- int **beta**
- int **blurKernelSize**
- int **morphologicalCloselterations**
- int **epsilon**
- int **regionAreaThresh**
- int **thresholdValue**
- std::vector< Region > **regions**

## Static Private Attributes

- static const std::string **DETECTOR\_TYPE** = "Regions"
- static const std::string **TRACKBAR\_ALPHA** = "Alpha"
- static const std::string **TRACKBAR\_BETA** = "Beta"
- static const std::string **TRACKBAR\_KERNEL** = "Gaussian blur kernel size"
- static const std::string **TRACKBAR\_MORPH** = "Morphological open, number of iterations"
- static const std::string **TRACKBAR\_CANNY** = "Canny lower threshold"
- static const std::string **TRACKBAR\_EPSILON** = "Epsilon"
- static const std::string **TRACKBAR\_REGION\_AREA\_THRESH** = "Region area threshold"
- static const std::string **TRACKBAR\_THRESHOLD** = "Threshold value"
- static const int **HIERARCHY\_NEXT\_INDEX** = 0
- static const int **HIERARCHY\_PREV\_INDEX** = 1
- static const int **HIERARCHY\_FIRST\_CHILD\_INDEX** = 2
- static const int **HIERARCHY\_PARENT\_INDEX** = 3
- static const bool **USE\_CANNY\_L2** = true
- static const bool **CONTOUR\_AREA\_ORIENTED** = false
- static const double **ALPHA\_REAL\_MIN** = 1.0
- static const double **ALPHA\_REAL\_MAX** = 3.0
- static const int **BETA\_REAL\_MIN** = -100
- static const int **BETA\_REAL\_MAX** = 100
- static const int **ALPHA\_MAX** = 1000
- static const int **BETA\_MAX** = 200
- static const int **KERNEL\_MAX** = 2000
- static const int **MORPH\_ITER\_MAX** = 100
- static const int **CANNY\_THRESH\_MAX** = 100
- static const int **EPSILON\_MAX** = 100
- static const int **REGION\_AREA\_THRESH\_MAX** = 200000
- static const int **THRESHOLD\_MAX** = 255

- static const int **THRESHOLD\_CLUSTEREDNESS** = 0
- static const int **INTENSITY\_MAX** = 255
- static const int **THRESHOLD\_HOLE\_AREA** = 1000
- static const bool **POLYGON\_CLOSED** = true
- static const int **DISPLAY\_LINE\_THICKNESS** = 10

## Additional Inherited Members

### 7.154.1 Detailed Description

Class for detecting regions of high intensity in grayscale images.

Definition at line 24 of file RegionDetector.hpp.

### 7.154.2 Constructor & Destructor Documentation

#### 7.154.2.1 RegionDetector::RegionDetector ( **bool debugMode = false** )

Definition at line 14 of file RegionDetector.cpp.

References alpha, multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, beta, blurKernelSize, epsilon, morphologicalCloselterations, regionAreaThresh, and thresholdValue.

#### 7.154.2.2 RegionDetector::~RegionDetector ( )

Definition at line 27 of file RegionDetector.cpp.

### 7.154.3 Member Function Documentation

#### 7.154.3.1 void RegionDetector::approximatePolygonOuterBorder ( **Polygon & polygon** ) [private]

Approximate the outer contour of the given polygon.

##### Parameters

|                |                   |
|----------------|-------------------|
| <i>polygon</i> | The given polygon |
|----------------|-------------------|

Definition at line 318 of file RegionDetector.cpp.

References epsilon.

Referenced by findRegions().

#### 7.154.3.2 void RegionDetector::changeContrastAndBrightness ( **cv::Mat & processedImage** ) [private]

Change the contrast and brightness of the image.

Change the contrast and brightness of the image by the factors alpha and gamma

##### Parameters

|                       |                     |
|-----------------------|---------------------|
| <i>processedImage</i> | The processed image |
|-----------------------|---------------------|

Definition at line 163 of file RegionDetector.cpp.

References alpha, beta, convertAlpha(), convertBeta(), and multiscale::analysis::Detector::image.

Referenced by processImageAndDetect().

**7.154.3.3 void RegionDetector::clearPreviousDetectionResults( ) [override], [private], [virtual]**

Clear the element present in the regions std::vector.

Implements [multiscale::analysis::Detector](#).

Definition at line 357 of file RegionDetector.cpp.

References regions.

**7.154.3.4 void RegionDetector::computeAverageClusterednessDegree( std::vector< Region > & regions ) [private]**

Compute the average clusteredness degree.

Parameters

|                |                          |
|----------------|--------------------------|
| <i>regions</i> | The regions in the image |
|----------------|--------------------------|

Definition at line 202 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, and sumOfAverageCentroidDistances().

Referenced by computeAverageMeasures().

**7.154.3.5 void RegionDetector::computeAverageDensity( std::vector< Region > & regions ) [private]**

Compute the average density.

Parameters

|                |                          |
|----------------|--------------------------|
| <i>regions</i> | The regions in the image |
|----------------|--------------------------|

Definition at line 237 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgDensity.

Referenced by computeAverageMeasures().

**7.154.3.6 void RegionDetector::computeAverageMeasures( std::vector< Region > & regions ) [private]**

Compute the average clusteredness degree and average density.

Parameters

|                |                          |
|----------------|--------------------------|
| <i>regions</i> | The regions in the image |
|----------------|--------------------------|

Definition at line 197 of file RegionDetector.cpp.

References computeAverageClusterednessDegree(), and computeAverageDensity().

Referenced by processImageAndDetect().

**7.154.3.7 double RegionDetector::convertAlpha( int alpha ) [private]**

Convert alpha from the range [0, ALPHA\_MAX] to [ALPHA\_REAL\_MIN, ALPHA\_REAL\_MAX].

Parameters

|              |       |
|--------------|-------|
| <i>alpha</i> | Alpha |
|--------------|-------|

Definition at line 408 of file RegionDetector.cpp.

References ALPHA\_MAX, ALPHA\_REAL\_MAX, and ALPHA\_REAL\_MIN.

Referenced by changeContrastAndBrightness().

7.154.3.8 int RegionDetector::convertBeta( int *beta* ) [private]

Convert beta from the range [0, BETA\_MAX] to [BETA\_REAL\_MIN, BETA\_REAL\_MAX].

## Parameters

|             |      |
|-------------|------|
| <i>beta</i> | Beta |
|-------------|------|

Definition at line 416 of file RegionDetector.cpp.

References BETA\_MAX, BETA\_REAL\_MAX, and BETA\_REAL\_MIN.

Referenced by changeContrastAndBrightness().

## 7.154.3.9 void RegionDetector::createDetectorSpecificTrackbars( ) [override], [private], [virtual]

Create the trackbars.

Implements [multiscale::analysis::Detector](#).

Definition at line 135 of file RegionDetector.cpp.

References alpha, ALPHA\_MAX, beta, BETA\_MAX, blurKernelSize, epsilon, EPSILON\_MAX, KERNEL\_MAX, MORPH\_ITER\_MAX, morphologicalCloselterations, REGION\_AREA\_THRESH\_MAX, regionAreaThresh, THRESHOLD\_MAX, thresholdValue, TRACKBAR\_ALPHA, TRACKBAR\_BETA, TRACKBAR\_EPSILON, TRACKBAR\_KERNEL, TRACKBAR\_MORPH, TRACKBAR\_REGION\_AREA\_THRESH, TRACKBAR\_THRESHOLD, and multiscale::analysis::Detector::WIN\_OUTPUT\_IMAGE.

7.154.3.10 Polygon RegionDetector::createPolygon( int *contourIndex*, const std::vector< std::vector< cv::Point > > & *contours*, const std::vector< cv::Vec4i > & *hierarchy* ) [private]

Create a new polygon considering the given contour index, contours and hierarchy information.

## Parameters

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| <i>contourIndex</i> | The index of the outer contour                           |
| <i>contours</i>     | The collection of all contours                           |
| <i>hierarchy</i>    | The information regarding the hierarchy between contours |

Definition at line 291 of file RegionDetector.cpp.

References setPolygonInnerContours(), and setPolygonOuterContour().

Referenced by createPolygonsFromContours().

7.154.3.11 std::vector< Polygon > RegionDetector::createPolygons( const std::vector< std::vector< cv::Point > > & *contours*, const std::vector< cv::Vec4i > & *hierarchy* ) [private]

Create polygons from the given contours and hierarchy information.

## Parameters

|                  |                                                          |
|------------------|----------------------------------------------------------|
| <i>contours</i>  | The given contours                                       |
| <i>hierarchy</i> | The information regarding the hierarchy between contours |

Definition at line 264 of file RegionDetector.cpp.

References createPolygonsFromContours(), and existContours().

Referenced by findPolygonsInImage().

```
7.154.3.12 void RegionDetector::createPolygonsFromContours (const std::vector< std::vector< cv::Point > > & contours,
 const std::vector< cv::Vec4i > & hierarchy, std::vector< Polygon > & polygons) [private]
```

Create polygons from the given contours and hierarchy information.

**Parameters**

|                  |                                                            |
|------------------|------------------------------------------------------------|
| <i>contours</i>  | The given contours                                         |
| <i>hierarchy</i> | The information regarding the hierarchy between contours   |
| <i>polygons</i>  | The collection of polygons created from the given contours |

Definition at line 279 of file RegionDetector.cpp.

References createPolygon(), HIERARCHY\_NEXT\_INDEX, and isValidContour().

Referenced by createPolygons().

**7.154.3.13 Region RegionDetector::createRegionFromPolygon ( const Polygon & *polygon* ) [private]**

Create a new region from the given polygon.

Process the polygon in order to get the required information (e.g. clusteredness, area etc.) and create a region using this information

**Parameters**

|                |                                |
|----------------|--------------------------------|
| <i>polygon</i> | Polygon determining the region |
|----------------|--------------------------------|

Definition at line 324 of file RegionDetector.cpp.

References multiscale::Geometry2D::distanceBtwPoints(), multiscale::Geometry2D::minimumDistancePoint<-Index(), multiscale::analysis::Detector::origin, multiscale::analysis::Detector::polygonAngle(), multiscale::analysis::Region, and regionDensity().

Referenced by findRegions().

**7.154.3.14 bool RegionDetector::existContours ( const std::vector< std::vector< cv::Point > > & *contours* ) [private]**

Check if the number of contours is greater than 0.

**Parameters**

|                 |                    |
|-----------------|--------------------|
| <i>contours</i> | The given contours |
|-----------------|--------------------|

Definition at line 275 of file RegionDetector.cpp.

Referenced by createPolygons().

**7.154.3.15 std::vector< Polygon > RegionDetector::findPolygonsInImage ( const cv::Mat & *image* ) [private]**

Find polygons in image.

**Parameters**

|              |           |
|--------------|-----------|
| <i>image</i> | The image |
|--------------|-----------|

Definition at line 248 of file RegionDetector.cpp.

References createPolygons().

Referenced by findRegions().

**7.154.3.16 void RegionDetector::findRegions ( const cv::Mat & *image*, std::vector< Region > & *regions* ) [private]**

Find the regions in the image.

Find the contours, approximate the polygons and extract the required information from them.

**Parameters**

|                |                          |
|----------------|--------------------------|
| <i>image</i>   | The image                |
| <i>regions</i> | The regions in the image |

Definition at line 185 of file RegionDetector.cpp.

References approximatePolygonOuterBorder(), createRegionFromPolygon(), and findPolygonsInImage().

Referenced by processImageAndDetect().

**7.154.3.17 int RegionDetector::getAlpha( )**

Get the value of field alpha.

Definition at line 29 of file RegionDetector.cpp.

References alpha.

Referenced by saveDetectorParameterValues().

**7.154.3.18 int RegionDetector::getBeta( )**

Get the value of field beta.

Definition at line 33 of file RegionDetector.cpp.

References beta.

Referenced by saveDetectorParameterValues().

**7.154.3.19 int RegionDetector::getBlurKernelSize( )**

Get the value of field blurKernelSize.

Definition at line 37 of file RegionDetector.cpp.

References blurKernelSize.

Referenced by saveDetectorParameterValues().

**7.154.3.20 std::vector< std::shared\_ptr< SpatialEntityPseudo3D > > RegionDetector::getCollectionOfSpatialEntityPseudo3D( ) [override], [private], [virtual]**

Get the collection of clusters detected in the image.

Implements [multiscale::analysis::Detector](#).

Definition at line 361 of file RegionDetector.cpp.

References multiscale::analysis::Region, and regions.

**7.154.3.21 std::string RegionDetector::getDetectorTypeAsString( ) [override], [private], [virtual]**

Get the type of the detector as a std::string.

Implements [multiscale::analysis::Detector](#).

Definition at line 147 of file RegionDetector.cpp.

References DETECTOR\_TYPE.

7.154.3.22 int RegionDetector::getEpsilon ( )

Get the value of field epsilon.

Definition at line 45 of file RegionDetector.cpp.

References epsilon.

Referenced by saveDetectorParameterValues().

7.154.3.23 int RegionDetector::getMorphologicalCloselterations ( )

Get the value of field morphologicalCloselterations.

Definition at line 41 of file RegionDetector.cpp.

References morphologicalCloselterations.

Referenced by saveDetectorParameterValues().

7.154.3.24 int RegionDetector::getOriginXCoordinate ( )

Get the value of field originXCoordinate.

Definition at line 53 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin.

7.154.3.25 int RegionDetector::getOriginYCoordinate ( )

Get the value of field originYCoordinate.

Definition at line 57 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin.

7.154.3.26 int RegionDetector::getRegionAreaThresh ( )

Get the value of field regionAreaThresh.

Definition at line 49 of file RegionDetector.cpp.

References regionAreaThresh.

Referenced by saveDetectorParameterValues().

7.154.3.27 std::vector< Region > const & RegionDetector::getRegions ( )

Get a const reference to the std::vector of detected regions.

Definition at line 65 of file RegionDetector.cpp.

References regions.

7.154.3.28 int RegionDetector::getThresholdValue ( )

Get the value of field thresholdValue.

Definition at line 61 of file RegionDetector.cpp.

References thresholdValue.

Referenced by saveDetectorParameterValues().

**7.154.3.29 void RegionDetector::initialiseDetectorSpecificFields( ) [override], [private], [virtual]**

Initialise the vision members.

Implements [multiscale::analysis::Detector](#).

Definition at line 123 of file RegionDetector.cpp.

References alpha, beta, blurKernelSize, epsilon, morphologicalCloselterations, regionAreaThresh, and threshold←Value.

**7.154.3.30 void RegionDetector::initialiseDetectorSpecificImageDependentFields( ) [override], [private], [virtual]**

Initialisation of the detector specific image dependent values.

Implements [multiscale::analysis::Detector](#).

Definition at line 133 of file RegionDetector.cpp.

**7.154.3.31 bool RegionDetector::isValidContour( const std::vector< cv::Point > & contour ) [private]**

Check if the contour is valid.

Check if the area determined by the contour > regionAreaThreshold

Parameters

|                |                   |
|----------------|-------------------|
| <i>contour</i> | The given contour |
|----------------|-------------------|

Definition at line 334 of file RegionDetector.cpp.

References CONTOUR\_AREA\_ORIENTED, and regionAreaThresh.

Referenced by [createPolygonsFromContours\(\)](#).

**7.154.3.32 bool RegionDetector::isValidHole( const std::vector< cv::Point > & hole ) [private]**

Check if the hole is valid.

Check if the area determined by the hole > THRESHOLD\_HOLE\_AREA

Parameters

|             |                         |
|-------------|-------------------------|
| <i>hole</i> | The contour of the hole |
|-------------|-------------------------|

Definition at line 340 of file RegionDetector.cpp.

References CONTOUR\_AREA\_ORIENTED, and THRESHOLD\_HOLE\_AREA.

Referenced by [setPolygonInnerContours\(\)](#).

**7.154.3.33 void RegionDetector::morphologicalClose( cv::Mat & image ) [private]**

Apply the morphological close operator on the image.

Parameters

|              |           |
|--------------|-----------|
| <i>image</i> | The image |
|--------------|-----------|

Definition at line 175 of file RegionDetector.cpp.

References morphologicalCloselterations.

Referenced by [processImageAndDetect\(\)](#).

7.154.3.34 void RegionDetector::outputRegionInnerBordersToImage ( const std::vector< std::vector< cv::Point > > & *innerBorders*, cv::Mat & *outputImage* ) [private]

Output the inner border polygons of a region to the outputImage instance.

#### Parameters

|                     |                                                         |
|---------------------|---------------------------------------------------------|
| <i>innerBorders</i> | The polygons defining the inner border(s) of the region |
| <i>outputImage</i>  | The given output image                                  |

Definition at line 400 of file RegionDetector.cpp.

References DISPLAY\_LINE\_THICKNESS, INTENSITY\_MAX, and POLYGON\_CLOSED.

Referenced by outputRegionToImage().

7.154.3.35 void RegionDetector::outputRegionOuterBorderToImage ( const std::vector< cv::Point > & *outerBorder*, cv::Mat & *outputImage* ) [private]

Output the outer border polygon of a region to the outputImage instance.

#### Parameters

|                    |                                                     |
|--------------------|-----------------------------------------------------|
| <i>outerBorder</i> | The polygon defining the outer border of the region |
| <i>outputImage</i> | The given output image                              |

Definition at line 394 of file RegionDetector.cpp.

References DISPLAY\_LINE\_THICKNESS, INTENSITY\_MAX, and POLYGON\_CLOSED.

Referenced by outputRegionToImage().

7.154.3.36 void RegionDetector::outputRegionToImage ( const Region & *region*, cv::Mat & *outputImage* ) [private]

Output the region to the outputImage instance.

#### Parameters

|                    |                        |
|--------------------|------------------------|
| <i>region</i>      | The given region       |
| <i>outputImage</i> | The given output image |

Definition at line 389 of file RegionDetector.cpp.

References multiscale::analysis::Region::getInnerBorderPolygons(), multiscale::analysis::Region::getOuterBorderPolygon(), multiscale::analysis::Detector::outputImage, outputRegionInnerBordersToImage(), and outputRegionOuterBorderToImage().

Referenced by outputResultsToImage().

7.154.3.37 void RegionDetector::outputResultsToImage ( ) [override], [private], [virtual]

Output the results to the outputImage instance.

Implements [multiscale::analysis::Detector](#).

Definition at line 371 of file RegionDetector.cpp.

References multiscale::analysis::Detector::image, multiscale::analysis::Detector::outputImage, outputRegionToImage(), and regions.

7.154.3.38 void RegionDetector::processImageAndDetect ( ) [override], [private], [virtual]

Process the given image.

Apply filters to the image, threshold it, find its contours, approximate the polygons from these contours. Afterwards, process the polygons to find their distance from the origin, their area and the angle determined by the points from the contour which are on the edge and the closest point to the origin. Return all the polygons together with the processed information as a std::vector of regions.

Implements [multiscale::analysis::Detector](#).

Definition at line 151 of file RegionDetector.cpp.

References [changeContrastAndBrightness\(\)](#), [computeAverageMeasures\(\)](#), [findRegions\(\)](#), [morphologicalClose\(\)](#), [regions](#), [smoothImage\(\)](#), and [thresholdImage\(\)](#).

#### 7.154.3.39 double RegionDetector::regionDensity ( const Polygon & *polygon* ) [private]

Compute the density of the area delimited by the given polygon.

The density is equal to the average intensity of the pixels in the area delimited by the given polygon divided by INTENSITY\_MAX.

Parameters

|                |                   |
|----------------|-------------------|
| <i>polygon</i> | The given polygon |
|----------------|-------------------|

Definition at line 346 of file RegionDetector.cpp.

References [multiscale::analysis::Detector::image](#), and [INTENSITY\\_MAX](#).

Referenced by [createRegionFromPolygon\(\)](#).

#### 7.154.3.40 void RegionDetector::setAlpha ( int *alpha* )

Set the value of field alpha.

Parameters

|              |                |
|--------------|----------------|
| <i>alpha</i> | Value of alpha |
|--------------|----------------|

Definition at line 69 of file RegionDetector.cpp.

References [alpha](#), and [multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag\(\)](#).

Referenced by [loadDetectorParameterValues\(\)](#).

#### 7.154.3.41 void RegionDetector::setBeta ( int *beta* )

Set the value of field beta.

Parameters

|             |               |
|-------------|---------------|
| <i>beta</i> | Value of beta |
|-------------|---------------|

Definition at line 75 of file RegionDetector.cpp.

References [beta](#), and [multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag\(\)](#).

Referenced by [loadDetectorParameterValues\(\)](#).

#### 7.154.3.42 void RegionDetector::setBlurKernelSize ( int *blurKernelSize* )

Set the value of field blurKernelSize.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>blurKernelSize</i> | Value of blurKernelSize |
|-----------------------|-------------------------|

Definition at line 81 of file RegionDetector.cpp.

References blurKernelSize, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

**7.154.3.43 void RegionDetector::setEpsilon ( int *epsilon* )**

Set the value of field epsilon.

**Parameters**

|                |                  |
|----------------|------------------|
| <i>epsilon</i> | Value of epsilon |
|----------------|------------------|

Definition at line 87 of file RegionDetector.cpp.

References epsilon, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

**7.154.3.44 void RegionDetector::setMorphologicalCloselterations ( int *morphologicalCloselterations* )**

Set the value of field morphologicalCloselterations.

**Parameters**

|                                     |                                       |
|-------------------------------------|---------------------------------------|
| <i>morphologicalCloselterations</i> | Value of morphologicalCloselterations |
|-------------------------------------|---------------------------------------|

Definition at line 93 of file RegionDetector.cpp.

References morphologicalCloselterations, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

**7.154.3.45 void RegionDetector::setOriginXCoordinate ( int *originXCoordinate* )**

Set the value of field originXCoordinate.

**Parameters**

|                          |                            |
|--------------------------|----------------------------|
| <i>originXCoordinate</i> | Value of originXCoordinate |
|--------------------------|----------------------------|

Definition at line 99 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

**7.154.3.46 void RegionDetector::setOriginYCoordinate ( int *originYCoordinate* )**

Set the value of field originYCoordinate.

**Parameters**

|                                |                            |
|--------------------------------|----------------------------|
| <i>originY←<br/>Coordinate</i> | Value of originYCoordinate |
|--------------------------------|----------------------------|

Definition at line 105 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin, and multiscale::analysis::Detector::setDetectorSpecificFields←InitialisationFlag().

**7.154.3.47 void RegionDetector::setPolygonInnerContours ( int *contourIndex*, const std::vector< std::vector< cv::Point > >  
& *contours*, const std::vector< cv::Vec4i > & *hierarchy*, Polygon & *polygon* ) [private]**

Set the inner contours of the polygon.

#### Parameters

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| <i>contourIndex</i> | The index of the outer contour                           |
| <i>contours</i>     | The collection of all contours                           |
| <i>hierarchy</i>    | The information regarding the hierarchy between contours |
| <i>polygon</i>      | The polygon for which the outer contour is set           |

Definition at line 306 of file RegionDetector.cpp.

References HIERARCHY\_PARENT\_INDEX, and isValidHole().

Referenced by createPolygon().

**7.154.3.48 void RegionDetector::setPolygonOuterContour ( int *contourIndex*, const std::vector< std::vector< cv::Point > >  
& *contours*, const std::vector< cv::Vec4i > & *hierarchy*, Polygon & *polygon* ) [private]**

Set the outer contour of the polygon.

#### Parameters

|                     |                                                          |
|---------------------|----------------------------------------------------------|
| <i>contourIndex</i> | The index of the outer contour                           |
| <i>contours</i>     | The collection of all contours                           |
| <i>hierarchy</i>    | The information regarding the hierarchy between contours |
| <i>polygon</i>      | The polygon for which the outer contour is set           |

Definition at line 301 of file RegionDetector.cpp.

Referenced by createPolygon().

**7.154.3.49 void RegionDetector::setRegionAreaThresh ( int *regionAreaThresh* )**

Set the value of field regionAreaThresh.

#### Parameters

|                               |                           |
|-------------------------------|---------------------------|
| <i>regionArea←<br/>Thresh</i> | Value of regionAreaThresh |
|-------------------------------|---------------------------|

Definition at line 111 of file RegionDetector.cpp.

References regionAreaThresh, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

**7.154.3.50 void RegionDetector::setThresholdValue ( int *thresholdValue* )**

Set the value of field thresholdValue.

**Parameters**

|                       |                         |
|-----------------------|-------------------------|
| <i>thresholdValue</i> | Value of thresholdValue |
|-----------------------|-------------------------|

Definition at line 117 of file RegionDetector.cpp.

References multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag(), and thresholdValue.

Referenced by loadDetectorParameterValues().

**7.154.3.51 void RegionDetector::smoothImage ( cv::Mat & *image* ) [private]**

Smooth out differences in the image.

Apply a Gaussian blur filter

**Parameters**

|              |           |
|--------------|-----------|
| <i>image</i> | The image |
|--------------|-----------|

Definition at line 167 of file RegionDetector.cpp.

References blurKernelSize.

Referenced by processImageAndDetect().

**7.154.3.52 double RegionDetector::sumOfAverageCentroidDistances ( std::vector< Region > & *regions* ) [private]**

Compute the average distances sum between regions' centroids.

**Parameters**

|                |                          |
|----------------|--------------------------|
| <i>regions</i> | The regions in the image |
|----------------|--------------------------|

Definition at line 217 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, and multiscale::Geometry2D::distanceBtwPoints().

Referenced by computeAverageClusterednessDegree().

**7.154.3.53 void RegionDetector::thresholdImage ( const cv::Mat & *image*, cv::Mat & *thresholdedImage* ) [private]**

Apply the threshold filter on the image.

**Parameters**

|                         |                       |
|-------------------------|-----------------------|
| <i>image</i>            | The image             |
| <i>thresholdedImage</i> | The thresholded image |

Definition at line 181 of file RegionDetector.cpp.

References THRESHOLD\_MAX, and thresholdValue.

Referenced by processImageAndDetect().

**7.154.4 Member Data Documentation****7.154.4.1 int multiscale::analysis::RegionDetector::alpha [private]**

Alpha for brightness and contrast adjustments

Definition at line 28 of file RegionDetector.hpp.

Referenced by changeContrastAndBrightness(), createDetectorSpecificTrackbars(), getAlpha(), initialiseDetectorSpecificFields(), RegionDetector(), and setAlpha().

**7.154.4.2 const int RegionDetector::ALPHA\_MAX = 1000 [static], [private]**

Definition at line 381 of file RegionDetector.hpp.

Referenced by convertAlpha(), and createDetectorSpecificTrackbars().

**7.154.4.3 const double RegionDetector::ALPHA\_REAL\_MAX = 3.0 [static], [private]**

Definition at line 376 of file RegionDetector.hpp.

Referenced by convertAlpha().

**7.154.4.4 const double RegionDetector::ALPHA\_REAL\_MIN = 1.0 [static], [private]**

Definition at line 375 of file RegionDetector.hpp.

Referenced by convertAlpha().

**7.154.4.5 int multiscale::analysis::RegionDetector::beta [private]**

Beta for brightness and contrast adjustments

Definition at line 29 of file RegionDetector.hpp.

Referenced by changeContrastAndBrightness(), createDetectorSpecificTrackbars(), getBeta(), initialiseDetectorSpecificFields(), RegionDetector(), and setBeta().

**7.154.4.6 const int RegionDetector::BETA\_MAX = 200 [static], [private]**

Definition at line 382 of file RegionDetector.hpp.

Referenced by convertBeta(), and createDetectorSpecificTrackbars().

**7.154.4.7 const int RegionDetector::BETA\_REAL\_MAX = 100 [static], [private]**

Definition at line 379 of file RegionDetector.hpp.

Referenced by convertBeta().

**7.154.4.8 const int RegionDetector::BETA\_REAL\_MIN = -100 [static], [private]**

Definition at line 378 of file RegionDetector.hpp.

Referenced by convertBeta().

**7.154.4.9 int multiscale::analysis::RegionDetector::blurKernelSize [private]**

Kernel size for Gaussian blur

Definition at line 30 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getBlurKernelSize(), initialiseDetectorSpecificFields(), RegionDetector(), setBlurKernelSize(), and smoothImage().

7.154.4.10 const int RegionDetector::CANNY\_THRESH\_MAX = 100 [static], [private]

Definition at line 385 of file RegionDetector.hpp.

7.154.4.11 const bool RegionDetector::CONTOUR\_AREA\_ORIENTED = false [static], [private]

Definition at line 373 of file RegionDetector.hpp.

Referenced by isValidContour(), and isValidHole().

7.154.4.12 const std::string RegionDetector::DETECTOR\_TYPE = "Regions" [static], [private]

Definition at line 356 of file RegionDetector.hpp.

Referenced by getDetectorTypeAsString().

7.154.4.13 const int RegionDetector::DISPLAY\_LINE\_THICKNESS = 10 [static], [private]

Definition at line 396 of file RegionDetector.hpp.

Referenced by outputRegionInnerBordersToImage(), and outputRegionOuterBorderToImage().

7.154.4.14 int multiscale::analysis::RegionDetector::epsilon [private]

Epsilon for polygon approximation

Definition at line 32 of file RegionDetector.hpp.

Referenced by approximatePolygonOuterBorder(), createDetectorSpecificTrackbars(), getEpsilon(), initialiseDetectorSpecificFields(), RegionDetector(), and setEpsilon().

7.154.4.15 const int RegionDetector::EPSILON\_MAX = 100 [static], [private]

Definition at line 386 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.16 const int RegionDetector::HIERARCHY\_FIRST\_CHILD\_INDEX = 2 [static], [private]

Definition at line 369 of file RegionDetector.hpp.

7.154.4.17 const int RegionDetector::HIERARCHY\_NEXT\_INDEX = 0 [static], [private]

Definition at line 367 of file RegionDetector.hpp.

Referenced by createPolygonsFromContours().

7.154.4.18 const int RegionDetector::HIERARCHY\_PARENT\_INDEX = 3 [static], [private]

Definition at line 370 of file RegionDetector.hpp.

Referenced by setPolygonInnerContours().

7.154.4.19 const int RegionDetector::HIERARCHY\_PREV\_INDEX = 1 [static], [private]

Definition at line 368 of file RegionDetector.hpp.

7.154.4.20 const int RegionDetector::INTENSITY\_MAX = 255 [static], [private]

Definition at line 390 of file RegionDetector.hpp.

Referenced by outputRegionInnerBordersToImage(), outputRegionOuterBorderToImage(), and regionDensity().

7.154.4.21 const int RegionDetector::KERNEL\_MAX = 2000 [static], [private]

Definition at line 383 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.22 const int RegionDetector::MORPH\_ITER\_MAX = 100 [static], [private]

Definition at line 384 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.23 int multiscale::analysis::RegionDetector::morphologicalCloselterations [private]

Number of iterations for morphological close operator

Definition at line 31 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getMorphologicalCloselterations(), initialiseDetectorSpecificFields(), morphologicalClose(), RegionDetector(), and setMorphologicalCloselterations().

7.154.4.24 const bool RegionDetector::POLYGON\_CLOSED = true [static], [private]

Definition at line 394 of file RegionDetector.hpp.

Referenced by outputRegionInnerBordersToImage(), and outputRegionOuterBorderToImage().

7.154.4.25 const int RegionDetector::REGION\_AREA\_THRESH\_MAX = 200000 [static], [private]

Definition at line 387 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.26 int multiscale::analysis::RegionDetector::regionAreaThresh [private]

Threshold for considering a region

Definition at line 33 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getRegionAreaThresh(), initialiseDetectorSpecificFields(), isValidContour(), RegionDetector(), and setRegionAreaThresh().

7.154.4.27 std::vector<Region> multiscale::analysis::RegionDetector::regions [private]

Regions detected in the image

Definition at line 36 of file RegionDetector.hpp.

Referenced by clearPreviousDetectionResults(), getCollectionOfSpatialEntityPseudo3D(), getRegions(), outputResultsToImage(), and processImageAndDetect().

7.154.4.28 const int RegionDetector::THRESHOLD\_CLUSTEREDNESS = 0 [static], [private]

Definition at line 389 of file RegionDetector.hpp.

7.154.4.29 const int RegionDetector::THRESHOLD\_HOLE\_AREA = 1000 [static], [private]

Definition at line 392 of file RegionDetector.hpp.

Referenced by isValidHole().

7.154.4.30 const int RegionDetector::THRESHOLD\_MAX = 255 [static], [private]

Definition at line 388 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), and thresholdImage().

7.154.4.31 int multiscale::analysis::RegionDetector::thresholdValue [private]

Value of the threshold for the threshold filter

Definition at line 34 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getThresholdValue(), initialiseDetectorSpecificFields(), RegionDetector(), setThresholdValue(), and thresholdImage().

7.154.4.32 const std::string RegionDetector::TRACKBAR\_ALPHA = "Alpha" [static], [private]

Definition at line 358 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.33 const std::string RegionDetector::TRACKBAR\_BETA = "Beta" [static], [private]

Definition at line 359 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.34 const std::string RegionDetector::TRACKBAR\_CANNY = "Canny lower threshold" [static], [private]

Definition at line 362 of file RegionDetector.hpp.

7.154.4.35 const std::string RegionDetector::TRACKBAR\_EPSILON = "Epsilon" [static], [private]

Definition at line 363 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.36 const std::string RegionDetector::TRACKBAR\_KERNEL = "Gaussian blur kernel size" [static], [private]

Definition at line 360 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.154.4.37 `const std::string RegionDetector::TRACKBAR_MORPH = "Morphological open, number of iterations"`  
[static], [private]

Definition at line 361 of file RegionDetector.hpp.

Referenced by `createDetectorSpecificTrackbars()`.

7.154.4.38 `const std::string RegionDetector::TRACKBAR_REGION_AREA_THRESH = "Region area threshold"` [static],  
[private]

Definition at line 364 of file RegionDetector.hpp.

Referenced by `createDetectorSpecificTrackbars()`.

7.154.4.39 `const std::string RegionDetector::TRACKBAR_THRESHOLD = "Threshold value"` [static], [private]

Definition at line 365 of file RegionDetector.hpp.

Referenced by `createDetectorSpecificTrackbars()`.

7.154.4.40 `const bool RegionDetector::USE_CANNY_L2 = true` [static], [private]

Definition at line 372 of file RegionDetector.hpp.

The documentation for this class was generated from the following files:

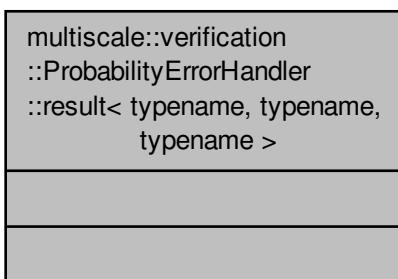
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/RegionDetector.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/RegionDetector.cpp

## 7.155 multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename > Struct Template Reference

Structure for specifying the type of the result.

```
#include <ProbabilityErrorHandler.hpp>
```

Collaboration diagram for multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >:



## Public Types

- `typedef void type`

### 7.155.1 Detailed Description

```
template<typename, typename, typename>struct multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >
```

Structure for specifying the type of the result.

Definition at line 23 of file `ProbabilityErrorHandler.hpp`.

### 7.155.2 Member Typedef Documentation

#### 7.155.2.1 template<typename , typename , typename > `typedef void multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >::type`

Definition at line 23 of file `ProbabilityErrorHandler.hpp`.

The documentation for this struct was generated from the following file:

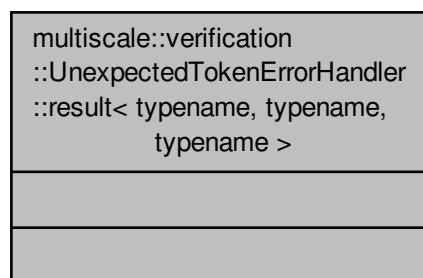
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/ProbabilityErrorHandler.hpp`

## 7.156 multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename > Struct Template Reference

Structure for specifying the type of the result.

```
#include <UnexpectedErrorHandler.hpp>
```

Collaboration diagram for `multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename >`:



## Public Types

- `typedef void type`

### 7.156.1 Detailed Description

```
template<typename, typename, typename>struct multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename >
```

Structure for specifying the type of the result.

Definition at line 23 of file UnexpectedErrorHandler.hpp.

### 7.156.2 Member Typedef Documentation

```
7.156.2.1 template<typename , typename , typename > typedef void multiscale::verification::UnexpectedTokenErrorHandler::result< typename, typename, typename >::type
```

Definition at line 23 of file UnexpectedErrorHandler.hpp.

The documentation for this struct was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/[UnexpectedErrorHandler.hpp](#)

## 7.157 multiscale::RGBColourGenerator Class Reference

Generate a RGB colour.

```
#include <RGBColourGenerator.hpp>
```

Collaboration diagram for multiscale::RGBColourGenerator:

| multiscale::RGBColourGenerator                                                                        |
|-------------------------------------------------------------------------------------------------------|
| + HUE_MIN<br>+ HUE_MAX<br>+ SATURATION<br>+ VALUE<br>- red<br>- green<br>- blue                       |
| + generate()<br>+ generate()<br>- convertHSVToRGB()<br>- computeRGBValues()<br>- convertRGBToString() |

### Public Member Functions

- std::string [generate](#) (double concentrationMin, double concentrationMax, double concentration)

*Generate a RGB colour for the given concentration.*

- cv::Scalar [generate](#) (cv::RNG &randomNumberGenerator)

*Generate a random RGB colour.*

## Static Public Attributes

- static const int [HUE\\_MIN](#) = 0
- static const int [HUE\\_MAX](#) = 120
- static const int [SATURATION](#) = 1
- static const int [VALUE](#) = 1

## Private Member Functions

- std::string [convertHSVToRGB](#) (double hue, double saturation, double value)  
*Convert a colour from HSV to RGB colour space.*
- void [computeRGBValues](#) (int huePrime, double X, double chroma, double m)  
*Compute RGB values from HSV specific values.*
- std::string [convertRGBToString](#) ()  
*Convert the RGB colour to a std::string.*

## Private Attributes

- double red
- double green
- double blue

### 7.157.1 Detailed Description

Generate a RGB colour.

Generate a RGB colour given the possible range for concentrations and the value of one of the concentrations

The conversion HSV->RGB is based on the wikipedia page on this topic

Definition at line 18 of file RGBColourGenerator.hpp.

### 7.157.2 Member Function Documentation

#### 7.157.2.1 void RGBColourGenerator::computeRGBValues ( int *huePrime*, double *X*, double *chroma*, double *m* ) [private]

Compute RGB values from HSV specific values.

##### Parameters

|                 |        |
|-----------------|--------|
| <i>huePrime</i> | Hue'   |
| <i>X</i>        | X      |
| <i>chroma</i>   | Chroma |
| <i>m</i>        | m      |

Definition at line 41 of file RGBColourGenerator.cpp.

References blue, green, and red.

Referenced by convertHSVToRGB().

7.157.2.2 `std::string RGBColourGenerator::convertHSVToRGB ( double hue, double saturation, double value )`  
[private]

Convert a colour from HSV to RGB colour space.

**Parameters**

|                   |            |
|-------------------|------------|
| <i>hue</i>        | Hue        |
| <i>saturation</i> | Saturation |
| <i>value</i>      | Value      |

Definition at line 27 of file RGBColourGenerator.cpp.

References computeRGBValues(), and convertRGBToString().

Referenced by generate().

**7.157.2.3 std::string RGBColourGenerator::convertRGBToString( ) [private]**

Convert the RGB colour to a std::string.

Definition at line 85 of file RGBColourGenerator.cpp.

References blue, green, and red.

Referenced by convertHSVToRGB().

**7.157.2.4 std::string RGBColourGenerator::generate( double concentrationMin, double concentrationMax, double concentration )**

Generate a RGB colour for the given concentration.

Generate a RGB colour considering the range of values a concentration can have and the value of the concentration

**Parameters**

|                             |                                                             |
|-----------------------------|-------------------------------------------------------------|
| <i>concentration</i><br>Min | The minimum of the range of values a concentration can take |
| <i>concentration</i><br>Max | The maximum of the range of values a concentration can take |
| <i>concentration</i>        | The concentration                                           |

Definition at line 11 of file RGBColourGenerator.cpp.

References convertHSVToRGB(), HUE\_MAX, HUE\_MIN, SATURATION, and VALUE.

Referenced by main(), and multiscale::analysis::SimulationClusterDetector::outputResultsToImage().

**7.157.2.5 cv::Scalar RGBColourGenerator::generate( cv::RNG & randomNumberGenerator )**

Generate a random RGB colour.

Generate a random RGB colour using the given random number generator

**Parameters**

|                                      |                         |
|--------------------------------------|-------------------------|
| <i>random</i><br>Number<br>Generator | Random number generator |
|--------------------------------------|-------------------------|

Definition at line 21 of file RGBColourGenerator.cpp.

**7.157.3 Member Data Documentation****7.157.3.1 double multiscale::RGBColourGenerator::blue [private]**

The amount of blue

Definition at line 24 of file RGBColourGenerator.hpp.

Referenced by computeRGBValues(), and convertRGBToString().

#### 7.157.3.2 double multiscale::RGBColourGenerator::green [private]

The amount of green

Definition at line 23 of file RGBColourGenerator.hpp.

Referenced by computeRGBValues(), and convertRGBToString().

#### 7.157.3.3 const int RGBColourGenerator::HUE\_MAX = 120 [static]

Definition at line 73 of file RGBColourGenerator.hpp.

Referenced by generate().

#### 7.157.3.4 const int RGBColourGenerator::HUE\_MIN = 0 [static]

Definition at line 72 of file RGBColourGenerator.hpp.

Referenced by generate().

#### 7.157.3.5 double multiscale::RGBColourGenerator::red [private]

The amount of red

Definition at line 22 of file RGBColourGenerator.hpp.

Referenced by computeRGBValues(), and convertRGBToString().

#### 7.157.3.6 const int RGBColourGenerator::SATURATION = 1 [static]

Definition at line 74 of file RGBColourGenerator.hpp.

Referenced by generate().

#### 7.157.3.7 const int RGBColourGenerator::VALUE = 1 [static]

Definition at line 75 of file RGBColourGenerator.hpp.

Referenced by generate().

The documentation for this class was generated from the following files:

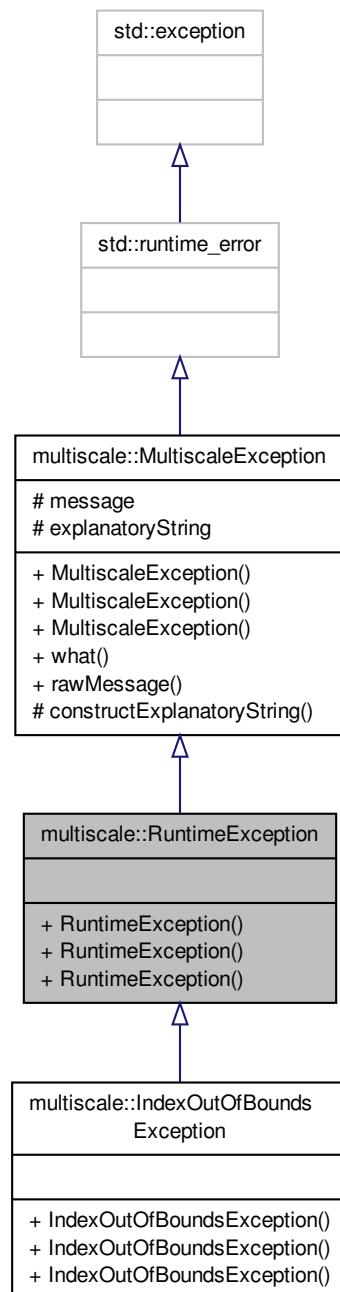
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[RGBColourGenerator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[RGBColourGenerator.cpp](#)

## 7.158 multiscale::RuntimeException Class Reference

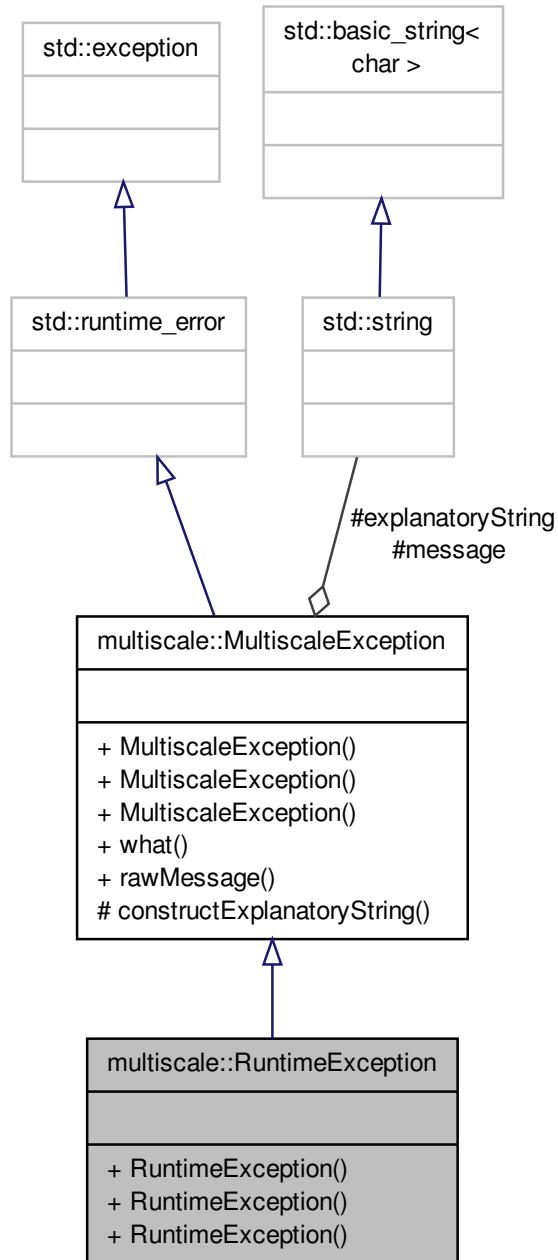
Class for representing runtime exceptions.

```
#include <RuntimeException.hpp>
```

Inheritance diagram for multiscale::RuntimeException:



Collaboration diagram for multiscale::RuntimeException:



## Public Member Functions

- `RuntimeException ()`
- `RuntimeException (const std::string &file, int line, const std::string &msg)`
- `RuntimeException (const std::string &file, int line, const char *msg)`

## Additional Inherited Members

### 7.158.1 Detailed Description

Class for representing runtime exceptions.

Definition at line 12 of file RuntimeException.hpp.

### 7.158.2 Constructor & Destructor Documentation

#### 7.158.2.1 multiscale::RuntimeException::RuntimeException( ) [inline]

Definition at line 16 of file RuntimeException.hpp.

#### 7.158.2.2 multiscale::RuntimeException::RuntimeException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 18 of file RuntimeException.hpp.

#### 7.158.2.3 multiscale::RuntimeException::RuntimeException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 23 of file RuntimeException.hpp.

The documentation for this class was generated from the following file:

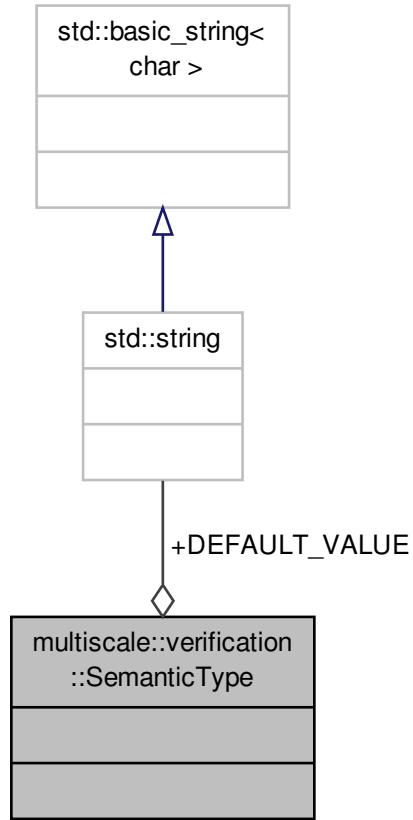
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/RuntimeException.hpp

## 7.159 multiscale::verification::SemanticType Class Reference

Enumeration for defining a semantic type.

```
#include <SemanticType.hpp>
```

Collaboration diagram for multiscale::verification::SemanticType:



## Static Public Attributes

- static const std::string `DEFAULT_VALUE` = ""

### 7.159.1 Detailed Description

Enumeration for defining a semantic type.

Definition at line 12 of file `SemanticType.hpp`.

### 7.159.2 Member Data Documentation

#### 7.159.2.1 `const std::string SemanticType::DEFAULT_VALUE = "" [static]`

The default semantic type value used when no explicit semantic type is associated to a spatial entity and/or numeric state variable

Definition at line 16 of file `SemanticType.hpp`.

Referenced by `multiscale::verification::SpatialTemporalDataReader::addNumericStateVariableToTimePoint()`, `multiscale::verification::TypeSemanticsTable::areDefaultSemanticCriteriaValues()`, `initialiseTrace()`, `multiscale::verification::TypeSemanticsTable::getSemanticType()`, `multiscale::verification::TypeSemanticsTable::setSemanticType()`.

::verification::SpatialTemporalDataReader::setSpatialEntitySemanticTypeValue(), and multiscale::verification::SpatialEntity::SpatialEntity().

The documentation for this class was generated from the following files:

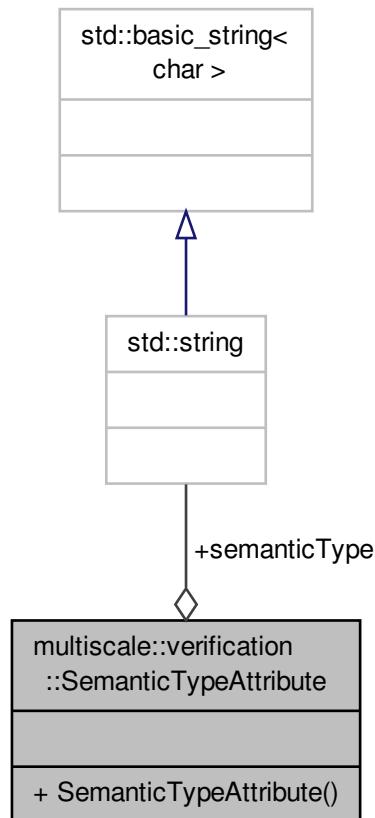
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SemanticType.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SemanticType.cpp

## 7.160 multiscale::verification::SemanticTypeAttribute Class Reference

Class for representing a semantic type attribute.

```
#include <SemanticTypeAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SemanticTypeAttribute:



### Public Member Functions

- [SemanticTypeAttribute](#) (const std::string &**semanticType**=SemanticType::DEFAULT\_VALUE)

## Public Attributes

- std::string [semanticType](#)

### 7.160.1 Detailed Description

Class for representing a semantic type attribute.

Definition at line 16 of file SemanticTypeAttribute.hpp.

### 7.160.2 Constructor & Destructor Documentation

7.160.2.1 multiscale::verification::SemanticTypeAttribute::SemanticTypeAttribute ( const std::string & *semanticType* = SemanticType::DEFAULT\_VALUE ) [inline]

Definition at line 24 of file SemanticTypeAttribute.hpp.

### 7.160.3 Member Data Documentation

7.160.3.1 std::string multiscale::verification::SemanticTypeAttribute::semanticType

The considered semantic type

Definition at line 20 of file SemanticTypeAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtType().

The documentation for this class was generated from the following file:

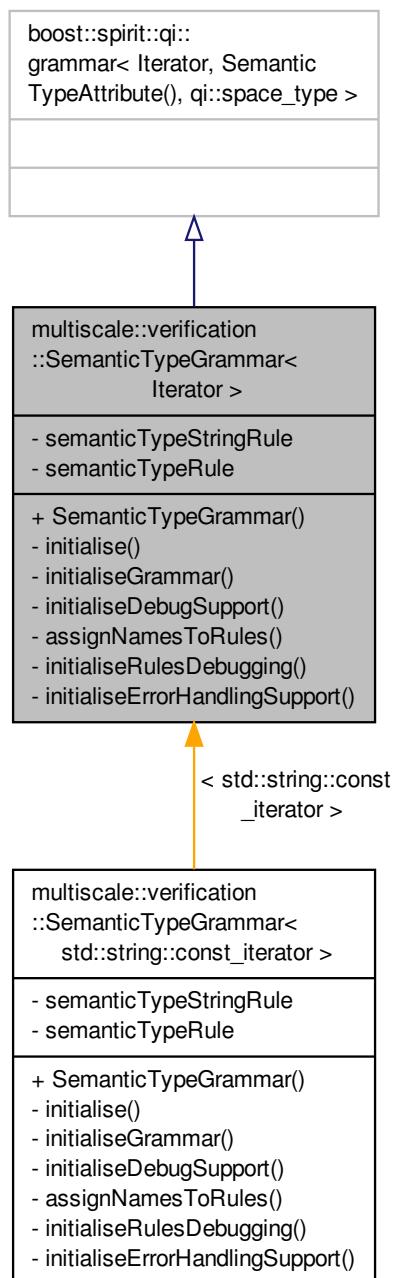
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp

## 7.161 multiscale::verification::SemanticTypeGrammar< Iterator > Class Template Reference

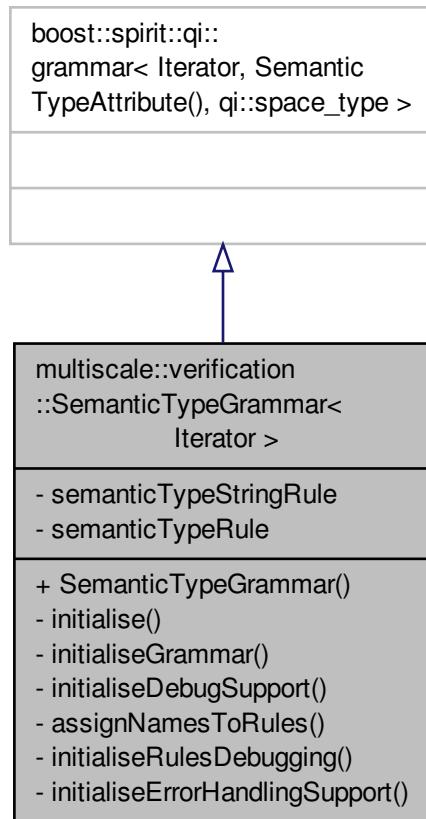
The grammar for parsing semantic type statements.

```
#include <SemanticTypeGrammar.hpp>
```

Inheritance diagram for multiscale::verification::SemanticTypeGrammar< Iterator >:



Collaboration diagram for multiscale::verification::SemanticTypeGrammar< Iterator >:



## Public Member Functions

- [SemanticTypeGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*
- void [initialiseErrorHandlingSupport \(\)](#)  
*Initialise the error handling routines.*

## Private Attributes

- [SemanticTypeStringGrammar](#)  
< Iterator > [semanticTypeStringRule](#)
- [qi::rule< Iterator,](#)  
[SemanticTypeAttribute\(\)](#),  
[qi::space\\_type > semanticTypeRule](#)

### 7.161.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::SemanticTypeGrammar< Iterator >
```

The grammar for parsing semantic type statements.

Definition at line 29 of file SemanticTypeGrammar.hpp.

### 7.161.2 Constructor & Destructor Documentation

```
7.161.2.1 template<typename Iterator > multiscale::verification::SemanticTypeGrammar< Iterator >::SemanticTypeGrammar()
```

Definition at line 19 of file SemanticTypeGrammarDefinition.hpp.

References multiscale::verification::SemanticTypeGrammar< Iterator >::initialise().

### 7.161.3 Member Function Documentation

```
7.161.3.1 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::assignNamesToRules() [private]
```

Assign names to the rules.

Definition at line 50 of file SemanticTypeGrammarDefinition.hpp.

```
7.161.3.2 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::initialise() [private]
```

Initialisation function.

Definition at line 26 of file SemanticTypeGrammarDefinition.hpp.

Referenced by multiscale::verification::SemanticTypeGrammar< Iterator >::SemanticTypeGrammar().

```
7.161.3.3 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::initialiseDebugSupport() [private]
```

Initialise debug support.

Definition at line 41 of file SemanticTypeGrammarDefinition.hpp.

```
7.161.3.4 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::initialiseErrorHandlingSupport() [private]
```

Initialise the error handling routines.

Definition at line 62 of file SemanticTypeGrammarDefinition.hpp.

---

7.161.3.5 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 34 of file SemanticTypeGrammarDefinition.hpp.

7.161.3.6 template<typename Iterator > void multiscale::verification::SemanticTypeGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 56 of file SemanticTypeGrammarDefinition.hpp.

#### 7.161.4 Member Data Documentation

7.161.4.1 template<typename Iterator> qi::rule<Iterator, SemanticTypeAttribute(), qi::space\_type> multiscale::verification::SemanticTypeGrammar< Iterator >::semanticTypeRule [private]

The rule for parsing a semantic type

Definition at line 43 of file SemanticTypeGrammar.hpp.

7.161.4.2 template<typename Iterator> SemanticTypeStringGrammar<Iterator> multiscale::verification::SemanticTypeGrammar< Iterator >::semanticTypeStringRule [private]

The rule for parsing a string representing a semantic type

Definition at line 37 of file SemanticTypeGrammar.hpp.

The documentation for this class was generated from the following files:

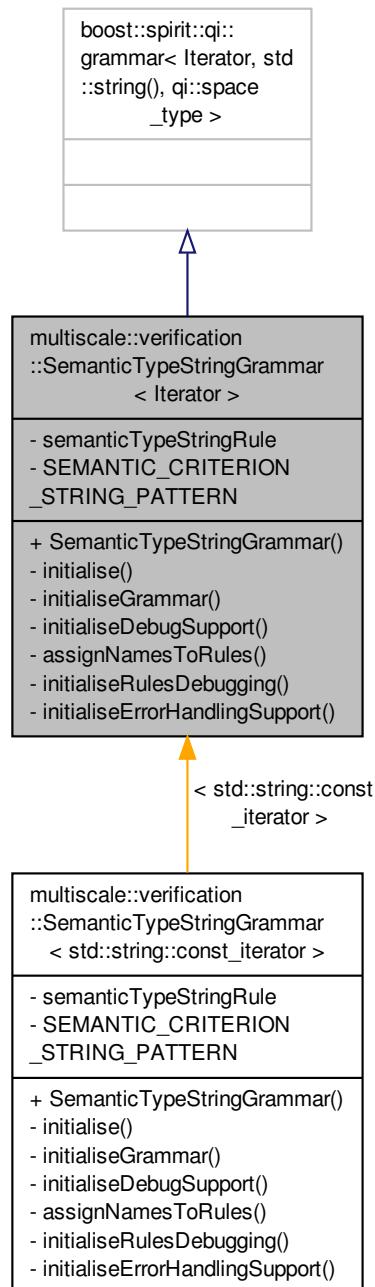
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SemanticTypeGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SemanticTypeGrammarDefinition.hpp](#)

## 7.162 multiscale::verification::SemanticTypeStringGrammar< Iterator > Class Template Reference

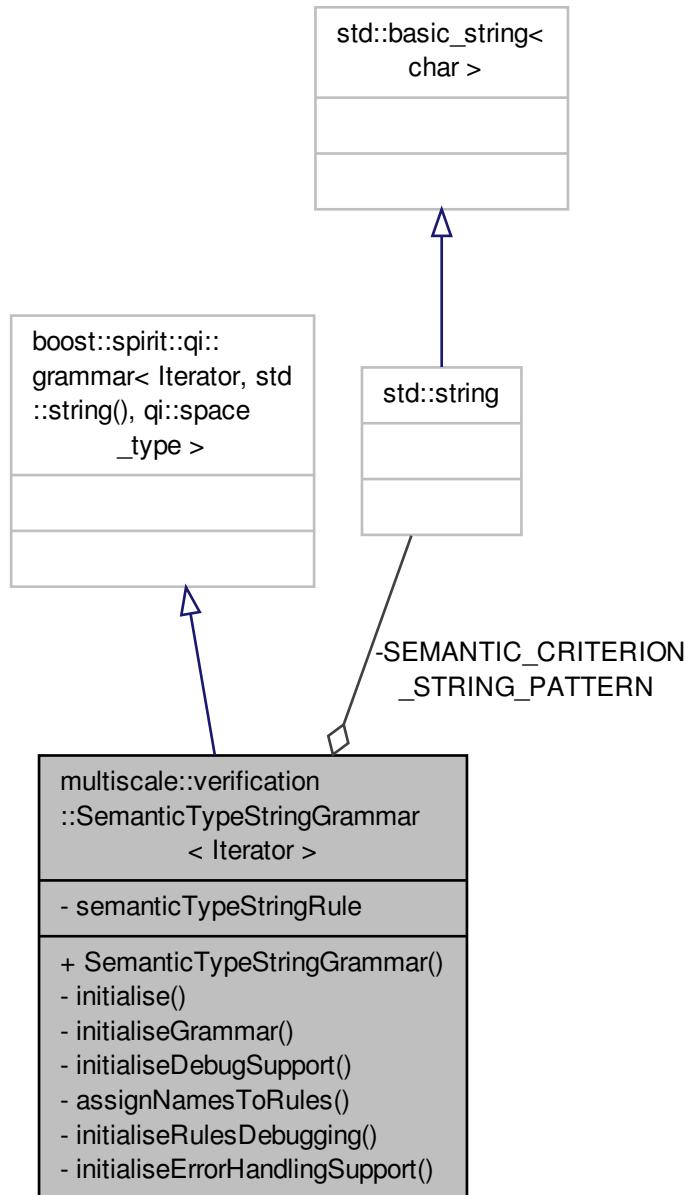
The grammar for parsing semantic type string statements.

```
#include <SemanticTypeStringGrammar.hpp>
```

Inheritance diagram for multiscale::verification::SemanticTypeStringGrammar< Iterator >:



Collaboration diagram for multiscale::verification::SemanticTypeStringGrammar< Iterator >:



## Public Member Functions

- [SemanticTypeStringGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)

*Initialisation function.*

- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*
- void [initialiseErrorHandlingSupport \(\)](#)  
*Initialise the error handling routines.*

## Private Attributes

- `qi::rule< Iterator, std::string(), qi::space_type > semanticTypeStringRule`

## Static Private Attributes

- static const std::string [SEMANTIC\\_CRITERION\\_STRING\\_PATTERN](#)

### 7.162.1 Detailed Description

`template<typename Iterator>class multiscale::verification::SemanticTypeStringGrammar< Iterator >`

The grammar for parsing semantic type string statements.

Definition at line 26 of file SemanticTypeStringGrammar.hpp.

### 7.162.2 Constructor & Destructor Documentation

7.162.2.1 `template<typename Iterator > multiscale::verification::SemanticTypeStringGrammar< Iterator >::SemanticTypeStringGrammar( )`

Definition at line 19 of file SemanticTypeStringGrammarDefinition.hpp.

References `multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialise()`.

### 7.162.3 Member Function Documentation

7.162.3.1 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::assignNamesToRules( ) [private]`

Assign names to the rules.

Definition at line 54 of file SemanticTypeStringGrammarDefinition.hpp.

7.162.3.2 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialise( ) [private]`

Initialisation function.

Definition at line 26 of file SemanticTypeStringGrammarDefinition.hpp.

Referenced by `multiscale::verification::SemanticTypeStringGrammar< Iterator >::SemanticTypeStringGrammar()`.

---

7.162.3.3 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialiseDebugSupport( ) [private]`

Initialise debug support.

Definition at line 45 of file SemanticTypeStringGrammarDefinition.hpp.

7.162.3.4 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]`

Initialise the error handling routines.

Definition at line 66 of file SemanticTypeStringGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.162.3.5 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialiseGrammar( ) [private]`

Initialise the grammar.

Definition at line 34 of file SemanticTypeStringGrammarDefinition.hpp.

7.162.3.6 `template<typename Iterator > void multiscale::verification::SemanticTypeStringGrammar< Iterator >::initialiseRulesDebugging( ) [private]`

Initialise the debugging of rules.

Definition at line 60 of file SemanticTypeStringGrammarDefinition.hpp.

## 7.162.4 Member Data Documentation

7.162.4.1 `template<typename Iterator> const std::string multiscale::verification::SemanticTypeStringGrammar< Iterator >::SEMANTIC_CRITERION_STRING_PATTERN [static], [private]`

Definition at line 64 of file SemanticTypeStringGrammar.hpp.

7.162.4.2 `template<typename Iterator> qi::rule<Iterator, std::string(), qi::space_type> multiscale::verification::SemanticTypeStringGrammar< Iterator >::semanticTypeStringRule [private]`

The rule for parsing a string representing a semantic type

Definition at line 34 of file SemanticTypeStringGrammar.hpp.

The documentation for this class was generated from the following files:

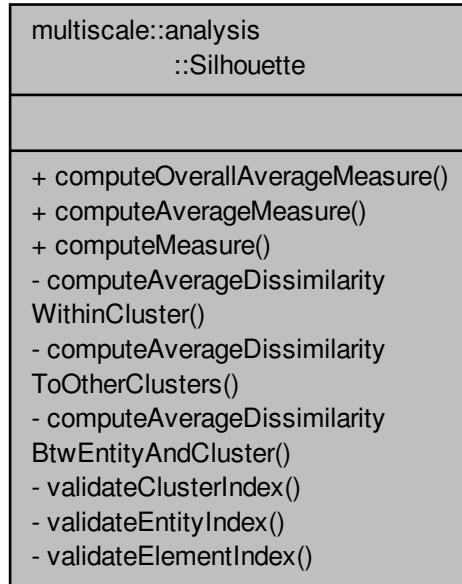
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SemanticTypeStringGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SemanticTypeStringGrammarDefinition.hpp](#)

## 7.163 multiscale::analysis::Silhouette Class Reference

Class for computing the "Silhouette" clustering index.

```
#include <Silhouette.hpp>
```

Collaboration diagram for multiscale::analysis::Silhouette:



## Static Public Member Functions

- static double `computeOverallAverageMeasure` (const std::vector< `Cluster` > &clusters)  
*Compute the overall average silhouette measure for the given collection of clusters.*
- static double `computeAverageMeasure` (std::size\_t clusterIndex, const std::vector< `Cluster` > &clusters)  
*Compute the average silhouette measure for the given cluster.*
- static double `computeMeasure` (std::size\_t entityIndex, std::size\_t clusterIndex, const std::vector< `Cluster` > &clusters)  
*Compute the silhouette measure for the given entity.*

## Static Private Member Functions

- static double `computeAverageDissimilarityWithinCluster` (std::size\_t entityIndex, std::size\_t clusterIndex, const std::vector< `Cluster` > &clusters)  
*Compute the average dissimilarity within cluster to which the entity belongs.*
- static double `computeAverageDissimilarityToOtherClusters` (std::size\_t entityIndex, std::size\_t clusterIndex, const std::vector< `Cluster` > &clusters)  
*Compute the average dissimilarity of the entity to the other clusters.*
- static double `computeAverageDissimilarityBtwEntityAndCluster` (std::size\_t entityIndex, std::size\_t entityIndex, std::size\_t clusterIndex, const std::vector< `Cluster` > &clusters)  
*Compute the average dissimilarity between entity and cluster.*
- static void `validateClusterIndex` (std::size\_t clusterIndex, std::size\_t totalNrOfClusters)  
*Check if the provided cluster index is valid.*
- static void `validateEntityIndex` (std::size\_t entityIndex, std::size\_t totalNrOfEntities)

*Check if the provided entity index is valid.*

- static void [validateElementIndex](#) (std::size\_t elementIndex, std::size\_t totalNrOfElements)

*Check if the provided element index is valid.*

### 7.163.1 Detailed Description

Class for computing the "Silhouette" clustering index.

Definition at line 12 of file Silhouette.hpp.

### 7.163.2 Member Function Documentation

7.163.2.1 double [Silhouette::computeAverageDissimilarityBtwEntityAndCluster](#) ( std::size\_t *entityIndex*, std::size\_t *entityClusterIndex*, std::size\_t *clusterIndex*, const std::vector< Cluster > & *clusters* ) [static], [private]

Compute the average dissimilarity between entity and cluster.

Parameters

|                           |                                                                           |
|---------------------------|---------------------------------------------------------------------------|
| <i>entityIndex</i>        | The index of the entity in the cluster for which the distance is computed |
| <i>entityClusterIndex</i> | The index of the cluster to which the entity belongs                      |
| <i>clusterIndex</i>       | The index of the cluster to which the average distance is computed        |
| <i>clusters</i>           | Collection of all clusters                                                |

Definition at line 91 of file Silhouette.cpp.

References multiscale::Geometry2D::distanceBtwPoints().

Referenced by [computeAverageDissimilarityToOtherClusters\(\)](#).

7.163.2.2 double [Silhouette::computeAverageDissimilarityToOtherClusters](#) ( std::size\_t *entityIndex*, std::size\_t *clusterIndex*, const std::vector< Cluster > & *clusters* ) [static], [private]

Compute the average dissimilarity of the entity to the other clusters.

Compute the average dissimilarity of the entity to the other clusters (i.e. clusters which are different from the cluster to which the entity belongs)

Parameters

|                     |                                                                                     |
|---------------------|-------------------------------------------------------------------------------------|
| <i>entityIndex</i>  | The index of the entity in the cluster for which the silhouette measure is computed |
| <i>clusterIndex</i> | The index of the cluster to which the entity belongs                                |
| <i>clusters</i>     | Collection of all clusters                                                          |

Definition at line 71 of file Silhouette.cpp.

References [computeAverageDissimilarityBtwEntityAndCluster\(\)](#).

Referenced by [computeMeasure\(\)](#).

7.163.2.3 double [Silhouette::computeAverageDissimilarityWithinCluster](#) ( std::size\_t *entityIndex*, std::size\_t *clusterIndex*, const std::vector< Cluster > & *clusters* ) [static], [private]

Compute the average dissimilarity within cluster to which the entity belongs.

**Parameters**

|                     |                                                                                     |
|---------------------|-------------------------------------------------------------------------------------|
| <i>entityIndex</i>  | The index of the entity in the cluster for which the silhouette measure is computed |
| <i>clusterIndex</i> | The index of the cluster to which the entity belongs                                |
| <i>clusters</i>     | Collection of all clusters                                                          |

Definition at line 53 of file Silhouette.cpp.

References multiscale::Geometry2D::distanceBtwPoints().

Referenced by computeMeasure().

**7.163.2.4 double Silhouette::computeAverageMeasure ( std::size\_t *clusterIndex*, const std::vector< Cluster > & *clusters* ) [static]**

Compute the average silhouette measure for the given cluster.

**Parameters**

|                     |                                                                               |
|---------------------|-------------------------------------------------------------------------------|
| <i>clusterIndex</i> | The index of the cluster for which the average silhouette measure is computed |
| <i>clusters</i>     | Collection of all clusters                                                    |

Definition at line 25 of file Silhouette.cpp.

References computeMeasure(), and validateClusterIndex().

Referenced by computeOverallAverageMeasure().

**7.163.2.5 double Silhouette::computeMeasure ( std::size\_t *entityIndex*, std::size\_t *clusterIndex*, const std::vector< Cluster > & *clusters* ) [static]**

Compute the silhouette measure for the given entity.

**Parameters**

|                     |                                                                                     |
|---------------------|-------------------------------------------------------------------------------------|
| <i>entityIndex</i>  | The index of the entity in the cluster for which the silhouette measure is computed |
| <i>clusterIndex</i> | The index of the cluster to which the entity belongs                                |
| <i>clusters</i>     | Collection of all clusters                                                          |

Definition at line 41 of file Silhouette.cpp.

References computeAverageDissimilarityToOtherClusters(), computeAverageDissimilarityWithinCluster(), validateClusterIndex(), and validateEntityIndex().

Referenced by computeAverageMeasure().

**7.163.2.6 double Silhouette::computeOverallAverageMeasure ( const std::vector< Cluster > & *clusters* ) [static]**

Compute the overall average silhouette measure for the given collection of clusters.

**Parameters**

|                 |                            |
|-----------------|----------------------------|
| <i>clusters</i> | Collection of all clusters |
|-----------------|----------------------------|

Definition at line 13 of file Silhouette.cpp.

References computeAverageMeasure().

Referenced by multiscale::analysis::ClusterDetector::computeClusterednessIndex().

**7.163.2.7 void Silhouette::validateClusterIndex ( std::size\_t *clusterIndex*, std::size\_t *totalNrOfClusters* ) [static], [private]**

Check if the provided cluster index is valid.

The cluster index `clusterIndex` (0-based indexing) is valid if and only if:  $0 \leq \text{clusterIndex} < \text{total number of clusters}$

## Parameters

|                          |                              |
|--------------------------|------------------------------|
| <i>clusterIndex</i>      | The index of the cluster     |
| <i>totalNrOfClusters</i> | The total number of clusters |

Definition at line 109 of file Silhouette.cpp.

References validateElementIndex().

Referenced by computeAverageMeasure(), and computeMeasure().

**7.163.2.8 void Silhouette::validateElementIndex ( std::size\_t elementIndex, std::size\_t totalNrOfElements ) [static], [private]**

Check if the provided element index is valid.

The element index *elementIndex* (0-based indexing) is valid if and only if:  $0 \leq \text{elementIndex} < \text{total number of elements}$

## Parameters

|                          |                              |
|--------------------------|------------------------------|
| <i>elementIndex</i>      | The index of the element     |
| <i>totalNrOfElements</i> | The total number of elements |

Definition at line 117 of file Silhouette.cpp.

References MS\_throw.

Referenced by validateClusterIndex(), and validateEntityIndex().

**7.163.2.9 void Silhouette::validateEntityIndex ( std::size\_t entityIndex, std::size\_t totalNrOfEntities ) [static], [private]**

Check if the provided entity index is valid.

The entity index *entityIndex* (0-based indexing) is valid if and only if:  $0 \leq \text{entityIndex} < \text{total number of entities}$

## Parameters

|                          |                              |
|--------------------------|------------------------------|
| <i>entityIndex</i>       | The index of the entity      |
| <i>totalNrOfEntities</i> | The total number of entities |

Definition at line 113 of file Silhouette.cpp.

References validateElementIndex().

Referenced by computeMeasure().

The documentation for this class was generated from the following files:

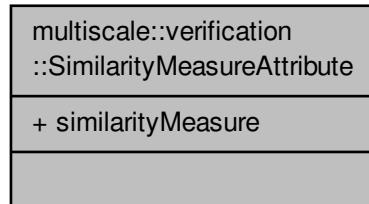
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[Silhouette.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[Silhouette.cpp](#)

## 7.164 multiscale::verification::SimilarityMeasureAttribute Class Reference

Class for representing a similarity measure attribute.

```
#include <SimilarityMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SimilarityMeasureAttribute:



## Public Attributes

- [SimilarityMeasureType similarityMeasure](#)

### 7.164.1 Detailed Description

Class for representing a similarity measure attribute.

Definition at line 28 of file SimilarityMeasureAttribute.hpp.

### 7.164.2 Member Data Documentation

#### 7.164.2.1 [SimilarityMeasureType multiscale::verification::SimilarityMeasureAttribute::similarityMeasure](#)

The similarity measure

Definition at line 32 of file SimilarityMeasureAttribute.hpp.

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

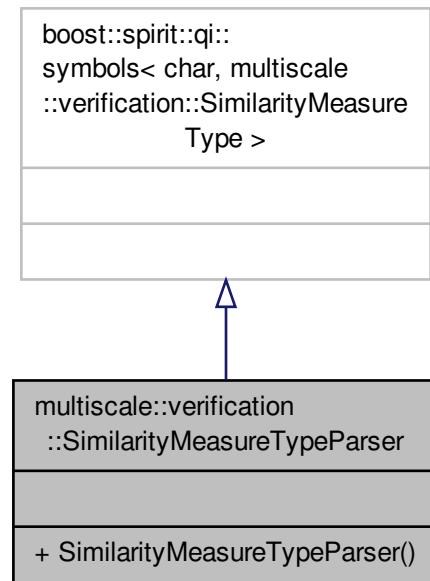
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityMeasureAttribute.hpp](#)

## 7.165 [multiscale::verification::SimilarityMeasureTypeParser Struct Reference](#)

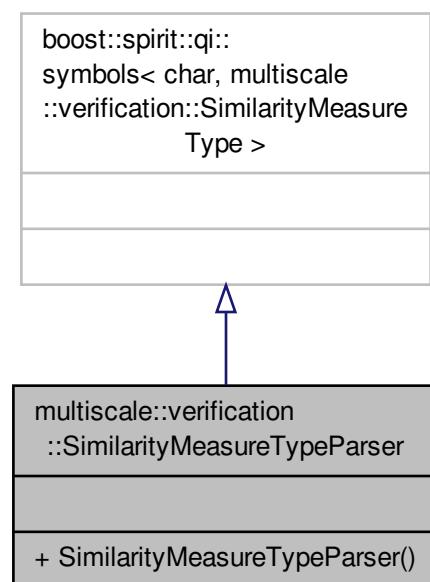
Symbol table and parser for the similarity measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::SimilarityMeasureTypeParser:



Collaboration diagram for multiscale::verification::SimilarityMeasureTypeParser:



## Public Member Functions

- [SimilarityMeasureTypeParser \(\)](#)

### 7.165.1 Detailed Description

Symbol table and parser for the similarity measure type.

Definition at line 156 of file [SymbolTables.hpp](#).

### 7.165.2 Constructor & Destructor Documentation

#### 7.165.2.1 multiscale::verification::SimilarityMeasureTypeParser::SimilarityMeasureTypeParser( ) [inline]

Definition at line 159 of file [SymbolTables.hpp](#).

References [multiscale::verification::Opposite](#), and [multiscale::verification::Similar](#).

The documentation for this struct was generated from the following file:

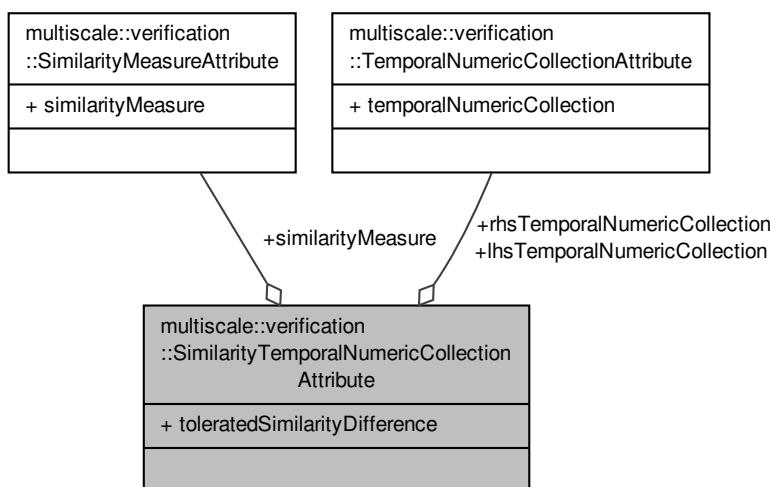
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.166 multiscale::verification::SimilarityTemporalNumericCollectionAttribute Class Reference

Class for representing a similarity temporal numeric collection attribute.

```
#include <SimilarityTemporalNumericCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SimilarityTemporalNumericCollectionAttribute:



## Public Attributes

- [SimilarityMeasureAttribute similarityMeasure](#)
- [TemporalNumericCollectionAttribute lhsTemporalNumericCollection](#)
- [TemporalNumericCollectionAttribute rhsTemporalNumericCollection](#)
- double toleratedSimilarityDifference

### 7.166.1 Detailed Description

Class for representing a similarity temporal numeric collection attribute.

Definition at line 16 of file [SimilarityTemporalNumericCollectionAttribute.hpp](#).

### 7.166.2 Member Data Documentation

#### 7.166.2.1 TemporalNumericCollectionAttribute multiscale::verification::SimilarityTemporalNumericCollectionAttribute<::lhsTemporalNumericCollection

The left hand side temporal numeric collection

Definition at line 23 of file [SimilarityTemporalNumericCollectionAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

#### 7.166.2.2 TemporalNumericCollectionAttribute multiscale::verification::SimilarityTemporalNumericCollectionAttribute<::rhsTemporalNumericCollection

The right hand side temporal numeric collection

Definition at line 25 of file [SimilarityTemporalNumericCollectionAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

#### 7.166.2.3 SimilarityMeasureAttribute multiscale::verification::SimilarityTemporalNumericCollectionAttribute::similarityMeasure

The similarity measure

Definition at line 21 of file [SimilarityTemporalNumericCollectionAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

#### 7.166.2.4 double multiscale::verification::SimilarityTemporalNumericCollectionAttribute::toleratedSimilarityDifference

The tolerated similarity difference

Definition at line 27 of file [SimilarityTemporalNumericCollectionAttribute.hpp](#).

Referenced by [multiscale::verification::LogicPropertyVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityTemporalNumericCollectionAttribute.hpp](#)

## 7.167 multiscale::analysis::SimulationClusterDetector Class Reference

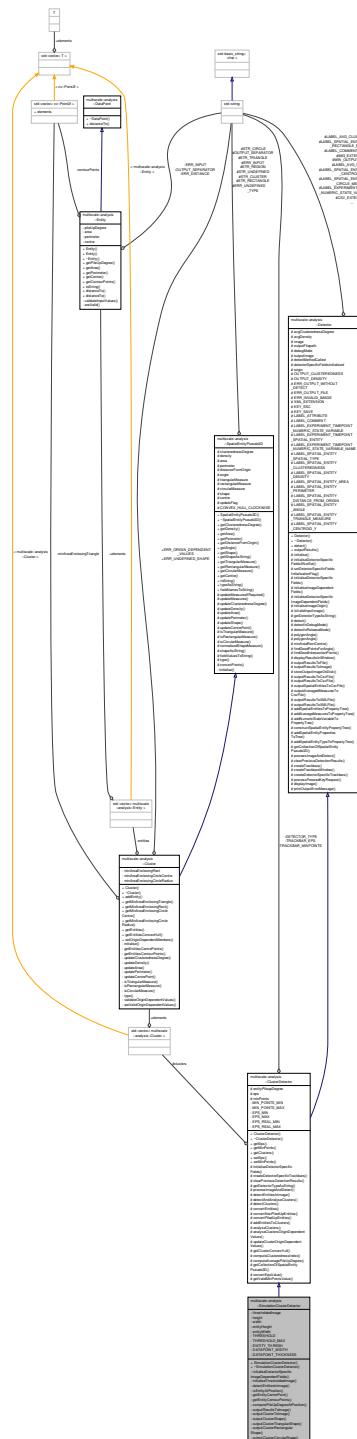
Class for detecting clusters in 2D images obtained from simulations.

```
#include <SimulationClusterDetector.hpp>
```

Inheritance diagram for multiscale::analysis::SimulationClusterDetector



## Collaboration diagram for multiscale::analysis::SimulationClusterDetector::



## Public Member Functions

- `SimulationClusterDetector` (unsigned int `height`, unsigned int `width`, int `maxPileupNumber`, double `maxPileupIntensity`, bool `debugMode=false`)
  - `~SimulationClusterDetector ()`

## Private Member Functions

- void `initialiseDetectorSpecificImageDependentFields ()` override  
*Initialise the image dependent values.*
- void `initialiseThresholdedImage ()`  
*Initialise the thresholdedImage field.*
- void `detectEntitiesInImage (std::vector< Entity > &entities)` override  
*Detect the entities in the image.*
- bool `isEntityAtPosition (int x, int y)`  
*Check if there is an entity in the image at the given position.*
- cv::Point2f `getEntityCentrePoint (int x, int y)`  
*Get the point representing the centre of the entity.*
- std::vector< cv::Point2f > `getEntityContourPoints (int x, int y)`  
*Get the points representing the contour of the entity.*
- unsigned int `computePileUpDegreeAtPosition (int x, int y)`  
*Compute the pile up degree at the given position.*
- void `outputResultsToImage ()` override  
*Display clusters on image.*
- void `outputClusterToImage (Cluster &cluster, cv::Scalar colour, cv::Mat &image)`  
*Display cluster on the image.*
- void `outputClusterShape (Cluster &cluster, cv::Scalar colour, cv::Mat &image)`  
*Draw the best matching shape (triangular, rectangular, circular) of the cluster on the image.*
- void `outputClusterTriangularShape (Cluster &cluster, cv::Scalar colour, cv::Mat &image)`  
*Draw the best matching triangular shape of the cluster on the image.*
- void `outputClusterRectangularShape (Cluster &cluster, cv::Scalar colour, cv::Mat &image)`  
*Draw the best matching rectangular shape of the cluster on the image.*
- void `outputClusterCircularShape (Cluster &cluster, cv::Scalar colour, cv::Mat &image)`  
*Draw the best matching circular shape of the cluster on the image.*

## Private Attributes

- cv::Mat `thresholdedImage`
- unsigned int `height`
- unsigned int `width`
- double `entityHeight`
- double `entityWidth`

## Static Private Attributes

- static const int `THRESHOLD` = 1
- static const int `THRESHOLD_MAX` = 255
- static const int `ENTITY_THRESH` = 200
- static const int `DATAPOINT_WIDTH` = 10
- static const int `DATAPOINT_THICKNESS` = -1

## Additional Inherited Members

### 7.167.1 Detailed Description

Class for detecting clusters in 2D images obtained from simulations.

Definition at line 15 of file SimulationClusterDetector.hpp.

## 7.167.2 Constructor & Destructor Documentation

7.167.2.1 `SimulationClusterDetector::SimulationClusterDetector ( unsigned int height, unsigned int width, int maxPileupNumber, double maxPileupIntensity, bool debugMode = false )`

### Parameters

|                                 |                                                                                  |
|---------------------------------|----------------------------------------------------------------------------------|
| <i>height</i>                   | Height of the grid used in the simulation                                        |
| <i>width</i>                    | Width of the grid used in the simulation                                         |
| <i>debugMode</i>                | Flag indicating if detector should run in debug mode or not                      |
| <i>maxPileup←<br/>Number</i>    | The maximum number of entities which can occupy a grid position at the same time |
| <i>maxPileup←<br/>Intensity</i> | The grayscale intensity of a maximally piled up grid position                    |

Definition at line 10 of file SimulationClusterDetector.cpp.

References entityHeight, entityWidth, height, and width.

7.167.2.2 `SimulationClusterDetector::~SimulationClusterDetector ( )`

Definition at line 20 of file SimulationClusterDetector.cpp.

## 7.167.3 Member Function Documentation

7.167.3.1 `unsigned int SimulationClusterDetector::computePileUpDegreeAtPosition ( int x, int y ) [private]`

Compute the pile up degree at the given position.

### Parameters

|          |                        |
|----------|------------------------|
| <i>x</i> | Coordinate for Ox axis |
| <i>y</i> | Coordinate for Oy axis |

Definition at line 76 of file SimulationClusterDetector.cpp.

References entityHeight, multiscale::analysis::ClusterDetector::entityPileupDegree, entityWidth, and multiscale::analysis::Detector::image.

Referenced by detectEntitiesInImage().

7.167.3.2 `void SimulationClusterDetector::detectEntitiesInImage ( std::vector< Entity > & entities ) [override], [private], [virtual]`

Detect the entities in the image.

Detect the entities in the image, compute their centre point and degree of pile up

### Parameters

|                 |                                |
|-----------------|--------------------------------|
| <i>entities</i> | Entities detected in the image |
|-----------------|--------------------------------|

Implements [multiscale::analysis::ClusterDetector](#).

Definition at line 33 of file SimulationClusterDetector.cpp.

References computePileUpDegreeAtPosition(), entityHeight, entityWidth, getEntityCentrePoint(), getEntityContourPoints(), height, isEntityAtPosition(), and width.

7.167.3.3 cv::Point2f SimulationClusterDetector::getEntityCentrePoint( int x, int y ) [private]

Get the point representing the centre of the entity.

**Parameters**

|          |               |
|----------|---------------|
| <i>x</i> | Ox coordinate |
| <i>y</i> | Oy coordinate |

Definition at line 57 of file SimulationClusterDetector.cpp.

References entityHeight, and entityWidth.

Referenced by detectEntitiesInImage().

#### 7.167.3.4 std::vector< cv::Point2f > SimulationClusterDetector::getEntityContourPoints ( int *x*, int *y* ) [private]

Get the points representing the contour of the entity.

**Parameters**

|          |               |
|----------|---------------|
| <i>x</i> | Ox coordinate |
| <i>y</i> | Oy coordinate |

Definition at line 64 of file SimulationClusterDetector.cpp.

References entityHeight, and entityWidth.

Referenced by detectEntitiesInImage().

#### 7.167.3.5 void SimulationClusterDetector::initialiseDetectorSpecificImageDependentFields ( ) [override], [private], [virtual]

Initialise the image dependent values.

Implements [multiscale::analysis::Detector](#).

Definition at line 22 of file SimulationClusterDetector.cpp.

References entityHeight, entityWidth, height, multiscale::analysis::Detector::image, initialiseThresholdedImage(), and width.

#### 7.167.3.6 void SimulationClusterDetector::initialiseThresholdedImage ( ) [private]

Initialise the thresholdedImage field.

Definition at line 29 of file SimulationClusterDetector.cpp.

References multiscale::analysis::Detector::image, THRESHOLD, THRESHOLD\_MAX, and thresholdedImage.

Referenced by initialiseDetectorSpecificImageDependentFields().

#### 7.167.3.7 bool SimulationClusterDetector::isEntityAtPosition ( int *x*, int *y* ) [private]

Check if there is an entity in the image at the given position.

**Parameters**

|          |                        |
|----------|------------------------|
| <i>x</i> | Coordinate for Ox axis |
| <i>y</i> | Coordinate for Oy axis |

Definition at line 49 of file SimulationClusterDetector.cpp.

References ENTITY\_THRESH, entityHeight, entityWidth, and thresholdedImage.

Referenced by detectEntitiesInImage().

```
7.167.3.8 void SimulationClusterDetector::outputClusterCircularShape (Cluster & cluster, cv::Scalar colour, cv::Mat & image
) [private]
```

Draw the best matching circular shape of the cluster on the image.

**Parameters**

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>cluster</i> | Cluster                                                       |
| <i>colour</i>  | Colour associated to all entities in the cluster              |
| <i>image</i>   | The image on which to display the cluster related information |

Definition at line 152 of file SimulationClusterDetector.cpp.

References DATAPOINT\_WIDTH, multiscale::analysis::Cluster::getMinAreaEnclosingCircleCentre(), and multiscale::analysis::Cluster::getMinAreaEnclosingCircleRadius().

Referenced by outputClusterShape().

**7.167.3.9 void SimulationClusterDetector::outputClusterRectangularShape ( Cluster & *cluster*, cv::Scalar *colour*, cv::Mat & *image* ) [private]**

Draw the best matching rectangular shape of the cluster on the image.

**Parameters**

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>cluster</i> | Cluster                                                       |
| <i>colour</i>  | Colour associated to all entities in the cluster              |
| <i>image</i>   | The image on which to display the cluster related information |

Definition at line 142 of file SimulationClusterDetector.cpp.

References DATAPOINT\_WIDTH, and multiscale::analysis::Cluster::getMinAreaEnclosingRect().

Referenced by outputClusterShape().

**7.167.3.10 void SimulationClusterDetector::outputClusterShape ( Cluster & *cluster*, cv::Scalar *colour*, cv::Mat & *image* ) [private]**

Draw the best matching shape (triangular, rectangular, circular) of the cluster on the image.

**Parameters**

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>cluster</i> | Cluster                                                       |
| <i>colour</i>  | Colour associated to all entities in the cluster              |
| <i>image</i>   | The image on which to display the cluster related information |

Definition at line 110 of file SimulationClusterDetector.cpp.

References multiscale::analysis::Circle, multiscale::analysis::Cluster::ERR\_UNDEFINED\_SHAPE, multiscale::analysis::SpatialEntityPseudo3D::getShape(), MS\_throw, outputClusterCircularShape(), outputClusterRectangularShape(), outputClusterTriangularShape(), multiscale::analysis::Rectangle, and multiscale::analysis::Triangle.

Referenced by outputClusterToImage().

**7.167.3.11 void SimulationClusterDetector::outputClusterToImage ( Cluster & *cluster*, cv::Scalar *colour*, cv::Mat & *image* ) [private]**

Display cluster on the image.

**Parameters**

|                |                                                  |
|----------------|--------------------------------------------------|
| <i>cluster</i> | Cluster                                          |
| <i>colour</i>  | Colour associated to all entities in the cluster |

|              |                                                               |
|--------------|---------------------------------------------------------------|
| <i>image</i> | The image on which to display the cluster related information |
|--------------|---------------------------------------------------------------|

Definition at line 100 of file SimulationClusterDetector.cpp.

References DATAPOINT\_THICKNESS, DATAPOINT\_WIDTH, multiscale::analysis::Cluster::getEntities(), and outputClusterShape().

Referenced by outputResultsToImage().

#### 7.167.3.12 void SimulationClusterDetector::outputClusterTriangularShape ( Cluster & *cluster*, cv::Scalar *colour*, cv::Mat & *image* ) [private]

Draw the best matching triangular shape of the cluster on the image.

Parameters

|                |                                                               |
|----------------|---------------------------------------------------------------|
| <i>cluster</i> | Cluster                                                       |
| <i>colour</i>  | Colour associated to all entities in the cluster              |
| <i>image</i>   | The image on which to display the cluster related information |

Definition at line 132 of file SimulationClusterDetector.cpp.

References DATAPOINT\_WIDTH, and multiscale::analysis::Cluster::getMinAreaEnclosingTriangle().

Referenced by outputClusterShape().

#### 7.167.3.13 void SimulationClusterDetector::outputResultsToImage ( ) [override], [private], [virtual]

Display clusters on image.

Implements [multiscale::analysis::Detector](#).

Definition at line 85 of file SimulationClusterDetector.cpp.

References multiscale::analysis::ClusterDetector::clusters, multiscale::RGBColourGenerator::generate(), multiscale::analysis::Detector::image, outputClusterToImage(), and multiscale::analysis::Detector::outputImage.

### 7.167.4 Member Data Documentation

#### 7.167.4.1 const int SimulationClusterDetector::DATAPOINT\_THICKNESS = -1 [static], [private]

Definition at line 139 of file SimulationClusterDetector.hpp.

Referenced by outputClusterToImage().

#### 7.167.4.2 const int SimulationClusterDetector::DATAPOINT\_WIDTH = 10 [static], [private]

Definition at line 138 of file SimulationClusterDetector.hpp.

Referenced by outputClusterCircularShape(), outputClusterRectangularShape(), outputClusterToImage(), and outputClusterTriangularShape().

#### 7.167.4.3 const int SimulationClusterDetector::ENTITY\_THRESH = 200 [static], [private]

Definition at line 136 of file SimulationClusterDetector.hpp.

Referenced by isEntityAtPosition().

**7.167.4.4 double multiscale::analysis::SimulationClusterDetector::entityHeight [private]**

Height of an entity

Definition at line 24 of file SimulationClusterDetector.hpp.

Referenced by computePileUpDegreeAtPosition(), detectEntitiesInImage(), getEntityCentrePoint(), getEntityContourPoints(), initialiseDetectorSpecificImageDependentFields(), isEntityAtPosition(), and SimulationClusterDetector().

**7.167.4.5 double multiscale::analysis::SimulationClusterDetector::entityWidth [private]**

Width of an entity

Definition at line 25 of file SimulationClusterDetector.hpp.

Referenced by computePileUpDegreeAtPosition(), detectEntitiesInImage(), getEntityCentrePoint(), getEntityContourPoints(), initialiseDetectorSpecificImageDependentFields(), isEntityAtPosition(), and SimulationClusterDetector().

**7.167.4.6 unsigned int multiscale::analysis::SimulationClusterDetector::height [private]**

Height of the grid used in the simulation

Definition at line 21 of file SimulationClusterDetector.hpp.

Referenced by detectEntitiesInImage(), initialiseDetectorSpecificImageDependentFields(), and SimulationClusterDetector().

**7.167.4.7 const int SimulationClusterDetector::THRESHOLD = 1 [static], [private]**

Definition at line 133 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage().

**7.167.4.8 const int SimulationClusterDetector::THRESHOLD\_MAX = 255 [static], [private]**

Definition at line 134 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage().

**7.167.4.9 cv::Mat multiscale::analysis::SimulationClusterDetector::thresholdedImage [private]**

Thresholded version of the image

Definition at line 19 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage(), and isEntityAtPosition().

**7.167.4.10 unsigned int multiscale::analysis::SimulationClusterDetector::width [private]**

Width of the grid used in the simulation

Definition at line 22 of file SimulationClusterDetector.hpp.

Referenced by detectEntitiesInImage(), initialiseDetectorSpecificImageDependentFields(), and SimulationClusterDetector().

The documentation for this class was generated from the following files:

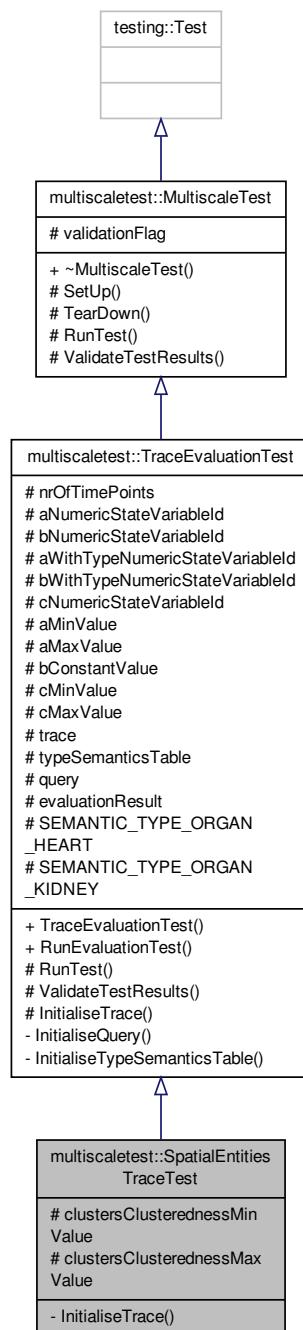
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/[SimulationClusterDetector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/[SimulationClusterDetector.cpp](#)

## 7.168 multiscaletest::SpatialEntitiesTraceTest Class Reference

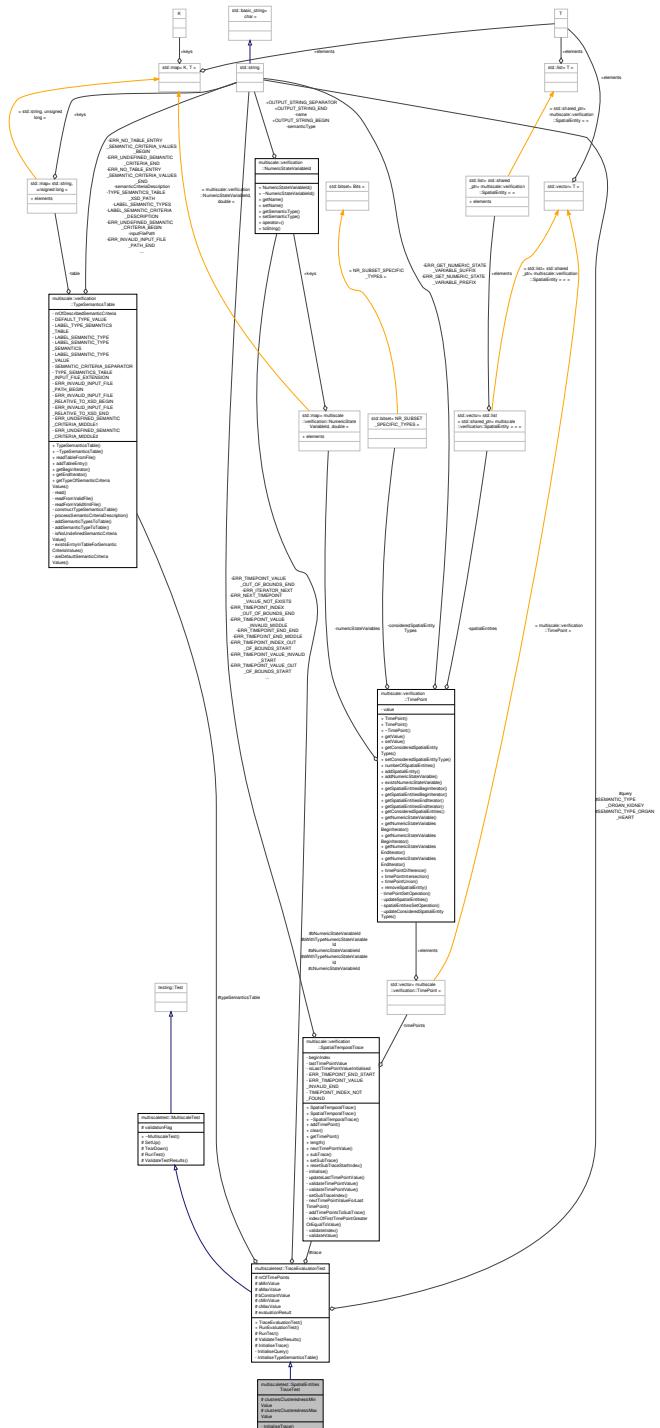
Class for testing evaluation of spatial entities-only traces.

```
#include <SpatialEntitiesTraceTest.hpp>
```

Inheritance diagram for multiscaletest::SpatialEntitiesTraceTest:



## Collaboration diagram for multiscaletest::SpatialEntitiesTraceTest:



## Protected Attributes

- double *clustersClusterednessMinValue*
  - double *clustersClusterednessMaxValue*

## Private Member Functions

- virtual void [InitialiseTrace \(\) override](#)

*Initialise the trace.*

## Additional Inherited Members

### 7.168.1 Detailed Description

Class for testing evaluation of spatial entities-only traces.

Definition at line 25 of file SpatialEntitiesTraceTest.hpp.

### 7.168.2 Member Function Documentation

#### 7.168.2.1 void multiscaletest::SpatialEntitiesTraceTest::InitialiseTrace ( ) [override], [private], [virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 39 of file SpatialEntitiesTraceTest.hpp.

### 7.168.3 Member Data Documentation

#### 7.168.3.1 double multiscaletest::SpatialEntitiesTraceTest::clustersClusterednessMaxValue [protected]

The maximum clusteredness value for the cluster spatial entity type

Definition at line 30 of file SpatialEntitiesTraceTest.hpp.

#### 7.168.3.2 double multiscaletest::SpatialEntitiesTraceTest::clustersClusterednessMinValue [protected]

The minimum clusteredness value for the cluster spatial entity type

Definition at line 29 of file SpatialEntitiesTraceTest.hpp.

The documentation for this class was generated from the following file:

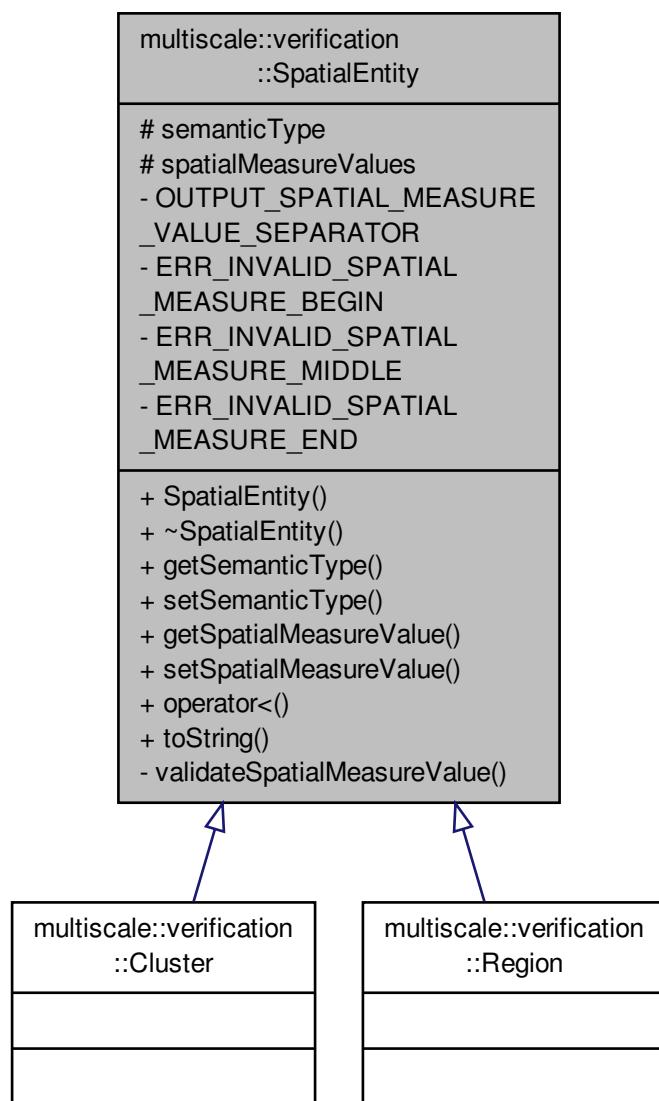
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[SpatialEntitiesTraceTest.hpp](#)

## 7.169 multiscale::verification::SpatialEntity Class Reference

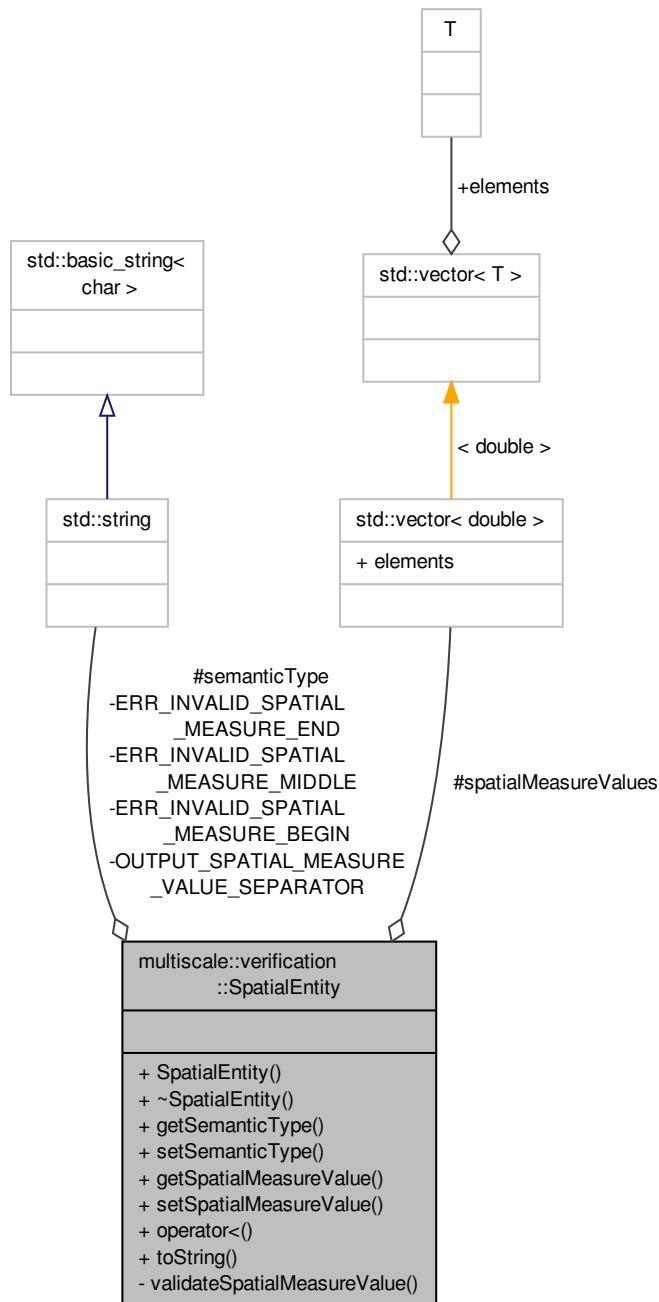
Class for representing a pseudo-3D spatial entity.

```
#include <SpatialEntity.hpp>
```

Inheritance diagram for multiscale::verification::SpatialEntity:



Collaboration diagram for multiscale::verification::SpatialEntity:



## Public Member Functions

- [SpatialEntity \(\)](#)
- [~SpatialEntity \(\)](#)
- [std::string getSemanticType \(\) const](#)  
*Get the semantic type.*
- [void setSemanticType \(const std::string &semanticType\)](#)

- Set the value of the semantic type.
- double `getSpatialMeasureValue` (const `SpatialMeasureType` &spatialMeasureType) const
  - Get the value of the given spatial measure.*
- void `setSpatialMeasureValue` (const `SpatialMeasureType` &spatialMeasureType, double spatialMeasureValue)
  - Set the value of the given spatial measure.*
- bool `operator<` (const `SpatialEntity` &rhsSpatialEntity) const
  - Overload the "<" operator for spatial entities.*
- std::string `toString` () const
  - Return a string representation of the spatial entity contents.*

## Protected Attributes

- std::string `semanticType`
- std::vector< double > `spatialMeasureValues`

## Private Member Functions

- void `validateSpatialMeasureValue` (double spatialMeasureValue, const `SpatialMeasureType` &spatialMeasureType)
  - Check if the provided value is valid considering the given spatial measure.*

## Static Private Attributes

- static const std::string `OUTPUT_SPATIAL_MEASURE_VALUE_SEPARATOR` = ", "
- static const std::string `ERR_INVALID_SPATIAL_MEASURE_BEGIN` = "The provided spatial measure value ("
- static const std::string `ERR_INVALID_SPATIAL_MEASURE_MIDDLE` = "is invalid for the given spatial measure type ("
- static const std::string `ERR_INVALID_SPATIAL_MEASURE_END` = "). Please change."

### 7.169.1 Detailed Description

Class for representing a pseudo-3D spatial entity.

Definition at line 17 of file `SpatialEntity.hpp`.

### 7.169.2 Constructor & Destructor Documentation

#### 7.169.2.1 `SpatialEntity::SpatialEntity()`

Definition at line 11 of file `SpatialEntity.cpp`.

References `multiscale::verification::SemanticType::DEFAULT_VALUE`, and `multiscale::verification::NR_SPATIAL_MEASURE_TYPES`.

#### 7.169.2.2 `SpatialEntity::~SpatialEntity()`

Definition at line 16 of file `SpatialEntity.cpp`.

### 7.169.3 Member Function Documentation

#### 7.169.3.1 std::string SpatialEntity::getSemanticType ( ) const

Get the semantic type.

Definition at line 18 of file SpatialEntity.cpp.

#### 7.169.3.2 double SpatialEntity::getSpatialMeasureValue ( const SpatialMeasureType & *spatialMeasureType* ) const

Get the value of the given spatial measure.

##### Parameters

|                           |                                                     |
|---------------------------|-----------------------------------------------------|
| <i>spatialMeasureType</i> | The spatial measure for which the value is returned |
|---------------------------|-----------------------------------------------------|

Definition at line 26 of file SpatialEntity.cpp.

References multiscale::verification::spatialmeasure::computeSpatialMeasureTypeIndex().

Referenced by multiscale::verification::SpatialMeasureEvaluator::evaluate().

#### 7.169.3.3 bool SpatialEntity::operator< ( const SpatialEntity & *rhsSpatialEntity* ) const

Overload the "<" operator for spatial entities.

In this implementation spatial entity se1 is smaller than spatial entity se2 (*se1 < se2*) if at least one of the fields in *se1 <* the corresponding field in *se2*

##### Parameters

|                         |                                                                            |
|-------------------------|----------------------------------------------------------------------------|
| <i>rhsSpatialEntity</i> | The spatial entity lying on the right hand side of the comparison operator |
|-------------------------|----------------------------------------------------------------------------|

Definition at line 41 of file SpatialEntity.cpp.

References multiscale::verification::NR\_SPATIAL\_MEASURE\_TYPES, semanticType, and spatialMeasureValues.

#### 7.169.3.4 void SpatialEntity::setSemanticType ( const std::string & *semanticType* )

Set the value of the semantic type.

##### Parameters

|                     |                                |
|---------------------|--------------------------------|
| <i>semanticType</i> | The value of the semantic type |
|---------------------|--------------------------------|

Definition at line 22 of file SpatialEntity.cpp.

#### 7.169.3.5 void SpatialEntity::setSpatialMeasureValue ( const SpatialMeasureType & *spatialMeasureType*, double *spatialMeasureValue* )

Set the value of the given spatial measure.

##### Parameters

|                           |                                                |
|---------------------------|------------------------------------------------|
| <i>spatialMeasureType</i> | The spatial measure for which the value is set |
|---------------------------|------------------------------------------------|

|                            |                               |
|----------------------------|-------------------------------|
| <i>spatialMeasureValue</i> | The new spatial measure value |
|----------------------------|-------------------------------|

Definition at line 32 of file SpatialEntity.cpp.

References multiscale::verification::spatialmeasure::computeSpatialMeasureTypeIndex().

#### 7.169.3.6 std::string SpatialEntity::toString( ) const

Return a string representation of the spatial entity contents.

Definition at line 58 of file SpatialEntity.cpp.

References multiscale::verification::NR\_SPATIAL\_MEASURE\_TYPES.

#### 7.169.3.7 void SpatialEntity::validateSpatialMeasureValue( double *spatialMeasureValue*, const SpatialMeasureType & *spatialMeasureType* ) [private]

Check if the provided value is valid considering the given spatial measure.

##### Parameters

|                            |                                                |
|----------------------------|------------------------------------------------|
| <i>spatialMeasureValue</i> | The new spatial measure value                  |
| <i>spatialMeasureType</i>  | The spatial measure for which the value is set |

Definition at line 70 of file SpatialEntity.cpp.

References multiscale::verification::spatialmeasure::getMaxValidSpatialMeasureValue(), multiscale::verification::spatialmeasure::getMinValidSpatialMeasureValue(), MS\_throw, and multiscale::verification::spatialmeasure::validateSpatialMeasureType().

### 7.169.4 Member Data Documentation

#### 7.169.4.1 const std::string SpatialEntity::ERR\_INVALID\_SPATIAL\_MEASURE\_BEGIN = "The provided spatial measure value (" [static], [private]

Definition at line 81 of file SpatialEntity.hpp.

#### 7.169.4.2 const std::string SpatialEntity::ERR\_INVALID\_SPATIAL\_MEASURE\_END = "). Please change." [static], [private]

Definition at line 83 of file SpatialEntity.hpp.

#### 7.169.4.3 const std::string SpatialEntity::ERR\_INVALID\_SPATIAL\_MEASURE\_MIDDLE = "is invalid for the given spatial measure type(" [static], [private]

Definition at line 82 of file SpatialEntity.hpp.

#### 7.169.4.4 const std::string SpatialEntity::OUTPUT\_SPATIAL\_MEASURE\_VALUE\_SEPARATOR = "," [static], [private]

Definition at line 79 of file SpatialEntity.hpp.

**7.169.4.5 std::string multiscale::verification::SpatialEntity::semanticType [protected]**

The semantic type of the spatial entity

Definition at line 21 of file SpatialEntity.hpp.

Referenced by operator<().

**7.169.4.6 std::vector<double> multiscale::verification::SpatialEntity::spatialMeasureValues [protected]**

The vector of spatial measures' values. The i-th spatial measure value in the vector corresponds to the i-th SpatialMeasureType enumeration value

Definition at line 23 of file SpatialEntity.hpp.

Referenced by operator<().

The documentation for this class was generated from the following files:

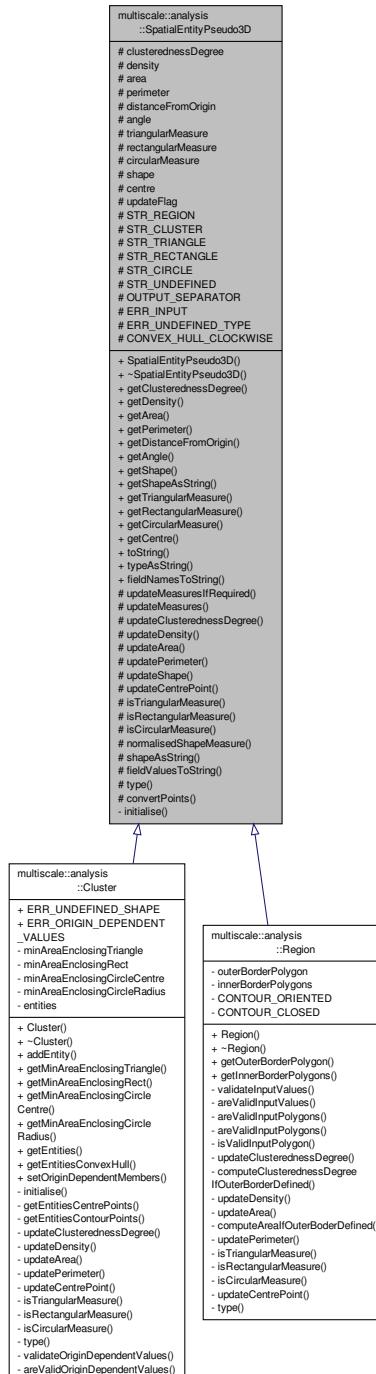
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[SpatialEntity.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[SpatialEntity.cpp](#)

**7.170 multiscale::analysis::SpatialEntityPseudo3D Class Reference**

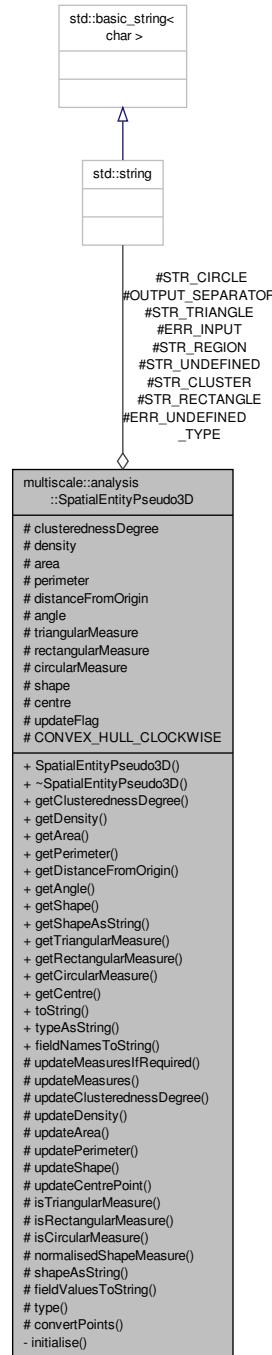
Class for representing a pseudo-3D (explicit 2D + implicit height) object.

```
#include <SpatialEntityPseudo3D.hpp>
```

Inheritance diagram for multiscale::analysis::SpatialEntityPseudo3D:



Collaboration diagram for multiscale::analysis::SpatialEntityPseudo3D:



## Public Member Functions

- [SpatialEntityPseudo3D \(\)](#)
- virtual [~SpatialEntityPseudo3D \(\)](#)
- [double getClusterednessDegree \(\)](#)

*Get the clusteredness degree.*

- [double getDensity \(\)](#)

- double `getArea ()`  
*Get the area.*
- double `getPerimeter ()`  
*Get the perimeter.*
- double `getDistanceFromOrigin ()`  
*Get the distance from the origin.*
- double `getAngle ()`  
*Get the angle.*
- Shape2D `getShape ()`  
*Get the shape best fitting the spatial collection.*
- std::string `getShapeAsString ()`  
*Get the shape best fitting the spatial collection as a std::string.*
- double `getTriangularMeasure ()`  
*Get the measure indicating how much the shape of the contour resembles a triangle.*
- double `getRectangularMeasure ()`  
*Get the measure indicating how much the shape of the contour resembles a rectangle.*
- double `getCircularMeasure ()`  
*Get the measure indicating how much the shape of the contour resembles a cv::circle.*
- cv::Point2f `getCentre ()`  
*Get the point defining the centre of the entity.*
- std::string `toString ()`  
*Get the std::string representation of all field values.*
- std::string `typeAsString ()`  
*Return the type of the pseudo 3D spatial entity as a std::string.*

## Static Public Member Functions

- static std::string `fieldNamesToString ()`  
*Get a std::string representation of all the field names printed in the "toString" method.*

## Protected Member Functions

- void `updateMeasuresIfRequired ()`  
*Update the values of all measures if required.*
- void `updateMeasures ()`  
*Update the values of all measures.*
- virtual void `updateClusterednessDegree ()=0`  
*Update the value of the clusteredness degree.*
- virtual void `updateDensity ()=0`  
*Update the value of the density.*
- virtual void `updateArea ()=0`  
*Update the value of the area.*
- virtual void `updatePerimeter ()=0`  
*Update the value of the perimeter.*
- void `updateShape ()`  
*Update the shape of the cluster.*
- virtual void `updateCentrePoint ()=0`  
*Update the point defining the centre of the cluster.*
- virtual double `isTriangularMeasure ()=0`

- `virtual double isRectangularMeasure ()=0`

*Get the measure that the cluster has a triangular shape.*
- `virtual double isCircularMeasure ()=0`

*Get the measure that the cluster has a rectangular shape.*
- `double normalisedShapeMeasure (double shapeArea)`

*Get the normalised shape measure ([0, 1]) that the cluster has a particular shape.*
- `std::string shapeAsString ()`

*Return the shape of the cluster as a std::string.*
- `std::string fieldValuesToString ()`

*Return the values of the fields as a std::string.*
- `virtual SpatialEntityPseudo3DType type ()=0`

*Return the type of the pseudo 3D spatial entity.*
- `std::vector< cv::Point2f > convertPoints (const std::vector< cv::Point > &points)`

*Convert the collection of points from type cv::Point to type cv::Point2f.*

## Protected Attributes

- `double clusterednessDegree`
- `double density`
- `double area`
- `double perimeter`
- `double distanceFromOrigin`
- `double angle`
- `double triangularMeasure`
- `double rectangularMeasure`
- `double circularMeasure`
- `Shape2D shape`
- `cv::Point2f centre`
- `bool updateFlag`

## Static Protected Attributes

- `static const std::string STR_REGION = "region"`
- `static const std::string STR_CLUSTER = "cluster"`
- `static const std::string STR_TRIANGLE = "triangular"`
- `static const std::string STR_RECTANGLE = "rectangular"`
- `static const std::string STR_CIRCLE = "circular"`
- `static const std::string STR_UNDEFINED = "undefined"`
- `static const std::string OUTPUT_SEPARATOR = ","`
- `static const std::string ERR_INPUT = "Invalid input parameters were provided to the constructor."`
- `static const std::string ERR_UNDEFINED_TYPE = "Pseudo 3D spatial entity of undefined type encountered."`
- `static const bool CONVEX_HULL_CLOCKWISE = true`

## Private Member Functions

- `void initialise ()`

*Initialisation function for the class.*

### 7.170.1 Detailed Description

Class for representing a pseudo-3D (explicit 2D + implicit height) object.

Definition at line 15 of file SpatialEntityPseudo3D.hpp.

### 7.170.2 Constructor & Destructor Documentation

#### 7.170.2.1 SpatialEntityPseudo3D::SpatialEntityPseudo3D ( )

Definition at line 8 of file SpatialEntityPseudo3D.cpp.

References initialise().

#### 7.170.2.2 SpatialEntityPseudo3D::~SpatialEntityPseudo3D ( ) [virtual]

Definition at line 12 of file SpatialEntityPseudo3D.cpp.

### 7.170.3 Member Function Documentation

#### 7.170.3.1 std::vector< cv::Point2f > SpatialEntityPseudo3D::convertPoints ( const std::vector< cv::Point > & points ) [protected]

Convert the collection of points from type cv::Point to type cv::Point2f.

Parameters

|               |                      |
|---------------|----------------------|
| <i>points</i> | Collection of points |
|---------------|----------------------|

Definition at line 202 of file SpatialEntityPseudo3D.cpp.

Referenced by multiscale::analysis::Region::isTriangularMeasure().

#### 7.170.3.2 std::string SpatialEntityPseudo3D::fieldNamesToString ( ) [static]

Get a std::string representation of all the field names printed in the "toString" method.

Definition at line 78 of file SpatialEntityPseudo3D.cpp.

Referenced by multiscale::analysis::Detector::outputResultsToCsvFile().

#### 7.170.3.3 std::string SpatialEntityPseudo3D::fieldValuesToString ( ) [protected]

Return the values of the fields as a std::string.

Definition at line 183 of file SpatialEntityPseudo3D.cpp.

References angle, area, centre, circularMeasure, clusterednessDegree, density, distanceFromOrigin, OUTPUT\_SEPARATOR, perimeter, rectangularMeasure, shapeAsString(), and triangularMeasure.

Referenced by toString().

#### 7.170.3.4 double SpatialEntityPseudo3D::getAngle ( )

Get the angle.

Definition at line 44 of file SpatialEntityPseudo3D.cpp.

References angle, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.5 double SpatialEntityPseudo3D::getArea( )

Get the area.

Definition at line 26 of file SpatialEntityPseudo3D.cpp.

References area, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.6 cv::Point2f SpatialEntityPseudo3D::getCentre( )

Get the point defining the centre of the entity.

Definition at line 72 of file SpatialEntityPseudo3D.cpp.

References centre, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.7 double SpatialEntityPseudo3D::getCircularMeasure( )

Get the measure indicating how much the shape of the contour resembles a cv::circle.

Definition at line 68 of file SpatialEntityPseudo3D.cpp.

References circularMeasure.

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.8 double SpatialEntityPseudo3D::getClusterednessDegree( )

Get the clusteredness degree.

Definition at line 14 of file SpatialEntityPseudo3D.cpp.

References clusterednessDegree, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.9 double SpatialEntityPseudo3D::getDensity( )

Get the density.

Definition at line 20 of file SpatialEntityPseudo3D.cpp.

References density, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

#### 7.170.3.10 double SpatialEntityPseudo3D::getDistanceFromOrigin( )

Get the distance from the origin.

Definition at line 38 of file SpatialEntityPseudo3D.cpp.

References distanceFromOrigin, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

**7.170.3.11 double SpatialEntityPseudo3D::getPerimeter( )**

Get the perimeter.

Definition at line 32 of file SpatialEntityPseudo3D.cpp.

References perimeter, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

**7.170.3.12 double SpatialEntityPseudo3D::getRectangularMeasure( )**

Get the measure indicating how much the shape of the contour resembles a rectangle.

Definition at line 64 of file SpatialEntityPseudo3D.cpp.

References rectangularMeasure.

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

**7.170.3.13 Shape2D SpatialEntityPseudo3D::getShape( )**

Get the shape best fitting the spatial collection.

Definition at line 50 of file SpatialEntityPseudo3D.cpp.

References shape, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterShape().

**7.170.3.14 std::string SpatialEntityPseudo3D::getShapeAsString( )**

Get the shape best fitting the spatial collection as a std::string.

Definition at line 56 of file SpatialEntityPseudo3D.cpp.

References shapeAsString().

**7.170.3.15 double SpatialEntityPseudo3D::getTriangularMeasure( )**

Get the measure indicating how much the shape of the contour resembles a triangle.

Definition at line 60 of file SpatialEntityPseudo3D.cpp.

References triangularMeasure.

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

**7.170.3.16 void SpatialEntityPseudo3D::initialise( ) [private]**

Initialisation function for the class.

Definition at line 212 of file SpatialEntityPseudo3D.cpp.

References area, circularMeasure, perimeter, rectangularMeasure, shape, triangularMeasure, multiscale::analysis::Undefined, and updateFlag.

Referenced by SpatialEntityPseudo3D().

**7.170.3.17 virtual double multiscale::analysis::SpatialEntityPseudo3D::isCircularMeasure( ) [protected], [pure virtual]**

Get the measure that the cluster has a circular shape.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateShape\(\)](#).

7.170.3.18 **virtual double multiscale::analysis::SpatialEntityPseudo3D::isRectangularMeasure( ) [protected], [pure virtual]**

Get the measure that the cluster has a rectangular shape.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateShape\(\)](#).

7.170.3.19 **virtual double multiscale::analysis::SpatialEntityPseudo3D::isTriangularMeasure( ) [protected], [pure virtual]**

Get the measure that the cluster has a triangular shape.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateShape\(\)](#).

7.170.3.20 **double SpatialEntityPseudo3D::normalisedShapeMeasure( double *shapeArea* ) [protected]**

Get the normalised shape measure ([0, 1]) that the cluster has a particular shape.

#### Parameters

|                  |                                  |
|------------------|----------------------------------|
| <i>shapeArea</i> | The area of the considered shape |
|------------------|----------------------------------|

Definition at line 149 of file [SpatialEntityPseudo3D.cpp](#).

References [area](#), [multiscale::Numeric::greaterOrEqual\(\)](#), and [multiscale::Numeric::lessOrEqual\(\)](#).

Referenced by [multiscale::analysis::Cluster::isCircularMeasure\(\)](#), [multiscale::analysis::Region::isCircularMeasure\(\)](#), [multiscale::analysis::Cluster::isRectangularMeasure\(\)](#), [multiscale::analysis::Region::isRectangularMeasure\(\)](#), [multiscale::analysis::Cluster::isTriangularMeasure\(\)](#), and [multiscale::analysis::Region::isTriangularMeasure\(\)](#).

7.170.3.21 **std::string SpatialEntityPseudo3D::shapeAsString( ) [protected]**

Return the shape of the cluster as a std::string.

Definition at line 161 of file [SpatialEntityPseudo3D.cpp](#).

References [multiscale::analysis::Circle](#), [multiscale::analysis::Rectangle](#), [shape](#), [STR\\_CIRCLE](#), [STR\\_RECTANGLE](#), [STR\\_TRIANGLE](#), [STR\\_UNDEFINED](#), [multiscale::analysis::Triangle](#), and [multiscale::analysis::Undefined](#).

Referenced by [fieldValuesToString\(\)](#), and [getShapeAsString\(\)](#).

7.170.3.22 **std::string SpatialEntityPseudo3D::toString( )**

Get the std::string representation of all field values.

Definition at line 85 of file [SpatialEntityPseudo3D.cpp](#).

References [fieldValuesToString\(\)](#), and [updateMeasuresIfRequired\(\)](#).

7.170.3.23 **virtual SpatialEntityPseudo3DType multiscale::analysis::SpatialEntityPseudo3D::type( ) [protected], [pure virtual]**

Return the type of the pseudo 3D spatial entity.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [typeAsString\(\)](#).

7.170.3.24 **std::string SpatialEntityPseudo3D::typeAsString( )**

Return the type of the pseudo 3D spatial entity as a std::string.

Definition at line 91 of file [SpatialEntityPseudo3D.cpp](#).

References [multiscale::analysis::Cluster](#), [ERR\\_UNDEFINED\\_TYPE](#), [MS\\_throw](#), [multiscale::analysis::Region](#), [STR\\_CLUSTER](#), [STR\\_REGION](#), [STR\\_UNDEFINED](#), and [type\(\)](#).

Referenced by [multiscale::analysis::Detector::addSpatialEntityTypeToPropertyTree\(\)](#).

7.170.3.25 **virtual void multiscale::analysis::SpatialEntityPseudo3D::updateArea( ) [protected], [pure virtual]**

Update the value of the area.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateMeasures\(\)](#).

7.170.3.26 **virtual void multiscale::analysis::SpatialEntityPseudo3D::updateCentrePoint( ) [protected], [pure virtual]**

Update the point defining the centre of the cluster.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateMeasures\(\)](#).

7.170.3.27 **virtual void multiscale::analysis::SpatialEntityPseudo3D::updateClusterednessDegree( ) [protected], [pure virtual]**

Update the value of the clusteredness degree.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateMeasures\(\)](#).

7.170.3.28 **virtual void multiscale::analysis::SpatialEntityPseudo3D::updateDensity( ) [protected], [pure virtual]**

Update the value of the density.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by [updateMeasures\(\)](#).

7.170.3.29 **void SpatialEntityPseudo3D::updateMeasures( ) [protected]**

Update the values of all measures.

Definition at line 120 of file [SpatialEntityPseudo3D.cpp](#).

References updateArea(), updateCentrePoint(), updateClusterednessDegree(), updateDensity(), updatePerimeter(), and updateShape().

Referenced by updateMeasuresIfRequired().

#### 7.170.3.30 void SpatialEntityPseudo3D::updateMeasuresIfRequired( ) [protected]

Update the values of all measures if required.

Definition at line 112 of file SpatialEntityPseudo3D.cpp.

References updateFlag, and updateMeasures().

Referenced by getAngle(), getArea(), getCentre(), getClusterednessDegree(), getDensity(), getDistanceFromOrigin(), multiscale::analysis::Cluster::getMinAreaEnclosingCircleCentre(), multiscale::analysis::Cluster::getMinAreaEnclosingCircleRadius(), multiscale::analysis::Cluster::getMinAreaEnclosingRect(), multiscale::analysis::Cluster::getMinAreaEnclosingTriangle(), getPerimeter(), getShape(), and toString().

#### 7.170.3.31 virtual void multiscale::analysis::SpatialEntityPseudo3D::updatePerimeter( ) [protected], [pure virtual]

Update the value of the perimeter.

Implemented in [multiscale::analysis::Region](#), and [multiscale::analysis::Cluster](#).

Referenced by updateMeasures().

#### 7.170.3.32 void SpatialEntityPseudo3D::updateShape( ) [protected]

Update the shape of the cluster.

Definition at line 129 of file SpatialEntityPseudo3D.cpp.

References multiscale::analysis::Circle, circularMeasure, isCircularMeasure(), isRectangularMeasure(), isTriangularMeasure(), multiscale::analysis::Rectangle, rectangularMeasure, shape, multiscale::analysis::Triangle, and triangularMeasure.

Referenced by updateMeasures().

### 7.170.4 Member Data Documentation

#### 7.170.4.1 double multiscale::analysis::SpatialEntityPseudo3D::angle [protected]

Angle of the region wrt the origin

Definition at line 31 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getAngle(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::setOriginDependentMembers().

#### 7.170.4.2 double multiscale::analysis::SpatialEntityPseudo3D::area [protected]

Area of the spatial collection

Definition at line 27 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Region::areValidInputValues(), fieldValuesToString(), getArea(), initialise(), normalisedShapeMeasure(), multiscale::analysis::Cluster::updateArea(), and multiscale::analysis::Region::updateArea().

**7.170.4.3 cv::Point2f multiscale::analysis::SpatialEntityPseudo3D::centre [protected]**

Point defining the centre of the spatial collection

Definition at line 41 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getCentre(), multiscale::analysis::Cluster::updateCentrePoint(), and multiscale::analysis::Region::updateCentrePoint().

**7.170.4.4 double multiscale::analysis::SpatialEntityPseudo3D::circularMeasure [protected]**

Measure ([0, 1]) indicating the similarity between the shape of the spatial collection and a circle

Definition at line 37 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getCircularMeasure(), initialise(), and updateShape().

**7.170.4.5 double multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree [protected]**

Degree of clusteredness

Definition at line 19 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Region::isValidInputValues(), fieldValuesToString(), getClusterednessDegree(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Cluster::updateClusterednessDegree(), and multiscale::analysis::Region::updateClusterednessDegree().

**7.170.4.6 const bool SpatialEntityPseudo3D::CONVEX\_HULL\_CLOCKWISE = true [static], [protected]**

Definition at line 175 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Cluster::getEntitiesConvexHull(), and multiscale::analysis::Region::isTriangularMeasure().

**7.170.4.7 double multiscale::analysis::SpatialEntityPseudo3D::density [protected]**

For regions: The average intensity of the pixels in the region normalised to the interval [0, 1]

For clusters: Degree of pile up

Definition at line 20 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getDensity(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::updateDensity().

**7.170.4.8 double multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin [protected]**

Distance from the origin

Definition at line 30 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getDistanceFromOrigin(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::setOriginDependentMembers().

**7.170.4.9 const std::string SpatialEntityPseudo3D::ERR\_INPUT = "Invalid input parameters were provided to the constructor." [static], [protected]**

Definition at line 172 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Region::validateInputValues().

7.170.4.10 `const std::string SpatialEntityPseudo3D::ERR_UNDEFINED_TYPE = "Pseudo 3D spatial entity of undefined type encountered."` [static], [protected]

Definition at line 173 of file SpatialEntityPseudo3D.hpp.

Referenced by `typeAsString()`.

7.170.4.11 `const std::string SpatialEntityPseudo3D::OUTPUT_SEPARATOR = ","` [static], [protected]

Definition at line 170 of file SpatialEntityPseudo3D.hpp.

Referenced by `fieldValuesToString()`.

7.170.4.12 `double multiscale::analysis::SpatialEntityPseudo3D::perimeter` [protected]

Perimeter of the spatial collection

Definition at line 28 of file SpatialEntityPseudo3D.hpp.

Referenced by `fieldValuesToString()`, `getPerimeter()`, `initialise()`, `multiscale::analysis::Cluster::updatePerimeter()`, and `multiscale::analysis::Region::updatePerimeter()`.

7.170.4.13 `double multiscale::analysis::SpatialEntityPseudo3D::rectangularMeasure` [protected]

Measure ([0, 1]) indicating the similarity between the shape of the spatial collection and a rectangle

Definition at line 35 of file SpatialEntityPseudo3D.hpp.

Referenced by `fieldValuesToString()`, `getRectangularMeasure()`, `initialise()`, and `updateShape()`.

7.170.4.14 `Shape2D multiscale::analysis::SpatialEntityPseudo3D::shape` [protected]

Shape of the spatial collection

Definition at line 40 of file SpatialEntityPseudo3D.hpp.

Referenced by `getShape()`, `initialise()`, `shapeAsString()`, and `updateShape()`.

7.170.4.15 `const std::string SpatialEntityPseudo3D::STR_CIRCLE = "circular"` [static], [protected]

Definition at line 167 of file SpatialEntityPseudo3D.hpp.

Referenced by `shapeAsString()`.

7.170.4.16 `const std::string SpatialEntityPseudo3D::STR_CLUSTER = "cluster"` [static], [protected]

Definition at line 163 of file SpatialEntityPseudo3D.hpp.

Referenced by `typeAsString()`.

7.170.4.17 `const std::string SpatialEntityPseudo3D::STR_RECTANGLE = "rectangular"` [static], [protected]

Definition at line 166 of file SpatialEntityPseudo3D.hpp.

Referenced by `shapeAsString()`.

7.170.4.18 const std::string SpatialEntityPseudo3D::STR\_REGION = "region" [static], [protected]

Definition at line 162 of file SpatialEntityPseudo3D.hpp.

Referenced by typeAsString().

7.170.4.19 const std::string SpatialEntityPseudo3D::STR\_TRIANGLE = "triangular" [static], [protected]

Definition at line 165 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString().

7.170.4.20 const std::string SpatialEntityPseudo3D::STR\_UNDEFINED = "undefined" [static], [protected]

Definition at line 168 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString(), and typeAsString().

7.170.4.21 double multiscale::analysis::SpatialEntityPseudo3D::triangularMeasure [protected]

Measure ([0, 1]) indicating the similarity between the shape of the spatial collection and a triangle

Definition at line 33 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getTriangularMeasure(), initialise(), and updateShape().

7.170.4.22 bool multiscale::analysis::SpatialEntityPseudo3D::updateFlag [protected]

Flag indicating if the field values dependent on the collection of entities need to be updated. This flag is used for lazy evaluation purposes, such that new field values are computed only when required

Definition at line 43 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Cluster::addEntity(), multiscale::analysis::Cluster::initialise(), initialise(), and updateMeasuresIfRequired().

The documentation for this class was generated from the following files:

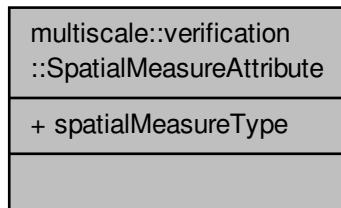
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SpatialEntityPseudo3D.cpp

## 7.171 multiscale::verification::SpatialMeasureAttribute Class Reference

Class for representing a spatial measure attribute.

```
#include <SpatialMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SpatialMeasureAttribute:



## Public Attributes

- [SpatialMeasureType spatialMeasureType](#)

### 7.171.1 Detailed Description

Class for representing a spatial measure attribute.

Definition at line 67 of file SpatialMeasureAttribute.hpp.

### 7.171.2 Member Data Documentation

#### 7.171.2.1 SpatialMeasureType multiscale::verification::SpatialMeasureAttribute::spatialMeasureType

The spatial measure type

Definition at line 71 of file SpatialMeasureAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection(), multiscale::verification::FilterNumericVisitor::operator()(), and multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

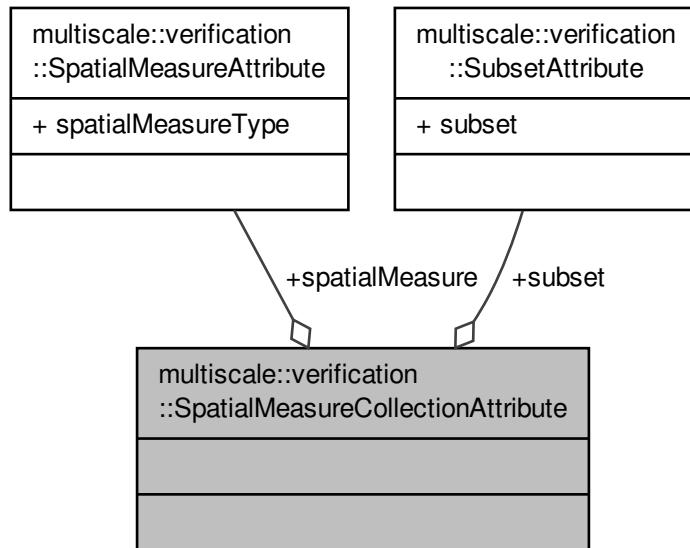
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp](#)

## 7.172 multiscale::verification::SpatialMeasureCollectionAttribute Class Reference

Class used to represent a spatial measure collection attribute.

```
#include <SpatialMeasureCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SpatialMeasureCollectionAttribute:



## Public Attributes

- [SpatialMeasureAttribute](#) `spatialMeasure`
- [SubsetAttribute](#) `subset`

### 7.172.1 Detailed Description

Class used to represent a spatial measure collection attribute.

Definition at line 15 of file `SpatialMeasureCollectionAttribute.hpp`.

### 7.172.2 Member Data Documentation

#### 7.172.2.1 SpatialMeasureAttribute multiscale::verification::SpatialMeasureCollectionAttribute::spatialMeasure

The spatial measure

Definition at line 19 of file `SpatialMeasureCollectionAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection()`.

#### 7.172.2.2 SubsetAttribute multiscale::verification::SpatialMeasureCollectionAttribute::subset

The considered subset

Definition at line 20 of file `SpatialMeasureCollectionAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection()`.

The documentation for this class was generated from the following file:

- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp

## **7.173 multiscale::verification::SpatialMeasureCollectionGrammar< Iterator > Class Template Reference**

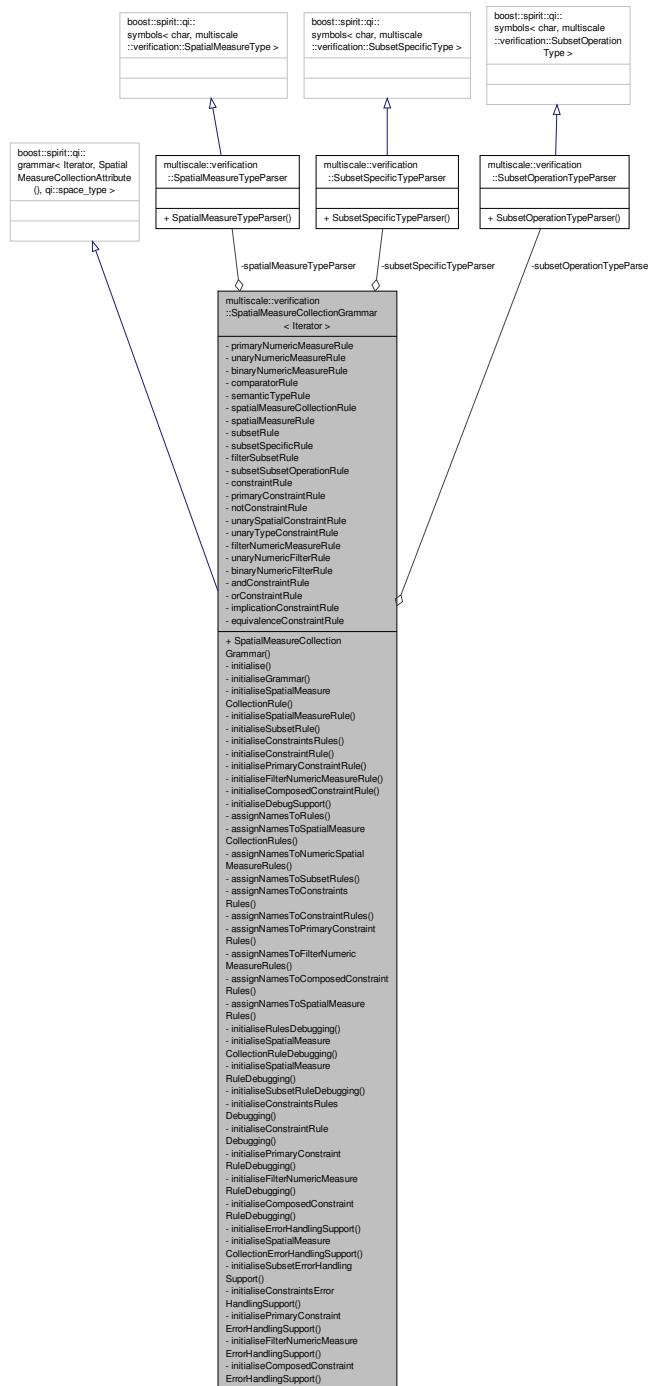
The grammar for parsing spatial measure collection statements.

```
#include <PrimaryNumericMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >:



Collaboration diagram for multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >:



## Public Member Functions

- [SpatialMeasureCollectionGrammar \(\)](#)

## Private Member Functions

- [void initialise \(\)](#)

- void **initialiseGrammar** ()  
*Initialise the grammar.*
- void **initialiseSpatialMeasureCollectionRule** ()  
*Initialise the spatial measure collection rule.*
- void **initialiseSpatialMeasureRule** ()  
*Initialise the spatial measure rule.*
- void **initialiseSubsetRule** ()  
*Initialise the subset rule.*
- void **initialiseConstraintsRules** ()  
*Initialise the constraints rules.*
- void **initialiseConstraintRule** ()  
*Initialise the constraint rule.*
- void **initialisePrimaryConstraintRule** ()  
*Initialise the primary constraint rule.*
- void **initialiseFilterNumericMeasureRule** ()  
*Initialise the filter numeric measure rule.*
- void **initialiseComposedConstraintRule** ()  
*Initialise the composed constraint rule.*
- void **initialiseDebugSupport** ()  
*Initialise debug support.*
- void **assignNamesToRules** ()  
*Assign names to the rules.*
- void **assignNamesToSpatialMeasureCollectionRules** ()  
*Assign names to the spatial measure collection rules.*
- void **assignNamesToNumericSpatialMeasureRules** ()  
*Assign names to the numeric spatial measure rules.*
- void **assignNamesToSubsetRules** ()  
*Assign names to the subset rules.*
- void **assignNamesToConstraintsRules** ()  
*Assign names to constraints rules.*
- void **assignNamesToConstraintRules** ()  
*Assign names to the constraint rules.*
- void **assignNamesToPrimaryConstraintRules** ()  
*Assign names to the primary constraint rules.*
- void **assignNamesToFilterNumericMeasureRules** ()  
*Assign names to the filter numeric measure rules.*
- void **assignNamesToComposedConstraintRules** ()  
*Assign names to the composed constraint rules.*
- void **assignNamesToSpatialMeasureRules** ()  
*Assign names to the spatial measure rules.*
- void **initialiseRulesDebugging** ()  
*Initialise the debugging of rules.*
- void **initialiseSpatialMeasureCollectionRuleDebugging** ()  
*Initialise debugging for the spatial measure collection rule.*
- void **initialiseSpatialMeasureRuleDebugging** ()  
*Initialise debugging for the numeric spatial measure rule.*
- void **initialiseSubsetRuleDebugging** ()  
*Initialise debugging for the subset rules.*
- void **initialiseConstraintsRulesDebugging** ()  
*Initialise the debugging of the constraints rules.*

- void **initialiseConstraintRuleDebugging ()**  
*Initialise debugging for the constraint rule.*
- void **initialisePrimaryConstraintRuleDebugging ()**  
*Initialise debugging for the primary constraint rules.*
- void **initialiseFilterNumericMeasureRuleDebugging ()**  
*Initialise debugging for the filter numeric measure rules.*
- void **initialiseComposedConstraintRuleDebugging ()**  
*Initialise debugging for the composed constraint rule.*
- void **initialiseErrorHandlingSupport ()**  
*Initialise the error handling routines.*
- void **initialiseSpatialMeasureCollectionErrorHandlingSupport ()**  
*Initialise the numeric measure collection error handling support.*
- void **initialiseSubsetErrorHandlingSupport ()**  
*Initialise the subset error handling support.*
- void **initialiseConstraintsErrorHandlingSupport ()**  
*Initialise the constraints error handling support.*
- void **initialisePrimaryConstraintErrorHandlingSupport ()**  
*Initialise the primary constraint error handling support.*
- void **initialiseFilterNumericMeasureErrorHandlingSupport ()**  
*Initialise the filter numeric measure error handling support.*
- void **initialiseComposedConstraintErrorHandlingSupport ()**  
*Initialise the composed constraint error handling support.*

## Private Attributes

- std::shared\_ptr  
`< PrimaryNumericMeasureGrammar`  
`< Iterator > > primaryNumericMeasureRule`
- **UnaryNumericMeasureGrammar**  
`< Iterator > unaryNumericMeasureRule`
- **BinaryNumericMeasureGrammar**  
`< Iterator > binaryNumericMeasureRule`
- **ComparatorGrammar**`< Iterator > comparatorRule`
- **SemanticTypeGrammar**`< Iterator > semanticTypeRule`
- qi::rule< Iterator,  
`SpatialMeasureCollectionAttribute()`,  
`qi::space_type > spatialMeasureCollectionRule`
- qi::rule< Iterator,  
`SpatialMeasureAttribute()`,  
`qi::space_type > spatialMeasureRule`
- qi::rule< Iterator,  
`SubsetAttribute()`,  
`qi::space_type > subsetRule`
- qi::rule< Iterator,  
`SubsetSpecificAttribute()`,  
`qi::space_type > subsetSpecificRule`
- qi::rule< Iterator,  
`FilterSubsetAttribute()`,  
`qi::space_type > filterSubsetRule`
- qi::rule< Iterator,  
`SubsetSubsetOperationAttribute()`,  
`qi::space_type > subsetSubsetOperationRule`

- qi::rule< Iterator,  
  **ConstraintAttribute()**,  
  qi::space\_type > **constraintRule**
- qi::rule< Iterator,  
  **PrimaryConstraintAttribute()**,  
  qi::space\_type > **primaryConstraintRule**
- qi::rule< Iterator,  
  **NotConstraintAttribute()**,  
  qi::space\_type > **notConstraintRule**
- qi::rule< Iterator,  
  **UnarySpatialConstraintAttribute()**,  
  qi::space\_type > **unarySpatialConstraintRule**
- qi::rule< Iterator,  
  **UnaryTypeConstraintAttribute()**,  
  qi::space\_type > **unaryTypeConstraintRule**
- qi::rule< Iterator,  
  **FilterNumericMeasureAttribute()**,  
  qi::space\_type > **filterNumericMeasureRule**
- qi::rule< Iterator,  
  **UnaryNumericFilterAttribute()**,  
  qi::space\_type > **unaryNumericFilterRule**
- qi::rule< Iterator,  
  **BinaryNumericFilterAttribute()**,  
  qi::space\_type > **binaryNumericFilterRule**
- qi::rule< Iterator,  
  **AndConstraintAttribute()**,  
  qi::space\_type > **andConstraintRule**
- qi::rule< Iterator,  
  **OrConstraintAttribute()**,  
  qi::space\_type > **orConstraintRule**
- qi::rule< Iterator,  
  **ImplicationConstraintAttribute()**,  
  qi::space\_type > **implicationConstraintRule**
- qi::rule< Iterator,  
  **EquivalenceConstraintAttribute()**,  
  qi::space\_type > **equivalenceConstraintRule**
- **SubsetSpecificTypeParser** **subsetSpecificTypeParser**
- **SubsetOperationTypeParser** **subsetOperationTypeParser**
- **SpatialMeasureTypeParser** **spatialMeasureTypeParser**

### 7.173.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >
```

The grammar for parsing spatial measure collection statements.

Definition at line 38 of file PrimaryNumericMeasureGrammar.hpp.

### 7.173.2 Constructor & Destructor Documentation

```
7.173.2.1 template<typename Iterator > multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::SpatialMeasureCollectionGrammar()
```

Definition at line 29 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialise().

### 7.173.3 Member Function Documentation

7.173.3.1 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToComposedConstraintRules( ) [private]`

Assign names to the composed constraint rules.

Definition at line 270 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.2 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToConstraintRules( ) [private]`

Assign names to the constraint rules.

Definition at line 247 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.3 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToConstraintsRules( ) [private]`

Assign names to constraints rules.

Definition at line 238 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.4 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToFilterNumericMeasureRules( ) [private]`

Assign names to the filter numeric measure rules.

Definition at line 262 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.5 `template<typename Iterator> void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToNumericSpatialMeasureRules( ) [private]`

Assign names to the numeric spatial measure rules.

7.173.3.6 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToPrimaryConstraintRules( ) [private]`

Assign names to the primary constraint rules.

Definition at line 253 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.7 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToRules( ) [private]`

Assign names to the rules.

Definition at line 208 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.8 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::assignNamesToSpatialMeasureCollectionRules( ) [private]`

Assign names to the spatial measure collection rules.

Definition at line 217 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.9 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::assignNamesToSpatialMeasureRules() [private]
```

Assign names to the spatial measure rules.

Definition at line 223 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.10 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::assignNamesToSubsetRules() [private]
```

Assign names to the subset rules.

Definition at line 229 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.11 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialise() [private]
```

Initialisation function.

Definition at line 42 of file SpatialMeasureCollectionGrammarDefinition.hpp.

Referenced by multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::SpatialMeasureCollectionGrammar().

```
7.173.3.12 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialiseComposedConstraintErrorHandlingSupport() [private]
```

Initialise the composed constraint error handling support.

Definition at line 426 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

```
7.173.3.13 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialiseComposedConstraintRule() [private]
```

Initialise the composed constraint rule.

Definition at line 183 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.14 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialiseComposedConstraintRuleDebugging() [private]
```

Initialise debugging for the composed constraint rule.

Definition at line 341 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.15 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialiseConstraintRule() [private]
```

Initialise the constraint rule.

Definition at line 119 of file SpatialMeasureCollectionGrammarDefinition.hpp.

```
7.173.3.16 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar<
Iterator >::initialiseConstraintRuleDebugging() [private]
```

Initialise debugging for the constraint rule.

Definition at line 318 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.17 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseConstraintsErrorHandlingSupport( ) [private]**

Initialise the constraints error handling support.

Definition at line 380 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.18 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseConstraintsRules( ) [private]**

Initialise the constraints rules.

Definition at line 110 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.19 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseConstraintsRulesDebugging( ) [private]**

Initialise the debugging of the constraints rules.

Definition at line 309 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.20 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseDebugSupport( ) [private]**

Initialise debug support.

Definition at line 199 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.21 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]**

Initialise the error handling routines.

Definition at line 350 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.22 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseFilterNumericMeasureErrorHandlingSupport( ) [private]**

Initialise the filter numeric measure error handling support.

Definition at line 409 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

**7.173.3.23 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseFilterNumericMeasureRule( ) [private]**

Initialise the filter numeric measure rule.

Definition at line 155 of file SpatialMeasureCollectionGrammarDefinition.hpp.

**7.173.3.24 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseFilterNumericMeasureRuleDebugging( ) [private]**

Initialise debugging for the filter numeric measure rules.

Definition at line 333 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.25 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseGrammar( ) [private]`

Initialise the grammar.

Definition at line 50 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.26 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialisePrimaryConstraintErrorHandlingSupport( ) [private]`

Initialise the primary constraint error handling support.

Definition at line 388 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.173.3.27 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialisePrimaryConstraintRule( ) [private]`

Initialise the primary constraint rule.

Definition at line 132 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.28 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialisePrimaryConstraintRuleDebugging( ) [private]`

Initialise debugging for the primary constraint rules.

Definition at line 324 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.29 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseRulesDebugging( ) [private]`

Initialise the debugging of rules.

Definition at line 279 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.30 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasureCollectionErrorHandlingSupport( ) [private]`

Initialise the numeric measure collection error handling support.

Initialise the spatial measure collection error handling support.

Definition at line 358 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.173.3.31 `template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasureCollectionRule( ) [private]`

Initialise the spatial measure collection rule.

Definition at line 59 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.32 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasureCollectionRuleDebugging( ) [private]

Initialise debugging for the spatial measure collection rule.

Definition at line 288 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.33 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasureRule( ) [private]

Initialise the spatial measure rule.

Definition at line 71 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.34 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSpatialMeasureRuleDebugging( ) [private]

Initialise debugging for the numeric spatial measure rule.

Initialise debugging for the spatial measure rule.

Definition at line 294 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.35 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSubsetErrorHandlingSupport( ) [private]

Initialise the subset error handling support.

Definition at line 367 of file SpatialMeasureCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.173.3.36 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSubsetRule( ) [private]

Initialise the subset rule.

Definition at line 78 of file SpatialMeasureCollectionGrammarDefinition.hpp.

7.173.3.37 template<typename Iterator > void multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::initialiseSubsetRuleDebugging( ) [private]

Initialise debugging for the subset rules.

Definition at line 300 of file SpatialMeasureCollectionGrammarDefinition.hpp.

#### 7.173.4 Member Data Documentation

7.173.4.1 template<typename Iterator> qi::rule<Iterator, AndConstraintAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::andConstraintRule [private]

The rule for parsing an "and" constraint

Definition at line 109 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.2 template<typename Iterator> qi::rule<Iterator, BinaryNumericFilterAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::binaryNumericFilterRule [private]

The rule for parsing a binary numeric filter measure

Definition at line 105 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.3 template<typename Iterator> BinaryNumericMeasureGrammar<Iterator> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::binaryNumericMeasureRule [private]

The grammar for parsing binary numeric measures

Definition at line 56 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.4 template<typename Iterator> ComparatorGrammar<Iterator> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::comparatorRule [private]

The grammar for parsing comparators

Definition at line 60 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.5 template<typename Iterator> qi::rule<Iterator, ConstraintAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::constraintRule [private]

The rule for parsing a constraint

Definition at line 85 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.6 template<typename Iterator> qi::rule<Iterator, EquivalenceConstraintAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::equivalenceConstraintRule [private]

The rule for parsing an "equivalence" constraint

Definition at line 116 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.7 template<typename Iterator> qi::rule<Iterator, FilterNumericMeasureAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::filterNumericMeasureRule [private]

The rule for parsing a filter numeric measure

Definition at line 99 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.8 template<typename Iterator> qi::rule<Iterator, FilterSubsetAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::filterSubsetRule [private]

The rule for parsing a subset filter

Definition at line 79 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.9 template<typename Iterator> qi::rule<Iterator, ImplicationConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::implicationConstraintRule**  
[private]

The rule for parsing an "implication" constraint

Definition at line 113 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.10 template<typename Iterator> qi::rule<Iterator, NotConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::notConstraintRule**  
[private]

The rule for parsing a "not" constraint

Definition at line 90 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.11 template<typename Iterator> qi::rule<Iterator, OrConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::orConstraintRule**  
[private]

The rule for parsing an "or" constraint

Definition at line 111 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.12 template<typename Iterator> qi::rule<Iterator, PrimaryConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::primaryConstraintRule**  
[private]

The rule for parsing a primary constraint

Definition at line 88 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.13 template<typename Iterator> std::shared\_ptr<PrimaryNumericMeasureGrammar<Iterator>>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::primaryNumericMeasureRule**  
[private]

The grammar for parsing primary numeric measures

Definition at line 49 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.14 template<typename Iterator> SemanticTypeGrammar<Iterator> **multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::semanticTypeRule**  
[private]

The grammar for parsing semantic types

Definition at line 63 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.15 template<typename Iterator> qi::rule<Iterator, SpatialMeasureCollectionAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::spatialMeasureCollectionRule**  
[private]

The rule for parsing a spatial measure collection

Definition at line 68 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.16 template<typename Iterator> qi::rule<Iterator, SpatialMeasureAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::spatialMeasureRule [private]

The rule for parsing a spatial measure

Definition at line 72 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.17 template<typename Iterator> SpatialMeasureTypeParser multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::spatialMeasureTypeParser [private]

The spatial measure type parser

Definition at line 127 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.18 template<typename Iterator> SubsetOperationTypeParser multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::subsetOperationTypeParser [private]

The subset operation type parser

Definition at line 124 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.19 template<typename Iterator> qi::rule<Iterator, SubsetAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::subsetRule [private]

The rule for parsing a subset

Definition at line 75 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.20 template<typename Iterator> qi::rule<Iterator, SubsetSpecificAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::subsetSpecificRule [private]

The rule for parsing a specific subset

Definition at line 77 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.21 template<typename Iterator> SubsetSpecificTypeParser multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::subsetSpecificTypeParser [private]

The subset specific type parser

Definition at line 122 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.22 template<typename Iterator> qi::rule<Iterator, SubsetSubsetOperationAttribute(), qi::space\_type> multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >::subsetSubsetOperationRule [private]

The rule for parsing a subset subset operation

Definition at line 81 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.23 template<typename Iterator> qi::rule<Iterator, UnaryNumericFilterAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar**< Iterator >::unaryNumericFilterRule  
[private]

The rule for parsing a unary numeric filter measure

Definition at line 102 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.24 template<typename Iterator> UnaryNumericMeasureGrammar<Iterator> **multiscale**←  
**::verification::SpatialMeasureCollectionGrammar**< Iterator >::unaryNumericMeasureRule  
[private]

The grammar for parsing unary numeric measures

Definition at line 53 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.25 template<typename Iterator> qi::rule<Iterator, UnarySpatialConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar**< Iterator >::unarySpatialConstraintRule  
[private]

The rule for parsing a unary spatial constraint

Definition at line 92 of file SpatialMeasureCollectionGrammar.hpp.

7.173.4.26 template<typename Iterator> qi::rule<Iterator, UnaryTypeConstraintAttribute(), qi::space\_type>  
**multiscale::verification::SpatialMeasureCollectionGrammar**< Iterator >::unaryTypeConstraintRule  
[private]

The rule for parsing a unary type constraint

Definition at line 95 of file SpatialMeasureCollectionGrammar.hpp.

The documentation for this class was generated from the following files:

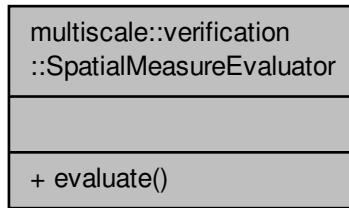
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[PrimaryNumericMeasureGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SpatialMeasureCollectionGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SpatialMeasureCollectionGrammarDefinition.hpp](#)

## 7.174 multiscale::verification::SpatialMeasureEvaluator Class Reference

Class for evaluating spatial measures.

```
#include <SpatialMeasureEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::SpatialMeasureEvaluator:



## Static Public Member Functions

- static double [evaluate](#) (const [SpatialEntity](#) &[spatialEntity](#), const [SpatialMeasureType](#) &[type](#))

*Return the value of the spatial measure for the given spatial entity.*

### 7.174.1 Detailed Description

Class for evaluating spatial measures.

Definition at line 13 of file [SpatialMeasureEvaluator.hpp](#).

### 7.174.2 Member Function Documentation

#### 7.174.2.1 static double multiscale::verification::SpatialMeasureEvaluator::evaluate ( const [SpatialEntity](#) & [spatialEntity](#), const [SpatialMeasureType](#) & [type](#) ) [inline], [static]

Return the value of the spatial measure for the given spatial entity.

##### Parameters

|                               |                                 |
|-------------------------------|---------------------------------|
| <a href="#">spatialEntity</a> | The given spatial entity        |
| <a href="#">type</a>          | The type of the spatial measure |

Definition at line 22 of file [SpatialMeasureEvaluator.hpp](#).

References [multiscale::verification::SpatialEntity::getSpatialMeasureValue\(\)](#), and [multiscale::verification::spatialmeasure::validateSpatialMeasureType\(\)](#).

Referenced by [multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtSpatialMeasure\(\)](#), [multiscale::verification::TimePointEvaluator::getSpatialMeasureValues\(\)](#), and [multiscale::verification::FilterNumericVisitor::operator\(\)](#).

The documentation for this class was generated from the following file:

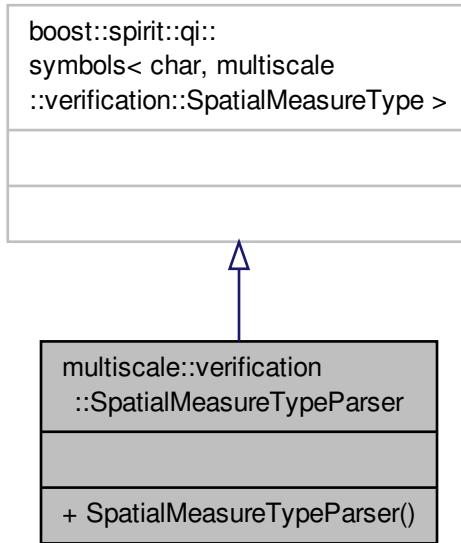
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[SpatialMeasureEvaluator.hpp](#)

## 7.175 multiscale::verification::SpatialMeasureTypeParser Struct Reference

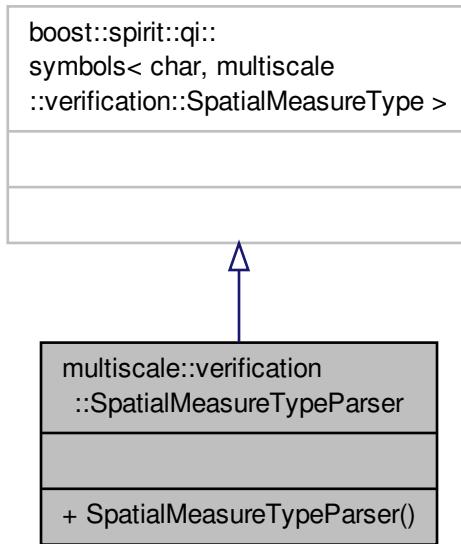
Symbol table and parser for the spatial measure type.

```
#include <SymbolTablesAutoGenerated.hpp>
```

Inheritance diagram for multiscale::verification::SpatialMeasureTypeParser:



Collaboration diagram for multiscale::verification::SpatialMeasureTypeParser:



## Public Member Functions

- [SpatialMeasureTypeParser \(\)](#)

### 7.175.1 Detailed Description

Symbol table and parser for the spatial measure type.

Definition at line 28 of file [SymbolTablesAutoGenerated.hpp](#).

### 7.175.2 Constructor & Destructor Documentation

#### 7.175.2.1 multiscale::verification::SpatialMeasureTypeParser::SpatialMeasureTypeParser( ) [inline]

Definition at line 30 of file [SymbolTablesAutoGenerated.hpp](#).

References [multiscale::verification::Angle](#), [multiscale::verification::Area](#), [multiscale::verification::CentroidX](#), [multiscale::verification::CentroidY](#), [multiscale::verification::CircleMeasure](#), [multiscale::verification::Clusteredness](#), [multiscale::verification::Density](#), [multiscale::verification::DistanceFromOrigin](#), [multiscale::verification::Perimeter](#), [multiscale::verification::RectangleMeasure](#), and [multiscale::verification::TriangleMeasure](#).

The documentation for this struct was generated from the following file:

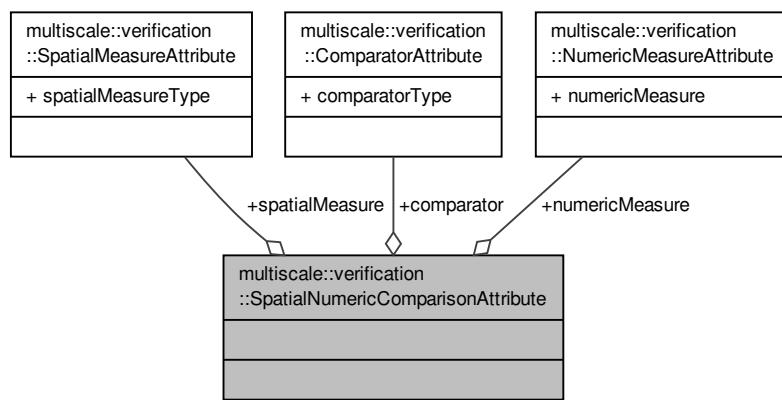
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTablesAutoGenerated.hpp\]\(#\)](#)

## 7.176 multiscale::verification::SpatialNumericComparisonAttribute Class Reference

Class for representing a spatial numeric comparison attribute.

```
#include <SpatialNumericComparisonAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SpatialNumericComparisonAttribute:



## Public Attributes

- [SpatialMeasureAttribute spatialMeasure](#)

- [ComparatorAttribute comparator](#)
- [NumericMeasureAttribute numericMeasure](#)

## 7.176.1 Detailed Description

Class for representing a spatial numeric comparison attribute.

Definition at line 19 of file [SpatialNumericComparisonAttribute.hpp](#).

## 7.176.2 Member Data Documentation

### 7.176.2.1 ComparatorAttribute multiscale::verification::SpatialNumericComparisonAttribute::comparator

The comparator

Definition at line 24 of file [SpatialNumericComparisonAttribute.hpp](#).

### 7.176.2.2 NumericMeasureAttribute multiscale::verification::SpatialNumericComparisonAttribute::numericMeasure

The numeric measure

Definition at line 25 of file [SpatialNumericComparisonAttribute.hpp](#).

### 7.176.2.3 SpatialMeasureAttribute multiscale::verification::SpatialNumericComparisonAttribute::spatialMeasure

The spatial measure

Definition at line 23 of file [SpatialNumericComparisonAttribute.hpp](#).

The documentation for this class was generated from the following file:

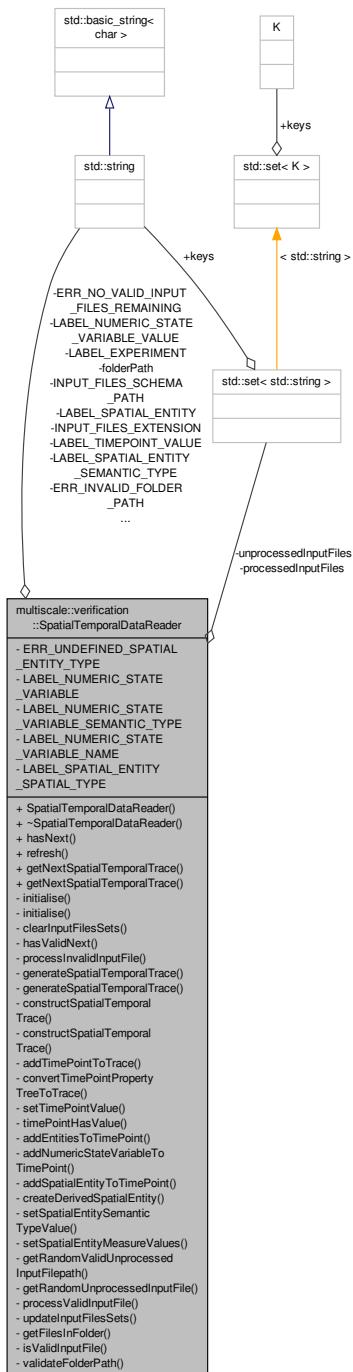
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialNumericComparisonAttribute.hpp](#)

## 7.177 multiscale::verification::SpatialTemporalDataReader Class Reference

Class for reading spatial temporal trace data from input files.

```
#include <SpatialTemporalDataReader.hpp>
```

Collaboration diagram for multiscale::verification::SpatialTemporalDataReader:



## Public Member Functions

- `SpatialTemporalDataReader (const std::string &folderPath)`
- `~SpatialTemporalDataReader ()`
- `bool hasNext ()`

*Check if there are any remaining valid unprocessed traces in the given folder.*

- `void refresh ()`

- Refresh the sets of processed and unprocessed traces' input files considering the given folder.*
- **SpatialTemporalTrace getNextSpatialTemporalTrace ()**

*Return the next spatial temporal trace.*
  - **SpatialTemporalTrace getNextSpatialTemporalTrace (std::string &tracePath)**

*Return the next spatial temporal trace and its path.*
- ### Private Member Functions
- void **initialise (const std::string &folderPath)**

*Initialise the sets for storing processed and unprocessed input files.*
  - void **initialise ()**

*Initialise the sets for storing processed and unprocessed input files.*
  - void **clearInputFilesSets ()**

*Clear the contents of the sets of processed and unprocessed input files.*
  - bool **hasValidNext ()**

*Check if there are any remaining valid unprocessed traces in the given folder.*
  - std::set< std::string >::iterator **processInvalidInputFile (const std::set< std::string >::iterator &invalidInputFileIterator)**

*Process the invalid input file to which the given iterator points.*
  - **SpatialTemporalTrace generateSpatialTemporalTrace ()**

*Generate the spatial temporal trace corresponding to the first valid unprocessed input file.*
  - **SpatialTemporalTrace generateSpatialTemporalTrace (std::string &tracePath)**

*Generate the spatial temporal trace corresponding to the first valid unprocessed input file.*
  - **SpatialTemporalTrace constructSpatialTemporalTrace (const std::string &inputFilepath)**

*Construct the spatial temporal trace corresponding to the first valid unprocessed input file.*
  - **SpatialTemporalTrace constructSpatialTemporalTrace (const pt::ptree &tree)**

*Construct the spatial temporal trace corresponding to the given property tree.*
  - void **addTimePointToTrace (const pt::ptree &timePointTree, SpatialTemporalTrace &trace)**

*Add a timepoint corresponding to the given property tree to the spatial temporal trace.*
  - void **convertTimePointPropertyTreeToTrace (const pt::ptree &timePointTree, TimePoint &timePoint)**

*Convert a time point from a property tree to a timepoint representation.*
  - void **setTimePointValue (const pt::ptree &timePointTree, TimePoint &timePoint)**

*Set the value of the timepoint considering the given timepoint tree.*
  - bool **timePointHasValue (const pt::ptree &propertyTree, unsigned long &value)**

*Check if the provided property tree contains the attribute "value".*
  - void **addEntitiesToTimePoint (const pt::ptree &timePointTree, TimePoint &timePoint)**

*Add the numeric state variables and spatial entities contained by the property tree to the given timepoint.*
  - void **addNumericStateVariableToTimePoint (const pt::ptree &numericStateVariableTree, TimePoint &timePoint)**

*Add the numeric state variable (provided as a tree) to the provided timepoint.*
  - void **addSpatialEntityToTimePoint (const pt::ptree &spatialEntityTree, TimePoint &timePoint)**

*Add the spatial entity contained by the property tree to the given timePoint.*
  - void **createDerivedSpatialEntity (const pt::ptree &spatialEntityTree, std::shared\_ptr< SpatialEntity > &spatialEntity, SubsetSpecificType &spatialEntityType)**

*Create a derived spatial entity considering the type specified in the given tree.*
  - void **setSpatialEntitySemanticTypeValue (const pt::ptree &spatialEntityTree, const std::shared\_ptr< SpatialEntity > &spatialEntity)**

*Initialise the spatial entity semantic type value using the given spatialEntityTree.*
  - void **setSpatialEntityMeasureValues (const pt::ptree &spatialEntityTree, const std::shared\_ptr< SpatialEntity > &spatialEntity)**

*Initialise the spatial entity measure values using the given spatialEntityTree.*

- std::set< std::string >::iterator [getRandomValidUnprocessedInputFilepath \(\)](#)  
*Get an iterator pointing to a random valid unprocessed input file.*
- std::set< std::string >::iterator [getRandomUnprocessedInputFile \(\)](#)  
*Get an iterator pointing to a random unprocessed input file.*
- void [processValidInputFile \(const std::set< std::string >::iterator &validInputFileIterator\)](#)  
*Process the valid input file to which the given iterator points.*
- void [updateInputFilesSets \(\)](#)  
*Update the sets of processed and unprocessed files by checking if the folder contents have been updated.*
- std::vector< std::string > [getFilesInFolder \(\)](#)  
*Get the collection of files stored in the input folder.*
- bool [isValidInputFile \(const std::string &inputFilepath\)](#)  
*Check if the given input file is valid.*
- void [validateFolderPath \(const std::string &folderPath\)](#)  
*Check if the given folder path is valid.*

## Private Attributes

- std::set< std::string > [processedInputFiles](#)
- std::set< std::string > [unprocessedInputFiles](#)
- std::string [FolderPath](#)

## Static Private Attributes

- static const std::string [ERR\\_INVALID\\_FOLDER\\_PATH](#) = "The provided path does not point to a folder. Please change."
- static const std::string [ERR\\_NO\\_VALID\\_INPUT\\_FILES\\_REMAINING](#) = "There are no valid unprocessed input files remaining."
- static const std::string [ERR\\_UNDEFINED\\_SPATIAL\\_ENTITY\\_TYPE](#) = "The provided spatial entity type is invalid."
- static const std::string [LABEL\\_EXPERIMENT](#) = "experiment"
- static const std::string [LABEL\\_TIMEPOINT\\_VALUE](#) = "<xmllattr>.value"
- static const std::string [LABEL\\_NUMERIC\\_STATE\\_VARIABLE](#) = "numericStateVariable"
- static const std::string [LABEL\\_NUMERIC\\_STATE\\_VARIABLE\\_SEMANTIC\\_TYPE](#) = "<xmllattr>.semanticType"
- static const std::string [LABEL\\_NUMERIC\\_STATE\\_VARIABLE\\_NAME](#) = "name"
- static const std::string [LABEL\\_NUMERIC\\_STATE\\_VARIABLE\\_VALUE](#) = "value"
- static const std::string [LABEL\\_SPATIAL\\_ENTITY](#) = "spatialEntity"
- static const std::string [LABEL\\_SPATIAL\\_ENTITY\\_SPATIAL\\_TYPE](#) = "<xmllattr>.spatialType"
- static const std::string [LABEL\\_SPATIAL\\_ENTITY\\_SEMANTIC\\_TYPE](#) = "<xmllattr>.semanticType"
- static const std::string [INPUT\\_FILES\\_EXTENSION](#) = ".xml"
- static const std::string [INPUT\\_FILES\\_SCHEMA\\_PATH](#) = "/usr/local/share/mule/config/verification/spatial-temporal/schema/MSTML\_L1V1.xsd"

### 7.177.1 Detailed Description

Class for reading spatial temporal trace data from input files.

Definition at line 20 of file SpatialTemporalDataReader.hpp.

## 7.177.2 Constructor & Destructor Documentation

### 7.177.2.1 SpatialTemporalDataReader::SpatialTemporalDataReader ( const std::string & *folderPath* )

Definition at line 18 of file SpatialTemporalDataReader.cpp.

References initialise().

### 7.177.2.2 SpatialTemporalDataReader::~SpatialTemporalDataReader ( )

Definition at line 22 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

## 7.177.3 Member Function Documentation

### 7.177.3.1 void SpatialTemporalDataReader::addEntitiesToTimePoint ( const pt::ptree & *timePointTree*, TimePoint & *timePoint* ) [private]

Add the numeric state variables and spatial entities contained by the property tree to the given timepoint.

#### Parameters

|                      |                         |
|----------------------|-------------------------|
| <i>timePointTree</i> | The given property tree |
| <i>timePoint</i>     | The given timepoint     |

Definition at line 186 of file SpatialTemporalDataReader.cpp.

References addNumericStateVariableToTimePoint(), addSpatialEntityToTimePoint(), LABEL\_NUMERIC\_STATE\_VARIABLE, and LABEL\_SPATIAL\_ENTITY.

Referenced by convertTimePointPropertyTreeToTrace().

### 7.177.3.2 void SpatialTemporalDataReader::addNumericStateVariableToTimePoint ( const pt::ptree & *numericStateVariableTree*, TimePoint & *timePoint* ) [private]

Add the numeric state variable (provided as a tree) to the provided timepoint.

#### Parameters

|                                 |                                                   |
|---------------------------------|---------------------------------------------------|
| <i>numericStateVariableTree</i> | The provided numeric state variable property tree |
| <i>timePoint</i>                | The given timepoint                               |

Definition at line 198 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::addNumericStateVariable(), multiscale::verification::SemanticType::DEFAULT\_VALUE, LABEL\_NUMERIC\_STATE\_VARIABLE\_NAME, LABEL\_NUMERIC\_STATE\_VARIABLE\_SEMANTIC\_TYPE, and LABEL\_NUMERIC\_STATE\_VARIABLE\_VALUE.

Referenced by addEntitiesToTimePoint().

### 7.177.3.3 void SpatialTemporalDataReader::addSpatialEntityToTimePoint ( const pt::ptree & *spatialEntityTree*, TimePoint & *timePoint* ) [private]

Add the spatial entity contained by the property tree to the given timePoint.

**Parameters**

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| <i>spatialEntityTree</i> | The given spatial entity represented as a property tree |
| <i>timePoint</i>         | The given timepoint                                     |

Definition at line 216 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::addSpatialEntity(), createDerivedSpatialEntity(), setSpatialEntityMeasureValues(), and setSpatialEntitySemanticTypeValue().

Referenced by addEntitiesToTimePoint().

**7.177.3.4 void SpatialTemporalDataReader::addTimePointToTrace ( const pt::ptree & *timePointTree*, SpatialTemporalTrace & *trace* ) [private]**

Add a timepoint corresponding to the given property tree to the spatial temporal trace.

**Parameters**

|                      |                                                  |
|----------------------|--------------------------------------------------|
| <i>timePointTree</i> | The property tree corresponding to the timepoint |
| <i>trace</i>         | The spatial temporal trace                       |

Definition at line 144 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::SpatialTemporalTrace::addTimePoint(), and convertTimePointPropertyTreeToTrace().

Referenced by constructSpatialTemporalTrace().

**7.177.3.5 void SpatialTemporalDataReader::clearInputFilesSets ( ) [private]**

Clear the contents of the sets of processed and unprocessed input files.

Definition at line 71 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

Referenced by initialise().

**7.177.3.6 SpatialTemporalTrace SpatialTemporalDataReader::constructSpatialTemporalTrace ( const std::string & *inputFilepath* ) [private]**

Construct the spatial temporal trace corresponding to the first valid unprocessed input file.

The unprocessed input file will be processed and returned as a property tree.

**Parameters**

|                      |                                       |
|----------------------|---------------------------------------|
| <i>inputFilepath</i> | The valid unprocessed input file path |
|----------------------|---------------------------------------|

Definition at line 124 of file SpatialTemporalDataReader.cpp.

Referenced by generateSpatialTemporalTrace().

**7.177.3.7 SpatialTemporalTrace SpatialTemporalDataReader::constructSpatialTemporalTrace ( const pt::ptree & *tree* ) [private]**

Construct the spatial temporal trace corresponding to the given property tree.

Definition at line 133 of file SpatialTemporalDataReader.cpp.

References addTimePointToTrace(), and LABEL\_EXPERIMENT.

7.177.3.8 void SpatialTemporalDataReader::convertTimePointPropertyTreeToTrace ( const pt::ptree & *timePointTree*,  
TimePoint & *timePoint* ) [private]

Convert a time point from a property tree to a timepoint representation.

**Parameters**

|                      |                                                               |
|----------------------|---------------------------------------------------------------|
| <i>timePointTree</i> | Property tree representation of the timepoint                 |
| <i>timePoint</i>     | The <a href="#">TimePoint</a> representation of the timepoint |

Definition at line 153 of file SpatialTemporalDataReader.cpp.

References [addEntitiesToTimePoint\(\)](#), and [setTimePointValue\(\)](#).

Referenced by [addTimePointToTrace\(\)](#).

**7.177.3.9 void SpatialTemporalDataReader::createDerivedSpatialEntity ( const pt::ptree & *spatialEntityTree*, std::shared\_ptr< SpatialEntity > & *spatialEntity*, SubsetSpecificType & *spatialEntityType* ) [private]**

Create a derived spatial entity considering the type specified in the given tree.

**Parameters**

|                          |                                                         |
|--------------------------|---------------------------------------------------------|
| <i>spatialEntityTree</i> | The given spatial entity represented as a property tree |
| <i>spatialEntity</i>     | The created spatial entity                              |
| <i>spatialEntityType</i> | The derived type of the spatial entity                  |

Definition at line 22 of file SpatialTemporalDataReaderAutoGenerated.cpp.

References [multiscale::verification::Clusters](#), [ERR\\_UNDEFINED\\_SPATIAL\\_ENTITY\\_TYPE](#), [LABEL\\_SPATIAL\\_ENTITY\\_SPATIAL\\_TYPE](#), [MS\\_throw](#), and [multiscale::verification::Regions](#).

Referenced by [addSpatialEntityToTimePoint\(\)](#).

**7.177.3.10 SpatialTemporalTrace SpatialTemporalDataReader::generateSpatialTemporalTrace ( ) [private]**

Generate the spatial temporal trace corresponding to the first valid unprocessed input file.

The unprocessed input file will be moved to the set of processed input files after creating the spatial temporal trace.

Definition at line 102 of file SpatialTemporalDataReader.cpp.

References [constructSpatialTemporalTrace\(\)](#), [getRandomValidUnprocessedInputFilepath\(\)](#), and [processValidInputFile\(\)](#).

Referenced by [getNextSpatialTemporalTrace\(\)](#).

**7.177.3.11 SpatialTemporalTrace SpatialTemporalDataReader::generateSpatialTemporalTrace ( std::string & *tracePath* ) [private]**

Generate the spatial temporal trace corresponding to the first valid unprocessed input file.

The unprocessed input file will be moved to the set of processed input files after creating the spatial temporal trace.

The path to the trace will be returned in the tracePath output parameter.

**Parameters**

|                  |                                        |
|------------------|----------------------------------------|
| <i>tracePath</i> | The path to the spatial temporal trace |
|------------------|----------------------------------------|

Definition at line 112 of file SpatialTemporalDataReader.cpp.

References [constructSpatialTemporalTrace\(\)](#), [getRandomValidUnprocessedInputFilepath\(\)](#), and [processValidInputFile\(\)](#).

**7.177.3.12 std::vector< std::string > SpatialTemporalDataReader::getFilesInFolder ( ) [private]**

Get the collection of files stored in the input folder.

Definition at line 293 of file SpatialTemporalDataReader.cpp.

References folderPath, multiscale::Filesystem::getFilesInFolder(), and INPUT\_FILES\_EXTENSION.

Referenced by updateInputFilesSets().

#### 7.177.3.13 SpatialTemporalTrace SpatialTemporalDataReader::getNextSpatialTemporalTrace ( )

Return the next spatial temporal trace.

Definition at line 38 of file SpatialTemporalDataReader.cpp.

References ERR\_NO\_VALID\_INPUT\_FILES\_REMAINING, generateSpatialTemporalTrace(), hasNext(), and MS\_throw.

Referenced by multiscale::verification::ModelCheckingManager::getNextSpatialTemporalTrace(), and readValidXmlFiles().

#### 7.177.3.14 SpatialTemporalTrace SpatialTemporalDataReader::getNextSpatialTemporalTrace ( std::string & tracePath )

Return the next spatial temporal trace and its path.

##### Parameters

|                  |                                        |
|------------------|----------------------------------------|
| <i>tracePath</i> | The path to the spatial temporal trace |
|------------------|----------------------------------------|

Definition at line 47 of file SpatialTemporalDataReader.cpp.

References ERR\_NO\_VALID\_INPUT\_FILES\_REMAINING, generateSpatialTemporalTrace(), hasNext(), and MS\_throw.

#### 7.177.3.15 std::set< std::string >::iterator SpatialTemporalDataReader::getRandomUnprocessedInputFile ( ) [private]

Get an iterator pointing to a random unprocessed input file.

Definition at line 261 of file SpatialTemporalDataReader.cpp.

References unprocessedInputFiles.

Referenced by getRandomValidUnprocessedInputFilepath().

#### 7.177.3.16 std::set< std::string >::iterator SpatialTemporalDataReader::getRandomValidUnprocessedInputFilepath ( ) [private]

Get an iterator pointing to a random valid unprocessed input file.

Definition at line 241 of file SpatialTemporalDataReader.cpp.

References ERR\_NO\_VALID\_INPUT\_FILES\_REMAINING, getRandomUnprocessedInputFile(), hasNext(), isValidInputFile(), MS\_throw, and processInvalidInputFile().

Referenced by generateSpatialTemporalTrace().

#### 7.177.3.17 bool SpatialTemporalDataReader::hasNext ( )

Check if there are any remaining valid unprocessed traces in the given folder.

This method does not automatically refresh the sets of input files.

Definition at line 28 of file SpatialTemporalDataReader.cpp.

References hasValidNext().

Referenced by `getNextSpatialTemporalTrace()`, `getRandomValidUnprocessedInputFilepath()`, `readValidXmlFiles()`, and `multiscale::verification::ModelCheckingManager::runModelCheckersForCurrentlyExistingTraces()`.

#### 7.177.3.18 `bool SpatialTemporalDataReader::isValidNext( ) [private]`

Check if there are any remaining valid unprocessed traces in the given folder.

Definition at line 77 of file `SpatialTemporalDataReader.cpp`.

References `isValidInputFile()`, `processInvalidInputFile()`, and `unprocessedInputFiles`.

Referenced by `hasNext()`.

#### 7.177.3.19 `void SpatialTemporalDataReader::initialise( const std::string & FolderPath ) [private]`

Initialise the sets for storing processed and unprocessed input files.

Parameters

|                   |                          |
|-------------------|--------------------------|
| <i>FolderPath</i> | Path to the input folder |
|-------------------|--------------------------|

Definition at line 56 of file `SpatialTemporalDataReader.cpp`.

References `FolderPath`, `initialise()`, and `validateFolderPath()`.

#### 7.177.3.20 `void SpatialTemporalDataReader::initialise( ) [private]`

Initialise the sets for storing processed and unprocessed input files.

Definition at line 65 of file `SpatialTemporalDataReader.cpp`.

References `clearInputFilesSets()`, and `updateInputFilesSets()`.

Referenced by `initialise()`, and `SpatialTemporalDataReader()`.

#### 7.177.3.21 `bool SpatialTemporalDataReader::isValidInputFile( const std::string & inputFilepath ) [private]`

Check if the given input file is valid.

An input file is valid if it is an xml file which conforms to the formal specification given in the xml schema (xsd file).

WARNING: The Timepoint class contains as members lists of spatial entities because the uniqueness of the spatial entities is determined using this method. If this method is no longer used then replace the lists in the Timepoint class with sets or unordered\_sets in order to ensure the uniqueness of the elements.

Parameters

|                      |                            |
|----------------------|----------------------------|
| <i>inputFilepath</i> | The path to the input file |
|----------------------|----------------------------|

Definition at line 298 of file `SpatialTemporalDataReader.cpp`.

References `INPUT_FILES_SCHEMA_PATH`, and `multiscale::XmlValidator::isValidXmlFile()`.

Referenced by `getRandomValidUnprocessedInputFilepath()`, and `isValidNext()`.

#### 7.177.3.22 `std::set< std::string >::iterator SpatialTemporalDataReader::processInvalidInputFile( const std::set< std::string >::iterator & invalidInputFileIterator ) [private]`

Process the invalid input file to which the given iterator points.

The iterator corresponds to a position in the list of unprocessed input files

**Parameters**

|                                 |                                                      |
|---------------------------------|------------------------------------------------------|
| <i>invalidInputFileIterator</i> | The iterator pointing to the invalid input file path |
|---------------------------------|------------------------------------------------------|

Definition at line 92 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

Referenced by getRandomValidUnprocessedInputFilepath(), and hasValidNext().

**7.177.3.23 void SpatialTemporalDataReader::processValidInputFile ( const std::set< std::string >::iterator & validInputFileIterator ) [private]**

Process the valid input file to which the given iterator points.

The iterator corresponds to a position in the list of unprocessed input files

**Parameters**

|                               |                                                    |
|-------------------------------|----------------------------------------------------|
| <i>validInputFileIterator</i> | The iterator pointing to the valid input file path |
|-------------------------------|----------------------------------------------------|

Definition at line 272 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

Referenced by generateSpatialTemporalTrace().

**7.177.3.24 void SpatialTemporalDataReader::refresh ( )**

Refresh the sets of processed and unprocessed traces' input files considering the given folder.

Definition at line 33 of file SpatialTemporalDataReader.cpp.

References updateInputFilesSets().

Referenced by multiscale::verification::ModelCheckingManager::updateTraceReader().

**7.177.3.25 void SpatialTemporalDataReader::setSpatialEntityMeasureValues ( const pt::ptree & spatialEntityTree, const std::shared\_ptr< SpatialEntity > & spatialEntity ) [private]**

Initialise the spatial entity measure values using the given spatialEntityTree.

**Parameters**

|                          |                                      |
|--------------------------|--------------------------------------|
| <i>spatialEntityTree</i> | The spatial entity tree              |
| <i>spatialEntity</i>     | The spatial entity to be initialised |

Definition at line 38 of file SpatialTemporalDataReaderAutoGenerated.cpp.

References multiscale::verification::Angle, multiscale::verification::Area, multiscale::verification::CentroidX, multiscale::verification::CentroidY, multiscale::verification::CircleMeasure, multiscale::verification::Clusteredness, multiscale::verification::Density, multiscale::verification::DistanceFromOrigin, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, and multiscale::verification::TriangleMeasure.

Referenced by addSpatialEntityToTimePoint().

**7.177.3.26 void SpatialTemporalDataReader::setSpatialEntitySemanticTypeValue ( const pt::ptree & spatialEntityTree, const std::shared\_ptr< SpatialEntity > & spatialEntity ) [private]**

Initialise the spatial entity semantic type value using the given spatialEntityTree.

If the value of the semantic type is not provided in the spatialEntityTree the default value 0 is used instead.

**Parameters**

|                          |                                      |
|--------------------------|--------------------------------------|
| <i>spatialEntityTree</i> | The spatial entity tree              |
| <i>spatialEntity</i>     | The spatial entity to be initialised |

Definition at line 229 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::SemanticType::DEFAULT\_VALUE, and LABEL\_SPATIAL\_ENTITY\_SEMANIC\_TYPE.

Referenced by addSpatialEntityToTimePoint().

**7.177.3.27 void SpatialTemporalDataReader::setTimePointValue ( const pt::ptree & *timePointTree*, TimePoint & *timePoint* ) [private]**

Set the value of the timepoint considering the given timepoint tree.

**Parameters**

|                      |                                                               |
|----------------------|---------------------------------------------------------------|
| <i>timePointTree</i> | Property tree representation of the timepoint                 |
| <i>timePoint</i>     | The <a href="#">TimePoint</a> representation of the timepoint |

Definition at line 160 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::setValue(), and timePointHasValue().

Referenced by convertTimePointPropertyTreeToTrace().

**7.177.3.28 bool SpatialTemporalDataReader::timePointHasValue ( const pt::ptree & *propertyTree*, unsigned long & *value* ) [private]**

Check if the provided property tree contains the attribute "value".

**Parameters**

|                     |                            |
|---------------------|----------------------------|
| <i>propertyTree</i> | The provided property tree |
| <i>value</i>        | The value (if it exists)   |

Definition at line 171 of file SpatialTemporalDataReader.cpp.

References LABEL\_TIMEPOINT\_VALUE.

Referenced by setTimePointValue().

**7.177.3.29 void SpatialTemporalDataReader::updateInputFilesSets ( ) [private]**

Update the sets of processed and unprocessed files by checking if the folder contents have been updated.

Definition at line 281 of file SpatialTemporalDataReader.cpp.

References getFilesInFolder(), processedInputFiles, and unprocessedInputFiles.

Referenced by initialise(), and refresh().

**7.177.3.30 void SpatialTemporalDataReader::validateFolderPath ( const std::string & *folderPath* ) [private]**

Check if the given folder path is valid.

The folder path is valid if it is a path pointing to a folder.

**Parameters**

|                         |  |
|-------------------------|--|
| <code>FolderPath</code> |  |
|-------------------------|--|

Definition at line 303 of file SpatialTemporalDataReader.cpp.

References `ERR_INVALID_FOLDER_PATH`, `multiscale::Filesystem::isValidFolderPath()`, and `MS_throw`.

Referenced by `initialise()`.

## 7.177.4 Member Data Documentation

**7.177.4.1** `const std::string SpatialTemporalDataReader::ERR_INVALID_FOLDER_PATH = "The provided path does not point to a folder. Please change." [static], [private]`

Definition at line 230 of file SpatialTemporalDataReader.hpp.

Referenced by `validateFolderPath()`.

**7.177.4.2** `const std::string SpatialTemporalDataReader::ERR_NO_VALID_INPUT_FILES_REMAINING = "There are no valid unprocessed input files remaining." [static], [private]`

Definition at line 231 of file SpatialTemporalDataReader.hpp.

Referenced by `getNextSpatialTemporalTrace()`, and `getRandomValidUnprocessedInputFilepath()`.

**7.177.4.3** `const std::string SpatialTemporalDataReader::ERR_UNDEFINED_SPATIAL_ENTITY_TYPE = "The provided spatial entity type is invalid." [static], [private]`

Definition at line 232 of file SpatialTemporalDataReader.hpp.

Referenced by `createDerivedSpatialEntity()`.

**7.177.4.4** `std::string multiscale::verification::SpatialTemporalDataReader::FolderPath [private]`

The path to the folder where all input files are stored

Definition at line 27 of file SpatialTemporalDataReader.hpp.

Referenced by `getFilesInFolder()`, and `initialise()`.

**7.177.4.5** `const std::string SpatialTemporalDataReader::INPUT_FILES_EXTENSION = ".xml" [static], [private]`

Definition at line 246 of file SpatialTemporalDataReader.hpp.

Referenced by `getFilesInFolder()`.

**7.177.4.6** `const std::string SpatialTemporalDataReader::INPUT_FILES_SCHEMA_PATH = "/usr/local/share/mule/config/verification/spatial-temporal/schema/MSTML_L1V1.xsd" [static], [private]`

Definition at line 247 of file SpatialTemporalDataReader.hpp.

Referenced by `isValidInputFile()`.

**7.177.4.7** `const std::string SpatialTemporalDataReader::LABEL_EXPERIMENT = "experiment" [static], [private]`

Definition at line 234 of file SpatialTemporalDataReader.hpp.

Referenced by `constructSpatialTemporalTrace()`.

7.177.4.8 `const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE = "numericStateVariable"`  
[static], [private]

Definition at line 237 of file SpatialTemporalDataReader.hpp.

Referenced by addEntitiesToTimePoint().

7.177.4.9 `const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE_NAME = "name"` [static],  
[private]

Definition at line 239 of file SpatialTemporalDataReader.hpp.

Referenced by addNumericStateVariableToTimePoint().

7.177.4.10 `const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE_SEMANTIC_TYPE =`  
"`<xmلاtr>.semanticType`" [static], [private]

Definition at line 238 of file SpatialTemporalDataReader.hpp.

Referenced by addNumericStateVariableToTimePoint().

7.177.4.11 `const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE_VALUE = "value"`  
[static], [private]

Definition at line 240 of file SpatialTemporalDataReader.hpp.

Referenced by addNumericStateVariableToTimePoint().

7.177.4.12 `const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY = "spatialEntity"` [static],  
[private]

Definition at line 242 of file SpatialTemporalDataReader.hpp.

Referenced by addEntitiesToTimePoint().

7.177.4.13 `const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_SEMANTIC_TYPE =`  
"`<xmلاtr>.semanticType`" [static], [private]

Definition at line 244 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntitySemanticTypeValue().

7.177.4.14 `const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_SPATIAL_TYPE = "<xmلاtr>.spatialType"`  
[static], [private]

Definition at line 243 of file SpatialTemporalDataReader.hpp.

Referenced by createDerivedSpatialEntity().

7.177.4.15 `const std::string SpatialTemporalDataReader::LABEL_TIMEPOINT_VALUE = "<xmلاtr>.value"` [static],  
[private]

Definition at line 235 of file SpatialTemporalDataReader.hpp.

Referenced by timePointHasValue().

7.177.4.16 std::set<std::string> multiscale::verification::SpatialTemporalDataReader::processedInputFiles [private]

The set of processed input files

Definition at line 24 of file SpatialTemporalDataReader.hpp.

Referenced by clearInputFilesSets(), processInvalidInputFile(), processValidInputFile(), updateInputFilesSets(), and ~SpatialTemporalDataReader().

7.177.4.17 std::set<std::string> multiscale::verification::SpatialTemporalDataReader::unprocessedInputFiles [private]

The set of unprocessed input files

Definition at line 25 of file SpatialTemporalDataReader.hpp.

Referenced by clearInputFilesSets(), getRandomUnprocessedInputFile(), hasValidNext(), processInvalidInputFile(), processValidInputFile(), updateInputFilesSets(), and ~SpatialTemporalDataReader().

The documentation for this class was generated from the following files:

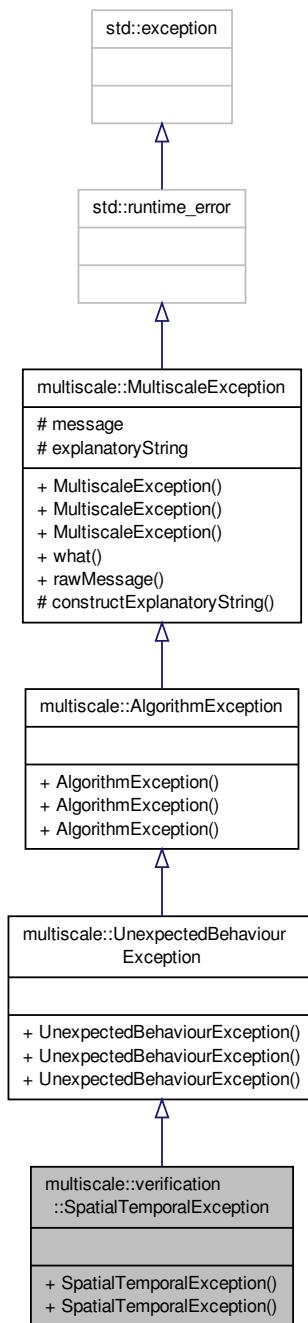
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/[SpatialTemporalDataReader.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[SpatialTemporalDataReader.cpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[SpatialTemporalDataReaderAutoGenerated.cpp](#)

## 7.178 multiscale::verification::SpatialTemporalException Class Reference

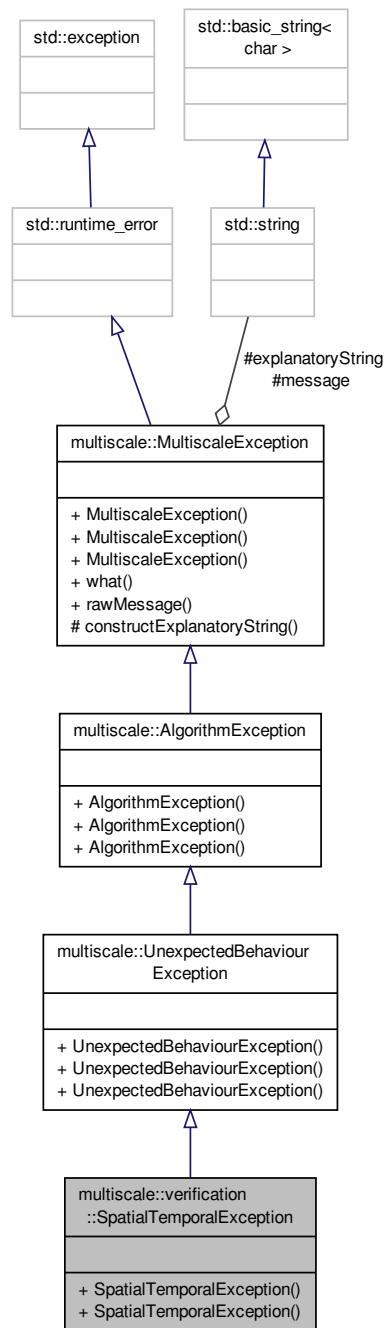
Class for representing a spatial temporal exception.

```
#include <SpatialTemporalException.hpp>
```

Inheritance diagram for multiscale::verification::SpatialTemporalException:



Collaboration diagram for multiscale::verification::SpatialTemporalException:



## Public Member Functions

- [SpatialTemporalException](#) (const std::string &file, int line, const std::string &msg)
- [SpatialTemporalException](#) (const std::string &file, int line, const char \*msg)

## Additional Inherited Members

### 7.178.1 Detailed Description

Class for representing a spatial temporal exception.

Definition at line 14 of file SpatialTemporalException.hpp.

### 7.178.2 Constructor & Destructor Documentation

7.178.2.1 `multiscale::verification::SpatialTemporalException::SpatialTemporalException ( const std::string & file, int line, const std::string & msg ) [inline], [explicit]`

Definition at line 18 of file SpatialTemporalException.hpp.

7.178.2.2 `multiscale::verification::SpatialTemporalException::SpatialTemporalException ( const std::string & file, int line, const char * msg ) [inline], [explicit]`

Definition at line 23 of file SpatialTemporalException.hpp.

The documentation for this class was generated from the following file:

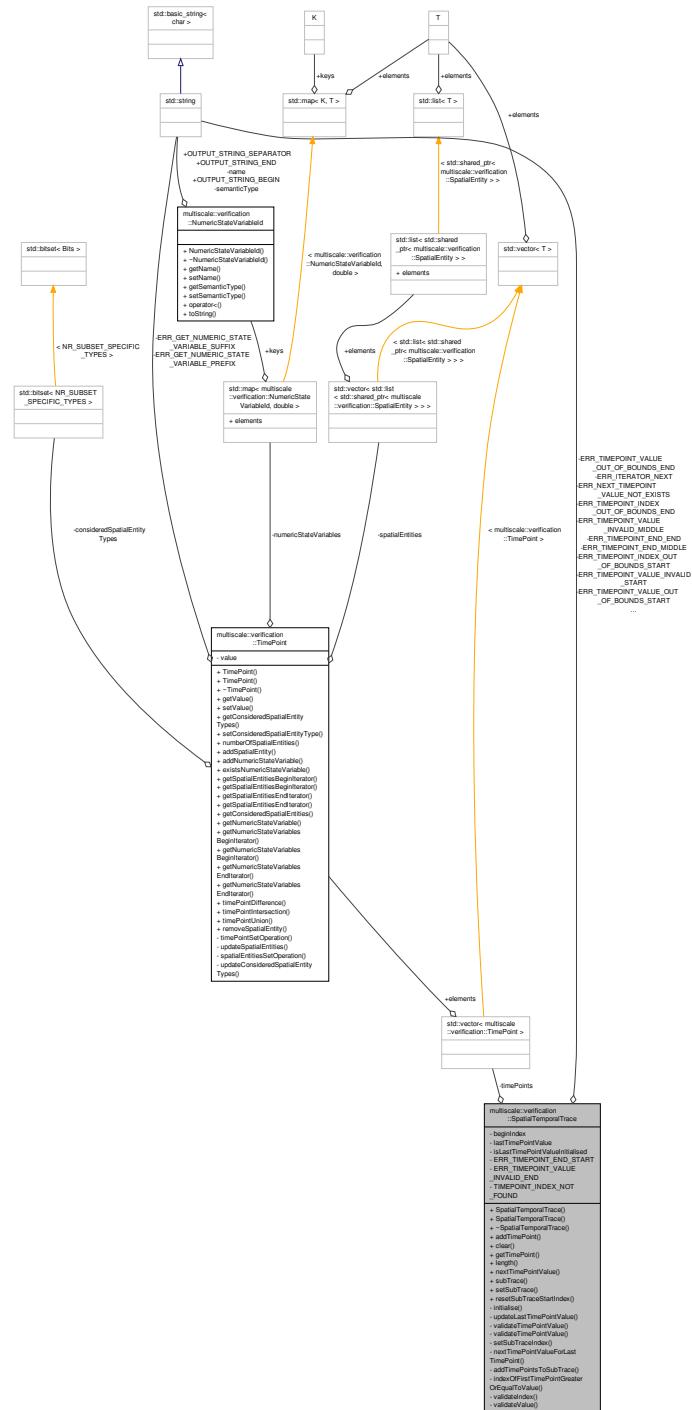
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp

## 7.179 multiscale::verification::SpatialTemporalTrace Class Reference

Class for representing a spatial temporal trace.

```
#include <SpatialTemporalTrace.hpp>
```

## Collaboration diagram for multiscale::verification::SpatialTemporalTrace



## Public Member Functions

- `SpatialTemporalTrace ()`
  - `SpatialTemporalTrace (const SpatialTemporalTrace &trace)`
  - `~SpatialTemporalTrace ()`
  - `void addTimePoint (const TimePoint &timePoint)`

*Add a time point to the array.*

- void `clear ()`  
*Clear all the stored timepoints and reinitialise.*
- `TimePoint getTimePoint (unsigned int index) const`  
*Get the time point at the given index in the array.*
- `unsigned int length () const`  
*Get the length of the spatial temporal trace (i.e. number of timepoints)*
- `unsigned long nextTimePointValue () const`  
*Get the value of the next timepoint considering beginIndex.*
- `SpatialTemporalTrace subTrace (unsigned int startIndex) const`  
*Get the subtrace containing timepoints with the index greater than the given index.*
- void `setSubTrace (unsigned long startValue)`  
*Set the subtrace containing timepoints with values greater than the given start value.*
- void `resetSubTraceStartIndex ()`  
*Reset the subtrace start index beginIndex to the value zero.*

## Private Member Functions

- void `initialise ()`  
*Initialise the member fields.*
- void `updateLastTimePointValue (TimePoint &timePoint)`  
*Update the last timepoint value.*
- void `validateTimePointValue (const TimePoint &timePoint)`  
*Check if the provided time point value is greater than the last time point value.*
- void `validateTimePointValue (unsigned long timePointValue)`  
*Check if the provided time point value is greater than the last time point value.*
- void `setSubTraceIndex (unsigned long startValue)`  
*Set the begin index for the subtrace starting with the given value.*
- `unsigned long nextTimePointValueForLastTimePoint () const`  
*Get the value of the next timepoint when beginIndex is the index of the last timepoint.*
- void `addTimePointsToSubTrace (SpatialTemporalTrace &subTrace, int startIndex, int endIndex) const`  
*Add the timepoints starting and ending with the given indices to the subtrace.*
- int `indexOfFirstTimePointGreaterOrEqualToValue (unsigned long value) const`  
*Get the index of the first timepoint which has a value greater than or equal to the given value.*
- void `validateIndex (unsigned int index) const`  
*Check if the provided index is smaller than the number of timepoints.*
- void `validateValue (unsigned long value) const`  
*Check if the provided value is smaller than or equal to the maximum timepoint value.*

## Private Attributes

- `unsigned int beginIndex`
- `std::vector< TimePoint > timePoints`
- `unsigned long lastTimePointValue`
- `bool isLastTimePointValueInitialised`

## Static Private Attributes

- static const std::string `ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_START` = "The provided timepoint index ("
- static const std::string `ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_END` = ") is out of bounds for the given spatial temporal trace."
- static const std::string `ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_START` = "The provided timepoint value ("
- static const std::string `ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_END` = ") is out of bounds for the given spatial temporal trace."
- static const std::string `ERR_TIMEPOINT_END_START` = "The provided end timepoint ("
- static const std::string `ERR_TIMEPOINT_END_MIDDLE` = ") should be greater or equal to the start timepoint ("
- static const std::string `ERR_TIMEPOINT_END_END` = ")."
- static const std::string `ERR_TIMEPOINT_VALUE_INVALID_START` = "The current timepoint value ("
- static const std::string `ERR_TIMEPOINT_VALUE_INVALID_MIDDLE` = ") should be greater than the previously added timepoint value ("
- static const std::string `ERR_TIMEPOINT_VALUE_INVALID_END` = ")."
- static const std::string `ERR_NEXT_TIMEPOINT_VALUE_NOT_EXISTS` = "The value of the last timepoint is the maximum value which can be represented by an unsigned long. Therefore a next timepoint value, which is greater than the value of the last timepoint, does not exist."
- static const std::string `ERR_ITERATOR_NEXT`
- static const int `TIMEPOINT_INDEX_NOT_FOUND` = -1

### 7.179.1 Detailed Description

Class for representing a spatial temporal trace.

Definition at line 15 of file SpatialTemporalTrace.hpp.

### 7.179.2 Constructor & Destructor Documentation

#### 7.179.2.1 SpatialTemporalTrace::SpatialTemporalTrace ( )

Definition at line 11 of file SpatialTemporalTrace.cpp.

References initialise().

#### 7.179.2.2 SpatialTemporalTrace::SpatialTemporalTrace ( const SpatialTemporalTrace & trace )

Definition at line 15 of file SpatialTemporalTrace.cpp.

#### 7.179.2.3 SpatialTemporalTrace::~SpatialTemporalTrace ( )

Definition at line 20 of file SpatialTemporalTrace.cpp.

### 7.179.3 Member Function Documentation

#### 7.179.3.1 void SpatialTemporalTrace::addTimePoint ( const TimePoint & timePoint )

Add a time point to the array.

**Parameters**

|                  |                               |
|------------------|-------------------------------|
| <i>timePoint</i> | Time point added to the array |
|------------------|-------------------------------|

Definition at line 22 of file SpatialTemporalTrace.cpp.

References timePoints, updateLastTimePointValue(), and validateTimePointValue().

Referenced by addTimePointsToSubTrace(), multiscale::verification::SpatialTemporalDataReader::addTimePointToTrace(), multiscale::verification::TemporalDataReader::createTimePointFromTokens(), multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues(), and initialiseTrace().

**7.179.3.2 void SpatialTemporalTrace::addTimePointsToSubTrace ( SpatialTemporalTrace & *subTrace*, int *startIndex*, int *endIndex* ) const [private]**

Add the timepoints starting and ending with the given indices to the subtrace.

**Parameters**

|                   |                              |
|-------------------|------------------------------|
| <i>subTrace</i>   | The resulting subtrace       |
| <i>startIndex</i> | The starting timepoint index |
| <i>endIndex</i>   | The end timepoint index      |

Definition at line 128 of file SpatialTemporalTrace.cpp.

References addTimePoint(), and timePoints.

Referenced by subTrace().

**7.179.3.3 void SpatialTemporalTrace::clear ( )**

Clear all the stored timepoints and reinitialise.

Definition at line 30 of file SpatialTemporalTrace.cpp.

References initialise().

Referenced by initialiseTrace().

**7.179.3.4 TimePoint SpatialTemporalTrace::getTimePoint ( unsigned int *index* ) const**

Get the time point at the given index in the array.

**Parameters**

|              |                                        |
|--------------|----------------------------------------|
| <i>index</i> | The index of the position in the array |
|--------------|----------------------------------------|

Definition at line 34 of file SpatialTemporalTrace.cpp.

References beginIndex, timePoints, and validateIndex().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollection(), multiscale::verification::TemporalNumericVisitor::operator()(), printSpatialTemporalTrace(), and printTrace().

**7.179.3.5 int SpatialTemporalTrace::indexOfFirstTimePointGreaterOrEqualToValue ( unsigned long *value* ) const [private]**

Get the index of the first timepoint which has a value greater than or equal to the given value.

**Parameters**

|                    |                 |
|--------------------|-----------------|
| <code>value</code> | The given value |
|--------------------|-----------------|

Definition at line 135 of file SpatialTemporalTrace.cpp.

References beginIndex, TIMEPOINT\_INDEX\_NOT\_FOUND, and timePoints.

Referenced by setSubTraceIndex().

#### 7.179.3.6 void SpatialTemporalTrace::initialise( ) [private]

Initialise the member fields.

Definition at line 70 of file SpatialTemporalTrace.cpp.

References beginIndex, isLastTimePointValueInitialised, lastTimePointValue, and timePoints.

Referenced by clear(), and SpatialTemporalTrace().

#### 7.179.3.7 unsigned int SpatialTemporalTrace::length( ) const

Get the length of the spatial temporal trace (i.e. number of timepoints)

Definition at line 40 of file SpatialTemporalTrace.cpp.

References beginIndex, and timePoints.

Referenced by multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluate(), printSpatialTemporalTrace(), and printTrace().

#### 7.179.3.8 unsigned long SpatialTemporalTrace::nextTimePointValue( ) const

Get the value of the next timepoint considering beginIndex.

Definition at line 44 of file SpatialTemporalTrace.cpp.

References beginIndex, nextTimePointValueForLastTimePoint(), and timePoints.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluatePrecedingLogicProperties(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollection(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollectionTimepoints(), and multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

#### 7.179.3.9 unsigned long SpatialTemporalTrace::nextTimePointValueForLastTimePoint( ) const [private]

Get the value of the next timepoint when beginIndex is the index of the last timepoint.

Return maximum unsigned long value if the value of the last timepoint is smaller than the maximum unsigned long value. Otherwise throw an exception.

Definition at line 120 of file SpatialTemporalTrace.cpp.

References ERR\_NEXT\_TIMEPOINT\_VALUE\_NOT\_EXISTS, MS\_throw, and timePoints.

Referenced by nextTimePointValue().

#### 7.179.3.10 void SpatialTemporalTrace::resetSubTracestartIndex( )

Reset the subtrace start index beginIndex to the value zero.

Definition at line 66 of file SpatialTemporalTrace.cpp.

References beginIndex.

#### 7.179.3.11 void SpatialTemporalTrace::setSubTrace ( unsigned long *startValue* )

Set the subtrace containing timepoints with values greater than the given start value.

**Parameters**

|                   |                                    |
|-------------------|------------------------------------|
| <i>startValue</i> | The starting value of the subtrace |
|-------------------|------------------------------------|

Definition at line 61 of file SpatialTemporalTrace.cpp.

References setSubTraceIndex(), and validateValue().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluatePrecedingLogicProperties(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollection(), multiscale::verification::NumericMeasureCollectionVisitor::evaluateTemporalNumericMeasureCollectionTimepoints(), and multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

#### 7.179.3.12 void SpatialTemporalTrace::setSubTraceIndex ( unsigned long *startValue* ) [private]

Set the begin index for the subtrace starting with the given value.

**Parameters**

|                   |                                              |
|-------------------|----------------------------------------------|
| <i>startValue</i> | The starting timepoint value of the subtrace |
|-------------------|----------------------------------------------|

Definition at line 110 of file SpatialTemporalTrace.cpp.

References beginIndex, indexOfFirstTimePointGreaterOrEqualToValue(), TIMEPOINT\_INDEX\_NOT\_FOUND, and timePoints.

Referenced by setSubTrace().

#### 7.179.3.13 SpatialTemporalTrace SpatialTemporalTrace::subTrace ( unsigned int *startIndex* ) const

Get the subtrace containing timepoints with the index greater than the given index.

**Parameters**

|                   |                                    |
|-------------------|------------------------------------|
| <i>startIndex</i> | The starting index of the subtrace |
|-------------------|------------------------------------|

Definition at line 52 of file SpatialTemporalTrace.cpp.

References addTimePointsToSubTrace(), beginIndex, timePoints, and validateIndex().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateNextKLogicProperty(), and multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericMeasure().

#### 7.179.3.14 void SpatialTemporalTrace::updateLastTimePointValue ( TimePoint & *timePoint* ) [private]

Update the last timepoint value.

**Parameters**

|                  |                          |
|------------------|--------------------------|
| <i>timePoint</i> | The last added timepoint |
|------------------|--------------------------|

Definition at line 79 of file SpatialTemporalTrace.cpp.

References multiscale::verification::TimePoint::getValue(), lastTimePointValue, and multiscale::verification::TimePoint::setValue().

Referenced by addTimePoint().

#### 7.179.3.15 void SpatialTemporalTrace::validateIndex ( *unsigned int index* ) const [private]

Check if the provided index is smaller than the number of timepoints.

**Parameters**

|              |                    |
|--------------|--------------------|
| <i>index</i> | The provided index |
|--------------|--------------------|

Definition at line 150 of file SpatialTemporalTrace.cpp.

References beginIndex, ERR\_TIMEPOINT\_INDEX\_OUT\_OF\_BOUNDS\_END, ERR\_TIMEPOINT\_INDEX\_OUT\_OF\_BOUNDS\_START, MS\_throw\_detailed, and timePoints.

Referenced by getTimePoint(), and subTrace().

#### 7.179.3.16 void SpatialTemporalTrace::validateTimePointValue ( *const TimePoint & timePoint* ) [private]

Check if the provided time point value is greater than the last time point value.

The timepoint is considered to be uninitialized if the value is equal to the maximum value which can be represented as an unsigned long. Otherwise if the timepoint value is less or equal to the lastTimePointValue then an exception is thrown.

**Parameters**

|                  |                      |
|------------------|----------------------|
| <i>timePoint</i> | The given time point |
|------------------|----------------------|

Definition at line 89 of file SpatialTemporalTrace.cpp.

References multiscale::verification::TimePoint::getValue().

Referenced by addTimePoint().

#### 7.179.3.17 void SpatialTemporalTrace::validateTimePointValue ( *unsigned long timePointValue* ) [private]

Check if the provided time point value is greater than the last time point value.

The timepoint is considered to be uninitialized if the value is equal to the maximum value which can be represented as an unsigned long. Otherwise if the timepoint value is less or equal to the lastTimePointValue then an exception is thrown.

**Parameters**

|                       |                            |
|-----------------------|----------------------------|
| <i>timePointValue</i> | The value of the timepoint |
|-----------------------|----------------------------|

Definition at line 95 of file SpatialTemporalTrace.cpp.

References ERR\_TIMEPOINT\_VALUE\_INVALID\_END, ERR\_TIMEPOINT\_VALUE\_INVALID\_MIDDLE, ERR\_TIMEPOINT\_VALUE\_INVALID\_START, isLastTimePointValueInitialised, lastTimePointValue, and MS\_throw\_detailed.

#### 7.179.3.18 void SpatialTemporalTrace::validateValue ( *unsigned long value* ) const [private]

Check if the provided value is smaller than or equal to the maximum timepoint value.

**Parameters**

|              |                    |
|--------------|--------------------|
| <i>value</i> | The provided value |
|--------------|--------------------|

Definition at line 157 of file SpatialTemporalTrace.cpp.

References ERR\_TIMEPOINT\_VALUE\_OUT\_OF\_BOUNDS\_END, ERR\_TIMEPOINT\_VALUE\_OUT\_OF\_BOUNDS\_START, MS\_throw\_detailed, and timePoints.

Referenced by `setSubTrace()`.

#### 7.179.4 Member Data Documentation

**7.179.4.1** `unsigned int multiscale::verification::SpatialTemporalTrace::beginIndex` [private]

The corresponding begin index

Definition at line 19 of file `SpatialTemporalTrace.hpp`.

Referenced by `getTimePoint()`, `indexOfFirstTimePointGreaterOrEqualToValue()`, `initialise()`, `length()`, `nextTimePointValue()`, `resetSubTraceStartIndex()`, `setSubTraceIndex()`, `subTrace()`, and `validateIndex()`.

**7.179.4.2** `const std::string SpatialTemporalTrace::ERR_ITERATOR_NEXT` [static], [private]

**Initial value:**

```
= "There is no next timepoint which the iterator can"
 " before"
 " available"
 " return. Please use the hasNext() method
 " to ensure there are further timepoints
 " before calling the next() method."
```

Definition at line 157 of file `SpatialTemporalTrace.hpp`.

**7.179.4.3** `const std::string SpatialTemporalTrace::ERR_NEXT_TIMEPOINT_VALUE_NOT_EXISTS` = "The value of the last
timepoint is the maximum value which can be represented by an unsigned long. Therefore a next timepoint value,
which is greater than the value of the last timepoint, does not exist." [static], [private]

Definition at line 155 of file `SpatialTemporalTrace.hpp`.

Referenced by `nextTimePointValueForLastTimePoint()`.

**7.179.4.4** `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_END` = ":" [static], [private]

Definition at line 149 of file `SpatialTemporalTrace.hpp`.

**7.179.4.5** `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_MIDDLE` = ")" should be greater or equal to the start
timepoint "(" [static], [private]

Definition at line 148 of file `SpatialTemporalTrace.hpp`.

**7.179.4.6** `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_START` = "The provided end timepoint (
[static], [private]

Definition at line 147 of file `SpatialTemporalTrace.hpp`.

**7.179.4.7** `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_END` = ") is out of bounds for
the given spatial temporal trace." [static], [private]

Definition at line 142 of file `SpatialTemporalTrace.hpp`.

Referenced by `validateIndex()`.

7.179.4.8 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_INDEX\_OUT\_OF\_BOUNDS\_START = "The provided timepoint index (" [static], [private]

Definition at line 141 of file SpatialTemporalTrace.hpp.

Referenced by validateIndex().

7.179.4.9 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_VALUE\_INVALID\_END = ")." [static], [private]

Definition at line 153 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.179.4.10 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_VALUE\_INVALID\_MIDDLE = ") should be greater than the previously added timepoint value (" [static], [private]

Definition at line 152 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.179.4.11 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_VALUE\_INVALID\_START = "The current timepoint value (" [static], [private]

Definition at line 151 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.179.4.12 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_VALUE\_OUT\_OF\_BOUNDS\_END = ") is out of bounds for the given spatial temporal trace." [static], [private]

Definition at line 145 of file SpatialTemporalTrace.hpp.

Referenced by validateValue().

7.179.4.13 const std::string SpatialTemporalTrace::ERR\_TIMEPOINT\_VALUE\_OUT\_OF\_BOUNDS\_START = "The provided timepoint value (" [static], [private]

Definition at line 144 of file SpatialTemporalTrace.hpp.

Referenced by validateValue().

7.179.4.14 bool multiscale::verification::SpatialTemporalTrace::isLastTimePointValueInitialised [private]

Flag to indicate if the last time point value was initialised

Definition at line 24 of file SpatialTemporalTrace.hpp.

Referenced by initialise(), and validateTimePointValue().

7.179.4.15 unsigned long multiscale::verification::SpatialTemporalTrace::lastTimePointValue [private]

The value of the last added timepoint

Definition at line 22 of file SpatialTemporalTrace.hpp.

Referenced by initialise(), updateLastTimePointValue(), and validateTimePointValue().

7.179.4.16 const int SpatialTemporalTrace::TIMEPOINT\_INDEX\_NOT\_FOUND = -1 [static], [private]

Definition at line 159 of file SpatialTemporalTrace.hpp.

Referenced by indexOfFirstTimePointGreaterOrEqualToValue(), and setSubTraceIndex().

7.179.4.17 std::vector<TimePoint> multiscale::verification::SpatialTemporalTrace::timePoints [private]

The array of time points

Definition at line 21 of file SpatialTemporalTrace.hpp.

Referenced by addTimePoint(), addTimePointsToSubTrace(), getTimePoint(), indexOfFirstTimePointGreaterOrEqualToValue(), initialise(), length(), nextTimePointValue(), nextTimePointValueForLastTimePoint(), setSubTraceIndex(), subTrace(), validateIndex(), and validateValue().

The documentation for this class was generated from the following files:

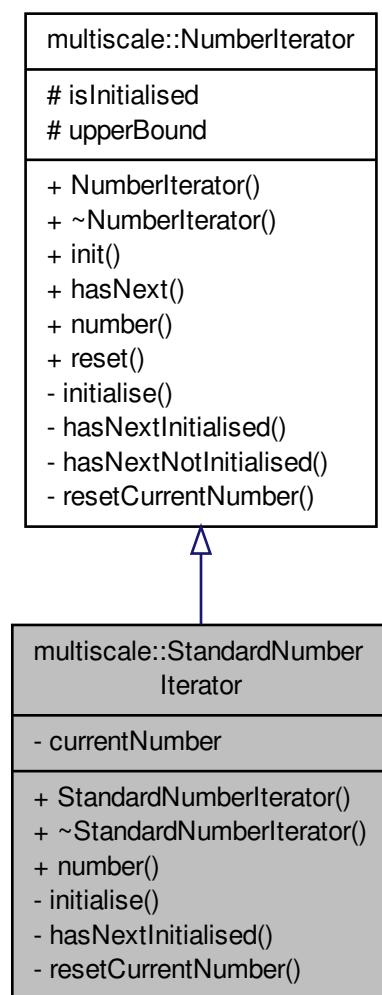
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[SpatialTemporalTrace.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[SpatialTemporalTrace.cpp](#)

## 7.180 multiscale::StandardNumberIterator Class Reference

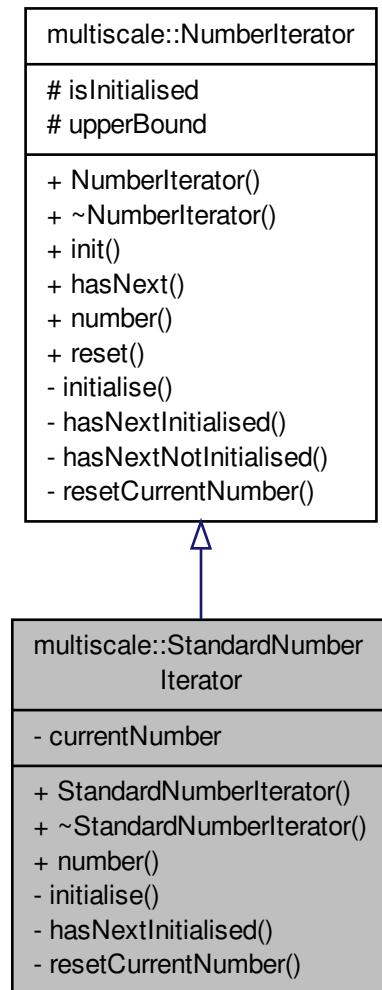
Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.

```
#include <StandardNumberIterator.hpp>
```

Inheritance diagram for multiscale::StandardNumberIterator:



Collaboration diagram for multiscale::StandardNumberIterator:



## Public Member Functions

- `StandardNumberIterator` (unsigned int `upperBound`)
- `~StandardNumberIterator ()`
- unsigned int `number ()`

*Get the number pointed by the iterator.*

## Private Member Functions

- void `initialise ()`  
*Initialise the value of the current number.*
- bool `hasNextInitialised ()`  
*Check if there is a next number when in initialised state.*
- void `resetCurrentNumber ()`  
*Reset the current number to the initial value.*

## Private Attributes

- unsigned int [currentNumber](#)

## Additional Inherited Members

### 7.180.1 Detailed Description

Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.

Definition at line 10 of file StandardNumberIterator.hpp.

### 7.180.2 Constructor & Destructor Documentation

#### 7.180.2.1 StandardNumberIterator::StandardNumberIterator ( unsigned int *upperBound* )

Definition at line 6 of file StandardNumberIterator.cpp.

References [initialise\(\)](#), and [multiscale::NumberIterator::reset\(\)](#).

#### 7.180.2.2 StandardNumberIterator::~StandardNumberIterator ( )

Definition at line 11 of file StandardNumberIterator.cpp.

### 7.180.3 Member Function Documentation

#### 7.180.3.1 bool StandardNumberIterator::hasNextInitialised ( ) [private], [virtual]

Check if there is a next number when in initialised state.

Implements [multiscale::NumberIterator](#).

Definition at line 19 of file StandardNumberIterator.cpp.

References [currentNumber](#), and [multiscale::NumberIterator::upperBound](#).

#### 7.180.3.2 void StandardNumberIterator::initialise ( ) [private], [virtual]

Initialise the value of the current number.

Implements [multiscale::NumberIterator](#).

Definition at line 17 of file StandardNumberIterator.cpp.

Referenced by [StandardNumberIterator\(\)](#).

#### 7.180.3.3 unsigned int StandardNumberIterator::number ( ) [virtual]

Get the number pointed by the iterator.

Implements [multiscale::NumberIterator](#).

Definition at line 13 of file StandardNumberIterator.cpp.

References [currentNumber](#).

**7.180.3.4 void StandardNumberIterator::resetCurrentNumber( ) [private], [virtual]**

Reset the current number to the initial value.

Implements [multiscale::NumberIterator](#).

Definition at line 29 of file StandardNumberIterator.cpp.

References currentNumber.

**7.180.4 Member Data Documentation****7.180.4.1 unsigned int multiscale::StandardNumberIterator::currentNumber [private]**

The current number

Definition at line 14 of file StandardNumberIterator.hpp.

Referenced by hasNextInitialised(), number(), and resetCurrentNumber().

The documentation for this class was generated from the following files:

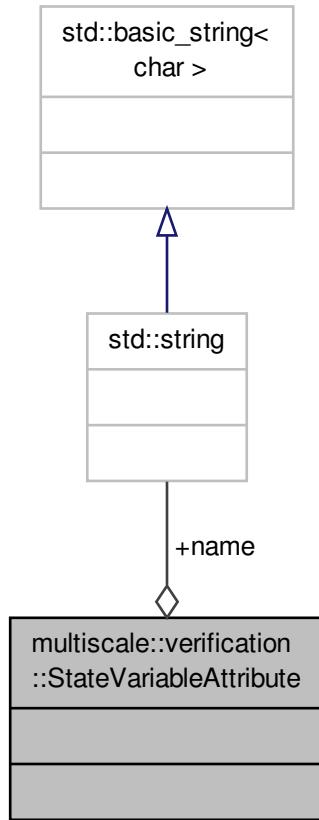
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/[StandardNumberIterator.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/[StandardNumberIterator.cpp](#)

**7.181 multiscale::verification::StateVariableAttribute Class Reference**

Class for representing a state variable attribute.

```
#include <StateVariableAttribute.hpp>
```

Collaboration diagram for multiscale::verification::StateVariableAttribute:



## Public Attributes

- `std::string name`

### 7.181.1 Detailed Description

Class for representing a state variable attribute.

Definition at line 14 of file StateVariableAttribute.hpp.

### 7.181.2 Member Data Documentation

#### 7.181.2.1 `std::string multiscale::verification::StateVariableAttribute::name`

Name of the state variable

Definition at line 18 of file StateVariableAttribute.hpp.

Referenced by `multiscale::verification::TemporalNumericVisitor::operator()()`, and `multiscale::verification::NumericVisitor::operator()()`.

The documentation for this class was generated from the following file:

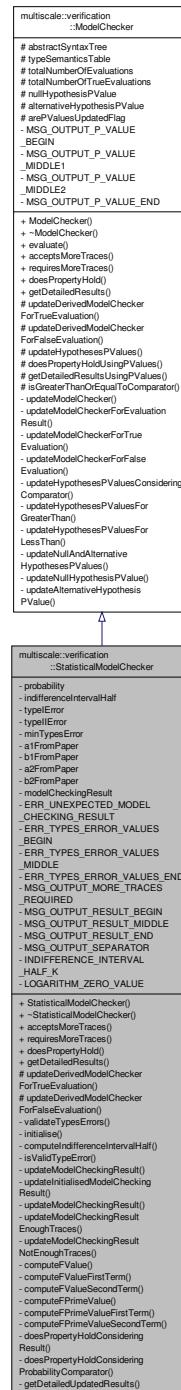
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[StateVariableAttribute.hpp](#)

## 7.182 multiscale::verification::StatisticalModelChecker Class Reference

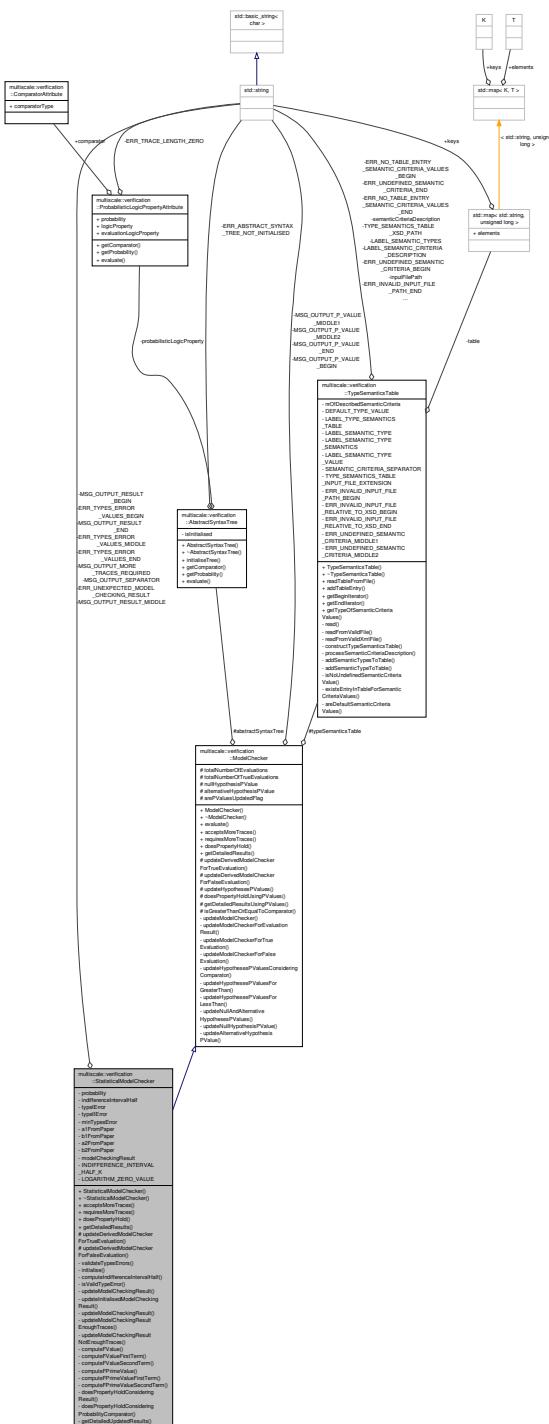
Class used to run statistical model checking tasks.

```
#include <StatisticalModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::StatisticalModelChecker:



Collaboration diagram for multiscale::verification::StatisticalModelChecker:



## Public Member Functions

- `StatisticalModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, const TypeSemanticsTable &typeSemanticsTable, double typeIError, double typeIIError)`
- `~StatisticalModelChecker ()`
- `bool acceptsMoreTraces () override`

*Check if more traces are accepted for evaluating the logic property.*

- bool `requiresMoreTraces ()` override  
*Check if more traces are required for evaluating the logic property.*
- bool `doesPropertyHold ()` override  
*Check if the given property holds.*
- std::string `getDetailedResults ()` override  
*Get the detailed description of the results.*

## Protected Member Functions

- void `updateDerivedModelCheckerForTrueEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.*
- void `updateDerivedModelCheckerForFalseEvaluation ()` override  
*Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.*

## Private Member Functions

- void `validateTypesErrors (double typeIError, double typeIIError)`  
*Validate the probability of type I and type II errors to occur.*
- void `initialise ()`  
*Initialisation of some of the class members.*
- double `computeIndifferenceIntervalHalf (double probability)`  
*Compute the value of the indifference interval half considering the given probability.*
- bool `isValidTypeError (double typeError)`  
*Check if the given type I/II error probability is valid.*
- void `updateModelCheckingResult ()`  
*Update the result of the model checking task.*
- void `updateInitialisedModelCheckingResult ()`  
*Update the result of the model checking task which was already initialised.*
- void `updateModelCheckingResult (double f, double fPrime)`  
*Update the result of the model checking task considering the given values.*
- void `updateModelCheckingResultEnoughTraces (double f, double fPrime)`  
*Update the result of the model checking task considering the given values when enough traces have been provided.*
- void `updateModelCheckingResultNotEnoughTraces ()`  
*Update the result of the model checking task when not enough traces were provided.*
- double `computeFValue ()`  
*Compute the value of f (from original paper)*
- double `computeFValueFirstTerm ()`  
*Compute the value of the first term of f (from original paper)*
- double `computeFValueSecondTerm ()`  
*Compute the value of the second term of f (from original paper)*
- double `computeFPrimeValue ()`  
*Compute the value of f' (from original paper)*
- double `computeFPrimeValueFirstTerm ()`  
*Compute the value of the first term of f' (from original paper)*
- double `computeFPrimeValueSecondTerm ()`  
*Compute the value of the second term of f' (from original paper)*
- bool `doesPropertyHoldConsideringResult ()`  
*Check if the given property holds considering the obtained model checking result.*
- bool `doesPropertyHoldConsideringProbabilityComparator (bool isNullHypothesisTrue)`

*Check if the given property holds considering the obtained answer and probability comparator (i.e. <=, >=)*

- std::string `getDetailedUpdatedResults ()`

*Get the detailed description of the updated results.*

## Private Attributes

- double `probability`
- double `indifferenceIntervalHalf`
- double `typeIError`
- double `typeIIError`
- double `minTypesError`
- double `a1FromPaper`
- double `b1FromPaper`
- double `a2FromPaper`
- double `b2FromPaper`
- StatisticalModelCheckingResult `modelCheckingResult`

## Static Private Attributes

- static const std::string `ERR_UNEXPECTED_MODEL_CHECKING_RESULT` = "An invalid statistical model checking result was obtained. Please check source code."
  - static const std::string `ERR_TYPES_ERROR_VALUES_BEGIN` = "The provided probabilities of type I and type II errors ("
  - static const std::string `ERR_TYPES_ERROR_VALUES_MIDDLE` = ", "
  - static const std::string `ERR_TYPES_ERROR_VALUES_END` = ") should be greater than zero and less or equal to 1. Please change."
  - static const std::string `MSG_OUTPUT_MORE_TRACES_REQUIRED` = "More traces are required to provide a true/false answer assuming the given probabilities of type I and type II errors. Probabilistic black-box model checking was used instead to provide an answer."
  - static const std::string `MSG_OUTPUT_RESULT_BEGIN` = "The provided answer is given for the `probability` of type I errors = "
  - static const std::string `MSG_OUTPUT_RESULT_MIDDLE` = " and the `probability` of type II errors = "
  - static const std::string `MSG_OUTPUT_RESULT_END` = ""
  - static const std::string `MSG_OUTPUT_SEPARATOR` = " "
  - static const unsigned int `INDIFFERENCE_INTERVAL_HALF_K` = (std::numeric\_limits<unsigned int>::max() >> 1)
- The value of this constant should be much greater than 1.*
- static const double `LOGARITHM_ZERO_VALUE` = (std::numeric\_limits<double>::lowest() / 1E+10)
- The value of this constant should be a large negative number.*

## Additional Inherited Members

### 7.182.1 Detailed Description

Class used to run statistical model checking tasks.

The implementation of this class is (partially) based on the algorithms described in the following paper:

C. H. Koh, S. K. Palaniappan, P. S. Thiagarajan, and L. Wong, 'Improved statistical model checking methods for pathway analysis', BMC Bioinformatics, vol. 13, no. Suppl 17, p. S15, Dec. 2012.

In our implementation the variables in the original paper (right hand side of the assignments) have been given the following new names (left hand side of assignments):

`probability` =  $\theta$

`indifference =  $\delta$`

`typeIError =  $\alpha$`

`typeIIError =  $\beta$`

`minTypesError =  $\gamma$`

`totalNumberOfEvaluations =  $n$`

`totalNumberOfTrueEvaluations =  $d$`

Definition at line 50 of file StatisticalModelChecker.hpp.

## 7.182.2 Constructor & Destructor Documentation

**7.182.2.1 StatisticalModelChecker::StatisticalModelChecker ( const AbstractSyntaxTree & *abstractSyntaxTree*, const TypeSemanticsTable & *typeSemanticsTable*, double *typeIError*, double *typeIIError* )**

Definition at line 13 of file StatisticalModelChecker.cpp.

References initialise(), minTypesError, typeIError, typeIIError, and validateTypesErrors().

**7.182.2.2 StatisticalModelChecker::~StatisticalModelChecker ( )**

Definition at line 27 of file StatisticalModelChecker.cpp.

## 7.182.3 Member Function Documentation

**7.182.3.1 bool StatisticalModelChecker::acceptsMoreTraces ( ) [override], [virtual]**

Check if more traces are accepted for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 29 of file StatisticalModelChecker.cpp.

References modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, and updateModelCheckingResult().

Referenced by requiresMoreTraces().

**7.182.3.2 double StatisticalModelChecker::computeFPrimeValue ( ) [private]**

Compute the value of  $f'$  (from original paper)

Definition at line 162 of file StatisticalModelChecker.cpp.

References computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(), multiscale::verification::ModelChecker::totalNumberOfEvaluations, and multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations.

Referenced by updateInitialisedModelCheckingResult().

**7.182.3.3 double StatisticalModelChecker::computeFPrimeValueFirstTerm ( ) [private]**

Compute the value of the first term of  $f'$  (from original paper)

If the value inside the logarithm is equal to zero than the returned value is equal to LOGARITHM\_ZERO\_VALUE. Otherwise the value of the logarithm is computed and returned.

Definition at line 172 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM\_ZERO\_VALUE, and probability.

Referenced by computeFPrimeValue().

#### 7.182.3.4 double StatisticalModelChecker::computeFPrimeValueSecondTerm( ) [private]

Compute the value of the second term of f' (from original paper)

If the value inside the logarithm is equal to zero than the returned value is equal to LOGARITHM\_ZERO\_VALUE. Otherwise the value of the logarithm is computed and returned.

Definition at line 181 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM\_ZERO\_VALUE, and probability.

Referenced by computeFPrimeValue().

#### 7.182.3.5 double StatisticalModelChecker::computeFValue( ) [private]

Compute the value of f (from original paper)

Definition at line 134 of file StatisticalModelChecker.cpp.

References computeFValueFirstTerm(), computeFValueSecondTerm(), multiscale::verification::ModelChecker::totalNumberOfEvaluations, and multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations.

Referenced by updateInitialisedModelCheckingResult().

#### 7.182.3.6 double StatisticalModelChecker::computeFValueFirstTerm( ) [private]

Compute the value of the first term of f (from original paper)

If the value inside the logarithm is equal to zero than the returned value is equal to LOGARITHM\_ZERO\_VALUE. Otherwise the value of the logarithm is computed and returned.

Definition at line 144 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM\_ZERO\_VALUE, and probability.

Referenced by computeFValue().

#### 7.182.3.7 double StatisticalModelChecker::computeFValueSecondTerm( ) [private]

Compute the value of the second term of f (from original paper)

If the value inside the logarithm is equal to zero than the returned value is equal to LOGARITHM\_ZERO\_VALUE. Otherwise the value of the logarithm is computed and returned.

Definition at line 153 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM\_ZERO\_VALUE, and probability.

Referenced by computeFValue().

#### 7.182.3.8 double StatisticalModelChecker::computeIndifferenceIntervalHalf( double probability ) [private]

Compute the value of the indifference interval half considering the given probability.

indifferenceIntervalHalf = max(0, min(probability, 1 - probability) - eps)

## Parameters

|                    |                              |
|--------------------|------------------------------|
| <i>probability</i> | The value of the probability |
|--------------------|------------------------------|

Definition at line 76 of file StatisticalModelChecker.cpp.

References INDIFFERENCE\_INTERVAL\_HALF\_K.

Referenced by initialise().

## 7.182.3.9 bool StatisticalModelChecker::doesPropertyHold( ) [override], [virtual]

Check if the given property holds.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 39 of file StatisticalModelChecker.cpp.

References doesPropertyHoldConsideringResult(), and updateModelCheckingResult().

7.182.3.10 bool StatisticalModelChecker::doesPropertyHoldConsideringProbabilityComparator ( bool *isNullHypothesisTrue* ) [private]

Check if the given property holds considering the obtained answer and probability comparator (i.e.  $\leq$ ,  $\geq$ )

For queries of type : a)  $P \geq \theta[\phi]$  the *isNullHypothesisTrue* flag value is returned b)  $P \leq \theta[\phi]$  the *!(isNullHypothesisTrue)* flag value is returned

## Parameters

|                             |                                                                                    |
|-----------------------------|------------------------------------------------------------------------------------|
| <i>isNullHypothesisTrue</i> | Flag indicating if the null hypothesis is true considering a $P \geq [\phi]$ query |
|-----------------------------|------------------------------------------------------------------------------------|

Definition at line 209 of file StatisticalModelChecker.cpp.

References multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator().

Referenced by doesPropertyHoldConsideringResult().

## 7.182.3.11 bool StatisticalModelChecker::doesPropertyHoldConsideringResult( ) [private]

Check if the given property holds considering the obtained model checking result.

Definition at line 190 of file StatisticalModelChecker.cpp.

References doesPropertyHoldConsideringProbabilityComparator(), multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues(), ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT, multiscale::verification::FA\_LSE, modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MS\_throw, and multiscale::verification::TRUE.

Referenced by doesPropertyHold().

## 7.182.3.12 std::string StatisticalModelChecker::getDetailedResults( ) [override], [virtual]

Get the detailed description of the results.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 45 of file StatisticalModelChecker.cpp.

References getDetailedUpdatedResults(), and updateModelCheckingResult().

### 7.182.3.13 std::string StatisticalModelChecker::getDetailedUpdatedResults( ) [private]

Get the detailed description of the updated results.

Definition at line 217 of file StatisticalModelChecker.cpp.

References multiscale::verification::ModelChecker::getDetailedResultsUsingPValues(), modelCheckingResult, multiscale::verification::MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_MORE\_TRACES\_REQUIRED, MSG\_OUTPUT\_RESULT\_BEGIN, MSG\_OUTPUT\_RESULT\_END, MSG\_OUTPUT\_RESULT\_MIDDLE, MSG\_OUTPUT\_SEPARATOR, multiscale::StringManipulator::toString(), typeIError, and typeIIError.

Referenced by getDetailedResults().

### 7.182.3.14 void StatisticalModelChecker::initialise( ) [private]

Initialisation of some of the class members.

Definition at line 66 of file StatisticalModelChecker.cpp.

References a1FromPaper, a2FromPaper, multiscale::verification::ModelChecker::abstractSyntaxTree, b1FromPaper, b2FromPaper, computeIndifferenceIntervalHalf(), multiscale::verification::AbstractSyntaxTree::getProbability(), indifferenceIntervalHalf, minTypesError, probability, typeIError, and typeIIError.

Referenced by StatisticalModelChecker().

### 7.182.3.15 bool StatisticalModelChecker::isValidTypeError( double typeError ) [private]

Check if the given type I/II error probability is valid.

The probability of the type I/II error to occur should be greater than zero and less than one

**Parameters**

|                  |                                               |
|------------------|-----------------------------------------------|
| <i>typeError</i> | The probability of a type I/II error to occur |
|------------------|-----------------------------------------------|

Definition at line 83 of file StatisticalModelChecker.cpp.

Referenced by validateTypesErrors().

### 7.182.3.16 bool StatisticalModelChecker::requiresMoreTraces( ) [override], [virtual]

Check if more traces are required for evaluating the logic property.

Implements [multiscale::verification::ModelChecker](#).

Definition at line 35 of file StatisticalModelChecker.cpp.

References acceptsMoreTraces().

### 7.182.3.17 void StatisticalModelChecker::updateDerivedModelCheckerForFalseEvaluation( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to false for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 53 of file StatisticalModelChecker.cpp.

7.182.3.18 void StatisticalModelChecker::updateDerivedModelCheckerForTrueEvaluation ( ) [override], [protected], [virtual]

Update the results of the derived model checker type considering that the logic property was evaluated to true for the last trace.

Do not do anything

Implements [multiscale::verification::ModelChecker](#).

Definition at line 51 of file StatisticalModelChecker.cpp.

7.182.3.19 void StatisticalModelChecker::updateInitialisedModelCheckingResult ( ) [private]

Update the result of the model checking task which was already initialised.

The name and semantics of the local variables a1, b1, a2, b2, f, fPrime, n, d correspond to the name and semantics of the variables used in the original paper.

Definition at line 98 of file StatisticalModelChecker.cpp.

References [computeFPrimeValue\(\)](#), [computeFValue\(\)](#), and [updateModelCheckingResult\(\)](#).

Referenced by [updateModelCheckingResult\(\)](#).

7.182.3.20 void StatisticalModelChecker::updateModelCheckingResult ( ) [private]

Update the result of the model checking task.

Definition at line 90 of file StatisticalModelChecker.cpp.

References [modelCheckingResult](#), [multiscale::verification::UNDECIDED](#), and [updateInitialisedModelCheckingResult\(\)](#).

Referenced by [acceptsMoreTraces\(\)](#), [doesPropertyHold\(\)](#), [getDetailedResults\(\)](#), and [updateInitialisedModelCheckingResult\(\)](#).

7.182.3.21 void StatisticalModelChecker::updateModelCheckingResult ( double *f*, double *fPrime* ) [private]

Update the result of the model checking task considering the given values.

#### Parameters

|               |                                                  |
|---------------|--------------------------------------------------|
| <i>f</i>      | The value of <i>f</i> (from the original paper)  |
| <i>fPrime</i> | The value of <i>f'</i> (from the original paper) |

Definition at line 105 of file StatisticalModelChecker.cpp.

References [a1FromPaper](#), [a2FromPaper](#), [b1FromPaper](#), [b2FromPaper](#), [updateModelCheckingResultEnoughTraces\(\)](#), and [updateModelCheckingResultNotEnoughTraces\(\)](#).

7.182.3.22 void StatisticalModelChecker::updateModelCheckingResultEnoughTraces ( double *f*, double *fPrime* ) [private]

Update the result of the model checking task considering the given values when enough traces have been provided.

#### Parameters

|          |                                                 |
|----------|-------------------------------------------------|
| <i>f</i> | The value of <i>f</i> (from the original paper) |
|----------|-------------------------------------------------|

|               |                                           |
|---------------|-------------------------------------------|
| <i>fPrime</i> | The value of f' (from the original paper) |
|---------------|-------------------------------------------|

Definition at line 118 of file StatisticalModelChecker.cpp.

References a1FromPaper, a2FromPaper, b1FromPaper, b2FromPaper, multiscale::verification::FALSE, indifference←IntervalHalf, modelCheckingResult, multiscale::verification::TRUE, and multiscale::verification::UNDECIDED.

Referenced by updateModelCheckingResult().

#### 7.182.3.23 void StatisticalModelChecker::updateModelCheckingResultNotEnoughTraces( ) [private]

Update the result of the model checking task when not enough traces were provided.

Definition at line 130 of file StatisticalModelChecker.cpp.

References modelCheckingResult, and multiscale::verification::MORE\_TRACES\_REQUIRED.

Referenced by updateModelCheckingResult().

#### 7.182.3.24 void StatisticalModelChecker::validateTypesErrors( double typeIError, double typeIIError ) [private]

Validate the probability of type I and type II errors to occur.

The probability of type I and type II errors to occur should be greater than zero and less than one

##### Parameters

|                    |                                             |
|--------------------|---------------------------------------------|
| <i>typeIError</i>  | The probability of a type I error to occur  |
| <i>typeIIError</i> | The probability of a type II error to occur |

Definition at line 55 of file StatisticalModelChecker.cpp.

References ERR\_TYPES\_ERROR\_VALUES\_BEGIN, ERR\_TYPES\_ERROR\_VALUES\_END, ERR\_TYPES\_ER←ROR\_VALUES\_MIDDLE, isValidTypeError(), MS\_throw, and multiscale::StringManipulator::toString().

Referenced by StatisticalModelChecker().

### 7.182.4 Member Data Documentation

#### 7.182.4.1 double multiscale::verification::StatisticalModelChecker::a1FromPaper [private]

The variable A1 (from the original paper)

Definition at line 64 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

#### 7.182.4.2 double multiscale::verification::StatisticalModelChecker::a2FromPaper [private]

The variable A2 (from the original paper)

Definition at line 66 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

#### 7.182.4.3 double multiscale::verification::StatisticalModelChecker::b1FromPaper [private]

The variable B1 (from the original paper)

Definition at line 65 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.182.4.4 double multiscale::verification::StatisticalModelChecker::b2FromPaper [private]

The variable B2 (from the original paper)

Definition at line 67 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.182.4.5 const std::string StatisticalModelChecker::ERR\_TYPES\_ERROR\_VALUES\_BEGIN = "The provided probabilities of type I and type II errors (" [static], [private]

Definition at line 215 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

7.182.4.6 const std::string StatisticalModelChecker::ERR\_TYPES\_ERROR\_VALUES\_END = ") should be greater than zero and less or equal to 1. Please change." [static], [private]

Definition at line 217 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

7.182.4.7 const std::string StatisticalModelChecker::ERR\_TYPES\_ERROR\_VALUES\_MIDDLE = "," [static], [private]

Definition at line 216 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

7.182.4.8 const std::string StatisticalModelChecker::ERR\_UNEXPECTED\_MODEL\_CHECKING\_RESULT = "An invalid statistical model checking result was obtained. Please check source code." [static], [private]

Definition at line 213 of file StatisticalModelChecker.hpp.

Referenced by doesPropertyHoldConsideringResult().

7.182.4.9 const unsigned int StatisticalModelChecker::INDIFFERENCE\_INTERVAL\_HALF\_K = (std::numeric\_limits<unsigned int>::max() >> 1) [static], [private]

The value of this constant should be much greater than 1.

Definition at line 227 of file StatisticalModelChecker.hpp.

Referenced by computeIndifferenceIntervalHalf().

7.182.4.10 double multiscale::verification::StatisticalModelChecker::indifferenceIntervalHalf [private]

Half of the size of the indifference interval

Definition at line 57 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(), computeFValueFirstTerm(), computeFValueSecondTerm(), initialise(), and updateModelCheckingResultEnoughTraces().

7.182.4.11 const double StatisticalModelChecker::LOGARITHM\_ZERO\_VALUE = (std::numeric\_limits<double>::lowest() / 1E+10) [static], [private]

The value of this constant should be a large negative number.

The value obtained when computing log(0)

This value will be further multiplied by non-negative integer numbers. In order to avoid overflow the lowest double value is divided by 1E10.

Definition at line 229 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(), computeFValueFirstTerm(), and computeFValueSecondTerm().

#### 7.182.4.12 double multiscale::verification::StatisticalModelChecker::minTypesError [private]

The minimum probability of type I and type II errors to occur

Definition at line 62 of file StatisticalModelChecker.hpp.

Referenced by initialise(), and StatisticalModelChecker().

#### 7.182.4.13 StatisticalModelCheckingResult multiscale::verification::StatisticalModelChecker::modelCheckingResult [private]

The result of the model checking task

Definition at line 69 of file StatisticalModelChecker.hpp.

Referenced by acceptsMoreTraces(), doesPropertyHoldConsideringResult(), getDetailedUpdatedResults(), updateModelCheckingResult(), updateModelCheckingResultEnoughTraces(), and updateModelCheckingResultNotEnoughTraces().

#### 7.182.4.14 const std::string StatisticalModelChecker::MSG\_OUTPUT\_MORE\_TRACES\_REQUIRED = "More traces are required to provide a true/false answer assuming the given probabilities of type I and type II errors. Probabilistic black-box model checking was used instead to provide an answer." [static], [private]

Definition at line 219 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

#### 7.182.4.15 const std::string StatisticalModelChecker::MSG\_OUTPUT\_RESULT\_BEGIN = "The provided answer is given for the probability of type I errors = " [static], [private]

Definition at line 221 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

#### 7.182.4.16 const std::string StatisticalModelChecker::MSG\_OUTPUT\_RESULT\_END = "" [static], [private]

Definition at line 223 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

#### 7.182.4.17 const std::string StatisticalModelChecker::MSG\_OUTPUT\_RESULT\_MIDDLE = " and the probability of type II errors = " [static], [private]

Definition at line 222 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.182.4.18 const std::string StatisticalModelChecker::MSG\_OUTPUT\_SEPARATOR = " " [static], [private]

Definition at line 225 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.182.4.19 double multiscale::verification::StatisticalModelChecker::probability [private]

The probability specified by the user for the logic property to be evaluated

Definition at line 54 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(), computeFValueFirstTerm(), computeFValueSecondTerm(), and initialise().

7.182.4.20 double multiscale::verification::StatisticalModelChecker::typeIError [private]

The probability of type I errors to occur

Definition at line 59 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults(), initialise(), and StatisticalModelChecker().

7.182.4.21 double multiscale::verification::StatisticalModelChecker::typeIIError [private]

The probability of type II errors to occur

Definition at line 60 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults(), initialise(), and StatisticalModelChecker().

The documentation for this class was generated from the following files:

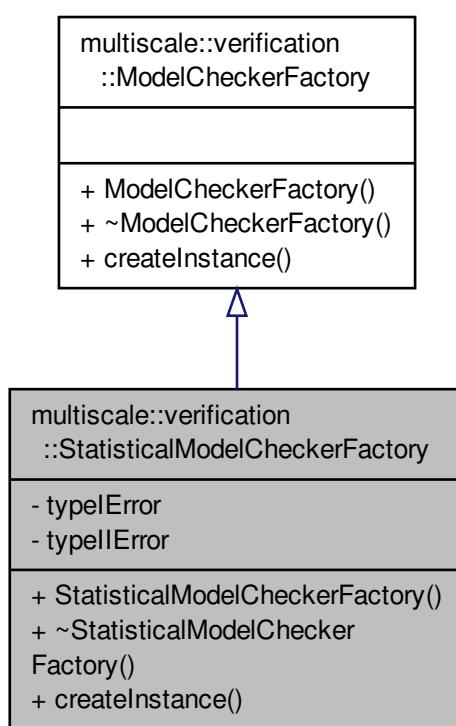
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[StatisticalModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[StatisticalModelChecker.cpp](#)

## 7.183 multiscale::verification::StatisticalModelCheckerFactory Class Reference

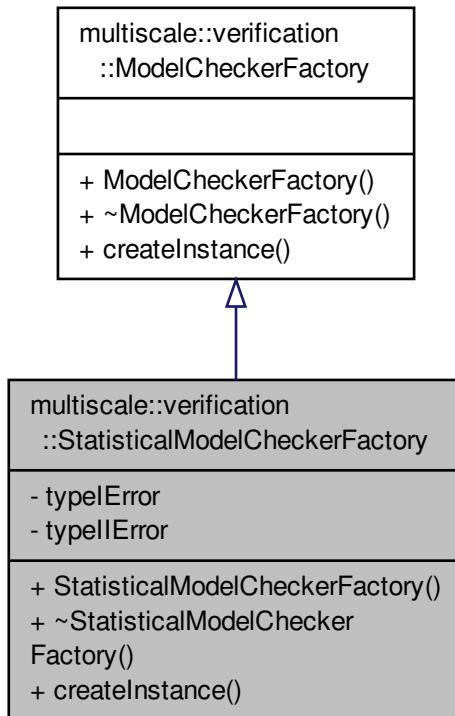
Class for creating [StatisticalModelChecker](#) instances.

```
#include <StatisticalModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::StatisticalModelCheckerFactory:



Collaboration diagram for multiscale::verification::StatisticalModelCheckerFactory:



## Public Member Functions

- `StatisticalModelCheckerFactory (double typeIError, double typeIIError)`
- `~StatisticalModelCheckerFactory ()`
- `std::shared_ptr< ModelChecker > createInstance (const AbstractSyntaxTree &abstractSyntaxTree, const TypeSemanticsTable &typeSemanticsTable) override`

*Create an instance of `StatisticalModelChecker`.*

## Private Attributes

- `double typeIError`
- `double typeIIError`

### 7.183.1 Detailed Description

Class for creating `StatisticalModelChecker` instances.

Definition at line 12 of file `StatisticalModelCheckerFactory.hpp`.

### 7.183.2 Constructor & Destructor Documentation

### 7.183.2.1 StatisticalModelCheckerFactory::StatisticalModelCheckerFactory ( double typeIError, double typeIIError )

Definition at line 7 of file StatisticalModelCheckerFactory.cpp.

### 7.183.2.2 StatisticalModelCheckerFactory::~StatisticalModelCheckerFactory ( )

Definition at line 12 of file StatisticalModelCheckerFactory.cpp.

## 7.183.3 Member Function Documentation

### 7.183.3.1 std::shared\_ptr< ModelChecker > StatisticalModelCheckerFactory::createInstance ( const AbstractSyntaxTree & abstractSyntaxTree, const TypeSemanticsTable & typeSemanticsTable ) [override], [virtual]

Create an instance of [StatisticalModelChecker](#).

#### Parameters

|                           |                                                                                       |
|---------------------------|---------------------------------------------------------------------------------------|
| <i>abstractSyntaxTree</i> | The abstract syntax tree representing the logic property to be checked                |
| <i>typeSemanticsTable</i> | The type semantics table mapping semantic criteria values to abstract natural numbers |

Implements [multiscale::verification::ModelCheckerFactory](#).

Definition at line 15 of file StatisticalModelCheckerFactory.cpp.

References typeIError, and typeIIError.

## 7.183.4 Member Data Documentation

### 7.183.4.1 double multiscale::verification::StatisticalModelCheckerFactory::typeIError [private]

The probability of a type I error

Definition at line 16 of file StatisticalModelCheckerFactory.hpp.

Referenced by [createInstance\(\)](#).

### 7.183.4.2 double multiscale::verification::StatisticalModelCheckerFactory::typeIIError [private]

The probability of a type II error

Definition at line 17 of file StatisticalModelCheckerFactory.hpp.

Referenced by [createInstance\(\)](#).

The documentation for this class was generated from the following files:

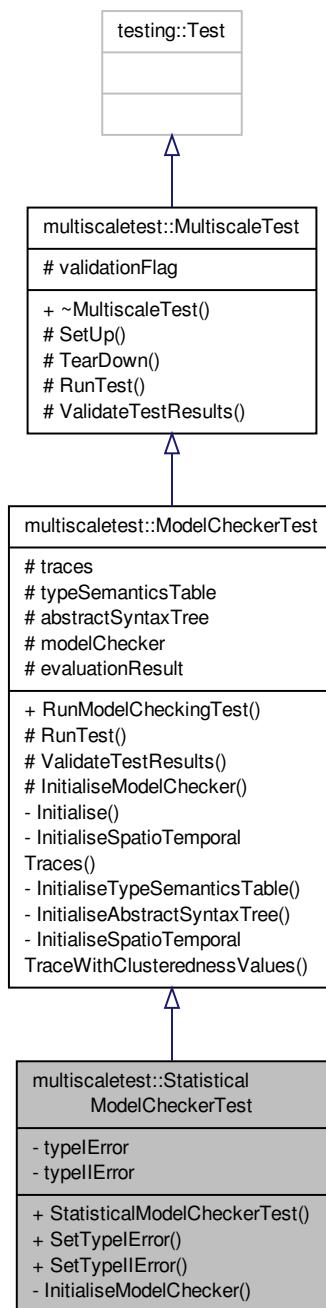
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[StatisticalModelCheckerFactory.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[StatisticalModelCheckerFactory.cpp](#)

## 7.184 multiscaletest::StatisticalModelCheckerTest Class Reference

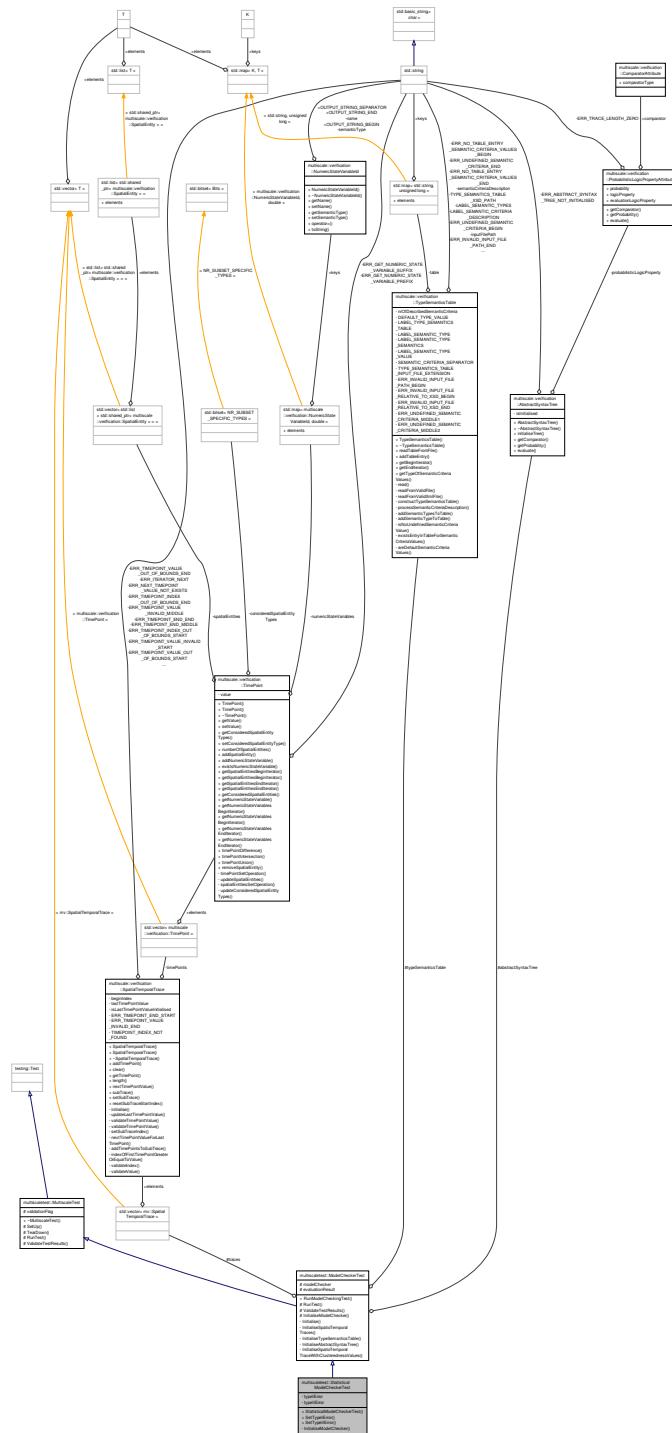
Class for testing the statistical model checker.

```
#include <StatisticalModelCheckerTest.hpp>
```

Inheritance diagram for multiscaletest::StatisticalModelCheckerTest:



## Collaboration diagram for multiscaletest::StatisticalModelCheckerTest:



## Public Member Functions

- `StatisticalModelCheckerTest ()`
  - `void SetTypeIError (double typeIError)`  
*Set the value of the type I error.*
  - `void SetTypeIIError (double typeIIError)`  
*Set the value of the type II error.*

*Set the value of the type I error.*

- void SetTypeIIError (double typeIIError)

*Set the value of the type II error.*

## Private Member Functions

- void [InitialiseModelChecker \(\) override](#)

*Initialise the model checker.*

## Private Attributes

- double [typeIError](#)
- double [typeIIError](#)

## Additional Inherited Members

### 7.184.1 Detailed Description

Class for testing the statistical model checker.

Definition at line 15 of file StatisticalModelCheckerTest.hpp.

### 7.184.2 Constructor & Destructor Documentation

#### 7.184.2.1 multiscaletest::StatisticalModelCheckerTest::StatisticalModelCheckerTest( ) [inline]

Definition at line 24 of file StatisticalModelCheckerTest.hpp.

### 7.184.3 Member Function Documentation

#### 7.184.3.1 void multiscaletest::StatisticalModelCheckerTest::InitialiseModelChecker( ) [override], [private], [virtual]

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 55 of file StatisticalModelCheckerTest.hpp.

#### 7.184.3.2 void multiscaletest::StatisticalModelCheckerTest::SetTypeIError( double typeIError )

Set the value of the type I error.

##### Parameters

|                         |                                            |
|-------------------------|--------------------------------------------|
| <code>typeIError</code> | The probability of type I errors occurring |
|-------------------------|--------------------------------------------|

Definition at line 47 of file StatisticalModelCheckerTest.hpp.

#### 7.184.3.3 void multiscaletest::StatisticalModelCheckerTest::SetTypeIIError( double typeIIError )

Set the value of the type II error.

##### Parameters

|                          |                                             |
|--------------------------|---------------------------------------------|
| <code>typeIIError</code> | The probability of type II errors occurring |
|--------------------------|---------------------------------------------|

Definition at line 51 of file StatisticalModelCheckerTest.hpp.

## 7.184.4 Member Data Documentation

7.184.4.1 double multiscaletest::StatisticalModelCheckerTest::typeIError [private]

The probability of type I errors

Definition at line 19 of file StatisticalModelCheckerTest.hpp.

7.184.4.2 double multiscaletest::StatisticalModelCheckerTest::typeIIError [private]

The probability of type II errors

Definition at line 20 of file StatisticalModelCheckerTest.hpp.

The documentation for this class was generated from the following file:

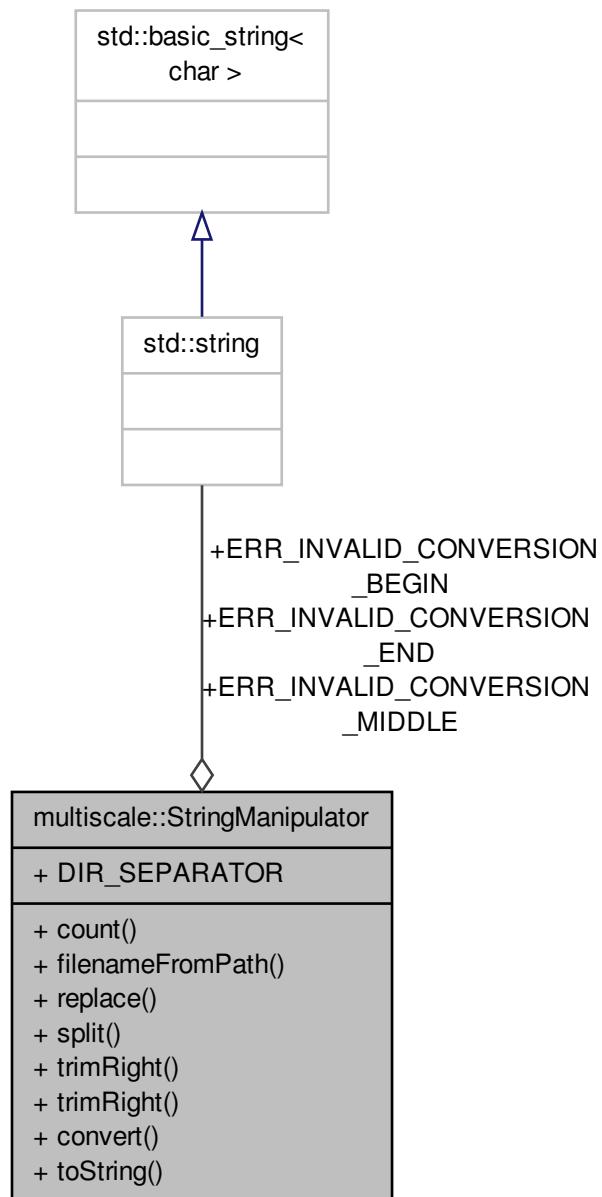
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/StatisticalModelCheckerTest.hpp

## 7.185 multiscale::StringManipulator Class Reference

Class for manipulating std::strings.

```
#include <StringManipulator.hpp>
```

Collaboration diagram for multiscale::StringManipulator:



## Static Public Member Functions

- static unsigned long `count` (char searchChar, const std::string &inputString)
 

*Count how many times character occurs in the given std::string.*
- static std::string `filenameFromPath` (const std::string &filepath)
 

*Obtain the file name from the given file path.*
- static std::string `replace` (const std::string &initialString, const std::string &replaceWhat, const std::string &replaceTo)
 

*Replace a substd::string of the given std::string with another std::string.*

- static std::vector< std::string > **split** (const std::string &initialString, const std::string &delimiter)
 

*Split the given std::string into a std::vector of std::strings considering the given delimiter.*
- static std::string **trimRight** (std::string &inputString)
 

*Remove the trailing "new line" characters from the end of the std::string.*
- static std::string **trimRight** (const std::string &inputString)
 

*Remove the trailing "new line" characters from the end of the std::string.*
- template<typename T>
 static T **convert** (const std::string &inputString)
 

*Convert the std::string to the given type.*
- template<typename T>
 static std::string **toString** (T variable)
 

*Convert the variable to a std::string.*

## Static Public Attributes

- static const char **DIR\_SEPARATOR** = '/'
- static const std::string **ERR\_INVALID\_CONVERSION\_BEGIN** = "The provided std::string ("
- static const std::string **ERR\_INVALID\_CONVERSION\_MIDDLE** = ") could not be converted to a "
- static const std::string **ERR\_INVALID\_CONVERSION\_END** = ". Please change."

### 7.185.1 Detailed Description

Class for manipulating std::strings.

Definition at line 15 of file StringManipulator.hpp.

### 7.185.2 Member Function Documentation

#### 7.185.2.1 template<typename T> static T multiscale::StringManipulator::convert ( const std::string & *inputString* ) [inline], [static]

Convert the std::string to the given type.

##### Parameters

|                    |                             |
|--------------------|-----------------------------|
| <i>inputString</i> | The given input std::string |
|--------------------|-----------------------------|

Definition at line 66 of file StringManipulator.hpp.

References **ERR\_INVALID\_CONVERSION\_BEGIN**, **ERR\_INVALID\_CONVERSION\_END**, **ERR\_INVALID\_CONVERSION\_MIDDLE**, and **MS\_throw**.

#### 7.185.2.2 unsigned long StringManipulator::count ( char *searchChar*, const std::string & *inputString* ) [static]

Count how many times character occurs in the given std::string.

##### Parameters

|                    |                                                                   |
|--------------------|-------------------------------------------------------------------|
| <i>searchChar</i>  | The search character of interest                                  |
| <i>inputString</i> | The input std::string in which the character will be searched for |

Definition at line 9 of file StringManipulator.cpp.

Referenced by **multiscale::verification::TypeSemanticsTable::isNoUndefinedSemanticCriteriaValue()**, and **multiscale::verification::TypeSemanticsTable::processSemanticCriteriaDescription()**.

7.185.2.3 `std::string StringManipulator::filenameFromPath ( const std::string & filepath ) [static]`

Obtain the file name from the given file path.

## Parameters

|                 |           |
|-----------------|-----------|
| <i>filepath</i> | File path |
|-----------------|-----------|

Definition at line 22 of file StringManipulator.cpp.

References DIR\_SEPARATOR.

Referenced by multiscale::video::PolarGnuplotScriptGenerator::generateHeader(), and multiscale::video::RectangularGnuplotScriptGenerator::generateHeader().

**7.185.2.4 std::string StringManipulator::replace ( const std::string & *initialString*, const std::string & *replaceWhat*, const std::string & *replaceTo* ) [static]**

Replace a substd::string of the given std::string with another std::string.

## Parameters

|                      |                                                                      |
|----------------------|----------------------------------------------------------------------|
| <i>initialString</i> | Initial std::string                                                  |
| <i>replaceWhat</i>   | Substd::string which will be replaced                                |
| <i>replaceTo</i>     | String which will be inserted instead of the replaceWhat std::string |

Definition at line 32 of file StringManipulator.cpp.

Referenced by multiscale::video::PolarGnuplotScriptGenerator::outputContent(), multiscale::video::PolarGnuplotScriptGenerator::outputHeader(), and multiscale::video::RectangularGnuplotScriptGenerator::outputHeader().

**7.185.2.5 std::vector< std::string > StringManipulator::split ( const std::string & *initialString*, const std::string & *delimiter* ) [static]**

Split the given std::string into a std::vector of std::strings considering the given delimiter.

## Parameters

|                      |                     |
|----------------------|---------------------|
| <i>initialString</i> | Initial std::string |
| <i>delimiter</i>     | Delimiter           |

Definition at line 45 of file StringManipulator.cpp.

Referenced by multiscale::verification::TemporalDataReader::readInputFileContents(), multiscale::verification::TemporalDataReader::readInputFileHeader(), multiscale::video::PolarCsvToInputFilesConverter::splitLineInConcentrations(), multiscale::video::RectangularCsvToInputFilesConverter::splitLineInConcentrations(), multiscale::video::RectangularEntityCsvToInputFilesConverter::splitLineInCoordinates(), multiscale::video::RectangularEntityCsvToInputFilesConverter::validateInputLine(), multiscale::video::PolarCsvToInputFilesConverter::validateInputLine(), and multiscale::video::RectangularCsvToInputFilesConverter::validateInputLine().

**7.185.2.6 template<typename T > static std::string multiscale::StringManipulator::toString ( T *variable* ) [inline], [static]**

Convert the variable to a std::string.

## Parameters

|                 |          |
|-----------------|----------|
| <i>variable</i> | Variable |
|-----------------|----------|

Definition at line 91 of file StringManipulator.hpp.

Referenced by multiscale::Numeric::combinations(), multiscale::XmlValidator::XmlValidationErrorHandler::constructExceptionMessage(), multiscale::verification::ApproximateProbabilisticModelChecker::getDetailedResults(), multiscale::verification::ModelChecker::getDetailedResultsUsingPValues(), multiscale::verification::ApproximateBayesianModelChecker::getDetailedUpdatedResults(), multiscale::verification::BayesianModelChecker::getDetailedUpdatedResults(), multiscale::verification::StatisticalModelChecker::getDetailedUpdatedResults(), multiscale::verification::CommandLineModelChecking::initialiseApproximateBayesianModelChecker(),

multiscale::verification::CommandLineModelChecking::initialiseApproximateProbabilisticModelChecker(), multiscale::verification::CommandLineModelChecking::initialiseBayesianModelChecker(), multiscale::verification::CommandLineModelChecking::initialiseStatisticalModelChecker(), multiscale::video::RectangularEntityCsvToInputFilesConverter::initOutputFile(), multiscale::video::PolarCsvToInputFilesConverter::initOutputFile(), multiscale::video::RectangularCsvToInputFilesConverter::initOutputFile(), multiscale::verification::ModelCheckingOutputWriter::printEvaluationResultsSummary(), multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage(), multiscale::verification::ModelCheckingOutputWriter::printTimeoutMessage(), multiscale::ConsolePrinter::unixColourCodeToString(), multiscale::verification::BayesianModelChecker::validateBayesFactorThreshold(), multiscale::verification::ApproximateProbabilisticModelChecker::validateInput(), multiscale::Numeric::validateLogBase(), multiscale::Numeric::validateLogNumber(), multiscale::BinomialDistribution::validateNrOfSuccesses(), multiscale::Distribution::validateProbability(), multiscale::BetaDistribution::validateShapeParameters(), multiscale::verification::BayesianModelChecker::validateShapeParameters(), multiscale::verification::ApproximateBayesianModelChecker::validateShapeParameters(), multiscale::verification::StatisticalModelChecker::validateTypesErrors(), and multiscale::verification::ApproximateBayesianModelChecker::validateVarianceThreshold().

#### 7.185.2.7 std::string StringManipulator::trimRight ( std::string & *inputString* ) [static]

Remove the trailing "new line" characters from the end of the std::string.

##### Parameters

|                    |                             |
|--------------------|-----------------------------|
| <i>inputString</i> | The given input std::string |
|--------------------|-----------------------------|

Definition at line 52 of file StringManipulator.cpp.

Referenced by multiscale::verification::ModelCheckingOutputWriter::printLogicPropertyWithTag(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage(), trimRight(), and multiscale::verification::ParserGrammarExceptionHandler::trimRight().

#### 7.185.2.8 std::string StringManipulator::trimRight ( const std::string & *inputString* ) [static]

Remove the trailing "new line" characters from the end of the std::string.

##### Parameters

|                    |                             |
|--------------------|-----------------------------|
| <i>inputString</i> | The given input std::string |
|--------------------|-----------------------------|

Definition at line 64 of file StringManipulator.cpp.

References trimRight().

### 7.185.3 Member Data Documentation

#### 7.185.3.1 const char StringManipulator::DIR\_SEPARATOR = '/' [static]

Definition at line 102 of file StringManipulator.hpp.

Referenced by filenameFromPath().

#### 7.185.3.2 const std::string StringManipulator::ERR\_INVALID\_CONVERSION\_BEGIN = "The provided std::string (" [static]

Definition at line 104 of file StringManipulator.hpp.

Referenced by convert().

#### 7.185.3.3 const std::string StringManipulator::ERR\_INVALID\_CONVERSION\_END = ". Please change." [static]

Definition at line 106 of file StringManipulator.hpp.

Referenced by convert().

7.185.3.4 `const std::string StringManipulator::ERR_INVALID_CONVERSION_MIDDLE = ") could not be converted to a "`  
`[static]`

Definition at line 105 of file StringManipulator.hpp.

Referenced by convert().

The documentation for this class was generated from the following files:

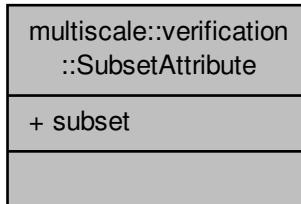
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/StringManipulator.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/StringManipulator.cpp

## 7.186 multiscale::verification::SubsetAttribute Class Reference

Class for representing a subset attribute.

```
#include <SubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SubsetAttribute:



### Public Attributes

- `SubsetAttributeType subset`

#### 7.186.1 Detailed Description

Class for representing a subset attribute.

Definition at line 29 of file SubsetAttribute.hpp.

#### 7.186.2 Member Data Documentation

##### 7.186.2.1 `SubsetAttributeType multiscale::verification::SubsetAttribute::subset`

The considered subset

Definition at line 33 of file SubsetAttribute.hpp.

Referenced by `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection()`, and `multiscale::verification::SubsetVisitor::operator()()`.

The documentation for this class was generated from the following file:

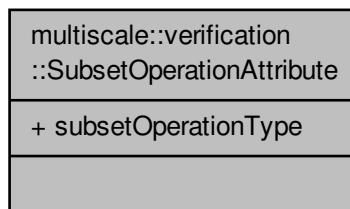
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetAttribute.hpp

## 7.187 multiscale::verification::SubsetOperationAttribute Class Reference

Class for representing a subset operation attribute.

```
#include <SubsetOperationAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SubsetOperationAttribute:



### Public Attributes

- [SubsetOperationType subsetOperationType](#)

#### 7.187.1 Detailed Description

Class for representing a subset operation attribute.

Definition at line 29 of file SubsetOperationAttribute.hpp.

#### 7.187.2 Member Data Documentation

##### 7.187.2.1 SubsetOperationType multiscale::verification::SubsetOperationAttribute::subsetOperationType

The subset operation type

Definition at line 33 of file SubsetOperationAttribute.hpp.

Referenced by multiscale::verification::SubsetVisitor::operator()().

The documentation for this class was generated from the following file:

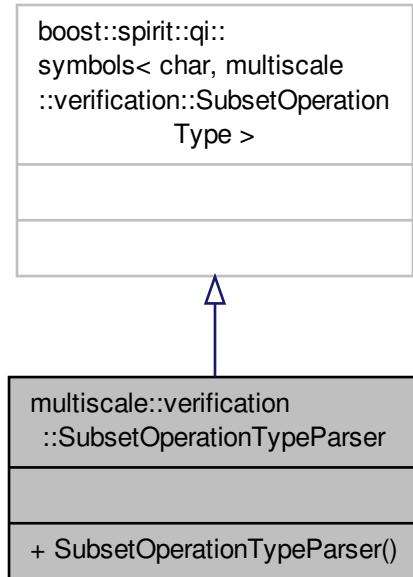
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp

## 7.188 multiscale::verification::SubsetOperationTypeParser Struct Reference

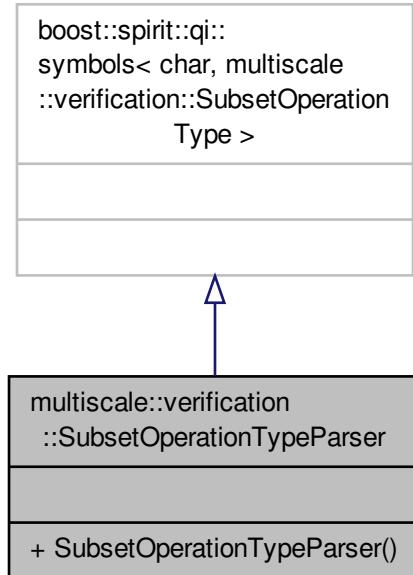
Symbol table and parser for the subset operation type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::SubsetOperationTypeParser:



Collaboration diagram for multiscale::verification::SubsetOperationTypeParser:



## Public Member Functions

- [SubsetOperationTypeParser \(\)](#)

### 7.188.1 Detailed Description

Symbol table and parser for the subset operation type.

Definition at line 169 of file [SymbolTables.hpp](#).

### 7.188.2 Constructor & Destructor Documentation

#### 7.188.2.1 multiscale::verification::SubsetOperationTypeParser::SubsetOperationTypeParser ( ) [inline]

Definition at line 172 of file [SymbolTables.hpp](#).

References [multiscale::verification::Difference](#), [multiscale::verification::Intersection](#), and [multiscale::verification::Union](#).

The documentation for this struct was generated from the following file:

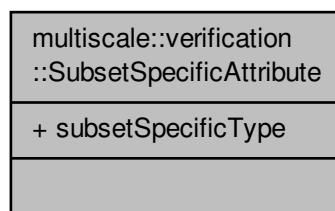
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.189 multiscale::verification::SubsetSpecificAttribute Class Reference

Class for representing a subset specific attribute.

```
#include <SubsetSpecificAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SubsetSpecificAttribute:



## Public Attributes

- [SubsetSpecificType subsetSpecificType](#)

### 7.189.1 Detailed Description

Class for representing a subset specific attribute.

Definition at line 55 of file [SubsetSpecificAttribute.hpp](#).

## 7.189.2 Member Data Documentation

### 7.189.2.1 SubsetSpecificType multiscale::verification::SubsetSpecificAttribute::subsetSpecificType

The specific subset type

Definition at line 59 of file `SubsetSpecificAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

The documentation for this class was generated from the following file:

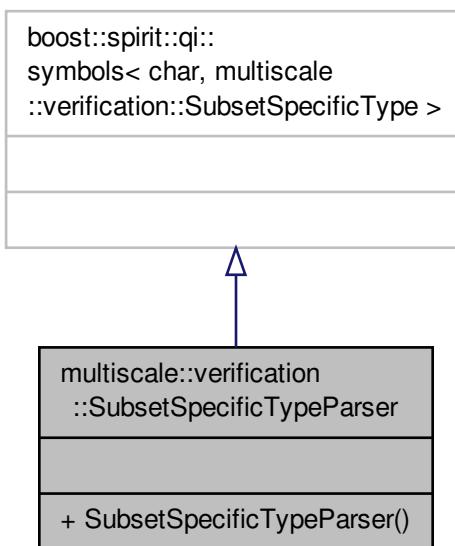
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[SubsetSpecificAttribute.hpp](#)

## 7.190 multiscale::verification::SubsetSpecificTypeParser Struct Reference

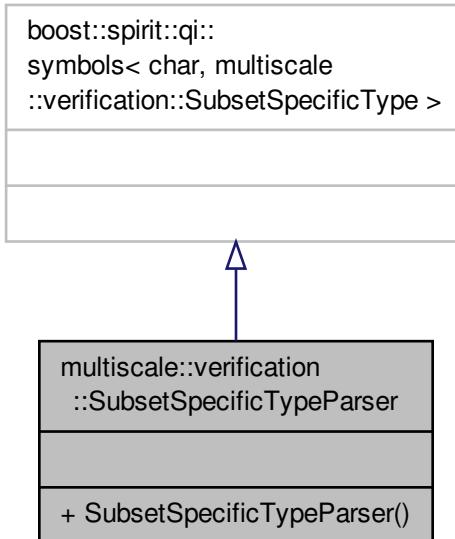
Symbol table and parser for a specific subset type.

```
#include <SymbolTablesAutoGenerated.hpp>
```

Inheritance diagram for multiscale::verification::SubsetSpecificTypeParser:



Collaboration diagram for multiscale::verification::SubsetSpecificTypeParser:



## Public Member Functions

- [SubsetSpecificTypeParser \(\)](#)

### 7.190.1 Detailed Description

Symbol table and parser for a specific subset type.

Definition at line 49 of file [SymbolTablesAutoGenerated.hpp](#).

### 7.190.2 Constructor & Destructor Documentation

#### 7.190.2.1 multiscale::verification::SubsetSpecificTypeParser::SubsetSpecificTypeParser( ) [inline]

Definition at line 51 of file [SymbolTablesAutoGenerated.hpp](#).

References multiscale::verification::Clusters, and multiscale::verification::Regions.

The documentation for this struct was generated from the following file:

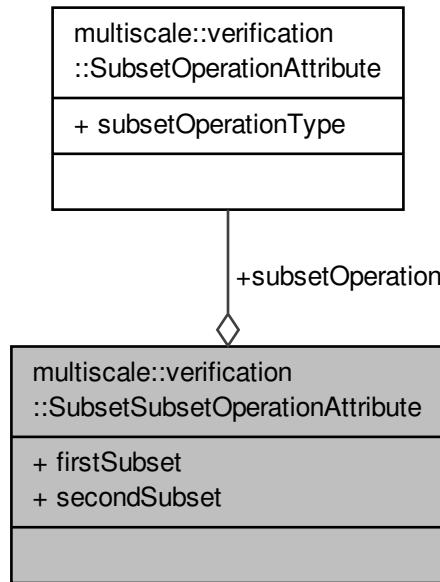
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTablesAutoGenerated.hpp](#)

## 7.191 multiscale::verification::SubsetSubsetOperationAttribute Class Reference

Class for representing a subset subset operation attribute.

```
#include <SubsetSubsetOperationAttribute.hpp>
```

Collaboration diagram for multiscale::verification::SubsetSubsetOperationAttribute:



## Public Attributes

- [SubsetOperationAttribute subsetOperation](#)
- [SubsetAttributeType firstSubset](#)
- [SubsetAttributeType secondSubset](#)

### 7.191.1 Detailed Description

Class for representing a subset subset operation attribute.

Definition at line 15 of file `SubsetSubsetOperationAttribute.hpp`.

### 7.191.2 Member Data Documentation

#### 7.191.2.1 `SubsetAttributeType multiscale::verification::SubsetSubsetOperationAttribute::firstSubset`

The first considered subset

Definition at line 20 of file `SubsetSubsetOperationAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

#### 7.191.2.2 `SubsetAttributeType multiscale::verification::SubsetSubsetOperationAttribute::secondSubset`

The second considered subset

Definition at line 21 of file `SubsetSubsetOperationAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

### 7.191.2.3 SubsetOperationAttribute multiscale::verification::SubsetSubsetOperationAttribute::subsetOperation

The employed subset operation

Definition at line 19 of file `SubsetSubsetOperationAttribute.hpp`.

Referenced by `multiscale::verification::SubsetVisitor::operator()()`.

The documentation for this class was generated from the following file:

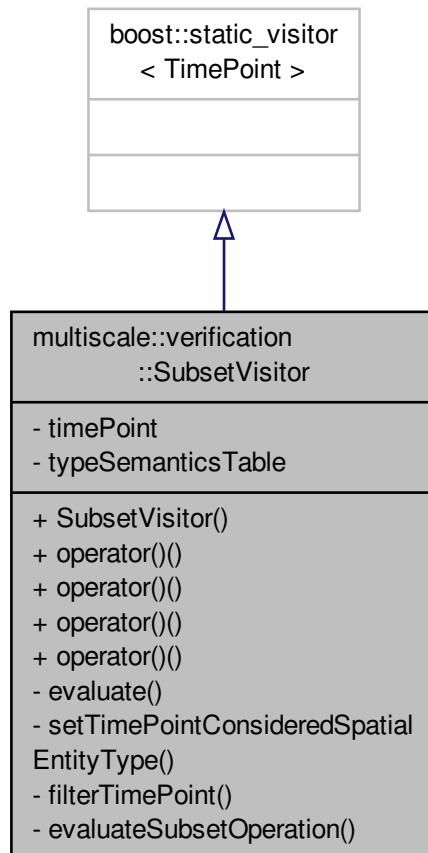
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSubsetOperationAttribute.hpp](#)

## 7.192 multiscale::verification::SubsetVisitor Class Reference

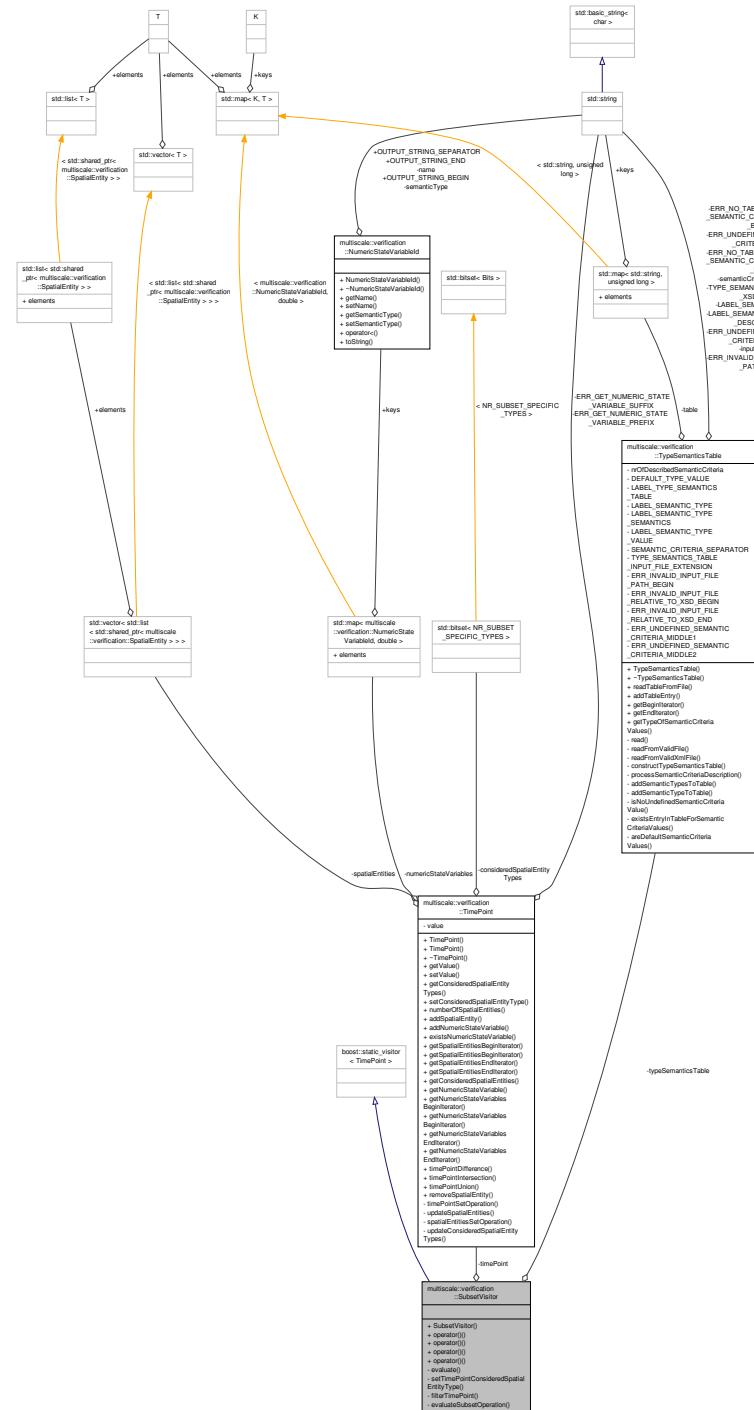
Class used to evaluate subsets.

```
#include <SubsetVisitor.hpp>
```

Inheritance diagram for `multiscale::verification::SubsetVisitor`:



## Collaboration diagram for multiscale::verification::SubsetVisitor:



## Public Member Functions

- `SubsetVisitor` (`const TimePoint &timePoint, const TypeSemanticsTable &typeSemanticsTable`)
  - `TimePoint operator()` (`const SubsetAttribute &subset`) const
    - Overloading the "()" operator for the `SubsetAttribute` alternative.*
  - `TimePoint operator()` (`const SubsetSpecificAttribute &subset`) const
    - Overloading the "()" operator for the `SubsetSpecificAttribute` alternative.*

- **TimePoint operator()** (const `FilterSubsetAttribute` &subset) const  
*Overloading the "()" operator for the `FilterSubsetAttribute` alternative.*
- **TimePoint operator()** (const `SubsetSubsetOperationAttribute` &subset) const  
*Overloading the "()" operator for the `SubsetSubsetOperationAttribute` alternative.*

## Private Member Functions

- **TimePoint evaluate** (const `SubsetAttributeType` &subset, const `TimePoint` &`timePoint`) const  
*Evaluate the subset considering the given timepoint.*
- **void setTimePointConsideredSpatialEntityType** (`TimePoint` &`timePoint`, const `SubsetSpecificType` &`subsetType`) const  
*Set the considered spatial entity type for the given timepoint using the specific subset type.*
- **TimePoint filterTimePoint** (const `TimePoint` &`timePoint`, const `ConstraintAttributeType` &`constraint`) const  
*Filter the given timepoint considering the provided constraint.*
- **TimePoint evaluateSubsetOperation** (const `SubsetOperationType` &`subsetOperation`, const `TimePoint` &`firstSubsetTimePoint`, const `TimePoint` &`secondSubsetTimePoint`) const  
*Evaluate subsetOperation against the given subsets timepoints.*

## Private Attributes

- const `TimePoint` & `timePoint`
- const `TypeSemanticsTable` & `typeSemanticsTable`

### 7.192.1 Detailed Description

Class used to evaluate subsets.

Definition at line 14 of file `SubsetVisitor.hpp`.

### 7.192.2 Constructor & Destructor Documentation

**7.192.2.1 multiscale::verification::SubsetVisitor::SubsetVisitor** ( const `TimePoint` & `timePoint`, const `TypeSemanticsTable` & `typeSemanticsTable` ) [inline]

Definition at line 23 of file `SubsetVisitor.hpp`.

Referenced by `evaluate()`.

### 7.192.3 Member Function Documentation

**7.192.3.1 TimePoint multiscale::verification::SubsetVisitor::evaluate** ( const `SubsetAttributeType` & `subset`, const `TimePoint` & `timePoint` ) const [inline], [private]

Evaluate the subset considering the given timepoint.

#### Parameters

|                        |                     |
|------------------------|---------------------|
| <code>subset</code>    | The subset          |
| <code>timePoint</code> | The given timepoint |

Definition at line 79 of file `SubsetVisitor.hpp`.

References `SubsetVisitor()`.

Referenced by `operator()()`.

7.192.3.2 **TimePoint multiscale::verification::SubsetVisitor::evaluateSubsetOperation ( const SubsetOperationType & *subsetOperation*, const TimePoint & *firstSubsetTimePoint*, const TimePoint & *secondSubsetTimePoint* ) const [inline], [private]**

Evaluate subsetOperation against the given subsets timepoints.

## Parameters

|                                           |                                                  |
|-------------------------------------------|--------------------------------------------------|
| <i>subsetOperation</i>                    | The considered subset operation                  |
| <i>firstSubset</i> ↵<br><i>TimePoint</i>  | The timepoint corresponding to the first subset  |
| <i>secondSubset</i> ↵<br><i>TimePoint</i> | The timepoint corresponding to the second subset |

Definition at line 108 of file SubsetVisitor.hpp.

References multiscale::verification::Difference, multiscale::ERR\_UNDEFINED\_ENUM\_VALUE, multiscale::verification::Intersection, MS\_throw, multiscale::verification::TimePoint::timePointDifference(), multiscale::verification::TimePoint::timePointIntersection(), multiscale::verification::TimePoint::timePointUnion(), and multiscale::verification::Union.

Referenced by operator()().

### 7.192.3.3 multiscale::verification::TimePoint multiscale::verification::SubsetVisitor::filterTimePoint ( const TimePoint & *timePoint*, const ConstraintAttributeType & *constraint* ) const [inline], [private]

Filter the given timepoint considering the provided constraint.

## Parameters

|                   |                         |
|-------------------|-------------------------|
| <i>timePoint</i>  | The given timepoint     |
| <i>constraint</i> | The provided constraint |

Definition at line 148 of file SubsetVisitor.hpp.

References typeSemanticsTable.

Referenced by operator()().

### 7.192.3.4 TimePoint multiscale::verification::SubsetVisitor::operator() ( const SubsetAttribute & *subset* ) const [inline]

Overloading the "(" operator for the [SubsetAttribute](#) alternative.

## Parameters

|               |            |
|---------------|------------|
| <i>subset</i> | The subset |
|---------------|------------|

Definition at line 30 of file SubsetVisitor.hpp.

References evaluate(), and multiscale::verification::SubsetAttribute::subset.

### 7.192.3.5 TimePoint multiscale::verification::SubsetVisitor::operator() ( const SubsetSpecificAttribute & *subset* ) const [inline]

Overloading the "(" operator for the [SubsetSpecificAttribute](#) alternative.

## Parameters

|               |                     |
|---------------|---------------------|
| <i>subset</i> | The specific subset |
|---------------|---------------------|

Definition at line 38 of file SubsetVisitor.hpp.

References setTimePointConsideredSpatialEntityType(), and multiscale::verification::SubsetSpecificAttribute::subsetSpecificType.

7.192.3.6 **TimePoint multiscale::verification::SubsetVisitor::operator()** ( **const FilterSubsetAttribute & subset** ) const  
[inline]

Overloading the "(" operator for the [FilterSubsetAttribute](#) alternative.

**Parameters**

|               |                   |
|---------------|-------------------|
| <i>subset</i> | The filter subset |
|---------------|-------------------|

Definition at line 50 of file SubsetVisitor.hpp.

References multiscale::verification::FilterSubsetAttribute::constraint, filterTimePoint(), setTimePointConsideredSpatialEntityType(), multiscale::verification::FilterSubsetAttribute::subsetSpecific, and multiscale::verification::SubsetSpecificAttribute::subsetSpecificType.

#### 7.192.3.7 TimePoint multiscale::verification::SubsetVisitor::operator() ( const SubsetSubsetOperationAttribute & *subset* ) const [inline]

Overloading the "()" operator for the [SubsetSubsetOperationAttribute](#) alternative.

**Parameters**

|               |                                       |
|---------------|---------------------------------------|
| <i>subset</i> | The subset subset operation attribute |
|---------------|---------------------------------------|

Definition at line 63 of file SubsetVisitor.hpp.

References evaluate(), evaluateSubsetOperation(), multiscale::verification::SubsetSubsetOperationAttribute::firstSubset, multiscale::verification::SubsetSubsetOperationAttribute::secondSubset, multiscale::verification::SubsetSubsetOperationAttribute::subsetOperation, and multiscale::verification::SubsetOperationAttribute::subsetOperationType.

#### 7.192.3.8 void multiscale::verification::SubsetVisitor::setTimePointConsideredSpatialEntityType ( TimePoint & *timePoint*, const SubsetSpecificType & *subsetType* ) const [inline], [private]

Set the considered spatial entity type for the given timepoint using the specific subset type.

**Parameters**

|                   |                          |
|-------------------|--------------------------|
| <i>timePoint</i>  | The given timepoint      |
| <i>subsetType</i> | The specific subset type |

Definition at line 89 of file SubsetVisitor.hpp.

References multiscale::verification::TimePoint::setConsideredSpatialEntityType().

Referenced by operator()().

### 7.192.4 Member Data Documentation

#### 7.192.4.1 const TimePoint& multiscale::verification::SubsetVisitor::timePoint [private]

The initial timepoint

Definition at line 18 of file SubsetVisitor.hpp.

#### 7.192.4.2 const TypeSemanticsTable& multiscale::verification::SubsetVisitor::typeSemanticsTable [private]

The considered type semantics table

Definition at line 19 of file SubsetVisitor.hpp.

Referenced by filterTimePoint().

The documentation for this class was generated from the following file:

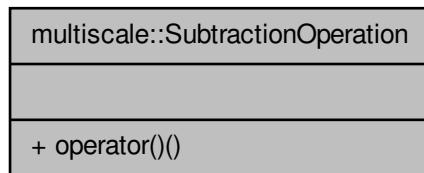
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SubsetVisitor.hpp

## 7.193 multiscale::SubtractionOperation Class Reference

Functor representing a subtraction operation.

```
#include <Numeric.hpp>
```

Collaboration diagram for multiscale::SubtractionOperation:



### Public Member Functions

- template<typename Operand >  
Operand [operator\(\)](#) (Operand operand1, Operand operand2) const  
*Subtract the two operands.*

#### 7.193.1 Detailed Description

Functor representing a subtraction operation.

Definition at line 67 of file Numeric.hpp.

#### 7.193.2 Member Function Documentation

7.193.2.1 template<typename Operand > Operand multiscale::SubtractionOperation::operator() ( Operand *operand1*, Operand *operand2* ) const [inline]

Subtract the two operands.

##### Parameters

|                 |                    |
|-----------------|--------------------|
| <i>operand1</i> | The first operand  |
| <i>operand2</i> | The second operand |

Definition at line 77 of file Numeric.hpp.

The documentation for this class was generated from the following file:

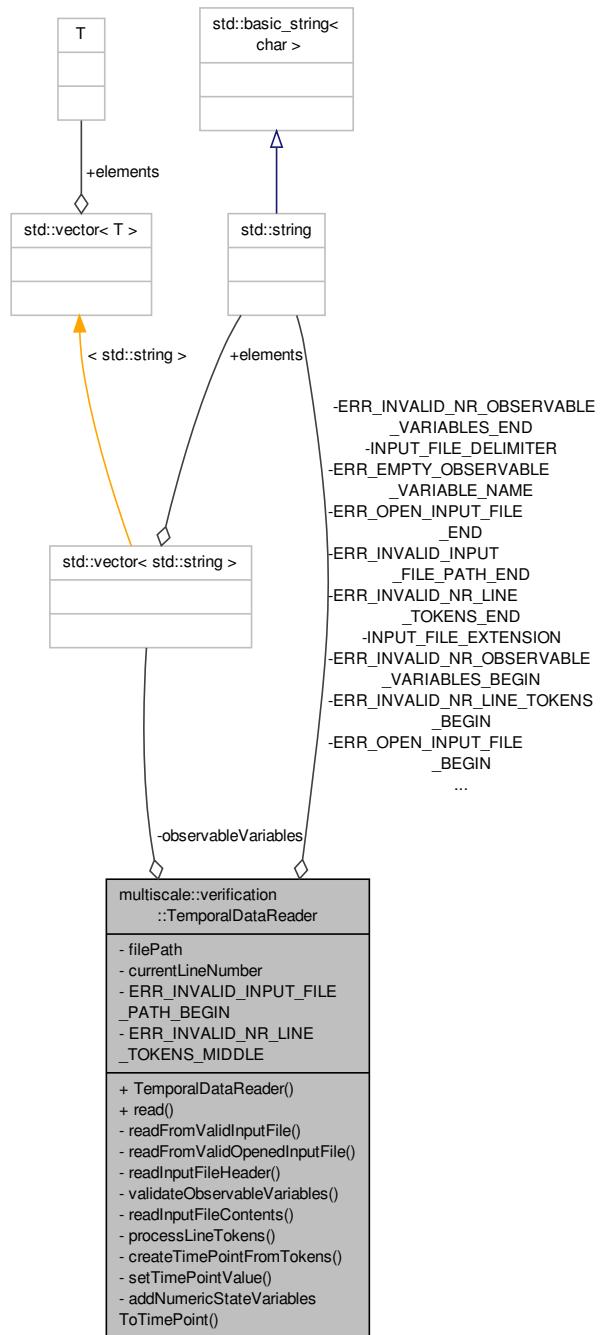
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[Numeric.hpp](#)

## 7.194 multiscale::verification::TemporalDataReader Class Reference

Class for reading (non-spatial) timeseries data from a .csv file.

```
#include <TemporalDataReader.hpp>
```

## Collaboration diagram for multiscale::verification::TemporalDataReader:



## Public Member Functions

- `TemporalDataReader ()`
  - `SpatialTemporalTrace read (const std::string &filePath)`

*Read the data from the input file and use it to construct a spatial temporal trace.*

## Private Member Functions

- `SpatialTemporalTrace readFromValidInputFile ()`  
*Read the data from the valid input file and construct a spatial temporal trace.*
- `void readFromValidOpenedInputFile (std::ifstream &fin, SpatialTemporalTrace &trace)`  
*Read the data from the valid opened input file and construct a spatial temporal trace.*
- `void readInputFileHeader (std::ifstream &fin, SpatialTemporalTrace &trace)`  
*Read the header row from the input file.*
- `void validateObservableVariables ()`  
*Validate the observable variables.*
- `void readInputFileContents (std::ifstream &fin, SpatialTemporalTrace &trace)`  
*Read the contents (excluding header row) from the input file.*
- `void processLineTokens (const std::vector< std::string > &lineTokens, SpatialTemporalTrace &trace)`  
*Check if the provided line tokens are valid and, if yes, add them to the trace.*
- `void createTimePointFromTokens (const std::vector< std::string > &lineTokens, SpatialTemporalTrace &trace)`  
*Create a new timepoint in the trace from the given tokens.*
- `void setTimePointValue (const std::vector< std::string > &lineTokens, TimePoint &timePoint)`  
*Set the value of the given timepoint considering the first token.*
- `void addNumericStateVariablesToTimePoint (const std::vector< std::string > &lineTokens, TimePoint &timePoint)`  
*Add the numeric state variable values to the timepoint.*

## Private Attributes

- `std::string filePath`
- `std::vector< std::string > observableVariables`
- `unsigned long currentLineNumber`

## Static Private Attributes

- `static const std::string INPUT_FILE_EXTENSION = ".csv"`
- `static const std::string INPUT_FILE_DELIMITER = ","`
- `static const std::string ERR_INVALID_INPUT_FILE_PATH_BEGIN = "The provided input file path ("`
- `static const std::string ERR_INVALID_INPUT_FILE_PATH_END = ") does not point to a file with the required extension. Please change."`
- `static const std::string ERR_OPEN_INPUT_FILE_BEGIN = "The provided input file ("`
- `static const std::string ERR_OPEN_INPUT_FILE_END = ") could not be opened. Please make sure it is available and not currently used by another process."`
- `static const std::string ERR_INVALID_NR_OBSERVABLE_VARIABLES_BEGIN = "The number of observable variables ("`
- `static const std::string ERR_INVALID_NR_OBSERVABLE_VARIABLES_END = ") should be greater or equal to two. Please change."`
- `static const std::string ERR_EMPTY_OBSERVABLE_VARIABLE_NAME = "The name of one of the observable variables is empty when it should contain at least one character. Please change."`
- `static const std::string ERR_INVALID_NR_LINE_TOKENS_BEGIN = "The number of tokens on line "`
- `static const std::string ERR_INVALID_NR_LINE_TOKENS_MIDDLE = " of input file "`
- `static const std::string ERR_INVALID_NR_LINE_TOKENS_END = " is different from the number of observable variables in the header. Please change."`

### 7.194.1 Detailed Description

Class for reading (non-spatial) timeseries data from a .csv file.

The format of the .csv input files is: Time, Observable 1, Observable 2, ..., Observable n T1, O11, O21, ..., On1 T2, O12, O22, ..., On2 ... Tm, O1m, O2m, ..., Onm where the first line contains the name of the observable variables (e.g. species) and the subsequent lines the values of these variables for a given timepoint (Ti, 1 <= i <= m)

Definition at line 26 of file TemporalDataReader.hpp.

### 7.194.2 Constructor & Destructor Documentation

#### 7.194.2.1 TemporalDataReader::TemporalDataReader( )

Definition at line 11 of file TemporalDataReader.cpp.

### 7.194.3 Member Function Documentation

#### 7.194.3.1 void TemporalDataReader::addNumericStateVariablesToTimePoint( const std::vector< std::string > & *lineTokens*, TimePoint & *timePoint* ) [private]

Add the numeric state variable values to the timepoint.

##### Parameters

|                   |                        |
|-------------------|------------------------|
| <i>lineTokens</i> | The given line tokens  |
| <i>timePoint</i>  | The provided timepoint |

Definition at line 146 of file TemporalDataReader.cpp.

References multiscale::verification::TimePoint::addNumericStateVariable(), and observableVariables.

Referenced by createTimePointFromTokens().

#### 7.194.3.2 void TemporalDataReader::createTimePointFromTokens( const std::vector< std::string > & *lineTokens*, SpatialTemporalTrace & *trace* ) [private]

Create a new timepoint in the trace from the given tokens.

##### Parameters

|                   |                                                                       |
|-------------------|-----------------------------------------------------------------------|
| <i>lineTokens</i> | The given line tokens                                                 |
| <i>trace</i>      | The spatial temporal trace created using the data from the input file |

Definition at line 129 of file TemporalDataReader.cpp.

References addNumericStateVariablesToTimePoint(), multiscale::verification::SpatialTemporalTrace::addTimePoint(), and setTimePointValue().

Referenced by processLineTokens().

#### 7.194.3.3 void TemporalDataReader::processLineTokens( const std::vector< std::string > & *lineTokens*, SpatialTemporalTrace & *trace* ) [private]

Check if the provided line tokens are valid and, if yes, add them to the trace.

**Parameters**

|                   |                                                                       |
|-------------------|-----------------------------------------------------------------------|
| <i>lineTokens</i> | The given line tokens                                                 |
| <i>trace</i>      | The spatial temporal trace created using the data from the input file |

Definition at line 113 of file TemporalDataReader.cpp.

References createTimePointFromTokens(), currentLineNumber, ERR\_INVALID\_NR\_LINE\_TOKENS\_BEGIN, E←RR\_INVALID\_NR\_LINE\_TOKENS\_END, ERR\_INVALID\_NR\_LINE\_TOKENS\_MIDDLE, filePath, MS\_throw, and observableVariables.

Referenced by readInputFileContents().

**7.194.3.4 SpatialTemporalTrace TemporalDataReader::read ( const std::string & filePath )**

Read the data from the input file and use it to construct a spatial temporal trace.

**Parameters**

|                 |                     |
|-----------------|---------------------|
| <i>filePath</i> | The input file path |
|-----------------|---------------------|

Definition at line 13 of file TemporalDataReader.cpp.

References currentLineNumber, ERR\_INVALID\_INPUT\_FILE\_PATH\_BEGIN, ERR\_INVALID\_INPUT\_FILE\_PA←TH\_END, filePath, INPUT\_FILE\_EXTENSION, multiscale::Filesystem::isValidFilePath(), MS\_throw, and readFromValidInputFile().

Referenced by analysePatterns(), and readAndPrintInputFile().

**7.194.3.5 SpatialTemporalTrace TemporalDataReader::readFromValidInputFile ( ) [private]**

Read the data from the valid input file and construct a spatial temporal trace.

Definition at line 32 of file TemporalDataReader.cpp.

References filePath, and readFromValidOpenedInputFile().

Referenced by read().

**7.194.3.6 void TemporalDataReader::readFromValidOpenedInputFile ( std::ifstream & fin, SpatialTemporalTrace & trace ) [private]**

Read the data from the valid opened input file and construct a spatial temporal trace.

**Parameters**

|              |                                                                       |
|--------------|-----------------------------------------------------------------------|
| <i>fin</i>   | The input file stream opened for the given input file                 |
| <i>trace</i> | The spatial temporal trace created using the data from the input file |

Definition at line 44 of file TemporalDataReader.cpp.

References ERR\_OPEN\_INPUT\_FILE\_BEGIN, ERR\_OPEN\_INPUT\_FILE\_END, filePath, MS\_throw, readInputFileContents(), and readInputFileHeader().

Referenced by readFromValidInputFile().

**7.194.3.7 void TemporalDataReader::readInputFileContents ( std::ifstream & fin, SpatialTemporalTrace & trace ) [private]**

Read the contents (excluding header row) from the input file.

**Parameters**

|              |                                                                       |
|--------------|-----------------------------------------------------------------------|
| <i>fin</i>   | The input file stream opened for the given input file                 |
| <i>trace</i> | The spatial temporal trace created using the data from the input file |

Definition at line 94 of file TemporalDataReader.cpp.

References `currentLineNumber`, `INPUT_FILE_DELIMITER`, `processLineTokens()`, and `multiscale::StringManipulator::split()`.

Referenced by `readFromValidOpenedInputFile()`.

**7.194.3.8 void TemporalDataReader::readInputFileHeader ( std::ifstream & *fin*, SpatialTemporalTrace & *trace* ) [private]**

Read the header row from the input file.

**Parameters**

|              |                                                                       |
|--------------|-----------------------------------------------------------------------|
| <i>fin</i>   | The input file stream opened for the given input file                 |
| <i>trace</i> | The spatial temporal trace created using the data from the input file |

Definition at line 58 of file TemporalDataReader.cpp.

References `currentLineNumber`, `INPUT_FILE_DELIMITER`, `observableVariables`, `multiscale::StringManipulator::split()`, and `validateObservableVariables()`.

Referenced by `readFromValidOpenedInputFile()`.

**7.194.3.9 void TemporalDataReader::setTimePointValue ( const std::vector< std::string > & *lineTokens*, TimePoint & *timePoint* ) [private]**

Set the value of the given timepoint considering the first token.

**Parameters**

|                   |                        |
|-------------------|------------------------|
| <i>lineTokens</i> | The given line tokens  |
| <i>timePoint</i>  | The provided timepoint |

Definition at line 139 of file TemporalDataReader.cpp.

References `multiscale::verification::TimePoint::setValue()`.

Referenced by `createTimePointFromTokens()`.

**7.194.3.10 void TemporalDataReader::validateObservableVariables ( ) [private]**

Validate the observable variables.

The observable variables collection is valid if it contains two or more elements.

Definition at line 72 of file TemporalDataReader.cpp.

References `ERR_EMPTY_OBSERVABLE_VARIABLE_NAME`, `ERR_INVALID_NR_OBSERVABLE_VARIABLES_BEGIN`, `ERR_INVALID_NR_OBSERVABLE_VARIABLES_END`, `MS_throw`, and `observableVariables`.

Referenced by `readInputFileHeader()`.

## 7.194.4 Member Data Documentation

**7.194.4.1 unsigned long multiscale::verification::TemporalDataReader::currentLineNumber [private]**

The current input file line number

Definition at line 34 of file TemporalDataReader.hpp.

Referenced by processLineTokens(), read(), readInputFileContents(), and readInputFileHeader().

7.194.4.2 `const std::string TemporalDataReader::ERR_EMPTY_OBSERVABLE_VARIABLE_NAME = "The name of one of the observable variables is empty when it should contain at least one character. Please change." [static], [private]`

Definition at line 124 of file TemporalDataReader.hpp.

Referenced by validateObservableVariables().

7.194.4.3 `const std::string TemporalDataReader::ERR_INVALID_INPUT_FILE_PATH_BEGIN = "The provided input file path (" [static], [private]`

Definition at line 115 of file TemporalDataReader.hpp.

Referenced by read().

7.194.4.4 `const std::string TemporalDataReader::ERR_INVALID_INPUT_FILE_PATH_END = ") does not point to a file with the required extension. Please change." [static], [private]`

Definition at line 116 of file TemporalDataReader.hpp.

Referenced by read().

7.194.4.5 `const std::string TemporalDataReader::ERR_INVALID_NR_LINE_TOKENS_BEGIN = "The number of tokens on line " [static], [private]`

Definition at line 126 of file TemporalDataReader.hpp.

Referenced by processLineTokens().

7.194.4.6 `const std::string TemporalDataReader::ERR_INVALID_NR_LINE_TOKENS_END = " is different from the number of observable variables in the header. Please change." [static], [private]`

Definition at line 128 of file TemporalDataReader.hpp.

Referenced by processLineTokens().

7.194.4.7 `const std::string TemporalDataReader::ERR_INVALID_NR_LINE_TOKENS_MIDDLE = " of input file " [static], [private]`

Definition at line 127 of file TemporalDataReader.hpp.

Referenced by processLineTokens().

7.194.4.8 `const std::string TemporalDataReader::ERR_INVALID_NR_OBSERVABLE_VARIABLES_BEGIN = "The number of observable variables (" [static], [private]`

Definition at line 121 of file TemporalDataReader.hpp.

Referenced by validateObservableVariables().

7.194.4.9 `const std::string TemporalDataReader::ERR_INVALID_NR_OBSERVABLE_VARIABLES_END = ") should be greater or equal to two. Please change." [static], [private]`

Definition at line 122 of file TemporalDataReader.hpp.

Referenced by validateObservableVariables().

7.194.4.10 `const std::string TemporalDataReader::ERR_OPEN_INPUT_FILE_BEGIN = "The provided input file (" [static], [private]`

Definition at line 118 of file TemporalDataReader.hpp.

Referenced by readFromValidOpenedInputFile().

7.194.4.11 `const std::string TemporalDataReader::ERR_OPEN_INPUT_FILE_END = ") could not be opened. Please make sure it is available and not currently used by another process." [static], [private]`

Definition at line 119 of file TemporalDataReader.hpp.

Referenced by readFromValidOpenedInputFile().

7.194.4.12 `std::string multiscale::verification::TemporalDataReader::filePath [private]`

The path to the input file

Definition at line 30 of file TemporalDataReader.hpp.

Referenced by processLineTokens(), read(), readFromValidInputFile(), and readFromValidOpenedInputFile().

7.194.4.13 `const std::string TemporalDataReader::INPUT_FILE_DELIMITER = "," [static], [private]`

Definition at line 113 of file TemporalDataReader.hpp.

Referenced by readInputFileContents(), and readInputFileHeader().

7.194.4.14 `const std::string TemporalDataReader::INPUT_FILE_EXTENSION = ".csv" [static], [private]`

Definition at line 111 of file TemporalDataReader.hpp.

Referenced by read().

7.194.4.15 `std::vector<std::string> multiscale::verification::TemporalDataReader::observableVariables [private]`

The names of the observable variables

Definition at line 32 of file TemporalDataReader.hpp.

Referenced by addNumericStateVariablesToTimePoint(), processLineTokens(), readInputFileHeader(), and validateObservableVariables().

The documentation for this class was generated from the following files:

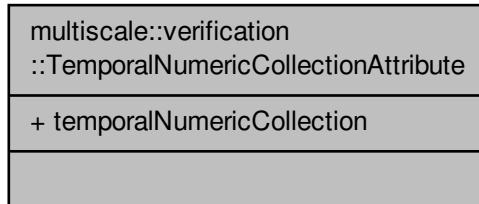
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/[TemporalDataReader.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[TemporalDataReader.cpp](#)

## 7.195 multiscale::verification::TemporalNumericCollectionAttribute Class Reference

Class for representing a temporal numeric collection attribute.

```
#include <TemporalNumericCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TemporalNumericCollectionAttribute:



### Public Attributes

- [TemporalNumericCollectionType temporalNumericCollection](#)

#### 7.195.1 Detailed Description

Class for representing a temporal numeric collection attribute.

Definition at line 29 of file `TemporalNumericCollectionAttribute.hpp`.

#### 7.195.2 Member Data Documentation

##### 7.195.2.1 [TemporalNumericCollectionType multiscale::verification::TemporalNumericCollectionAttribute::temporalNumericCollection](#)

The temporal numeric collection

Definition at line 33 of file `TemporalNumericCollectionAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionEvaluator::evaluateTemporalNumericCollection()`.

The documentation for this class was generated from the following file:

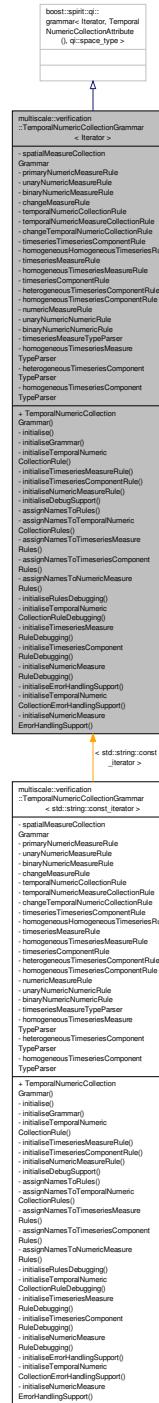
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp](#)

## 7.196 multiscale::verification::TemporalNumericCollectionGrammar< Iterator > Class Template Reference

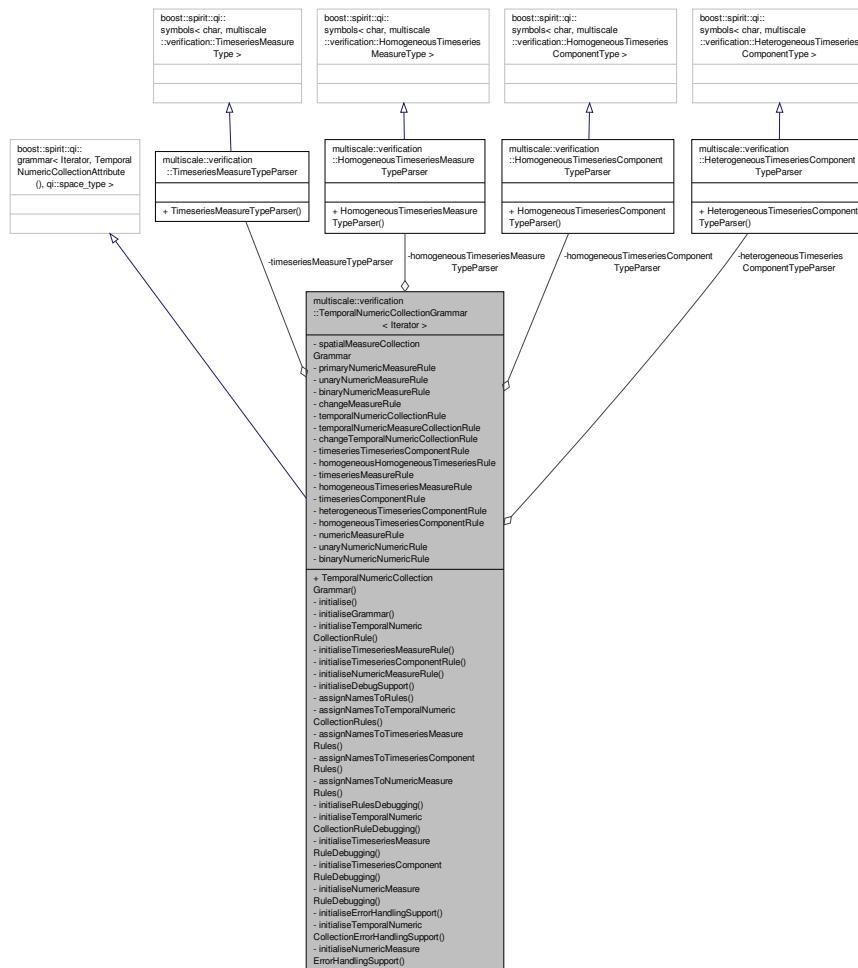
The grammar for parsing temporal numeric collection statements.

```
#include <TemporalNumericCollectionGrammar.hpp>
```

Inheritance diagram for multiscale::verification::TemporalNumericCollectionGrammar< Iterator >:



Collaboration diagram for multiscale::verification::TemporalNumericCollectionGrammar< Iterator >:



## Public Member Functions

- [TemporalNumericCollectionGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseTemporalNumericCollectionRule \(\)](#)  
*Initialise the temporal numeric collection rule.*
- void [initialiseTimeseriesMeasureRule \(\)](#)  
*Initialise the timeseries measure rule.*
- void [initialiseTimeseriesComponentRule \(\)](#)  
*Initialise the timeseries component rule.*
- void [initialiseNumericMeasureRule \(\)](#)  
*Initialise the numeric measure rule.*

- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [assignNamesToTemporalNumericCollectionRules \(\)](#)  
*Assign names to the temporal numeric collection rule.*
- void [assignNamesToTimeseriesMeasureRules \(\)](#)  
*Assign names to the timeseries measure rule.*
- void [assignNamesToTimeseriesComponentRules \(\)](#)  
*Assign names to the timeseries component rule.*
- void [assignNamesToNumericMeasureRules \(\)](#)  
*Assign names to the numeric measure rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*
- void [initialiseTemporalNumericCollectionRuleDebugging \(\)](#)  
*Initialise debugging for the temporal numeric collection rule.*
- void [initialiseTimeseriesMeasureRuleDebugging \(\)](#)  
*Initialise debugging for the timeseries measure rule.*
- void [initialiseTimeseriesComponentRuleDebugging \(\)](#)  
*Initialise debugging for the timeseries component rule.*
- void [initialiseNumericMeasureRuleDebugging \(\)](#)  
*Initialise debugging for the numeric measure rule.*
- void [initialiseErrorHandlingSupport \(\)](#)  
*Initialise the error handling routines.*
- void [initialiseTemporalNumericCollectionErrorHandlingSupport \(\)](#)  
*Initialise the temporal numeric collection error handling support.*
- void [initialiseNumericMeasureErrorHandlingSupport \(\)](#)  
*Initialise the numeric measure error handling support.*

## Private Attributes

- std::shared\_ptr  
  < [SpatialMeasureCollectionGrammar](#)  
  < Iterator > > [spatialMeasureCollectionGrammar](#)
- std::shared\_ptr  
  < [PrimaryNumericMeasureGrammar](#)  
  < Iterator > > [primaryNumericMeasureRule](#)
- [UnaryNumericMeasureGrammar](#)  
  < Iterator > [unaryNumericMeasureRule](#)
- [BinaryNumericMeasureGrammar](#)  
  < Iterator > [binaryNumericMeasureRule](#)
- [ChangeMeasureGrammar](#)< Iterator > [changeMeasureRule](#)
- qi::rule< Iterator,  
  [TemporalNumericCollectionAttribute\(\)](#),  
  qi::space\_type > [temporalNumericCollectionRule](#)
- qi::rule< Iterator,  
  [TemporalNumericMeasureCollectionAttribute\(\)](#),  
  qi::space\_type > [temporalNumericMeasureCollectionRule](#)
- qi::rule< Iterator,  
  [ChangeTemporalNumericCollectionAttribute\(\)](#),  
  qi::space\_type > [changeTemporalNumericCollectionRule](#)

- qi::rule< Iterator,  
TimeseriesTimeseriesComponentAttribute(),  
qi::space\_type > timeseriesTimeseriesComponentRule
- qi::rule< Iterator,  
HomogeneousHomogeneousTimeseriesAttribute(),  
qi::space\_type > homogeneousHomogeneousTimeseriesRule
- qi::rule< Iterator,  
TimeseriesMeasureAttribute(),  
qi::space\_type > timeseriesMeasureRule
- qi::rule< Iterator,  
HomogeneousTimeseriesMeasureAttribute(),  
qi::space\_type > homogeneousTimeseriesMeasureRule
- qi::rule< Iterator,  
TimeseriesComponentAttribute(),  
qi::space\_type > timeseriesComponentRule
- qi::rule< Iterator,  
HeterogeneousTimeseriesComponentAttribute(),  
qi::space\_type > heterogeneousTimeseriesComponentRule
- qi::rule< Iterator,  
HomogeneousTimeseriesComponentAttribute(),  
qi::space\_type > homogeneousTimeseriesComponentRule
- qi::rule< Iterator,  
NumericMeasureAttribute(),  
qi::space\_type > numericMeasureRule
- qi::rule< Iterator,  
UnaryNumericNumericAttribute(),  
qi::space\_type > unaryNumericNumericRule
- qi::rule< Iterator,  
BinaryNumericNumericAttribute(),  
qi::space\_type > binaryNumericNumericRule
- TimeseriesMeasureTypeParser timeseriesMeasureTypeParser
- HomogeneousTimeseriesMeasureTypeParser homogeneousTimeseriesMeasureTypeParser
- HeterogeneousTimeseriesComponentTypeParser heterogeneousTimeseriesComponentTypeParser
- HomogeneousTimeseriesComponentTypeParser homogeneousTimeseriesComponentTypeParser

### 7.196.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::TemporalNumericCollectionGrammar< Iterator >
```

The grammar for parsing temporal numeric collection statements.

Definition at line 34 of file TemporalNumericCollectionGrammar.hpp.

### 7.196.2 Constructor & Destructor Documentation

```
7.196.2.1 template<typename Iterator > multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::TemporalNumericCollectionGrammar()
```

Definition at line 31 of file TemporalNumericCollectionGrammarDefinition.hpp.

References multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialise(), multiscale< ::verification::TemporalNumericCollectionGrammar< Iterator >::primaryNumericMeasureRule, and multiscale< ::verification::TemporalNumericCollectionGrammar< Iterator >::spatialMeasureCollectionGrammar.

### 7.196.3 Member Function Documentation

7.196.3.1 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::assignNamesToNumericMeasureRules( ) [private]`

Assign names to the numeric measure rules.

Definition at line 206 of file TemporalNumericCollectionGrammarDefinition.hpp.

7.196.3.2 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::assignNamesToRules( ) [private]`

Assign names to the rules.

Definition at line 172 of file TemporalNumericCollectionGrammarDefinition.hpp.

7.196.3.3 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::assignNamesToTemporalNumericCollectionRules( ) [private]`

Assign names to the temporal numeric collection rule.

Assign names to the temporal numeric collection rules.

Definition at line 181 of file TemporalNumericCollectionGrammarDefinition.hpp.

7.196.3.4 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::assignNamesToTimeseriesComponentRules( ) [private]`

Assign names to the timeseries component rule.

Assign names to the timeseries component rules.

Definition at line 198 of file TemporalNumericCollectionGrammarDefinition.hpp.

7.196.3.5 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::assignNamesToTimeseriesMeasureRules( ) [private]`

Assign names to the timeseries measure rule.

Assign names to the timeseries measure rules.

Definition at line 191 of file TemporalNumericCollectionGrammarDefinition.hpp.

7.196.3.6 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::initialise( ) [private]`

Initialisation function.

Definition at line 47 of file TemporalNumericCollectionGrammarDefinition.hpp.

Referenced by `multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::TemporalNumericCollectionGrammar()`.

7.196.3.7 `template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<Iterator>::initialiseDebugSupport( ) [private]`

Initialise debug support.

Definition at line 163 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.8 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseErrorHandlingSupport() [private]
```

Initialise the error handling routines.

Definition at line 257 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.9 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseGrammar() [private]
```

Initialise the grammar.

Definition at line 55 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.10 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseNumericMeasureErrorHandlingSupport() [private]
```

Initialise the numeric measure error handling support.

Definition at line 285 of file TemporalNumericCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

```
7.196.3.11 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseNumericMeasureRule() [private]
```

Initialise the numeric measure rule.

Definition at line 136 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.12 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseNumericMeasureRuleDebugging() [private]
```

Initialise debugging for the numeric measure rule.

Definition at line 249 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.13 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseRulesDebugging() [private]
```

Initialise the debugging of rules.

Definition at line 215 of file TemporalNumericCollectionGrammarDefinition.hpp.

```
7.196.3.14 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseTemporalNumericCollectionErrorHandlingSupport() [private]
```

Initialise the temporal numeric collection error handling support.

Definition at line 264 of file TemporalNumericCollectionGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

```
7.196.3.15 template<typename Iterator> void multiscale::verification::TemporalNumericCollectionGrammar<
Iterator>::initialiseTemporalNumericCollectionRule() [private]
```

Initialise the temporal numeric collection rule.

Definition at line 64 of file TemporalNumericCollectionGrammarDefinition.hpp.

**7.196.3.16 template<typename Iterator > void multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTemporalNumericCollectionRuleDebugging( ) [private]**

Initialise debugging for the temporal numeric collection rule.

Definition at line 224 of file TemporalNumericCollectionGrammarDefinition.hpp.

**7.196.3.17 template<typename Iterator > void multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTimeseriesComponentRule( ) [private]**

Initialise the timeseries component rule.

Definition at line 122 of file TemporalNumericCollectionGrammarDefinition.hpp.

**7.196.3.18 template<typename Iterator > void multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTimeseriesComponentRuleDebugging( ) [private]**

Initialise debugging for the timeseries component rule.

Definition at line 241 of file TemporalNumericCollectionGrammarDefinition.hpp.

**7.196.3.19 template<typename Iterator > void multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTimeseriesMeasureRule( ) [private]**

Initialise the timeseries measure rule.

Definition at line 112 of file TemporalNumericCollectionGrammarDefinition.hpp.

**7.196.3.20 template<typename Iterator > void multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::initialiseTimeseriesMeasureRuleDebugging( ) [private]**

Initialise debugging for the timeseries measure rule.

Initialise debugging for the timeseries measures rule.

Definition at line 234 of file TemporalNumericCollectionGrammarDefinition.hpp.

## 7.196.4 Member Data Documentation

**7.196.4.1 template<typename Iterator> BinaryNumericMeasureGrammar<Iterator> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::binaryNumericMeasureRule [private]**

The grammar for parsing binary numeric measures

Definition at line 53 of file TemporalNumericCollectionGrammar.hpp.

**7.196.4.2 template<typename Iterator> qi::rule<Iterator, BinaryNumericNumericAttribute(), qi::space\_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::binaryNumericNumericRule [private]**

The rule for parsing a binary numeric numeric attribute

Definition at line 99 of file TemporalNumericCollectionGrammar.hpp.

---

```
7.196.4.3 template<typename Iterator> ChangeMeasureGrammar<Iterator> multiscale<-
 ::verification::TemporalNumericCollectionGrammar< Iterator >::changeMeasureRule
 [private]
```

The grammar for parsing change measures

Definition at line 57 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.4 template<typename Iterator> qi::rule<Iterator, ChangeTemporalNumericCollectionAttribute(),-
 qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::changeTemporalNumericCollectionRule [private]
```

The rule for parsing a change temporal numeric collections attribute

Definition at line 68 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.5 template<typename Iterator> qi::rule<Iterator, HeterogeneousTimeseriesComponentAttribute(),-
 qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::heterogeneousTimeseriesComponentRule [private]
```

The rule for parsing a heterogeneous timeseries component

Definition at line 87 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.6 template<typename Iterator> HeterogeneousTimeseriesComponentTypeParser
 multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::heterogeneousTimeseriesComponentTypeParser [private]
```

The heterogeneous timeseries component type parser

Definition at line 111 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.7 template<typename Iterator> qi::rule<Iterator, HomogeneousHomogeneousTimeseriesAttribute(),-
 qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::homogeneousHomogeneousTimeseriesRule [private]
```

The rule for parsing a homogeneous homogeneous timeseries measure attribute

Definition at line 74 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.8 template<typename Iterator> qi::rule<Iterator, HomogeneousTimeseriesComponentAttribute(),-
 qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::homogeneousTimeseriesComponentRule [private]
```

The rule for parsing a homogeneous timeseries component

Definition at line 90 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.9 template<typename Iterator> HomogeneousTimeseriesComponentTypeParser
 multiscale::verification::TemporalNumericCollectionGrammar< Iterator
 >::homogeneousTimeseriesComponentTypeParser [private]
```

The homogeneous timeseries component type parser

Definition at line 114 of file TemporalNumericCollectionGrammar.hpp.

7.196.4.10 template<typename Iterator> qi::rule<Iterator, HomogeneousTimeseriesMeasureAttribute(), qi::space\_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::homogeneousTimeseriesMeasureRule [private]

The rule for parsing a homogeneous timeseries measure

Definition at line 80 of file TemporalNumericCollectionGrammar.hpp.

7.196.4.11 template<typename Iterator> HomogeneousTimeseriesMeasureTypeParser multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::homogeneousTimeseriesMeasureTypeParser [private]

The homogeneous timeseries measure type parser

Definition at line 107 of file TemporalNumericCollectionGrammar.hpp.

7.196.4.12 template<typename Iterator> qi::rule<Iterator, NumericMeasureAttribute(), qi::space\_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::numericMeasureRule [private]

The rule for parsing a numeric measure

Definition at line 94 of file TemporalNumericCollectionGrammar.hpp.

7.196.4.13 template<typename Iterator> std::shared\_ptr<PrimaryNumericMeasureGrammar<Iterator>> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::primaryNumericMeasureRule [private]

The grammar for parsing primary numeric measures

Definition at line 47 of file TemporalNumericCollectionGrammar.hpp.

Referenced by multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::TemporalNumericCollectionGrammar().

7.196.4.14 template<typename Iterator> std::shared\_ptr<SpatialMeasureCollectionGrammar<Iterator>> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::spatialMeasureCollectionGrammar [private]

The grammar for parsing spatial measure collection which will be passed by reference to the primary numeric measure grammar

Definition at line 42 of file TemporalNumericCollectionGrammar.hpp.

Referenced by multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::TemporalNumericCollectionGrammar().

7.196.4.15 template<typename Iterator> qi::rule<Iterator, TemporalNumericCollectionAttribute(), qi::space\_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::temporalNumericCollectionRule [private]

The rule for parsing temporal numeric collections

Definition at line 62 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.16 template<typename Iterator> qi::rule<Iterator, TemporalNumericMeasureCollectionAttribute(),
qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
>::temporalNumericMeasureCollectionRule [private]
```

The rule for parsing temporal numeric measure collections

Definition at line 65 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.17 template<typename Iterator> qi::rule<Iterator, TimeseriesComponentAttribute(), qi::space_type>
multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::timeseriesComponentRule
[private]
```

The rule for parsing a timeseries component

Definition at line 84 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.18 template<typename Iterator> qi::rule<Iterator, TimeseriesMeasureAttribute(), qi::space_type>
multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::timeseriesMeasureRule
[private]
```

The rule for parsing a timeseries measure

Definition at line 78 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.19 template<typename Iterator> TimeseriesMeasureTypeParser multiscale::verification<~
::TemporalNumericCollectionGrammar< Iterator >::timeseriesMeasureTypeParser
[private]
```

The timeseries measure type parser

Definition at line 105 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.20 template<typename Iterator> qi::rule<Iterator, TimeseriesTimeseriesComponentAttribute(),
qi::space_type> multiscale::verification::TemporalNumericCollectionGrammar< Iterator
>::timeseriesTimeseriesComponentRule [private]
```

The rule for parsing a timeseries timeseries component attribute

Definition at line 71 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.21 template<typename Iterator> UnaryNumericMeasureGrammar<Iterator> multiscale<~
::verification::TemporalNumericCollectionGrammar< Iterator >::unaryNumericMeasureRule
[private]
```

The grammar for parsing unary numeric measures

Definition at line 50 of file TemporalNumericCollectionGrammar.hpp.

```
7.196.4.22 template<typename Iterator> qi::rule<Iterator, UnaryNumericNumericAttribute(), qi::space_type>
multiscale::verification::TemporalNumericCollectionGrammar< Iterator >::unaryNumericNumericRule
[private]
```

The rule for parsing a unary numeric numeric attribute

Definition at line 96 of file TemporalNumericCollectionGrammar.hpp.

The documentation for this class was generated from the following files:

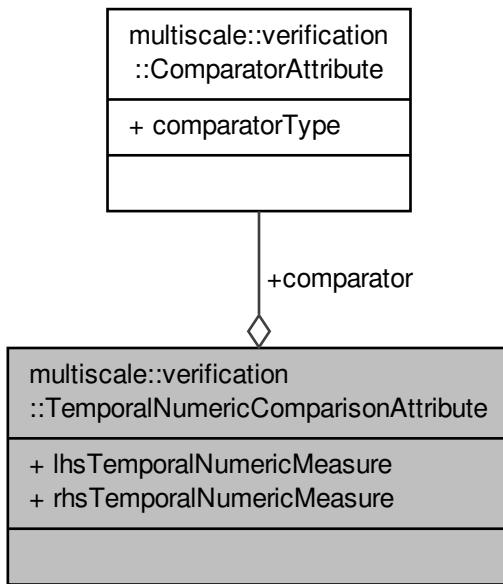
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[TemporalNumericCollectionGrammar.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[TemporalNumericCollectionGrammarDefinition.hpp](#)

## 7.197 multiscale::verification::TemporalNumericComparisonAttribute Class Reference

Class for representing a temporal numeric comparison attribute.

```
#include <TemporalNumericComparisonAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TemporalNumericComparisonAttribute:



### Public Attributes

- `TemporalNumericMeasureType lhsTemporalNumericMeasure`
- `ComparatorAttribute comparator`
- `TemporalNumericMeasureType rhsTemporalNumericMeasure`

#### 7.197.1 Detailed Description

Class for representing a temporal numeric comparison attribute.

Definition at line 15 of file `TemporalNumericComparisonAttribute.hpp`.

#### 7.197.2 Member Data Documentation

### 7.197.2.1 ComparatorAttribute multiscale::verification::TemporalNumericComparisonAttribute::comparator

The comparator

Definition at line 22 of file TemporalNumericComparisonAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericComparison().

### 7.197.2.2 TemporalNumericMeasureType multiscale::verification::TemporalNumericComparisonAttribute::lhsTemporal< NumericMeasure

The temporal numeric measure preceding the comparator

Definition at line 20 of file TemporalNumericComparisonAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericComparison().

### 7.197.2.3 TemporalNumericMeasureType multiscale::verification::TemporalNumericComparisonAttribute::rhsTemporal< NumericMeasure

The temporal numeric measure succeeding the comparator

Definition at line 24 of file TemporalNumericComparisonAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateTemporalNumericComparison().

The documentation for this class was generated from the following file:

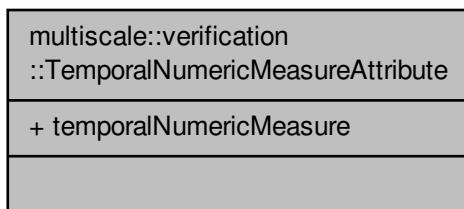
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[TemporalNumericComparisonAttribute.hpp](#)

## 7.198 multiscale::verification::TemporalNumericMeasureAttribute Class Reference

Class for representing a temporal numeric measure attribute.

```
#include <TemporalNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TemporalNumericMeasureAttribute:



### Public Attributes

- [TemporalNumericMeasureType temporalNumericMeasure](#)

### 7.198.1 Detailed Description

Class for representing a temporal numeric measure attribute.

Definition at line 32 of file TemporalNumericMeasureAttribute.hpp.

### 7.198.2 Member Data Documentation

#### 7.198.2.1 TemporalNumericMeasureType multiscale::verification::TemporalNumericMeasureAttribute::temporal< NumericMeasure

The temporal numeric measure

Definition at line 36 of file TemporalNumericMeasureAttribute.hpp.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()().

The documentation for this class was generated from the following file:

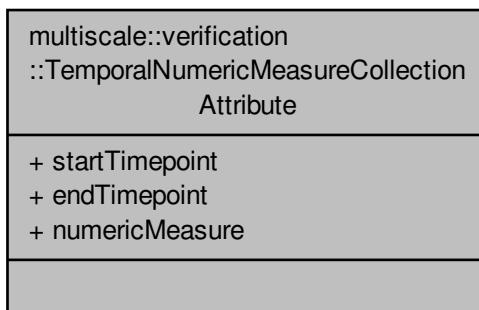
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[TemporalNumericMeasureAttribute.hpp](#)

## 7.199 multiscale::verification::TemporalNumericMeasureCollectionAttribute Class Reference

Class for representing temporal numeric measure collection attributes.

```
#include <TemporalNumericMeasureCollectionAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TemporalNumericMeasureCollectionAttribute:



### Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [NumericMeasureType](#) [numericMeasure](#)

### 7.199.1 Detailed Description

Class for representing temporal numeric measure collection attributes.

Definition at line 14 of file TemporalNumericMeasureCollectionAttribute.hpp.

### 7.199.2 Member Data Documentation

#### 7.199.2.1 unsigned long multiscale::verification::TemporalNumericMeasureCollectionAttribute::endTimepoint

The considered end timepoint

Definition at line 19 of file TemporalNumericMeasureCollectionAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

#### 7.199.2.2 NumericMeasureType multiscale::verification::TemporalNumericMeasureCollectionAttribute::numericMeasure

The numeric measure

Definition at line 21 of file TemporalNumericMeasureCollectionAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

#### 7.199.2.3 unsigned long multiscale::verification::TemporalNumericMeasureCollectionAttribute::startTimepoint

The considered start timepoint

Definition at line 18 of file TemporalNumericMeasureCollectionAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[TemporalNumericMeasureCollectionAttribute.hpp](#)

## 7.200 multiscale::verification::TemporalNumericMeasureGrammar< Iterator > Class Template Reference

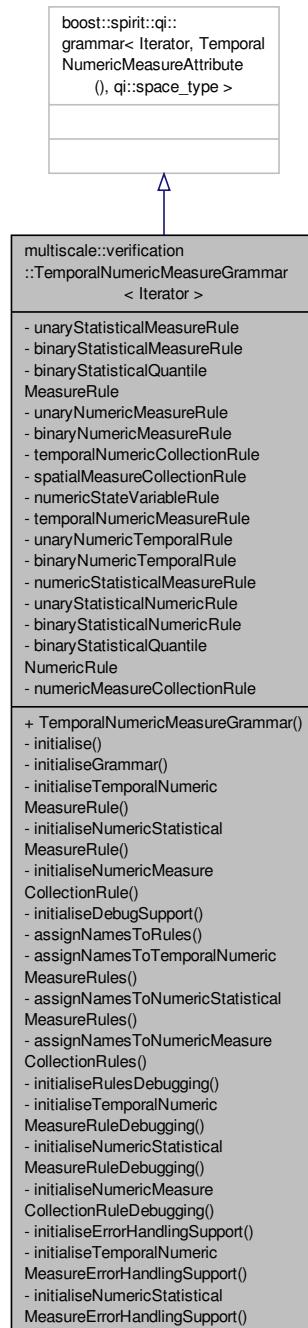
The grammar for parsing temporal numeric measure statements.

```
#include <TemporalNumericMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::TemporalNumericMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::TemporalNumericMeasureGrammar< Iterator >:



## Public Member Functions

- [TemporalNumericMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)

- `Initialisation function.`
- `void initialiseGrammar ()`  
*Initialise the grammar.*
- `void initialiseTemporalNumericMeasureRule ()`  
*Initialise the temporal numeric measure rule.*
- `void initialiseNumericStatisticalMeasureRule ()`  
*Initialise the numeric statistical measure rules.*
- `void initialiseNumericMeasureCollectionRule ()`  
*Initialise the numeric measure collection rule.*
- `void initialiseDebugSupport ()`  
*Initialise debug support.*
- `void assignNamesToRules ()`  
*Assign names to the rules.*
- `void assignNamesToTemporalNumericMeasureRules ()`  
*Assign names to the temporal numeric measure rules.*
- `void assignNamesToNumericStatisticalMeasureRules ()`  
*Assign names to the numeric statistical measure rules.*
- `void assignNamesToNumericMeasureCollectionRules ()`  
*Assign names to the numeric measure collection rules.*
- `void initialiseRulesDebugging ()`  
*Initialise the debugging of rules.*
- `void initialiseTemporalNumericMeasureRuleDebugging ()`  
*Initialise debugging for the temporal numeric measure rule.*
- `void initialiseNumericStatisticalMeasureRuleDebugging ()`  
*Initialise debugging for the numeric statistical measure rule.*
- `void initialiseNumericMeasureCollectionRuleDebugging ()`  
*Initialise debugging for the numeric measure collection rule.*
- `void initialiseErrorHandlingSupport ()`  
*Initialise the error handling routines.*
- `void initialiseTemporalNumericMeasureErrorHandlingSupport ()`  
*Initialise the temporal numeric measure error handling support.*
- `void initialiseNumericStatisticalMeasureErrorHandlingSupport ()`  
*Initialise the numeric statistical measure error handling support.*

## Private Attributes

- `UnaryStatisticalMeasureGrammar`  
< Iterator > `unaryStatisticalMeasureRule`
- `BinaryStatisticalMeasureGrammar`  
< Iterator > `binaryStatisticalMeasureRule`
- `BinaryStatisticalQuantileMeasureGrammar`  
< Iterator > `binaryStatisticalQuantileMeasureRule`
- `UnaryNumericMeasureGrammar`  
< Iterator > `unaryNumericMeasureRule`
- `BinaryNumericMeasureGrammar`  
< Iterator > `binaryNumericMeasureRule`
- `TemporalNumericCollectionGrammar`  
< Iterator > `temporalNumericCollectionRule`
- `SpatialMeasureCollectionGrammar`  
< Iterator > `spatialMeasureCollectionRule`
- `NumericStateVariableGrammar`  
< Iterator > `numericStateVariableRule`

- qi::rule< Iterator,  
  TemporalNumericMeasureAttribute(),  
  qi::space\_type > temporalNumericMeasureRule
- qi::rule< Iterator,  
  UnaryNumericTemporalAttribute(),  
  qi::space\_type > unaryNumericTemporalRule
- qi::rule< Iterator,  
  BinaryNumericTemporalAttribute(),  
  qi::space\_type > binaryNumericTemporalRule
- qi::rule< Iterator,  
  NumericStatisticalMeasureAttribute(),  
  qi::space\_type > numericStatisticalMeasureRule
- qi::rule< Iterator,  
  UnaryStatisticalNumericAttribute(),  
  qi::space\_type > unaryStatisticalNumericRule
- qi::rule< Iterator,  
  BinaryStatisticalNumericAttribute(),  
  qi::space\_type > binaryStatisticalNumericRule
- qi::rule< Iterator,  
  BinaryStatisticalQuantileNumericAttribute(),  
  qi::space\_type > binaryStatisticalQuantileNumericRule
- qi::rule< Iterator,  
  NumericMeasureCollectionAttribute(),  
  qi::space\_type > numericMeasureCollectionRule

### 7.200.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::TemporalNumericMeasureGrammar< Iterator >
```

The grammar for parsing temporal numeric measure statements.

Definition at line 38 of file TemporalNumericMeasureGrammar.hpp.

### 7.200.2 Constructor & Destructor Documentation

```
7.200.2.1 template<typename Iterator > multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::TemporalNumericMeasureGrammar()
```

Definition at line 29 of file TemporalNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialise().

### 7.200.3 Member Function Documentation

```
7.200.3.1 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::assignNamesToNumericMeasureCollectionRules() [private]
```

Assign names to the numeric measure collection rules.

Definition at line 160 of file TemporalNumericMeasureGrammarDefinition.hpp.

```
7.200.3.2 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::assignNamesToNumericStatisticalMeasureRules() [private]
```

Assign names to the numeric statistical measure rules.

Definition at line 151 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.3 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::assignNamesToRules( ) [private]

Assign names to the rules.

Definition at line 135 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.4 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::assignNamesToTemporalNumericMeasureRules( ) [private]

Assign names to the temporal numeric measure rules.

Definition at line 143 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.5 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialise( ) [private]

Initialisation function.

Definition at line 36 of file TemporalNumericMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::TemporalNumericMeasureGrammar().

7.200.3.6 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseDebugSupport( ) [private]

Initialise debug support.

Definition at line 126 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.7 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseErrorHandlingSupport( ) [private]

Initialise the error handling routines.

Definition at line 197 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.8 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseGrammar( ) [private]

Initialise the grammar.

Definition at line 44 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.9 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseNumericMeasureCollectionRule( ) [private]

Initialise the numeric measure collection rule.

Definition at line 118 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.10 template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseNumericMeasureCollectionRuleDebugging( ) [private]

Initialise debugging for the numeric measure collection rule.

Definition at line 191 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.11 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseNumericStatisticalMeasureErrorHandlingSupport( ) [private]**

Initialise the numeric statistical measure error handling support.

Definition at line 221 of file TemporalNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.200.3.12 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseNumericStatisticalMeasureRule( ) [private]**

Initialise the numeric statistical measure rules.

Definition at line 81 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.13 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseNumericStatisticalMeasureRuleDebugging( ) [private]**

Initialise debugging for the numeric statistical measure rule.

Definition at line 182 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.14 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]**

Initialise the debugging of rules.

Definition at line 166 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.15 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseTemporalNumericMeasureErrorHandlingSupport( ) [private]**

Initialise the temporal numeric measure error handling support.

Definition at line 204 of file TemporalNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::handleUnexpectedTokenError.

7.200.3.16 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseTemporalNumericMeasureRule( ) [private]**

Initialise the temporal numeric measure rule.

Definition at line 52 of file TemporalNumericMeasureGrammarDefinition.hpp.

7.200.3.17 **template<typename Iterator > void multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::initialiseTemporalNumericMeasureRuleDebugging( ) [private]**

Initialise debugging for the temporal numeric measure rule.

Definition at line 174 of file TemporalNumericMeasureGrammarDefinition.hpp.

## 7.200.4 Member Data Documentation

---

7.200.4.1 template<typename Iterator> **BinaryNumericMeasureGrammar<Iterator>** **multiscale**<  
**::verification::TemporalNumericMeasureGrammar< Iterator >**::binaryNumericMeasureRule  
[private]

The grammar for parsing a binary numeric measure

Definition at line 59 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.2 template<typename Iterator> qi::rule<Iterator, BinaryNumericTemporalAttribute(), qi::space\_type>  
**multiscale**<**verification::TemporalNumericMeasureGrammar< Iterator >**::binaryNumericTemporalRule  
[private]

The rule for parsing a binary numeric temporal attribute

Definition at line 82 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.3 template<typename Iterator> **BinaryStatisticalMeasureGrammar<Iterator>** **multiscale**<  
**::verification::TemporalNumericMeasureGrammar< Iterator >**::binaryStatisticalMeasureRule  
[private]

The grammar for parsing a binary statistical measure

Definition at line 49 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.4 template<typename Iterator> qi::rule<Iterator, BinaryStatisticalNumericAttribute(), qi::space\_type>  
**multiscale**<**verification::TemporalNumericMeasureGrammar< Iterator >**::binaryStatisticalNumericRule  
[private]

The rule for parsing a binary statistical numeric attribute

Definition at line 92 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.5 template<typename Iterator> **BinaryStatisticalQuantileMeasureGrammar<Iterator>** **multiscale**<  
**::verification::TemporalNumericMeasureGrammar< Iterator >**::binaryStatisticalQuantileMeasureRule  
[private]

The grammar for parsing a binary statistical quantile measure

Definition at line 52 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.6 template<typename Iterator> qi::rule<Iterator, BinaryStatisticalQuantileNumericAttribute(),  
qi::space\_type> **multiscale**<**verification::TemporalNumericMeasureGrammar< Iterator >**::binaryStatisticalQuantileNumericRule  
[private]

The rule for parsing a binary statistical quantile numeric attribute

Definition at line 95 of file TemporalNumericMeasureGrammar.hpp.

7.200.4.7 template<typename Iterator> qi::rule<Iterator, NumericMeasureCollectionAttribute(),  
qi::space\_type> **multiscale**<**verification::TemporalNumericMeasureGrammar< Iterator >**::numericMeasureCollectionRule  
[private]

The rule for parsing numeric measure collections

Definition at line 99 of file TemporalNumericMeasureGrammar.hpp.

---

```
7.200.4.8 template<typename Iterator> NumericStateVariableGrammar<Iterator> multiscale<-
 ::verification::TemporalNumericMeasureGrammar< Iterator >::numericStateVariableRule
 [private]
```

The grammar for parsing a numeric state variable

Definition at line 70 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.9 template<typename Iterator> qi::rule<Iterator, NumericStatisticalMeasureAttribute(), qi::space_type>
 multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::numericStatisticalMeasureRule
 [private]
```

The rule for parsing a numeric statistical measure

Definition at line 86 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.10 template<typename Iterator> SpatialMeasureCollectionGrammar<Iterator> multiscale<-
 ::verification::TemporalNumericMeasureGrammar< Iterator >::spatialMeasureCollectionRule
 [private]
```

The grammar for parsing a spatial measure collection

Definition at line 66 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.11 template<typename Iterator> TemporalNumericCollectionGrammar<Iterator> multiscale<-
 ::verification::TemporalNumericMeasureGrammar< Iterator >::temporalNumericCollectionRule
 [private]
```

The grammar for parsing a temporal numeric collection

Definition at line 63 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.12 template<typename Iterator> qi::rule<Iterator, TemporalNumericMeasureAttribute(),
 qi::space_type> multiscale::verification::TemporalNumericMeasureGrammar< Iterator
 >::temporalNumericMeasureRule [private]
```

The rule for parsing a temporal numeric measure

Definition at line 76 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.13 template<typename Iterator> UnaryNumericMeasureGrammar<Iterator> multiscale<-
 ::verification::TemporalNumericMeasureGrammar< Iterator >::unaryNumericMeasureRule
 [private]
```

The grammar for parsing a unary numeric measure

Definition at line 56 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.14 template<typename Iterator> qi::rule<Iterator, UnaryNumericTemporalAttribute(), qi::space_type>
 multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::unaryNumericTemporalRule
 [private]
```

The rule for parsing a unary numeric temporal attribute

Definition at line 79 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.15 template<typename Iterator> UnaryStatisticalMeasureGrammar<Iterator> multiscale<~
::verification::TemporalNumericMeasureGrammar< Iterator >::unaryStatisticalMeasureRule
[private]
```

The grammar for parsing a unary statistical measure

Definition at line 46 of file TemporalNumericMeasureGrammar.hpp.

```
7.200.4.16 template<typename Iterator> qi::rule<Iterator, UnaryStatisticalNumericAttribute(), qi::space_type>
multiscale::verification::TemporalNumericMeasureGrammar< Iterator >::unaryStatisticalNumericRule
[private]
```

The rule for parsing a unary statistical numeric attribute

Definition at line 89 of file TemporalNumericMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

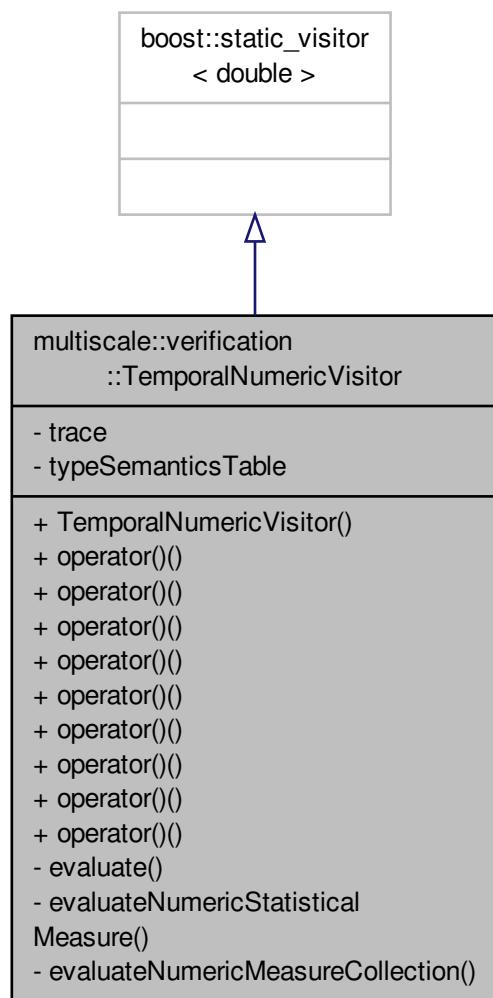
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[TemporalNumericMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[TemporalNumericMeasureGrammarDefinition.hpp](#)

## 7.201 multiscale::verification::TemporalNumericVisitor Class Reference

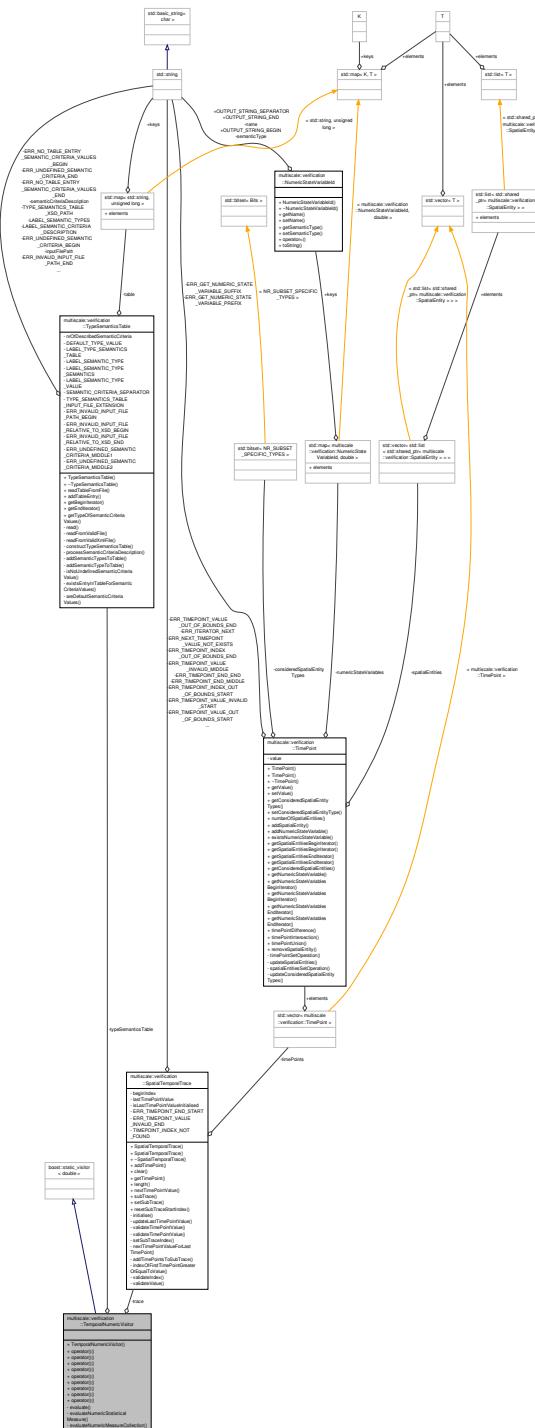
Class for evaluating temporal numeric measures.

```
#include <TemporalNumericVisitor.hpp>
```

## Inheritance diagram for multiscale::verification::TemporalNumericVisitor:



## Collaboration diagram for multiscale::verification::TemporalNumericVisitor



## Public Member Functions

- `TemporalNumericVisitor` (const `SpatialTemporalTrace` &`trace`, const `TypeSemanticsTable` &`typeSemanticsTable`)
  - double `operator()` (const `TemporalNumericMeasureAttribute` &`temporalNumericMeasure`) const  
*Overloading the "()" operator for the `TemporalNumericMeasureAttribute` alternative.*
  - double `operator()` (double `realNumber`) const

- Overloading the "()" operator for the real number alternative.
- double operator() (const [NumericStateVariableAttribute](#) &numericStateVariable) const
  - Overloading the "()" operator for the [NumericStateVariableAttribute](#) alternative.
- double operator() (const [NumericStatisticalMeasureAttribute](#) &numericStatisticalMeasure) const
  - Overloading the "()" operator for the [NumericStatisticalMeasureAttribute](#) alternative.
- double operator() (const [UnaryNumericTemporalAttribute](#) &unaryNumericTemporalMeasure) const
  - Overloading the "()" operator for the [UnaryNumericTemporalAttribute](#) alternative.
- double operator() (const [BinaryNumericTemporalAttribute](#) &binaryNumericTemporalMeasure) const
  - Overloading the "()" operator for the [BinaryNumericTemporalAttribute](#) alternative.
- double operator() (const [UnaryStatisticalNumericAttribute](#) &unaryStatisticalNumericAttribute) const
  - Overloading the "()" operator for the [UnaryStatisticalNumericAttribute](#) alternative.
- double operator() (const [BinaryStatisticalNumericAttribute](#) &binaryStatisticalNumericAttribute) const
  - Overloading the "()" operator for the [BinaryStatisticalNumericAttribute](#) alternative.
- double operator() (const [BinaryStatisticalQuantileNumericAttribute](#) &binaryStatisticalQuantileNumericAttribute) const
  - Overloading the "()" operator for the [BinaryStatisticalQuantileNumericAttribute](#) alternative.

## Private Member Functions

- double evaluate (const [TemporalNumericMeasureType](#) &temporalNumericMeasure) const
  - Evaluate the given temporal numeric measure considering the trace field.
- double evaluateNumericStatisticalMeasure (const [NumericStatisticalMeasureType](#) &numericStatisticalMeasure) const
  - Evaluate the given numeric statistical measure considering the trace field.
- std::vector< double > evaluateNumericMeasureCollection (const [NumericMeasureCollectionAttribute](#) &numericMeasureCollection) const
  - Evaluate the given numeric measure collection considering the trace field.

## Private Attributes

- const [SpatialTemporalTrace](#) & [trace](#)
- const [TypeSemanticsTable](#) & [typeSemanticsTable](#)

### 7.201.1 Detailed Description

Class for evaluating temporal numeric measures.

Definition at line 19 of file TemporalNumericVisitor.hpp.

### 7.201.2 Constructor & Destructor Documentation

**7.201.2.1 multiscale::verification::TemporalNumericVisitor::TemporalNumericVisitor ( const [SpatialTemporalTrace](#) & [trace](#), const [TypeSemanticsTable](#) & [typeSemanticsTable](#) ) [inline]**

Definition at line 28 of file TemporalNumericVisitor.hpp.

Referenced by [evaluate\(\)](#), and [evaluateNumericStatisticalMeasure\(\)](#).

### 7.201.3 Member Function Documentation

7.201.3.1 double multiscale::verification::TemporalNumericVisitor::evaluate ( const TemporalNumericMeasureType & *temporalNumericMeasure* ) const [inline], [private]

Evaluate the given temporal numeric measure considering the trace field.

**Parameters**

|                                                     |                                    |
|-----------------------------------------------------|------------------------------------|
| <i>temporal</i><br><i>Numeric</i><br><i>Measure</i> | The given temporal numeric measure |
|-----------------------------------------------------|------------------------------------|

Definition at line 180 of file TemporalNumericVisitor.hpp.

References TemporalNumericVisitor().

Referenced by operator()().

**7.201.3.2 std::vector< double > multiscale::verification::TemporalNumericVisitor::evaluateNumericMeasureCollection ( const NumericMeasureCollectionAttribute & *numericMeasureCollection* ) const [inline], [private]**

Evaluate the given numeric measure collection considering the trace field.

**Parameters**

|                                                       |                                      |
|-------------------------------------------------------|--------------------------------------|
| <i>numeric</i><br><i>Measure</i><br><i>Collection</i> | The given numeric measure collection |
|-------------------------------------------------------|--------------------------------------|

Definition at line 223 of file TemporalNumericVisitor.hpp.

References multiscale::verification::NumericMeasureCollectionAttribute::numericMeasureCollection, trace, and typeSemanticsTable.

Referenced by operator()().

**7.201.3.3 double multiscale::verification::TemporalNumericVisitor::evaluateNumericStatisticalMeasure ( const NumericStatisticalMeasureType & *numericStatisticalMeasure* ) const [inline], [private]**

Evaluate the given numeric statistical measure considering the trace field.

**Parameters**

|                                                        |                                       |
|--------------------------------------------------------|---------------------------------------|
| <i>numeric</i><br><i>Statistical</i><br><i>Measure</i> | The given numeric statistical measure |
|--------------------------------------------------------|---------------------------------------|

Definition at line 190 of file TemporalNumericVisitor.hpp.

References TemporalNumericVisitor().

Referenced by operator()().

**7.201.3.4 double multiscale::verification::TemporalNumericVisitor::operator() ( const TemporalNumericMeasureAttribute & *temporalNumericMeasure* ) const [inline]**

Overloading the "(") operator for the [TemporalNumericMeasureAttribute](#) alternative.

**Parameters**

|                                                     |                              |
|-----------------------------------------------------|------------------------------|
| <i>temporal</i><br><i>Numeric</i><br><i>Measure</i> | The temporal numeric measure |
|-----------------------------------------------------|------------------------------|

Definition at line 37 of file TemporalNumericVisitor.hpp.

References evaluate(), and multiscale::verification::TemporalNumericMeasureAttribute::temporalNumericMeasure.

7.201.3.5 double multiscale::verification::TemporalNumericVisitor::operator() ( double *realNumber* ) const [inline]

Overloading the "()" operator for the real number alternative.

**Parameters**

|                   |                 |
|-------------------|-----------------|
| <i>realNumber</i> | The real number |
|-------------------|-----------------|

Definition at line 46 of file TemporalNumericVisitor.hpp.

**7.201.3.6 double multiscale::verification::TemporalNumericVisitor::operator() ( const NumericStateVariableAttribute & numericStateVariable ) const [inline]**

Overloading the "(") operator for the [NumericStateVariableAttribute](#) alternative.

**Parameters**

|                             |                            |
|-----------------------------|----------------------------|
| <i>numericStateVariable</i> | The numeric state variable |
|-----------------------------|----------------------------|

Definition at line 55 of file TemporalNumericVisitor.hpp.

References multiscale::verification::TimePoint::getNumericStateVariable(), multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::StateVariableAttribute::name, multiscale::verification::NumericStateVariableAttribute::semanticType, and multiscale::verification::NumericStateVariableAttribute::stateVariable.

**7.201.3.7 double multiscale::verification::TemporalNumericVisitor::operator() ( const NumericStatisticalMeasureAttribute & numericStatisticalMeasure ) const [inline]**

Overloading the "(") operator for the [NumericStatisticalMeasureAttribute](#) alternative.

**Parameters**

|                                  |                                           |
|----------------------------------|-------------------------------------------|
| <i>numericStatisticalMeasure</i> | The numeric statistical measure attribute |
|----------------------------------|-------------------------------------------|

Definition at line 71 of file TemporalNumericVisitor.hpp.

References evaluateNumericStatisticalMeasure(), and multiscale::verification::NumericStatisticalMeasureAttribute::numericStatisticalMeasure.

**7.201.3.8 double multiscale::verification::TemporalNumericVisitor::operator() ( const UnaryNumericTemporalAttribute & unaryNumericTemporalMeasure ) const [inline]**

Overloading the "(") operator for the [UnaryNumericTemporalAttribute](#) alternative.

**Parameters**

|                                    |                                    |
|------------------------------------|------------------------------------|
| <i>unaryNumericTemporalMeasure</i> | The unary numeric temporal measure |
|------------------------------------|------------------------------------|

Definition at line 82 of file TemporalNumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluate(), and multiscale::verification::UnaryNumericTemporalAttribute::temporalNumericMeasure.

**7.201.3.9 double multiscale::verification::TemporalNumericVisitor::operator() ( const BinaryNumericTemporalAttribute & binaryNumericTemporalMeasure ) const [inline]**

Overloading the "(") operator for the [BinaryNumericTemporalAttribute](#) alternative.

**Parameters**

|                                     |                                     |
|-------------------------------------|-------------------------------------|
| <i>binaryNumericTemporalMeasure</i> | The binary numeric temporal measure |
|-------------------------------------|-------------------------------------|

Definition at line 97 of file TemporalNumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluate(), multiscale::verification::BinaryNumericTemporalAttribute::firstTemporalNumericMeasure, and multiscale::verification::BinaryNumericTemporalAttribute::secondTemporalNumericMeasure.

#### 7.201.3.10 double multiscale::verification::TemporalNumericVisitor::operator() ( const UnaryStatisticalNumericAttribute & unaryStatisticalNumericAttribute ) const [inline]

Overloading the "(") operator for the [UnaryStatisticalNumericAttribute](#) alternative.

**Parameters**

|                                         |                                         |
|-----------------------------------------|-----------------------------------------|
| <i>unaryStatisticalNumericAttribute</i> | The unary statistical numeric attribute |
|-----------------------------------------|-----------------------------------------|

Definition at line 117 of file TemporalNumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), evaluateNumericMeasureCollection(), multiscale::verification::UnaryStatisticalNumericAttribute::numericMeasureCollection, multiscale::verification::UnaryStatisticalNumericAttribute::unaryStatisticalMeasure, and multiscale::verification::UnaryStatisticalMeasure::unaryStatisticalMeasureType.

#### 7.201.3.11 double multiscale::verification::TemporalNumericVisitor::operator() ( const BinaryStatisticalNumericAttribute & binaryStatisticalNumericAttribute ) const [inline]

Overloading the "(") operator for the [BinaryStatisticalNumericAttribute](#) alternative.

**Parameters**

|                                          |                                          |
|------------------------------------------|------------------------------------------|
| <i>binaryStatisticalNumericAttribute</i> | The binary statistical numeric attribute |
|------------------------------------------|------------------------------------------|

Definition at line 134 of file TemporalNumericVisitor.hpp.

References multiscale::verification::BinaryStatisticalNumericAttribute::binaryStatisticalMeasure, multiscale::verification::BinaryStatisticalMeasureAttribute::binaryStatisticalMeasureType, multiscale::verification::NumericEvaluator::evaluate(), evaluateNumericMeasureCollection(), multiscale::verification::BinaryStatisticalNumericAttribute::firstNumericMeasureCollection, and multiscale::verification::BinaryStatisticalNumericAttribute::secondNumericMeasureCollection.

#### 7.201.3.12 double multiscale::verification::TemporalNumericVisitor::operator() ( const BinaryStatisticalQuantileNumericAttribute & binaryStatisticalQuantileNumericAttribute ) const [inline]

Overloading the "(") operator for the [BinaryStatisticalQuantileNumericAttribute](#) alternative.

**Parameters**

|                                                                                                                                                           |                                                   |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| $\begin{array}{l} \textit{binary} \leftarrow \\ \textit{Statistical} \leftarrow \\ \textit{Quantile} \leftarrow \\ \textit{NumericAttribute} \end{array}$ | The binary statistical quantile numeric attribute |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|

Definition at line 157 of file TemporalNumericVisitor.hpp.

References multiscale::verification::BinaryStatisticalQuantileNumericAttribute::binaryStatisticalQuantileMeasure, multiscale::verification::NumericEvaluator::evaluate(), evaluateNumericMeasureCollection(), multiscale::verification::BinaryStatisticalQuantileNumericAttribute::numericMeasureCollection, and multiscale::verification::BinaryStatisticalQuantileNumericAttribute::parameter.

#### 7.201.4 Member Data Documentation

##### 7.201.4.1 const SpatialTemporalTrace& multiscale::verification::TemporalNumericVisitor::trace [private]

The considered spatial temporal trace

Definition at line 23 of file TemporalNumericVisitor.hpp.

Referenced by evaluateNumericMeasureCollection().

##### 7.201.4.2 const TypeSemanticsTable& multiscale::verification::TemporalNumericVisitor::typeSemanticsTable [private]

The type semantics table

Definition at line 24 of file TemporalNumericVisitor.hpp.

Referenced by evaluateNumericMeasureCollection().

The documentation for this class was generated from the following file:

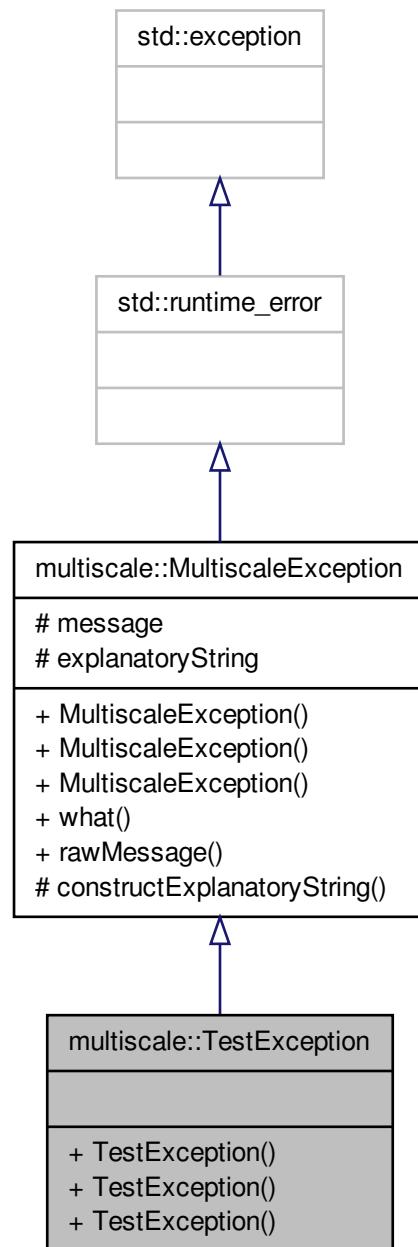
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[TemporalNumericVisitor.hpp](#)

## 7.202 multiscale::TestException Class Reference

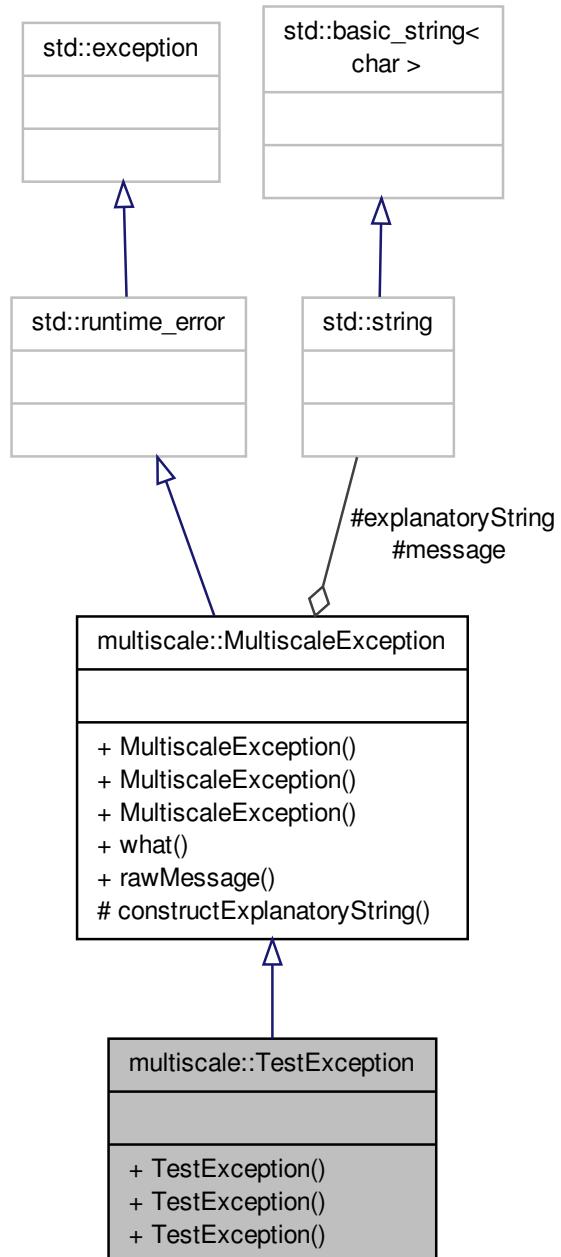
Class for representing testing exceptions.

```
#include <TestException.hpp>
```

Inheritance diagram for multiscale::TestException:



Collaboration diagram for multiscale::TestException:



## Public Member Functions

- [TestException \(\)](#)
- [TestException \(const std::string &file, int line, const std::string &msg\)](#)
- [TestException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.202.1 Detailed Description

Class for representing testing exceptions.

Definition at line 12 of file TestException.hpp.

### 7.202.2 Constructor & Destructor Documentation

#### 7.202.2.1 multiscale::TestException::TestException( ) [inline]

Definition at line 16 of file TestException.hpp.

#### 7.202.2.2 multiscale::TestException::TestException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 18 of file TestException.hpp.

#### 7.202.2.3 multiscale::TestException::TestException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 23 of file TestException.hpp.

The documentation for this class was generated from the following file:

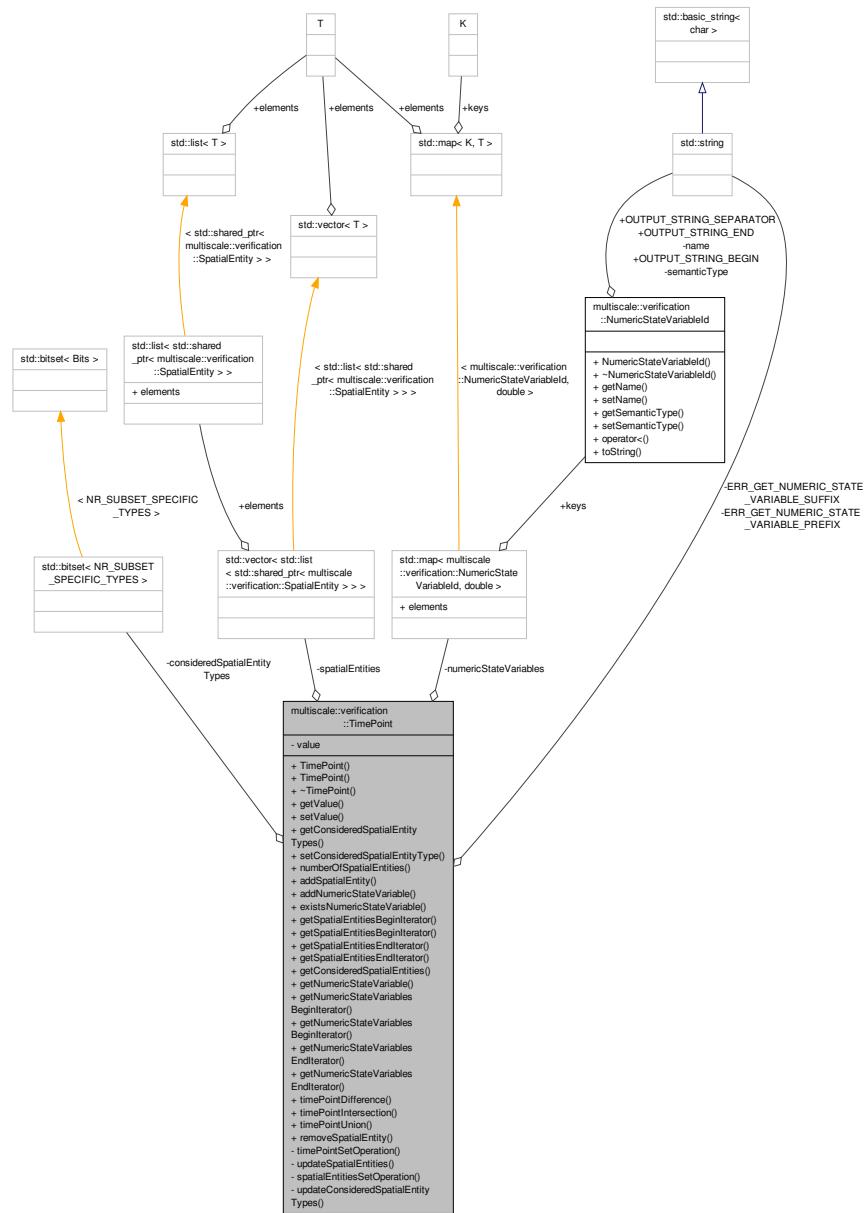
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/TestException.hpp

## 7.203 multiscale::verification::TimePoint Class Reference

Class for representing a timepoint.

```
#include <TimePoint.hpp>
```

## Collaboration diagram for multiscale::verification::TimePoint:



## Public Member Functions

- `TimePoint` (`unsigned long value=std::numeric_limits< unsigned long >::max()`)
  - `TimePoint` (`const TimePoint &timePoint`)
  - `~TimePoint ()`
  - `unsigned long getValue () const`

*Get the value of the timepoint.*

- void `setValue` (unsigned long `value`)

*Set the value of the timepoint.*

- std::bitset

< NR SUBSET SPECIFIC TYPES > getConsideredSpatialEntityTypes ()

*Get the considered spatial entity type.*

- void `setConsideredSpatialEntityType` (const `SubsetSpecificType` &consideredSpatialEntityType)  
*Set the considered spatial entity type to the given type.*
- double `numberOfSpatialEntities` () const  
*Get the number of considered spatial entities.*
- void `addSpatialEntity` (const `std::shared_ptr< SpatialEntity >` &spatialEntity, const `SubsetSpecificType` &spatialEntityType)  
*Add a spatial entity of the given type to the list of spatial entities.*
- void `addNumericStateVariable` (const `NumericStateVariableId` &id, double `value`)  
*Add a numeric state variable to the map.*
- bool `existsNumericStateVariable` (const `NumericStateVariableId` &id)  
*Check if the numeric state variable with the given id exists.*
- `std::list< std::shared_ptr< SpatialEntity > >::iterator getSpatialEntitiesBeginIterator` (const `SubsetSpecificType` &spatialEntityType)  
*Get the begin iterator for the spatial entities of the given type.*
- `std::list< std::shared_ptr< SpatialEntity > >::const_iterator getSpatialEntitiesBeginIterator` (const `SubsetSpecificType` &spatialEntityType) const  
*Get the begin iterator for the spatial entities of the given type.*
- `std::list< std::shared_ptr< SpatialEntity > >::iterator getSpatialEntitiesEndIterator` (const `SubsetSpecificType` &spatialEntityType)  
*Get the end iterator for the spatial entities of the given type.*
- `std::list< std::shared_ptr< SpatialEntity > >::const_iterator getSpatialEntitiesEndIterator` (const `SubsetSpecificType` &spatialEntityType) const  
*Get the end iterator for the spatial entities of the given type.*
- `std::vector< std::shared_ptr< SpatialEntity > > getConsideredSpatialEntities` () const  
*Get the collection of considered spatial entities.*
- double `getNumericStateVariable` (const `NumericStateVariableId` &id) const  
*Get the value of the numeric state variable with the given id.*
- `std::map< NumericStateVariableId, double >::iterator getNumericStateVariablesBeginIterator` ()  
*Get the begin iterator for the collection of numeric state variables.*
- `std::map< NumericStateVariableId, double >::const_iterator getNumericStateVariablesBeginIterator` () const  
*Get the begin iterator for the collection of numeric state variables.*
- `std::map< NumericStateVariableId, double >::iterator getNumericStateVariablesEndIterator` ()  
*Get the end iterator for the collection of numeric state variables.*
- `std::map< NumericStateVariableId, double >::const_iterator getNumericStateVariablesEndIterator` () const  
*Get the end iterator for the collection of numeric state variables.*
- void `timePointDifference` (const `TimePoint` &timePoint)  
*Compute the difference of this timepoint and the given timepoint (spatial entities only)*
- void `timePointIntersection` (const `TimePoint` &timePoint)  
*Compute the intersection of this timepoint and the given timepoint (spatial entities only)*
- void `timePointUnion` (const `TimePoint` &timePoint)  
*Compute the union of this timepoint and the given timepoint (spatial entities only)*

- std::list< std::shared\_ptr< SpatialEntity > >::iterator removeSpatialEntity (std::list< std::shared\_ptr< SpatialEntity > >::iterator &position, const SubsetSpecificType &spatialEntityType)

*Remove the spatial entity of the given type from the given position.*

## Private Member Functions

- void timePointSetOperation (const TimePoint &timePoint, const SubsetOperationType &setOperationType)  
*Compute the given set operation of this timepoint and the given timepoint considering the given set operation type.*
- void updateSpatialEntities (const TimePoint &timePoint, const SubsetOperationType &setOperationType)  
*Apply the set operation to the collection of spatial entities from this and the given timepoint.*
- std::list< std::shared\_ptr< SpatialEntity > >::spatialEntitiesSetOperation (const TimePoint &timePoint, const SubsetOperationType &setOperationType, const SubsetSpecificType &spatialEntityTypeIndex)  
*Compute the given set operation on the set of spatial entities of the given type from this and the provided timepoint.*
- void updateConsideredSpatialEntityTypes (const std::bitset< NR\_SUBSET\_SPECIFIC\_TYPES > &consideredSpatialEntityTypes, const SubsetOperationType &setOperationType)  
*Update the considered spatial entity type of this timepoint considering the given setOperationType and consideredSpatialEntityTypes.*

## Private Attributes

- unsigned long value
- std::vector< std::list< std::shared\_ptr< SpatialEntity > > > spatialEntities
- std::map< NumericStateVariableId, double > numericStateVariables
- std::bitset< NR\_SUBSET\_SPECIFIC\_TYPES > consideredSpatialEntityTypes

## Static Private Attributes

- static const std::string ERR\_GET\_NUMERIC\_STATE\_VARIABLE\_PREFIX = "The numeric state variable with the given id "
- static const std::string ERR\_GET\_NUMERIC\_STATE\_VARIABLE\_SUFFIX = " does not exist."

### 7.203.1 Detailed Description

Class for representing a timepoint.

Definition at line 24 of file TimePoint.hpp.

### 7.203.2 Constructor & Destructor Documentation

#### 7.203.2.1 TimePoint::TimePoint ( unsigned long value = std::numeric\_limits<unsigned long>::max () )

Definition at line 13 of file TimePoint.cpp.

References multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES.

7.203.2.2 TimePoint::TimePoint ( const TimePoint & *timePoint* )

Definition at line 20 of file TimePoint.cpp.

## 7.203.2.3 TimePoint::~TimePoint ( )

Definition at line 25 of file TimePoint.cpp.

## 7.203.3 Member Function Documentation

7.203.3.1 void TimePoint::addNumericStateVariable ( const NumericStateVariableId & *id*, double *value* )

Add a numeric state variable to the map.

If a numeric state variable with the same id exists then the value of the existing numeric state variable will be replaced by the provided new value.

## Parameters

|              |                                                   |
|--------------|---------------------------------------------------|
| <i>id</i>    | The id (name, type) of the numeric state variable |
| <i>value</i> | The value of the numeric state variable           |

Definition at line 74 of file TimePoint.cpp.

References numericStateVariables, and value.

Referenced by multiscale::verification::TemporalDataReader::addNumericStateVariablesToTimePoint(), and multiscale::verification::SpatialTemporalDataReader::addNumericStateVariableToTimePoint().

7.203.3.2 void TimePoint::addSpatialEntity ( const std::shared\_ptr< SpatialEntity > & *spatialEntity*, const SubsetSpecificType & *spatialEntityType* )

Add a spatial entity of the given type to the list of spatial entities.

## Parameters

|                          |                                |
|--------------------------|--------------------------------|
| <i>spatialEntity</i>     | The spatial entity             |
| <i>spatialEntityType</i> | The type of the spatial entity |

Definition at line 64 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::SpatialTemporalDataReader::addSpatialEntityToTimePoint(), and multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues().

7.203.3.3 bool TimePoint::existsNumericStateVariable ( const NumericStateVariableId & *id* )

Check if the numeric state variable with the given id exists.

## Parameters

|           |                                      |
|-----------|--------------------------------------|
| <i>id</i> | The id of the numeric state variable |
|-----------|--------------------------------------|

Definition at line 79 of file TimePoint.cpp.

References numericStateVariables.

### 7.203.3.4 `std::vector< std::shared_ptr< SpatialEntity > > TimePoint::getConsideredSpatialEntities( ) const`

Get the collection of considered spatial entities.

Definition at line 128 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES, and spatialEntities.

Referenced by multiscale::verification::TimePointEvaluator::getSpatialMeasureValues().

### 7.203.3.5 `std::bitset< NR_SUBSET_SPECIFIC_TYPES > TimePoint::getConsideredSpatialEntityTypes( )`

Get the considered spatial entity type.

Definition at line 38 of file TimePoint.cpp.

References consideredSpatialEntityTypes.

Referenced by multiscale::verification::ConstraintVisitor::evaluateSpatialMeasureConstraint(), and multiscale::verification::ConstraintVisitor::evaluateTypeConstraint().

### 7.203.3.6 `double TimePoint::getNumericStateVariable( const NumericStateVariableId & id ) const`

Get the value of the numeric state variable with the given id.

Get the value of the numeric state variable with the given id if it exists and throw an exception otherwise

**Parameters**

|           |                                      |
|-----------|--------------------------------------|
| <i>id</i> | The id of the numeric state variable |
|-----------|--------------------------------------|

Definition at line 142 of file TimePoint.cpp.

References ERR\_GET\_NUMERIC\_STATE\_VARIABLE\_PREFIX, ERR\_GET\_NUMERIC\_STATE\_VARIABLE\_SUFFIX, MS\_throw\_detailed, and numericStateVariables.

Referenced by multiscale::verification::TemporalNumericVisitor::operator()(), and multiscale::verification::NumericVisitor::operator()().

### 7.203.3.7 `std::map< NumericStateVariableId, double >::iterator TimePoint::getNumericStateVariablesBeginIterator( )`

Get the begin iterator for the collection of numeric state variables.

Definition at line 154 of file TimePoint.cpp.

References numericStateVariables.

Referenced by printTimePoint().

### 7.203.3.8 `std::map< NumericStateVariableId, double >::const_iterator TimePoint::getNumericStateVariablesBeginIterator( ) const`

Get the begin iterator for the collection of numeric state variables.

Definition at line 159 of file TimePoint.cpp.

References numericStateVariables.

### 7.203.3.9 `std::map< NumericStateVariableId, double >::iterator TimePoint::getNumericStateVariablesEndIterator( )`

Get the end iterator for the collection of numeric state variables.

Definition at line 164 of file TimePoint.cpp.

References numericStateVariables.

Referenced by printTimePoint().

**7.203.3.10 std::map< NumericStateVariableId, double >::const\_iterator TimePoint::getNumericStateVariablesEndIterator ( ) const**

Get the end iterator for the collection of numeric state variables.

Definition at line 169 of file TimePoint.cpp.

References numericStateVariables.

**7.203.3.11 std::list< std::shared\_ptr< SpatialEntity > >::iterator TimePoint::getSpatialEntitiesBeginIterator ( const SubsetSpecificType & spatialEntityType )**

Get the begin iterator for the spatial entities of the given type.

Return the spatial entities begin iterator if the considered spatial entity type is of the given type. Otherwise return the spatial entities end iterator.

#### Parameters

|                          |                                  |
|--------------------------|----------------------------------|
| <i>spatialEntityType</i> | The type of the spatial entities |
|--------------------------|----------------------------------|

Definition at line 84 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringEqualComparator(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringNonEqualComparator(), multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues(), printSpatialEntities(), and spatialEntitiesSetOperation().

**7.203.3.12 std::list< std::shared\_ptr< SpatialEntity > >::const\_iterator TimePoint::getSpatialEntitiesBeginIterator ( const SubsetSpecificType & spatialEntityType ) const**

Get the begin iterator for the spatial entities of the given type.

Return the spatial entities begin iterator if the considered spatial entity type is of the given type. Otherwise return the spatial entities end iterator.

#### Parameters

|                          |                                  |
|--------------------------|----------------------------------|
| <i>spatialEntityType</i> | The type of the spatial entities |
|--------------------------|----------------------------------|

Definition at line 97 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

**7.203.3.13 std::list< std::shared\_ptr< SpatialEntity > >::iterator TimePoint::getSpatialEntitiesEndIterator ( const SubsetSpecificType & spatialEntityType )**

Get the end iterator for the spatial entities of the given type.

**Parameters**

|                          |                                  |
|--------------------------|----------------------------------|
| <i>spatialEntityType</i> | The type of the spatial entities |
|--------------------------|----------------------------------|

Definition at line 110 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringEqualComparator(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringNonEqualComparator(), printSpatialEntities(), and spatialEntitiesSetOperation().

#### 7.203.3.14 std::list< std::shared\_ptr< SpatialEntity > >::const\_iterator TimePoint::getSpatialEntitiesEndIterator ( const SubsetSpecificType & *spatialEntityType* ) const

Get the end iterator for the spatial entities of the given type.

**Parameters**

|                          |                                  |
|--------------------------|----------------------------------|
| <i>spatialEntityType</i> | The type of the spatial entities |
|--------------------------|----------------------------------|

Definition at line 119 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

#### 7.203.3.15 unsigned long TimePoint::getValue ( ) const

Get the value of the timepoint.

Definition at line 28 of file TimePoint.cpp.

References value.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateChangeLhsTemporalNumericMeasure(), printTimePoint(), multiscale::verification::SpatialTemporalTrace::updateLastTimePointValue(), and multiscale::verification::SpatialTemporalTrace::validateTimePointValue().

#### 7.203.3.16 double TimePoint::numberOfSpatialEntities ( ) const

Get the number of considered spatial entities.

Definition at line 51 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES, and spatialEntities.

#### 7.203.3.17 std::list< std::shared\_ptr< SpatialEntity > >::iterator TimePoint::removeSpatialEntity ( std::list< std::shared\_ptr< SpatialEntity > >::iterator & *position*, const SubsetSpecificType & *spatialEntityType* )

Remove the spatial entity of the given type from the given position.

**Parameters**

|                          |                                                  |
|--------------------------|--------------------------------------------------|
| <i>position</i>          | The position of the spatial entity to be removed |
| <i>spatialEntityType</i> | The type of the spatial entity                   |

Definition at line 189 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringEqualComparator(), multiscale::verification::ConstraintVisitor::filterSpatialEntitiesWrtTypeConsideringNonEqualComparator(), and multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues().

#### 7.203.3.18 void TimePoint::setConsideredSpatialEntityType ( const SubsetSpecificType & *consideredSpatialEntityType* )

Set the considered spatial entity type to the given type.

##### Parameters

|                                    |                                             |
|------------------------------------|---------------------------------------------|
| <i>consideredSpatialEntityType</i> | The considered type of the spatial entities |
|------------------------------------|---------------------------------------------|

Definition at line 43 of file TimePoint.cpp.

References multiscale::verification::subsetsspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, and multiscale::verification::subsetsspecific::validateSubsetSpecificType().

Referenced by multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithClusterednessValues(), and multiscale::verification::SubsetVisitor::setTimePointConsideredSpatialEntityType().

#### 7.203.3.19 void TimePoint::setValue ( unsigned long *value* )

Set the value of the timepoint.

##### Parameters

|              |                            |
|--------------|----------------------------|
| <i>value</i> | The value of the timepoint |
|--------------|----------------------------|

Definition at line 33 of file TimePoint.cpp.

References value.

Referenced by multiscale::verification::TemporalDataReader::setTimePointValue(), multiscale::verification::SpatialTemporalDataReader::setTimePointValue(), and multiscale::verification::SpatialTemporalTrace::updateLastTimePointValue().

#### 7.203.3.20 std::list< std::shared\_ptr< SpatialEntity > > TimePoint::spatialEntitiesSetOperation ( const TimePoint & *timePoint*, const SubsetOperationType & *setOperationType*, const SubsetSpecificType & *spatialEntityTypeIndex* ) [private]

Compute the given set operation on the set of spatial entities of the given type from this and the provided timepoint.

##### Parameters

|                               |                                          |
|-------------------------------|------------------------------------------|
| <i>timePoint</i>              | The given timepoint                      |
| <i>setOperationType</i>       | The considered set operation type        |
| <i>spatialEntityTypeIndex</i> | The considered spatial entity type index |

Definition at line 212 of file TimePoint.cpp.

References multiscale::verification::Difference, getSpatialEntitiesBeginIterator(), getSpatialEntitiesEndIterator(), multiscale::verification::Intersection, and multiscale::verification::Union.

Referenced by updateSpatialEntities().

### 7.203.3.21 void TimePoint::timePointDifference ( const TimePoint & *timePoint* )

Compute the difference of this timepoint and the given timepoint (spatial entities only)

Compute the difference of this timepoint and the given timepoint by taking into account the value of considered  
SpatialEntityType

Spatial entities belonging to the first and not to the second timepoint will be included in the resulting timepoint.

The consideredSpatialEntityType of the resulting timepoint will be the consideredSpatialEntityType of this timepoint.

#### Parameters

|                  |                     |
|------------------|---------------------|
| <i>timePoint</i> | The given timepoint |
|------------------|---------------------|

Definition at line 174 of file TimePoint.cpp.

References multiscale::verification::Difference, and timePointSetOperation().

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::  
ConstraintVisitor::operator()().

### 7.203.3.22 void TimePoint::timePointIntersection ( const TimePoint & *timePoint* )

Compute the intersection of this timepoint and the given timepoint (spatial entities only)

Compute the intersection of this timepoint and the given timepoint by taking into account the value of considered  
SpatialEntityType

Spatial entities belonging both to the first and the second timepoint will be included in the resulting timepoint.

The consideredSpatialEntityType of the resulting timepoint will be the intersection of the timepoints' considered  
SpatialEntityTypes.

#### Parameters

|                  |                     |
|------------------|---------------------|
| <i>timePoint</i> | The given timepoint |
|------------------|---------------------|

Definition at line 179 of file TimePoint.cpp.

References multiscale::verification::Intersection, and timePointSetOperation().

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::  
ConstraintVisitor::operator()().

### 7.203.3.23 void TimePoint::timePointSetOperation ( const TimePoint & *timePoint*, const SubsetOperationType & *setOperationType* ) [private]

Compute the given set operation of this timepoint and the given timepoint considering the given set operation type.

#### Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>timePoint</i>        | The given timepoint               |
| <i>setOperationType</i> | The considered set operation type |

Definition at line 197 of file TimePoint.cpp.

References consideredSpatialEntityTypes, updateConsideredSpatialEntityTypes(), and updateSpatialEntities().

Referenced by timePointDifference(), timePointIntersection(), and timePointUnion().

### 7.203.3.24 void TimePoint::timePointUnion ( const TimePoint & *timePoint* )

Compute the union of this timepoint and the given timepoint (spatial entities only)

Compute the union of this timepoint and the given timepoint by taking into account the value of consideredSpatialEntityTypes.

Spatial entities belonging either to the first or the second timepoint will be included in the resulting timepoint.

The consideredSpatialEntityType of the resulting timepoint will be the union of the timepoints' consideredSpatialEntityTypes.

#### Parameters

|                  |                     |
|------------------|---------------------|
| <i>timePoint</i> | The given timepoint |
|------------------|---------------------|

Definition at line 184 of file TimePoint.cpp.

References timePointSetOperation(), and multiscale::verification::Union.

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::ConstraintVisitor::operator()().

**7.203.3.25 void TimePoint::updateConsideredSpatialEntityTypes ( const std::bitset< NR\_SUBSET\_SPECIFIC\_TYPES > & *consideredSpatialEntityTypes*, const SubsetOperationType & *setOperationType* ) [private]**

Update the considered spatial entity type of this timepoint considering the given setOperationType and consideredSpatialEntityTypes.

Definition at line 246 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::Difference, multiscale::verification::Intersection, and multiscale::verification::Union.

Referenced by timePointSetOperation().

**7.203.3.26 void TimePoint::updateSpatialEntities ( const TimePoint & *timePoint*, const SubsetOperationType & *setOperationType* ) [private]**

Apply the set operation to the collection of spatial entities from this and the given timepoint.

#### Parameters

|                         |                                   |
|-------------------------|-----------------------------------|
| <i>timePoint</i>        | The given timepoint               |
| <i>setOperationType</i> | The considered set operation type |

Definition at line 203 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES, spatialEntities, and spatialEntitiesSetOperation().

Referenced by timePointSetOperation().

## 7.203.4 Member Data Documentation

**7.203.4.1 std::bitset<NR\_SUBSET\_SPECIFIC\_TYPES> multiscale::verification::TimePoint::consideredSpatialEntityTypes [private]**

The collection of bits recording the considered spatial entity types. The i-th bit corresponds to the i-th SubsetSpecificType enum value. If the bit is set true then the corresponding subset specific type is considered. Otherwise it is not.

Definition at line 49 of file TimePoint.hpp.

Referenced by getConsideredSpatialEntities(), getConsideredSpatialEntityTypes(), getSpatialEntitiesBeginIterator(), numberofSpatialEntities(), setConsideredSpatialEntityType(), timePointSetOperation(), and updateConsideredSpatialEntityTypes().

7.203.4.2 `const std::string TimePoint::ERR_GET_NUMERIC_STATE_VARIABLE_PREFIX = "The numeric state variable with the given id "` [static], [private]

Definition at line 265 of file TimePoint.hpp.

Referenced by `getNumericStateVariable()`.

7.203.4.3 `const std::string TimePoint::ERR_GET_NUMERIC_STATE_VARIABLE_SUFFIX = " does not exist."` [static], [private]

Definition at line 266 of file TimePoint.hpp.

Referenced by `getNumericStateVariable()`.

7.203.4.4 `std::map<NumericStateVariableId, double> multiscale::verification::TimePoint::numericStateVariables` [private]

The associative map for storing numeric state variables

Definition at line 46 of file TimePoint.hpp.

Referenced by `addNumericStateVariable()`, `existsNumericStateVariable()`, `getNumericStateVariable()`, `getNumericStateVariablesBeginIterator()`, and `getNumericStateVariablesEndIterator()`.

7.203.4.5 `std::vector<std::list<std::shared_ptr<SpatialEntity>>> multiscale::verification::TimePoint::spatialEntities` [private]

The meta-list of spatial entities smart pointers. The i-th spatial entities list in the meta-list corresponds to the i-th `SubsetSpecificType` enumeration value

Definition at line 40 of file TimePoint.hpp.

Referenced by `addSpatialEntity()`, `getConsideredSpatialEntities()`, `getSpatialEntitiesBeginIterator()`, `getSpatialEntitiesEndIterator()`, `numberOfSpatialEntities()`, `removeSpatialEntity()`, and `updateSpatialEntities()`.

7.203.4.6 `unsigned long multiscale::verification::TimePoint::value` [private]

The value of the timepoint within a simulation/experiment

Definition at line 28 of file TimePoint.hpp.

Referenced by `addNumericStateVariable()`, `getValue()`, and `setValue()`.

The documentation for this class was generated from the following files:

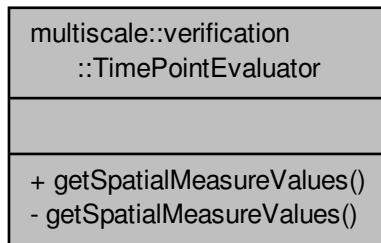
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[TimePoint.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[TimePoint.cpp](#)

## 7.204 multiscale::verification::TimePointEvaluator Class Reference

Class used to evaluate timepoints.

```
#include <TimePointEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::TimePointEvaluator:



## Static Public Member Functions

- static std::vector< double > [getSpatialMeasureValues](#) (const [TimePoint](#) &timePoint, const [SpatialMeasureType](#) &spatialMeasure)

*Return the spatial measure values for all considered spatial entities in the given timepoint.*

## Static Private Member Functions

- static void [getSpatialMeasureValues](#) (const std::vector< std::shared\_ptr< [SpatialEntity](#) >> &consideredSpatialEntities, std::vector< double > &spatialMeasureValues, const [SpatialMeasureType](#) &spatialMeasure)

*Return the spatial measure values for all considered spatial entities in the given timepoint.*

### 7.204.1 Detailed Description

Class used to evaluate timepoints.

Definition at line 12 of file TimePointEvaluator.hpp.

### 7.204.2 Member Function Documentation

#### 7.204.2.1 static std::vector<double> multiscale::verification::TimePointEvaluator::getSpatialMeasureValues ( const TimePoint & timePoint, const SpatialMeasureType & spatialMeasure ) [inline], [static]

Return the spatial measure values for all considered spatial entities in the given timepoint.

##### Parameters

|                             |                                |
|-----------------------------|--------------------------------|
| <code>timePoint</code>      | The considered timepoint       |
| <code>spatialMeasure</code> | The considered spatial measure |

Definition at line 21 of file TimePointEvaluator.hpp.

References multiscale::verification::TimePoint::getConsideredSpatialEntities().

Referenced by multiscale::verification::NumericMeasureCollectionEvaluator::evaluateSpatialMeasureCollection().

```
7.204.2.2 static void multiscale::verification::TimePointEvaluator::getSpatialMeasureValues (const std::vector<
std::shared_ptr< SpatialEntity >> & consideredSpatialEntities, std::vector< double > & spatialMeasureValues,
const SpatialMeasureType & spatialMeasure) [inline], [static], [private]
```

Return the spatial measure values for all considered spatial entities in the given timepoint.

**Parameters**

|                                  |                                                        |
|----------------------------------|--------------------------------------------------------|
| <i>consideredSpatialEntities</i> | The considered spatial entities                        |
| <i>spatialMeasureValues</i>      | The collection of values for the given spatial measure |
| <i>spatialMeasure</i>            | The considered spatial measure                         |

Definition at line 41 of file TimePointEvaluator.hpp.

References multiscale::verification::SpatialMeasureEvaluator::evaluate().

The documentation for this class was generated from the following file:

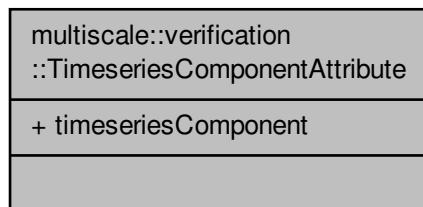
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[TimePointEvaluator.hpp](#)

## 7.205 multiscale::verification::TimeseriesComponentAttribute Class Reference

Class for representing a timeseries component attribute.

```
#include <TimeseriesComponentAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TimeseriesComponentAttribute:



### Public Attributes

- [TimeseriesComponentType timeseriesComponent](#)

#### 7.205.1 Detailed Description

Class for representing a timeseries component attribute.

Definition at line 22 of file TimeseriesComponentAttribute.hpp.

#### 7.205.2 Member Data Documentation

##### 7.205.2.1 TimeseriesComponentType multiscale::verification::TimeseriesComponentAttribute::timeseriesComponent

The timeseries component

Definition at line 26 of file TimeseriesComponentAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::evaluateTimeseriesComponent().

The documentation for this class was generated from the following file:

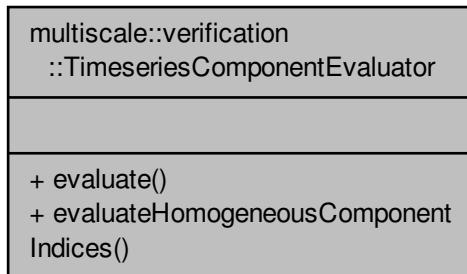
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesComponentAttribute.hpp

## 7.206 multiscale::verification::TimeseriesComponentEvaluator Class Reference

Class for evaluating timeseries components.

```
#include <TimeseriesComponentEvaluator.hpp>
```

Collaboration diagram for multiscale::verification::TimeseriesComponentEvaluator:



### Classes

- class [HomogeneousComponentEvaluator](#)
- class [UniformHomogeneousComponentEvaluator](#)

### Static Public Member Functions

- static std::vector< std::size\_t > **evaluate** (const [HomogeneousTimeseriesComponentAttribute](#) &homogeneousTimeseriesComponent, const std::vector< double > &values)

*Evaluate the homogeneous timeseries component considering the given collection of values.*

- template<typename Relation , template< typename > class HomogeneousTimeseriesComponentEvaluator> static std::vector< std::size\_t > **evaluateHomogeneousComponentIndices** (const std::vector< double > &values, const Relation &relation, const HomogeneousTimeseriesComponentEvaluator< Relation > &evaluator)

*Compute the set of (start, end) indices pointing to nonuniform homogeneous timeseries components.*

### 7.206.1 Detailed Description

Class for evaluating timeseries components.

Definition at line 15 of file TimeseriesComponentEvaluator.hpp.

## 7.206.2 Member Function Documentation

7.206.2.1 static std::vector<std::size\_t> multiscale::verification::TimeseriesComponentEvaluator::evaluate ( const HomogeneousTimeseriesComponentAttribute & *homogeneousTimeseriesComponent*, const std::vector<double> & *values* ) [inline], [static]

Evaluate the homogeneous timeseries component considering the given collection of values.

**Parameters**

|                                       |                                            |
|---------------------------------------|--------------------------------------------|
| <i>homogeneousTimeseriesComponent</i> | The given homogeneous timeseries component |
| <i>values</i>                         | The given collection of values             |

Definition at line 25 of file TimeseriesComponentEvaluator.hpp.

References multiscale::verification::Ascent, multiscale::verification::Descent, multiscale::ERR\_UNDEFED\_ENUM\_VALUE, evaluateHomogeneousComponentIndices(), multiscale::verification::HomogeneousTimeseriesComponentAttribute::homogeneousTimeseriesComponent, MS\_throw, multiscale::verification::Plateau, multiscale::verification::UniformAscent, and multiscale::verification::UniformDescent.

Referenced by multiscale::verification::TimeseriesComponentVisitor::operator()(), and multiscale::verification::NumericMeasureCollectionVisitor::operator()().

```
7.206.2.2 template<typename Relation , template< typename > class HomogeneousTimeseriesComponentEvaluator>
static std::vector<std::size_t> multiscale::verification::TimeseriesComponentEvaluator::evaluate<->
HomogeneousComponentIndices (const std::vector< double > & values, const Relation & relation, const
HomogeneousTimeseriesComponentEvaluator< Relation > & evaluator) [inline], [static]
```

Compute the set of (start, end) indices pointing to nonuniform homogeneous timeseries components.

The value of relation depending on the considered homogeneous component type is:

- (uniform) ascent: "<"
- (uniform) descent: ">"
- plateau: "="

**Parameters**

|                  |                                |
|------------------|--------------------------------|
| <i>values</i>    | The collection of given values |
| <i>relation</i>  | The considered relation        |
| <i>evaluator</i> | The considered evaluator       |

Definition at line 87 of file TimeseriesComponentEvaluator.hpp.

Referenced by evaluate().

The documentation for this class was generated from the following file:

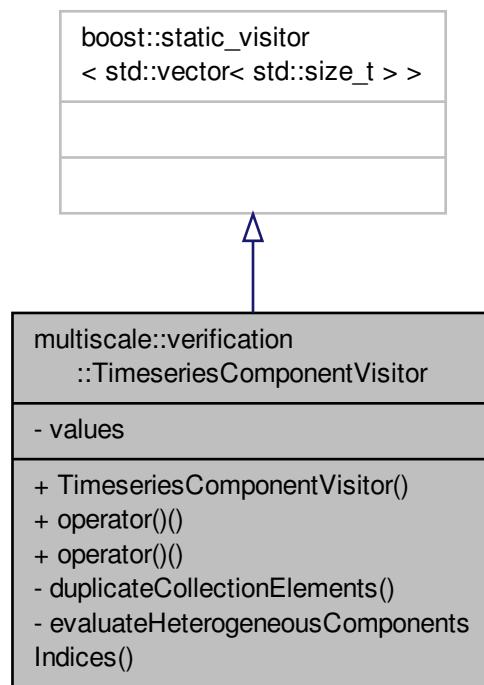
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp

## 7.207 multiscale::verification::TimeseriesComponentVisitor Class Reference

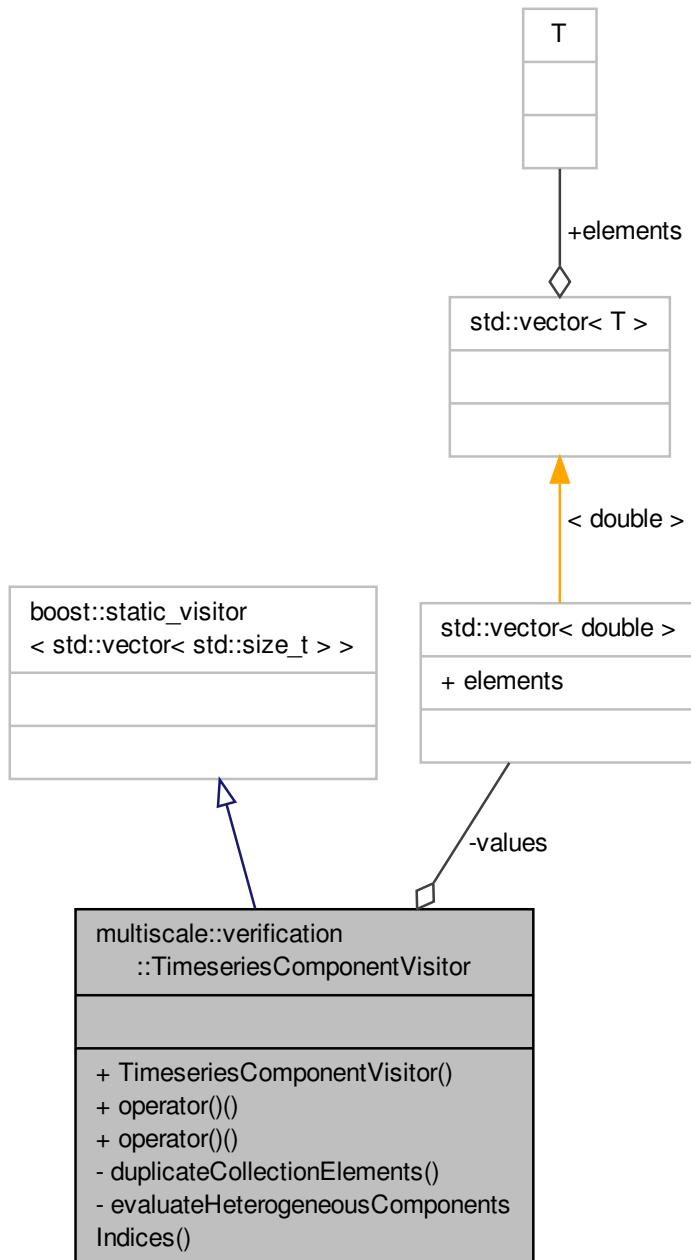
Class for evaluating timeseries components.

```
#include <TimeseriesComponentVisitor.hpp>
```

Inheritance diagram for multiscale::verification::TimeseriesComponentVisitor:



Collaboration diagram for multiscale::verification::TimeseriesComponentVisitor:



## Public Member Functions

- `TimeseriesComponentVisitor (const std::vector< double > &values)`
- `std::vector< std::size_t > operator() (const HeterogeneousTimeseriesComponentAttribute &heterogeneousTimeseriesComponent) const`  
*Overloading the "()" operator for the `HeterogeneousTimeseriesComponentAttribute` alternative.*
- `std::vector< std::size_t > operator() (const HomogeneousTimeseriesComponentAttribute &homogeneousTimeseriesComponent) const`

Overloading the "()" operator for the [HomogeneousTimeseriesComponentAttribute](#) alternative.

## Private Member Functions

- `std::vector< std::size_t > duplicateCollectionElements (const std::vector< std::size_t > &collection) const`  
*Duplicate each element in the given indices collection.*
- `template<typename Relation >`  
`std::vector< std::size_t > evaluateHeterogeneousComponentsIndices (const std::vector< double > &values, const Relation &relation) const`  
*Compute the set of indices pointing to the heterogeneous components in the given values collection.*

## Private Attributes

- `const std::vector< double > & values`

### 7.207.1 Detailed Description

Class for evaluating timeseries components.

The output values of the visitor is a collection of indices pointing to the start and end positions of the timeseries components identified in the given collection of values. In case of heterogeneous timeseries components the start position = end position, while in the case of homogeneous timeseries components start position != end position.

Definition at line 24 of file TimeseriesComponentVisitor.hpp.

### 7.207.2 Constructor & Destructor Documentation

#### 7.207.2.1 multiscale::verification::TimeseriesComponentVisitor::TimeseriesComponentVisitor ( const std::vector< double > & values ) [inline]

Definition at line 32 of file TimeseriesComponentVisitor.hpp.

### 7.207.3 Member Function Documentation

#### 7.207.3.1 std::vector<std::size\_t> multiscale::verification::TimeseriesComponentVisitor::duplicateCollectionElements ( const std::vector< std::size\_t > & collection ) const [inline], [private]

Duplicate each element in the given indices collection.

Definition at line 77 of file TimeseriesComponentVisitor.hpp.

Referenced by `operator()`.

#### 7.207.3.2 template<typename Relation > std::vector<std::size\_t> multiscale::verification::TimeseriesComponentVisitor::evaluateHeterogeneousComponentsIndices ( const std::vector< double > & values, const Relation & relation ) const [inline], [private]

Compute the set of indices pointing to the heterogeneous components in the given values collection.

A value located at index  $i$  in the collection,  $0 < i < (\text{values.size() - 1})$ , is a heterogeneous component if and only if `relation(values[i - 1], values[i])` and `!relation(values[i], values[i + 1])` and `!equal_to(values[i], values[i + 1])`

**Parameters**

|                 |                                |
|-----------------|--------------------------------|
| <i>values</i>   | The collection of given values |
| <i>relation</i> | The considered relation        |

Definition at line 105 of file `TimeseriesComponentVisitor.hpp`.

Referenced by `operator()`.

**7.207.3.3** `std::vector<std::size_t> multiscale::verification::TimeseriesComponentVisitor::operator() ( const HeterogeneousTimeseriesComponentAttribute & heterogeneousTimeseriesComponent ) const [inline]`

Overloading the `"()"` operator for the `HeterogeneousTimeseriesComponentAttribute` alternative.

**Parameters**

|                                         |                                        |
|-----------------------------------------|----------------------------------------|
| <i>heterogeneousTimeseriesComponent</i> | The heterogeneous timeseries component |
|-----------------------------------------|----------------------------------------|

Definition at line 39 of file `TimeseriesComponentVisitor.hpp`.

References `duplicateCollectionElements()`, `multiscale::ERR_UNDEFINED_ENUM_VALUE`, `evaluateHeterogeneousComponentsIndices()`, `multiscale::verification::HeterogeneousTimeseriesComponentAttribute::heterogeneousTimeseriesComponent`, `MS_throw`, `multiscale::verification::Peak`, and `multiscale::verification::Valley`.

**7.207.3.4** `std::vector<std::size_t> multiscale::verification::TimeseriesComponentVisitor::operator() ( const HomogeneousTimeseriesComponentAttribute & homogeneousTimeseriesComponent ) const [inline]`

Overloading the `"()"` operator for the `HomogeneousTimeseriesComponentAttribute` alternative.

**Parameters**

|                                       |                                      |
|---------------------------------------|--------------------------------------|
| <i>homogeneousTimeseriesComponent</i> | The homogeneous timeseries component |
|---------------------------------------|--------------------------------------|

Definition at line 66 of file `TimeseriesComponentVisitor.hpp`.

References `multiscale::verification::TimeseriesComponentEvaluator::evaluate()`.

## 7.207.4 Member Data Documentation

**7.207.4.1** `const std::vector<double> & multiscale::verification::TimeseriesComponentVisitor::values [private]`

The collection of considered values

Definition at line 28 of file `TimeseriesComponentVisitor.hpp`.

The documentation for this class was generated from the following file:

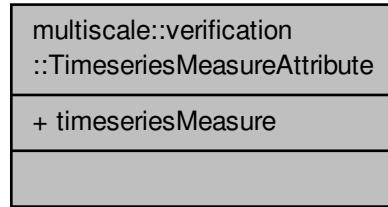
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/`TimeseriesComponentVisitor.hpp`

## 7.208 multiscale::verification::TimeseriesMeasureAttribute Class Reference

Class for representing a timeseries measure attribute.

```
#include <TimeseriesMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TimeseriesMeasureAttribute:



## Public Attributes

- [TimeseriesMeasureType timeseriesMeasure](#)

### 7.208.1 Detailed Description

Class for representing a timeseries measure attribute.

Definition at line 28 of file TimeseriesMeasureAttribute.hpp.

### 7.208.2 Member Data Documentation

#### 7.208.2.1 TimeseriesMeasureType multiscale::verification::TimeseriesMeasureAttribute::timeseriesMeasure

The timeseries measure

Definition at line 32 of file TimeseriesMeasureAttribute.hpp.

Referenced by multiscale::verification::NumericMeasureCollectionVisitor::operator()().

The documentation for this class was generated from the following file:

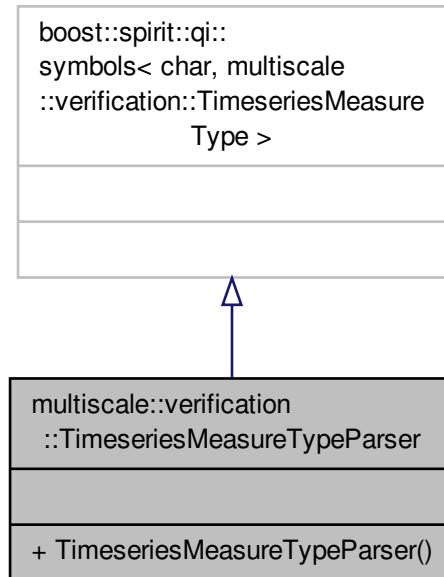
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesMeasureAttribute.hpp](#)

## 7.209 multiscale::verification::TimeseriesMeasureTypeParser Struct Reference

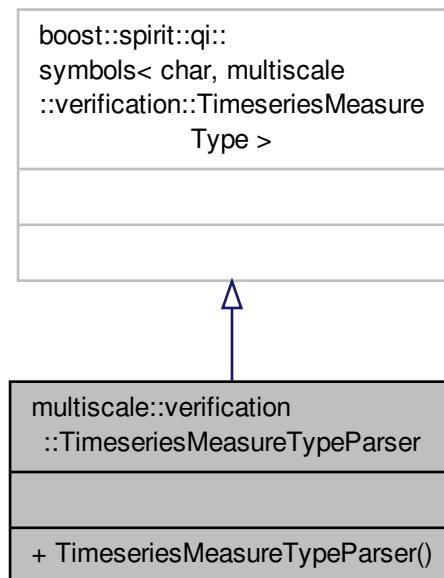
Symbol table and parser for the timeseries measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::TimeseriesMeasureTypeParser:



Collaboration diagram for multiscale::verification::TimeseriesMeasureTypeParser:



## Public Member Functions

- [TimeseriesMeasureTypeParser \(\)](#)

### 7.209.1 Detailed Description

Symbol table and parser for the timeseries measure type.

Definition at line 183 of file [SymbolTables.hpp](#).

### 7.209.2 Constructor & Destructor Documentation

#### 7.209.2.1 multiscale::verification::TimeseriesMeasureTypeParser::TimeseriesMeasureTypeParser( ) [inline]

Definition at line 186 of file [SymbolTables.hpp](#).

References [multiscale::verification::EnteringTime](#), and [multiscale::verification::EnteringValue](#).

The documentation for this struct was generated from the following file:

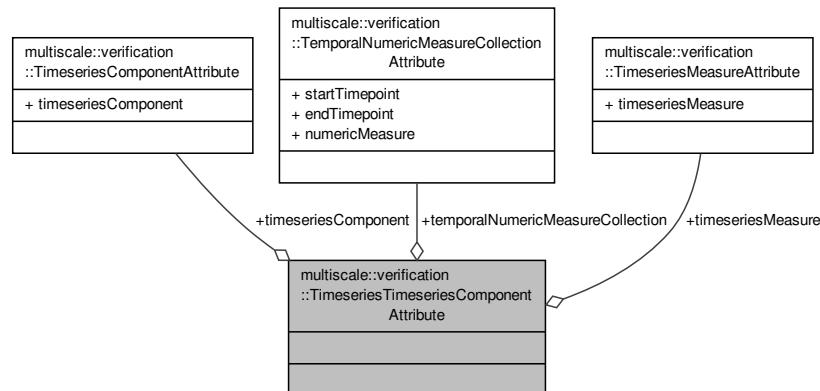
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.210 multiscale::verification::TimeseriesTimeseriesComponentAttribute Class Reference

Class for representing a timeseries timeseries component attribute.

```
#include <TimeseriesTimeseriesComponentAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TimeseriesTimeseriesComponentAttribute:



## Public Attributes

- [TimeseriesMeasureAttribute timeseriesMeasure](#)
- [TimeseriesComponentAttribute timeseriesComponent](#)
- [TemporalNumericMeasureCollectionAttribute temporalNumericMeasureCollection](#)

### 7.210.1 Detailed Description

Class for representing a timeseries timeseries component attribute.

Definition at line 17 of file `TimeseriesTimeseriesComponentAttribute.hpp`.

### 7.210.2 Member Data Documentation

#### 7.210.2.1 `TemporalNumericMeasureCollectionAttribute multiscale::verification::TimeseriesTimeseriesComponentAttribute::temporalNumericMeasureCollection`

The temporal numeric collection

Definition at line 26 of file `TimeseriesTimeseriesComponentAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

#### 7.210.2.2 `TimeseriesComponentAttribute multiscale::verification::TimeseriesTimeseriesComponentAttribute::timeseriesComponent`

The timeseries component

Definition at line 24 of file `TimeseriesTimeseriesComponentAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

#### 7.210.2.3 `TimeseriesMeasureAttribute multiscale::verification::TimeseriesTimeseriesComponentAttribute::timeseriesMeasure`

The timeseries measure

Definition at line 22 of file `TimeseriesTimeseriesComponentAttribute.hpp`.

Referenced by `multiscale::verification::NumericMeasureCollectionVisitor::operator()()`.

The documentation for this class was generated from the following file:

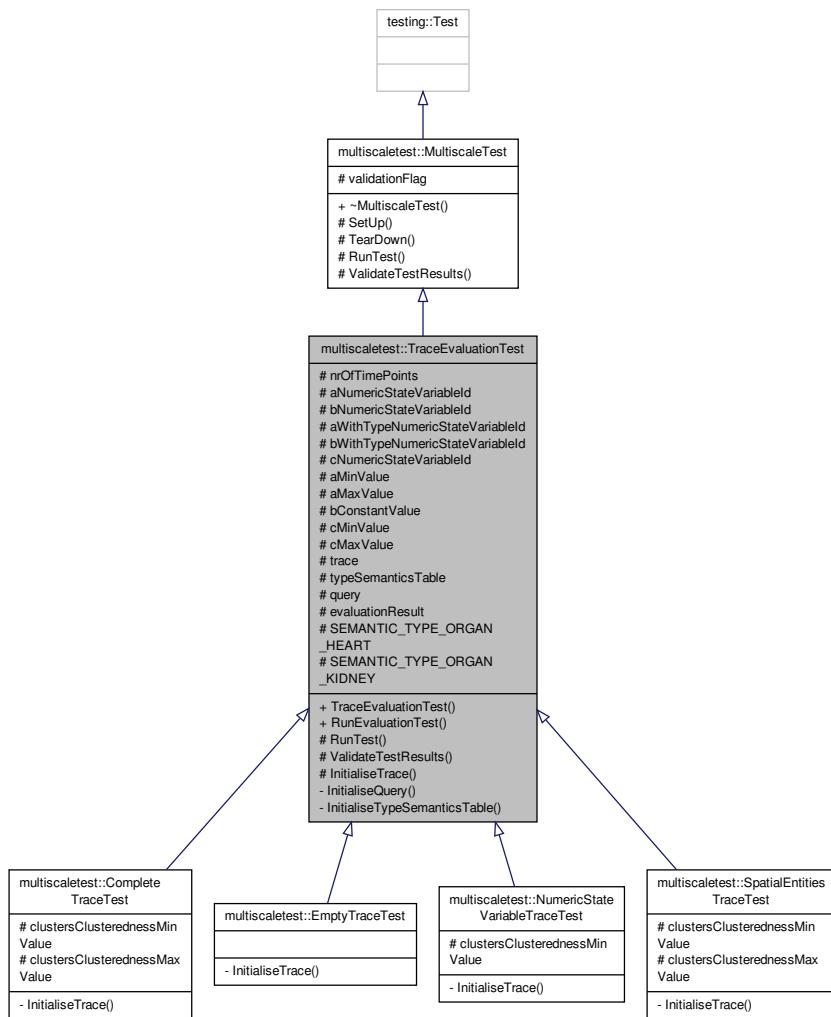
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesTimeseriesComponentAttribute.hpp`

## 7.211 `multiscaletest::TraceEvaluationTest` Class Reference

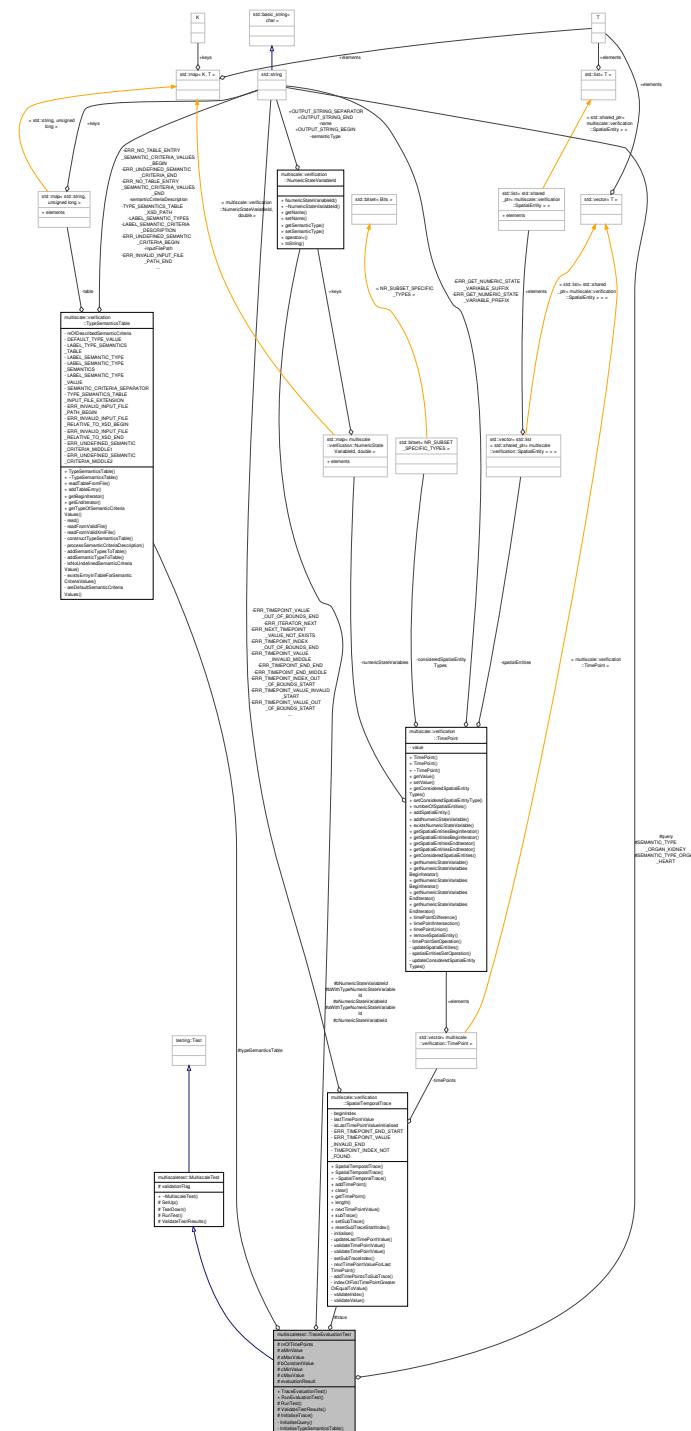
Class for testing evaluation of traces.

```
#include <TraceEvaluationTest.hpp>
```

Inheritance diagram for multiscaletest::TraceEvaluationTest:



## Collaboration diagram for multiscaletest::TraceEvaluationTest:



## Public Member Functions

- `TraceEvaluationTest ()`
  - `bool RunEvaluationTest (const std::string &query)`

*Run the test with the given string.*

## Protected Member Functions

- virtual void [RunTest \(\) override](#)  
*Run the test.*
- virtual void [ValidateTestResults \(\) override](#)  
*Validate the results of the test.*
- virtual void [InitialiseTrace \(\)=0](#)  
*Initialise the trace.*

## Protected Attributes

- std::size\_t [nrOfTimePoints](#)
- NumericStateVariableId [aNumericStateVariableId](#)
- NumericStateVariableId [bNumericStateVariableId](#)
- NumericStateVariableId [aWithTypeNumericStateVariableId](#)
- NumericStateVariableId [bWithTypeNumericStateVariableId](#)
- NumericStateVariableId [cNumericStateVariableId](#)
- double [aMinValue](#)
- double [aMaxValue](#)
- double [bConstantValue](#)
- double [cMinValue](#)
- double [cMaxValue](#)
- mv::SpatialTemporalTrace [trace](#)
- mv::TypeSemanticsTable [typeSemanticsTable](#)
- std::string [query](#)
- bool [evaluationResult](#)

## Static Protected Attributes

- static const std::string [SEMANTIC\\_TYPE\\_ORGAN\\_HEART](#) = "Organ.Heart"
- static const std::string [SEMANTIC\\_TYPE\\_ORGAN\\_KIDNEY](#) = "Organ.Kidney"

## Private Member Functions

- void [InitialiseQuery \(const std::string &query\)](#)  
*Initialise the query.*
- void [InitialiseTypeSemanticsTable \(\)](#)  
*Initialise the type semantics table.*

### 7.211.1 Detailed Description

Class for testing evaluation of traces.

Definition at line 23 of file TraceEvaluationTest.hpp.

### 7.211.2 Constructor & Destructor Documentation

#### 7.211.2.1 TraceEvaluationTest::TraceEvaluationTest ( )

Definition at line 6 of file TraceEvaluationTest.cpp.

### 7.211.3 Member Function Documentation

7.211.3.1 void TraceEvaluationTest::InitialiseQuery ( const std::string & *query* ) [private]

Initialise the query.

**Parameters**

|              |                 |
|--------------|-----------------|
| <i>query</i> | The given query |
|--------------|-----------------|

Definition at line 40 of file TraceEvaluationTest.cpp.

References [query](#).

Referenced by [RunEvaluationTest\(\)](#).

**7.211.3.2 virtual void multiscaletest::TraceEvaluationTest::InitialiseTrace( ) [protected], [pure virtual]**

Initialise the trace.

Implemented in [multiscaletest::CompleteTraceTest](#), [multiscaletest::SpatialEntitiesTraceTest](#), [multiscaletest::NumericStateVariableTraceTest](#), and [multiscaletest::EmptyTraceTest](#).

Referenced by [RunEvaluationTest\(\)](#).

**7.211.3.3 void TraceEvaluationTest::InitialiseTypeSemanticsTable( ) [private]**

Initialise the type semantics table.

Definition at line 44 of file TraceEvaluationTest.cpp.

References [multiscale::verification::TypeSemanticsTable::addTableEntry\(\)](#), SEMANTIC\_TYPE\_ORGAN\_HEART, SEMANTIC\_TYPE\_ORGAN\_KIDNEY, and [typeSemanticsTable](#).

Referenced by [RunEvaluationTest\(\)](#).

**7.211.3.4 bool TraceEvaluationTest::RunEvaluationTest( const std::string & *query* )**

Run the test with the given string.

**Parameters**

|              |                 |
|--------------|-----------------|
| <i>query</i> | The given query |
|--------------|-----------------|

Definition at line 16 of file TraceEvaluationTest.cpp.

References [evaluationResult](#), [InitialiseQuery\(\)](#), [InitialiseTrace\(\)](#), [InitialiseTypeSemanticsTable\(\)](#), [RunTest\(\)](#), and [ValidateTestResults\(\)](#).

**7.211.3.5 void TraceEvaluationTest::RunTest( ) [override], [protected], [virtual]**

Run the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 27 of file TraceEvaluationTest.cpp.

References [ERR\\_MSG\\_TEST](#), [multiscale::verification::AbstractSyntaxTree::evaluate\(\)](#), [evaluationResult](#), [MS\\_throw](#), [multiscale::verification::Parser::parse\(\)](#), [query](#), [trace](#), and [typeSemanticsTable](#).

Referenced by [RunEvaluationTest\(\)](#).

**7.211.3.6 void TraceEvaluationTest::ValidateTestResults( ) [override], [protected], [virtual]**

Validate the results of the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 38 of file TraceEvaluationTest.cpp.

Referenced by RunEvaluationTest().

#### 7.211.4 Member Data Documentation

7.211.4.1 **double multiscaletest::TraceEvaluationTest::a.MaxValue** [protected]

The maximum value of numeric state variable "A"

Definition at line 41 of file TraceEvaluationTest.hpp.

7.211.4.2 **double multiscaletest::TraceEvaluationTest::a.MinValue** [protected]

The minimum value of numeric state variable "A"

Definition at line 40 of file TraceEvaluationTest.hpp.

7.211.4.3 **NumericStateVariableId multiscaletest::TraceEvaluationTest::aNumericStateVariableId** [protected]

The id of the numeric state variable "A" (no type)

Definition at line 30 of file TraceEvaluationTest.hpp.

7.211.4.4 **NumericStateVariableId multiscaletest::TraceEvaluationTest::aWithTypeNumericStateVariableId** [protected]

The id of the numeric state variable "A" (with type)

Definition at line 34 of file TraceEvaluationTest.hpp.

7.211.4.5 **double multiscaletest::TraceEvaluationTest::b.ConstantValue** [protected]

The constant value of numeric state variable "B"

Definition at line 42 of file TraceEvaluationTest.hpp.

7.211.4.6 **NumericStateVariableId multiscaletest::TraceEvaluationTest::b.NumericStateVariableId** [protected]

The id of the numeric state variable "B" (no type)

Definition at line 32 of file TraceEvaluationTest.hpp.

7.211.4.7 **NumericStateVariableId multiscaletest::TraceEvaluationTest::bWithTypeNumericStateVariableId** [protected]

The id of the numeric state variable "B" (with type)

Definition at line 36 of file TraceEvaluationTest.hpp.

7.211.4.8 **double multiscaletest::TraceEvaluationTest::c.MaxValue** [protected]

The maximum value of numeric state variable "C"

Definition at line 44 of file TraceEvaluationTest.hpp.

7.211.4.9 `double multiscaletest::TraceEvaluationTest::cMinValue [protected]`

The minimum value of numeric state variable "C"

Definition at line 43 of file TraceEvaluationTest.hpp.

7.211.4.10 `NumericStateVariableId multiscaletest::TraceEvaluationTest::cNumericStateVariableId [protected]`

The id of the numeric state variable "C"

Definition at line 38 of file TraceEvaluationTest.hpp.

7.211.4.11 `bool multiscaletest::TraceEvaluationTest::evaluationResult [protected]`

The result of the evaluation

Definition at line 53 of file TraceEvaluationTest.hpp.

Referenced by RunEvaluationTest(), and RunTest().

7.211.4.12 `std::size_t multiscaletest::TraceEvaluationTest::nrOfTimePoints [protected]`

The number of timepoints in the trace

Definition at line 27 of file TraceEvaluationTest.hpp.

7.211.4.13 `std::string multiscaletest::TraceEvaluationTest::query [protected]`

The query to be checked

Definition at line 51 of file TraceEvaluationTest.hpp.

Referenced by InitialiseQuery(), and RunTest().

7.211.4.14 `const std::string multiscaletest::TraceEvaluationTest::SEMANTIC_TYPE_ORGAN_HEART = "Organ.Heart" [static], [protected]`

Definition at line 90 of file TraceEvaluationTest.hpp.

Referenced by InitialiseTypeSemanticsTable().

7.211.4.15 `const std::string multiscaletest::TraceEvaluationTest::SEMANTIC_TYPE_ORGAN_KIDNEY = "Organ.Kidney" [static], [protected]`

Definition at line 91 of file TraceEvaluationTest.hpp.

Referenced by InitialiseTypeSemanticsTable().

7.211.4.16 `mv::SpatialTemporalTrace multiscaletest::TraceEvaluationTest::trace [protected]`

The spatial temporal trace

Definition at line 47 of file TraceEvaluationTest.hpp.

Referenced by RunTest().

### 7.211.4.17 mv::TypeSemanticsTable multiscaletest::TraceEvaluationTest::typeSemanticsTable [protected]

The type semantics table

Definition at line 49 of file TraceEvaluationTest.hpp.

Referenced by InitialiseTypeSemanticsTable(), and RunTest().

The documentation for this class was generated from the following files:

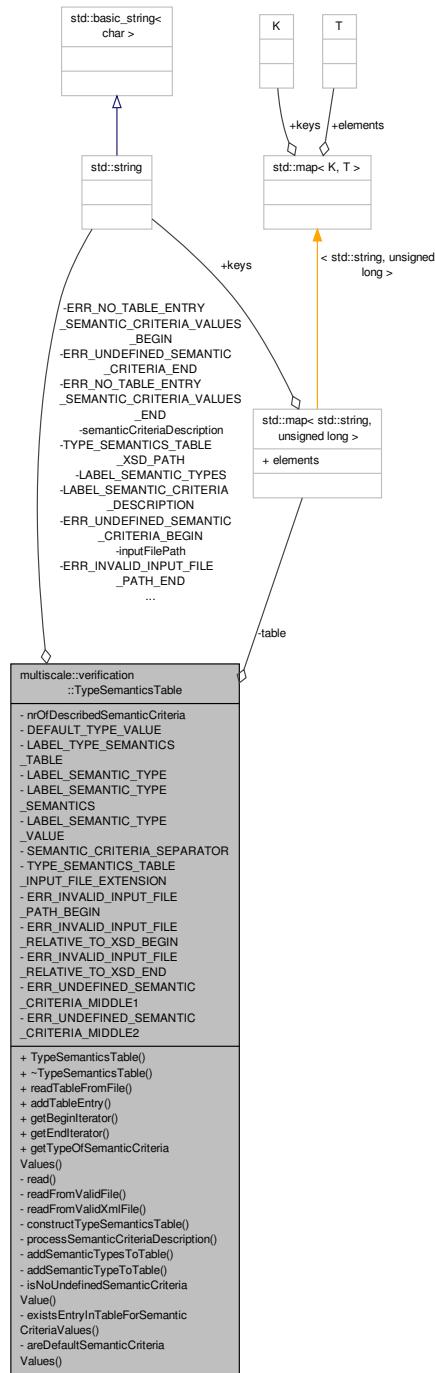
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[TraceEvaluationTest.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[TraceEvaluationTest.cpp](#)

## 7.212 multiscale::verification::TypeSemanticsTable Class Reference

Class for defining a type semantics table.

```
#include <TypeSemanticsTable.hpp>
```

Collaboration diagram for multiscale::verification::TypeSemanticsTable:



## Public Member Functions

- `TypeSemanticsTable ()`
- `~TypeSemanticsTable ()`
- void `readTableFromFile (const std::string &inputFilePath)`

*Read the type semantics table from the provided input file.*

- void `addTableEntry (const std::string &semanticCriteriaValues, unsigned long type)`

- Add a mapping to the table between semantic criteria values and an abstract type.
- std::map< std::string, unsigned long >
  - ::const\_iterator **getBeginIterator** () const
 

Get a const reference to the iterator pointing at the beginning of the table.
  - std::map< std::string, unsigned long >
    - ::const\_iterator **getEndIterator** () const
 

Get a const reference to the iterator pointing at the end of the table.
  - unsigned long **getTypeOfSemanticCriteriaValues** (const std::string &semanticCriteriaValues) const
 

Get the type corresponding to the semantic criteria values.

## Private Member Functions

- void **read** ()
 

Read the type semantics table from the provided input file.
- void **readFromValidFile** ()
 

Read the type semantics table from the provided valid input file.
- void **readFromValidXmlFile** ()
 

Read the type semantics table from the provided valid xml input file.
- void **constructTypeSemanticsTable** (const pt::ptree &propertyTree)
 

Construct a type semantics table from the provided property tree.
- void **processSemanticCriteriaDescription** (const pt::ptree &propertyTree)
 

Process semantic criteria description and initialise the corresponding fields.
- void **addSemanticTypesToTable** (const pt::ptree &propertyTree)
 

Add the semantic types to the type semantics table.
- void **addSemanticTypeToTable** (const pt::ptree &semanticTypeTree)
 

Add a semantic type to the type semantics table.
- bool **isNoUndefinedSemanticCriteriaValue** (const std::string &semanticCriteriaValues)
 

Check if the provided semantic criteria value is valid.
- bool **existsEntryInTableForSemanticCriteriaValues** (const std::string &semanticCriteriaValues) const
 

Check if an entry exists in the table for the provided semantic criteria values.
- bool **areDefaultSemanticCriteriaValues** (const std::string &semanticCriteriaValues) const
 

Check if the provided semantic criteria values equal the default semantic criteria value.

## Private Attributes

- std::string **inputFilePath**
- std::string **semanticCriteriaDescription**
- unsigned long **nrOfDescribedSemanticCriteria**
- std::map< std::string, unsigned long > **table**

## Static Private Attributes

- static const unsigned long **DEFAULT\_TYPE\_VALUE** = 0
- static const std::string **LABEL\_TYPE\_SEMATICS\_TABLE** = "typeSemanticsTable"
- static const std::string **LABEL\_SEMANTIC\_CRITERIA\_DESCRIPTION** = **LABEL\_TYPE\_SEMATICS\_TABLE** + ".semanticCriteria.<xmllat>.description"
- static const std::string **LABEL\_SEMANTIC\_TYPES** = **LABEL\_TYPE\_SEMATICS\_TABLE** + ".semanticTypes"

- static const std::string `LABEL_SEMANTIC_TYPE` = "type"
- static const std::string `LABEL_SEMANTIC_TYPE_SEMATICS` = "<xmlattr>.semantics"
- static const std::string `LABEL_SEMANTIC_TYPE_VALUE` = "<xmlattr>.value"
- static const char `SEMANTIC_CRITERIA_SEPARATOR` = ''
- static const std::string `TYPE_SEMATICS_TABLE_INPUT_FILE_EXTENSION` = ".xml"
- static const std::string `TYPE_SEMATICS_TABLE_XSD_PATH` = "/usr/local/share/mule/config/verification/spatial-temporal/schema/type\_semantics\_table.xsd"
- static const std::string `ERR_INVALID_INPUT_FILE_PATH_BEGIN` = "The provided input file path ("
- static const std::string `ERR_INVALID_INPUT_FILE_PATH_END` = ") does not point to a regular file with the required extension (" + `TYPE_SEMATICS_TABLE_INPUT_FILE_EXTENSION` + "). Please change."
- static const std::string `ERR_INVALID_INPUT_FILE_RELATIVE_TO_XSD_BEGIN` = "The provided xml input file ("
- static const std::string `ERR_INVALID_INPUT_FILE_RELATIVE_TO_XSD_END` = ") is invalid relative to the xsd file (" + `TYPE_SEMATICS_TABLE_XSD_PATH` + ")."
- static const std::string `ERR_UNDEFINED_SEMANTIC_CRITERIA_BEGIN` = "The number of semantic criteria provided in the semantic description ("
- static const std::string `ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE1` = ") of input file ("
- static const std::string `ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE2` = ") is smaller than the number of semantic criteria considered in one of the semantic types definitions ("
- static const std::string `ERR_UNDEFINED_SEMANTIC_CRITERIA_END` = "). Please extend the semantic criteria description accordingly."
- static const std::string `ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_BEGIN` = "There is no type semantics `table` entry for the provided semantic criteria values ("
- static const std::string `ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_END` = "). Please change."

### 7.212.1 Detailed Description

Class for defining a type semantics table.

Definition at line 18 of file TypeSemanticsTable.hpp.

### 7.212.2 Constructor & Destructor Documentation

#### 7.212.2.1 TypeSemanticsTable::TypeSemanticsTable( )

Definition at line 11 of file TypeSemanticsTable.cpp.

#### 7.212.2.2 TypeSemanticsTable::~TypeSemanticsTable( )

Definition at line 13 of file TypeSemanticsTable.cpp.

### 7.212.3 Member Function Documentation

#### 7.212.3.1 void TypeSemanticsTable::addSemanticTypesToTable( const pt::ptree & *propertyTree* ) [private]

Add the semantic types to the type semantics table.

##### Parameters

|                           |                                                       |
|---------------------------|-------------------------------------------------------|
| <code>propertyTree</code> | The property tree corresponding to the xml input file |
|---------------------------|-------------------------------------------------------|

Definition at line 99 of file TypeSemanticsTable.cpp.

References addSemanticTypeToTable(), and `LABEL_SEMANTIC_TYPES`.

Referenced by constructTypeSemanticsTable().

### 7.212.3.2 void TypeSemanticsTable::addSemanticTypeToTable ( const pt::ptree & semanticTypeTree ) [private]

Add a semantic type to the type semantics table.

**Parameters**

|                         |                                                      |
|-------------------------|------------------------------------------------------|
| <i>semanticTypeTree</i> | The property tree corresponding to the semantic type |
|-------------------------|------------------------------------------------------|

Definition at line 108 of file TypeSemanticsTable.cpp.

References addTableEntry(), ERR\_UNDEFINED\_SEMANTIC\_CRITERIA\_BEGIN, ERR\_UNDEFINED\_SEMANTIC\_CRITERIA\_END, ERR\_UNDEFINED\_SEMANTIC\_CRITERIA\_MIDDLE1, ERR\_UNDEFINED\_SEMANTIC\_CRITERIA\_MIDDLE2, inputFilePath, isNoUndefinedSemanticCriteriaValue(), LABEL\_SEMANTIC\_TYPE\_SEMANTICS, LABEL\_SEMANTIC\_TYPE\_VALUE, MS\_throw, and semanticCriteriaDescription.

Referenced by addSemanticTypesToTable().

### 7.212.3.3 void TypeSemanticsTable::addTableEntry ( const std::string & semanticCriteriaValues, unsigned long type )

Add a mapping to the table between semantic criteria values and an abstract type.

If another type T was already associated with the semantic criteria values, T will be overwritten by the newly provided type.

**Parameters**

|                               |                                                                                                                                                   |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>semanticCriteriaValues</i> | The considered semantic criteria values specified in the "semanticCriterion1Value.semanticCriterion2Value. .... . semanticCriterionNValue" format |
| <i>type</i>                   | The abstract natural number representing the type associated to the given semantic criteria values                                                |

Definition at line 128 of file TypeSemanticsTable.cpp.

References table.

Referenced by addSemanticTypeToTable(), and multiscaletest::TraceEvaluationTest::InitialiseTypeSemanticsTable().

### 7.212.3.4 bool TypeSemanticsTable::areDefaultSemanticCriteriaValues ( const std::string & semanticCriteriaValues ) const [private]

Check if the provided semantic criteria values equal the default semantic criteria value.

**Parameters**

|                               |                                       |
|-------------------------------|---------------------------------------|
| <i>semanticCriteriaValues</i> | The provided semantic criteria values |
|-------------------------------|---------------------------------------|

Definition at line 148 of file TypeSemanticsTable.cpp.

References multiscale::verification::SemanticType::DEFAULT\_VALUE.

Referenced by getTypeOfSemanticCriteriaValues().

### 7.212.3.5 void TypeSemanticsTable::constructTypeSemanticsTable ( const pt::ptree & propertyTree ) [private]

Construct a type semantics table from the provided property tree.

**Parameters**

|                     |                                                       |
|---------------------|-------------------------------------------------------|
| <i>propertyTree</i> | The property tree corresponding to the xml input file |
|---------------------|-------------------------------------------------------|

Definition at line 86 of file TypeSemanticsTable.cpp.

References addSemanticTypesToTable(), and processSemanticCriteriaDescription().

Referenced by readFromValidXmlFile().

**7.212.3.6 bool TypeSemanticsTable::existsEntryInTableForSemanticCriteriaValues ( const std::string & *semanticCriteriaValues* ) const [private]**

Check if an entry exists in the table for the provided semantic criteria values.

**Parameters**

|                               |                                       |
|-------------------------------|---------------------------------------|
| <i>semanticCriteriaValues</i> | The provided semantic criteria values |
|-------------------------------|---------------------------------------|

Definition at line 141 of file TypeSemanticsTable.cpp.

References table.

Referenced by getTypeOfSemanticCriteriaValues().

**7.212.3.7 std::map< std::string, unsigned long >::const\_iterator TypeSemanticsTable::getBeginIterator ( ) const**

Get a const reference to the iterator pointing at the beginning of the table.

Definition at line 22 of file TypeSemanticsTable.cpp.

References table.

Referenced by printTypeSemanticsTableContents().

**7.212.3.8 std::map< std::string, unsigned long >::const\_iterator TypeSemanticsTable::getEndIterator ( ) const**

Get a const reference to the iterator pointing at the end of the table.

Definition at line 26 of file TypeSemanticsTable.cpp.

References table.

Referenced by printTypeSemanticsTableContents().

**7.212.3.9 unsigned long TypeSemanticsTable::getTypeOfSemanticCriteriaValues ( const std::string & *semanticCriteriaValues* ) const**

Get the type corresponding to the semantic criteria values.

**Parameters**

|                               |                                                                            |
|-------------------------------|----------------------------------------------------------------------------|
| <i>semanticCriteriaValues</i> | The semantic criteria values for which the type (if it exists) is computed |
|-------------------------------|----------------------------------------------------------------------------|

Definition at line 30 of file TypeSemanticsTable.cpp.

References areDefaultSemanticCriteriaValues(), DEFAULT\_TYPE\_VALUE, ERR\_NO\_TABLE\_ENTRY\_SEMANTIC\_CRITERIA\_VALUES\_BEGIN, ERR\_NO\_TABLE\_ENTRY\_SEMANTIC\_CRITERIA\_VALUES\_END, existsEntryInTableForSemanticCriteriaValues(), MS\_throw, and table.

Referenced by multiscale::verification::ConstraintVisitor::translateSemanticTypeToAbstractNaturalNumber().

7.212.3.10 `bool TypeSemanticsTable::isNoUndefinedSemanticCriteriaValue ( const std::string & semanticCriteriaValues ) [private]`

Check if the provided semantic criteria value is valid.

Check if at least one of the provided semantic criteria values is undefined

**Parameters**

|                                     |                                       |
|-------------------------------------|---------------------------------------|
| <code>semanticCriteriaValues</code> | The provided semantic criteria values |
|-------------------------------------|---------------------------------------|

Definition at line 133 of file TypeSemanticsTable.cpp.

References multiscale::StringManipulator::count(), nrOfDescribedSemanticCriteria, and SEMANTIC\_CRITERIA\_SEPARATOR.

Referenced by addSemanticTypeToTable().

7.212.3.11 `void TypeSemanticsTable::processSemanticCriteriaDescription ( const pt::ptree & propertyTree ) [private]`

Process semantic criteria description and initialise the corresponding fields.

**Parameters**

|                           |                                                       |
|---------------------------|-------------------------------------------------------|
| <code>propertyTree</code> | The property tree corresponding to the xml input file |
|---------------------------|-------------------------------------------------------|

Definition at line 91 of file TypeSemanticsTable.cpp.

References multiscale::StringManipulator::count(), LABEL\_SEMANTIC\_CRITERIA\_DESCRIPTION, nrOfDescribedSemanticCriteria, SEMANTIC\_CRITERIA\_SEPARATOR, and semanticCriteriaDescription.

Referenced by constructTypeSemanticsTable().

7.212.3.12 `void TypeSemanticsTable::read ( ) [private]`

Read the type semantics table from the provided input file.

Definition at line 49 of file TypeSemanticsTable.cpp.

References ERR\_INVALID\_INPUT\_FILE\_PATH\_BEGIN, ERR\_INVALID\_INPUT\_FILE\_PATH\_END, inputFilePath, multiscale::Filesystem::isValidFilePath(), MS\_throw, readFromValidFile(), and TYPE\_SEMATICS\_TABLE\_INPUT\_FILE\_EXTENSION.

Referenced by readTableFromFile().

7.212.3.13 `void TypeSemanticsTable::readFromValidFile ( ) [private]`

Read the type semantics table from the provided valid input file.

Precondition: The provided input file path points to a regular file with the correct extension.

Definition at line 62 of file TypeSemanticsTable.cpp.

References ERR\_INVALID\_INPUT\_FILE\_RELATIVE\_TO\_XSD\_BEGIN, ERR\_INVALID\_INPUT\_FILE\_RELATIVE\_TO\_XSD\_END, inputFilePath, multiscale::XmlValidator::isValidXmlFile(), MS\_throw, readFromValidXmlFile(), and TYPE\_SEMATICS\_TABLE\_XSD\_PATH.

Referenced by read().

7.212.3.14 `void TypeSemanticsTable::readFromValidXmlFile ( ) [private]`

Read the type semantics table from the provided valid xml input file.

Precondition: The provided input file path points to a valid xml file (relative to xsd).

Definition at line 78 of file TypeSemanticsTable.cpp.

References constructTypeSemanticsTable(), and inputFilePath.

Referenced by readFromValidFile().

#### 7.212.3.15 void TypeSemanticsTable::readTableFromFile ( const std::string & *inputFilePath* )

Read the type semantics table from the provided input file.

Warning: The contents of the existing table will be overwritten by the contents stored in the file.

##### Parameters

|                      |                            |
|----------------------|----------------------------|
| <i>inputFilePath</i> | The path to the input file |
|----------------------|----------------------------|

Definition at line 15 of file TypeSemanticsTable.cpp.

References inputFilePath, and read().

Referenced by multiscale::verification::ModelCheckingManager::initialiseTypeSemanticsTable(), and readTypeSemanticsTable().

## 7.212.4 Member Data Documentation

### 7.212.4.1 const unsigned long TypeSemanticsTable::DEFAULT\_TYPE\_VALUE = 0 [static], [private]

Definition at line 138 of file TypeSemanticsTable.hpp.

Referenced by getTypeOfSemanticCriteriaValues().

### 7.212.4.2 const std::string TypeSemanticsTable::ERR\_INVALID\_INPUT\_FILE\_PATH\_BEGIN = "The provided input file path (" [static], [private]

Definition at line 153 of file TypeSemanticsTable.hpp.

Referenced by read().

### 7.212.4.3 const std::string TypeSemanticsTable::ERR\_INVALID\_INPUT\_FILE\_PATH\_END = ") does not point to a regular file with the required extension (" + TYPE\_SEMATICS\_TABLE\_INPUT\_FILE\_EXTENSION + "). Please change." [static], [private]

Definition at line 154 of file TypeSemanticsTable.hpp.

Referenced by read().

### 7.212.4.4 const std::string TypeSemanticsTable::ERR\_INVALID\_INPUT\_FILE\_RELATIVE\_TO\_XSD\_BEGIN = "The provided xml input file (" [static], [private]

Definition at line 156 of file TypeSemanticsTable.hpp.

Referenced by readFromValidFile().

### 7.212.4.5 const std::string TypeSemanticsTable::ERR\_INVALID\_INPUT\_FILE\_RELATIVE\_TO\_XSD\_END = ") is invalid relative to the xsd file (" + TYPE\_SEMATICS\_TABLE\_XSD\_PATH + "). " [static], [private]

Definition at line 157 of file TypeSemanticsTable.hpp.

Referenced by readFromValidFile().

7.212.4.6 `const std::string TypeSemanticsTable::ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_BEGIN = "There is no type semantics table entry for the provided semantic criteria values (" [static], [private]`

Definition at line 164 of file TypeSemanticsTable.hpp.

Referenced by `getTypeOfSemanticCriteriaValues()`.

7.212.4.7 `const std::string TypeSemanticsTable::ERR_NO_TABLE_ENTRY_SEMANTIC_CRITERIA_VALUES_END = "). Please change." [static], [private]`

Definition at line 165 of file TypeSemanticsTable.hpp.

Referenced by `getTypeOfSemanticCriteriaValues()`.

7.212.4.8 `const std::string TypeSemanticsTable::ERR_UNDEFINED_SEMANTIC_CRITERIA_BEGIN = "The number of semantic criteria provided in the semantic description (" [static], [private]`

Definition at line 159 of file TypeSemanticsTable.hpp.

Referenced by `addSemanticTypeToTable()`.

7.212.4.9 `const std::string TypeSemanticsTable::ERR_UNDEFINED_SEMANTIC_CRITERIA_END = "). Please extend the semantic criteria description accordingly." [static], [private]`

Definition at line 162 of file TypeSemanticsTable.hpp.

Referenced by `addSemanticTypeToTable()`.

7.212.4.10 `const std::string TypeSemanticsTable::ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE1 = ") of input file (" [static], [private]`

Definition at line 160 of file TypeSemanticsTable.hpp.

Referenced by `addSemanticTypeToTable()`.

7.212.4.11 `const std::string TypeSemanticsTable::ERR_UNDEFINED_SEMANTIC_CRITERIA_MIDDLE2 = ") is smaller than the number of semantic criteria considered in one of the semantic types definitions (" [static], [private]`

Definition at line 161 of file TypeSemanticsTable.hpp.

Referenced by `addSemanticTypeToTable()`.

7.212.4.12 `std::string multiscale::verification::TypeSemanticsTable::inputFilePath [private]`

The path to the input file

Definition at line 23 of file TypeSemanticsTable.hpp.

Referenced by `addSemanticTypeToTable()`, `read()`, `readFromValidFile()`, `readFromValidXmlFile()`, and `readTable<FromFile()`.

7.212.4.13 `const std::string TypeSemanticsTable::LABEL_SEMANTIC_CRITERIA_DESCRIPTION = LABEL_TYPE_SEMATICS_TABLE + ".semanticCriteria.<xmllattr>.description" [static], [private]`

Definition at line 141 of file TypeSemanticsTable.hpp.

Referenced by `processSemanticCriteriaDescription()`.

7.212.4.14 const std::string TypeSemanticsTable::LABEL\_SEMANTIC\_TYPE = "type" [static], [private]

Definition at line 144 of file TypeSemanticsTable.hpp.

7.212.4.15 const std::string TypeSemanticsTable::LABEL\_SEMANTIC\_TYPE\_SEMATICS = "<xmلاtr>.semantics" [static], [private]

Definition at line 145 of file TypeSemanticsTable.hpp.

Referenced by addSemanticTypeToTable().

7.212.4.16 const std::string TypeSemanticsTable::LABEL\_SEMANTIC\_TYPE\_VALUE = "<xmلاtr>.value" [static], [private]

Definition at line 146 of file TypeSemanticsTable.hpp.

Referenced by addSemanticTypeToTable().

7.212.4.17 const std::string TypeSemanticsTable::LABEL\_SEMANTIC\_TYPES = LABEL\_TYPE\_SEMATICS\_TABLE + ".semanticTypes" [static], [private]

Definition at line 142 of file TypeSemanticsTable.hpp.

Referenced by addSemanticTypesToTable().

7.212.4.18 const std::string TypeSemanticsTable::LABEL\_TYPE\_SEMATICS\_TABLE = "typeSemanticsTable" [static], [private]

Definition at line 140 of file TypeSemanticsTable.hpp.

7.212.4.19 unsigned long multiscale::verification::TypeSemanticsTable::nrOfDescribedSemanticCriteria [private]

The number of considered semantic criteria

Definition at line 28 of file TypeSemanticsTable.hpp.

Referenced by isNoUndefinedSemanticCriteriaValue(), and processSemanticCriteriaDescription().

7.212.4.20 const char TypeSemanticsTable::SEMANTIC\_CRITERIA\_SEPARATOR = ':' [static], [private]

Definition at line 148 of file TypeSemanticsTable.hpp.

Referenced by isNoUndefinedSemanticCriteriaValue(), and processSemanticCriteriaDescription().

7.212.4.21 std::string multiscale::verification::TypeSemanticsTable::semanticCriteriaDescription [private]

The description of the considered semantic criteria

Definition at line 26 of file TypeSemanticsTable.hpp.

Referenced by addSemanticTypeToTable(), and processSemanticCriteriaDescription().

7.212.4.22 std::map<std::string, unsigned long> multiscale::verification::TypeSemanticsTable::table [private]

The type semantics table mapping the semantic criteria values to abstract positive natural numbers

Definition at line 31 of file TypeSemanticsTable.hpp.

Referenced by `addTableEntry()`, `existsEntryInTableForSemanticCriteriaValues()`, `getBeginIterator()`, `getEndIterator()`, and `getTypeOfSemanticCriteriaValues()`.

**7.212.4.23** `const std::string TypeSemanticsTable::TYPE_SEMATICS_TABLE_INPUT_FILE_EXTENSION = ".xml"`  
`[static], [private]`

Definition at line 150 of file `TypeSemanticsTable.hpp`.

Referenced by `read()`.

**7.212.4.24** `const std::string TypeSemanticsTable::TYPE_SEMATICS_TABLE_XSD_PATH =`  
`"/usr/local/share/mule/config/verification/spatial-temporal/schema/type_semantics_table.xsd"` `[static],`  
`[private]`

Definition at line 151 of file `TypeSemanticsTable.hpp`.

Referenced by `readFromValidFile()`.

The documentation for this class was generated from the following files:

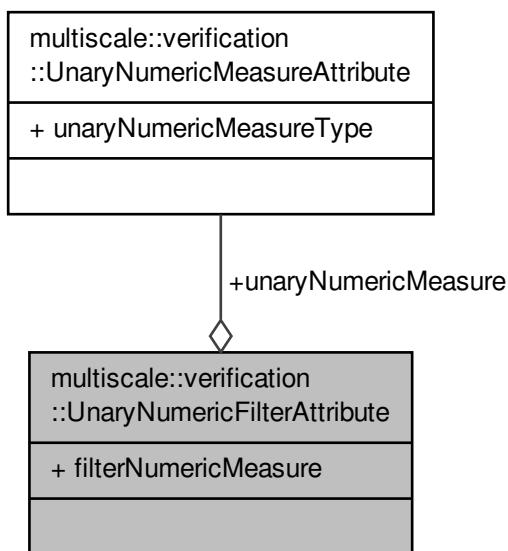
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp`
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/TypeSemanticsTable.cpp`

## 7.213 multiscale::verification::UnaryNumericFilterAttribute Class Reference

Class for representing a unary numeric filter attribute.

```
#include <UnaryNumericFilterAttribute.hpp>
```

Collaboration diagram for `multiscale::verification::UnaryNumericFilterAttribute`:



## Public Attributes

- [UnaryNumericMeasureAttribute unaryNumericMeasure](#)
- [FilterNumericMeasureAttributeType filterNumericMeasure](#)

### 7.213.1 Detailed Description

Class for representing a unary numeric filter attribute.

Definition at line 15 of file [UnaryNumericFilterAttribute.hpp](#).

### 7.213.2 Member Data Documentation

#### 7.213.2.1 FilterNumericMeasureAttributeType multiscale::verification::UnaryNumericFilterAttribute::filterNumericMeasure

The considered filter numeric measure

Definition at line 20 of file [UnaryNumericFilterAttribute.hpp](#).

Referenced by [multiscale::verification::FilterNumericVisitor::operator\(\)\(\)](#).

#### 7.213.2.2 UnaryNumericMeasureAttribute multiscale::verification::UnaryNumericFilterAttribute::unaryNumericMeasure

The unary numeric measure

Definition at line 19 of file [UnaryNumericFilterAttribute.hpp](#).

Referenced by [multiscale::verification::FilterNumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

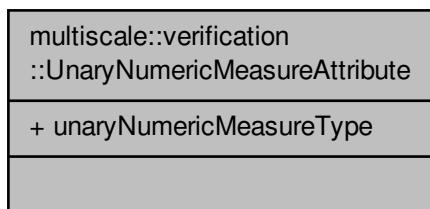
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/\*\*UnaryNumericFilterAttribute.hpp\*\*](#)

## 7.214 multiscale::verification::UnaryNumericMeasureAttribute Class Reference

Class for representing a unary numeric measure attribute.

```
#include <UnaryNumericMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryNumericMeasureAttribute:



## Public Attributes

- [UnaryNumericMeasureType unaryNumericMeasureType](#)

### 7.214.1 Detailed Description

Class for representing a unary numeric measure attribute.

Definition at line 33 of file [UnaryNumericMeasureAttribute.hpp](#).

### 7.214.2 Member Data Documentation

#### 7.214.2.1 [UnaryNumericMeasureType multiscale::verification::UnaryNumericMeasureAttribute::unaryNumericMeasureType](#)

The unary numeric measure type

Definition at line 37 of file [UnaryNumericMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::FilterNumericVisitor::operator\(\)](#).

The documentation for this class was generated from the following file:

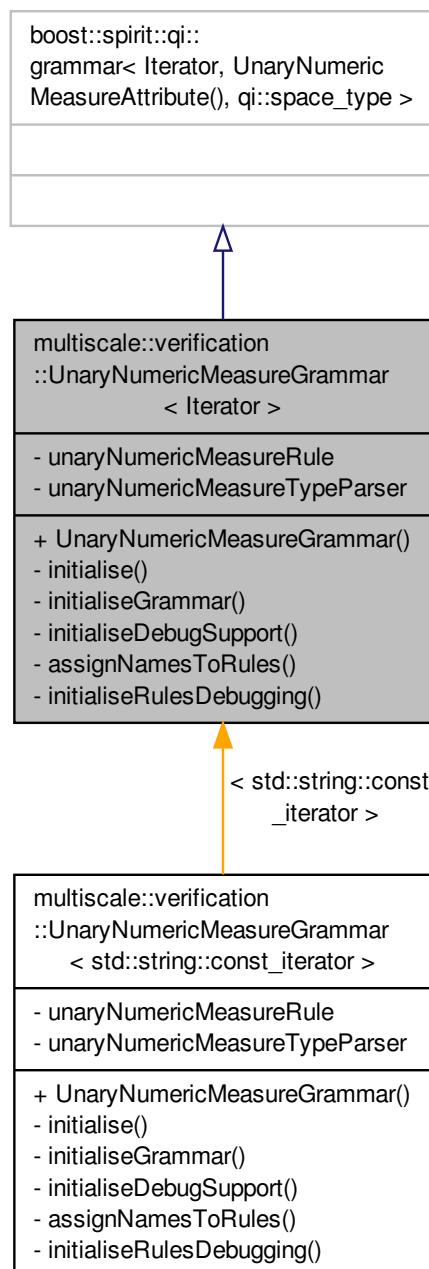
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/\[UnaryNumericMeasureAttribute.hpp\]\(#\)](#)

## 7.215 [multiscale::verification::UnaryNumericMeasureGrammar< Iterator >](#) Class Template Reference

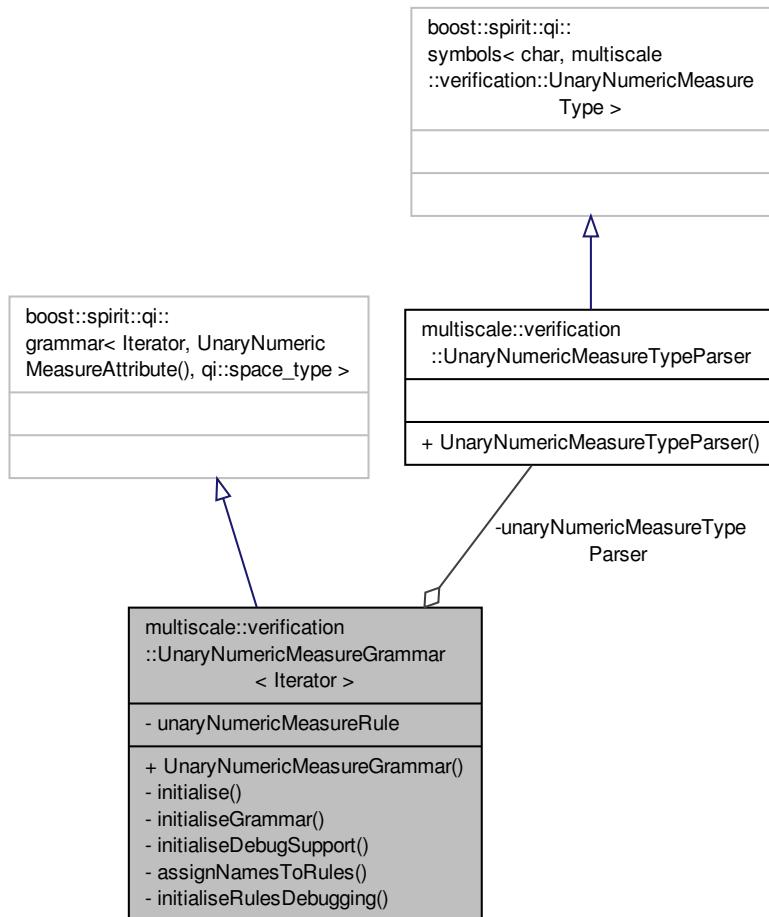
The grammar for parsing unary numeric measure statements.

```
#include <UnaryNumericMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::UnaryNumericMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::UnaryNumericMeasureGrammar< Iterator >:



## Public Member Functions

- [UnaryNumericMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*

## Private Attributes

- qi::rule< Iterator,  
    [UnaryNumericMeasureAttribute\(\)](#),  
    qi::space\_type > unaryNumericMeasureRule
- [UnaryNumericMeasureTypeParser](#) unaryNumericMeasureTypeParser

### 7.215.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::UnaryNumericMeasureGrammar< Iterator >
```

The grammar for parsing unary numeric measure statements.

Definition at line 30 of file UnaryNumericMeasureGrammar.hpp.

### 7.215.2 Constructor & Destructor Documentation

```
7.215.2.1 template<typename Iterator > multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::UnaryNumericMeasureGrammar()
```

Definition at line 23 of file UnaryNumericMeasureGrammarDefinition.hpp.

References multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::initialise().

### 7.215.3 Member Function Documentation

```
7.215.3.1 template<typename Iterator > void multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::assignNamesToRules() [private]
```

Assign names to the rules.

Definition at line 56 of file UnaryNumericMeasureGrammarDefinition.hpp.

```
7.215.3.2 template<typename Iterator > void multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::initialise() [private]
```

Initialisation function.

Definition at line 33 of file UnaryNumericMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::UnaryNumericMeasureGrammar().

```
7.215.3.3 template<typename Iterator > void multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::initialiseDebugSupport() [private]
```

Initialise debug support.

Definition at line 47 of file UnaryNumericMeasureGrammarDefinition.hpp.

```
7.215.3.4 template<typename Iterator > void multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::initialiseGrammar() [private]
```

Initialise the grammar.

Definition at line 40 of file UnaryNumericMeasureGrammarDefinition.hpp.

---

7.215.3.5 template<typename Iterator > void multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 62 of file UnaryNumericMeasureGrammarDefinition.hpp.

#### 7.215.4 Member Data Documentation

7.215.4.1 template<typename Iterator> qi::rule<Iterator, UnaryNumericMeasureAttribute(), qi::space\_type> multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::unaryNumericMeasureRule [private]

The rule for parsing a unary numeric measure

Definition at line 36 of file UnaryNumericMeasureGrammar.hpp.

7.215.4.2 template<typename Iterator> UnaryNumericMeasureTypeParser multiscale::verification::UnaryNumericMeasureGrammar< Iterator >::unaryNumericMeasureTypeParser [private]

The unary numeric measure type parser

Definition at line 41 of file UnaryNumericMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

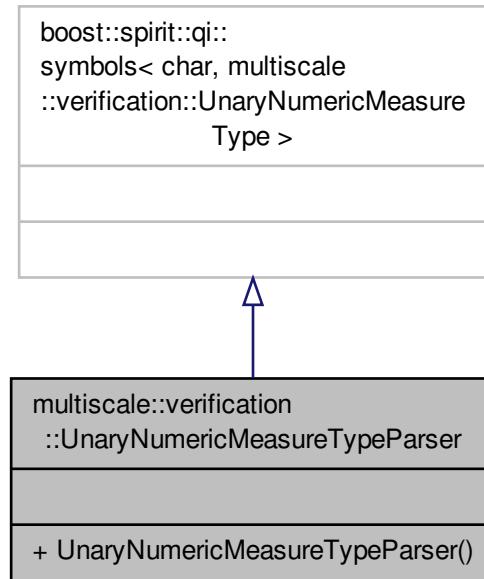
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[UnaryNumericMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[UnaryNumericMeasureGrammarDefinition.hpp](#)

#### 7.216 multiscale::verification::UnaryNumericMeasureTypeParser Struct Reference

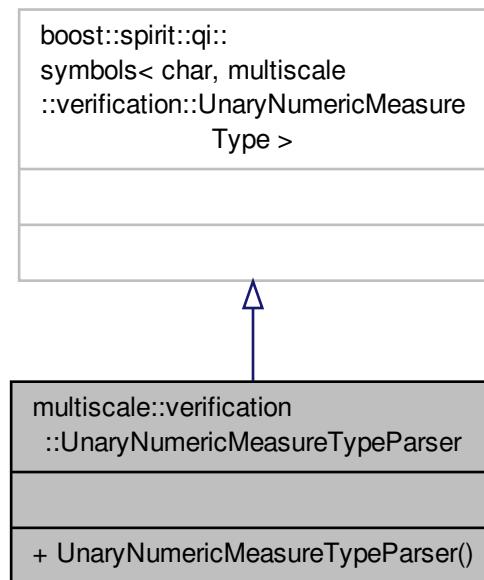
Symbol table and parser for the unary numeric measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::UnaryNumericMeasureTypeParser:



Collaboration diagram for multiscale::verification::UnaryNumericMeasureTypeParser:



## Public Member Functions

- [UnaryNumericMeasureTypeParser \(\)](#)

### 7.216.1 Detailed Description

Symbol table and parser for the unary numeric measure type.

Definition at line 196 of file [SymbolTables.hpp](#).

### 7.216.2 Constructor & Destructor Documentation

#### 7.216.2.1 multiscale::verification::UnaryNumericMeasureTypeParser::UnaryNumericMeasureTypeParser ( ) [inline]

Definition at line 199 of file [SymbolTables.hpp](#).

References [multiscale::verification::Abs](#), [multiscale::verification::Ceil](#), [multiscale::verification::Floor](#), [multiscale::verification::Round](#), [multiscale::verification::Sign](#), [multiscale::verification::Sqrt](#), and [multiscale::verification::Trunc](#).

The documentation for this struct was generated from the following file:

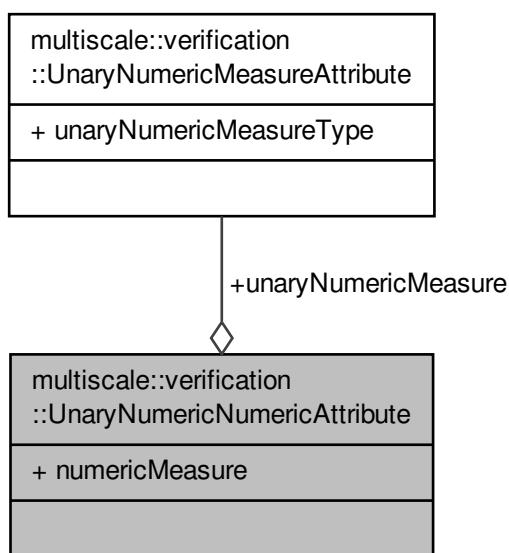
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

## 7.217 multiscale::verification::UnaryNumericNumericAttribute Class Reference

Class for representing a unary numeric numeric measure attribute.

```
#include <UnaryNumericNumericAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryNumericNumericAttribute:



## Public Attributes

- [UnaryNumericMeasureAttribute unaryNumericMeasure](#)
- [NumericMeasureType numericMeasure](#)

### 7.217.1 Detailed Description

Class for representing a unary numeric numeric measure attribute.

Definition at line 15 of file [UnaryNumericNumericAttribute.hpp](#).

### 7.217.2 Member Data Documentation

#### 7.217.2.1 NumericMeasureType multiscale::verification::UnaryNumericNumericAttribute::numericMeasure

The considered numeric measure

Definition at line 20 of file [UnaryNumericNumericAttribute.hpp](#).

Referenced by [multiscale::verification::NumericVisitor::operator\(\)\(\)](#).

#### 7.217.2.2 UnaryNumericMeasureAttribute multiscale::verification::UnaryNumericNumericAttribute::unaryNumericMeasure

The unary numeric measure

Definition at line 19 of file [UnaryNumericNumericAttribute.hpp](#).

The documentation for this class was generated from the following file:

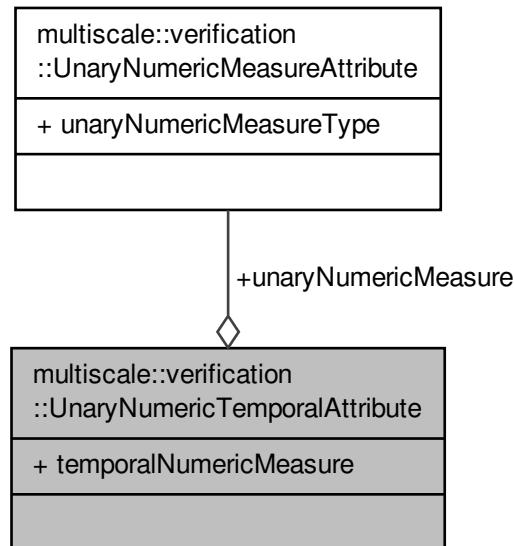
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/\*UnaryNumericNumericAttribute.hpp\*](#)

## 7.218 multiscale::verification::UnaryNumericTemporalAttribute Class Reference

Class for representing a unary numeric temporal measure attribute.

```
#include <UnaryNumericTemporalAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryNumericTemporalAttribute:



## Public Attributes

- [UnaryNumericMeasureAttribute](#) `unaryNumericMeasure`
- [TemporalNumericMeasureType](#) `temporalNumericMeasure`

### 7.218.1 Detailed Description

Class for representing a unary numeric temporal measure attribute.

Definition at line 15 of file `UnaryNumericTemporalAttribute.hpp`.

### 7.218.2 Member Data Documentation

#### 7.218.2.1 TemporalNumericMeasureType multiscale::verification::UnaryNumericTemporalAttribute::temporalNumericMeasure

The considered temporal numeric measure

Definition at line 22 of file `UnaryNumericTemporalAttribute.hpp`.

Referenced by `multiscale::verification::TemporalNumericVisitor::operator()()`.

#### 7.218.2.2 UnaryNumericMeasureAttribute multiscale::verification::UnaryNumericTemporalAttribute::unaryNumericMeasure

The unary numeric measure

Definition at line 20 of file `UnaryNumericTemporalAttribute.hpp`.

The documentation for this class was generated from the following file:

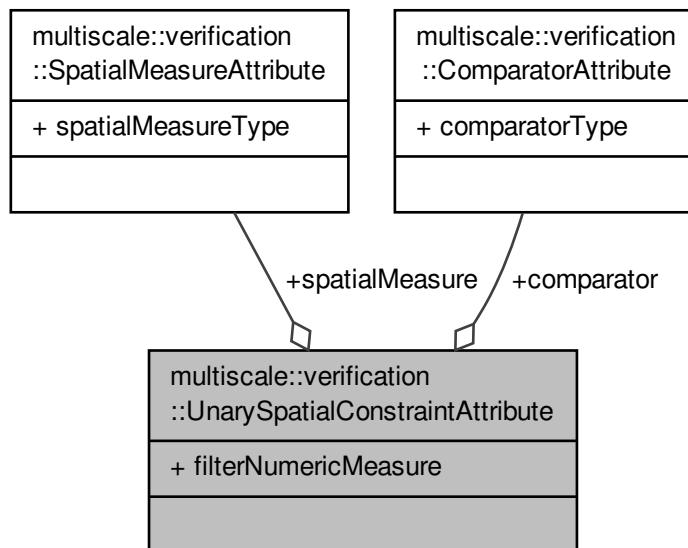
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericTemporalAttribute.hpp

## 7.219 multiscale::verification::UnarySpatialConstraintAttribute Class Reference

Class for representing a "unary" spatial constraint attribute.

```
#include <UnarySpatialConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnarySpatialConstraintAttribute:



### Public Attributes

- `SpatialMeasureAttribute spatialMeasure`
- `ComparatorAttribute comparator`
- `FilterNumericMeasureAttributeType filterNumericMeasure`

### 7.219.1 Detailed Description

Class for representing a "unary" spatial constraint attribute.

Definition at line 16 of file UnarySpatialConstraintAttribute.hpp.

### 7.219.2 Member Data Documentation

#### 7.219.2.1 ComparatorAttribute multiscale::verification::UnarySpatialConstraintAttribute::comparator

The comparator

Definition at line 21 of file UnarySpatialConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

### 7.219.2.2 FilterNumericMeasureAttributeType multiscale::verification::UnarySpatialConstraintAttribute::filterNumericMeasure

The filter numeric measure

Definition at line 22 of file UnarySpatialConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

### 7.219.2.3 SpatialMeasureAttribute multiscale::verification::UnarySpatialConstraintAttribute::spatialMeasure

The spatial measure

Definition at line 20 of file UnarySpatialConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

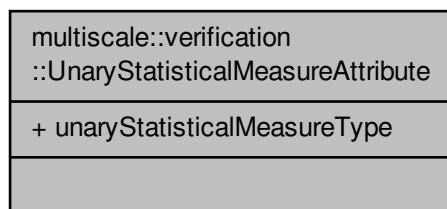
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnarySpatialConstraintAttribute.hpp](#)

## 7.220 multiscale::verification::UnaryStatisticalMeasureAttribute Class Reference

Class for representing a unary statistical measure attribute.

```
#include <UnaryStatisticalMeasureAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryStatisticalMeasureAttribute:



### Public Attributes

- [UnaryStatisticalMeasureType unaryStatisticalMeasureType](#)

### 7.220.1 Detailed Description

Class for representing a unary statistical measure attribute.

Definition at line 40 of file UnaryStatisticalMeasureAttribute.hpp.

## 7.220.2 Member Data Documentation

### 7.220.2.1 UnaryStatisticalMeasureType multiscale::verification::UnaryStatisticalMeasureAttribute::unaryStatisticalMeasureType

The unary statistical measure type

Definition at line 44 of file [UnaryStatisticalMeasureAttribute.hpp](#).

Referenced by [multiscale::verification::TemporalNumericVisitor::operator\(\)\(\)](#), and [multiscale::verification::NumericVisitor::operator\(\)\(\)](#).

The documentation for this class was generated from the following file:

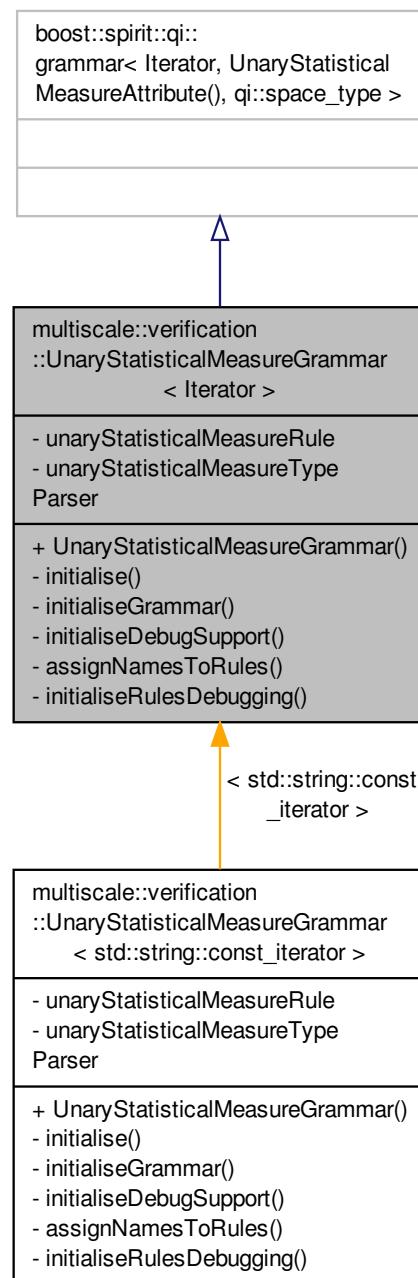
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/\*\*UnaryStatisticalMeasureAttribute.hpp\*\*](#)

## 7.221 multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator > Class Template Reference

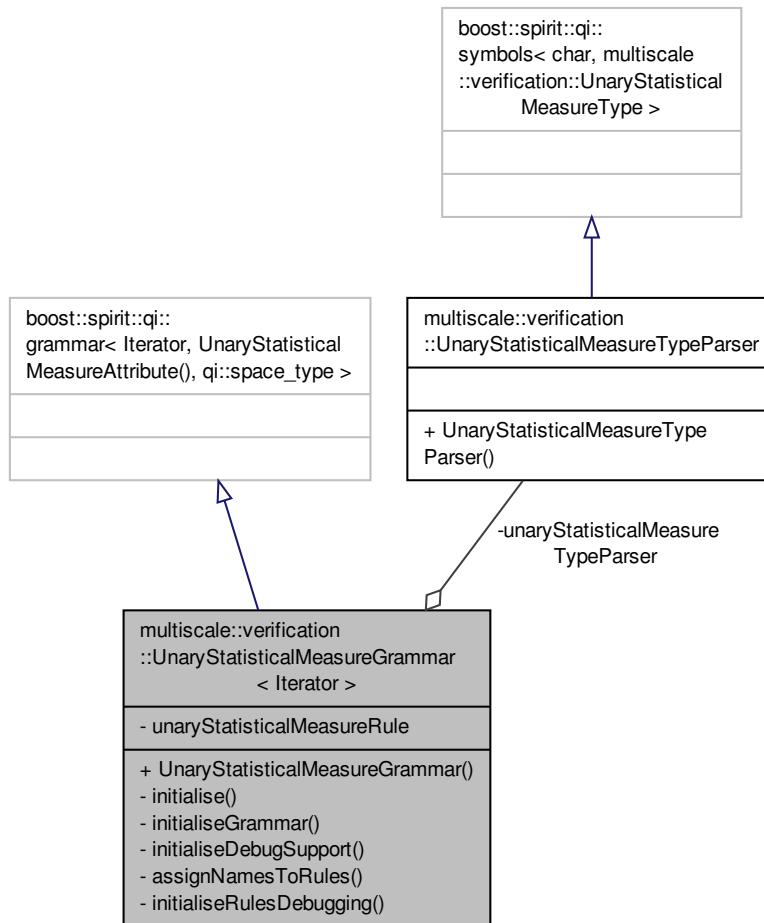
The grammar for parsing unary statistical measure statements.

```
#include <UnaryStatisticalMeasureGrammar.hpp>
```

Inheritance diagram for multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >:



Collaboration diagram for multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >:



## Public Member Functions

- [UnaryStatisticalMeasureGrammar \(\)](#)

## Private Member Functions

- void [initialise \(\)](#)  
*Initialisation function.*
- void [initialiseGrammar \(\)](#)  
*Initialise the grammar.*
- void [initialiseDebugSupport \(\)](#)  
*Initialise debug support.*
- void [assignNamesToRules \(\)](#)  
*Assign names to the rules.*
- void [initialiseRulesDebugging \(\)](#)  
*Initialise the debugging of rules.*

## Private Attributes

- qi::rule< Iterator,  
[UnaryStatisticalMeasureAttribute\(\)](#),  
 qi::space\_type > unaryStatisticalMeasureRule
- [UnaryStatisticalMeasureTypeParser](#) unaryStatisticalMeasureTypeParser

### 7.221.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >
```

The grammar for parsing unary statistical measure statements.

Definition at line 30 of file UnaryStatisticalMeasureGrammar.hpp.

### 7.221.2 Constructor & Destructor Documentation

7.221.2.1 **template<typename Iterator > multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::UnaryStatisticalMeasureGrammar( )**

Definition at line 23 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

References multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::initialise().

### 7.221.3 Member Function Documentation

7.221.3.1 **template<typename Iterator > void multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::assignNamesToRules( ) [private]**

Assign names to the rules.

Definition at line 56 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

7.221.3.2 **template<typename Iterator > void multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::initialise( ) [private]**

Initialisation function.

Definition at line 33 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

Referenced by multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::UnaryStatisticalMeasureGrammar().

7.221.3.3 **template<typename Iterator > void multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::initialiseDebugSupport( ) [private]**

Initialise debug support.

Definition at line 47 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

7.221.3.4 **template<typename Iterator > void multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::initialiseGrammar( ) [private]**

Initialise the grammar.

Definition at line 40 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

---

7.221.3.5 template<typename Iterator > void multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::initialiseRulesDebugging( ) [private]

Initialise the debugging of rules.

Definition at line 62 of file UnaryStatisticalMeasureGrammarDefinition.hpp.

## 7.221.4 Member Data Documentation

7.221.4.1 template<typename Iterator> qi::rule<Iterator, UnaryStatisticalMeasureAttribute(), qi::space\_type> multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::unaryStatisticalMeasureRule [private]

The rule for parsing a unary statistical measure

Definition at line 36 of file UnaryStatisticalMeasureGrammar.hpp.

7.221.4.2 template<typename Iterator> UnaryStatisticalMeasureTypeParser multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >::unaryStatisticalMeasureTypeParser [private]

The unary statistical measure type parser

Definition at line 42 of file UnaryStatisticalMeasureGrammar.hpp.

The documentation for this class was generated from the following files:

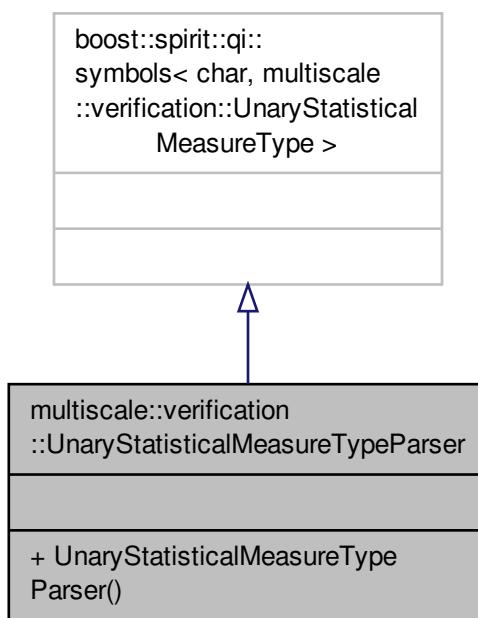
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[UnaryStatisticalMeasureGrammar.hpp](#)
  
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[UnaryStatisticalMeasureGrammarDefinition.hpp](#)

## 7.222 multiscale::verification::UnaryStatisticalMeasureTypeParser Struct Reference

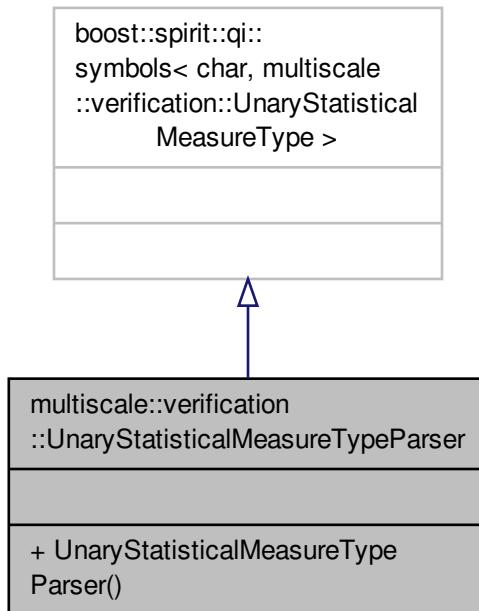
Symbol table and parser for the unary statistical measure type.

```
#include <SymbolTables.hpp>
```

Inheritance diagram for multiscale::verification::UnaryStatisticalMeasureTypeParser:



Collaboration diagram for multiscale::verification::UnaryStatisticalMeasureTypeParser:



## Public Member Functions

- [UnaryStatisticalMeasureTypeParser \(\)](#)

### 7.222.1 Detailed Description

Symbol table and parser for the unary statistical measure type.

Definition at line 214 of file [SymbolTables.hpp](#).

### 7.222.2 Constructor & Destructor Documentation

#### 7.222.2.1 multiscale::verification::UnaryStatisticalMeasureTypeParser::UnaryStatisticalMeasureTypeParser ( ) [inline]

Definition at line 217 of file [SymbolTables.hpp](#).

References [multiscale::verification::Avg](#), [multiscale::verification::Count](#), [multiscale::verification::Geomean](#), [multiscale::verification::Harmean](#), [multiscale::verification::Kurt](#), [multiscale::verification::Max](#), [multiscale::verification::Median](#), [multiscale::verification::Min](#), [multiscale::verification::Mode](#), [multiscale::verification::Product](#), [multiscale::verification::Skew](#), [multiscale::verification::Stdev](#), [multiscale::verification::Sum](#), and [multiscale::verification::Var](#).

The documentation for this struct was generated from the following file:

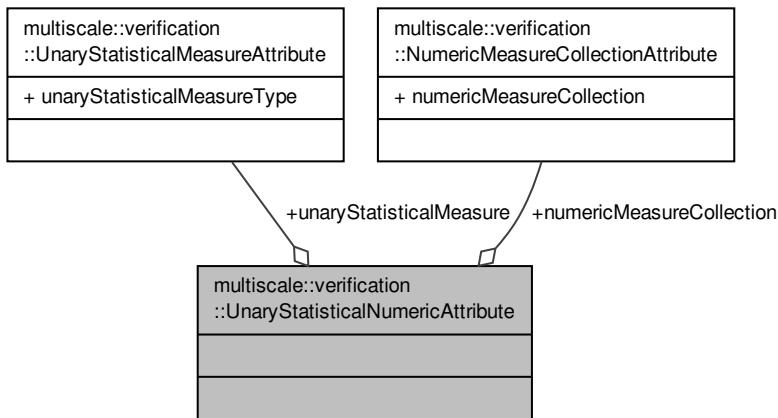
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp](#)

## 7.223 multiscale::verification::UnaryStatisticalNumericAttribute Class Reference

Class for representing a unary statistical numeric attribute.

```
#include <UnaryStatisticalNumericAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryStatisticalNumericAttribute:



### Public Attributes

- [UnaryStatisticalMeasureAttribute unaryStatisticalMeasure](#)
- [NumericMeasureCollectionAttribute numericMeasureCollection](#)

#### 7.223.1 Detailed Description

Class for representing a unary statistical numeric attribute.

Definition at line 15 of file `UnaryStatisticalNumericAttribute.hpp`.

#### 7.223.2 Member Data Documentation

##### 7.223.2.1 NumericMeasureCollectionAttribute multiscale::verification::UnaryStatisticalNumericAttribute::numericMeasureCollection

The considered numeric measure collection

Definition at line 20 of file `UnaryStatisticalNumericAttribute.hpp`.

Referenced by `multiscale::verification::TemporalNumericVisitor::operator()()`.

##### 7.223.2.2 UnaryStatisticalMeasureAttribute multiscale::verification::UnaryStatisticalNumericAttribute::unaryStatisticalMeasure

The unary statistical measure

Definition at line 19 of file `UnaryStatisticalNumericAttribute.hpp`.

Referenced by `multiscale::verification::TemporalNumericVisitor::operator()()`.

The documentation for this class was generated from the following file:

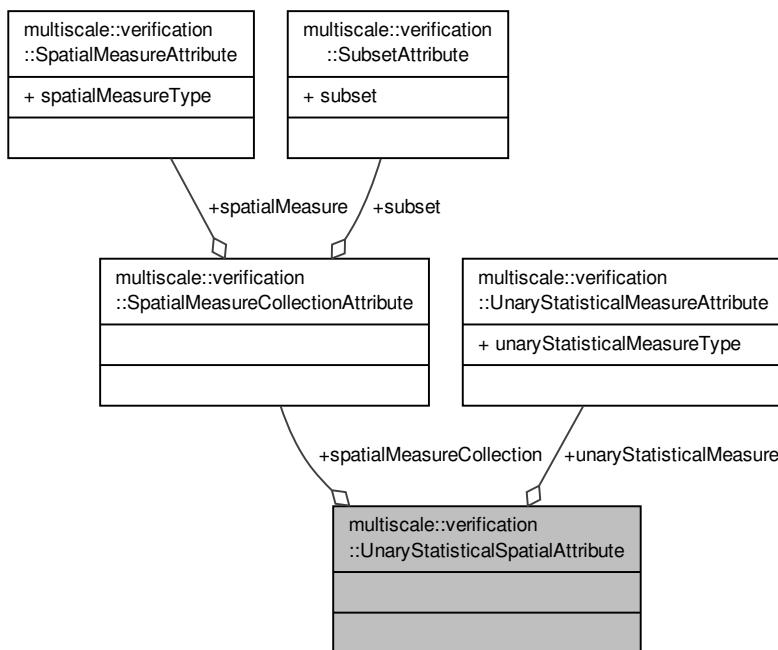
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnaryStatisticalNumericAttribute.hpp](#)

## 7.224 multiscale::verification::UnaryStatisticalSpatialAttribute Class Reference

Class for representing a unary statistical spatial attribute.

```
#include <UnaryStatisticalSpatialAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryStatisticalSpatialAttribute:



### Public Attributes

- [UnaryStatisticalMeasureAttribute unaryStatisticalMeasure](#)
- [SpatialMeasureCollectionAttribute spatialMeasureCollection](#)

#### 7.224.1 Detailed Description

Class for representing a unary statistical spatial attribute.

Definition at line 15 of file `UnaryStatisticalSpatialAttribute.hpp`.

#### 7.224.2 Member Data Documentation

### 7.224.2.1 SpatialMeasureCollectionAttribute multiscale::verification::UnaryStatisticalSpatialAttribute::spatialMeasure← Collection

The considered spatial measure collection

Definition at line 20 of file UnaryStatisticalSpatialAttribute.hpp.

Referenced by multiscale::verification::NumericVisitor::operator()().

### 7.224.2.2 UnaryStatisticalMeasureAttribute multiscale::verification::UnaryStatisticalSpatialAttribute::unaryStatistical← Measure

The unary statistical measure

Definition at line 19 of file UnaryStatisticalSpatialAttribute.hpp.

Referenced by multiscale::verification::NumericVisitor::operator()().

The documentation for this class was generated from the following file:

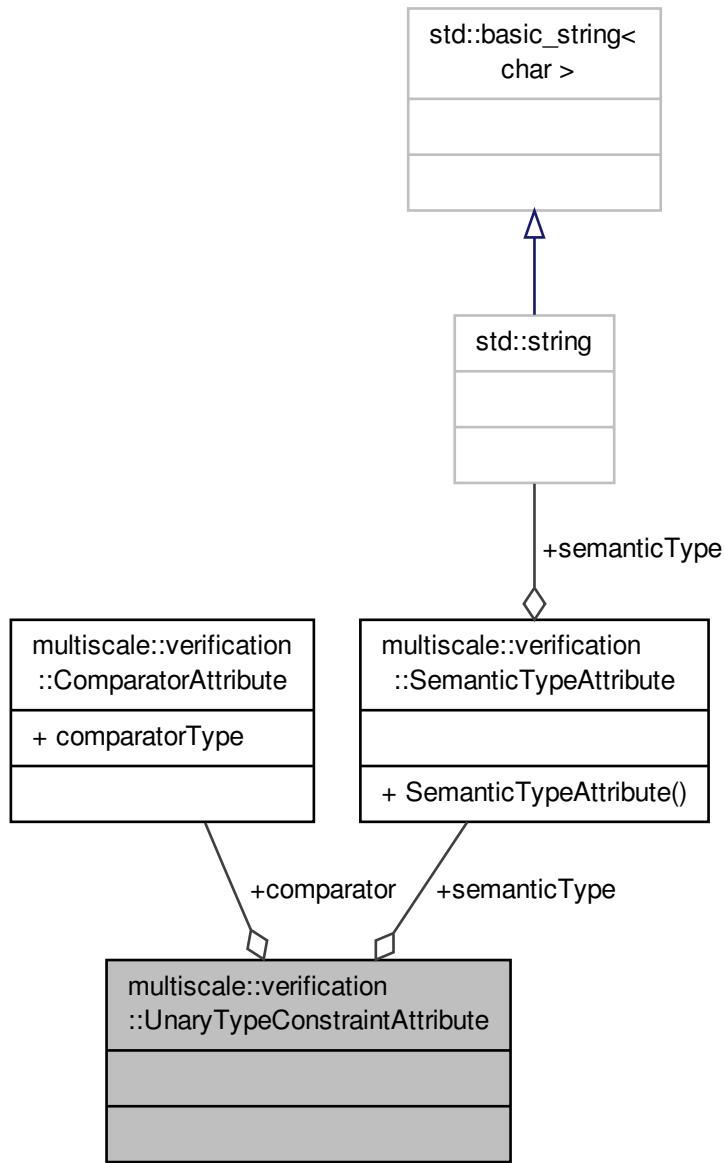
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnaryStatisticalSpatialAttribute.hpp](#)

## 7.225 multiscale::verification::UnaryTypeConstraintAttribute Class Reference

Class for representing a "unary" type constraint attribute.

```
#include <UnaryTypeConstraintAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnaryTypeConstraintAttribute:



## Public Attributes

- `ComparatorAttribute comparator`
- `SemanticTypeAttribute semanticType`

### 7.225.1 Detailed Description

Class for representing a "unary" type constraint attribute.

Definition at line 15 of file `UnaryTypeConstraintAttribute.hpp`.

## 7.225.2 Member Data Documentation

### 7.225.2.1 ComparatorAttribute multiscale::verification::UnaryTypeConstraintAttribute::comparator

The comparator

Definition at line 19 of file UnaryTypeConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

### 7.225.2.2 SemanticTypeAttribute multiscale::verification::UnaryTypeConstraintAttribute::semanticType

The considered semantic type

Definition at line 20 of file UnaryTypeConstraintAttribute.hpp.

Referenced by multiscale::verification::ConstraintVisitor::operator()().

The documentation for this class was generated from the following file:

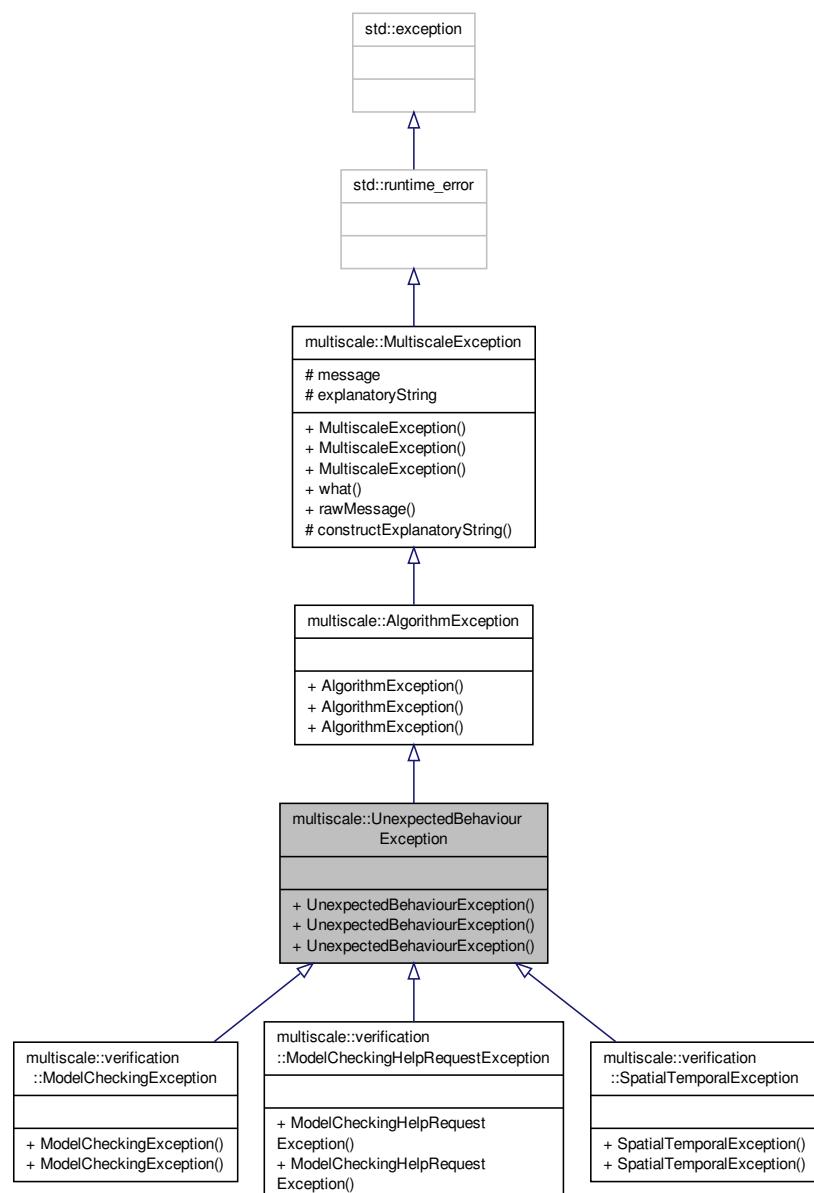
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnaryTypeConstraintAttribute.hpp](#)

## 7.226 multiscale::UnexpectedBehaviourException Class Reference

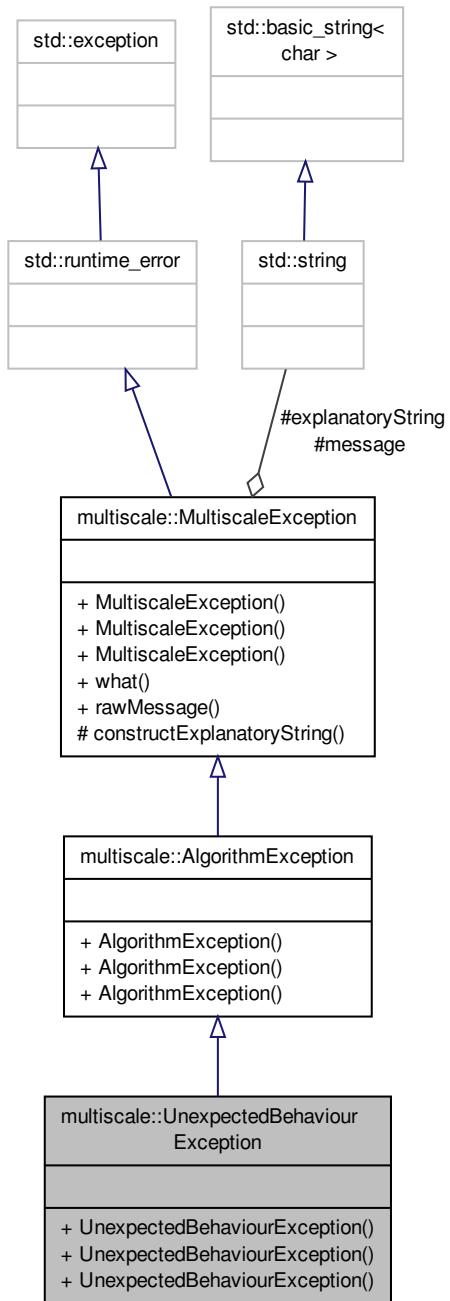
Class for representing unexpected behaviour exceptions.

```
#include <UnexpectedBehaviourException.hpp>
```

Inheritance diagram for multiscale::UnexpectedBehaviourException:



Collaboration diagram for multiscale::UnexpectedBehaviourException:



## Public Member Functions

- [UnexpectedBehaviourException \(\)](#)
- [UnexpectedBehaviourException \(const std::string &file, int line, const std::string &msg\)](#)
- [UnexpectedBehaviourException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.226.1 Detailed Description

Class for representing unexpected behaviour exceptions.

Definition at line 12 of file UnexpectedBehaviourException.hpp.

### 7.226.2 Constructor & Destructor Documentation

#### 7.226.2.1 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException( ) [inline]

Definition at line 16 of file UnexpectedBehaviourException.hpp.

#### 7.226.2.2 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]

Definition at line 18 of file UnexpectedBehaviourException.hpp.

#### 7.226.2.3 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]

Definition at line 23 of file UnexpectedBehaviourException.hpp.

The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/[UnexpectedBehaviourException.hpp](#)

## 7.227 multiscale::verification::UnexpectedErrorHandler Struct Reference

Structure for defining the error handler for unexpected token errors.

```
#include <UnexpectedErrorHandler.hpp>
```

Collaboration diagram for multiscale::verification::UnexpectedErrorHandler:

|                                                      |
|------------------------------------------------------|
| multiscale::verification<br>::UnexpectedErrorHandler |
|                                                      |
| + operator()<br>- getExpectedTokenAsString()         |

## Classes

- struct [result](#)

*Structure for specifying the type of the result.*

## Public Member Functions

- template<typename Iterator>  
void [operator\(\)](#) (qi::info const &expectedToken, Iterator errorPosition, Iterator last) const  
*Overloaded operator.*

## Private Member Functions

- std::string [getExpectedTokenAsString](#) (qi::info const &expectedToken) const  
*Convert the expected token to a string.*

### 7.227.1 Detailed Description

Structure for defining the error handler for unexpected token errors.

Definition at line 17 of file UnexpectedErrorHandler.hpp.

### 7.227.2 Member Function Documentation

- 7.227.2.1 std::string multiscale::verification::UnexpectedErrorHandler::getExpectedTokenAsString ( qi::info const & *expectedToken* ) const [inline], [private]

Convert the expected token to a string.

Convert the expected token to a string and remove enclosing quotes

#### Parameters

|                      |                                        |
|----------------------|----------------------------------------|
| <i>expectedToken</i> | The expected token (not a std::string) |
|----------------------|----------------------------------------|

Definition at line 46 of file UnexpectedErrorHandler.hpp.

Referenced by [operator\(\)\(\)](#).

- 7.227.2.2 template<typename Iterator> void multiscale::verification::UnexpectedErrorHandler::operator() ( qi::info const & *expectedToken*, Iterator *errorPosition*, Iterator *last* ) const [inline]

Overloaded operator.

#### Parameters

|                      |                                           |
|----------------------|-------------------------------------------|
| <i>expectedToken</i> | The expected token                        |
| <i>errorPosition</i> | Iterator pointing to the error position   |
| <i>last</i>          | Iterator pointing to the end of the query |

Definition at line 32 of file UnexpectedErrorHandler.hpp.

References [getExpectedTokenAsString\(\)](#).

The documentation for this struct was generated from the following file:

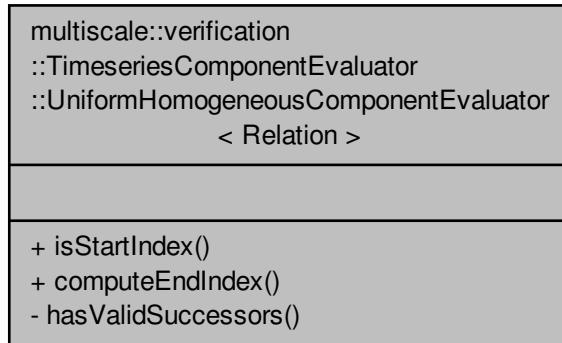
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/[UnexpectedErrorHandler.hpp](#)

7.228

~~multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation > Class Template Reference~~ 1037  
7.228 ~~multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneous~~ ←  
ComponentEvaluator< Relation > Class Template Reference

#include <TimeseriesComponentEvaluator.hpp>

Collaboration diagram for multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >:



## Public Member Functions

- bool `startIndex (std::size_t index, std::size_t nrOfValues, const std::vector< double > &values, const Relation &relation) const`

*Check if the given index is a uniform homogeneous component start index.*

- `std::size_t computeEndIndex (std::size_t startIndex, std::size_t nrOfValues, const std::vector< double > &values, const Relation &relation) const`

*Compute the uniform homogeneous component end index considering the given start index.*

## Private Member Functions

- bool `hasValidSuccessors (std::size_t index, std::size_t nrOfValues, const std::vector< double > &values, const Relation &relation) const`

*Check if a valid uniform homogeneous subcomponent starts from the given index.*

### 7.228.1 Detailed Description

~~template<typename Relation>class multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >~~

Definition at line 210 of file TimeseriesComponentEvaluator.hpp.

### 7.228.2 Member Function Documentation

---

7.228.2.1 template<typename Relation > std::size\_t multiscale::verification::TimeseriesComponentEvaluator::←  
UniformHomogeneousComponentEvaluator< Relation >::computeEndIndex ( std::size\_t startIndex,  
std::size\_t nrOfValues, const std::vector< double > & values, const Relation & relation ) const [inline]

Compute the uniform homogeneous component end index considering the given start index.

A value located at index  $i$  in the collection,  $1 < i < (\text{values.size() - 1})$ , is a nonuniform homogeneous component ending index if and only if  $\text{relation}(\text{values}[i - 2], \text{values}[i - 1])$ ,  $\text{relation}(\text{values}[i - 1], \text{values}[i])$ ,  $(\text{values}[i - 2] - \text{values}[i - 1]) = \text{values}[i - 1] - \text{values}[i]$  and  $\text{!relation}(\text{values}[i], \text{values}[i + 1])$  or  $(\text{values}[i - 1] - \text{values}[i]) != (\text{values}[i] - \text{values}[i + 1])$ , respectively  $\text{relation}(\text{values}[i - 2], \text{values}[i - 1])$ ,  $\text{relation}(\text{values}[i - 1], \text{values}[i])$  and  $(\text{values}[i - 2] - \text{values}[i - 1]) = (\text{values}[i - 1] - \text{values}[i])$  if  $i = (\text{values.size() - 1})$ .

#### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>startIndex</i> | The given start index           |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |
| <i>relation</i>   | The considered relation         |

Definition at line 265 of file TimeseriesComponentEvaluator.hpp.

References multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::isValidSuccessors(), and multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::isStartIndex().

7.228.2.2 template<typename Relation > bool multiscale::verification::TimeseriesComponentEvaluator::←  
UniformHomogeneousComponentEvaluator< Relation >::isValidSuccessors ( std::size\_t index,  
std::size\_t nrOfValues, const std::vector< double > & values, const Relation & relation ) const [inline],  
[private]

Check if a valid uniform homogeneous subcomponent starts from the given index.

#### Parameters

|                   |                                 |
|-------------------|---------------------------------|
| <i>index</i>      | The given index                 |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |
| <i>relation</i>   | The considered relation         |

Definition at line 294 of file TimeseriesComponentEvaluator.hpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::computeEndIndex(), and multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::isStartIndex().

7.228.2.3 template<typename Relation > bool multiscale::verification::TimeseriesComponentEvaluator::←  
UniformHomogeneousComponentEvaluator< Relation >::isStartIndex ( std::size\_t index, std::size\_t  
nrOfValues, const std::vector< double > & values, const Relation & relation ) const [inline]

Check if the given index is a uniform homogeneous component start index.

A value located at index  $i$  in the collection,  $0 < i < (\text{values.size() - 2})$ , is a uniform homogeneous component starting index if and only if  $(\text{!relation}(\text{values}[i - 1], \text{values}[i]))$  or  $(\text{values}[i - 2] - \text{values}[i - 1]) != (\text{values}[i - 1] - \text{values}[i])$ ,  $\text{relation}(\text{values}[i], \text{values}[i + 1])$ ,  $\text{relation}(\text{values}[i + 1], \text{values}[i + 2])$  and  $(\text{values}[i] - \text{values}[i + 1]) = (\text{values}[i + 1] - \text{values}[i + 2])$ , respectively  $\text{relation}(\text{values}[i], \text{values}[i + 1])$ ,  $\text{relation}(\text{values}[i + 1], \text{values}[i + 2])$ , and  $(\text{values}[i] - \text{values}[i + 1]) = (\text{values}[i + 1] - \text{values}[i + 2])$  if  $i = 0$ .

**Parameters**

|                   |                                 |
|-------------------|---------------------------------|
| <i>index</i>      | The given index                 |
| <i>nrOfValues</i> | The number of considered values |
| <i>values</i>     | The collection of values        |
| <i>relation</i>   | The considered relation         |

Definition at line 228 of file TimeseriesComponentEvaluator.hpp.

References multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::isValidSuccessors().

Referenced by multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >::computeEndIndex().

The documentation for this class was generated from the following file:

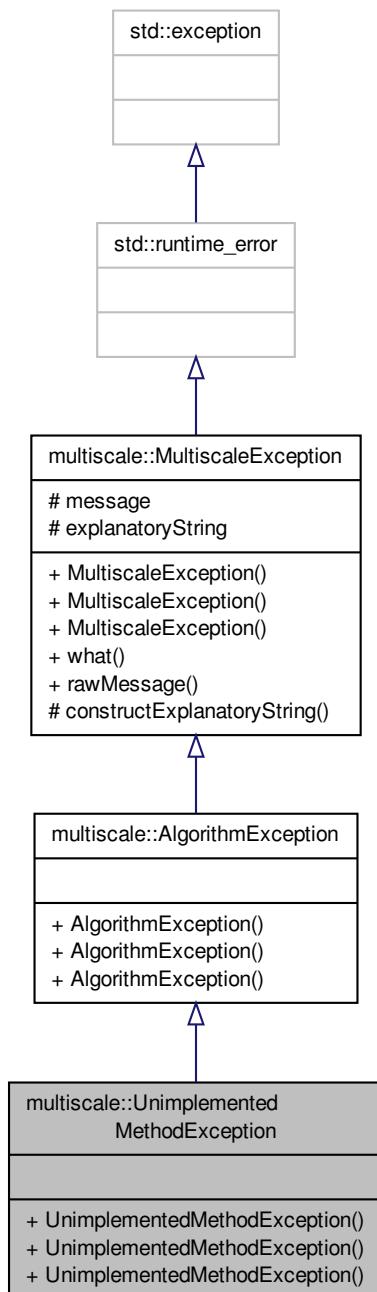
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp

## 7.229 multiscale::UnimplementedMethodException Class Reference

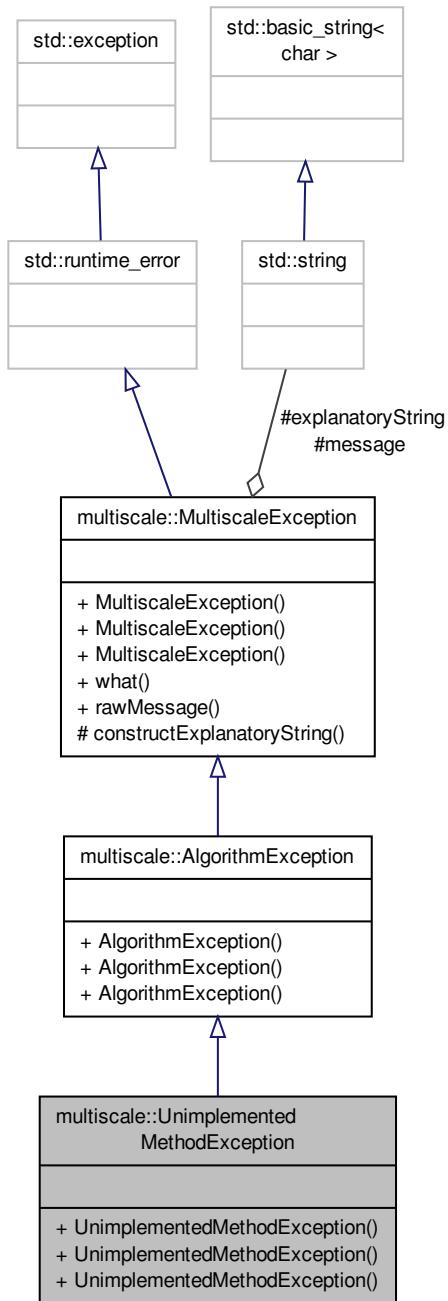
Class for representing unimplemented method exceptions.

```
#include <UnimplementedMethodException.hpp>
```

Inheritance diagram for multiscale::UnimplementedMethodException:



Collaboration diagram for multiscale::UnimplementedMethodException:



## Public Member Functions

- [UnimplementedMethodException \(\)](#)
- [UnimplementedMethodException \(const std::string &file, int line, const std::string &msg\)](#)
- [UnimplementedMethodException \(const std::string &file, int line, const char \\*msg\)](#)

## Additional Inherited Members

### 7.229.1 Detailed Description

Class for representing unimplemented method exceptions.

Definition at line 12 of file UnimplementedMethodException.hpp.

### 7.229.2 Constructor & Destructor Documentation

**7.229.2.1 multiscale::UnimplementedMethodException::UnimplementedMethodException( ) [inline]**

Definition at line 16 of file UnimplementedMethodException.hpp.

**7.229.2.2 multiscale::UnimplementedMethodException::UnimplementedMethodException( const std::string & *file*, int *line*, const std::string & *msg* ) [inline], [explicit]**

Definition at line 18 of file UnimplementedMethodException.hpp.

**7.229.2.3 multiscale::UnimplementedMethodException::UnimplementedMethodException( const std::string & *file*, int *line*, const char \* *msg* ) [inline], [explicit]**

Definition at line 23 of file UnimplementedMethodException.hpp.

The documentation for this class was generated from the following file:

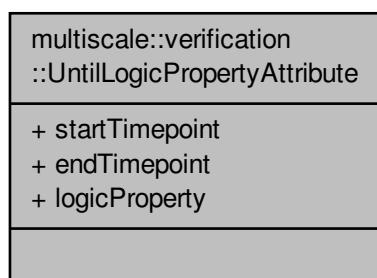
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/[UnimplementedMethodException.hpp](#)

## 7.230 multiscale::verification::UntilLogicPropertyAttribute Class Reference

Class for representing an "until" logic property attribute.

```
#include <UntilLogicPropertyAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UntilLogicPropertyAttribute:



## Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [LogicPropertyAttributeType logicProperty](#)

### 7.230.1 Detailed Description

Class for representing an "until" logic property attribute.

Definition at line 14 of file UntilLogicPropertyAttribute.hpp.

### 7.230.2 Member Data Documentation

#### 7.230.2.1 unsigned long multiscale::verification::UntilLogicPropertyAttribute::endTimepoint

The considered end timepoint

Definition at line 19 of file UntilLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

#### 7.230.2.2 LogicPropertyAttributeType multiscale::verification::UntilLogicPropertyAttribute::logicProperty

The logic property following the "until" operator

Definition at line 20 of file UntilLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

#### 7.230.2.3 unsigned long multiscale::verification::UntilLogicPropertyAttribute::startTimepoint

The considered start timepoint

Definition at line 18 of file UntilLogicPropertyAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

The documentation for this class was generated from the following file:

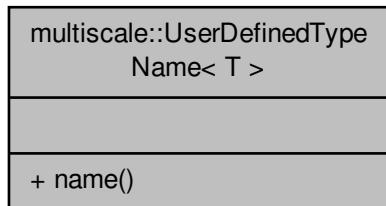
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UntilLogicPropertyAttribute.hpp](#)

## 7.231 multiscale::UserDefinedTypeName< T > Class Template Reference

Class for representing a user defined type name.

```
#include <UserDefinedTypeName.hpp>
```

Collaboration diagram for multiscale::UserDefinedTypeName< T >:



## Static Public Member Functions

- static std::string [name \(\)](#)

*Retrieve the name of the given type.*

### 7.231.1 Detailed Description

`template<typename T>class multiscale::UserDefinedTypeName< T >`

Class for representing a user defined type name.

Definition at line 12 of file `UserDefinedTypeName.hpp`.

### 7.231.2 Member Function Documentation

**7.231.2.1 `template<typename T > static std::string multiscale::UserDefinedTypeName< T >::name ( )`**  
`[inline], [static]`

Retrieve the name of the given type.

Definition at line 17 of file `UserDefinedTypeName.hpp`.

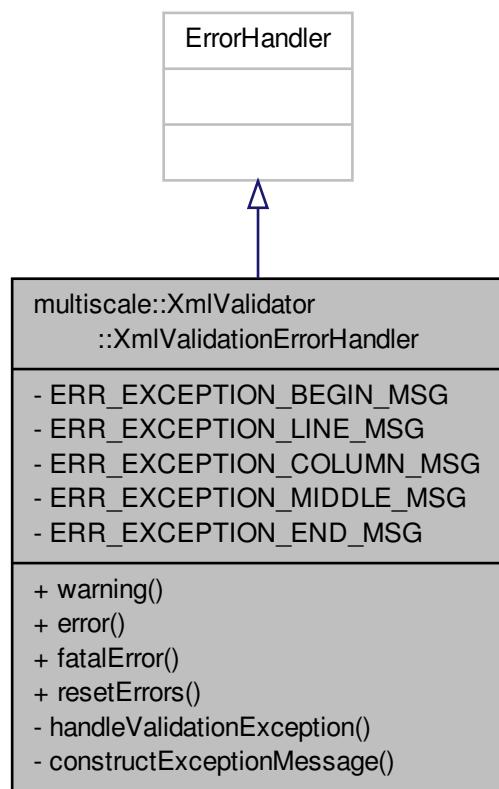
The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/[UserDefinedTypeName.hpp](#)

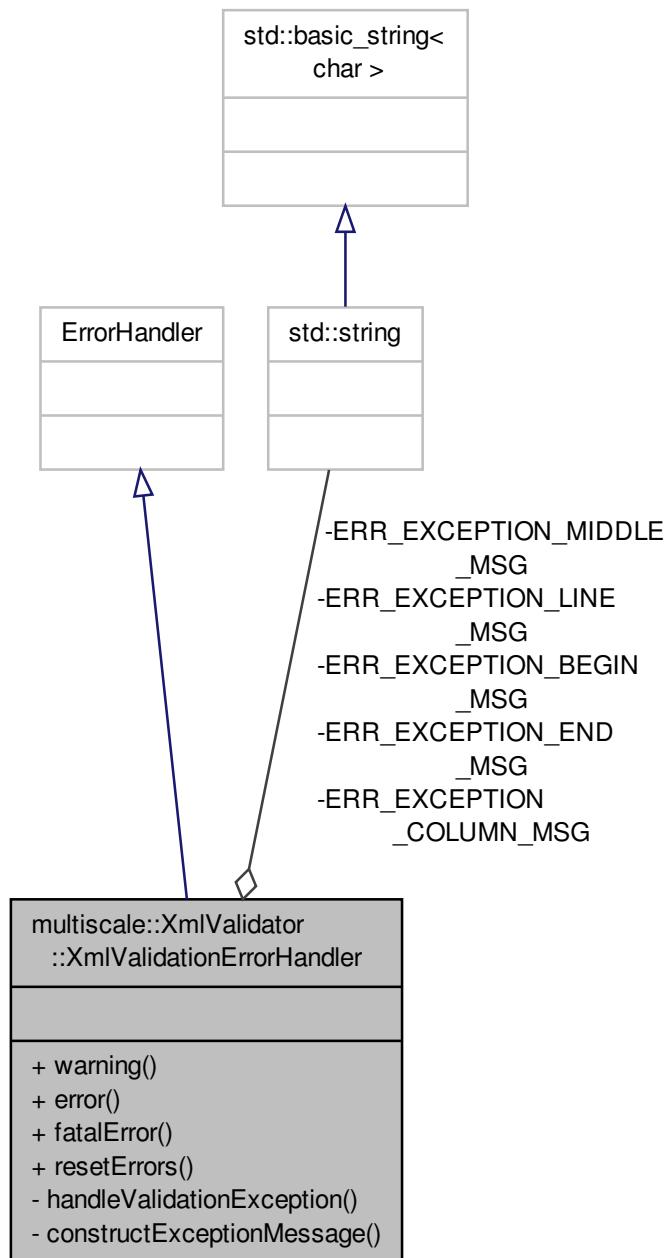
## 7.232 multiscale::XmlValidator::XmlValidationErrorHandler Class Reference

Class used for handling errors during the xml file validation process.

Inheritance diagram for multiscale::XmlValidator::XmlValidationErrorHandler:



Collaboration diagram for multiscale::XmlValidator::XmlValidationErrorHandler:



## Public Member Functions

- void **warning** (const SAXParseException &ex) override  
*Handle warning messages.*
- void **error** (const SAXParseException &ex) override  
*Handle recoverable error messages.*
- void **fatalError** (const SAXParseException &ex) override

- Handle non-recoverable error messages.*
- void [resetErrors \(\) override](#)  
*Reinitialise the error handler.*

### Private Member Functions

- void [handleValidationException \(const SAXParseException &ex\)](#)  
*Handle the exception thrown during the validation process.*
- std::string [constructExceptionMessage \(const SAXParseException &ex\)](#)  
*Construct the exception message for the given exception.*

### Static Private Attributes

- static const std::string [ERR\\_EXCEPTION\\_BEGIN\\_MSG](#) = "An [error](#) occurred at "
- static const std::string [ERR\\_EXCEPTION\\_LINE\\_MSG](#) = "line "
- static const std::string [ERR\\_EXCEPTION\\_COLUMN\\_MSG](#) = ", column "
- static const std::string [ERR\\_EXCEPTION\\_MIDDLE\\_MSG](#) = " and the [error](#) message is \\"
- static const std::string [ERR\\_EXCEPTION\\_END\\_MSG](#) = "\"."

### 7.232.1 Detailed Description

Class used for handling errors during the xml file validation process.

Definition at line 111 of file XmlValidator.hpp.

### 7.232.2 Member Function Documentation

#### 7.232.2.1 std::string XmlValidator::XmlValidationErrorHandler::constructExceptionMessage ( const SAXParseException & ex ) [private]

Construct the exception message for the given exception.

##### Parameters

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <code>ex</code> | The exception thrown during the validation process |
|-----------------|----------------------------------------------------|

Definition at line 108 of file XmlValidator.cpp.

References multiscale::StringManipulator::toString().

#### 7.232.2.2 void XmlValidator::XmlValidationErrorHandler::error ( const SAXParseException & ex ) [override]

Handle recoverable error messages.

##### Parameters

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <code>ex</code> | The exception thrown during the validation process |
|-----------------|----------------------------------------------------|

Definition at line 92 of file XmlValidator.cpp.

#### 7.232.2.3 void XmlValidator::XmlValidationErrorHandler::fatalError ( const SAXParseException & ex ) [override]

Handle non-recoverable error messages.

**Parameters**

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <code>ex</code> | The exception thrown during the validation process |
|-----------------|----------------------------------------------------|

Definition at line 96 of file XmlValidator.cpp.

**7.232.2.4 void XmlValidator::XmlValidationErrorHandler::handleValidationException ( const SAXParseException & ex ) [private]**

Handle the exception thrown during the validation process.

**Parameters**

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <code>ex</code> | The exception thrown during the validation process |
|-----------------|----------------------------------------------------|

Definition at line 102 of file XmlValidator.cpp.

References [MS\\_throw](#).

Referenced by [warning\(\)](#).

**7.232.2.5 void XmlValidator::XmlValidationErrorHandler::resetErrors ( ) [override]**

Reinitialise the error handler.

Definition at line 100 of file XmlValidator.cpp.

**7.232.2.6 void XmlValidator::XmlValidationErrorHandler::warning ( const SAXParseException & ex ) [override]**

Handle warning messages.

**Parameters**

|                 |                                                    |
|-----------------|----------------------------------------------------|
| <code>ex</code> | The exception thrown during the validation process |
|-----------------|----------------------------------------------------|

Definition at line 88 of file XmlValidator.cpp.

References [handleValidationException\(\)](#).

## 7.232.3 Member Data Documentation

**7.232.3.1 const std::string XmlValidator::XmlValidationErrorHandler::ERR\_EXCEPTION\_BEGIN\_MSG = "An error occurred at " [static], [private]**

Definition at line 151 of file XmlValidator.hpp.

**7.232.3.2 const std::string XmlValidator::XmlValidationErrorHandler::ERR\_EXCEPTION\_COLUMN\_MSG = ", column " [static], [private]**

Definition at line 154 of file XmlValidator.hpp.

**7.232.3.3 const std::string XmlValidator::XmlValidationErrorHandler::ERR\_EXCEPTION\_END\_MSG = "\\". " [static], [private]**

Definition at line 157 of file XmlValidator.hpp.

```
7.232.3.4 const std::string XmlValidator::XmlValidationErrorHandler::ERR_EXCEPTION_LINE_MSG = "line " [static],
[private]
```

Definition at line 153 of file XmlValidator.hpp.

```
7.232.3.5 const std::string XmlValidator::XmlValidationErrorHandler::ERR_EXCEPTION_MIDDLE_MSG = " and the error
message is \\" [static], [private]
```

Definition at line 155 of file XmlValidator.hpp.

The documentation for this class was generated from the following files:

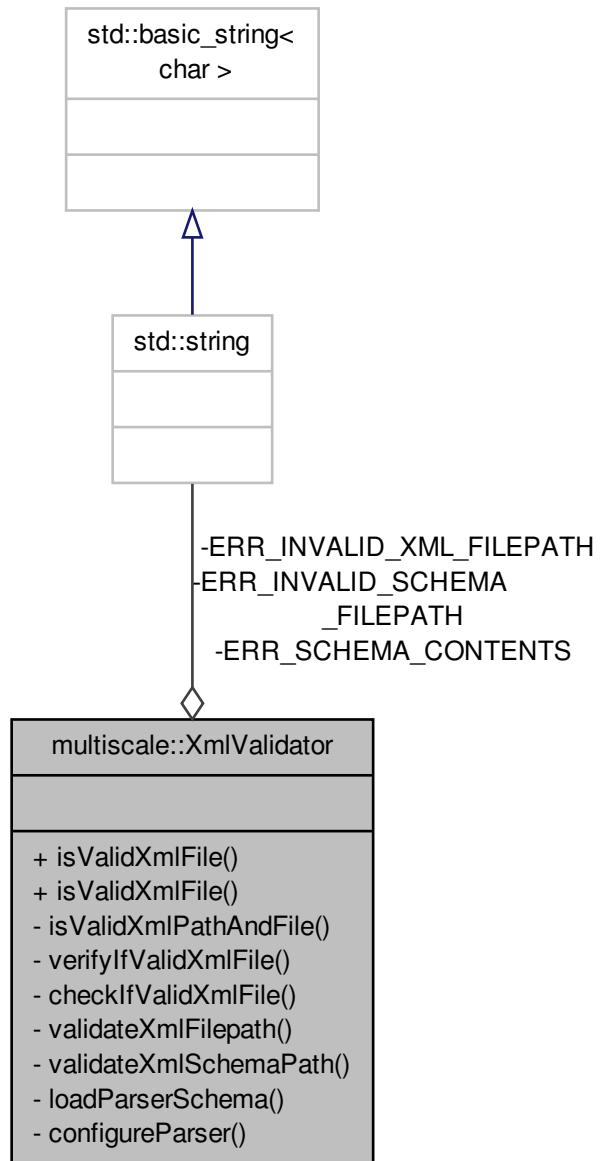
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[XmlValidator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[XmlValidator.cpp](#)

## 7.233 multiscale::XmlValidator Class Reference

Class used to validate xml files.

```
#include <XmlValidator.hpp>
```

## Collaboration diagram for multiscale::XmlValidator:



## Classes

- class `XmlValidationErrorHandler`

*Class used for handling errors during the xml file validation process.*

## Static Public Member Functions

- static bool `isValidXmlFile` (const std::string &xmlFilepath, const std::string &xmlSchemaPath)

*Check if the given xml file is valid considering the provided xml schema (xsd file)*

- static bool `isValidXmlFile` (const std::string &xmlFilepath, const std::string &xmlSchemaPath, std::string &xmlErrorMessage)

*Check if the given xml file is valid considering the provided xml schema (xsd file)*

## Static Private Member Functions

- static bool `isValidXmlPathAndFile` (const std::string &xmlFilepath, const std::string &xmlSchemaPath, std::string &xmlErrorMessage)
 

*Check if the given xml file is valid considering the provided xml schema (xsd file)*
- static bool `verifyIfValidXmlFile` (const std::string &xmlFilepath, const std::string &xmlSchemaPath, std::string &xmlErrorMessage)
 

*Check if the given xml file is valid considering the provided xml schema (xsd file)*
- static bool `checkIfValidXmlFile` (const std::string &xmlFilepath, const std::string &xmlSchemaPath)
 

*Check if the given xml file is valid considering the provided xml schema (xsd file)*
- static void `validateXmlFilepath` (const std::string &xmlFilepath)
 

*Check if the provided xml file path is valid.*
- static void `validateXmlSchemaPath` (const std::string &xmlSchemaPath)
 

*Check if the provided xml schema file path is valid.*
- static void `loadParserSchema` (const std::string &xmlSchemaPath, XercesDOMParser &parser)
 

*Load the xml schema using the given parser.*
- static void `configureParser` (XercesDOMParser &parser)
 

*Configure the given parser.*

## Static Private Attributes

- static const std::string `ERR_INVALID_XML_FILEPATH` = "The provided xml file path is invalid. Please change."
- static const std::string `ERR_INVALID_SCHEMA_FILEPATH` = "The provided xml schema file path is invalid. Please change."
- static const std::string `ERR_SCHEMA_CONTENTS` = "The provided xml schema is invalid. Please verify the xml schema contents."

### 7.233.1 Detailed Description

Class used to validate xml files.

Definition at line 18 of file XmlValidator.hpp.

### 7.233.2 Member Function Documentation

**7.233.2.1** `bool XmlValidator::checkIfValidXmlFile ( const std::string & xmlFilepath, const std::string & xmlSchemaPath )`  
`[static], [private]`

Check if the given xml file is valid considering the provided xml schema (xsd file)

The validation is performed using the Xerces C++ library.

#### Parameters

|                          |                          |
|--------------------------|--------------------------|
| <code>xmlFilepath</code> | The path to the xml file |
|--------------------------|--------------------------|

|                            |                                 |
|----------------------------|---------------------------------|
| <code>xmlSchemaPath</code> | The path to the xml schema file |
|----------------------------|---------------------------------|

Definition at line 48 of file XmlValidator.cpp.

References configureParser(), and loadParserSchema().

Referenced by verifyIfValidXmlFile().

#### 7.233.2.2 void XmlValidator::configureParser ( XercesDOMParser & parser ) [static], [private]

Configure the given parser.

##### Parameters

|                     |                          |
|---------------------|--------------------------|
| <code>parser</code> | The given xml DOM parser |
|---------------------|--------------------------|

Definition at line 77 of file XmlValidator.cpp.

Referenced by checkIfValidXmlFile().

#### 7.233.2.3 bool XmlValidator::isValidXmlFile ( const std::string & xmlFilepath, const std::string & xmlSchemaPath ) [static]

Check if the given xml file is valid considering the provided xml schema (xsd file)

The validation is performed using the Xerces C++ library.

##### Parameters

|                            |                                 |
|----------------------------|---------------------------------|
| <code>xmlFilepath</code>   | The path to the xml file        |
| <code>xmlSchemaPath</code> | The path to the xml schema file |

Definition at line 12 of file XmlValidator.cpp.

Referenced by checkIfValidXmlFile(), multiscale::verification::SpatialTemporalDataReader::isValidInputFile(), and multiscale::verification::TypeSemanticsTable::readFromValidFile().

#### 7.233.2.4 bool XmlValidator::isValidXmlFile ( const std::string & xmlFilepath, const std::string & xmlSchemaPath, std::string & xmlErrorMessage ) [static]

Check if the given xml file is valid considering the provided xml schema (xsd file)

The validation is performed using the Xerces C++ library.

In case the xml file is not valid the error message is retrieved in xmlErrorMessage.

##### Parameters

|                              |                                                            |
|------------------------------|------------------------------------------------------------|
| <code>xmlFilepath</code>     | The path to the xml file                                   |
| <code>xmlSchemaPath</code>   | The path to the xml schema file                            |
| <code>xmlErrorMessage</code> | The error message explaining why the xml file is not valid |

Definition at line 18 of file XmlValidator.cpp.

References isValidXmlPathAndFile().

#### 7.233.2.5 bool XmlValidator::isValidXmlPathAndFile ( const std::string & xmlFilepath, const std::string & xmlSchemaPath, std::string & xmlErrorMessage ) [static], [private]

Check if the given xml file is valid considering the provided xml schema (xsd file)

The validation is performed using the Xerces C++ library.

In case the xml file is not valid the error message is retrieved in `xmlErrorMessage`.

**Parameters**

|                         |                                                            |
|-------------------------|------------------------------------------------------------|
| <i>xmlFilepath</i>      | The path to the xml file                                   |
| <i>xmlSchemaPath</i>    | The path to the xml schema file                            |
| <i>xmlError Message</i> | The error message explaining why the xml file is not valid |

Definition at line 29 of file XmlValidator.cpp.

References validateXmlFilepath(), validateXmlSchemaPath(), and verifyIfValidXmlFile().

Referenced by isValidXmlFile().

**7.233.2.6 void XmlValidator::loadParserSchema ( const std::string & *xmlSchemaPath*, XercesDOMParser & *parser* ) [static], [private]**

Load the xml schema using the given parser.

**Parameters**

|                      |                                 |
|----------------------|---------------------------------|
| <i>parser</i>        | The given xml DOM parser        |
| <i>xmlSchemaPath</i> | The file path to the xml schema |

Definition at line 71 of file XmlValidator.cpp.

References ERR\_SCHEMA\_CONTENTS, and MS\_throw.

Referenced by checkIfValidXmlFile().

**7.233.2.7 void XmlValidator::validateXmlFilepath ( const std::string & *xmlFilepath* ) [static], [private]**

Check if the provided xml file path is valid.

**Parameters**

|                    |                          |
|--------------------|--------------------------|
| <i>xmlFilepath</i> | The path to the xml file |
|--------------------|--------------------------|

Definition at line 59 of file XmlValidator.cpp.

References ERR\_INVALID\_XML\_FILEPATH, multiscale::Filesystem::isValidFilePath(), and MS\_throw.

Referenced by isValidXmlPathAndFile().

**7.233.2.8 void XmlValidator::validateXmlSchemaPath ( const std::string & *xmlSchemaPath* ) [static], [private]**

Check if the provided xml schema file path is valid.

**Parameters**

|                      |                            |
|----------------------|----------------------------|
| <i>xmlSchemaPath</i> | The path to the xml schema |
|----------------------|----------------------------|

Definition at line 65 of file XmlValidator.cpp.

References ERR\_INVALID\_SCHEMA\_FILEPATH, multiscale::Filesystem::isValidFilePath(), and MS\_throw.

Referenced by isValidXmlPathAndFile().

**7.233.2.9 bool XmlValidator::verifyIfValidXmlFile ( const std::string & *xmlFilepath*, const std::string & *xmlSchemaPath*, std::string & *xmlErrorMessage* ) [static], [private]**

Check if the given xml file is valid considering the provided xml schema (xsd file)

The validation is performed using the Xerces C++ library.

In case the xml file is not valid the error message is retrieved in *xmlErrorMessage*.

**Parameters**

|                        |                                                            |
|------------------------|------------------------------------------------------------|
| <i>xmlFilepath</i>     | The path to the xml file                                   |
| <i>xmlSchemaPath</i>   | The path to the xml schema file                            |
| <i>xmlErrorMessage</i> | The error message explaining why the xml file is not valid |

Definition at line 37 of file XmlValidator.cpp.

References checkIfValidXmlFile(), and multiscale::MultiscaleException::rawMessage().

Referenced by isValidXmlPathAndFile().

### 7.233.3 Member Data Documentation

**7.233.3.1** `const std::string XmlValidator::ERR_INVALID_SCHEMA_FILEPATH = "The provided xml schema file path is invalid. Please change."` [static], [private]

Definition at line 104 of file XmlValidator.hpp.

Referenced by validateXmlSchemaPath().

**7.233.3.2** `const std::string XmlValidator::ERR_INVALID_XML_FILEPATH = "The provided xml file path is invalid. Please change."` [static], [private]

Definition at line 103 of file XmlValidator.hpp.

Referenced by validateXmlFilepath().

**7.233.3.3** `const std::string XmlValidator::ERR_SCHEMA_CONTENTS = "The provided xml schema is invalid. Please verify the xml schema contents."` [static], [private]

Definition at line 106 of file XmlValidator.hpp.

Referenced by loadParserSchema().

The documentation for this class was generated from the following files:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/[XmlValidator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[XmlValidator.cpp](#)



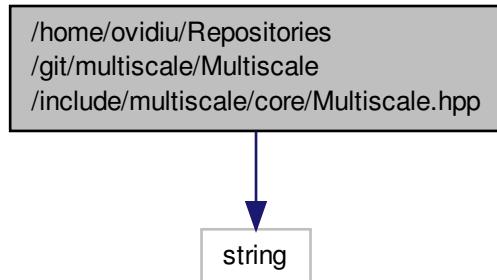
# Chapter 8

## File Documentation

### 8.1 config/mainpage.dox File Reference

### 8.2 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/Multiscale.hpp File Reference

```
#include <string>
Include dependency graph for Multiscale.hpp:
```



### Namespaces

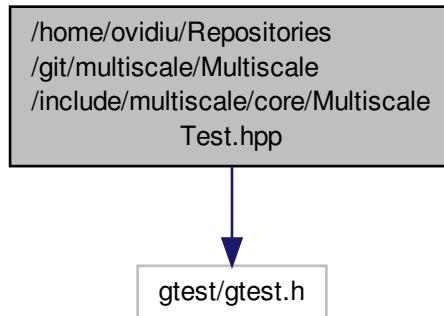
- multiscale

### Variables

- const int multiscale::EXEC\_SUCCESS\_CODE = 0
- const int multiscale::EXEC\_ERR\_CODE = 1
- const std::string multiscale::ERR\_MSG = "An error occurred: "
- const std::string multiscale::ERR\_UNDEFINED\_ENUM\_VALUE = "The provided enumeration value is invalid. Please use one of the available enumeration values instead."
- const std::string multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_BEGIN = "The provided index value ("
- const std::string multiscale::ERR\_INDEX\_OUT\_OF\_BOUNDS\_END = ") is invalid. Please change."

## 8.3 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/MultiscaleTest.hpp File Reference

```
#include "gtest/gtest.h"
Include dependency graph for MultiscaleTest.hpp:
```



### Classes

- class [multiscaletest::MultiscaleTest](#)

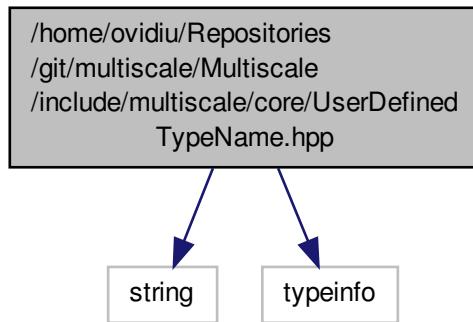
### Namespaces

- [multiscaletest](#)

## 8.4 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/UserDefinedTypeName.hpp File Reference

```
#include <string>
#include <typeinfo>
```

Include dependency graph for UserDefinedTypeName.hpp:



## Classes

- class [multiscale::UserDefinedTypeName< T >](#)  
*Class for representing a user defined type name.*

## Namespaces

- [multiscale](#)

## Macros

- [#define DEFINE\\_TYPE\\_NAME\(Type, Name\)](#)

### 8.4.1 Macro Definition Documentation

#### 8.4.1.1 #define DEFINE\_TYPE\_NAME( Type, Name )

##### Value:

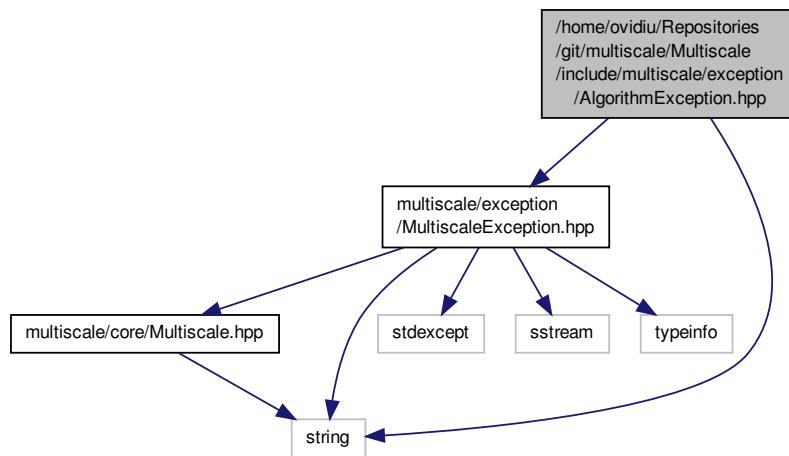
```
template <>
 class UserDefinedTypeName<Type> {
 public:
 static std::string name() {
 return Name;
 }
 };

```

Definition at line 25 of file UserDefinedTypeName.hpp.

## 8.5 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/AlgorithmException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include <string>
Include dependency graph for AlgorithmException.hpp:
```



### Classes

- class [multiscale::AlgorithmException](#)

*Class for representing algorithm exceptions.*

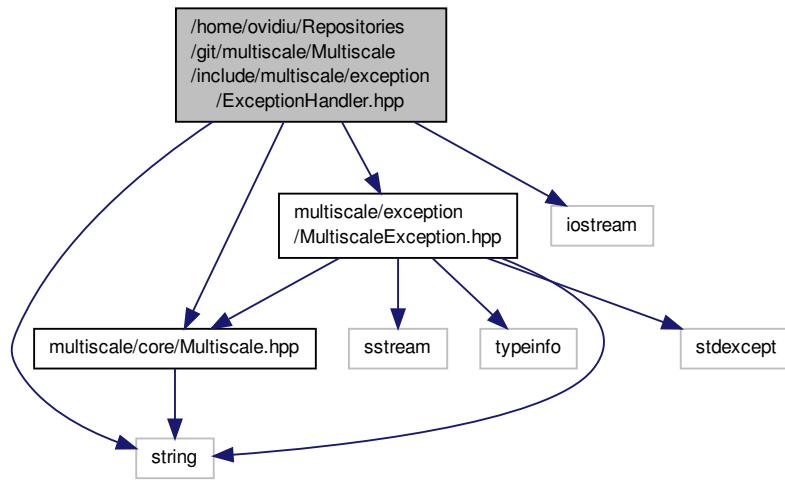
### Namespaces

- [multiscale](#)

## 8.6 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ExceptionHandler.hpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/exception/MultiscaleException.hpp"
#include <iostream>
#include <string>
```

Include dependency graph for ExceptionHandler.hpp:



## Classes

- class [multiscale::ExceptionHandler](#)

*Exception handler class.*

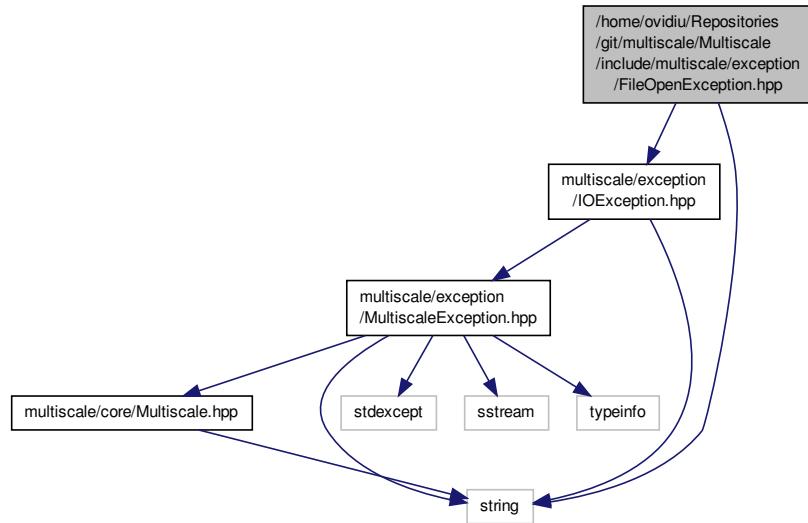
## Namespaces

- [multiscale](#)

## 8.7 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/← FileOpenException.hpp File Reference

```
#include "multiscale/exception/IOException.hpp"
#include <string>
```

Include dependency graph for FileOpenException.hpp:



## Classes

- class [multiscale::FileOpenException](#)

*Class for representing exceptions when opening a file.*

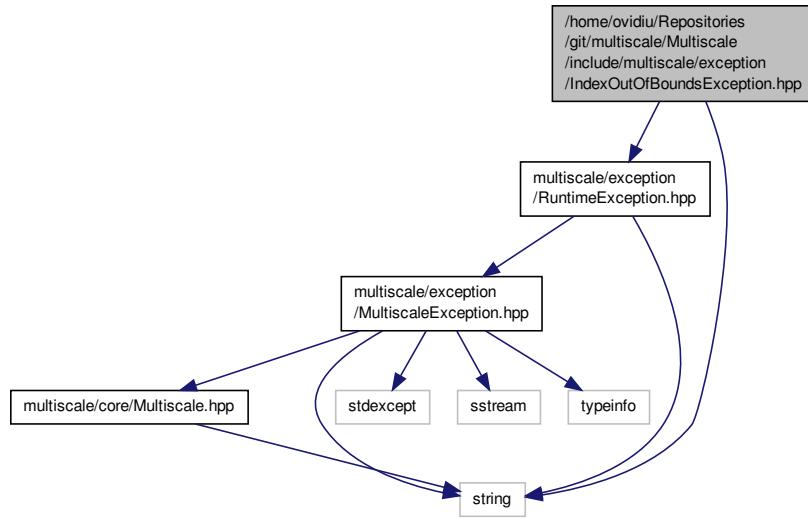
## Namespaces

- [multiscale](#)

## 8.8 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/IndexOutOfBoundsException.hpp File Reference

```
#include "multiscale/exception/RuntimeException.hpp"
#include <string>
```

Include dependency graph for IndexOutOfBoundsException.hpp:



## Classes

- class [multiscale::IndexOutOfBoundsException](#)

*Class for representing an index out of bounds exception.*

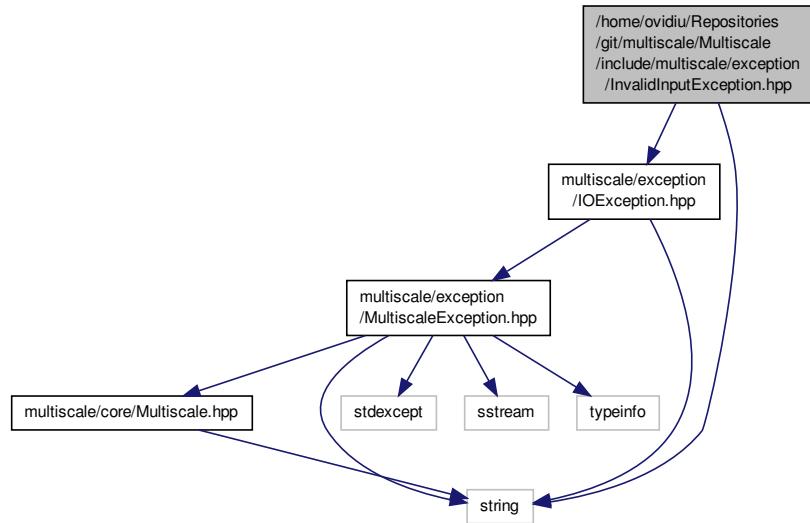
## Namespaces

- [multiscale](#)

## 8.9 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/InvalidInputException.hpp File Reference

```
#include "multiscale/exception/IOException.hpp"
#include <string>
```

Include dependency graph for InvalidInputException.hpp:



## Classes

- class [multiscale::InvalidInputException](#)

*Class for representing invalid input exceptions.*

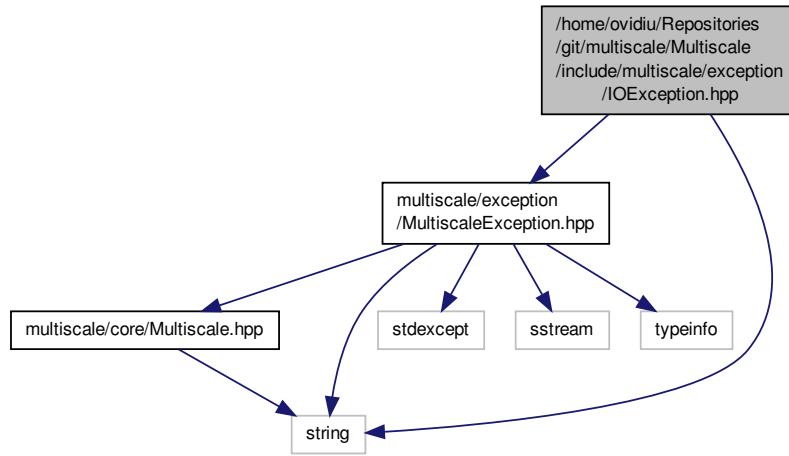
## Namespaces

- [multiscale](#)

## 8.10 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/I OException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include <string>
```

Include dependency graph for IOException.hpp:



## Classes

- class [multiscale::IOException](#)

*Class for representing input and output exceptions.*

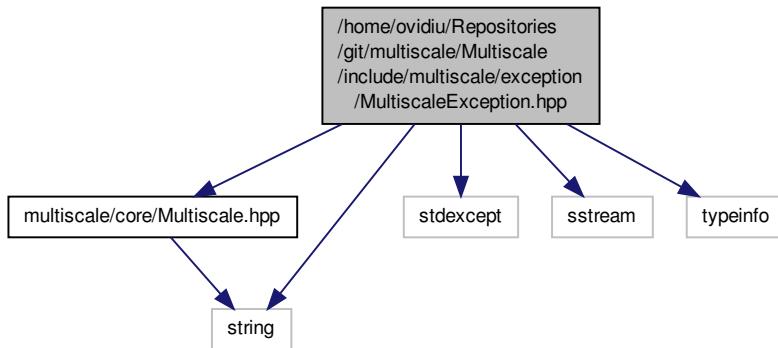
## Namespaces

- [multiscale](#)

## 8.11 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ MultiscaleException.hpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include <stdexcept>
#include <string>
#include <iostream>
#include <typeinfo>
```

Include dependency graph for MultiscaleException.hpp:



## Classes

- class [multiscale::MultiscaleException](#)

*Parent exception class for the project.*

## Namespaces

- [multiscale](#)

## Macros

- `#define MS_throw(ex, msg) (throw ex(__FILE__, __LINE__, msg))`
- `#define MS_throw_detailed(ex, startMsg, msg, endMsg) (throw ex(__FILE__, __LINE__, startMsg + msg + endMsg))`

### 8.11.1 Macro Definition Documentation

#### 8.11.1.1 #define MS\_throw( ex, msg ) (throw ex(\_\_FILE\_\_, \_\_LINE\_\_, msg))

Definition at line 11 of file MultiscaleException.hpp.

Referenced by multiscale::verification::TypeSemanticsTable::addSemanticTypeToTable(), multiscale::Numeric::applyOperation(), multiscale::verification::CommandLineModelChecking::areInvalidModelCheckingArguments(), multiscale::verification::CommandLineModelChecking::areModelCheckingTypeSpecificArgumentsPresent(), multiscale::verification::LogicPropertyVisitor::areSimilarValues(), multiscale::Numeric::combinations(), multiscale::video::RectangularEntityCsvToInputFilesConverter::computeCoordinate(), multiscale::verification::ChangeMeasureEvaluator::computeNumericMeasureValueChange(), multiscale::video::RectangularEntityCsvToInputFilesConverter::computeSimulationTime(), multiscale::video::RectangularCsvToInputFilesConverter::computeSimulationTime(), multiscale::video::PolarCsvToInputFilesConverter::computeSimulationTime(), multiscale::StringManipulator::convert(), multiscale::analysis::MatFactory::create(), multiscale::verification::SpatialTemporalDataReader::createDerivedSpatialEntity(), multiscale::analysis::Detector::detect(), multiscale::verification::ApproximateBayesianModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::BayesianModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::StatisticalModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::TimeseriesComponentEvaluator::evaluate(), multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluate(), multiscale::verification::AbstractSyntaxTree::evaluate(),

**File Reference**

multiscale::verification::NumericMeasureCollectionVisitor::evaluateHomogeneousHomogeneousTimeseries(),  
 multiscale::verification::SubsetVisitor::evaluateSubsetOperation(),    multiscale::verification::NumericMeasure::  
 CollectionVisitor::evaluateTimeseriesTimeseriesComponent(),    multiscale::OperatingSystem::executeProgram::  
 AndVerifyPath(),    multiscale::MinEnclosingTriangleFinder::find(),    multiscale::MinEnclosingTriangleFinder::find::  
 VertexCOnSideB(),    multiscale::verification::spatialmeasure::getMaxValidSpatialMeasureValue(),    multiscale::  
 ::verification::spatialmeasure::getMinValidSpatialMeasureValue(),    multiscale::verification::SpatialTemporalData::  
 Reader::getNextSpatialTemporalTrace(),    multiscale::verification::SpatialTemporalDataReader::getRandomValid::  
 UnprocessedInputFilepath(),    multiscale::verification::TypeSemanticsTable::getTypeOfSemanticCriteriaValues(),  
 multiscale::verification::ParserGrammarExceptionHandler::handleExtraInputException(),    multiscale::verification::  
 CommandLineModelChecking::handleHelpRequest(),    multiscale::verification::ParserGrammarExceptionHandler::  
 ::handleProbabilityException(),    multiscale::verification::ParserGrammarExceptionHandler::handleUnexpected::  
 TokenException(),    multiscale::verification::ParserGrammarExceptionHandler::handleUnparseableInputException(),  
 multiscale::XmlValidator::XmlValidationErrorHandler::handleValidationException(),    multiscale::verification::  
 CommandLineModelChecking::initialise(),    multiscale::verification::CommandLineModelChecking::initialiseModel::  
 Checker(),    multiscale::video::RectangularEntityCsvToInputFilesConverter::initInputFile(),    multiscale::analysis::  
 MatFactory::initInputFile(),    multiscale::video::PolarCsvToInputFilesConverter::initInputFile(),    multiscale::video::  
 ::RectangularCsvToInputFilesConverter::initInputFile(),    multiscale::video::PolarCsvToInputFilesConverter::init::  
 MaximumConcentration(),    multiscale::video::RectangularCsvToInputFilesConverter::initMaximumConcentration(),  
 multiscale::analysis::CircularMatFactory::isValidViewerImage(),    multiscale::analysis::RectangularMatFactory::  
 ::isValidViewerImage(),    multiscale::XmlValidator::loadParserSchema(),    multiscale::Filesystem::nativeFormat::  
 FilePath(),    multiscale::verification::SpatialTemporalTrace::nextTimePointValueForLastTimePoint(),    multiscale::  
 ::verification::TimeseriesComponentVisitor::operator()(),    multiscale::analysis::SimulationClusterDetector::output::  
 ClusterShape(),    multiscale::analysis::Detector::outputResultsToCsvFile(),    multiscale::analysis::CircularMat::  
 Factory::processConcentrations(),    multiscale::analysis::RectangularMatFactory::processConcentrations(),    multiscale::  
 ::verification::TemporalDataReader::processLineTokens(),    multiscale::verification::TemporalDataReader::read(),  
 multiscale::verification::TypeSemanticsTable::read(),    multiscale::video::CartesianToConcentrationsConverter::  
 ::readConcentrations(),    multiscale::video::CartesianToPolarConverter::readConcentrations(),    multiscale::verification::  
 ::TypeSemanticsTable::readFromValidFile(),    multiscale::verification::TemporalDataReader::readFromValid::  
 OpenedInputFile(),    multiscale::video::CartesianToConcentrationsConverter::readHeaderLine(),    multiscale::  
 ::video::CartesianToPolarConverter::readHeaderLine(),    multiscale::video::CartesianToConcentrationsConverter::  
 ::readInputData(),    multiscale::video::CartesianToPolarConverter::readInputData(),    multiscale::verification::  
 LogicPropertyDataReader::readLogicPropertiesFromFile(),    multiscale::verification::LogicPropertyDataReader::  
 ::readLogicPropertiesFromValidFilepath(),    multiscale::verification::CommandLineModelChecking::remove::  
 ModelCheckingTypeSpecificArguments(),    multiscaletest::TraceEvaluationTest::RunTest(),    multiscale::analysis::  
 ::SpatialEntityPseudo3D::typeAsString(),    multiscale::MinEnclosingTriangleFinder::updateSideB(),    multiscale::  
 ::verification::BayesianModelChecker::validateBayesFactorThreshold(),    multiscale::video::RectangularEntityCsv::  
 ToInputFilesConverter::validateCoordinate(),    multiscale::analysis::Silhouette::validateElementIndex(),    multiscale::  
 ::video::RectangularEntityCsvToInputFilesConverter::validateEntitiesGrid(),    multiscale::verification::Spatial::  
 TemporalDataReader::validateFolderPath(),    multiscale::video::RectangularEntityCsvToInputFilesConverter::  
 ::validateInput(),    multiscale::verification::ApproximateProbabilisticModelChecker::validateInput(),    multiscale::  
 ::video::PolarCsvToInputFilesConverter::validateInput(),    multiscale::video::RectangularCsvToInputFilesConverter::  
 ::validateInput(),    multiscale::video::RectangularEntityCsvToInputFilesConverter::validateInputLine(),    multiscale::  
 ::video::PolarCsvToInputFilesConverter::validateInputLine(),    multiscale::video::RectangularCsvToInputFiles::  
 Converter::validateInputLine(),    multiscale::analysis::Region::validateInputValues(),    multiscale::analysis::Entity::  
 ::validateInputValues(),    multiscale::video::RectangularEntityCsvToInputFilesConverter::validateMaxNrOfEntities::  
 PerPosition(),    multiscale::BinomialDistribution::validateNrOfSuccesses(),    multiscale::verification::Temporal::  
 DataReader::validateObservableVariables(),    multiscale::analysis::Cluster::validateOriginDependentValues(),  
 multiscale::Distribution::validateProbability(),    multiscale::video::PolarCsvToInputFilesConverter::validateSelected::  
 ConcentrationIndex(),    multiscale::video::RectangularCsvToInputFilesConverter::validateSelectedConcentration::  
 Index(),    multiscale::BetaDistribution::validateShapeParameters(),    multiscale::verification::BayesianModel::  
 Checker::validateShapeParameters(),    multiscale::verification::ApproximateBayesianModelChecker::validate::  
 ShapeParameters(),    multiscale::video::RectangularEntityCsvToInputFilesConverter::validateSimulationTime(),  
 multiscale::verification::spatialmeasure::validateSpatialMeasureType(),    multiscale::verification::spatialmeasure::  
 ::validateSpatialMeasureTypeIndex(),    multiscale::verification::SpatialEntity::validateSpatialMeasureValue(),    multiscale::  
 ::verification::subsetspecific::validateSubsetSpecificType(),    multiscale::verification::subsetspecific::validate::  
 SubsetSpecificTypeIndex(),    multiscale::verification::StatisticalModelChecker::validateTypesErrors(),    multiscale::  
 ::verification::ApproximateBayesianModelChecker::validateVarianceThreshold(),    multiscale::XmlValidator::  
 ::validateXmlFilepath(), and multiscale::XmlValidator::validateXmlSchemaPath().

```
8.11.1.2 #define MS_throw_detailed(ex, startMsg, msg, endMsg) (throw ex(__FILE__, __LINE__, startMsg + msg + endMsg))
```

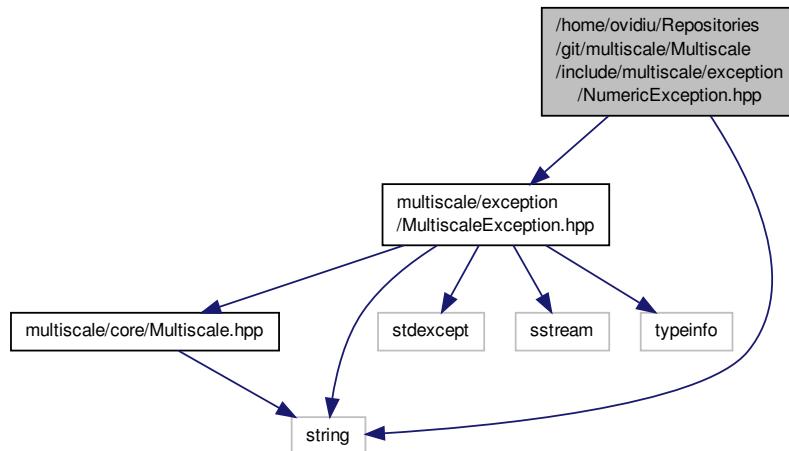
Definition at line 12 of file MultiscaleException.hpp.

Referenced by multiscale::verification::TimePoint::getNumericStateVariable(), multiscale::verification::SpatialTemporalTrace::validateIndex(), multiscale::Numeric::validateLogBase(), multiscale::Numeric::validateLogNumber(), multiscale::Numeric::validatePercentile(), multiscale::Numeric::validateQuartile(), multiscale::verification::SpatialTemporalTrace::validateTimePointValue(), and multiscale::verification::SpatialTemporalTrace::validateValue().

## 8.12 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ NumericException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include <string>
```

Include dependency graph for NumericException.hpp:



### Classes

- class [multiscale::NumericException](#)  
*Class for representing algorithm exceptions.*

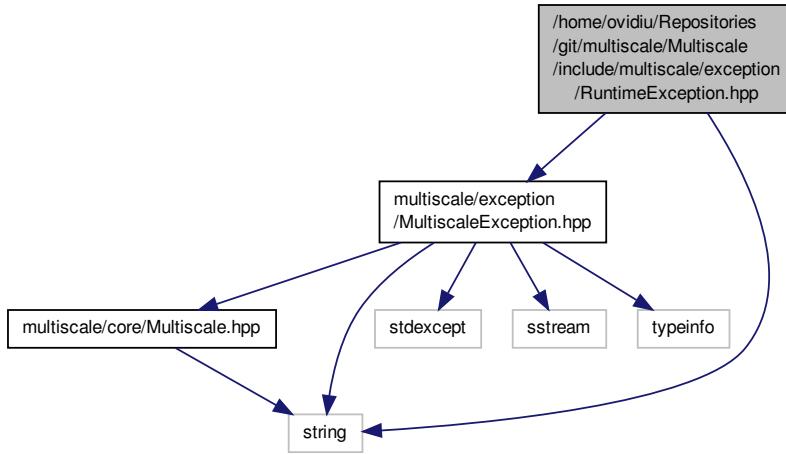
### Namespaces

- [multiscale](#)

## 8.13 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/ RuntimeException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include <string>
```

Include dependency graph for RuntimeException.hpp:



## Classes

- class [multiscale::RuntimeException](#)

*Class for representing runtime exceptions.*

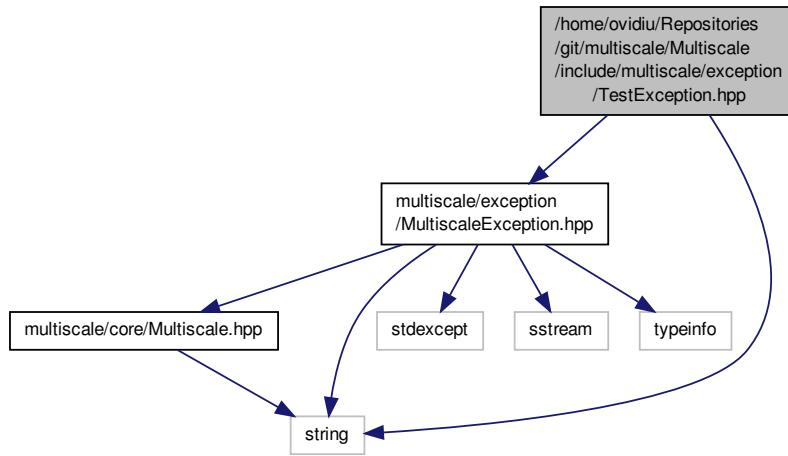
## Namespaces

- [multiscale](#)

## 8.14 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/[TestException.hpp](#) File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include <string>
```

Include dependency graph for TestException.hpp:



## Classes

- class [multiscale::TestException](#)

*Class for representing testing exceptions.*

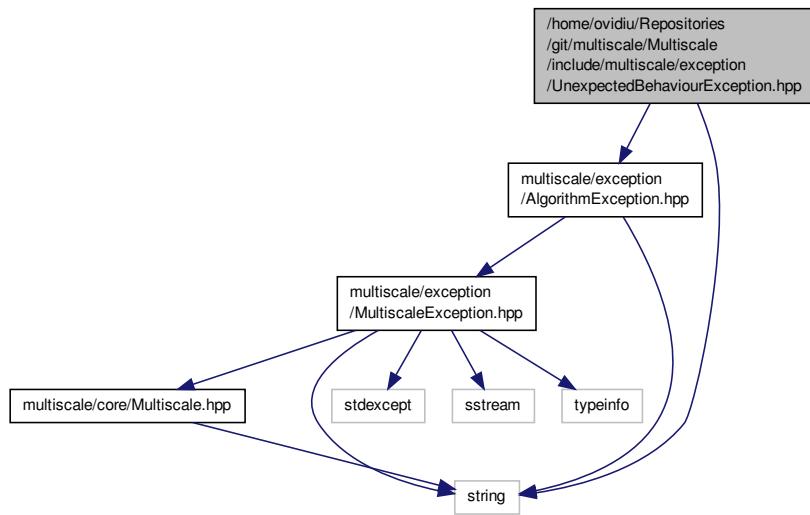
## Namespaces

- [multiscale](#)

## 8.15 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/UnexpectedBehaviourException.hpp File Reference

```
#include "multiscale/exception/AlgorithmException.hpp"
#include <string>
```

Include dependency graph for UnexpectedBehaviourException.hpp:



## Classes

- class [multiscale::UnexpectedBehaviourException](#)

*Class for representing unexpected behaviour exceptions.*

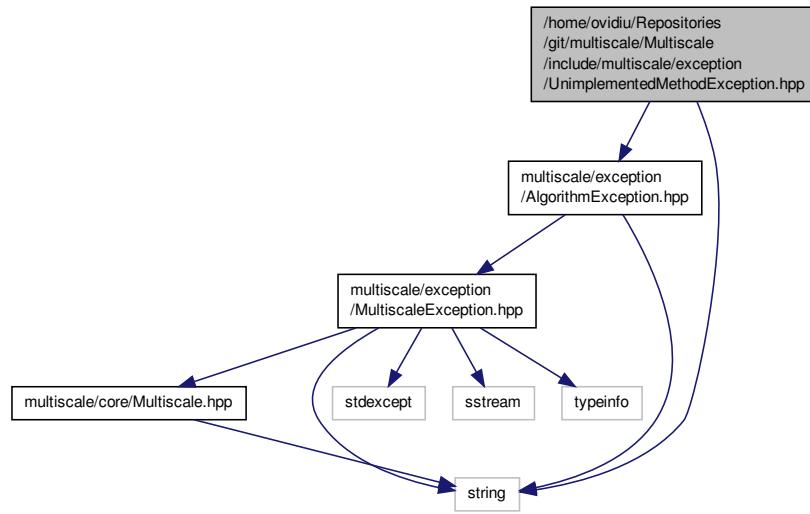
## Namespaces

- [multiscale](#)

## 8.16 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/UnimplementedMethodException.hpp File Reference

```
#include "multiscale/exception/AlgorithmException.hpp"
#include <string>
```

Include dependency graph for UnimplementedMethodException.hpp:



## Classes

- class [multiscale::UnimplementedMethodException](#)

*Class for representing unimplemented method exceptions.*

## Namespaces

- [multiscale](#)

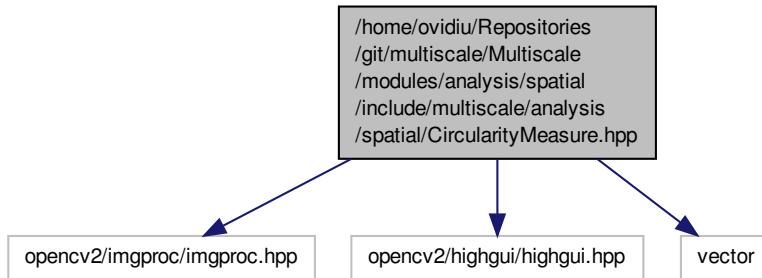
## Variables

- const std::string [multiscale::ERR\\_UNIMPLEMENTED\\_METHOD](#) = "The method you tried to call is not implemented. Please change."

## 8.17 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/CircularityMeasure.hpp File Reference

```
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <vector>
```

Include dependency graph for CircularityMeasure.hpp:



## Classes

- class [multiscale::analysis::CircularityMeasure](#)

*Class for computing the circularity measure for the given collection of points.*

## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

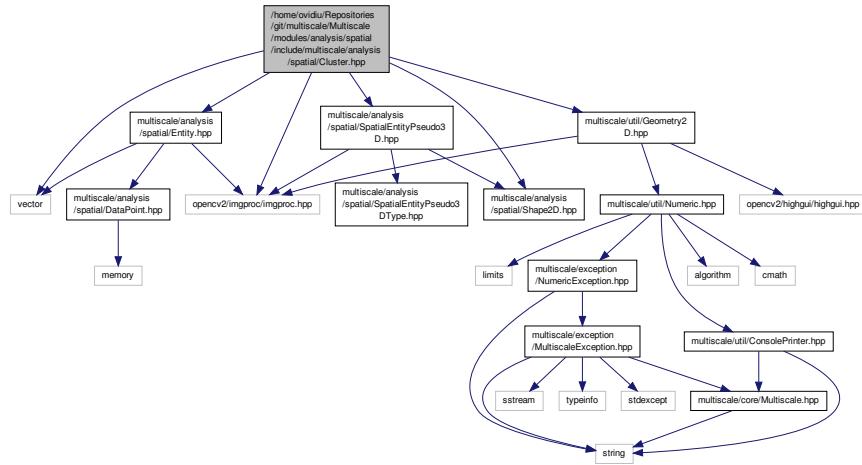
## 8.18 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Cluster.hpp File Reference

```

#include "multiscale/analysis/spatial/Entity.hpp"
#include "multiscale/analysis/spatial/Shape2D.hpp"
#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <vector>

```

Include dependency graph for Cluster.hpp:



## Classes

- class [multiscale::analysis::Cluster](#)

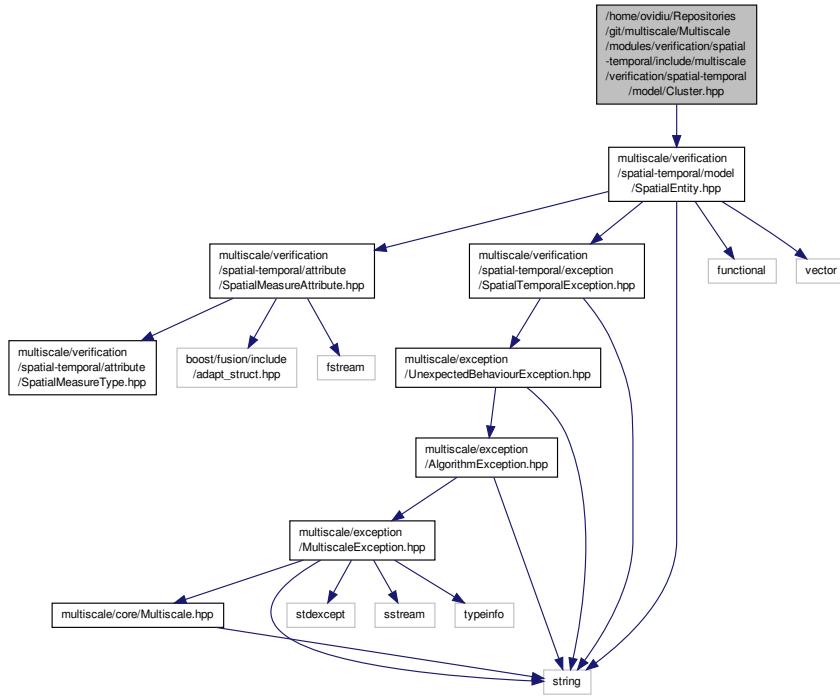
*Class for representing a cluster of entities in an image.*

## Namespaces

- multiscale
- multiscale::analysis

## 8.19 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Cluster.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SpatialEntity.hpp"
```



## Classes

- class multiscale::verification::Cluster

*Class for representing a cluster.*

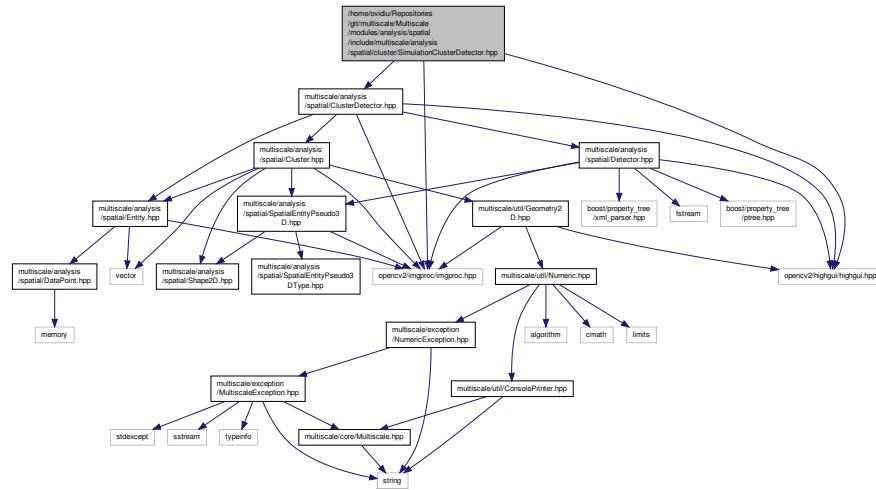
## Namespaces

- multiscale
- multiscale::verification

## 8.20 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp File Reference

```
#include "multiscale/analysis/spatial/ClusterDetector.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
```

Include dependency graph for SimulationClusterDetector.hpp:



## Classes

- class [multiscale::analysis::SimulationClusterDetector](#)

*Class for detecting clusters in 2D images obtained from simulations.*

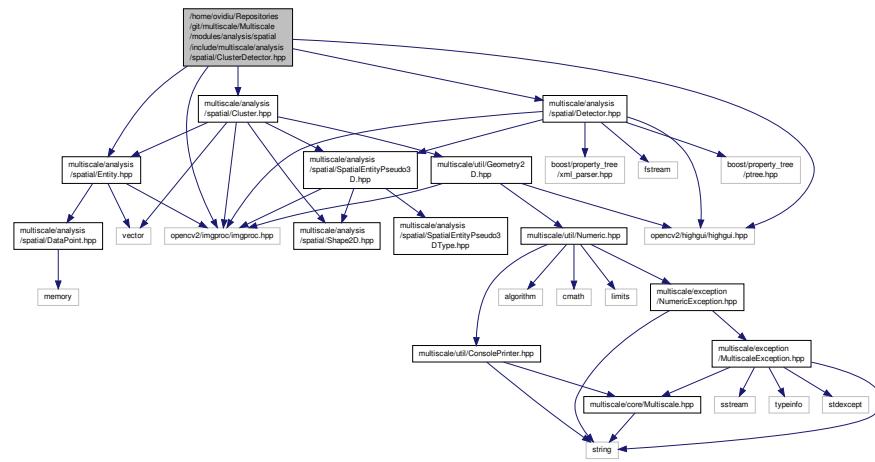
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.21 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp File Reference

```
#include "multiscale/analysis/spatial/Cluster.hpp"
#include "multiscale/analysis/spatial/Detector.hpp"
#include "multiscale/analysis/spatial/Entity.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
```

Include dependency graph for ClusterDetector.hpp:



## Classes

- class [multiscale::analysis::ClusterDetector](#)

*Class for detecting clusters in 2D images.*

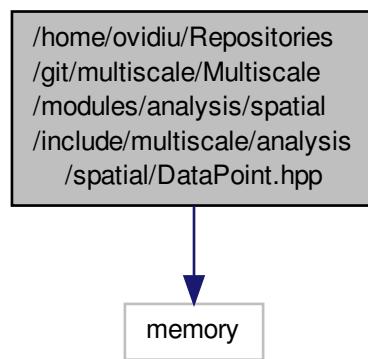
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.22 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/DataPoint.hpp File Reference

#include <memory>

Include dependency graph for DataPoint.hpp:



## Classes

- class [multiscale::analysis::DataPoint](#)

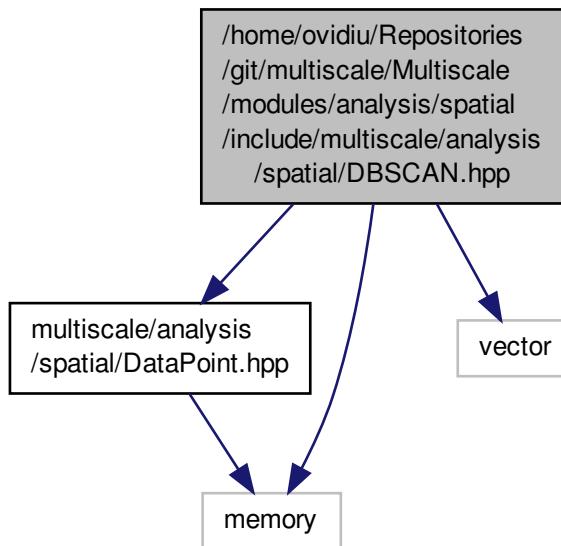
*Class for representing a data point.*

## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.23 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/DBSCAN.hpp File Reference

```
#include "multiscale/analysis/spatial/DataPoint.hpp"
#include <memory>
#include <vector>
Include dependency graph for DBSCAN.hpp:
```



## Classes

- class [multiscale::analysis::DBSCAN](#)

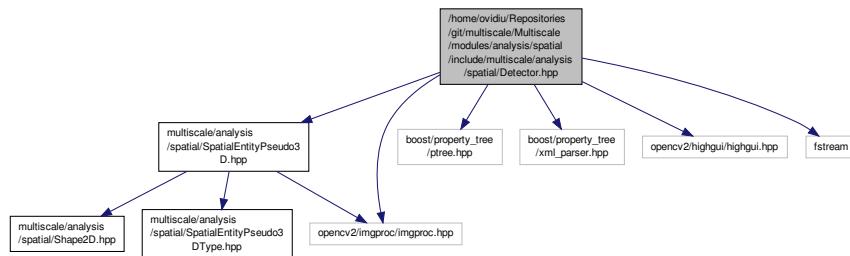
*Class which implements an improved version of the DBSCAN algorithm.*

## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

```
#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp"
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <fstream>
```

Include dependency graph for Detector.hpp:



## Classes

- class [multiscale::analysis::Detector](#)

*Abstract class for detecting entities of interest in images.*

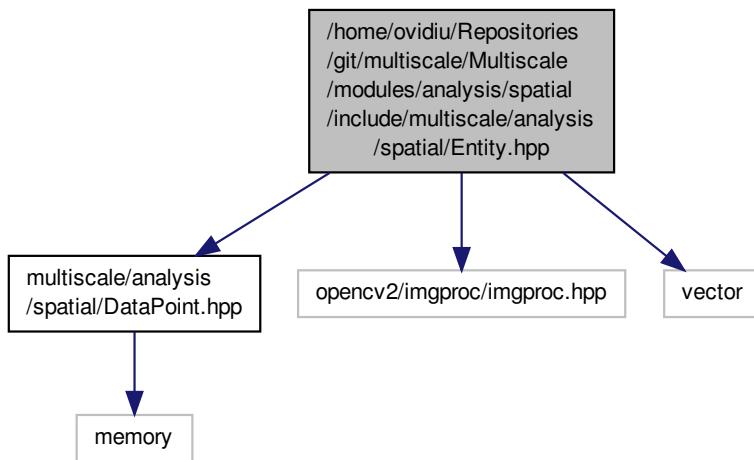
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.25 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Entity.hpp File Reference

```
#include "multiscale/analysis/spatial/DataPoint.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <vector>
```

Include dependency graph for Entity.hpp:



## Classes

- class [multiscale::analysis::Entity](#)

*Class for representing an entity in an image (e.g. cell, organism etc.)*

## Namespaces

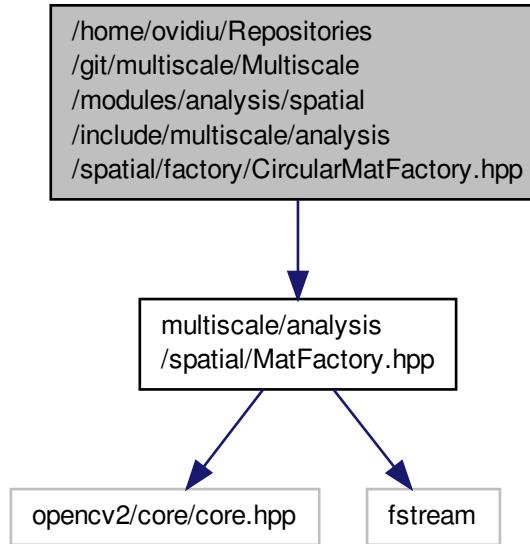
- [multiscale](#)
- [multiscale::analysis](#)

## 8.26 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/an...

### CircularMatFactory.hpp File Reference

```
#include "multiscale/analysis/spatial/MatFactory.hpp"
```

Include dependency graph for CircularMatFactory.hpp:



## Classes

- class [multiscale::analysis::CircularMatFactory](#)

*Class for creating a Mat object considering a circular grid.*

## Namespaces

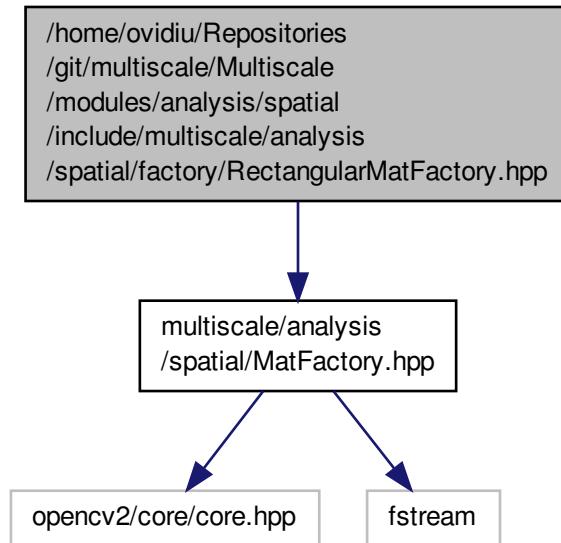
- [multiscale](#)
- [multiscale::analysis](#)

## 8.27 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factory/RectangularMatFactory.hpp File Reference

---

```
#include "multiscale/analysis/spatial/MatFactory.hpp"
```

Include dependency graph for RectangularMatFactory.hpp:



## Classes

- class [multiscale::analysis::RectangularMatFactory](#)

*Class for creating a cv::Mat object considering a rectangular grid.*

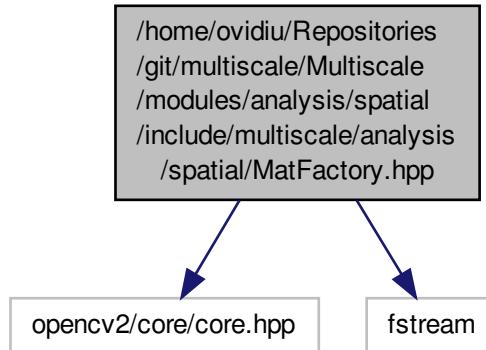
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.28 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/MatFactory.hpp File Reference

```
#include "opencv2/core/core.hpp"
#include <fstream>
```

Include dependency graph for MatFactory.hpp:



## Classes

- class [multiscale::analysis::MatFactory](#)

*Class for creating a cv::Mat object.*

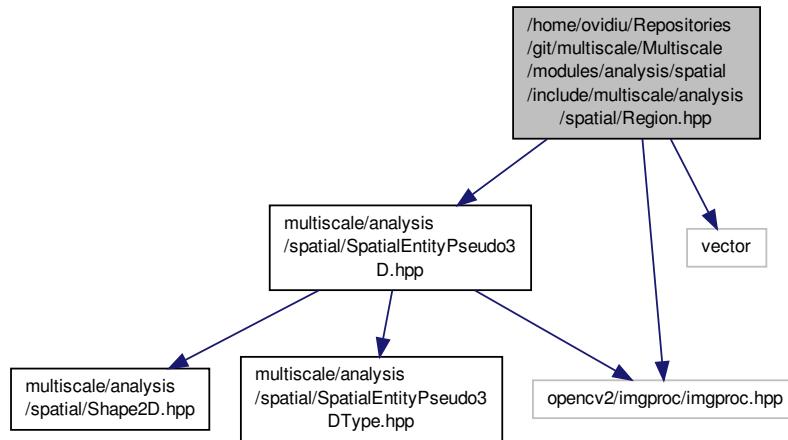
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.29 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Region.hpp File Reference

```
#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include <vector>
```

Include dependency graph for Region.hpp:



## Classes

- class [multiscale::analysis::Region](#)

*Class for representing a region.*

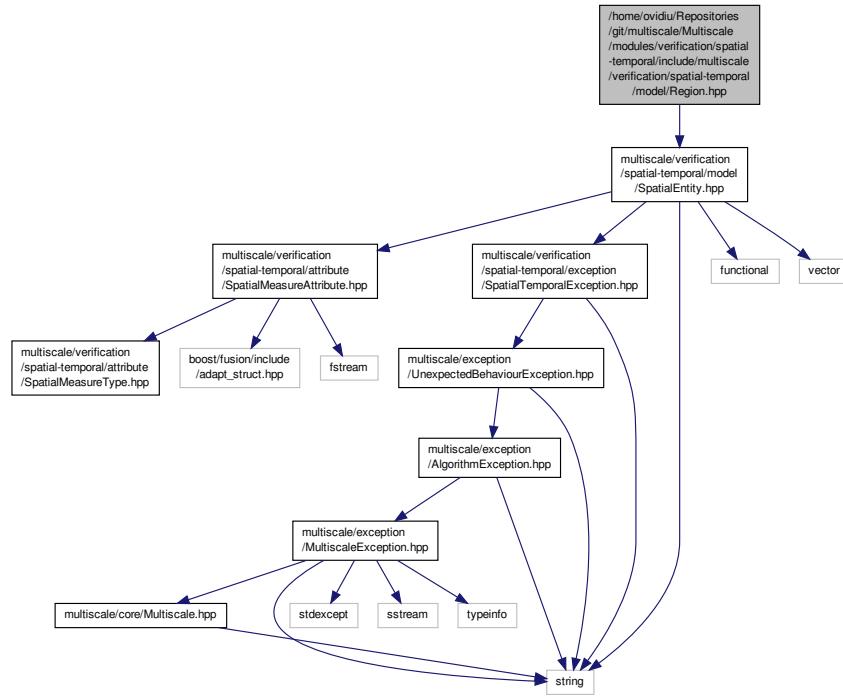
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.30 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Region.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SpatialEntity.hpp"
```

Include dependency graph for Region.hpp:



## Classes

- class [multiscale::verification::Region](#)

*Class for representing a region.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

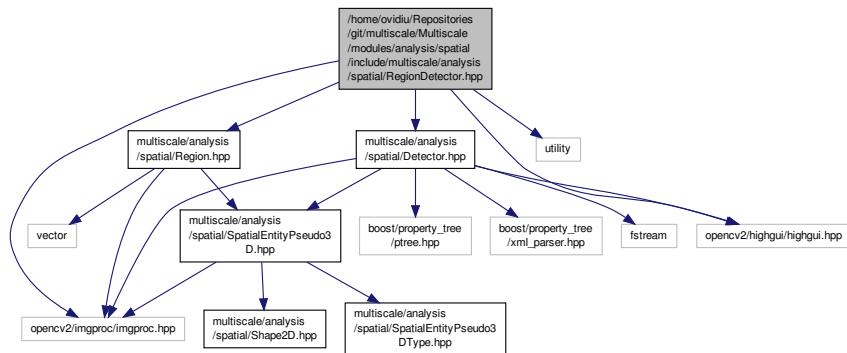
## 8.31 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/RegionDetector.hpp File Reference

```

#include "multiscale/analysis/spatial/Detector.hpp"
#include "multiscale/analysis/spatial/Region.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
#include <utility>

```

Include dependency graph for RegionDetector.hpp:



## Classes

- class [multiscale::analysis::RegionDetector](#)

*Class for detecting regions of high intensity in grayscale images.*

## Namespaces

- multiscale
- multiscale::analysis

## TypeDefs

- typedef std::pair< std::vector< cv::Point >, std::vector< std::vector< cv::Point > > > [multiscale::analysis::Polygon](#)

## 8.32 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/Shape2D.hpp File Reference

## Namespaces

- multiscale
- multiscale::analysis

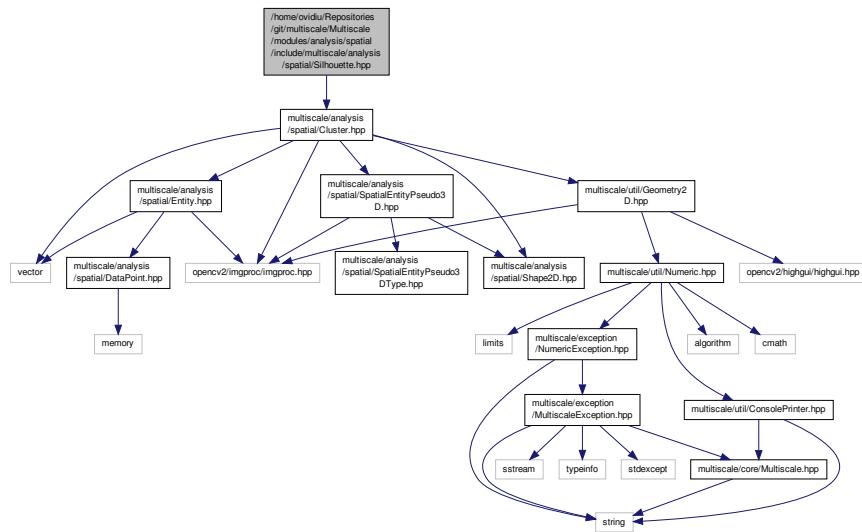
## Enumerations

- enum [multiscale::analysis::Shape2D](#) : unsigned int { multiscale::analysis::Shape2D::Triangle = 0, multiscale::analysis::Shape2D::Rectangle, multiscale::analysis::Shape2D::Circle, multiscale::analysis::Shape2D::Undefined }

*Enumeration for determining the type of a 2D shape.*

## Silhouette.hpp File Reference

```
#include "multiscale/analysis/spatial/Cluster.hpp"
Include dependency graph for Silhouette.hpp:
```



## Classes

- class [multiscale::analysis::Silhouette](#)

*Class for computing the "Silhouette" clustering index.*

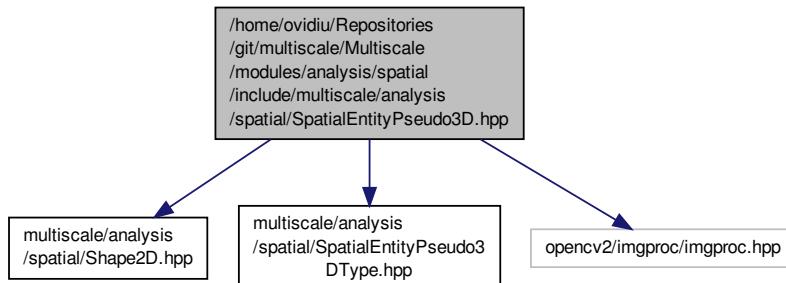
## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.34 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/an SpatialEntityPseudo3D.hpp File Reference

```
#include "multiscale/analysis/spatial/Shape2D.hpp"
#include "multiscale/analysis/spatial/SpatialEntityPseudo3DType.hpp"
#include "opencv2/imgproc/imgproc.hpp"
```

Include dependency graph for SpatialEntityPseudo3D.hpp:



## Classes

- class [multiscale::analysis::SpatialEntityPseudo3D](#)  
*Class for representing a pseudo-3D (explicit 2D + implicit height) object.*

## Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

## 8.35 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/SpatialEntityPseudo3DType.hpp File Reference

### Namespaces

- [multiscale](#)
- [multiscale::analysis](#)

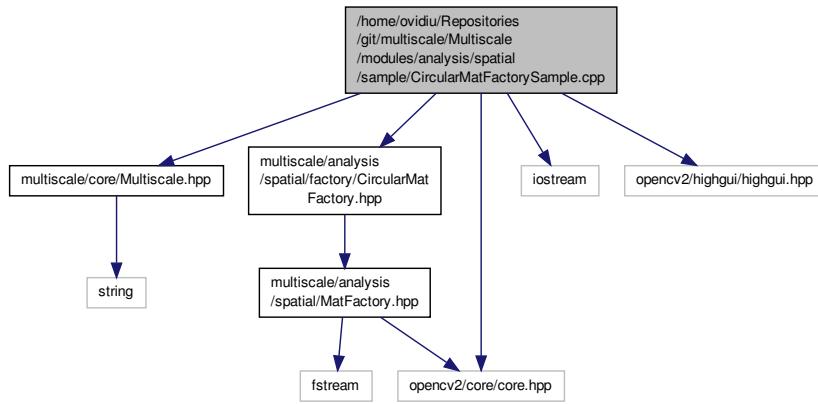
### Enumerations

- enum [multiscale::analysis::SpatialEntityPseudo3DType](#) : unsigned int { [multiscale::analysis::SpatialEntityPseudo3DType::Cluster](#) = 0, [multiscale::analysis::SpatialEntityPseudo3DType::Region](#) }  
*Enumeration for determining the type of a pseudo 3D entity.*

## 8.36 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/CircularMatFactorySample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/analysis/spatial/factory/CircularMatFactory.hpp"
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
```

Include dependency graph for CircularMatFactorySample.cpp:



## Functions

- int main ()

### 8.36.1 Function Documentation

#### 8.36.1.1 int main ( )

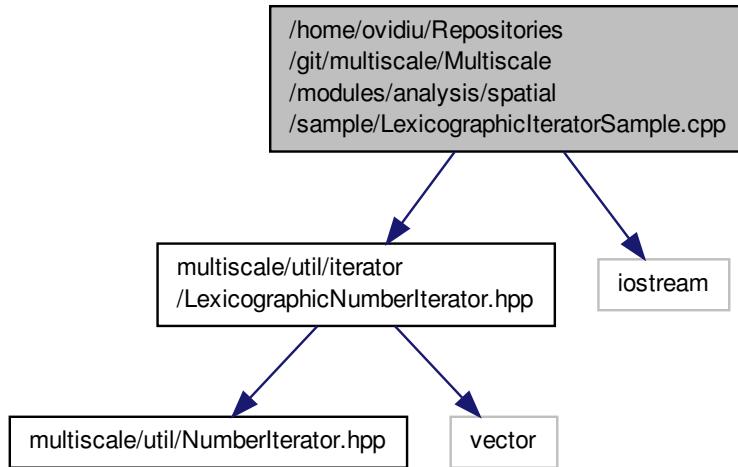
Definition at line 13 of file CircularMatFactorySample.cpp.

References multiscale::analysis::CircularMatFactory::createFromViewerImage(), multiscale::EXEC\_SUCCESS\_CODE, and multiscale::analysis::CircularMatFactory::maxColourBarIntensityFromViewerImage().

## 8.37 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/LexicographicIteratorSample.cpp File Reference

```
#include "multiscale/util/iterator/LexicographicNumberIterator.hpp"
#include <iostream>
```

Include dependency graph for LexicographicIteratorSample.cpp:



## Functions

- int `main ()`

### 8.37.1 Function Documentation

#### 8.37.1.1 int main ( )

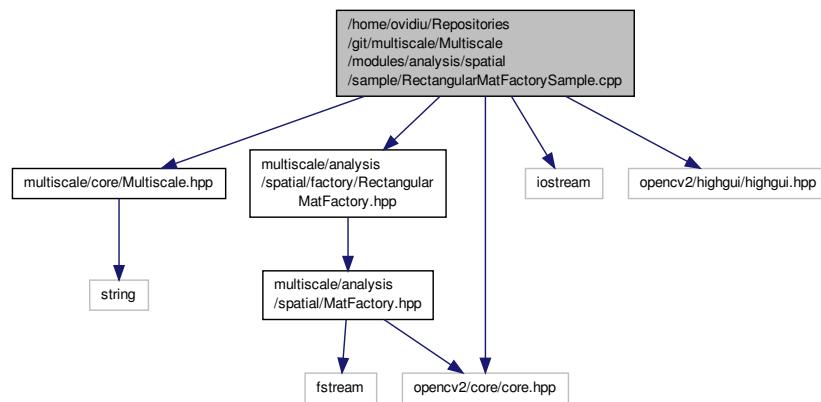
Definition at line 9 of file LexicographicIteratorSample.cpp.

References multiscale::NumberIterator::hasNext(), and multiscale::LexicographicNumberIterator::number().

## 8.38 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/RectangularMatFactorySample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp"
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
```

Include dependency graph for RectangularMatFactorySample.cpp:



## Functions

- int main ()

### 8.38.1 Function Documentation

### 8.38.1.1 int main( )

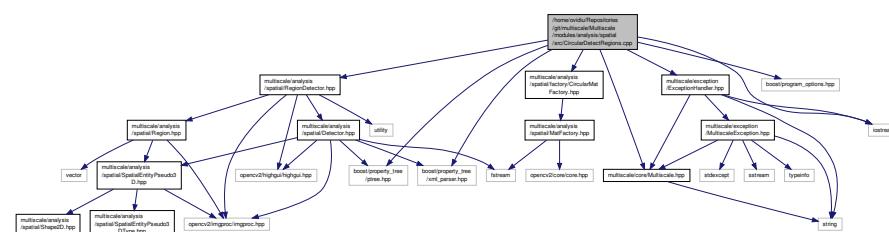
Definition at line 13 of file RectangularMatFactorySample.cpp.

References multiscale::analysis::RectangularMatFactory::createFromViewerImage(), multiscale::EXEC\_SUCCESS\_CODE, and multiscale::analysis::RectangularMatFactory::maxColourBarIntensityFromViewerImage().

## 8.39 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← CircularDetectRegions.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/analysis/spatial/RegionDetector.hpp"
#include "multiscale/analysis/spatial/factory/CircularMatFactory.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for CircularDetectRegions.cpp



## Functions

- `po::variables_map initArgumentsConfig (po::options_description &usageDescription, int argc, char **argv)`
- `void printHelpInformation (const po::variables_map &vm, const po::options_description &usageDescription)`
- `void printWrongParameters ()`
- `bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, bool &debugFlag, int argc, char **argv)`
- `void loadDetectorParameterValues (RegionDetector &detector)`
- `void saveDetectorParameterValues (RegionDetector &detector)`
- `void loadDetectorParameterValues (RegionDetector &detector, bool debugMode)`
- `void saveDetectorParameterValues (RegionDetector &detector, bool debugMode)`
- `int main (int argc, char **argv)`

## Variables

- `const std::string CONFIG_FILE = "/usr/local/share/mule/config/analysis/spatial/circular_region_detector.xml"`
- `const std::string LABEL_ROOT_COMMENT = "<xmlcomment>"`
- `const std::string LABEL_ALPHA = "detector.alpha"`
- `const std::string LABEL_BETA = "detector.beta"`
- `const std::string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"`
- `const std::string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS = "detector.morphologicalCloseIterations"`
- `const std::string LABEL_EPSILON = "detector.epsilon"`
- `const std::string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"`
- `const std::string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"`
- `const std::string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."`

### 8.39.1 Function Documentation

**8.39.1.1 `bool areValidParameters ( std::string & inputfilepath, std::string & outputfilename, bool & debugFlag, int argc, char ** argv )`**

Definition at line 71 of file CircularDetectRegions.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

Referenced by `main()`.

**8.39.1.2 `po::variables_map initArgumentsConfig ( po::options_description & usageDescription, int argc, char ** argv )`**

Definition at line 47 of file CircularDetectRegions.cpp.

Referenced by `areValidParameters()`.

**8.39.1.3 `void loadDetectorParameterValues ( RegionDetector & detector )`**

Definition at line 99 of file CircularDetectRegions.cpp.

References `CONFIG_FILE`, `LABEL_ALPHA`, `LABEL_BETA`, `LABEL_BLUR_KERNEL_SIZE`, `LABEL_EPSILON`, `LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS`, `LABEL_REGION_AREA_THRESH`, `LABEL_THRESHOLD_OLD_VALUE`, `multiscale::analysis::RegionDetector::setAlpha()`, `multiscale::analysis::RegionDetector::setBeta()`, `multiscale::analysis::RegionDetector::setBlurKernelSize()`, `multiscale::analysis::RegionDetector::setEpsilon()`, `multiscale::analysis::RegionDetector::setMorphologicalCloseIterations()`, `multiscale::analysis::RegionDetector::setRegionAreaThresh()`, and `multiscale::analysis::RegionDetector::setThresholdValue()`.

Referenced by `loadDetectorParameterValues()`, and `main()`.

---

8.39.1.4 void loadDetectorParameterValues ( **RegionDetector & detector, bool debugMode** )

Definition at line 133 of file CircularDetectRegions.cpp.

References loadDetectorParameterValues().

8.39.1.5 int main ( **int argc, char \*\* argv** )

Definition at line 145 of file CircularDetectRegions.cpp.

References areValidParameters(), multiscale::analysis::CircularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, loadDetectorParameterValues(), multiscale::analysis::Detector::outputResults(), multiscale::ExceptionHandler::printDetailedErrorMessage(), printWrongParameters(), and saveDetectorParameterValues().

8.39.1.6 void printHelpInformation ( **const po::variables\_map & vm, const po::options\_description & usageDescription** )

Definition at line 60 of file CircularDetectRegions.cpp.

Referenced by areValidParameters().

8.39.1.7 void printWrongParameters ( )

Definition at line 65 of file CircularDetectRegions.cpp.

References multiscale::ERR\_MSG.

Referenced by main().

8.39.1.8 void saveDetectorParameterValues ( **RegionDetector & detector** )

Definition at line 114 of file CircularDetectRegions.cpp.

References CONFIG\_FILE, multiscale::analysis::RegionDetector::getAlpha(), multiscale::analysis::RegionDetector::getBeta(), multiscale::analysis::RegionDetector::getBlurKernelSize(), multiscale::analysis::RegionDetector::getEpsilon(), multiscale::analysis::RegionDetector::getMorphologicalCloseIterations(), multiscale::analysis::RegionDetector::getRegionAreaThresh(), multiscale::analysis::RegionDetector::getThresholdValue(), LABEL\_ALPHA, LABEL\_BETA, LABEL\_BLUR\_KERNEL\_SIZE, LABEL\_EPSILON, LABEL\_MORPHOLOGICAL\_CLOSE\_ITERATIONS, LABEL\_REGION\_AREA\_THRESH, LABEL\_ROOT\_COMMENT, LABEL\_THRESHOLD\_VALUE, and ROOT\_COMMENT.

Referenced by main(), and saveDetectorParameterValues().

8.39.1.9 void saveDetectorParameterValues ( **RegionDetector & detector, bool debugMode** )

Definition at line 138 of file CircularDetectRegions.cpp.

References saveDetectorParameterValues().

## 8.39.2 Variable Documentation

8.39.2.1 const std::string CONFIG\_FILE = "/usr/local/share/mule/config/analysis/spatial/circular\_region\_detector.xml"

Definition at line 32 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.2 const std::string LABEL\_ALPHA = "detector.alpha"

Definition at line 35 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.3 const std::string LABEL\_BETA = "detector.beta"

Definition at line 36 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.4 const std::string LABEL\_BLUR\_KERNEL\_SIZE = "detector.blurKernelSize"

Definition at line 37 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.5 const std::string LABEL\_EPSILON = "detector.epsilon"

Definition at line 39 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.6 const std::string LABEL\_MORPHOLOGICAL\_CLOSE\_ITERATIONS = "detector.morphologicalCloselterations"

Definition at line 38 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.7 const std::string LABEL\_REGION\_AREA\_THRESH = "detector.regionAreaThresh"

Definition at line 40 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.8 const std::string LABEL\_ROOT\_COMMENT = "<xmlcomment>"

Definition at line 34 of file CircularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

8.39.2.9 const std::string LABEL\_THRESHOLD\_VALUE = "detector.thresholdValue"

Definition at line 41 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.39.2.10 const std::string ROOT\_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

Definition at line 43 of file CircularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

## 8.40

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/CircularityMeasure.cpp

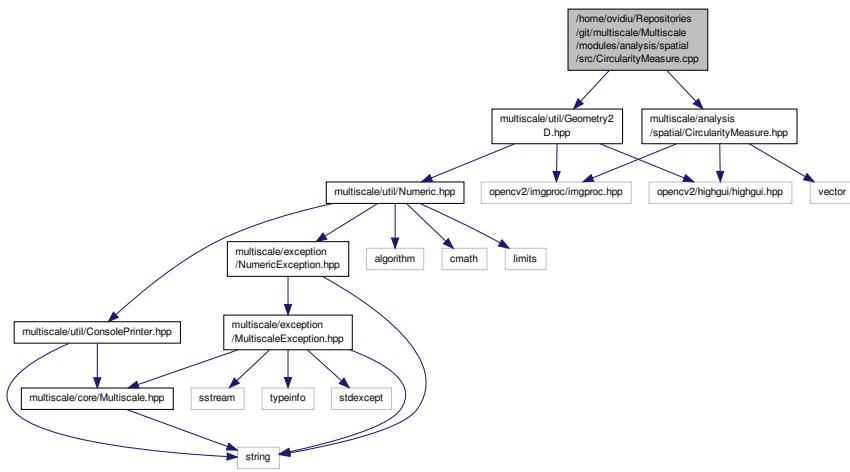
File Reference

1095

8.40 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/

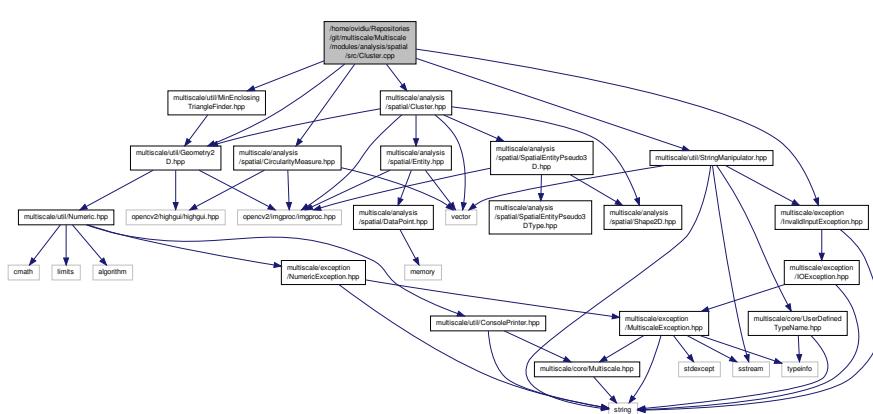
### CircularityMeasure.cpp File Reference

```
#include "multiscale/analysis/spatial/CircularityMeasure.hpp"
#include "multiscale/util/Geometry2D.hpp"
Include dependency graph for CircularityMeasure.cpp:
```



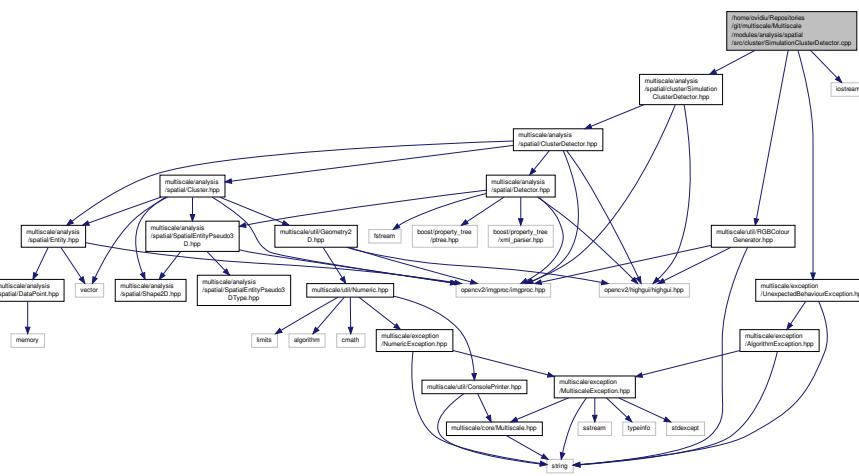
## 8.41 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Cluster.cpp File Reference

```
#include "multiscale/analysis/spatial/CircularityMeasure.hpp"
#include "multiscale/analysis/spatial/Cluster.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include "multiscale/util/MinEnclosingTriangleFinder.hpp"
#include "multiscale/util/StringManipulator.hpp"
Include dependency graph for Cluster.cpp:
```



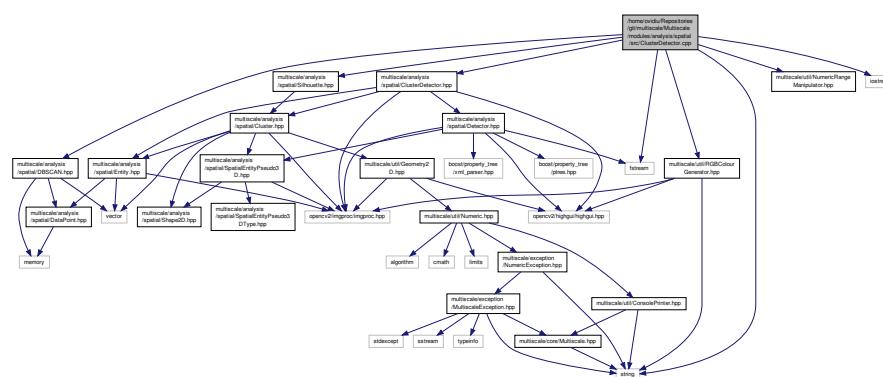
## 8.42 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/[SimulationClusterDetector.cpp](#) File Reference

```
#include "multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/RGBColourGenerator.hpp"
#include <iostream>
Include dependency graph for SimulationClusterDetector.cpp:
```



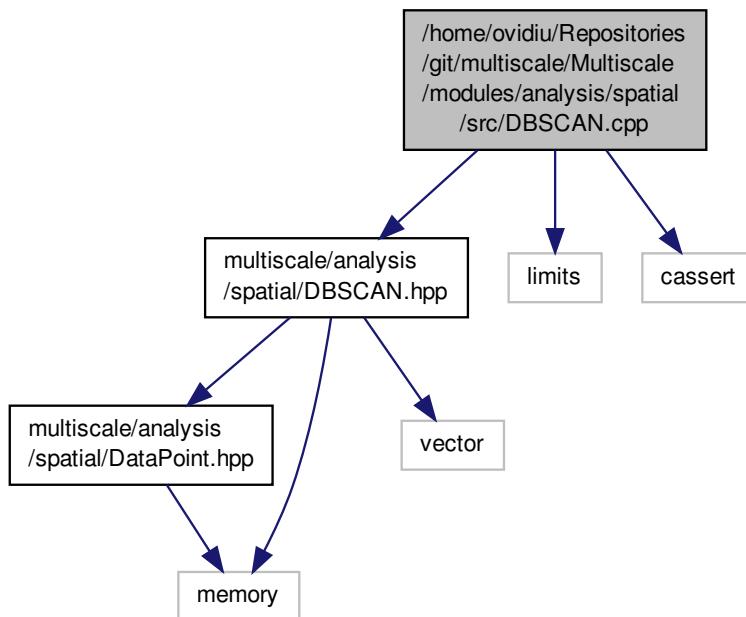
## 8.43 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← ClusterDetector.cpp File Reference

```
#include "multiscale/analysis/spatial/ClusterDetector.hpp"
#include "multiscale/analysis/spatial/DBSCAN.hpp"
#include "multiscale/analysis/spatial/Silhouette.hpp"
#include "multiscale/util/NumericRangeManipulator.hpp"
#include "multiscale/util/RGBColourGenerator.hpp"
#include <iostream>
#include <fstream>
#include <string>
```



## 8.44 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/DBSCAN.cpp File Reference

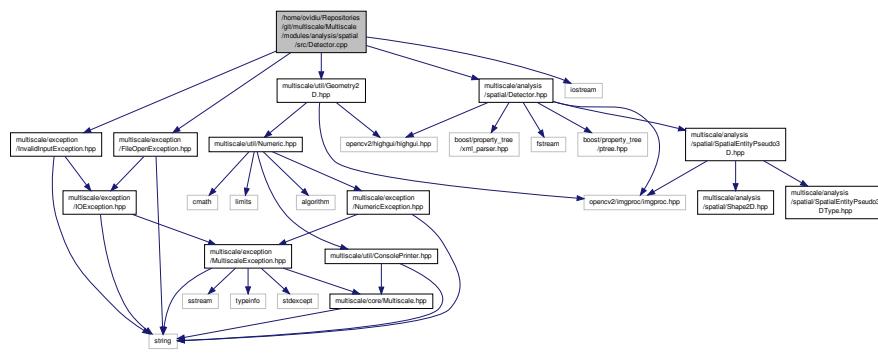
```
#include "multiscale/analysis/spatial/DBSCAN.hpp"
#include <limits>
#include <cassert>
Include dependency graph for DBSCAN.cpp:
```



## 8.45 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Detector.cpp File Reference

```
#include "multiscale/analysis/spatial/Detector.hpp"
#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include <iostream>
```

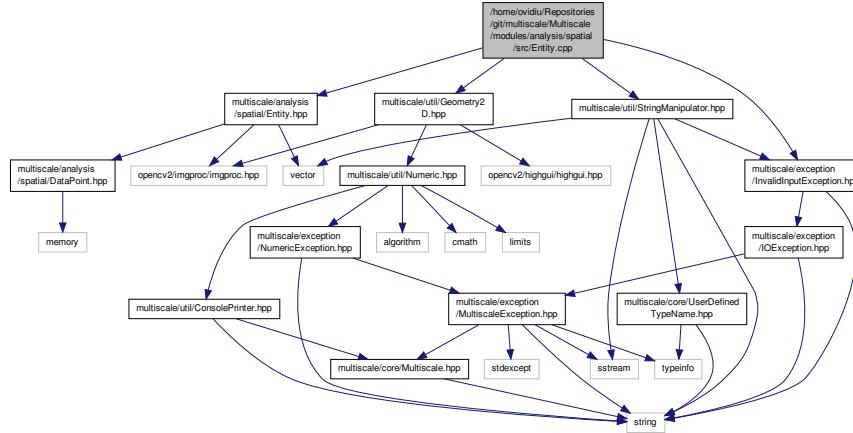
Include dependency graph for Detector.cpp:



## 8.46 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Entity.cpp File Reference

```
#include "multiscale/analysis/spatial/Entity.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/Geometry2D.hpp"
```

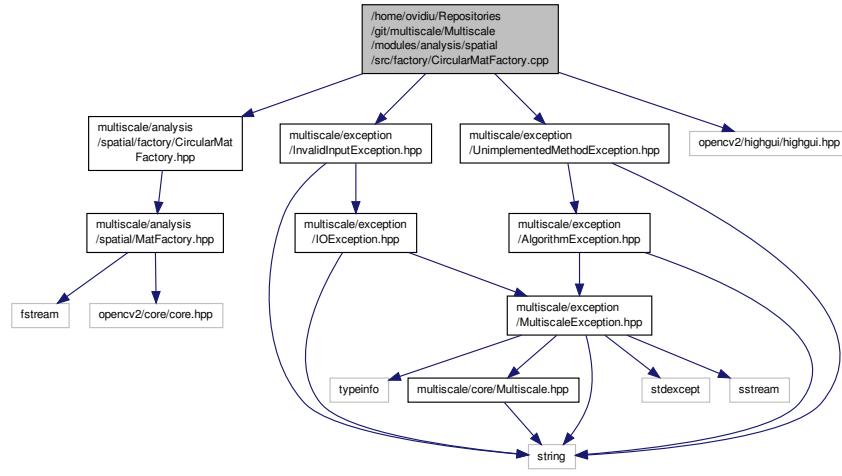
Include dependency graph for Entity.cpp:



## 8.47 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/CircularMatFactory.cpp File Reference

```
#include "multiscale/analysis/spatial/factory/CircularMatFactory.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/UnimplementedMethodException.hpp"
#include "opencv2/highgui/highgui.hpp"
```

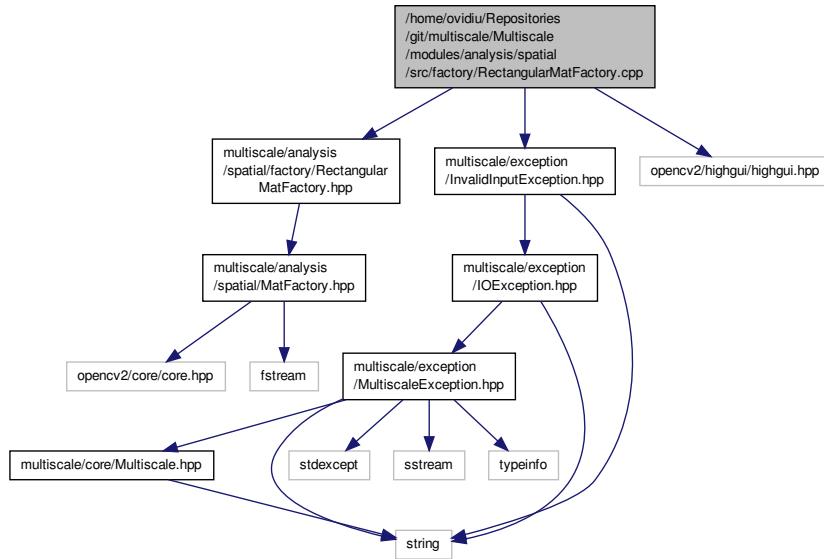
Include dependency graph for CircularMatFactory.cpp:



## 8.48 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/← RectangularMatFactory.cpp File Reference

```
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "opencv2/highgui/highgui.hpp"
```

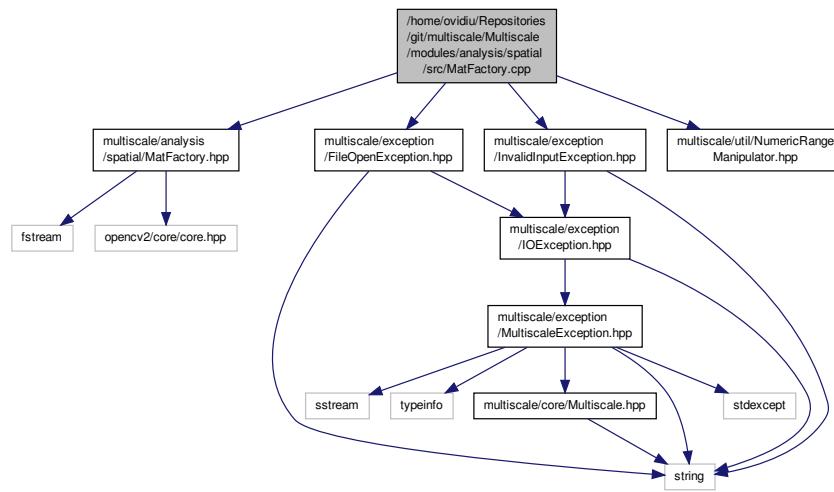
Include dependency graph for RectangularMatFactory.cpp:



## 8.49 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← MatFactory.cpp File Reference

```
#include "multiscale/analysis/spatial/MatFactory.hpp"
#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/NumericRangeManipulator.hpp"
Include dependency graph for MatFactory.cpp:
```

Include dependency graph for MatFactory.cpp:

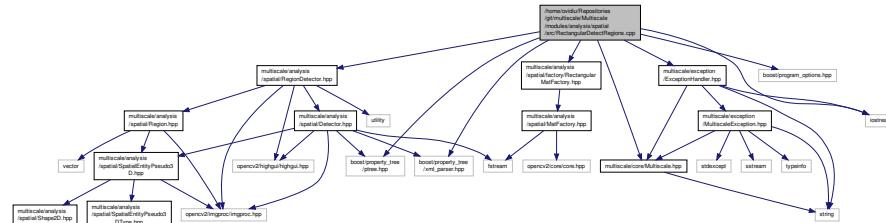


## 8.50 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← RectangularDetectRegions.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/analysis/spatial/RegionDetector.hpp"
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for RectangularDetectRegions.cpp:

Include dependency graph for RectangularDetectRegions.cpp:



- `po::variables_map initArgumentsConfig (po::options_description &usageDescription, int argc, char **argv)`
- `void printHelpInformation (const po::variables_map &vm, const po::options_description &usageDescription)`
- `void printWrongParameters ()`
- `bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, bool &debugFlag, int argc, char **argv)`
- `void loadDetectorParameterValues (RegionDetector &detector)`
- `void saveDetectorParameterValues (RegionDetector &detector)`
- `void loadDetectorParameterValues (RegionDetector &detector, bool debugMode)`
- `void saveDetectorParameterValues (RegionDetector &detector, bool debugMode)`
- `int main (int argc, char **argv)`

## Variables

- `const std::string CONFIG_FILE = "/usr/local/share/mule/config/analysis/spatial/rectangular_region_detector.xml"`
- `const std::string LABEL_ROOT_COMMENT = "<xmlcomment>"`
- `const std::string LABEL_ALPHA = "detector.alpha"`
- `const std::string LABEL_BETA = "detector.beta"`
- `const std::string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"`
- `const std::string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS = "detector.morphologicalCloseIterations"`
- `const std::string LABEL_EPSILON = "detector.epsilon"`
- `const std::string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"`
- `const std::string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"`
- `const std::string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."`

### 8.50.1 Function Documentation

**8.50.1.1 `bool areValidParameters ( std::string & inputfilepath, std::string & outputfilename, bool & debugFlag, int argc, char ** argv )`**

Definition at line 71 of file RectangularDetectRegions.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

Referenced by `main()`.

**8.50.1.2 `po::variables_map initArgumentsConfig ( po::options_description & usageDescription, int argc, char ** argv )`**

Definition at line 47 of file RectangularDetectRegions.cpp.

Referenced by `areValidParameters()`.

**8.50.1.3 `void loadDetectorParameterValues ( RegionDetector & detector )`**

Definition at line 99 of file RectangularDetectRegions.cpp.

References `CONFIG_FILE`, `LABEL_ALPHA`, `LABEL_BETA`, `LABEL_BLUR_KERNEL_SIZE`, `LABEL_EPSILON`, `LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS`, `LABEL_REGION_AREA_THRESH`, `LABEL_THRESHOLD_OLD_VALUE`, `multiscale::analysis::RegionDetector::setAlpha()`, `multiscale::analysis::RegionDetector::setBeta()`, `multiscale::analysis::RegionDetector::setBlurKernelSize()`, `multiscale::analysis::RegionDetector::setEpsilon()`, `multiscale::analysis::RegionDetector::setMorphologicalCloseIterations()`, `multiscale::analysis::RegionDetector::setRegionAreaThresh()`, and `multiscale::analysis::RegionDetector::setThresholdValue()`.

Referenced by `loadDetectorParameterValues()`, and `main()`.

#### 8.50.1.4 void loadDetectorParameterValues ( **RegionDetector & detector**, **bool debugMode** )

Definition at line 133 of file RectangularDetectRegions.cpp.

References loadDetectorParameterValues().

#### 8.50.1.5 int main ( **int argc**, **char \*\* argv** )

Definition at line 145 of file RectangularDetectRegions.cpp.

References areValidParameters(), multiscale::analysis::RectangularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, loadDetectorParameterValues(), multiscale::analysis::Detector::outputResults(), multiscale::ExceptionHandler::printDetailedErrorMessage(), printWrongParameters(), and saveDetectorParameterValues().

#### 8.50.1.6 void printHelpInformation ( **const po::variables\_map & vm**, **const po::options\_description & usageDescription** )

Definition at line 60 of file RectangularDetectRegions.cpp.

Referenced by areValidParameters().

#### 8.50.1.7 void printWrongParameters ( )

Definition at line 65 of file RectangularDetectRegions.cpp.

References multiscale::ERR\_MSG.

Referenced by main().

#### 8.50.1.8 void saveDetectorParameterValues ( **RegionDetector & detector** )

Definition at line 114 of file RectangularDetectRegions.cpp.

References CONFIG\_FILE, multiscale::analysis::RegionDetector::getAlpha(), multiscale::analysis::RegionDetector::getBeta(), multiscale::analysis::RegionDetector::getBlurKernelSize(), multiscale::analysis::RegionDetector::getEpsilon(), multiscale::analysis::RegionDetector::getMorphologicalCloseIterations(), multiscale::analysis::RegionDetector::getRegionAreaThresh(), multiscale::analysis::RegionDetector::getThresholdValue(), LABEL\_ALPHA, LABEL\_BETA, LABEL\_BLUR\_KERNEL\_SIZE, LABEL\_EPSILON, LABEL\_MORPHOLOGICAL\_CLOSE\_ITERATIONS, LABEL\_REGION\_AREA\_THRESH, LABEL\_ROOT\_COMMENT, LABEL\_THRESHOLD\_VALUE, and ROOT\_COMMENT.

Referenced by main(), and saveDetectorParameterValues().

#### 8.50.1.9 void saveDetectorParameterValues ( **RegionDetector & detector**, **bool debugMode** )

Definition at line 138 of file RectangularDetectRegions.cpp.

References saveDetectorParameterValues().

### 8.50.2 Variable Documentation

#### 8.50.2.1 const std::string CONFIG\_FILE = "/usr/local/share/mule/config/analysis/spatial/rectangular\_region\_detector.xml"

Definition at line 32 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

---

```
8.50.2.2 const std::string LABEL_ALPHA = "detector.alpha"
```

Definition at line 35 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.3 const std::string LABEL_BETA = "detector.beta"
```

Definition at line 36 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.4 const std::string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"
```

Definition at line 37 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.5 const std::string LABEL_EPSILON = "detector.epsilon"
```

Definition at line 39 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.6 const std::string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS = "detector.morphologicalCloselterations"
```

Definition at line 38 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.7 const std::string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"
```

Definition at line 40 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.8 const std::string LABEL_ROOT_COMMENT = "<xmlcomment>"
```

Definition at line 34 of file RectangularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

```
8.50.2.9 const std::string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"
```

Definition at line 41 of file RectangularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

```
8.50.2.10 const std::string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."
```

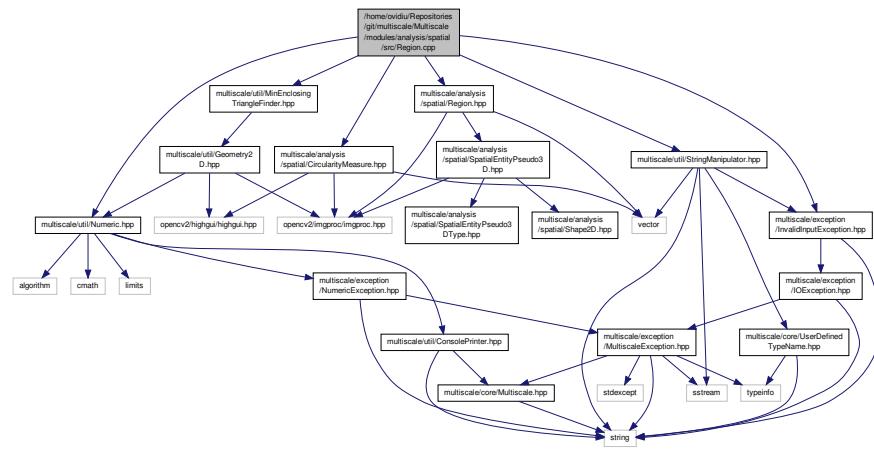
Definition at line 43 of file RectangularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

## 8.51 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Region.cpp File Reference

```
#include "multiscale/analysis/spatial/CircularityMeasure.hpp"
#include "multiscale/analysis/spatial/Region.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/MinEnclosingTriangleFinder.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
Include dependency graph for Region.cpp:
```

Include dependency graph for Region.cpp:

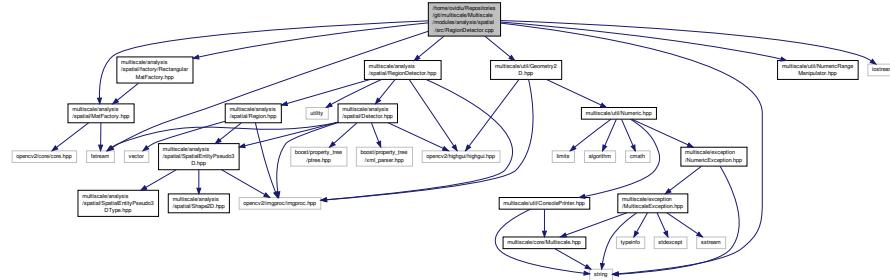


## 8.52 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← RegionDetector.cpp File Reference

```
#include "multiscale/analysis/spatial/MatFactory.hpp"
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp"
#include "multiscale/analysis/spatial/RegionDetector.hpp"
#include "multiscale/util/NumericRangeManipulator.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include <iostream>
#include <fstream>
#include <string>
```

Include dependency graph for RegionDetector.cpp:

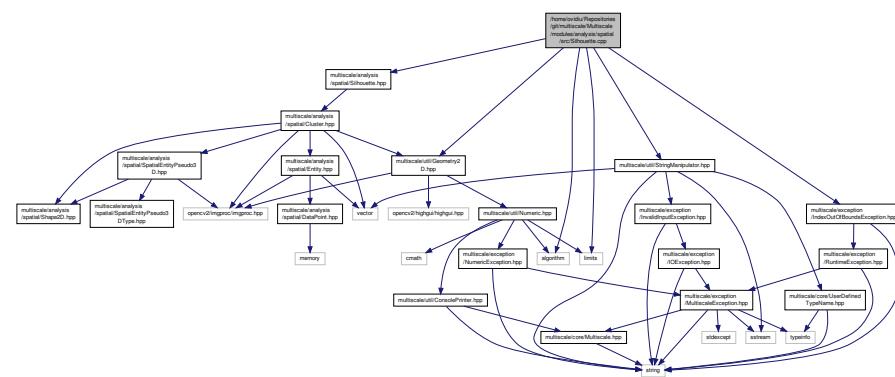
Include dependency graph for RegionDetector.cpp:



## 8.53 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← Silhouette.cpp File Reference

```
#include "multiscale/analysis/spatial/Silhouette.hpp"
#include "multiscale/exception/IndexOutOfBoundsException.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <algorithm>
#include <limits>
```

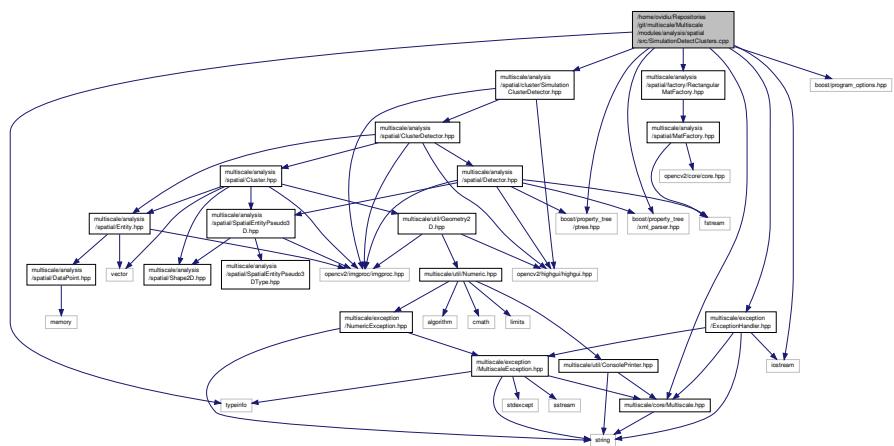
Include dependency graph for Silhouette.cpp:



## 8.54 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← SimulationDetectClusters.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/analysis/spatial/cluster/SimulationClusterDetector.hpp"
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include <boost/program_options.hpp>
#include <iostream>
#include <typeinfo>
```

Include dependency graph for SimulationDetectClusters.cpp:



## Functions

- po::variables\_map initArgumentsConfig (po::options\_description &usageDescription, int argc, char \*\*argv)
  - void printHelpInformation (const po::variables\_map &vm, const po::options\_description &usageDescription)
  - void printWrongParameters ()
  - bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, bool &debugFlag, unsigned int &height, unsigned int &width, unsigned int &maxPileup, int argc, char \*\*argv)
  - void loadDetectorParameterValues (SimulationClusterDetector &detector)
  - void saveDetectorParameterValues (SimulationClusterDetector &detector)
  - void loadDetectorParameterValues (SimulationClusterDetector &detector, bool debugMode)
  - void saveDetectorParameterValues (SimulationClusterDetector &detector, bool debugMode)
  - int main (int argc, char \*\*argv)

## Variables

- const std::string **CONFIG\_FILE** = "/usr/local/share/mule/config/analysis/spatial/simulation\_cluster\_detector.xml"
  - const std::string **LABEL\_ROOT\_COMMENT** = "<xmlcomment>"
  - const std::string **LABEL\_EPS** = "detector.eps"
  - const std::string **LABEL\_MINPOINTS** = "detector.minPoints"
  - const std::string **ROOT\_COMMENT** = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

## 8.54.1 Function Documentation

8.54.1.1 bool areValidParameters ( std::string & *inputfilepath*, std::string & *outputfilename*, bool & *debugFlag*, unsigned int & *height*, unsigned int & *width*, unsigned int & *maxPileup*, int *argc*, char \*\* *argv* )

Definition at line 70 of file SimulationDetectClusters.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

Referenced by main().

**8.54 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/SimulationDetectClusters.cpp File**

**Reference**

**1107**

**8.54.1.2 po::variables\_map initArgumentsConfig ( po::options\_description & usageDescription, int argc, char \*\* argv )**

Definition at line 43 of file SimulationDetectClusters.cpp.

Referenced by areValidParameters().

**8.54.1.3 void loadDetectorParameterValues ( SimulationClusterDetector & detector )**

Definition at line 104 of file SimulationDetectClusters.cpp.

References CONFIG\_FILE, LABEL\_EPS, LABEL\_MINPOINTS, multiscale::analysis::ClusterDetector::setEps(), and multiscale::analysis::ClusterDetector::setMinPoints().

Referenced by loadDetectorParameterValues(), and main().

**8.54.1.4 void loadDetectorParameterValues ( SimulationClusterDetector & detector, bool debugMode )**

Definition at line 128 of file SimulationDetectClusters.cpp.

References loadDetectorParameterValues().

**8.54.1.5 int main ( int argc, char \*\* argv )**

Definition at line 140 of file SimulationDetectClusters.cpp.

References areValidParameters(), multiscale::analysis::RectangularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, loadDetectorParameterValues(), multiscale::analysis::RectangularMatFactory::maxColourBarIntensityFromViewerImage(), multiscale::analysis::Detector::outputResults(), multiscale::ExceptionHandler::printDetailedErrorMessage(), printWrongParameters(), and saveDetectorParameterValues().

**8.54.1.6 void printHelpInformation ( const po::variables\_map & vm, const po::options\_description & usageDescription )**

Definition at line 59 of file SimulationDetectClusters.cpp.

Referenced by areValidParameters().

**8.54.1.7 void printWrongParameters ( )**

Definition at line 64 of file SimulationDetectClusters.cpp.

References multiscale::ERR\_MSG.

Referenced by main().

**8.54.1.8 void saveDetectorParameterValues ( SimulationClusterDetector & detector )**

Definition at line 114 of file SimulationDetectClusters.cpp.

References CONFIG\_FILE, multiscale::analysis::ClusterDetector::getEps(), multiscale::analysis::ClusterDetector::getMinPoints(), LABEL\_EPS, LABEL\_MINPOINTS, LABEL\_ROOT\_COMMENT, and ROOT\_COMMENT.

Referenced by main(), and saveDetectorParameterValues().

**8.54.1.9 void saveDetectorParameterValues ( SimulationClusterDetector & detector, bool debugMode )**

Definition at line 133 of file SimulationDetectClusters.cpp.

References saveDetectorParameterValues().

## 8.54.2 Variable Documentation

8.54.2.1 `const std::string CONFIG_FILE = "/usr/local/share/mule/config/analysis/spatial/simulation_cluster_detector.xml"`

Definition at line 33 of file SimulationDetectClusters.cpp.

Referenced by `loadDetectorParameterValues()`, and `saveDetectorParameterValues()`.

8.54.2.2 `const std::string LABEL_EPS = "detector.eps"`

Definition at line 36 of file SimulationDetectClusters.cpp.

Referenced by `loadDetectorParameterValues()`, and `saveDetectorParameterValues()`.

8.54.2.3 `const std::string LABEL_MINPOINTS = "detector.minPoints"`

Definition at line 37 of file SimulationDetectClusters.cpp.

Referenced by `loadDetectorParameterValues()`, and `saveDetectorParameterValues()`.

8.54.2.4 `const std::string LABEL_ROOT_COMMENT = "<xmlcomment>"`

Definition at line 35 of file SimulationDetectClusters.cpp.

Referenced by `saveDetectorParameterValues()`.

8.54.2.5 `const std::string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."`

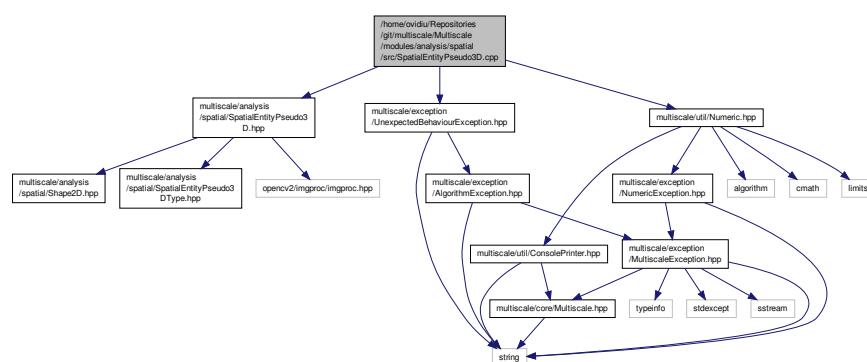
Definition at line 39 of file SimulationDetectClusters.cpp.

Referenced by `saveDetectorParameterValues()`.

## 8.55 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/← SpatialEntityPseudo3D.cpp File Reference

```
#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/Numeric.hpp"
```

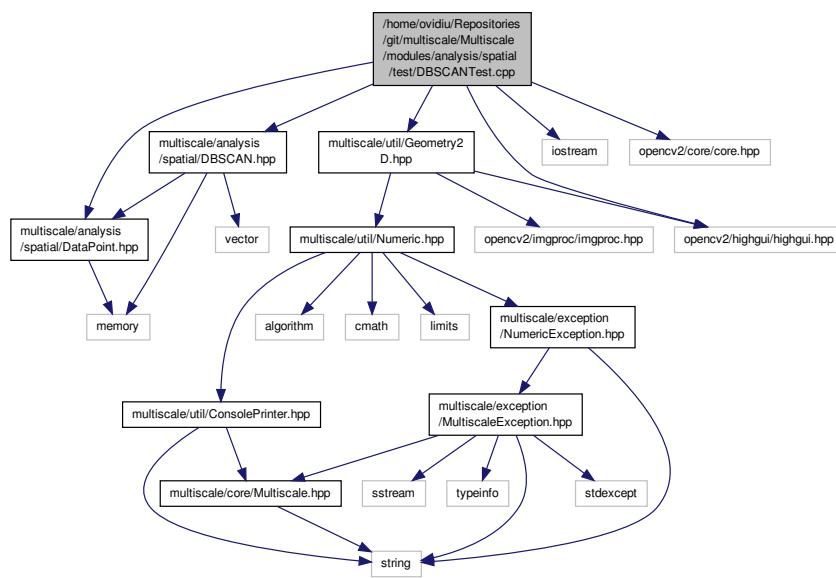
Include dependency graph for SpatialEntityPseudo3D.cpp:



## 8.56 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/ DBSCANTest.cpp File Reference

```
#include "multiscale/analysis/spatial/DataPoint.hpp"
#include "multiscale/analysis/spatial/DBSCAN.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include <iostream>
#include <opencv2/core/core.hpp>
#include <opencv2/highgui/highgui.hpp>
```

Include dependency graph for DBSCANTest.cpp:



### Classes

- class [EuclideanDataPoint](#)

### Functions

- std::vector< std::shared\_ptr< [DataPoint](#) > > [convertPoints](#) (std::vector< [EuclideanDataPoint](#) > &points)
- void [printResults](#) (const std::vector< int > &clusterIndexes)
- void [runTest](#) (std::vector< [EuclideanDataPoint](#) > &points, double eps, int minPoints)
- void [runTest1](#) ()
- void [runTest2](#) ()
- void [runTest3](#) ()
- void [runTest4](#) ()
- void [runTest5](#) ()
- void [runTests](#) ()
- int [main](#) ()

#### 8.56.1 Function Documentation

8.56.1.1 `std::vector<std::shared_ptr<DataPoint>> convertPoints ( std::vector< EuclideanDataPoint > & points )`

Definition at line 35 of file DBSCANTest.cpp.

Referenced by runTest().

8.56.1.2 `int main ( )`

Definition at line 131 of file DBSCANTest.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, and runTests().

8.56.1.3 `void printResults ( const std::vector< int > & clusterIndexes )`

Definition at line 46 of file DBSCANTest.cpp.

Referenced by runTest().

8.56.1.4 `void runTest ( std::vector< EuclideanDataPoint > & points, double eps, int minPoints )`

Definition at line 55 of file DBSCANTest.cpp.

References convertPoints(), printResults(), and multiscale::analysis::DBSCAN::run().

Referenced by runTest1(), runTest2(), runTest3(), runTest4(), and runTest5().

8.56.1.5 `void runTest1 ( )`

Definition at line 65 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.56.1.6 `void runTest2 ( )`

Definition at line 76 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.56.1.7 `void runTest3 ( )`

Definition at line 93 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.56.1.8 `void runTest4 ( )`

Definition at line 104 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.56.1.9 void runTest5( )

Definition at line 111 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.56.1.10 void runTests( )

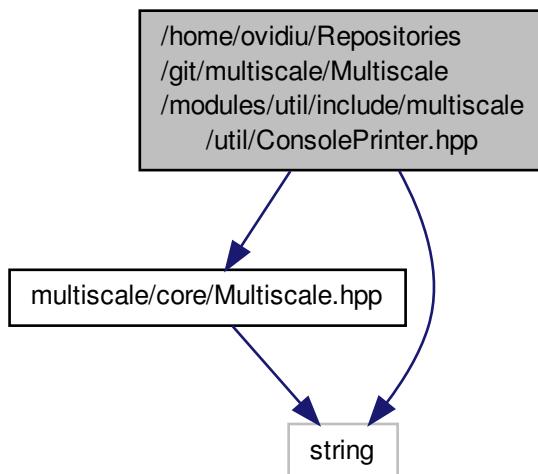
Definition at line 122 of file DBSCANTest.cpp.

References runTest1(), runTest2(), runTest3(), runTest4(), and runTest5().

Referenced by main().

## 8.57 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/ConsolePrinter.hpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include <string>
Include dependency graph for ConsolePrinter.hpp:
```



### Classes

- class [multiscale::ConsolePrinter](#)

*Class used to print (coloured) messages to the console.*

### Namespaces

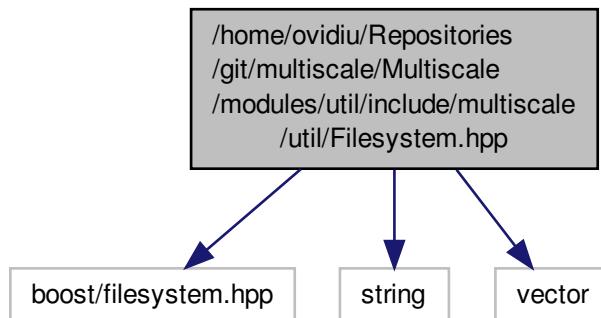
- [multiscale](#)

## Enumerations

- enum multiscale::UnixColourCode : unsigned int {
 multiscale::UnixColourCode::BLACK = 0, multiscale::UnixColourCode::RED = 1, multiscale::UnixColourCode::GREEN = 2, multiscale::UnixColourCode::YELLOW = 3,
 multiscale::UnixColourCode::BLUE = 4, multiscale::UnixColourCode::MAGENTA = 5, multiscale::UnixColourCode::CYAN = 6, multiscale::UnixColourCode::WHITE = 7 }
- enum multiscale::WindowsColourCode : unsigned int {
 multiscale::WindowsColourCode::BLACK = 0, multiscale::WindowsColourCode::DARK\_BLUE = 1,
 multiscale::WindowsColourCode::DARK\_GREEN = 2, multiscale::WindowsColourCode::DARK\_CYAN = 3,
 multiscale::WindowsColourCode::DARK\_RED = 4, multiscale::WindowsColourCode::DARK\_MAGENTA = 5,
 multiscale::WindowsColourCode::DARK\_YELLOW = 6, multiscale::WindowsColourCode::DARK\_WHITE = 7,
 multiscale::WindowsColourCode::GRAY = 8, multiscale::WindowsColourCode::BLUE = 9, multiscale::WindowsColourCode::GREEN = 10, multiscale::WindowsColourCode::CYAN = 11,
 multiscale::WindowsColourCode::RED = 12, multiscale::WindowsColourCode::MAGENTA = 13, multiscale::WindowsColourCode::YELLOW = 14, multiscale::WindowsColourCode::WHITE = 15 }
- enum multiscale::ColourCode : unsigned int {
 multiscale::ColourCode::BLACK = 0, multiscale::ColourCode::RED = 1, multiscale::ColourCode::GREEN = 2,
 multiscale::ColourCode::YELLOW = 3, multiscale::ColourCode::BLUE = 4, multiscale::ColourCode::MAGENTA = 5, multiscale::ColourCode::CYAN = 6, multiscale::ColourCode::WHITE = 7 }

## 8.58 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Filesyste... File Reference

```
#include <boost/filesystem.hpp>
#include <string>
#include <vector>
Include dependency graph for Filesystem.hpp:
```



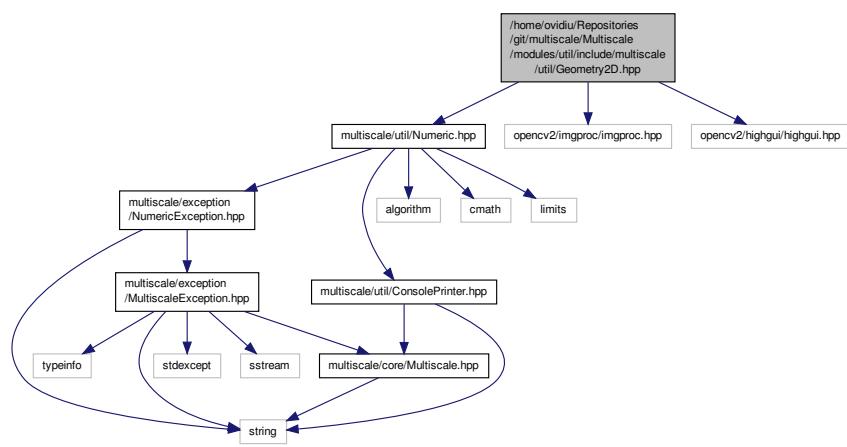
## Classes

- class multiscale::Filesystem  
*Class containing methods for interacting with the filesystem.*

- multiscale

## 8.59 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/Geometry2D.hpp File Reference

```
#include "multiscale/util/Numeric.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/highgui/highgui.hpp"
Include dependency graph for Geometry2D.hpp:
```



## Classes

- class [multiscale::Geometry2D](#)

*Two-dimensional geometric operations.*

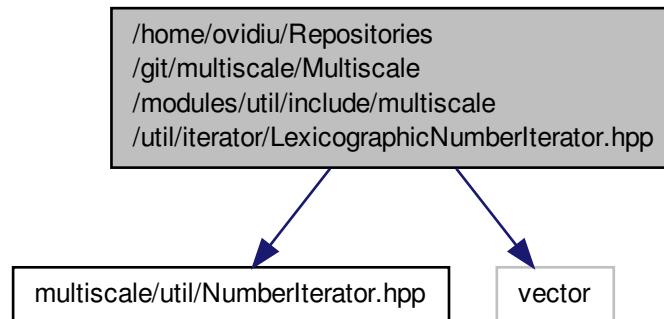
## Namespaces

- multiscale

## 8.60 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/LexicographicNumberIterator.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
#include <vector>
```

Include dependency graph for LexicographicNumberIterator.hpp:



## Classes

- class [multiscale::LexicographicNumberIterator](#)

*Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "—" .*

## Namespaces

- [multiscale](#)

## 8.61 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/NumberIteratorType.hpp File Reference

### Namespaces

- [multiscale](#)

### Enumerations

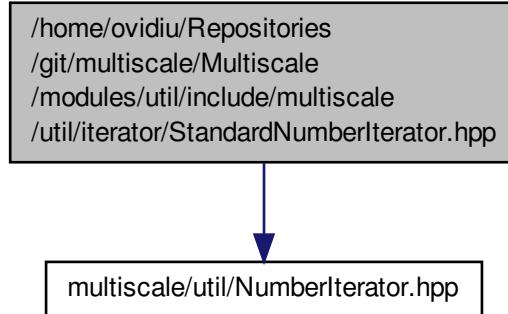
- enum [multiscale::NumberIteratorType](#) { [multiscale::STANDARD](#) = 1, [multiscale::LEXICOGRAPHIC](#) = 2 }

*The type of the number iterator.*

## 8.62 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/StandardNumberIterator.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
```

Include dependency graph for StandardNumberIterator.hpp:



## Classes

- class [multiscale::StandardNumberIterator](#)

*Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.*

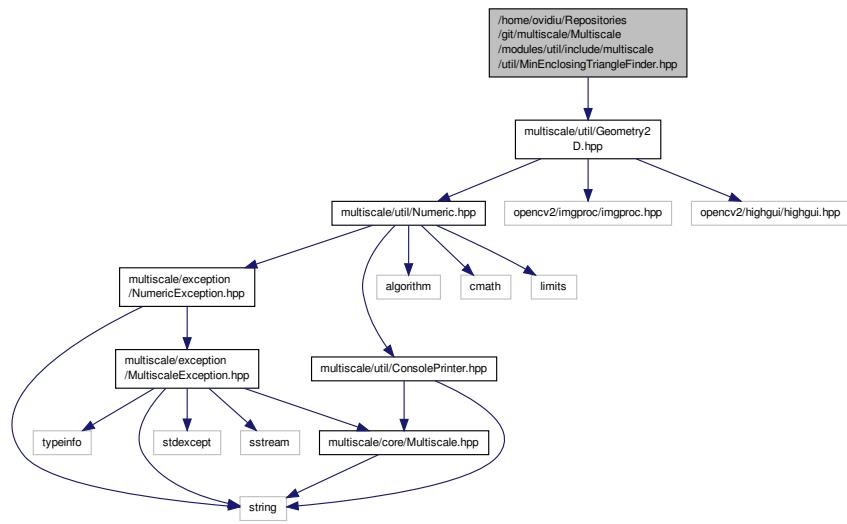
## Namespaces

- [multiscale](#)

## 8.63 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/MinEnclosingTriangleFinder.hpp File Reference

```
#include "multiscale/util/Geometry2D.hpp"
```

Include dependency graph for MinEnclosingTriangleFinder.hpp:



## Classes

- class [multiscale::MinEnclosingTriangleFinder](#)  
*Class for computing the minimum area enclosing triangle for a given polygon.*

## Namespaces

- multiscale

## 8.64 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumberIterator.hpp File Reference

## Classes

- class multiscale::NumberIterator  
*Abstract class representing a number iterator.*

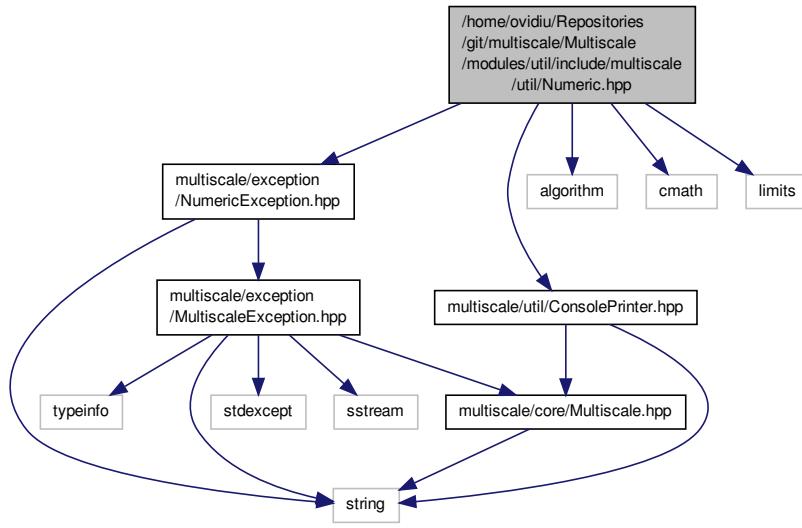
## Namespaces

- multiscale

## 8.65 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/← Numeric.hpp File Reference

```
#include "multiscale/exception/NumericException.hpp"
#include "multiscale/util/ConsolePrinter.hpp"
#include <algorithm>
#include <cmath>
#include <limits>
```

Include dependency graph for Numeric.hpp:



## Classes

- class `multiscale::AdditionOperation`  
*Functor representing an addition operation.*
- class `multiscale::DivisionOperation`  
*Functor representing a division operation.*
- class `multiscale::MultiplicationOperation`  
*Functor representing a multiplication operation.*
- class `multiscale::SubtractionOperation`  
*Functor representing a subtraction operation.*
- class `multiscale::Numeric`  
*Class for processing numeric (shorts, ints, floats, doubles etc.) expressions.*

## Namespaces

- `multiscale`

## 8.66 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/NumericRangeManipulator.hpp File Reference

## Classes

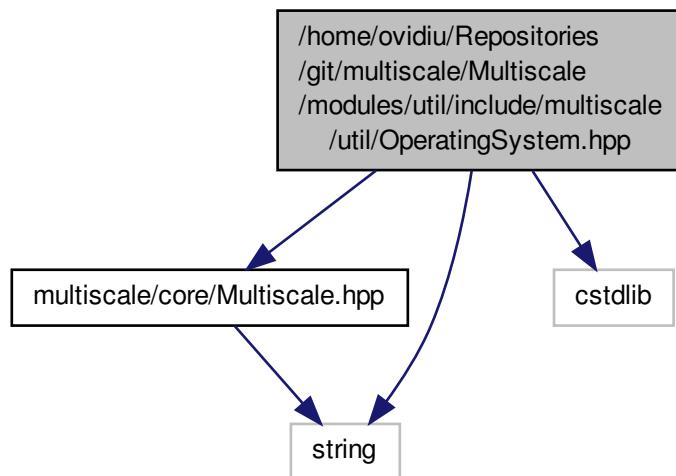
- class `multiscale::NumericRangeManipulator`  
*Operations for ranges of numeric values.*

## Namespaces

- `multiscale`

## 8.67 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/... OperatingSystem.hpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include <cstdlib>
#include <string>
Include dependency graph for OperatingSystem.hpp:
```



### Classes

- class [multiscale::OperatingSystem](#)

*Class for executing operating system related functions.*

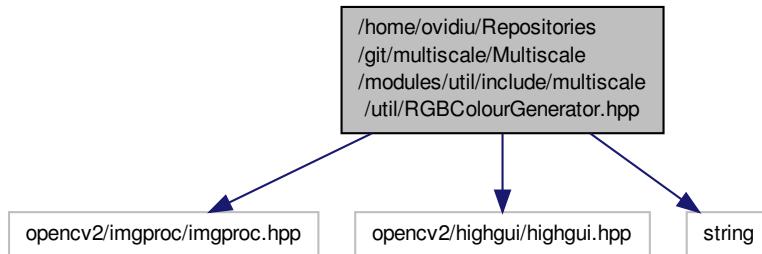
### Namespaces

- [multiscale](#)

## 8.68 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/... RGBColourGenerator.hpp File Reference

```
#include <opencv2/imgproc/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <string>
```

Include dependency graph for RGBColourGenerator.hpp:



## Classes

- class [multiscale::RGBColourGenerator](#)

*Generate a RGB colour.*

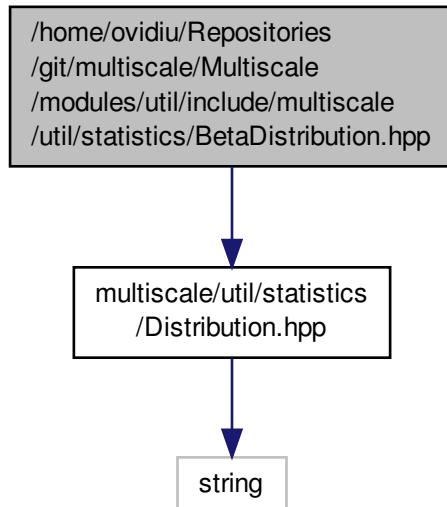
## Namespaces

- [multiscale](#)

## 8.69 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/ BetaDistribution.hpp File Reference

```
#include "multiscale/util/statistics/Distribution.hpp"
```

Include dependency graph for BetaDistribution.hpp:



## Classes

- class [multiscale::BetaDistribution](#)

*Class for analysing Beta distributed data.*

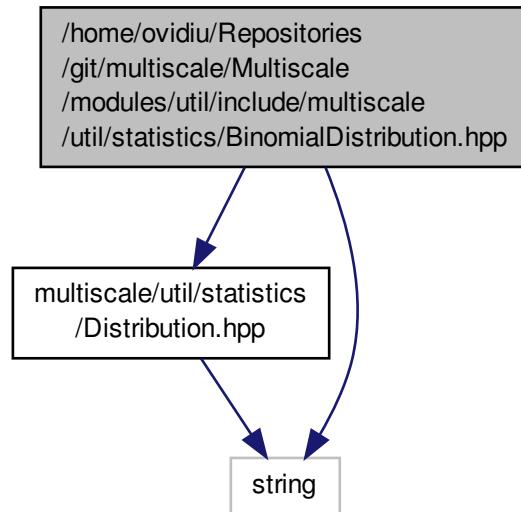
## Namespaces

- [multiscale](#)

## 8.70 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/ BinomialDistribution.hpp File Reference

```
#include "multiscale/util/statistics/Distribution.hpp"
#include <string>
```

Include dependency graph for BinomialDistribution.hpp:



## Classes

- class [multiscale::BinomialDistribution](#)

*Class for analysing Binomial distributed data.*

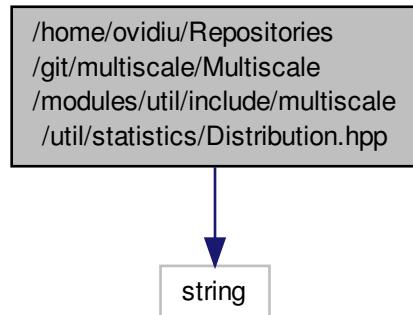
## Namespaces

- [multiscale](#)

## 8.71 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/ Distribution.hpp File Reference

```
#include <string>
```

Include dependency graph for Distribution.hpp:



## Classes

- class [multiscale::Distribution](#)

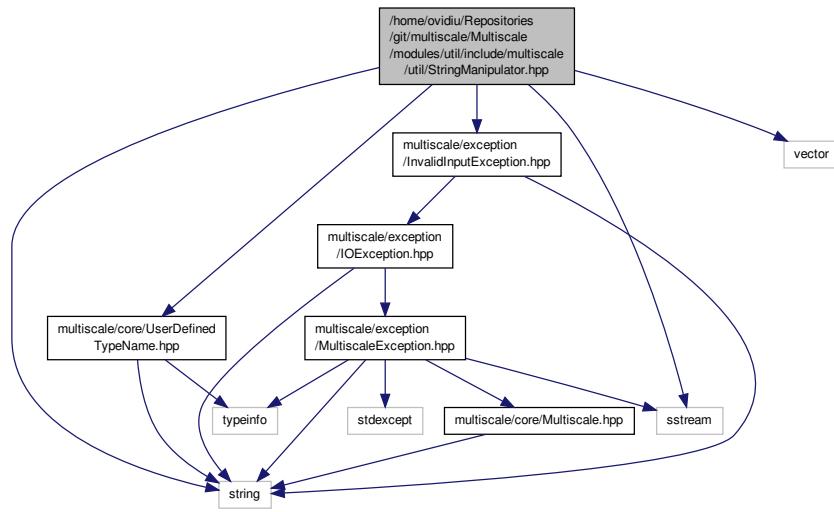
## Namespaces

- [multiscale](#)

## 8.72 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/\_← StringManipulator.hpp File Reference

```
#include "multiscale/core/UserDefinedTypeName.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include <string>
#include <vector>
#include <iostream>
```

Include dependency graph for StringManipulator.hpp:



## Classes

- class multiscale::StringManipulator  
*Class for manipulating std::strings.*

# Namespaces

- multiscale

## 8.73 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/ XmlValidator.hpp File Reference

```
#include <xercesc/framework/LocalFileInputSource.hpp>
#include <xercesc/parsers/XercesDOMParser.hpp>
#include <xercesc/sax/Error Handler.hpp>
#include <xercesc/sax/SAXParseException.hpp>
#include <xercesc/validators/common/Grammar.hpp>
#include <string>
```

**Include dependency graph for XmlValidator.hpp:**

Include dependency graph for XmlValidator.hpp:



## Classes

- class multiscale::XmlValidator

*Class used to validate xml files.*

- class [multiscale::XmlValidator::XmlValidationErrorHandler](#)

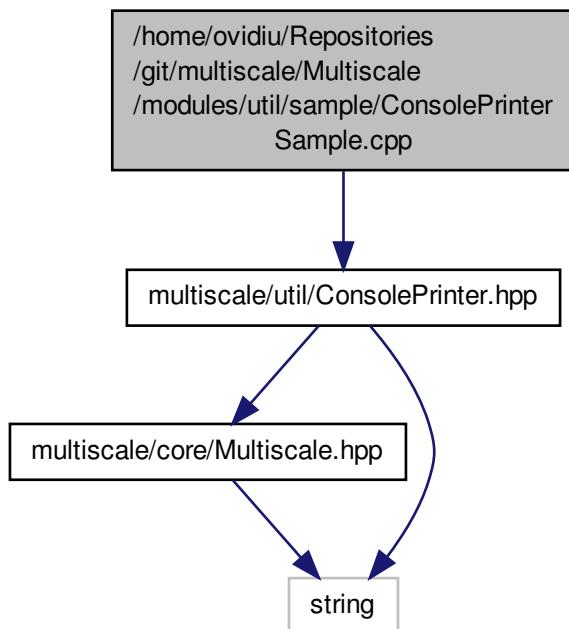
*Class used for handling errors during the xml file validation process.*

## Namespaces

- [multiscale](#)

## 8.74 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/ConsolePrinterSample.cpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp"
Include dependency graph for ConsolePrinterSample.cpp:
```



## Functions

- int [main \(\)](#)

## Variables

- const std::string [SAMPLE\\_TAG](#) = "[ SAMPLE ]"
- const std::string [SAMPLE\\_MSG](#) = "This is a sample message."

## 8.74.1 Function Documentation

### 8.74.1.1 int main( )

Definition at line 10 of file ConsolePrinterSample.cpp.

References multiscale::BLUE, multiscale::CYAN, multiscale::MAGENTA, multiscale::ConsolePrinter::printColouredMessage(), multiscale::ConsolePrinter::printColouredMessageWithColouredTag(), multiscale::ConsolePrinter::printMessage(), multiscale::ConsolePrinter::printMessageWithColouredTag(), multiscale::RED, SAMPLE\_MSG, and SAMPLE\_TAG.

## 8.74.2 Variable Documentation

### 8.74.2.1 const std::string SAMPLE\_MSG = "This is a sample message."

Definition at line 6 of file ConsolePrinterSample.cpp.

Referenced by main().

### 8.74.2.2 const std::string SAMPLE\_TAG = "[ SAMPLE ]"

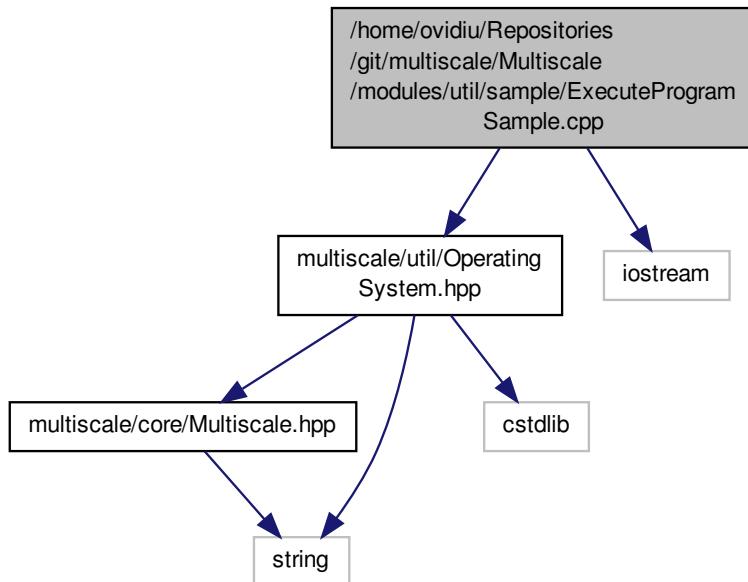
Definition at line 5 of file ConsolePrinterSample.cpp.

Referenced by main().

## 8.75 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/ExecuteProgramSample.cpp File Reference

```
#include "multiscale/util/OperatingSystem.hpp"
#include <iostream>
```

Include dependency graph for ExecuteProgramSample.cpp:



## Functions

- int `main` (int argc, char \*\*argv)

### 8.75.1 Function Documentation

#### 8.75.1.1 int main ( int argc, char \*\* argv )

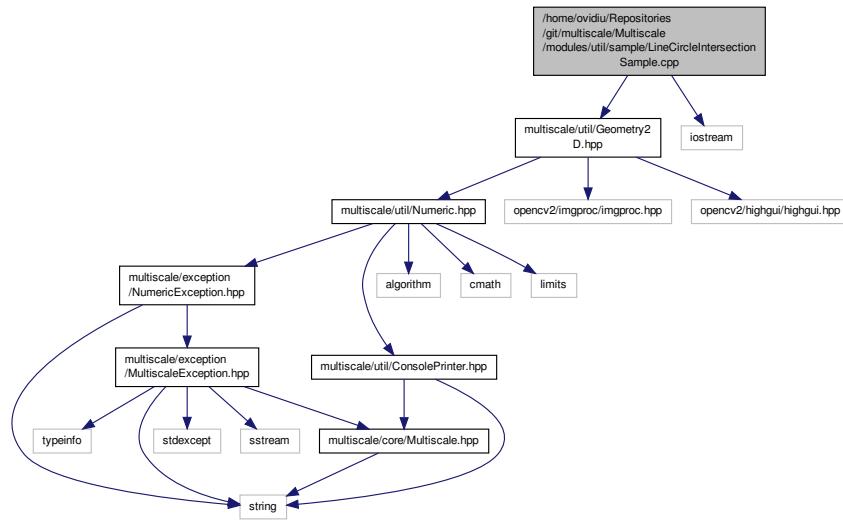
Definition at line 9 of file `ExecuteProgramSample.cpp`.

References `multiscale::EXEC_SUCCESS_CODE`, and `multiscale::OperatingSystem::executeProgram()`.

## 8.76 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/LineCircleIntersectionSample.cpp File Reference

```
#include "multiscale/util/Geometry2D.hpp"
#include <iostream>
```

Include dependency graph for LineCircleIntersectionSample.cpp:



## Functions

- void [printPoints](#) (const std::vector< cv::Point2f > &points)
- int [main](#) ()

### 8.76.1 Function Documentation

#### 8.76.1.1 int main ( )

Definition at line 20 of file LineCircleIntersectionSample.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, multiscale::Geometry2D::lineCircleIntersection(), multiscale::Geometry2D::lineSegmentCircleIntersection(), and printPoints().

#### 8.76.1.2 void printPoints ( const std::vector< cv::Point2f > & points )

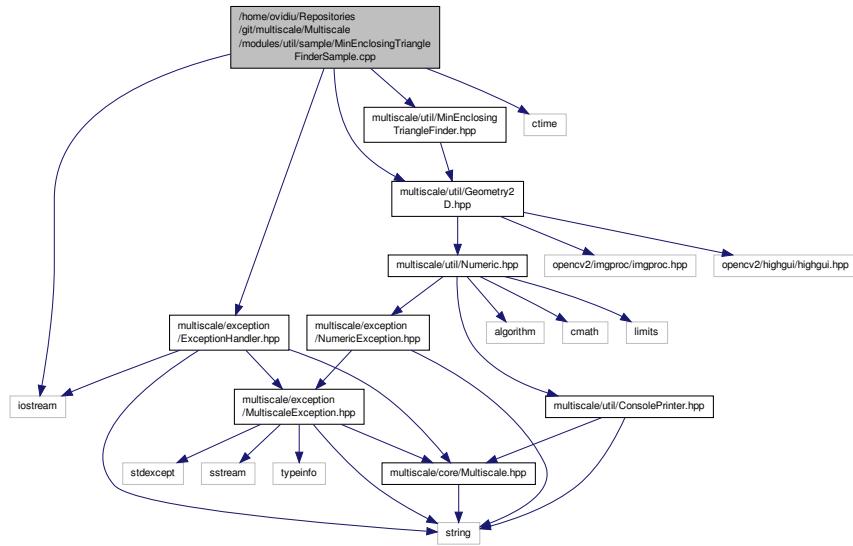
Definition at line 10 of file LineCircleIntersectionSample.cpp.

Referenced by main().

## 8.77 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/MinEnclosingTriangleFinderSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include "multiscale/util/MinEnclosingTriangleFinder.hpp"
#include <ctime>
#include <iostream>
```

Include dependency graph for MinEnclosingTriangleFinderSample.cpp:



## Functions

- std::vector< cv::Point2f > [generateRandomSetOf2DPoints](#) (int nrOfPoints)
- void [printPolygon](#) (const std::vector< cv::Point2f > &points)
- void [outputMinEnclosingTriangleFinderResults](#) (const std::vector< cv::Point2f > &minEnclosingTriangle, const std::vector< cv::Point2f > &points)
- bool [arePointsEnclosed](#) (const std::vector< cv::Point2f > &points, const std::vector< cv::Point2f > &triangle)
- bool [isTriangleTouchingPolygon](#) (const std::vector< cv::Point2f > &convexPolygon, const std::vector< cv::Point2f > &triangle)
- bool [isOneEdgeFlush](#) (const std::vector< cv::Point2f > &convexPolygon, const std::vector< cv::Point2f > &triangle)
- bool [isValidTriangle](#) (const std::vector< cv::Point2f > &points, const std::vector< cv::Point2f > &triangle)
- void [runMinEnclosingTriangleFinder](#) (const std::vector< cv::Point2f > &points)
- void [runMinEnclosingTriangleFinderUsingRandomPolygons](#) ()
- void [runMinEnclosingTriangleFinder](#) ()
- int [main](#) (int argc, char \*\*argv)

## Variables

- const std::string [WIN\\_MIN\\_AREA\\_TRIANGLE](#) = "Minimum area enclosing triangle"
- const int [KEY\\_ESC](#) = 27
- const int [RADIUS](#) = 1
- const int [LINE\\_THICKNESS](#) = 50
- const int [NR RAND POLYGONS](#) = 50
- const int [MAX POLYGON POINTS](#) = 100
- const int [POLYGON POINT X MAX](#) = 500
- const int [POLYGON POINT Y MAX](#) = 500
- const double [POINT IN TRIANGLE THRESH](#) = 1E-4

---

8.77.1 Function Documentation

8.77.1.1 `bool arePointsEnclosed ( const std::vector< cv::Point2f > & points, const std::vector< cv::Point2f > & triangle )`

Definition at line 82 of file MinEnclosingTriangleFinderSample.cpp.

References POINT\_IN\_TRIANGLE\_THRESH.

Referenced by isValidTriangle().

8.77.1.2 `std::vector<cv::Point2f> generateRandomSetOf2DPoints ( int nrOfPoints )`

Definition at line 26 of file MinEnclosingTriangleFinderSample.cpp.

References POLYGON\_POINT\_X\_MAX, and POLYGON\_POINT\_Y\_MAX.

Referenced by runMinEnclosingTriangleFinderUsingRandomPolygons().

8.77.1.3 `bool isOneEdgeFlush ( const std::vector< cv::Point2f > & convexPolygon, const std::vector< cv::Point2f > & triangle )`

Definition at line 121 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::Geometry2D::isPointOnLineSegment().

Referenced by isValidTriangle().

8.77.1.4 `bool isTriangleTouchingPolygon ( const std::vector< cv::Point2f > & convexPolygon, const std::vector< cv::Point2f > & triangle )`

Definition at line 97 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::Geometry2D::isPointOnLineSegment(), and multiscale::Geometry2D::middlePoint().

Referenced by isValidTriangle().

8.77.1.5 `bool isValidTriangle ( const std::vector< cv::Point2f > & points, const std::vector< cv::Point2f > & triangle )`

Definition at line 139 of file MinEnclosingTriangleFinderSample.cpp.

References arePointsEnclosed(), isOneEdgeFlush(), and isTriangleTouchingPolygon().

Referenced by runMinEnclosingTriangleFinder().

8.77.1.6 `int main ( int argc, char ** argv )`

Definition at line 194 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(), and runMinEnclosingTriangleFinder().

8.77.1.7 `void outputMinEnclosingTriangleFinderResults ( const std::vector< cv::Point2f > & minEnclosingTriangle, const std::vector< cv::Point2f > & points )`

Definition at line 54 of file MinEnclosingTriangleFinderSample.cpp.

References LINE\_THICKNESS, POLYGON\_POINT\_X\_MAX, POLYGON\_POINT\_Y\_MAX, printPolygon(), RADIUS, and WIN\_MIN\_AREA\_TRIANGLE.

Referenced by runMinEnclosingTriangleFinder().

8.77.1.8 void printPolygon ( const std::vector< cv::Point2f > & *points* )

Definition at line 38 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

8.77.1.9 void runMinEnclosingTriangleFinder ( const std::vector< cv::Point2f > & *points* )

Definition at line 152 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::MinEnclosingTriangleFinder::find(), isValidTriangle(), and outputMinEnclosingTriangleFinderResults().

Referenced by main(), and runMinEnclosingTriangleFinderUsingRandomPolygons().

8.77.1.10 void runMinEnclosingTriangleFinder ( )

Definition at line 189 of file MinEnclosingTriangleFinderSample.cpp.

References runMinEnclosingTriangleFinderUsingRandomPolygons().

8.77.1.11 void runMinEnclosingTriangleFinderUsingRandomPolygons ( )

Definition at line 168 of file MinEnclosingTriangleFinderSample.cpp.

References generateRandomSetOf2DPoints(), KEY\_ESC, MAX\_POLYGON\_POINTS, and runMinEnclosingTriangleFinder().

Referenced by runMinEnclosingTriangleFinder().

## 8.77.2 Variable Documentation

8.77.2.1 const int KEY\_ESC = 27

Definition at line 13 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by runMinEnclosingTriangleFinderUsingRandomPolygons().

8.77.2.2 const int LINE\_THICKNESS = 50

Definition at line 16 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

8.77.2.3 const int MAX\_POLYGON\_POINTS = 100

Definition at line 18 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by runMinEnclosingTriangleFinderUsingRandomPolygons().

8.77.2.4 const int NR RAND POLYGONS = 50

Definition at line 17 of file MinEnclosingTriangleFinderSample.cpp.

---

8.77.2.5 `const double POINT_IN_TRIANGLE_THRESH = 1E-4`

Definition at line 22 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by multiscaletest::MinEnclosingTriangleFinderTest::ArePointsEnclosed(), and arePointsEnclosed().

8.77.2.6 `const int POLYGON_POINT_X_MAX = 500`

Definition at line 19 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by generateRandomSetOf2DPoints(), and outputMinEnclosingTriangleFinderResults().

8.77.2.7 `const int POLYGON_POINT_Y_MAX = 500`

Definition at line 20 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by generateRandomSetOf2DPoints(), and outputMinEnclosingTriangleFinderResults().

8.77.2.8 `const int RADIUS = 1`

Definition at line 15 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

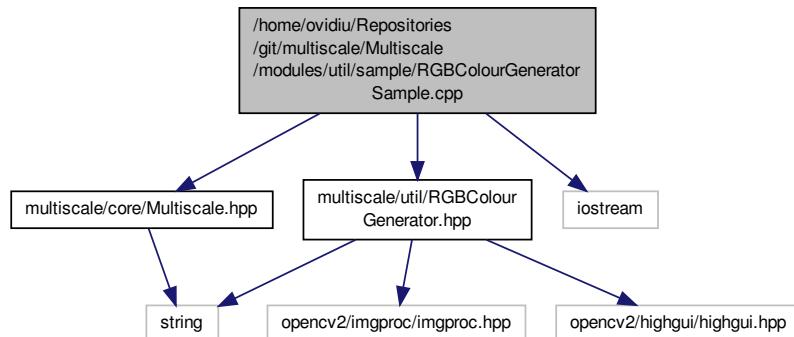
8.77.2.9 `const std::string WIN_MIN_AREA_TRIANGLE = "Minimum area enclosing triangle"`

Definition at line 11 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

## 8.78 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/RGBColourGeneratorSample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/util/RGBColourGenerator.hpp"
#include <iostream>
Include dependency graph for RGBColourGeneratorSample.cpp:
```



## Functions

- int [main \(\)](#)

### 8.78.1 Function Documentation

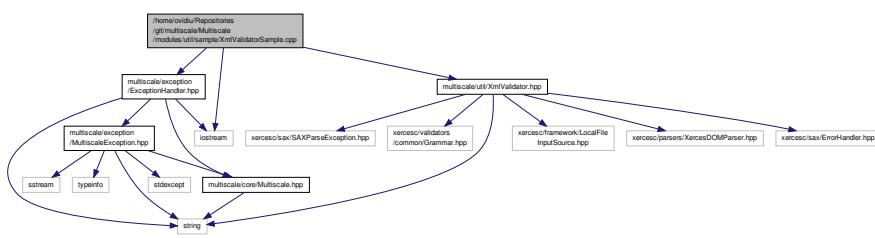
#### 8.78.1.1 int main ( )

Definition at line 9 of file RGBColourGeneratorSample.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, and multiscale::RGBColourGenerator::generate().

## 8.79 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/XmlValidatorSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/util/XmlValidator.hpp"
#include <iostream>
Include dependency graph for XmlValidatorSample.cpp:
```



## Functions

- void [printValidationResult](#) (bool validationResult, const std::string &xmlFilepath, const std::string &xmlError Message)
- void [checkIfValidXmlFile](#) (const std::string &xmlFilepath, const std::string &xmlSchemaFilepath)
- void [validateXmlFile](#) (int argc, char \*\*argv)
- int [main](#) (int argc, char \*\*argv)

### 8.79.1 Function Documentation

#### 8.79.1.1 void checkIfValidXmlFile ( const std::string & xmlFilepath, const std::string & xmlSchemaFilepath )

Definition at line 23 of file XmlValidatorSample.cpp.

References multiscale::XmlValidator::isValidXmlFile(), and printValidationResult().

Referenced by validateXmlFile().

#### 8.79.1.2 int main ( int argc, char \*\* argv )

Definition at line 44 of file XmlValidatorSample.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, and validateXmlFile().

8.79.1.3 void printValidationResult ( bool validationResult, const std::string & xmlfilepath, const std::string & xmleErrorMessage )

Definition at line 10 of file XmlValidatorSample.cpp.

Referenced by checkIfValidXmlFile().

8.79.1.4 void validateXmlFile ( int argc, char \*\* argv )

Definition at line 32 of file XmlValidatorSample.cpp.

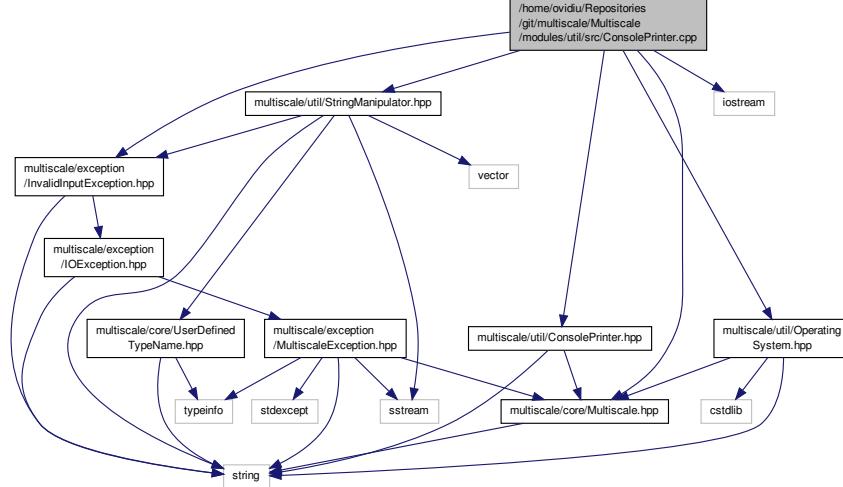
References checkIfValidXmlFile(), and multiscale::ExceptionHandler::printDetailedErrorMessage().

Referenced by main().

## 8.80 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ConsolePrinter.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/ConsolePrinter.hpp"
#include "multiscale/util/OperatingSystem.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <iostream>
```

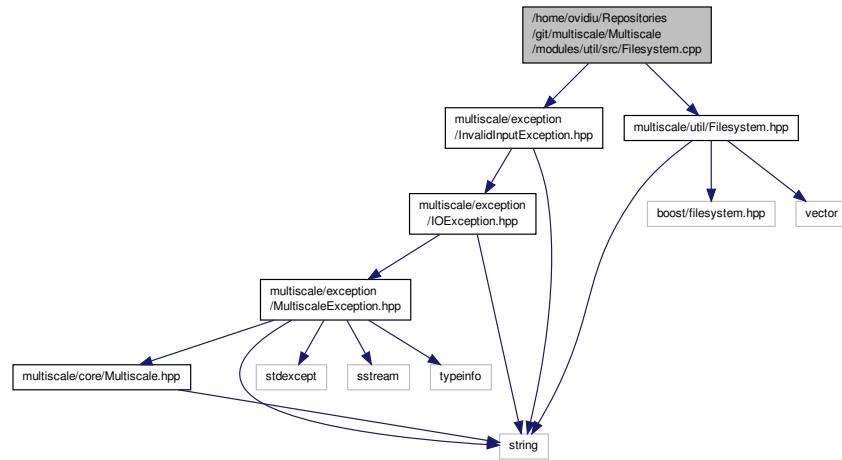
Include dependency graph for ConsolePrinter.cpp:



## 8.81 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Filesystem.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/Filesystem.hpp"
```

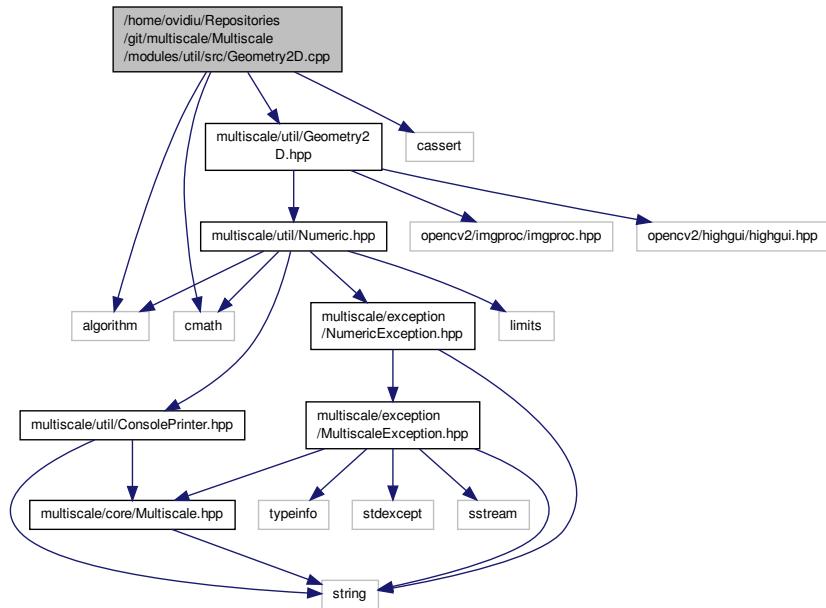
Include dependency graph for Filesystem.cpp:



## 8.82 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Geometry2D.cpp File Reference

```
#include "multiscale/util/Geometry2D.hpp"
#include <algorithm>
#include <cassert>
#include <cmath>
```

Include dependency graph for Geometry2D.cpp:



8.83 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/LexicographicNumberIterator.cpp File

Reference

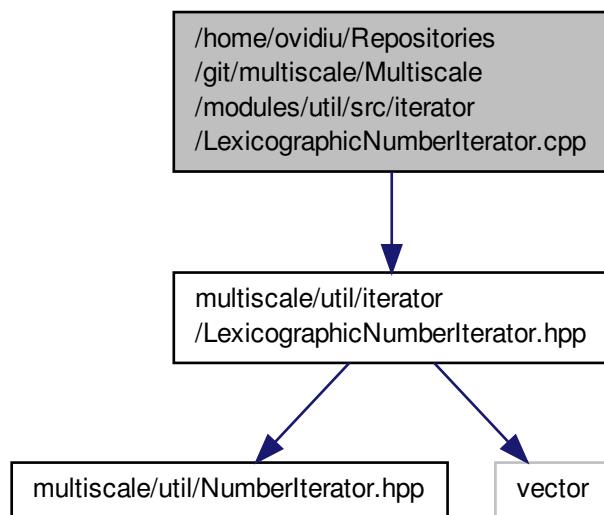
8.83 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/

1135

## LexicographicNumberIterator.cpp File Reference

```
#include "multiscale/util/iterator/LexicographicNumberIterator.hpp"
```

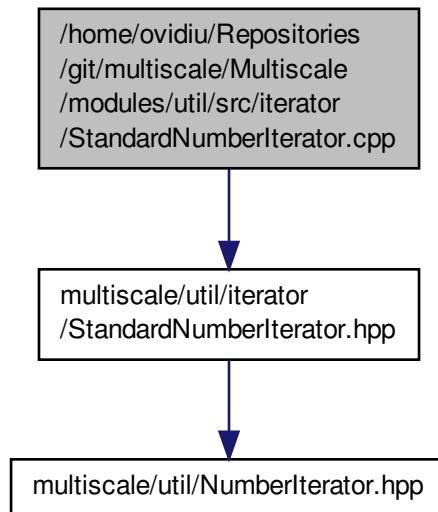
Include dependency graph for LexicographicNumberIterator.cpp:



8.84 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/ StandardNumberIterator.cpp File Reference

```
#include "multiscale/util/iterator/StandardNumberIterator.hpp"
```

Include dependency graph for StandardNumberIterator.cpp:



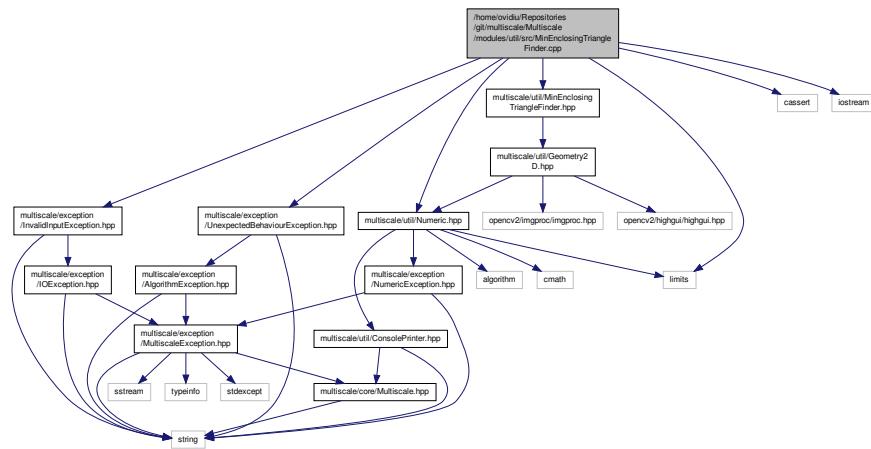
## 8.85 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/MinEnclosingTriangleFinder.cpp File Reference

```

#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/MinEnclosingTriangleFinder.hpp"
#include "multiscale/util/Numeric.hpp"
#include <cassert>
#include <iostream>
#include <limits>

```

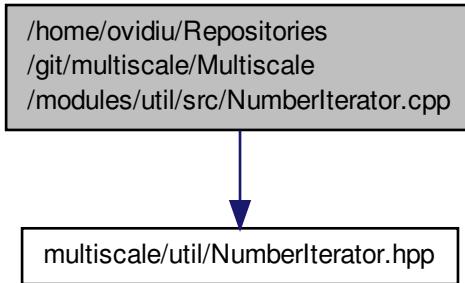
Include dependency graph for MinEnclosingTriangleFinder.cpp:



## 8.86 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/NumberIterator.cpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
```

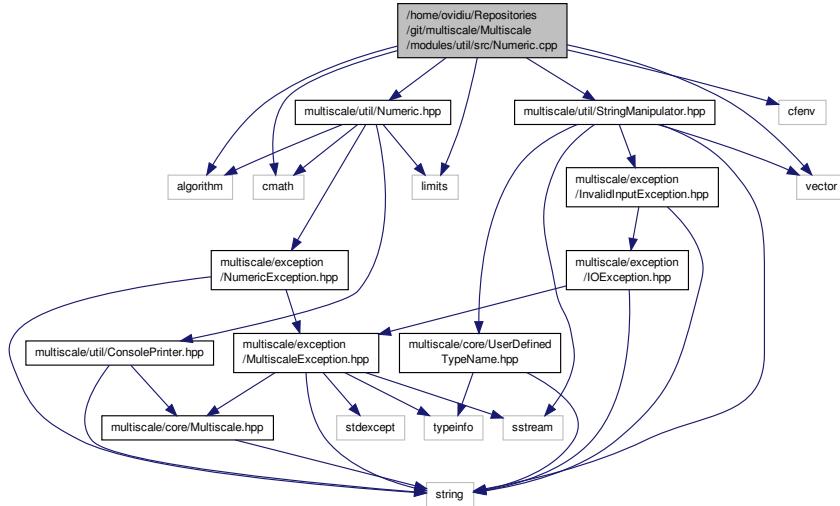
Include dependency graph for NumberIterator.cpp:



## 8.87 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/Numeric.cpp File Reference

```
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <algorithm>
#include <cfenv>
#include <cmath>
#include <limits>
#include <vector>
```

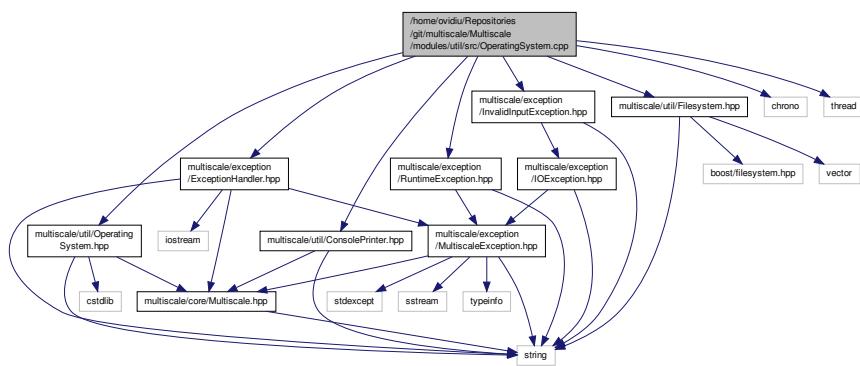
Include dependency graph for Numeric.cpp:



## 8.88 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/OperatingSystem.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/RuntimeException.hpp"
#include "multiscale/util/ConsolePrinter.hpp"
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/util/OperatingSystem.hpp"
#include <chrono>
#include <thread>
```

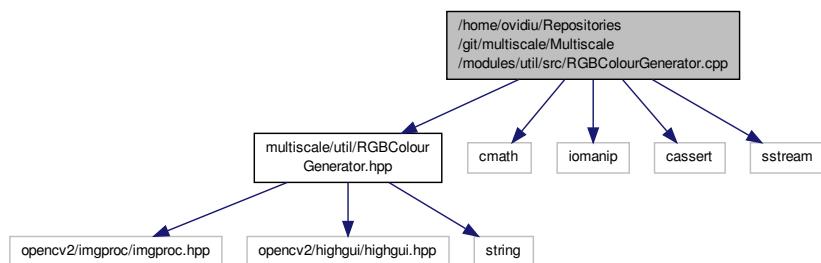
Include dependency graph for OperatingSystem.cpp:



## 8.89 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/RGBColourGenerator.cpp File Reference

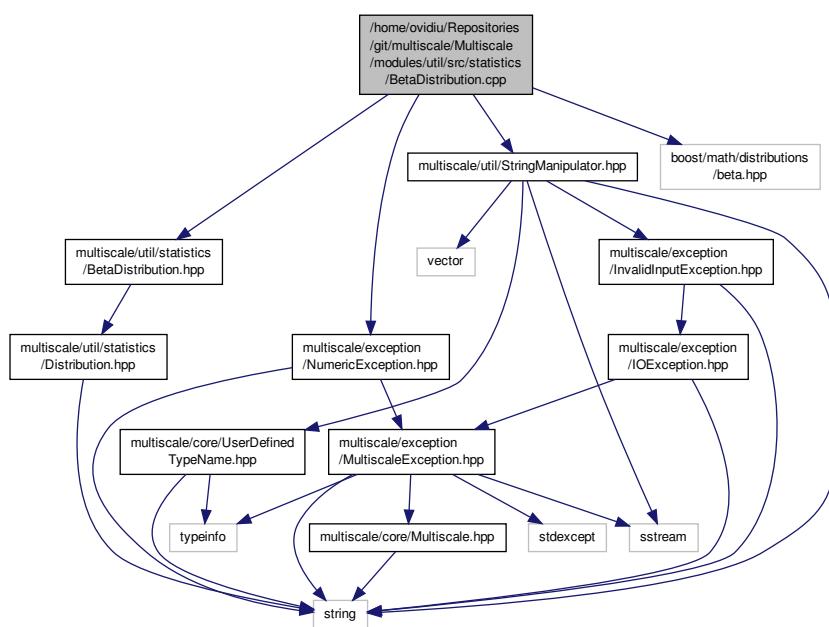
```
#include "multiscale/util/RGBColourGenerator.hpp"
#include <cmath>
#include <iomanip>
#include <cassert>
#include <sstream>
```

Include dependency graph for RGBColourGenerator.cpp:



8.90 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/[BetaDistribution.cpp File Reference](#)

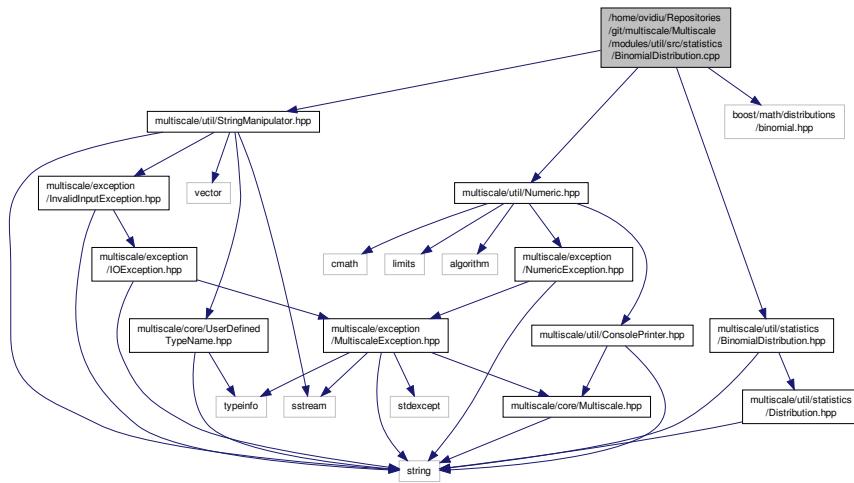
```
#include "multiscale/exception/NumericException.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/statistics/BetaDistribution.hpp"
#include <boost/math/distributions/beta.hpp>
Include dependency graph for BetaDistribution.cpp:
```



8.91 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/[BinomialDistribution.cpp File Reference](#)

```
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/statistics/BinomialDistribution.hpp"
#include <boost/math/distributions/binomial.hpp>
```

Include dependency graph for BinomialDistribution.cpp:



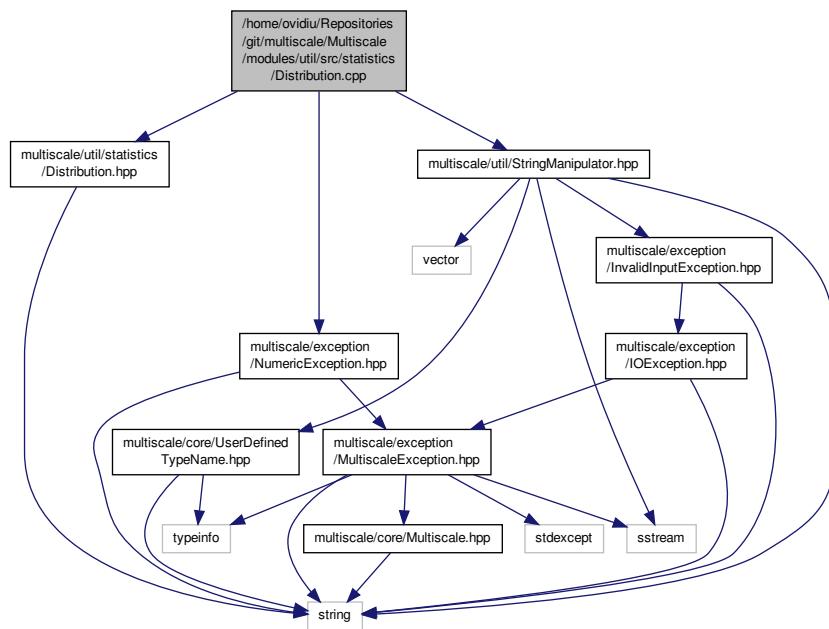
## 8.92 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/← Distribution.cpp File Reference

```

#include "multiscale/exception/NumericException.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/statistics/Distribution.hpp"

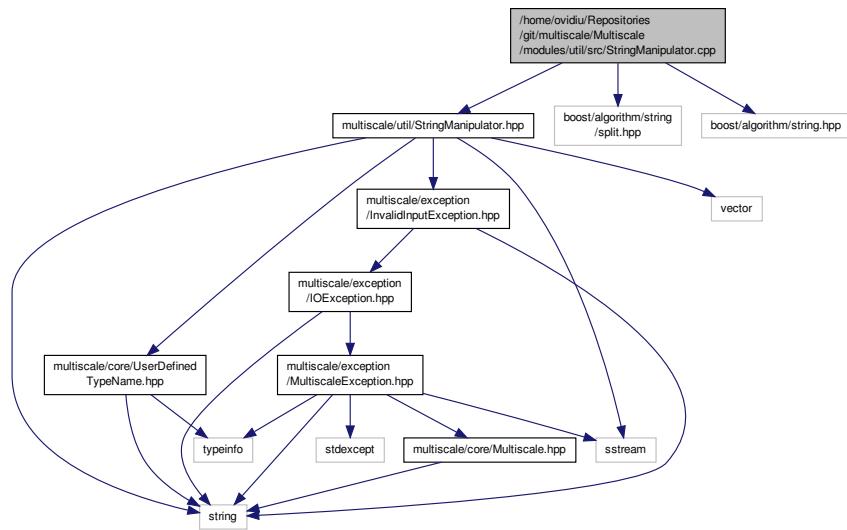
```

Include dependency graph for Distribution.cpp:



## 8.93 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/StringManipulator.cpp

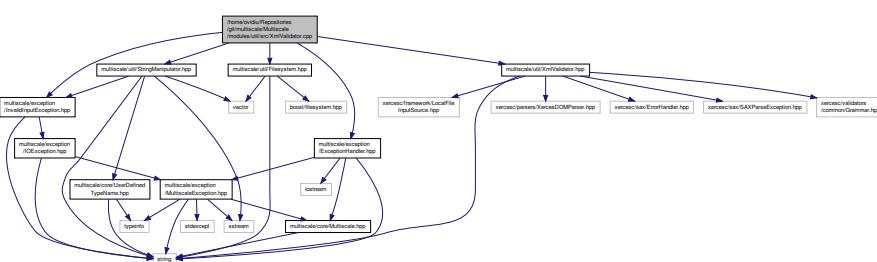
```
#include "multiscale/util/StringManipulator.hpp"
#include <boost/algorithm/string/split.hpp>
#include <boost/algorithm/string.hpp>
Include dependency graph for StringManipulator.cpp:
```



## 8.94 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/XmlValidator.cpp

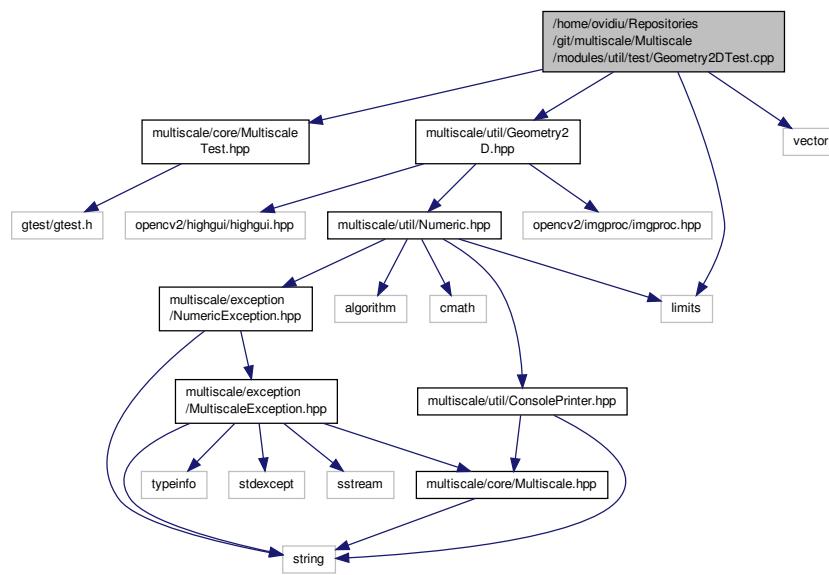
```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/XmlValidator.hpp"
```

Include dependency graph for XmlValidator.cpp:



## 8.95 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/Geometry2DTest.cpp File Reference

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/util/Geometry2D.hpp"
#include <limits>
#include <vector>
Include dependency graph for Geometry2DTest.cpp:
```



### Functions

- `TEST (Geometry2D, TriangleArea)`
- `TEST (Geometry2D, PointOnLineSegment)`
- int `main (int argc, char **argv)`

### Variables

- const double `DOUBLE_COMP_ERROR` = 1E-6

#### 8.95.1 Function Documentation

##### 8.95.1.1 int main ( int argc, char \*\* argv )

Definition at line 35 of file Geometry2DTest.cpp.

##### 8.95.1.2 TEST ( Geometry2D , TriangleArea )

Definition at line 14 of file Geometry2DTest.cpp.

References `multiscale::Geometry2D::areaOfTriangle()`, and `DOUBLE_COMP_ERROR`.

Definition at line 25 of file Geometry2DTest.cpp.

References multiscale::Geometry2D::isPointOnLineSegment().

## 8.95.2 Variable Documentation

### 8.95.2.1 const double DOUBLE\_COMP\_ERROR = 1E-6

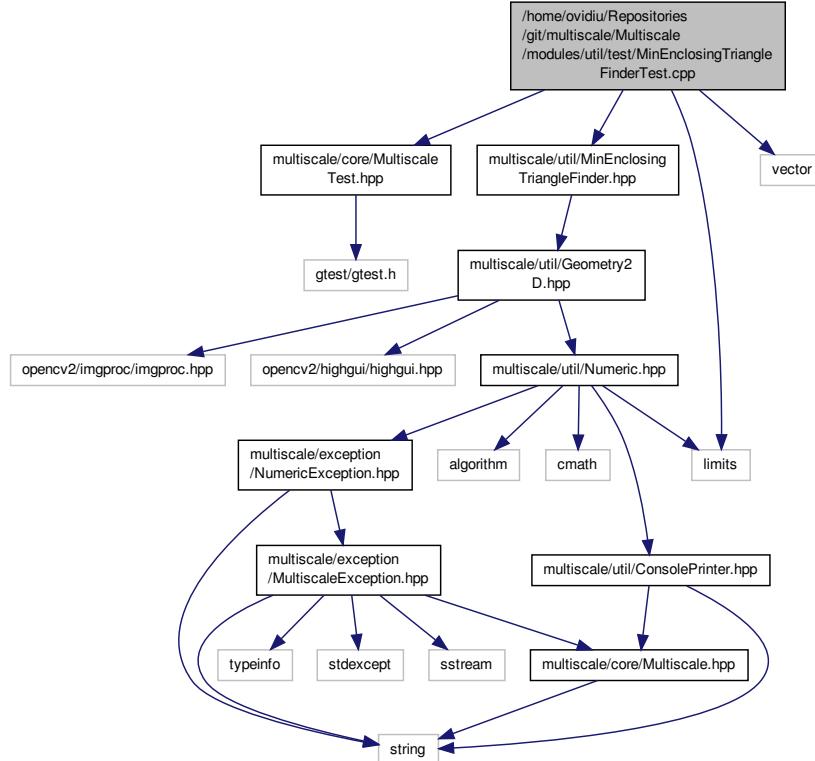
Definition at line 10 of file Geometry2DTest.cpp.

Referenced by TEST().

## 8.96 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/MinEnclosingTriangleFinderTest.cpp File Reference

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/util/MinEnclosingTriangleFinder.hpp"
#include <limits>
#include <vector>
```

Include dependency graph for MinEnclosingTriangleFinderTest.cpp:



## Classes

- class multiscaletest::MinEnclosingTriangleFinderTest

*Class for testing the minimum enclosing triangle algorithm.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(MinEnclosingTriangleFinderTest, TestNoPoints\)](#)
- [TEST\\_F \(MinEnclosingTriangleFinderTest, TestNegativeCoordinates\)](#)
- [TEST\\_F \(MinEnclosingTriangleFinderTest, TestVaryingNumberOfPoints\)](#)
- [TEST\\_F \(MinEnclosingTriangleFinderTest, TestRandomPoints\)](#)
- int [main \(int argc, char \\*\\*argv\)](#)

### 8.96.1 Function Documentation

#### 8.96.1.1 int main ( int argc, char \*\* argv )

Definition at line 334 of file MinEnclosingTriangleFinderTest.cpp.

#### 8.96.1.2 TEST\_F ( MinEnclosingTriangleFinderTest , TestNoPoints )

Definition at line 310 of file MinEnclosingTriangleFinderTest.cpp.

#### 8.96.1.3 TEST\_F ( MinEnclosingTriangleFinderTest , TestNegativeCoordinates )

Definition at line 314 of file MinEnclosingTriangleFinderTest.cpp.

#### 8.96.1.4 TEST\_F ( MinEnclosingTriangleFinderTest , TestVaryingNumberOfPoints )

Definition at line 320 of file MinEnclosingTriangleFinderTest.cpp.

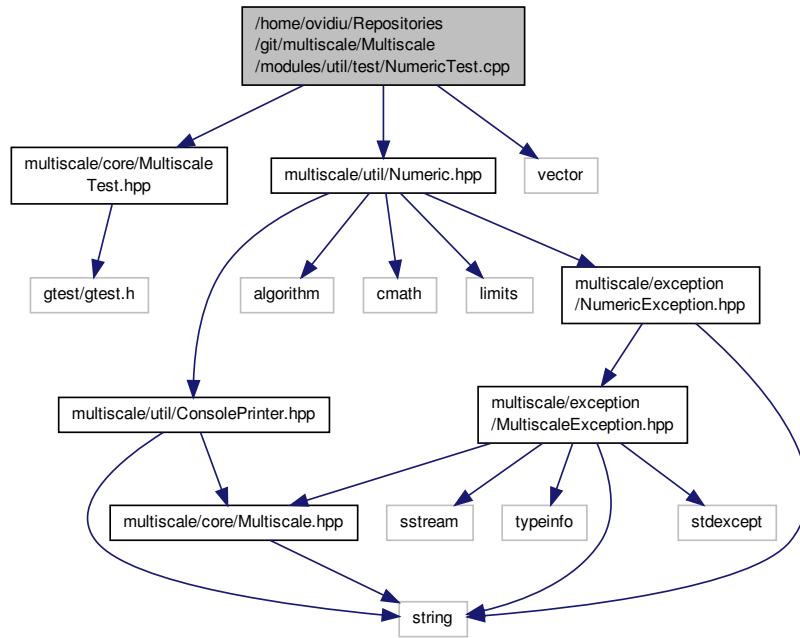
#### 8.96.1.5 TEST\_F ( MinEnclosingTriangleFinderTest , TestRandomPoints )

Definition at line 328 of file MinEnclosingTriangleFinderTest.cpp.

## 8.97 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/NumericTest.cpp File Reference

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/util/Numeric.hpp"
#include <vector>
```

Include dependency graph for NumericTest.cpp:



## Functions

- `TEST (Numeric, GreaterOrEqual)`
- `TEST (Numeric, LessOrEqual)`
- `TEST (Numeric, AlmostEqual)`
- `TEST (Numeric, Average)`
- `TEST (Numeric, Combinations)`
- `TEST (Numeric, Covariance)`
- `TEST (Numeric, Factorial)`
- `TEST (Numeric, GeometricMean)`
- `TEST (Numeric, HarmonicMean)`
- `TEST (Numeric, Kurtosis)`
- `TEST (Numeric, Maximum)`
- `TEST (Numeric, Median)`
- `TEST (Numeric, Minimum)`
- `TEST (Numeric, Mode)`
- `TEST (Numeric, Percentile)`
- `TEST (Numeric, Product)`
- `TEST (Numeric, Quartile)`
- `TEST (Numeric, Skew)`
- `TEST (Numeric, StandardDeviation)`
- `TEST (Numeric, Sum)`
- `TEST (Numeric, Variance)`
- `int main (int argc, char **argv)`

## Variables

- `const double DOUBLE_COMP_ERROR = 1E-6`

### 8.97.1 Function Documentation

#### 8.97.1.1 int main ( int argc, char \*\* argv )

Definition at line 246 of file NumericTest.cpp.

#### 8.97.1.2 TEST( Numeric , GreaterOrEqual )

Definition at line 13 of file NumericTest.cpp.

References multiscale::Numeric::greaterOrEqual().

#### 8.97.1.3 TEST( Numeric , LessOrEqual )

Definition at line 25 of file NumericTest.cpp.

References multiscale::Numeric::lessOrEqual().

#### 8.97.1.4 TEST( Numeric , AlmostEqual )

Definition at line 37 of file NumericTest.cpp.

References multiscale::Numeric::almostEqual().

#### 8.97.1.5 TEST( Numeric , Average )

Definition at line 50 of file NumericTest.cpp.

References multiscale::Numeric::average(), and DOUBLE\_COMP\_ERROR.

#### 8.97.1.6 TEST( Numeric , Combinations )

Definition at line 59 of file NumericTest.cpp.

References multiscale::Numeric::combinations(), and DOUBLE\_COMP\_ERROR.

#### 8.97.1.7 TEST( Numeric , Covariance )

Definition at line 66 of file NumericTest.cpp.

References multiscale::Numeric::covariance(), and DOUBLE\_COMP\_ERROR.

#### 8.97.1.8 TEST( Numeric , Factorial )

Definition at line 75 of file NumericTest.cpp.

References multiscale::Numeric::factorial().

#### 8.97.1.9 TEST( Numeric , GeometricMean )

Definition at line 82 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::geometricMean().

### 8.97.1.10 TEST( Numeric , HarmonicMean )

Definition at line 90 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::harmonicMean().

### 8.97.1.11 TEST( Numeric , Kurtosis )

Definition at line 98 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::kurtosis().

### 8.97.1.12 TEST( Numeric , Maximum )

Definition at line 106 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::maximum().

### 8.97.1.13 TEST( Numeric , Median )

Definition at line 116 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::median().

### 8.97.1.14 TEST( Numeric , Minimum )

Definition at line 127 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::minimum().

### 8.97.1.15 TEST( Numeric , Mode )

Definition at line 138 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::mode().

### 8.97.1.16 TEST( Numeric , Percentile )

Definition at line 151 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::percentile().

### 8.97.1.17 TEST( Numeric , Product )

Definition at line 164 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::product().

### 8.97.1.18 TEST( Numeric , Quartile )

Definition at line 180 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::quartile().

**8.97.1.19 TEST( Numeric , Skew )**

Definition at line 195 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::skew().

**8.97.1.20 TEST( Numeric , StandardDeviation )**

Definition at line 207 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::standardDeviation().

**8.97.1.21 TEST( Numeric , Sum )**

Definition at line 219 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::sum().

**8.97.1.22 TEST( Numeric , Variance )**

Definition at line 232 of file NumericTest.cpp.

References DOUBLE\_COMP\_ERROR, and multiscale::Numeric::variance().

**8.97.2 Variable Documentation****8.97.2.1 const double DOUBLE\_COMP\_ERROR = 1E-6**

Definition at line 9 of file NumericTest.cpp.

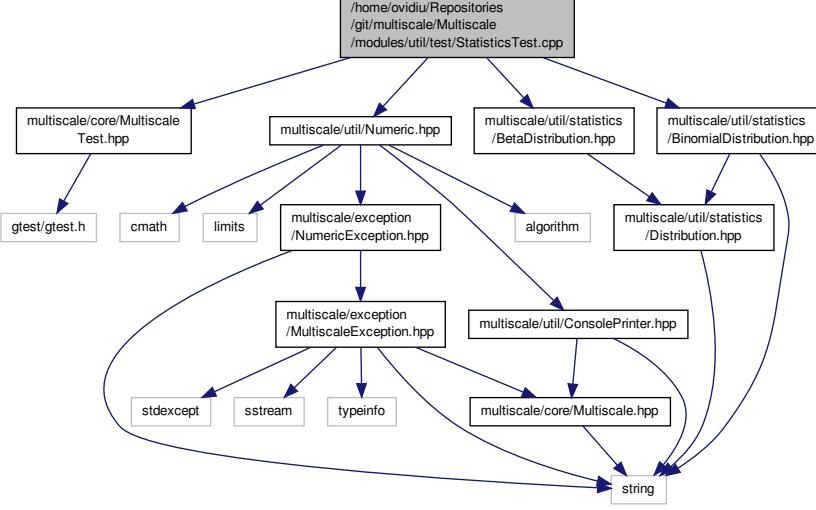
Referenced by TEST().

**8.98 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/StatisticsTest.cpp File Reference**

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/statistics/BetaDistribution.hpp"
#include "multiscale/util/statistics/BinomialDistribution.hpp"
```

**Reference**

Include dependency graph for StatisticsTest.cpp:



## Functions

- [TEST \(Statistics, BinomialPDF\)](#)
- [TEST \(Statistics, BinomialCDF\)](#)
- [TEST \(Statistics, BetaCDF\)](#)

### 8.98.1 Function Documentation

#### 8.98.1.1 TEST ( Statistics , BinomialPDF )

Definition at line 11 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BinomialDistribution::pdf().

#### 8.98.1.2 TEST ( Statistics , BinomialCDF )

Definition at line 21 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BinomialDistribution::cdf().

#### 8.98.1.3 TEST ( Statistics , BetaCDF )

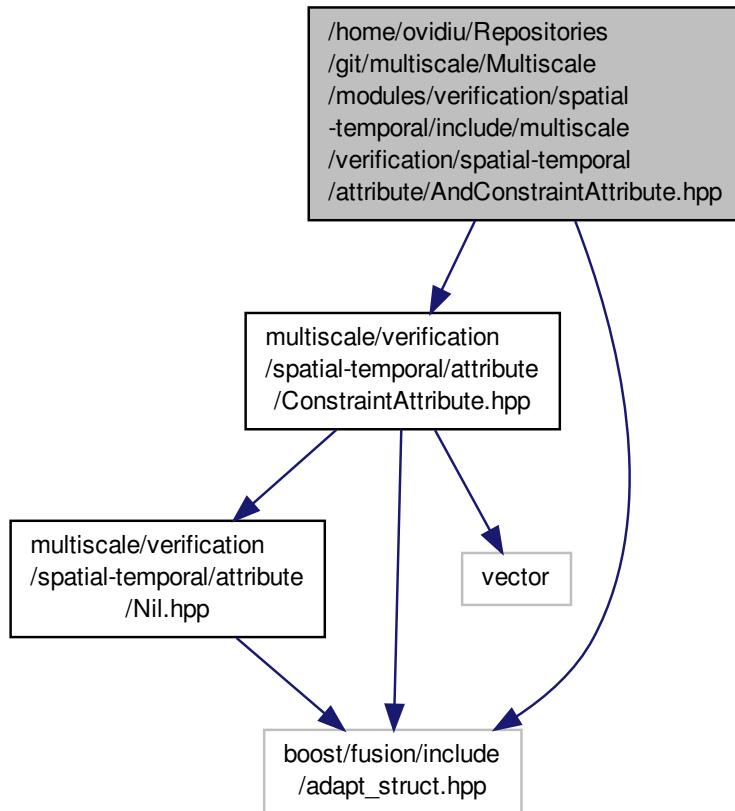
Definition at line 30 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BetaDistribution::cdf().

## 8.99 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
```

```
Attribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for AndConstraintAttribute.hpp:
```



## Classes

- class [multiscale::verification::AndConstraintAttribute](#)  
*Class for representing an "and" constraint attribute.*

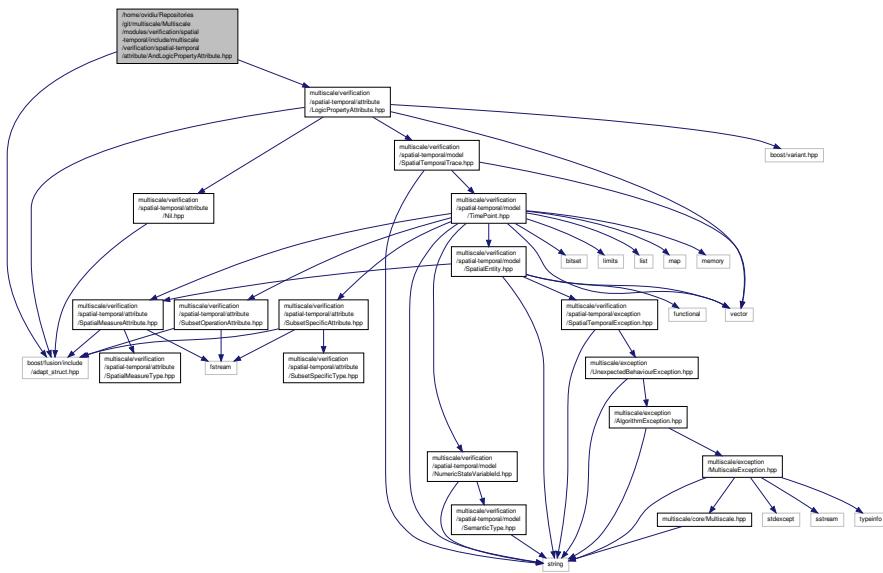
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.100 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for AndLogicPropertyAttribute.hpp:



## Classes

- class multiscale::verification::AndLogicPropertyAttribute

*Class for representing an "and" logic property attribute.*

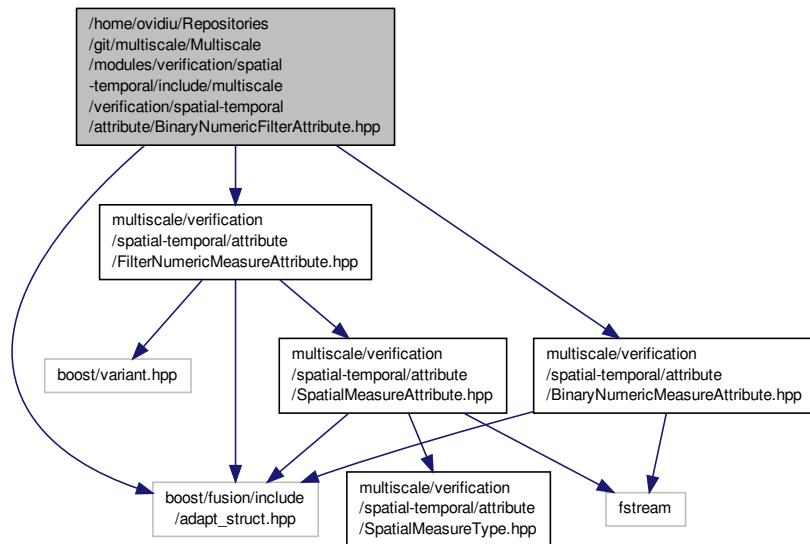
## Namespaces

- multiscale
  - multiscale::verification

8.101 [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericFilterAttribute.hpp](#) File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/FilterNumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for BinaryNumericFilterAttribute.hpp:



## Classes

- class [multiscale::verification::BinaryNumericFilterAttribute](#)

*Class for representing a binary numeric filter attribute.*

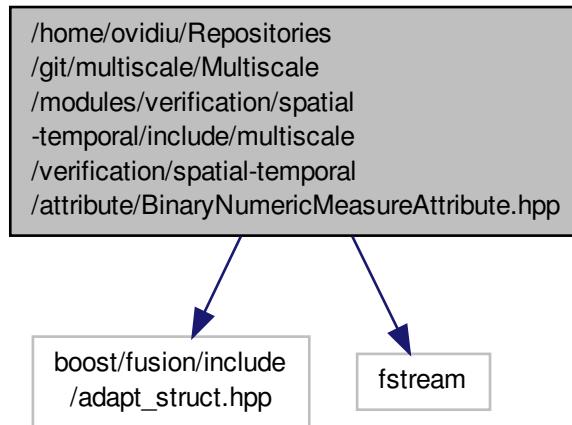
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.102 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for BinaryNumericMeasureAttribute.hpp:



## Classes

- class `multiscale::verification::BinaryNumericMeasureAttribute`  
*Class for representing a binary numeric measure attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## Enumerations

- enum `multiscale::verification::BinaryNumericMeasureType` : unsigned int {  
  `multiscale::verification::BinaryNumericMeasureType::Add` = 0, `multiscale::verification::BinaryNumericMeasureType::Div`, `multiscale::verification::BinaryNumericMeasureType::Log`, `multiscale::verification::BinaryNumericMeasureType::Mod`,  
  `multiscale::verification::BinaryNumericMeasureType::Multiply`, `multiscale::verification::BinaryNumericMeasureType::Power`, `multiscale::verification::BinaryNumericMeasureType::Subtract` }

*Enumeration for representing a binary numeric measure type.*

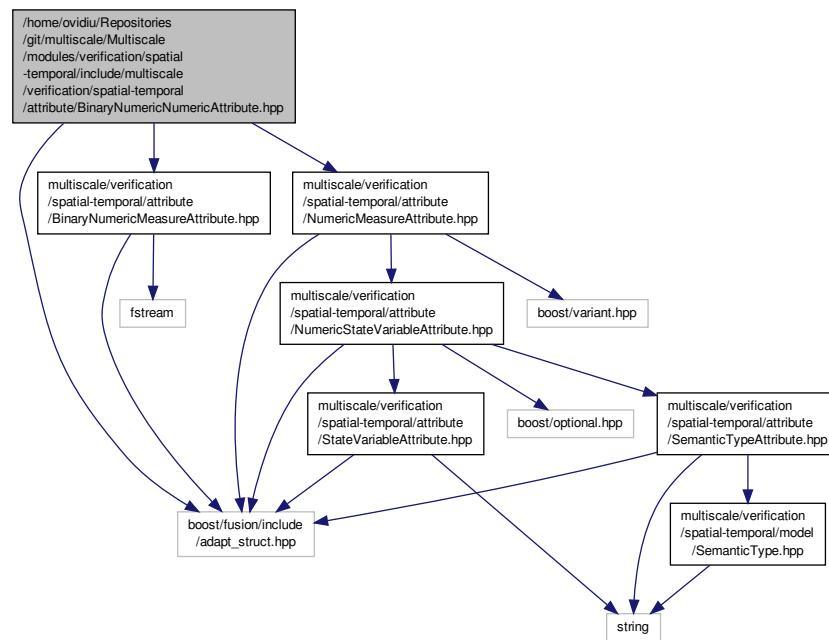
## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const BinaryNumericMeasureType &binaryNumericMeasureType)`

*Overload the output stream operator for the enumeration.*

## 8.103 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for BinaryNumericNumericAttribute.hpp:
```



### Classes

- class [multiscale::verification::BinaryNumericNumericAttribute](#)  
*Class for representing a binary numeric numeric measure attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

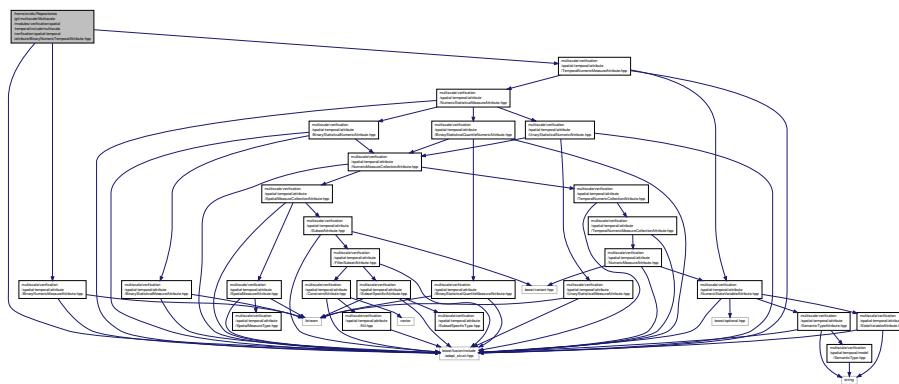
## 8.104 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericTemporalAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericAttribute.hpp"
```

MeasureAttribute.hpp"

```
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for BinaryNumericTemporalAttribute.hpp:



## Classes

- class `multiscale::verification::BinaryNumericTemporalAttribute`

*Class for representing a binary numeric temporal measure attribute.*

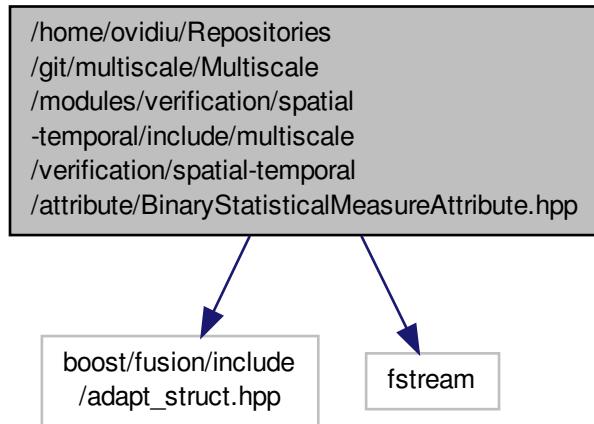
## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.105 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for BinaryStatisticalMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::BinaryStatisticalMeasureAttribute](#)  
*Class for representing a binary statistical measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::BinaryStatisticalMeasureType](#) : unsigned int { [multiscale::verification::BinaryStatisticalMeasureType::Covar](#) = 0 }  
*Enumeration for representing a binary statistical measure type.*

## Functions

- `std::ostream & multiscale::verification::operator<<` (`std::ostream &out, const BinaryStatisticalMeasureType &binaryStatisticalMeasureType`)  
*Overload the output stream operator for the enumeration.*

## 8.106 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatistical→
```

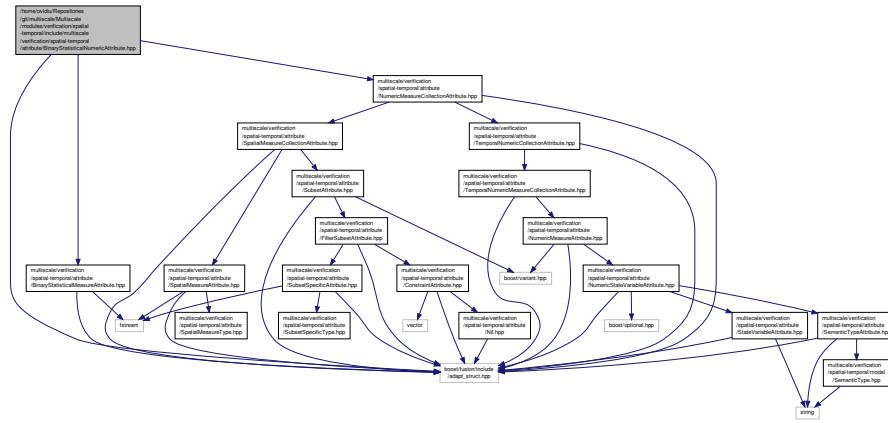
## 8.107 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasure.h

### Attribute.hpp File

[Reference](#)

1157

```
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for BinaryStatisticalNumericAttribute.hpp:
```



## Classes

- class `multiscale::verification::BinaryStatisticalNumericAttribute`

*Class for representing a binary statistical numeric attribute.*

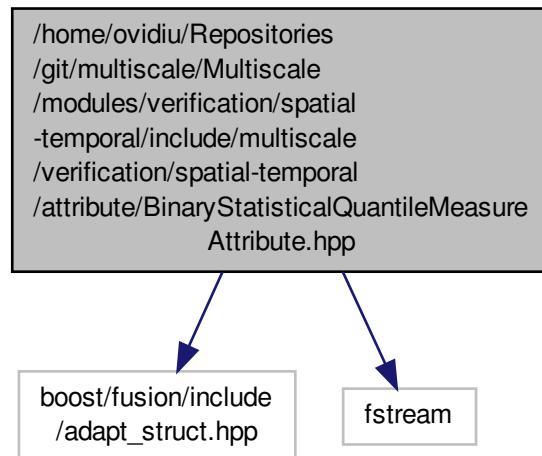
## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.107 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for BinaryStatisticalQuantileMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::BinaryStatisticalQuantileMeasureAttribute](#)  
*Class for representing a binary statistical quantile measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::BinaryStatisticalQuantileMeasureType](#) : unsigned int { [multiscale::verification::BinaryStatisticalQuantileMeasureType::Percentile](#) = 0, [multiscale::verification::BinaryStatisticalQuantileMeasureType::Quartile](#) }

*Enumeration for representing a binary statistical quantile measure type.*

## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const BinaryStatisticalQuantileMeasureType &binaryStatisticalQuantileMeasureType)`  
*Overload the output stream operator for the enumeration.*

8.108 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileNumeric.c

## Attribute.hpp File

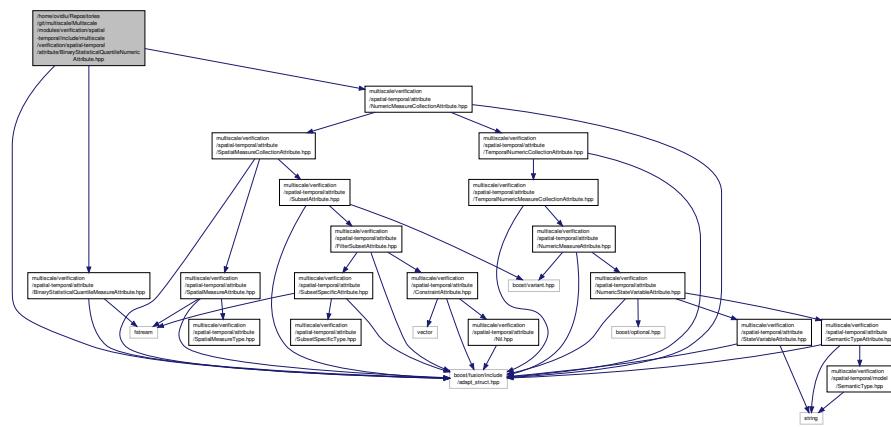
Attribute  
8.108  
Reference

<https://doi.org/10.1016/j.jcp.2019.07.033>

[temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatistical](#)

# QuantileNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for BinaryStatisticalQuantileNumericAttribute.hpp:
```



## Classes

- class multiscale::verification::BinaryStatisticalQuantileNumericAttribute

*Class for representing a binary statistical quantile numeric attribute.*

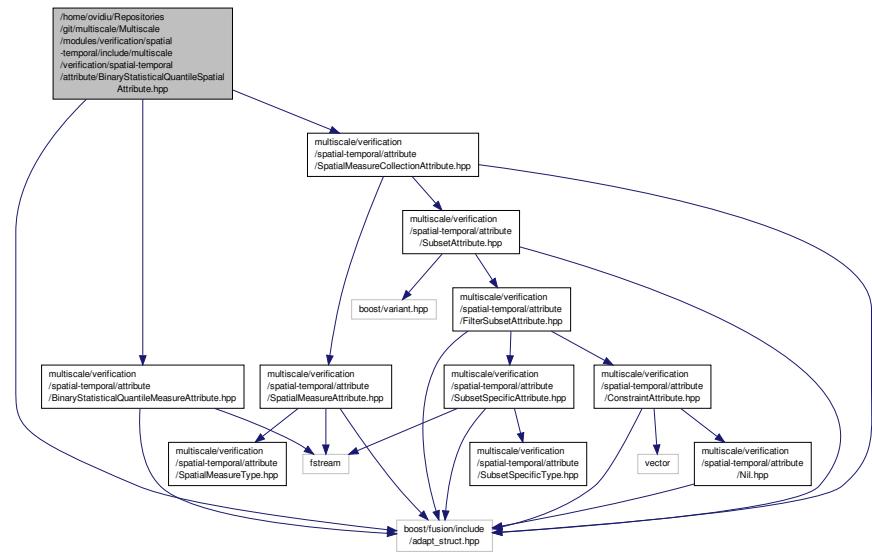
## Namespaces

- multiscale
  - multiscale::verification

8.109 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileSpatialAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for BinaryStatisticalQuantileSpatialAttribute.hpp:



## Classes

- class [multiscale::verification::BinaryStatisticalQuantileSpatialAttribute](#)

*Class for representing a binary statistical quantile spatial attribute.*

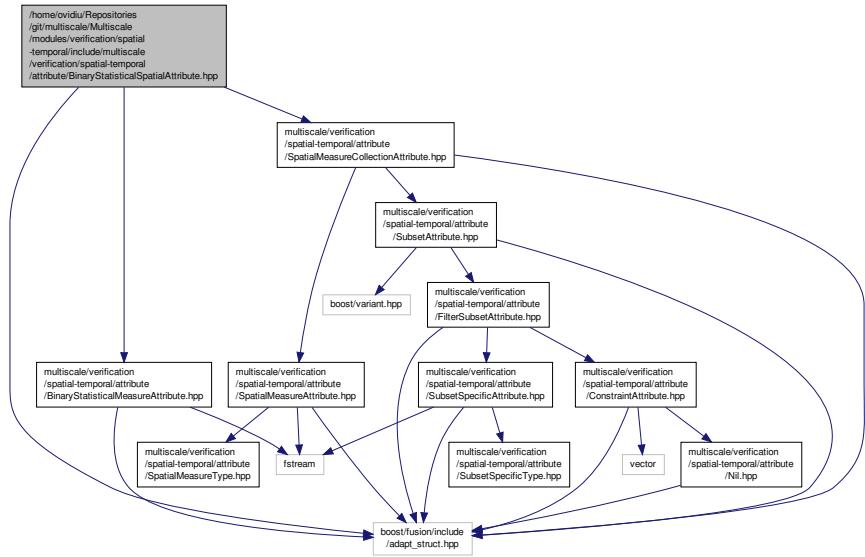
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.110 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryStatisticalSpatialAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for BinaryStatisticalSpatialAttribute.hpp:



## Classes

- class `multiscale::verification::BinaryStatisticalSpatialAttribute`

*Class for representing a binary statistical spatial attribute.*

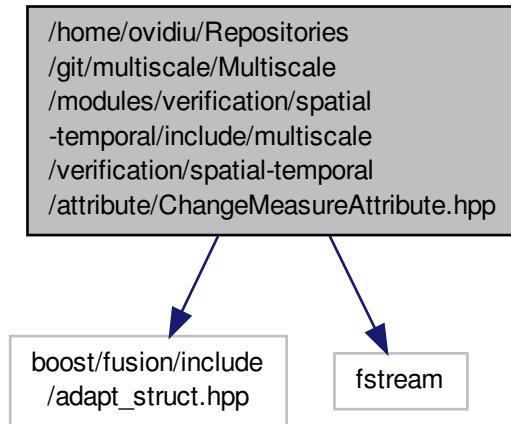
## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.111 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for ChangeMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::ChangeMeasureAttribute](#)  
*Class for representing a change measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::ChangeMeasureType](#) : unsigned int { [multiscale::verification::ChangeMeasureType::Derivative](#) = 0, [multiscale::verification::ChangeMeasureType::Ratio](#) }  
*Enumeration for representing a change measure type.*

## Functions

- [std::ostream & multiscale::verification::operator<<](#) (std::ostream &out, const ChangeMeasureType &changeMeasureType)  
*Overload the output stream operator for the enumeration.*

## 8.112 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericCollectionAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp"
```

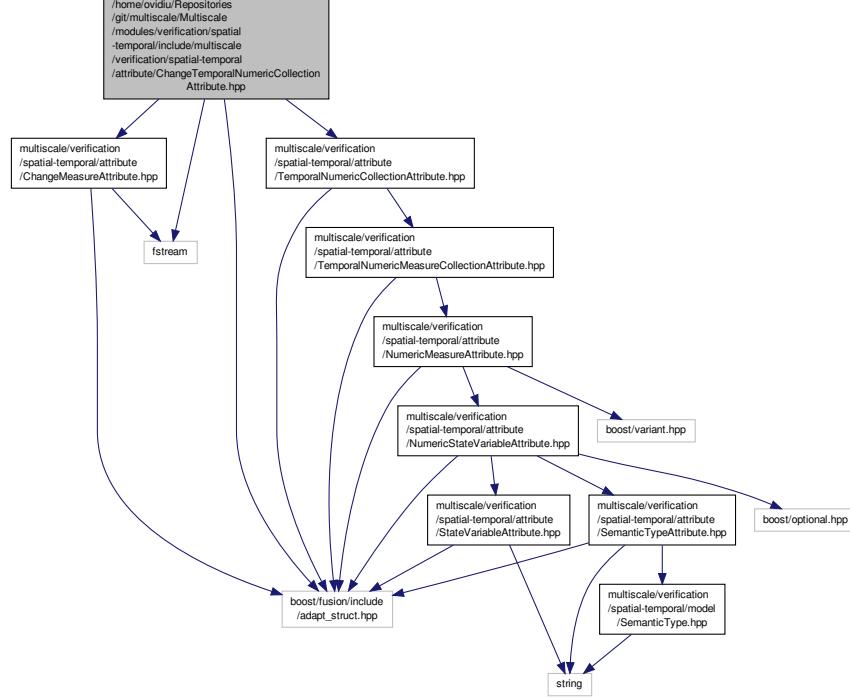
8.113 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericMeasure.h  
Attribute.h File

Attribute.hpp

1163

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for ChangeTemporalNumericCollectionAttribute.hpp:



## Classes

- class multiscale::verification::ChangeTemporalNumericCollectionAttribute

*Class for representing a change temporal numeric collection attribute.*

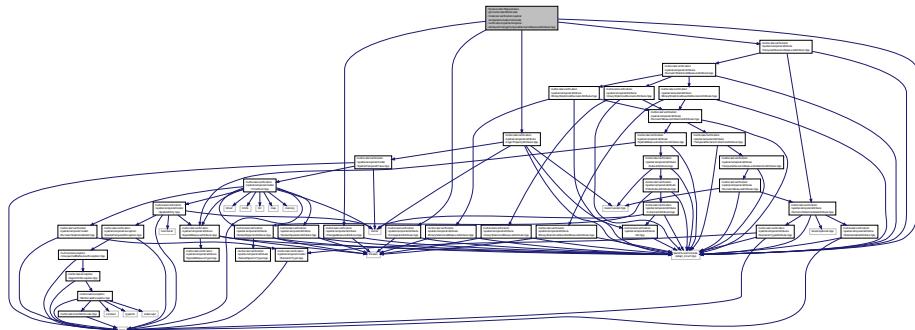
## Namespaces

- multiscale
- multiscale::verification

8.113 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/LogicProperty.hpp"
```

```
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for ChangeTemporalNumericMeasureAttribute.hpp:
```



## Classes

- class [multiscale::verification::ChangeTemporalNumericMeasureAttribute](#)

*Class for representing a change temporal numeric measure attribute.*

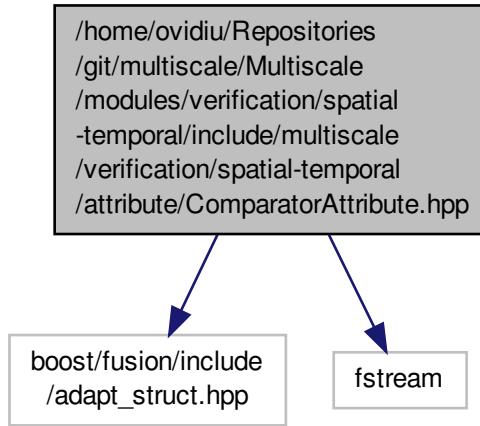
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.114 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for ComparatorAttribute.hpp:



## Classes

- class [multiscale::verification::ComparatorAttribute](#)

*Class for representing a comparator attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::ComparatorType](#) : unsigned int {  
    multiscale::verification::ComparatorType::GreaterThan = 0,   multiscale::verification::ComparatorType::GreaterThanOrEqual,   multiscale::verification::ComparatorType::LessThan,   multiscale::verification::ComparatorType::LessThanOrEqual,  
    multiscale::verification::ComparatorType::Equal }

*Enumeration for representing a comparator type.*

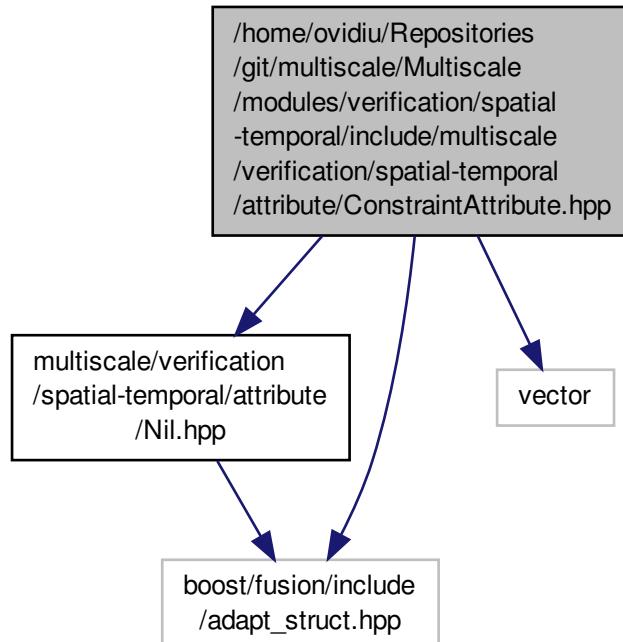
## Functions

- [std::ostream & multiscale::verification::operator<< \(std::ostream &out, const ComparatorType &comparatorType\)](#)

*Overload the output stream operator for the enumeration.*

## 8.115 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/Nil.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <vector>
Include dependency graph for ConstraintAttribute.hpp:
```



### Classes

- class [multiscale::verification::ConstraintAttribute](#)

*Class for representing a constraint attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

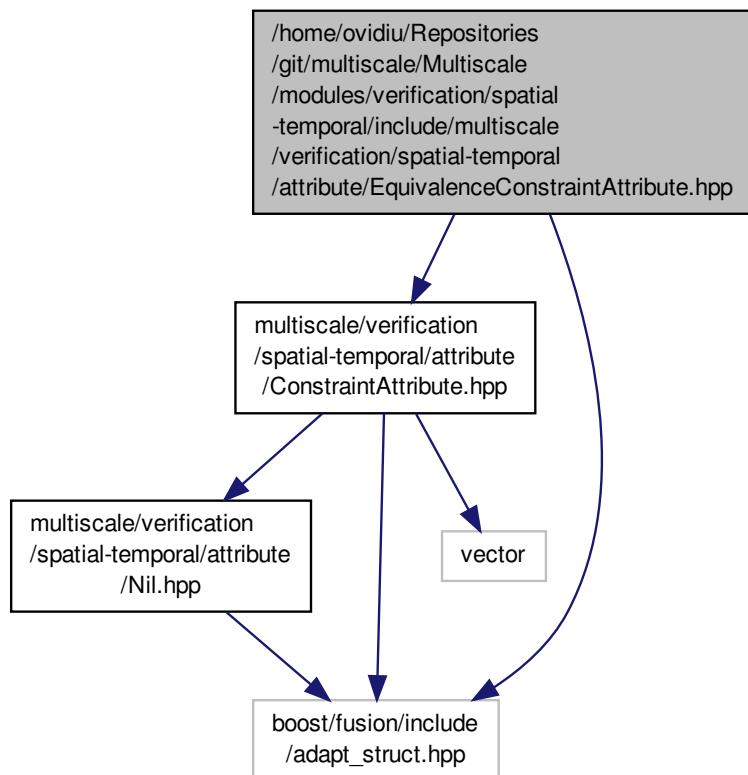
- `typedef boost::variant< Nil,  
boost::recursive_wrapper  
< ConstraintAttribute >  
, boost::recursive_wrapper  
< OrConstraintAttribute >  
, boost::recursive_wrapper  
< AndConstraintAttribute >  
, boost::recursive_wrapper  
< ImplicationConstraintAttribute >  
, boost::recursive_wrapper  
< EquivalenceConstraintAttribute >  
, boost::recursive_wrapper  
< PrimaryConstraintAttribute > > multiscale::verification::ConstraintAttributeType`

*Variant for a constraint attribute type.*

## 8.116 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/EquivalenceConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for EquivalenceConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::EquivalenceConstraintAttribute](#)  
*Class for representing an "equivalence" constraint attribute.*

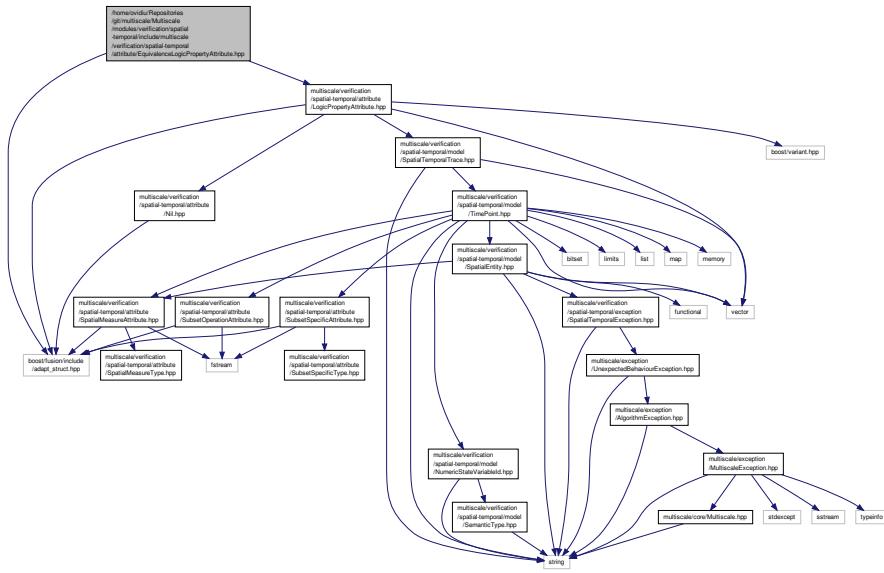
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.117 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/EquivalenceLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for EquivalenceLogicPropertyAttribute.hpp:



## Classes

- class multiscale::verification::EquivalenceLogicPropertyAttribute

*Class for representing an "equivalence" logic property attribute.*

## Namespaces

- multiscale
- multiscale::verification

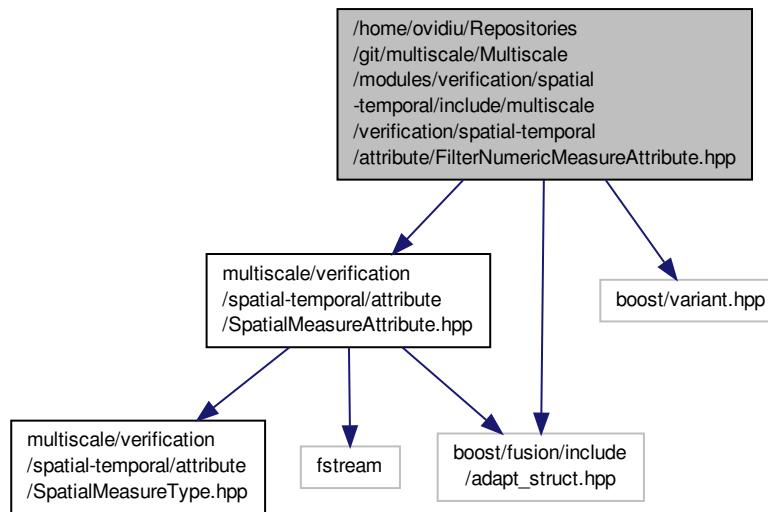
## 8.118 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FilterNumericMeasureAttribute.hpp File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>

```

Include dependency graph for FilterNumericMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::FilterNumericMeasureAttribute](#)

*Class for representing a filter numeric measure.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

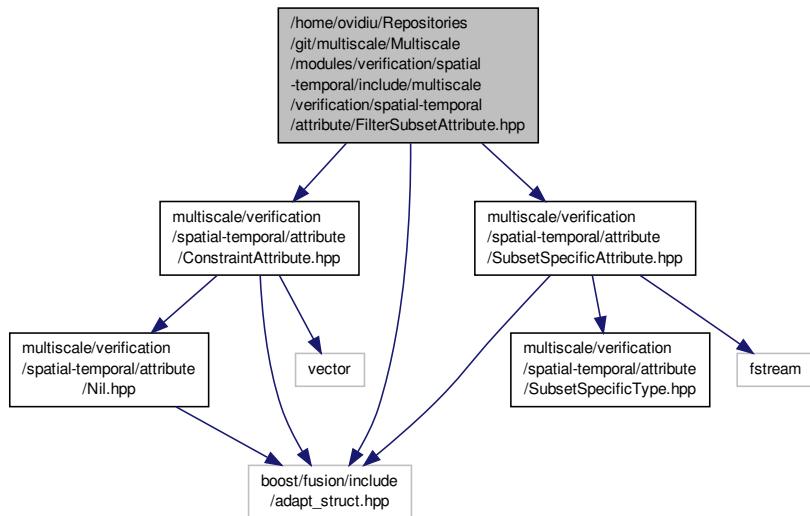
## TypeDefs

- `typedef boost::variant< SpatialMeasureAttribute, boost::recursive_wrapper< PrimaryNumericMeasureAttribute >, boost::recursive_wrapper< UnaryNumericFilterAttribute >, boost::recursive_wrapper< BinaryNumericFilterAttribute >, boost::recursive_wrapper< FilterNumericMeasureAttribute > > multiscale::verification::FilterNumericMeasureAttributeType`

*Variant for a filter numeric measure attribute.*

8.119 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FilterSubsetAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for FilterSubsetAttribute.hpp:
```



## Classes

- class [multiscale::verification::FilterSubsetAttribute](#)

*Class for representing a filter subset attribute.*

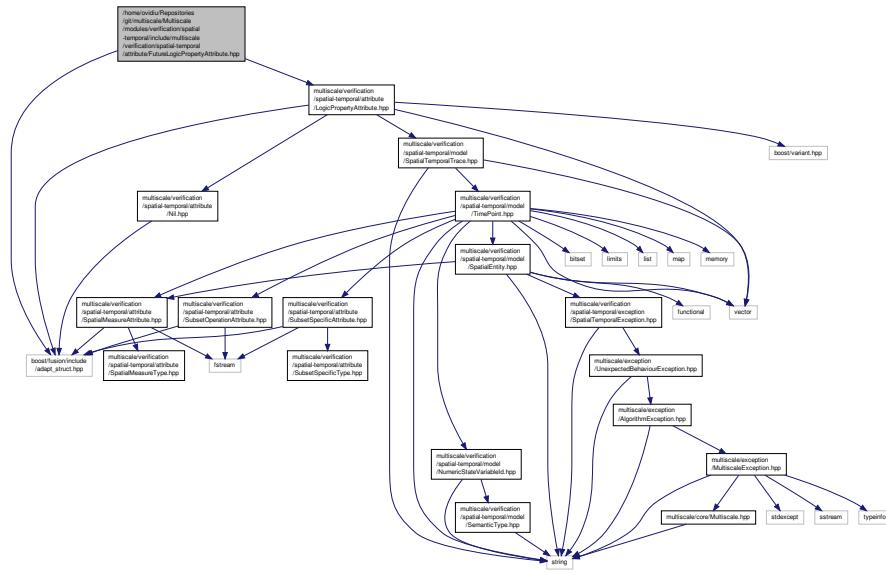
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

8.120 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/FutureLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for FutureLogicPropertyAttribute.hpp:



## Classes

- class multiscale::verification::FutureLogicPropertyAttribute

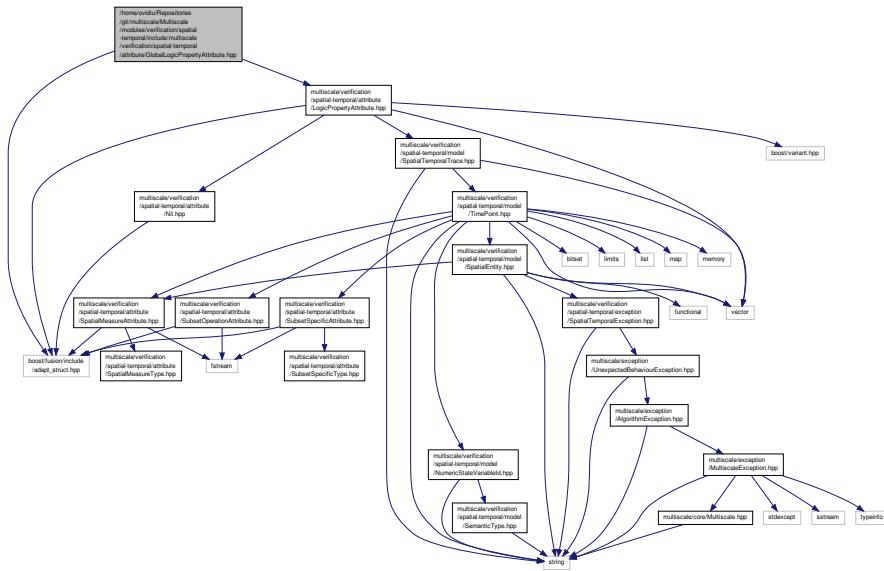
*Class for representing a "future" logic property attribute.*

## Namespaces

- multiscale
  - multiscale::verification

8.121 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/GlobalLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```



## Classes

- class multiscale::verification::GlobalLogicPropertyAttribute

*Class for representing a "globally" logic property attribute.*

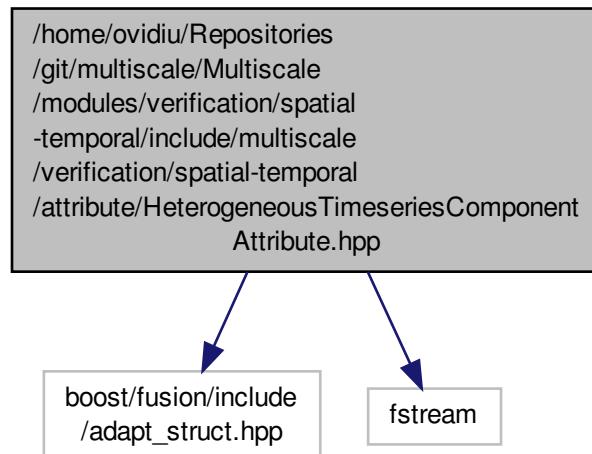
## Namespaces

- multiscale
  - multiscale::verification

8.122 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HeterogeneousTimeseriesComponentAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <iostream>
```

Include dependency graph for HeterogeneousTimeseriesComponentAttribute.hpp:



## Classes

- class [multiscale::verification::HeterogeneousTimeseriesComponentAttribute](#)  
*Class for representing a heterogeneous timeseries component attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::HeterogeneousTimeseriesComponentType](#) : unsigned int { [multiscale::verification::HeterogeneousTimeseriesComponentType::Peak](#) = 0, [multiscale::verification::HeterogeneousTimeseriesComponentType::Valley](#) }

*Enumeration for representing a heterogeneous timeseries component type.*

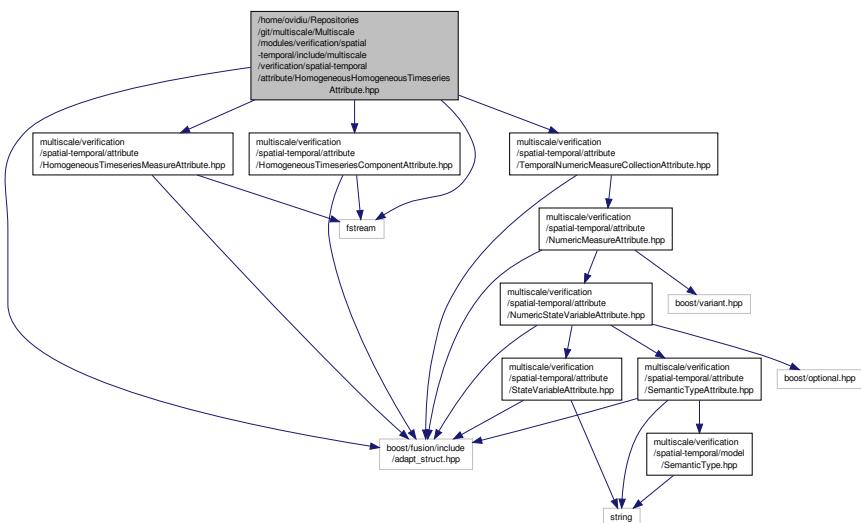
## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const HeterogeneousTimeseriesComponentType &heterogeneousTimeseriesComponentType)`  
*Overload the output stream operator for the enumeration.*

8.123 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousHomogeneousTimeseriesAttribute.hpp File Reference  
8.123 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousHomogeneousTimeseriesAttribute.hpp File Reference [175]

```
#include "multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for HomogeneousHomogeneousTimeseriesAttribute.hpp:



## Classes

- class [multiscale::verification::HomogeneousHomogeneousTimeseriesAttribute](#)  
*Class for representing a homogeneous homogeneous timeseries attribute.*

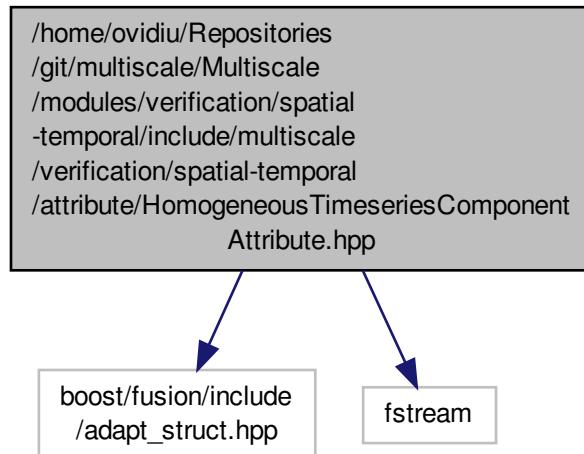
## Namespaces

- multiscale
- multiscale::verification

8.124 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for HomogeneousTimeseriesComponentAttribute.hpp:



## Classes

- class `multiscale::verification::HomogeneousTimeseriesComponentAttribute`

*Class for representing a homogeneous timeseries component attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## Enumerations

- enum `multiscale::verification::HomogeneousTimeseriesComponentType` : unsigned int {
 `multiscale::verification::HomogeneousTimeseriesComponentType::Ascent = 0,`
`multiscale::verification::HomogeneousTimeseriesComponentType::Descent,`
`multiscale::verification::HomogeneousTimeseriesComponentType::Plateau,`
`multiscale::verification::HomogeneousTimeseriesComponentType::UniformAscent,`
`multiscale::verification::HomogeneousTimeseriesComponentType::UniformDescent }`

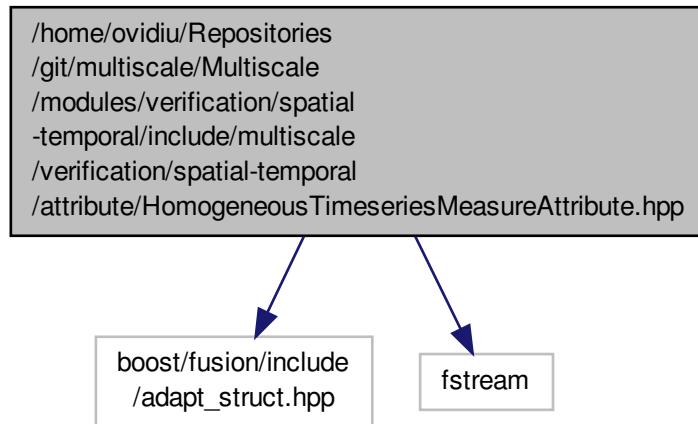
*Enumeration for representing a homogeneous timeseries component type.*

## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const HomogeneousTimeseriesComponentType &homogeneousTimeseriesComponentType)`

*Overload the output stream operator for the enumeration.*

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
Include dependency graph for HomogeneousTimeseriesMeasureAttribute.hpp:
```



## Classes

- class `multiscale::verification::HomogeneousTimeseriesMeasureAttribute`  
*Class for representing a homogeneous timeseries measure attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## Enumerations

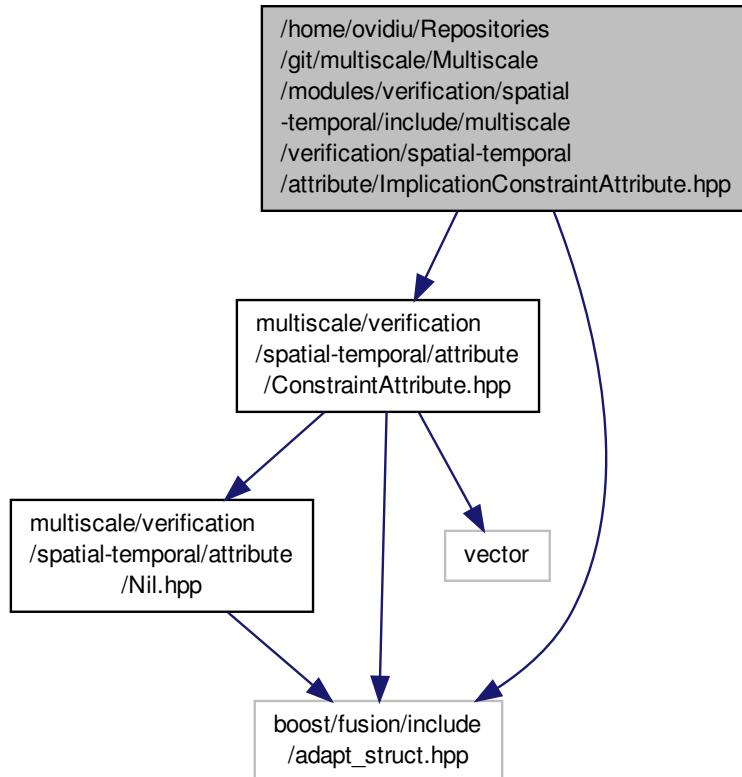
- enum `multiscale::verification::HomogeneousTimeseriesMeasureType` : `unsigned int` { `multiscale::verification::HomogeneousTimeseriesMeasureType::TimeSpan = 0`, `multiscale::verification::HomogeneousTimeseriesMeasureType::Values` }  
*Enumeration for representing a homogeneous timeseries measure type.*

## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const HomogeneousTimeseriesMeasureType &homogeneousTimeseriesMeasureType)`  
*Overload the output stream operator for the enumeration.*

## 8.126 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for ImplicationConstraintAttribute.hpp:
```



### Classes

- class [multiscale::verification::ImplicationConstraintAttribute](#)

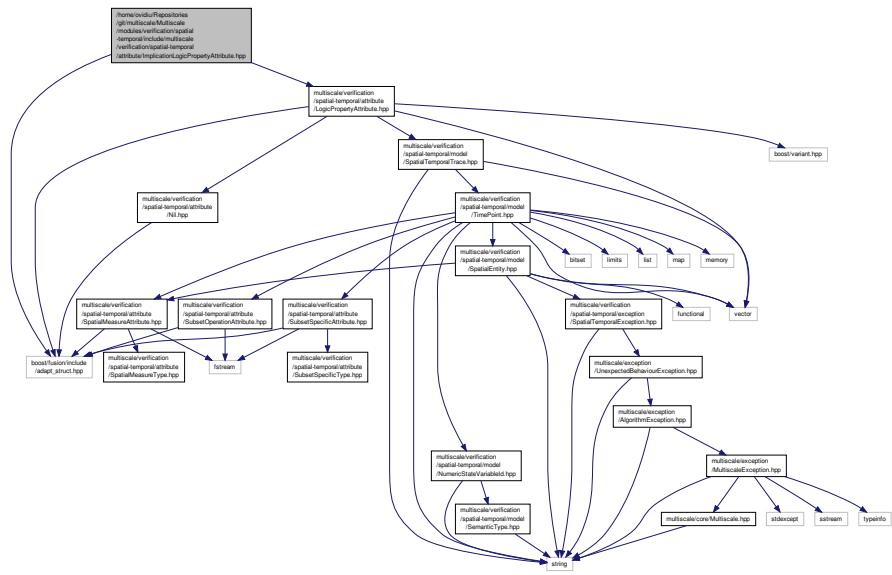
*Class for representing an "implication" constraint attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

**8.127** /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationLogicPropertyAttribute.hpp  
**File Reference** 1179  
8.127 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ImplicationLogicPropertyAttribute.hpp File Reference ↵

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for ImplicationLogicPropertyAttribute.hpp:
```



## Classes

- class [multiscale::verification::ImplicationLogicPropertyAttribute](#)  
*Class for representing an "implication" logic property attribute.*

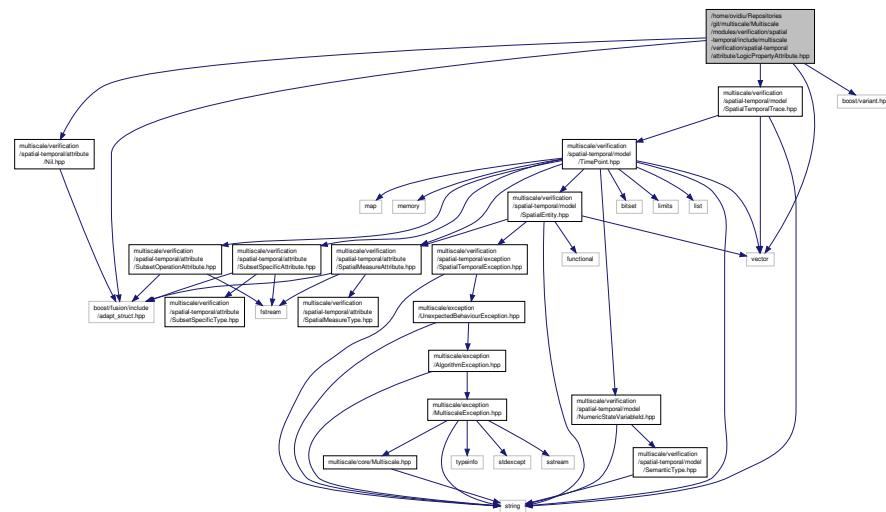
## Namespaces

- multiscale
  - multiscale::verification

8.128 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/Nil.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
#include <vector>
```

Include dependency graph for LogicPropertyAttribute.hpp:



## Classes

- class [multiscale::verification::LogicPropertyAttribute](#)

*Class for representing a logic property attribute.*

## Namespaces

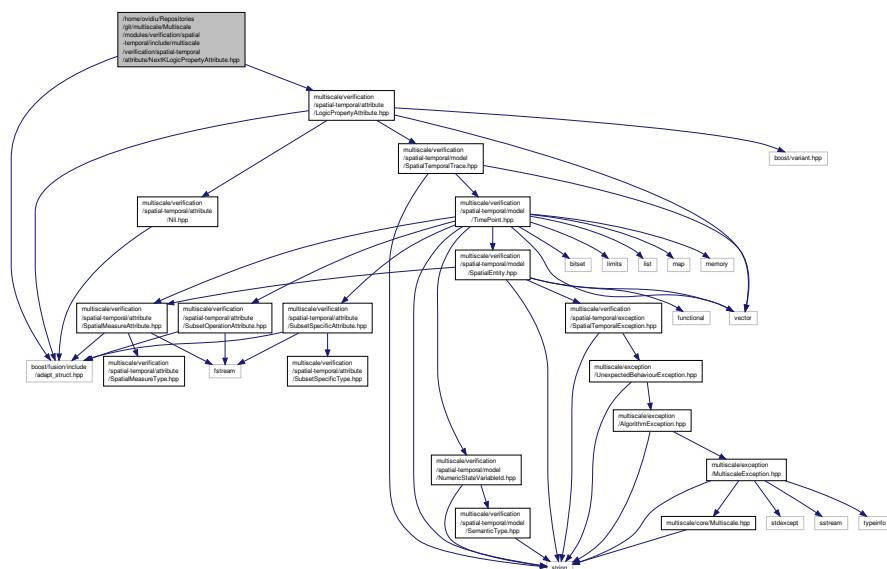
- [multiscale](#)
- [multiscale::verification](#)

## Typedefs

- [typedef boost::variant< Nil,  
boost::recursive\\_wrapper  
< LogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< OrLogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< AndLogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< ImplicationLogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< EquivalenceLogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< UntilLogicPropertyAttribute >  
, boost::recursive\\_wrapper  
< PrimaryLogicPropertyAttribute > > multiscale::verification::LogicPropertyAttributeType](#)

*Variant for the logic property attribute.*

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for NextKLogicPropertyAttribute.hpp:
```



## Classes

- class multiscale::verification::NextKLogicPropertyAttribute

*Class for representing a "next K" logic property attribute.*

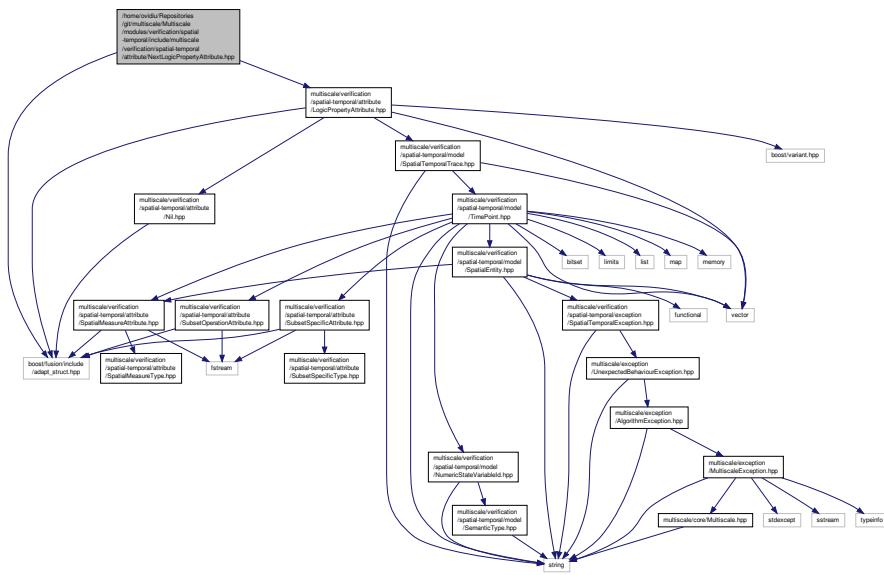
## Namespaces

- multiscale
  - multiscale::verification

8.130 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for NextLogicPropertyAttribute.hpp:



## Classes

- class multiscale::verification::NextLogicPropertyAttribute

*Class for representing a "next" logic property attribute.*

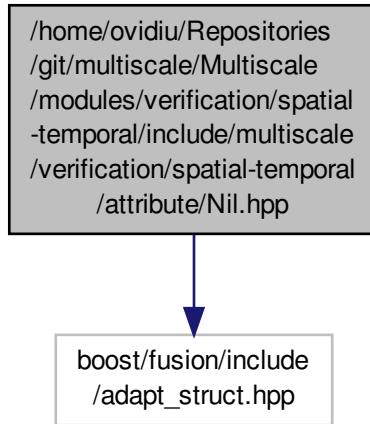
## Namespaces

- multiscale
- multiscale::verification

## 8.131 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Nil.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for Nil.hpp:



## Classes

- class [multiscale::verification::Nil](#)

*A class used to avoid run-time errors when defining a variant type.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Functions

- [BOOST\\_FUSION\\_ADAPT\\_STRUCT \(multiscale::verification::Nil,\)](#)

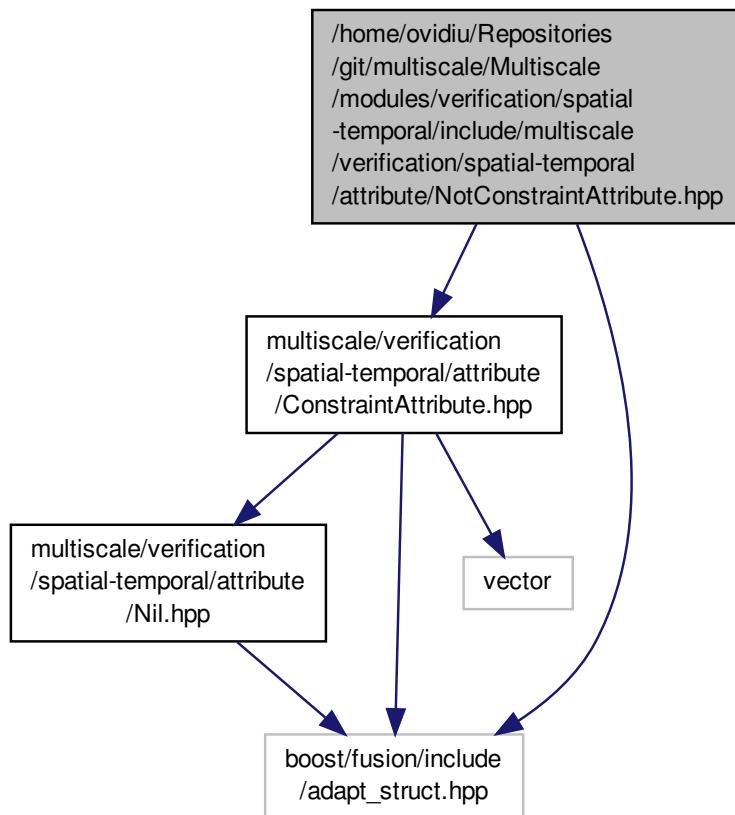
### 8.131.1 Function Documentation

#### 8.131.1.1 [BOOST\\_FUSION\\_ADAPT\\_STRUCT \( multiscale::verification::Nil \)](#)

## 8.132 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for NotConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::NotConstraintAttribute](#)

*Class for representing a "not" constraint attribute.*

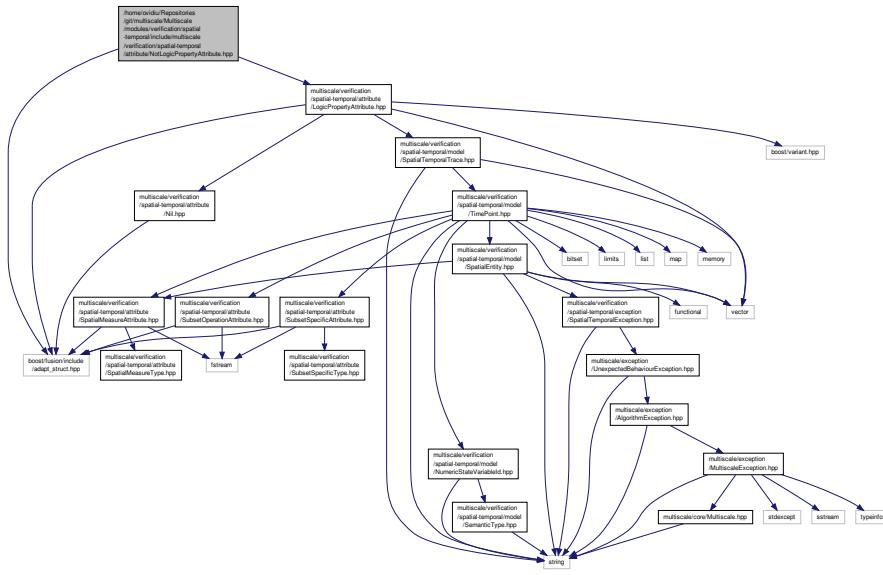
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.133 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NotLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for NotLogicPropertyAttribute.hpp:



## Classes

- class [multiscale::verification::NotLogicPropertyAttribute](#)

*Class for representing a "not" logic property attribute.*

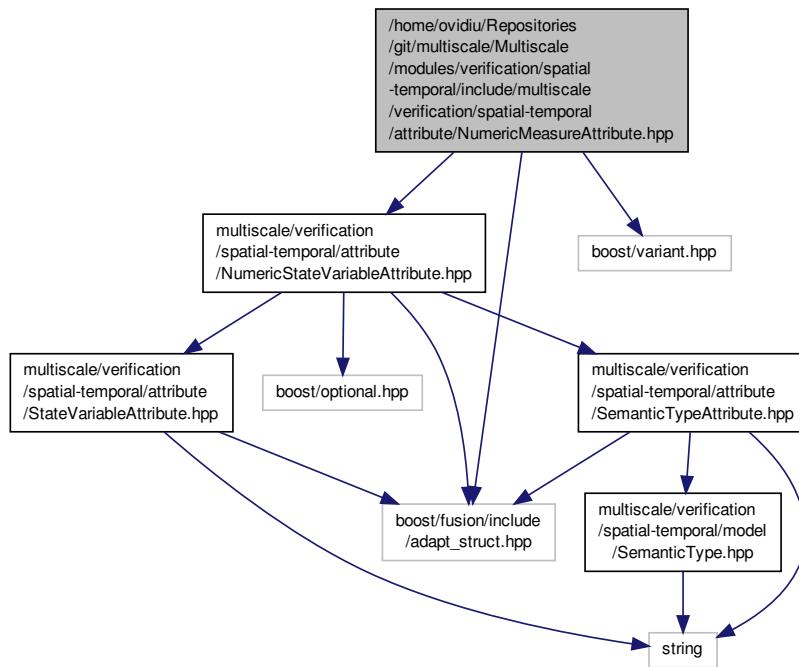
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.134 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
```

Include dependency graph for NumericMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::NumericMeasureAttribute](#)  
*Class for representing a numeric measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

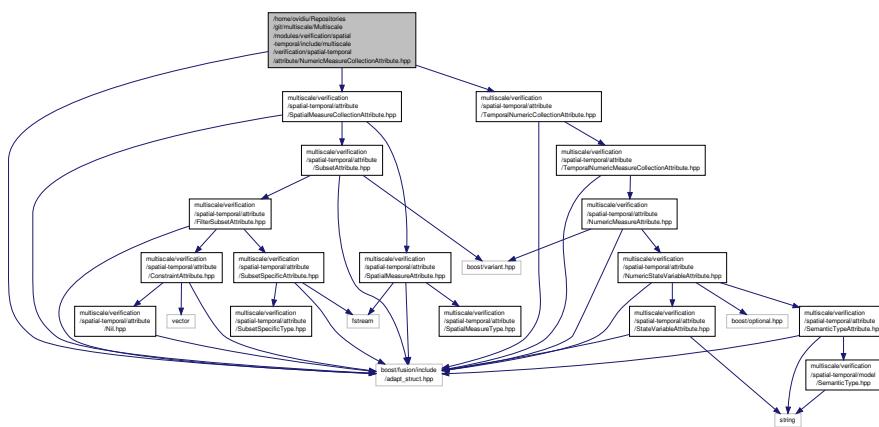
## TypeDefs

- `typedef boost::variant< double,  
 NumericStateVariableAttribute,  
 boost::recursive_wrapper  
 < NumericSpatialMeasureAttribute >  
 , boost::recursive_wrapper  
 < PrimaryNumericMeasureAttribute >  
 , boost::recursive_wrapper  
 < UnaryNumericNumericAttribute >  
 , boost::recursive_wrapper  
 < BinaryNumericNumericAttribute >  
 , boost::recursive_wrapper  
 < NumericMeasureAttribute > > multiscale::verification::NumericMeasureType`

*Variant for the numeric measure attribute.*

8.135 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp  
**File Reference** 1187  
**8.135** /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for NumericMeasureCollectionAttribute.hpp:
```



## Classes

- class [multiscale::verification::NumericMeasureCollectionAttribute](#)  
*Class for representing a numeric measure collection attribute.*

## Namespaces

- multiscale
- multiscale::verification

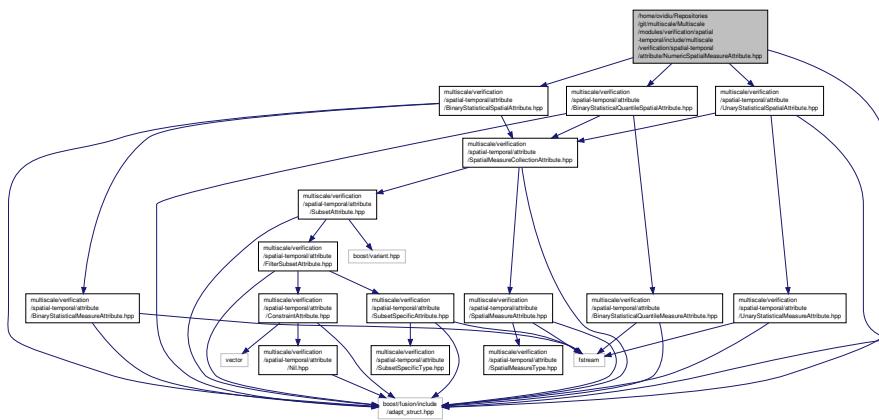
## Typedefs

- typedef boost::variant< SpatialMeasureCollectionAttribute, TemporalNumericCollectionAttribute > [multiscale::verification::NumericMeasureCollectionType](#)  
*Variant for the numeric measure collection attribute.*

8.136 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericSpatialMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileSpatialAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatistical
```

```
SpatialAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for NumericSpatialMeasureAttribute.hpp:
```



## Classes

- class [multiscale::verification::NumericSpatialMeasureAttribute](#)  
*Class for representing a numeric spatial measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

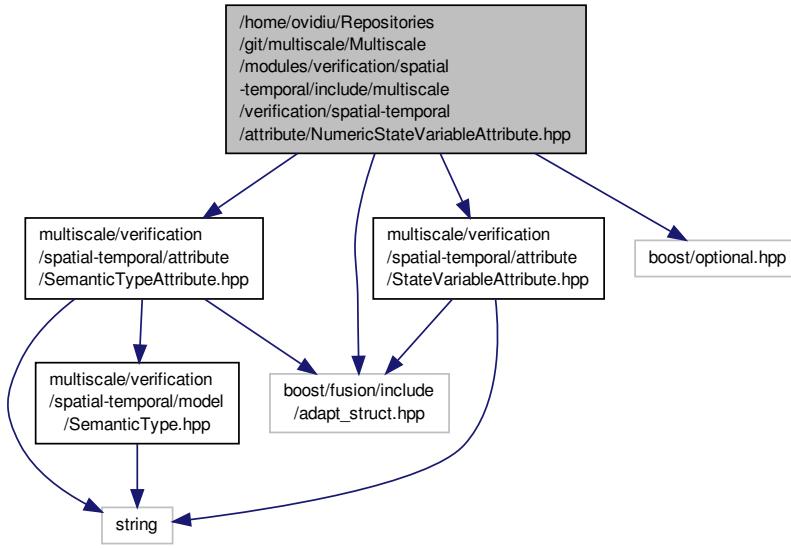
## Typedefs

- typedef `boost::variant<UnaryStatisticalSpatialAttribute, BinaryStatisticalSpatialAttribute, BinaryStatisticalQuantileSpatialAttribute, boost::recursive_wrapper<NumericSpatialMeasureAttribute>>` [multiscale::verification::NumericSpatialMeasureType](#)  
*Variant for a numeric spatial measure attribute.*

## 8.137 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/StateVariableAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/optional.hpp>
```

Include dependency graph for NumericStateVariableAttribute.hpp:



## Classes

- class [multiscale::verification::NumericStateVariableAttribute](#)

*Class for representing a numeric state variable attribute.*

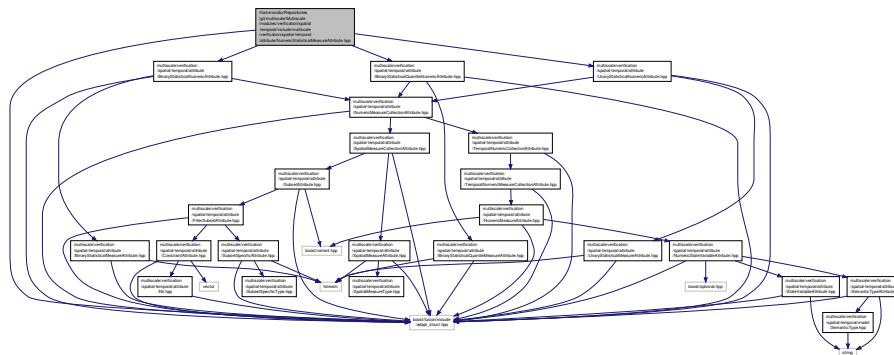
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.138 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NumericStatisticalMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalNumericAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileNumericAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalNumericAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for NumericStatisticalMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::NumericStatisticalMeasureAttribute](#)

*Class for representing a numeric statistical measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Typedefs

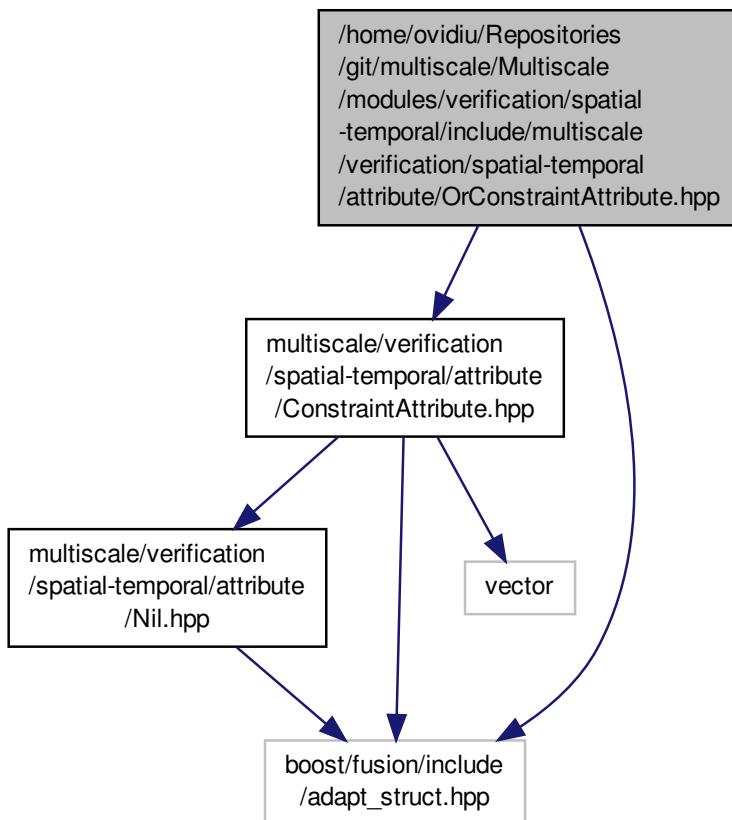
- typedef boost::variant<UnaryStatisticalNumericAttribute, BinaryStatisticalNumericAttribute, BinaryStatisticalQuantileNumericAttribute> [multiscale::verification::NumericStatisticalMeasureType](#)

*Variant for the numeric statistical measure attribute.*

## 8.139 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ConstraintAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for OrConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::OrConstraintAttribute](#)  
*Class for representing an "or" constraint attribute.*

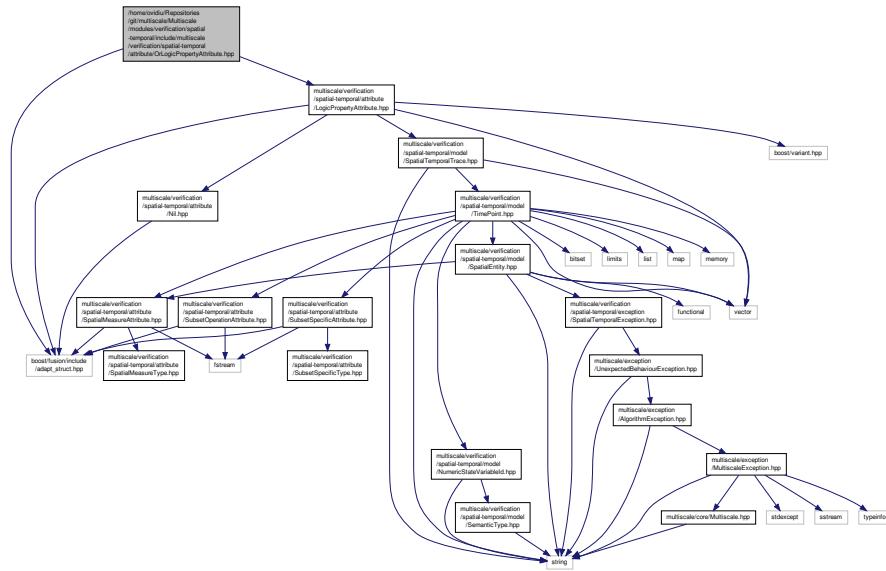
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.140 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/OrLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for OrLogicPropertyAttribute.hpp:



## Classes

- class `multiscale::verification::OrLogicPropertyAttribute`

*Class for representing an "or" logic property attribute.*

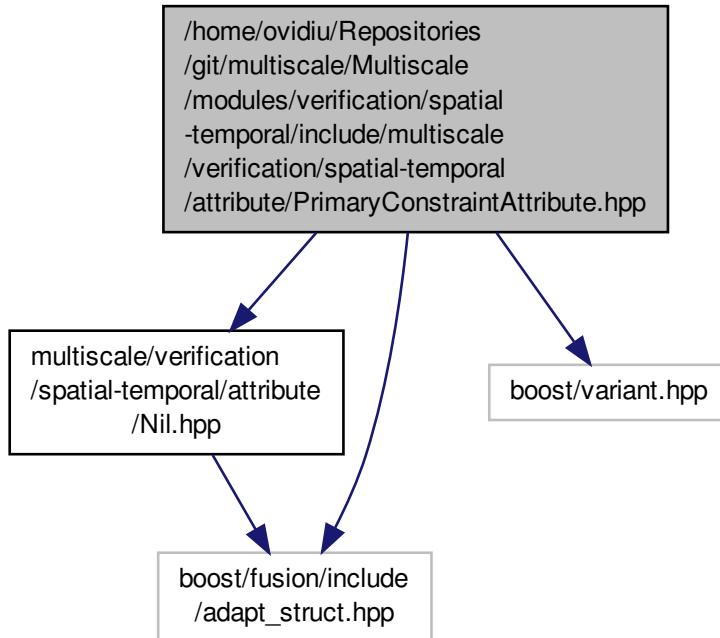
## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.141 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/Nil.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
```

Include dependency graph for PrimaryConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::PrimaryConstraintAttribute](#)

*Class for representing a primary constraint attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## TypeDefs

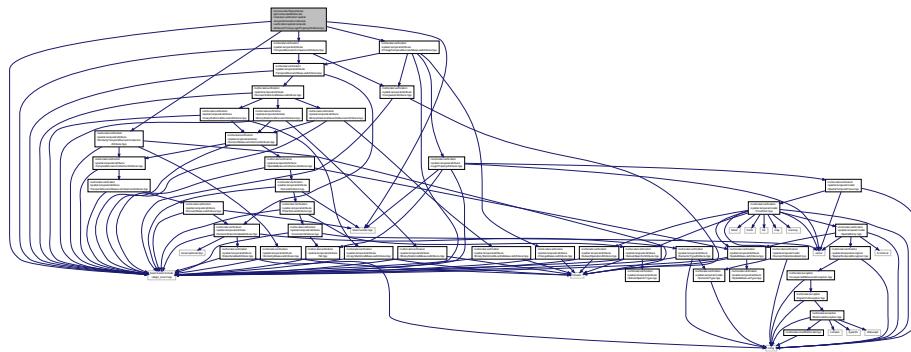
- typedef boost::variant< Nil,  
boost::recursive\_wrapper  
< ConstraintAttribute >  
, boost::recursive\_wrapper  
< NotConstraintAttribute >  
, boost::recursive\_wrapper  
< UnarySpatialConstraintAttribute >  
, boost::recursive\_wrapper  
< UnaryTypeConstraintAttribute > > [multiscale::verification::PrimaryConstraintAttributeType](#)

*Variant for a primary constraint attribute.*

## 8.142 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ChangeTemporalNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SimilarityTemporalNumericCollectionAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericComparisonAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
```

Include dependency graph for PrimaryLogicPropertyAttribute.hpp:



### Classes

- class [multiscale::verification::PrimaryLogicPropertyAttribute](#)  
*Class for representing a primary logic property attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### Typedefs

- [typedef boost::variant<TemporalNumericComparisonAttribute, ChangeTemporalNumericMeasureAttribute, SimilarityTemporalNumericCollectionAttribute, boost::recursive\\_wrapper<NotLogicPropertyAttribute>, boost::recursive\\_wrapper<FutureLogicPropertyAttribute>, boost::recursive\\_wrapper<GlobalLogicPropertyAttribute>, boost::recursive\\_wrapper<NextLogicPropertyAttribute>, boost::recursive\\_wrapper<NextKLogicPropertyAttribute>, boost::recursive\\_wrapper](#)

< LogicPropertyAttribute >> multiscale::verification::PrimaryLogicPropertyAttributeType

*Variant for representing a primary logic property type.*

## Functions

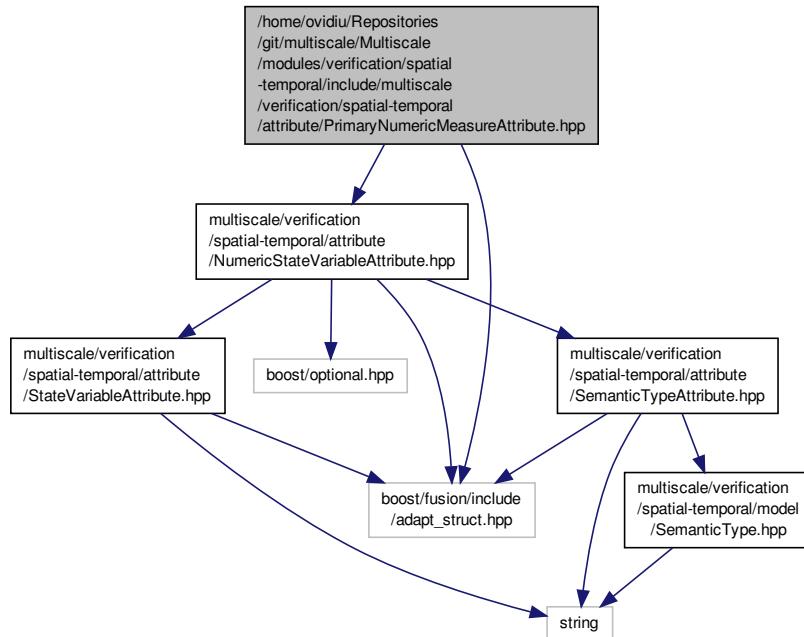
- `BOOST_FUSION_ADAPT_STRUCT` (`multiscale::verification::PrimaryLogicPropertyAttribute, (multiscale::verification::PrimaryLogicPropertyAttributeType, primaryLogicProperty)`)

### 8.142.1 Function Documentation

8.142.1.1 `BOOST_FUSION_ADAPT_STRUCT ( multiscale::verification::PrimaryLogicPropertyAttribute , (multiscale::verification::PrimaryLogicPropertyAttributeType, primaryLogicProperty) )`

## 8.143 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryNumericMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for PrimaryNumericMeasureAttribute.hpp:
```



## Classes

- class `multiscale::verification::PrimaryNumericMeasureAttribute`

*Class for representing a primary numeric measure attribute.*

## Namespaces

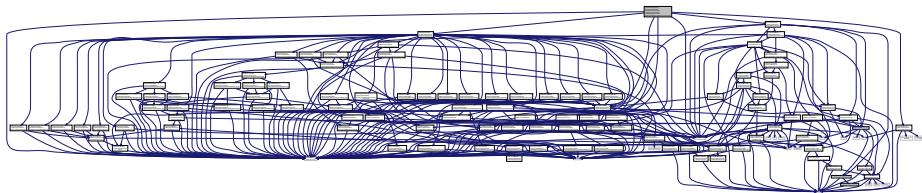
- [multiscale](#)
- [multiscale::verification](#)

## Typedefs

- [typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive\\_wrapper< NumericSpatialMeasureAttribute >, boost::recursive\\_wrapper< PrimaryNumericMeasureAttribute > > multiscale::verification::PrimaryNumericMeasureAttributeType](#)  
*Variant for the primary numeric measure attribute.*

## 8.144 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp"
#include "multiscale/verification/spatial-temporal/visitor/LogicPropertyVisitor.hpp"
#include "boost/fusion/include/adapt_struct.hpp"
#include <boost/variant.hpp>
Include dependency graph for ProbabilisticLogicPropertyAttribute.hpp:
```



## Classes

- class [multiscale::verification::ProbabilisticLogicPropertyAttribute](#)  
*Class for representing a probabilistic logic property attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

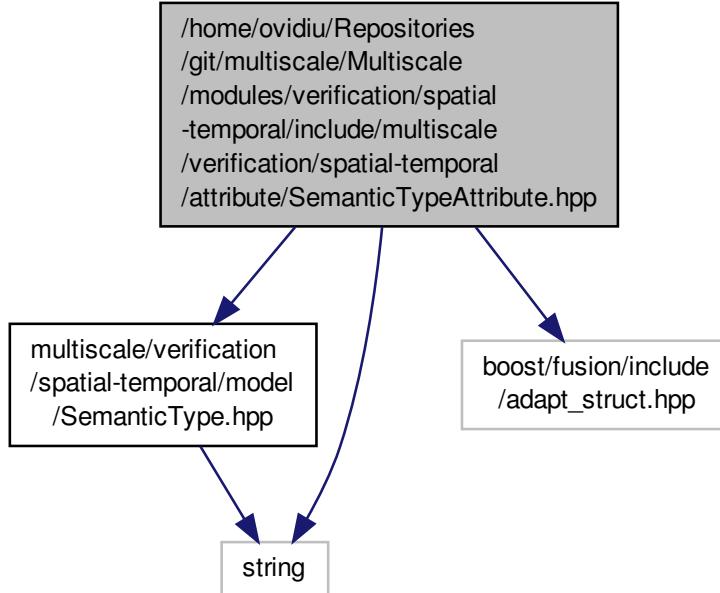
- BOOST\_FUSION\_ADAPT\_STRUCT ([multiscale::verification::ProbabilisticLogicPropertyAttribute](#),  
[\(multiscale::verification::ComparatorAttribute, comparator\)\(double, probability\)](#)([multiscale::verification::LogicPropertyAttributeType](#), logicProperty))

#### 8.144.1 Function Documentation

8.144.1.1 BOOST\_FUSION\_ADAPT\_STRUCT ( [multiscale::verification::ProbabilisticLogicPropertyAttribute](#) , [\(multiscale::verification::ComparatorAttribute, comparator\)\(double, probability\)](#)([multiscale::verification::LogicPropertyAttributeType](#), logicProperty) )

### 8.145 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <string>
Include dependency graph for SemanticTypeAttribute.hpp:
```



#### Classes

- class [multiscale::verification::SemanticTypeAttribute](#)

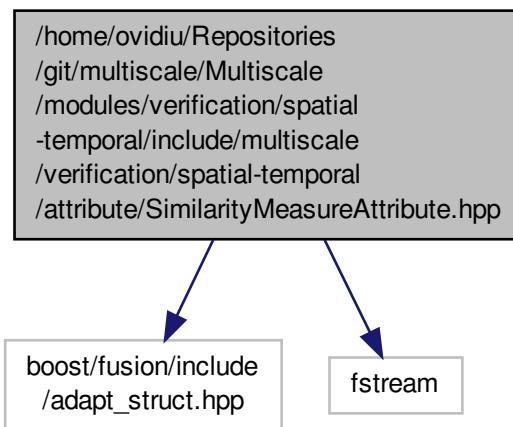
*Class for representing a semantic type attribute.*

## Namespaces

- multiscale
- multiscale::verification

## 8.146 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
Include dependency graph for SimilarityMeasureAttribute.hpp:
```



## Classes

- class [multiscale::verification::SimilarityMeasureAttribute](#)  
*Class for representing a similarity measure attribute.*

## Namespaces

- multiscale
- multiscale::verification

## Enumerations

- enum [multiscale::verification::SimilarityMeasureType](#) : unsigned int { [multiscale::verification::SimilarityMeasureType::Opposite](#) = 0, [multiscale::verification::SimilarityMeasureType::Similar](#) }
- Enumeration for representing a similarity measure type.*

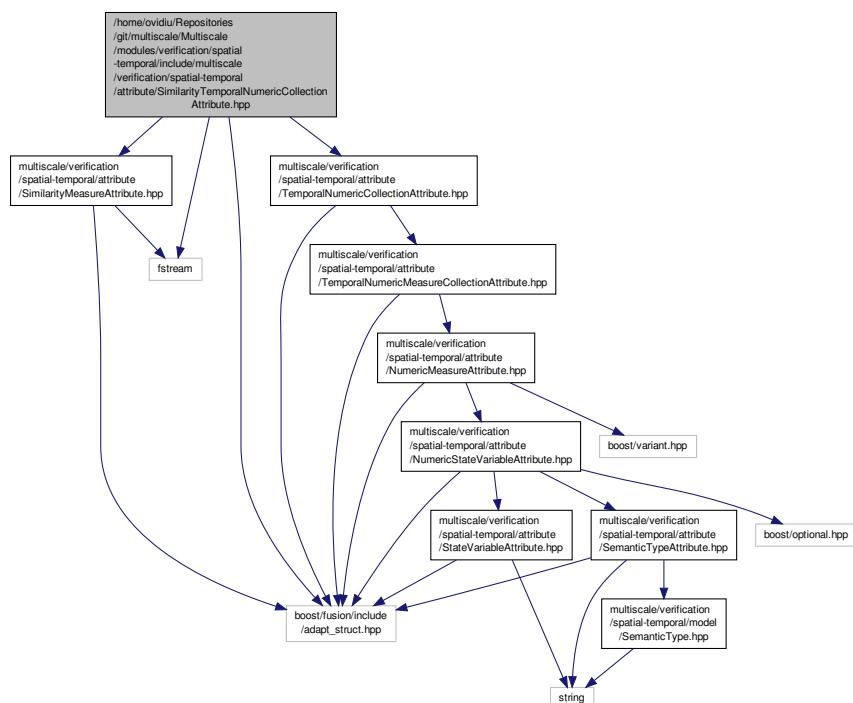
- std::ostream & multiscale::verification::operator<< (std::ostream &out, const SimilarityMeasureType &similarityMeasureType)

*Overload the output stream operator for the enumeration.*

## 8.147 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SimilarityTemporalNumericCollectionAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SimilarityMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for SimilarityTemporalNumericCollectionAttribute.hpp:



## Classes

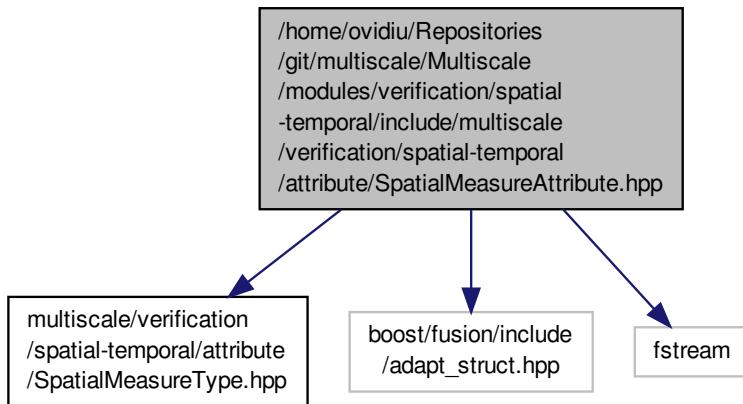
- class `multiscale::verification::SimilarityTemporalNumericCollectionAttribute`  
*Class for representing a similarity temporal numeric collection attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.148 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureType.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
Include dependency graph for SpatialMeasureAttribute.hpp:
```



### Classes

- class [multiscale::verification::SpatialMeasureAttribute](#)  
*Class for representing a spatial measure attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)
- [multiscale::verification::spatialmeasure](#)

### Functions

- void [multiscale::verification::spatialmeasure::validateSpatialMeasureType](#) (const [SpatialMeasureType](#) &[spatialMeasureType](#))  
*Check if the given spatial measure type is valid.*
- void [multiscale::verification::spatialmeasure::validateSpatialMeasureTypeIndex](#) (const std::size\_t &[spatialMeasureTypeIndex](#))  
*Check if the given spatial measure type index is valid.*
- size\_t [multiscale::verification::spatialmeasure::computeSpatialMeasureTypeIndex](#) (const [SpatialMeasureType](#) &[spatialMeasureType](#))  
*Compute the index of the spatial measure type.*

**8.149 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp**  
**File Reference**

**1201**

- `SpatialMeasureType multiscale::verification::spatialmeasure::computeSpatialMeasureType (const std::size_t & spatialMeasureTypeIndex)`

*Compute the spatial measure type from the given index.*
- `double multiscale::verification::spatialmeasure::getMinValidSpatialMeasureValue (const SpatialMeasureType & spatialMeasureType)`

*Get the minimum valid value for the given spatial measure type.*
- `double multiscale::verification::spatialmeasure::getMaxValidSpatialMeasureValue (const SpatialMeasureType & spatialMeasureType)`

*Get the maximum valid value for the given spatial measure type.*
- `std::ostream & multiscale::verification::operator<< (std::ostream & out, const SpatialMeasureType & spatialMeasureType)`

*Overload the output stream operator for the enumeration.*

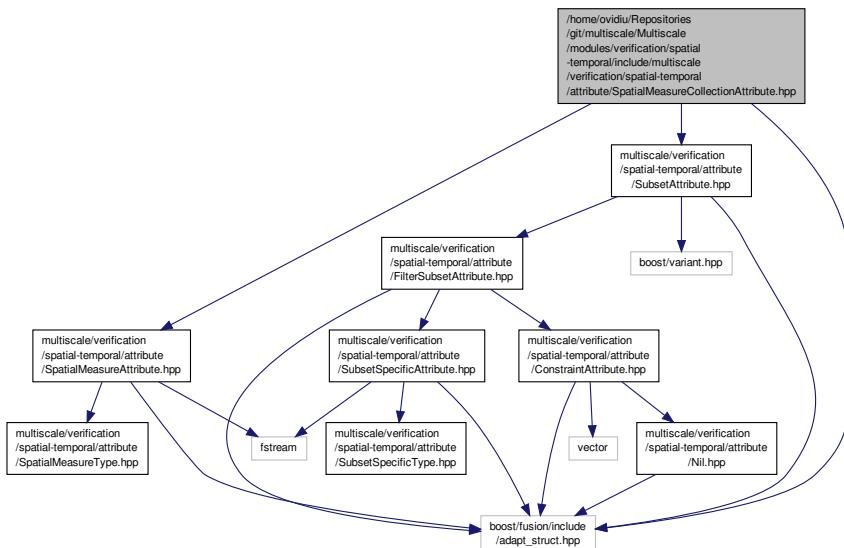
**Variables**

- `static const std::size_t multiscale::verification::NR_SPATIAL_MEASURE_TYPES = static_cast<std::size_t>(SpatialMeasureType::NrOfSpatialMeasureTypeEntries)`

*An std::size\_t constant which stores the number of spatial measure type entries.*

**8.149 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp File Reference**

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for SpatialMeasureCollectionAttribute.hpp:
```



## Classes

- class [multiscale::verification::SpatialMeasureCollectionAttribute](#)

*Class used to represent a spatial measure collection attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.150 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialMeasureType.hpp File Reference

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### Enumerations

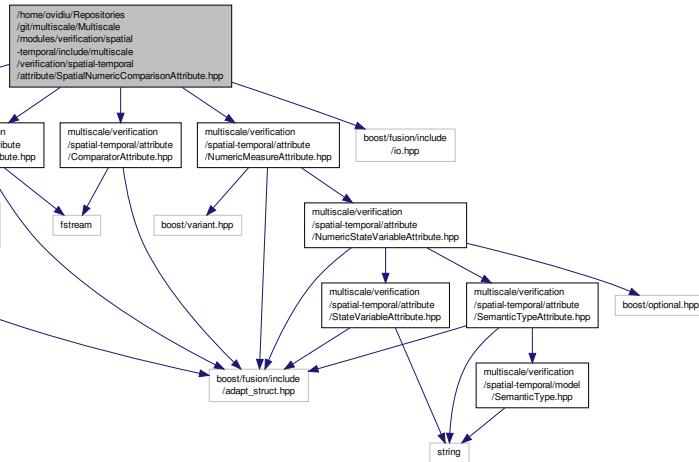
- enum [multiscale::verification::SpatialMeasureType](#) : unsigned int {
 multiscale::verification::SpatialMeasureType::Clusteredness = 0, [multiscale::verification::SpatialMeasureType::Density](#), [multiscale::verification::SpatialMeasureType::Area](#), [multiscale::verification::SpatialMeasureType::Perimeter](#), [multiscale::verification::SpatialMeasureType::DistanceFromOrigin](#), [multiscale::verification::SpatialMeasureType::Angle](#), [multiscale::verification::SpatialMeasureType::TriangleMeasure](#), [multiscale::verification::SpatialMeasureType::RectangleMeasure](#), [multiscale::verification::SpatialMeasureType::CircleMeasure](#), [multiscale::verification::SpatialMeasureType::CentroidX](#), [multiscale::verification::SpatialMeasureType::CentroidY](#), [multiscale::verification::SpatialMeasureType::NrOfSpatialMeasureTypeEntries](#) }

*Enumeration for representing the types of spatial measures.*

## 8.151 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SpatialNumericComparisonAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/fusion/include/io.hpp>
```

Include dependency graph for SpatialNumericComparisonAttribute.hpp:



## Classes

- class [multiscale::verification::SpatialNumericComparisonAttribute](#)

*Class for representing a spatial numeric comparison attribute.*

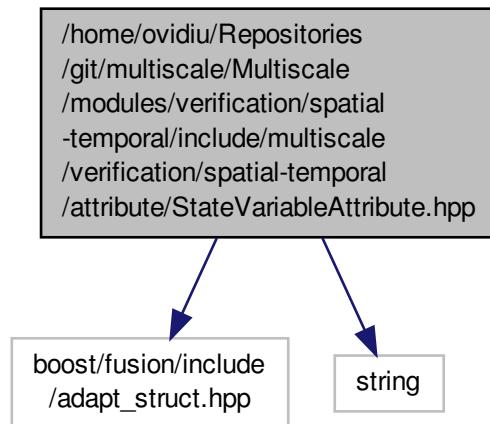
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.152 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/StateVariableAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <string>
```

Include dependency graph for StateVariableAttribute.hpp:



## Classes

- class [multiscale::verification::StateVariableAttribute](#)

*Class for representing a state variable attribute.*

## Namespaces

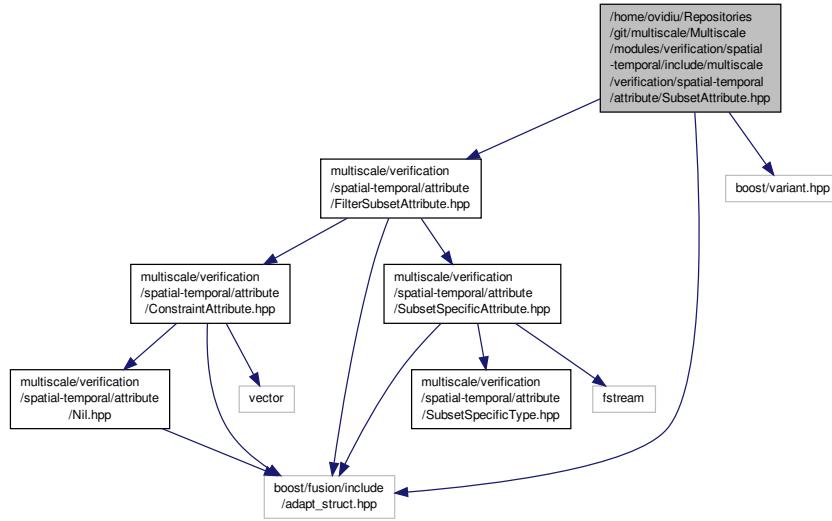
- [multiscale](#)
- [multiscale::verification](#)

## 8.153 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetAttribute.hpp

### File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/FilterSubsetAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
```

Include dependency graph for SubsetAttribute.hpp:



## Classes

- class [multiscale::verification::SubsetAttribute](#)

*Class for representing a subset attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## TypeDefs

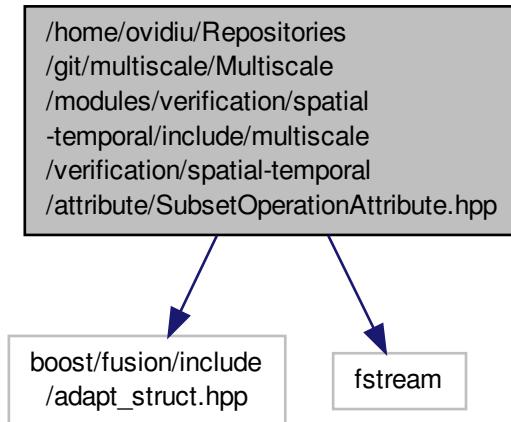
- [typedef boost::variant<SubsetSpecificAttribute, FilterSubsetAttribute, boost::recursive\\_wrapper<SubsetSubsetOperationAttribute>, boost::recursive\\_wrapper<SubsetAttribute>> multiscale::verification::SubsetAttributeType](#)

*Variant for a subset attribute.*

## 8.154 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for SubsetOperationAttribute.hpp:



## Classes

- class [multiscale::verification::SubsetOperationAttribute](#)  
*Class for representing a subset operation attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::SubsetOperationType](#) : unsigned int { [multiscale::verification::SubsetOperationType::Difference](#) = 0, [multiscale::verification::SubsetOperationType::Intersection](#), [multiscale::verification::SubsetOperationType::Union](#) }  
*Enumeration for representing the types of subset operations.*

## Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const SubsetOperationType &subsetOperationType)`  
*Overload the output stream operator for the enumeration.*

## 8.155 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp File Reference

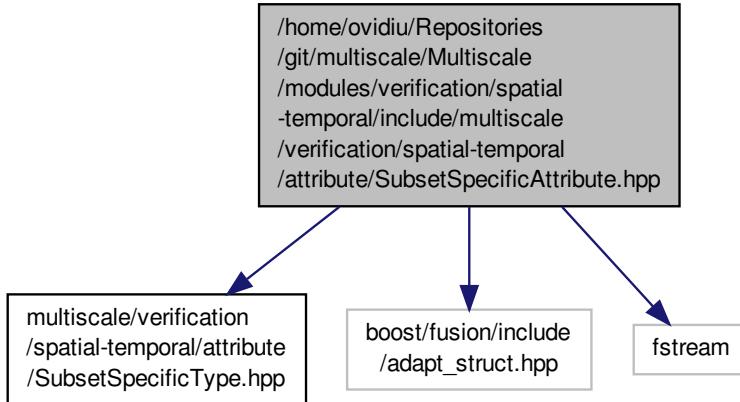
```
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
```

Reference

Type.hpp"

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for SubsetSpecificAttribute.hpp:



## Classes

- class multiscale::verification::SubsetSpecificAttribute

*Class for representing a subset specific attribute.*

## Namespaces

- multiscale
- multiscale::verification
- multiscale::verification::subsetspecific

## Functions

- void multiscale::verification::subsetspecific::validateSubsetSpecificType (const SubsetSpecificType &subsetSpecificType)  
*Check if the given subset specific type is valid.*
- void multiscale::verification::subsetspecific::validateSubsetSpecificTypeIndex (const std::size\_t &subsetSpecificTypeIndex)  
*Check if the given subset specific type index is valid.*
- size\_t multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex (const SubsetSpecificType &subsetSpecificType)  
*Compute the index of the subset specific type.*
- SubsetSpecificType multiscale::verification::subsetspecific::computeSubsetSpecificType (const std::size\_t &subsetSpecificTypeIndex)  
*Compute the subset specific type from the given index.*
- std::ostream & multiscale::verification::operator<< (std::ostream &out, const SubsetSpecificType &subsetSpecificType)  
*Overload the output stream operator for the enumeration.*

## Variables

- static const std::size\_t [multiscale::verification::NR\\_SUBSET\\_SPECIFIC\\_TYPES](#) = static\_cast<std::size\_t>(SubsetSpecificType::NrOfSubsetSpecificTypeEntries)

*An std::size\_t constant which stores the number of subset specific type entries.*

## 8.156 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSpecificType.hpp File Reference

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### Enumerations

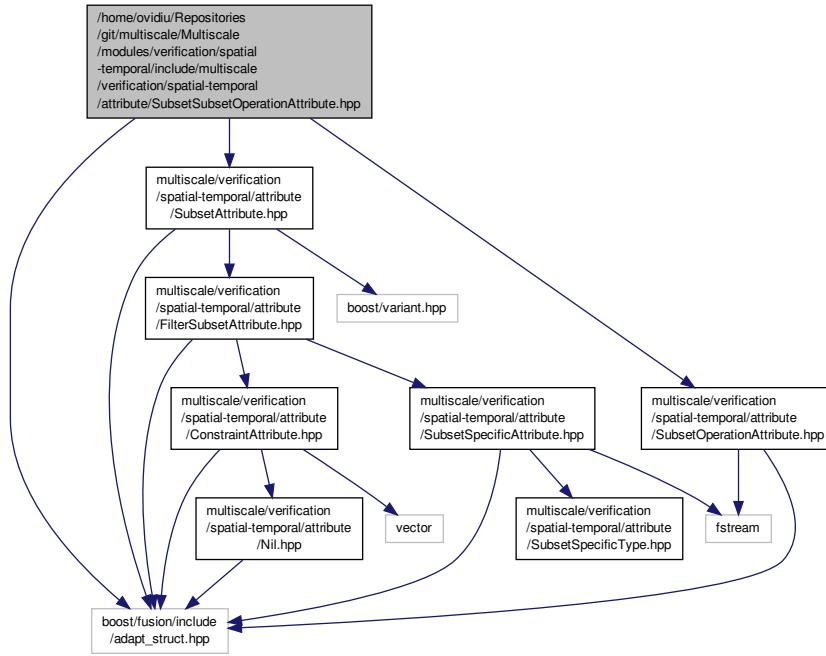
- enum [multiscale::verification::SubsetSpecificType](#) : unsigned int { [multiscale::verification::SubsetSpecificType::Clusters](#) = 0, [multiscale::verification::SubsetSpecificType::Regions](#), [multiscale::verification::SubsetSpecificType::NrOfSubsetSpecificTypeEntries](#) }

*Enumeration for representing a specific subset type.*

## 8.157 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSubsetOperationAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SubsetAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for SubsetSubsetOperationAttribute.hpp:



## Classes

- class [multiscale::verification::SubsetSubsetOperationAttribute](#)  
*Class for representing a subset subset operation attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.158 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/NotLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/FutureLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/GlobalLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NextLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NextKLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/OrLogicPropertyAttribute.hpp"

```

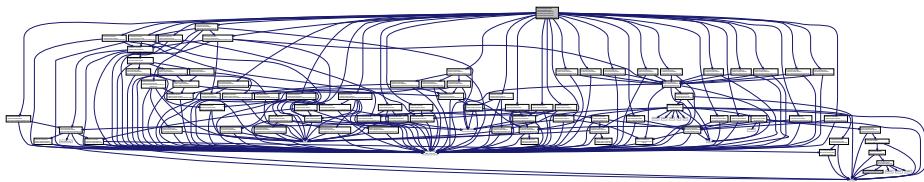
```
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/AndLogicProperty<-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Implication<-
LogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Equivalence<-
LogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UntilLogic<-
PropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/PrimaryLogic<-
PropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumeric<-
ComparisonAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ChangeTemporal<-
NumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumeric<-
TemporalAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumeric<-
TemporalAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ChangeTemporal<-
NumericCollectionAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Timeseries<-
TimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Homogeneous<-
HomogeneousTimeseriesAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/PrimaryNumeric<-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumeric<-
NumericAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumeric<-
NumericAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericSpatial<-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/PrimaryConstraint<-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NotConstraint<-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/OrConstraint<-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/AndConstraint<-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Implication<-
ConstraintAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Equivalence<-
ConstraintAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnarySpatial<-
ConstraintAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryType<-
ConstraintAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetSubset<-
OperationAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumeric<-
FilterAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumeric<-
FilterAttribute.hpp"
```

8.159 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.h

### Attribute.hpp File

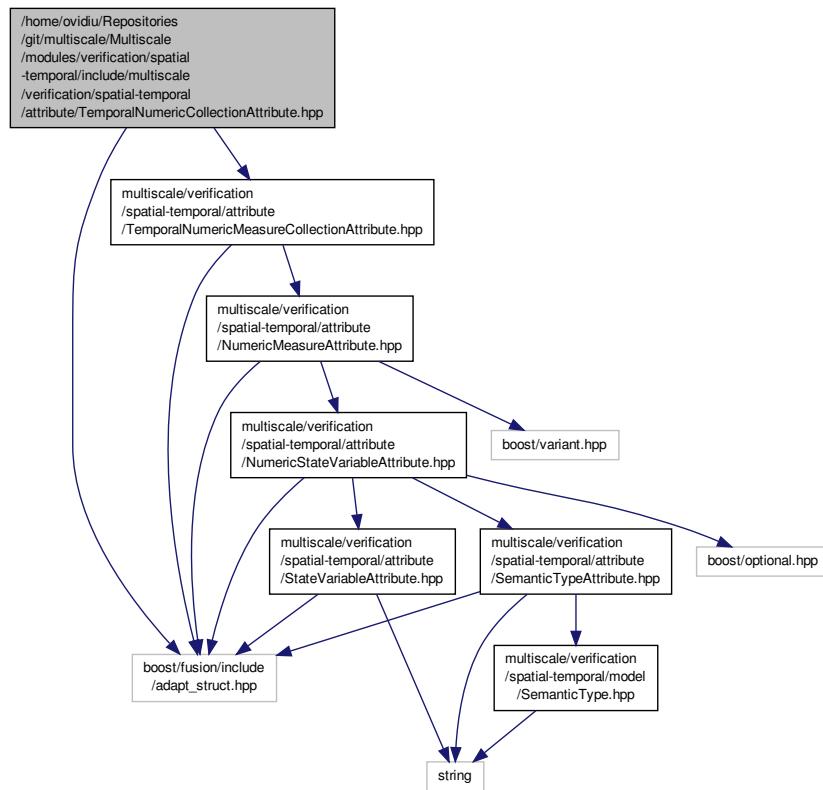
Include dependency graph for SynthesizedAttribute.hpp:

1211



8.159 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for TemporalNumericCollectionAttribute.hpp:
```



## Classes

- class [multiscale::verification::TemporalNumericCollectionAttribute](#)

*Class for representing a temporal numeric collection attribute.*

## Namespaces

- multiscale
- multiscale::verification

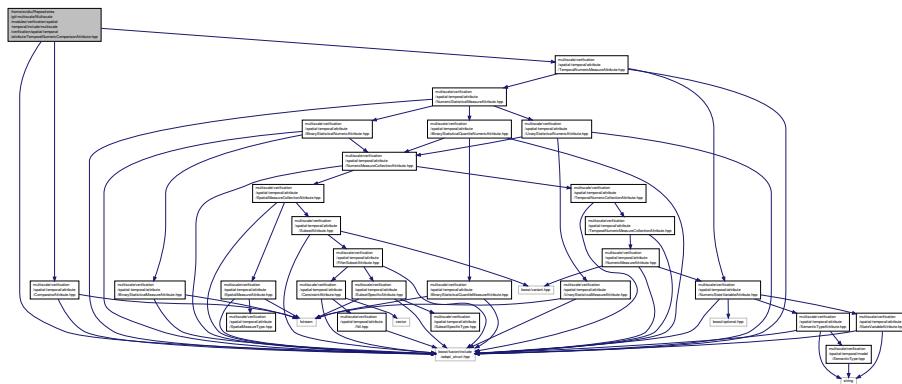
## Typedefs

- `typedef boost::variant<TemporalNumericMeasureCollectionAttribute, boost::recursive_wrapper<ChangeTemporalNumericCollectionAttribute>, boost::recursive_wrapper<TimeseriesTimeseriesComponentAttribute>, boost::recursive_wrapper<HomogeneousHomogeneousTimeseriesAttribute>> > > multiscale::verification::TemporalNumericCollectionType`

*Variant for the temporal numeric collection attribute.*

## 8.160 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericComparisonAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for TemporalNumericComparisonAttribute.hpp:
```



## Classes

- class `multiscale::verification::TemporalNumericComparisonAttribute`

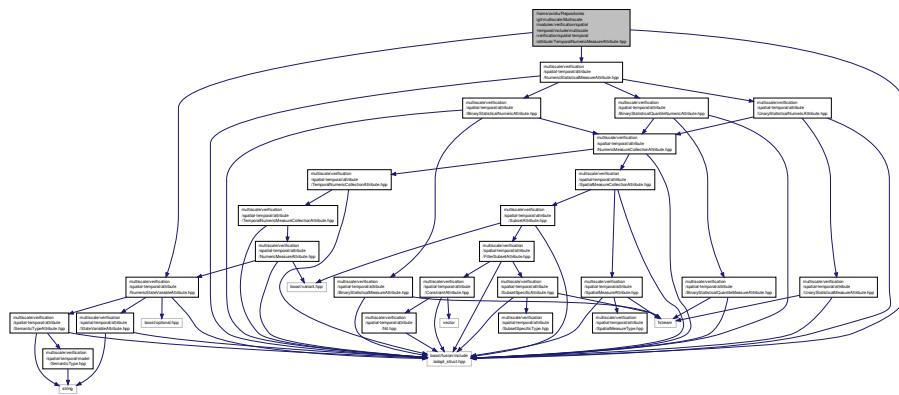
*Class for representing a temporal numeric comparison attribute.*

## Namespaces

- multiscale
- multiscale::verification

## 8.161 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericStatisticalMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for TemporalNumericMeasureAttribute.hpp:
```



## Classes

- class `multiscale::verification::TemporalNumericMeasureAttribute`

*Class for representing a temporal numeric measure attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

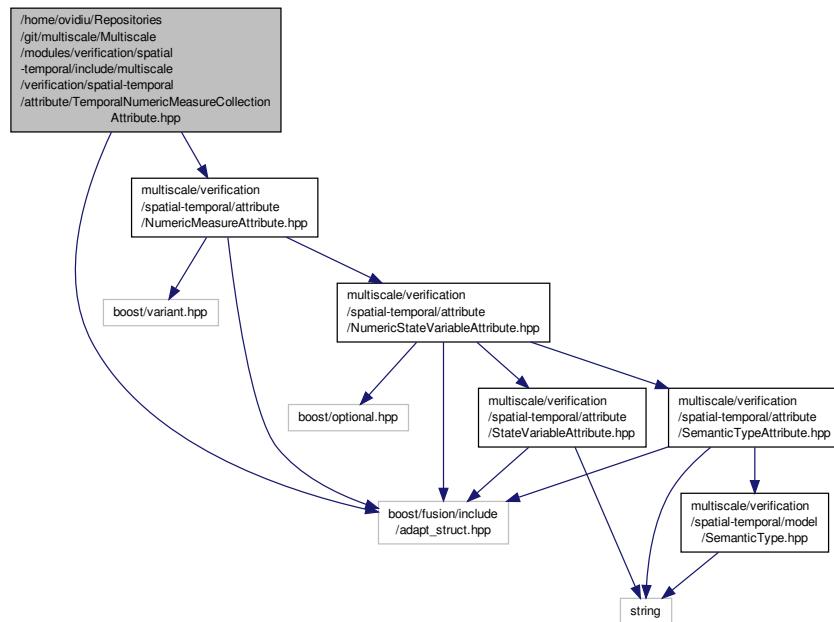
## TypeDefs

- `typedef boost::variant< double,  
NumericStateVariableAttribute,  
NumericStatisticalMeasureAttribute,  
boost::recursive_wrapper  
< UnaryNumericTemporalAttribute >  
, boost::recursive_wrapper  
< BinaryNumericTemporalAttribute >  
, boost::recursive_wrapper  
< TemporalNumericMeasureAttribute > > multiscale::verification::TemporalNumericMeasureType`

*Variant for the temporal numeric measure attribute.*

## 8.162 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureCollectionAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for TemporalNumericMeasureCollectionAttribute.hpp:
```



### Classes

- class [multiscale::verification::TemporalNumericMeasureCollectionAttribute](#)  
*Class for representing temporal numeric measure collection attributes.*

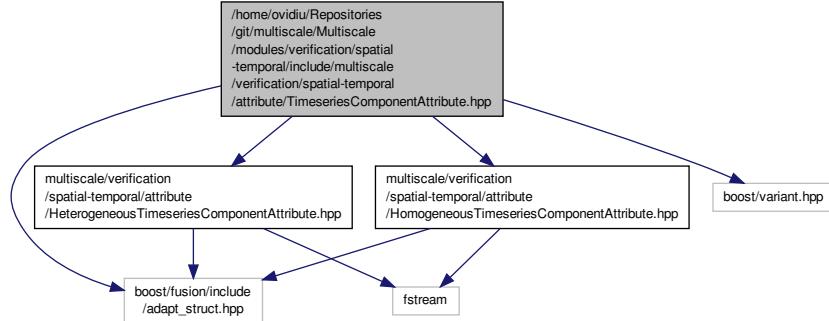
### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.163 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesComponentAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/HeterogeneousTimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Homogeneous<
```

```
TimeseriesComponentAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <boost/variant.hpp>
Include dependency graph for TimeseriesComponentAttribute.hpp:
```



## Classes

- class [multiscale::verification::TimeseriesComponentAttribute](#)

*Class for representing a timeseries component attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Typedefs

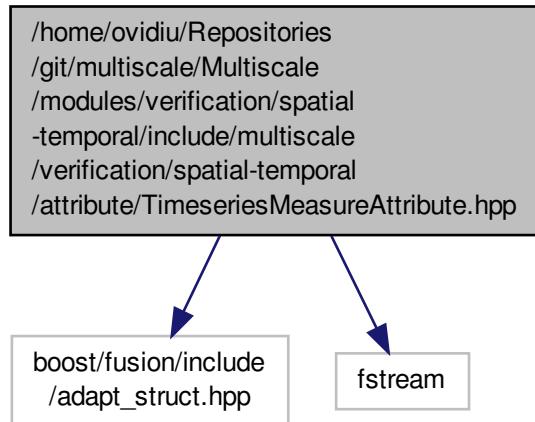
- [typedef boost::variant<HeterogeneousTimeseriesComponentAttribute, HomogeneousTimeseriesComponentAttribute> multiscale::verification::TimeseriesComponentType](#)

*Variant for the timeseries component attribute.*

## 8.164 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for TimeseriesMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::TimeseriesMeasureAttribute](#)  
*Class for representing a timeseries measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::TimeseriesMeasureType](#) : unsigned int { [multiscale::verification::TimeseriesMeasureType::EnteringTime](#) = 0, [multiscale::verification::TimeseriesMeasureType::EnteringValue](#) }  
*Enumeration for representing a timeseries measure type.*

## Functions

- [std::ostream & multiscale::verification::operator<<](#) ([std::ostream &out](#), const [TimeseriesMeasureType](#) &[timeseriesMeasureType](#))  
*Overload the output stream operator for the enumeration.*

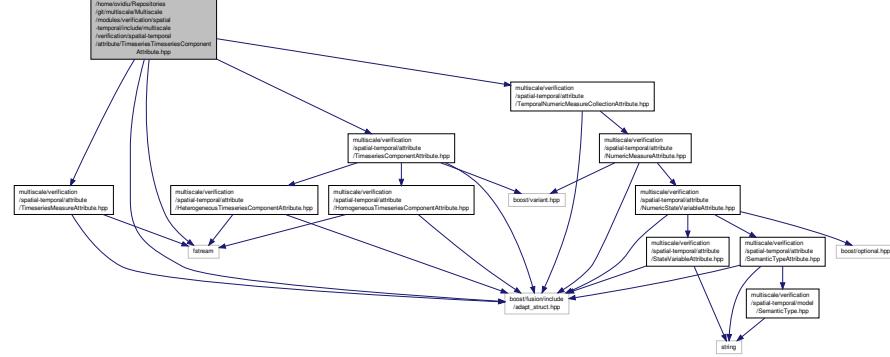
## 8.165 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TimeseriesTimeseriesComponentAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TimeseriesTimeseriesComponentAttribute.hpp"
```

MeasureAttribute.hpp"

```
#include "multiscale/verification/spatial-temporal/attribute/TimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for TimeseriesTimeseriesComponentAttribute.hpp.



## Classes

- class multiscale::verification::TimeseriesTimeseriesComponentAttribute

*Class for representing a timeseries timeseries component attribute.*

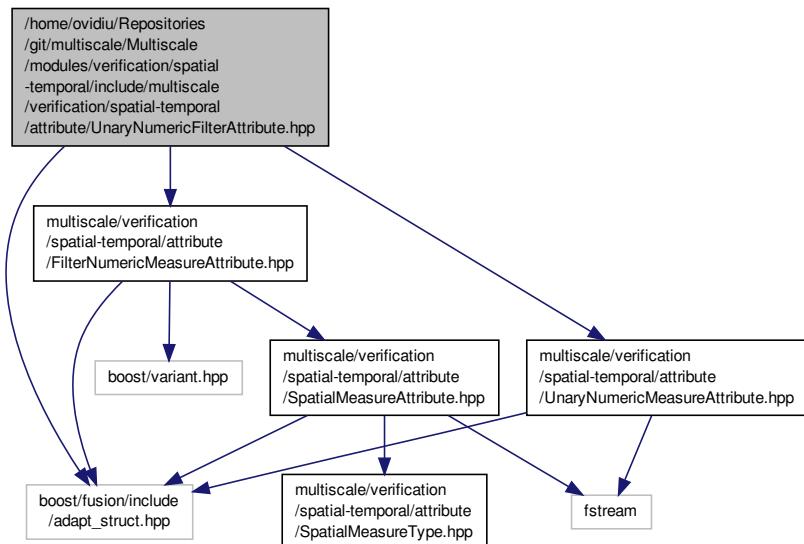
## Namespaces

- multiscale
  - multiscale::verification

8.166 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericFilterAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/FilterNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for UnaryNumericFilterAttribute.hpp:



## Classes

- class [multiscale::verification::UnaryNumericFilterAttribute](#)

*Class for representing a unary numeric filter attribute.*

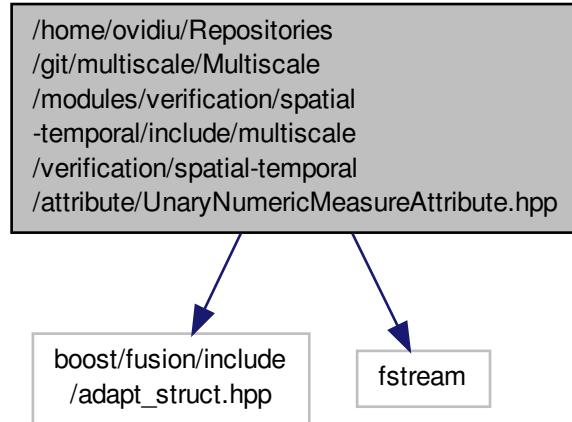
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.167 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for UnaryNumericMeasureAttribute.hpp:



## Classes

- class [multiscale::verification::UnaryNumericMeasureAttribute](#)  
*Class for representing a unary numeric measure attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

- enum [multiscale::verification::UnaryNumericMeasureType](#) : unsigned int {  
    [multiscale::verification::UnaryNumericMeasureType::Abs](#) = 0,   [multiscale::verification::UnaryNumericMeasureType::Cell](#),   [multiscale::verification::UnaryNumericMeasureType::Floor](#),   [multiscale::verification::UnaryNumericMeasureType::Round](#),  
    [multiscale::verification::UnaryNumericMeasureType::Sign](#),   [multiscale::verification::UnaryNumericMeasureType::Sqrt](#),   [multiscale::verification::UnaryNumericMeasureType::Trunc](#) }

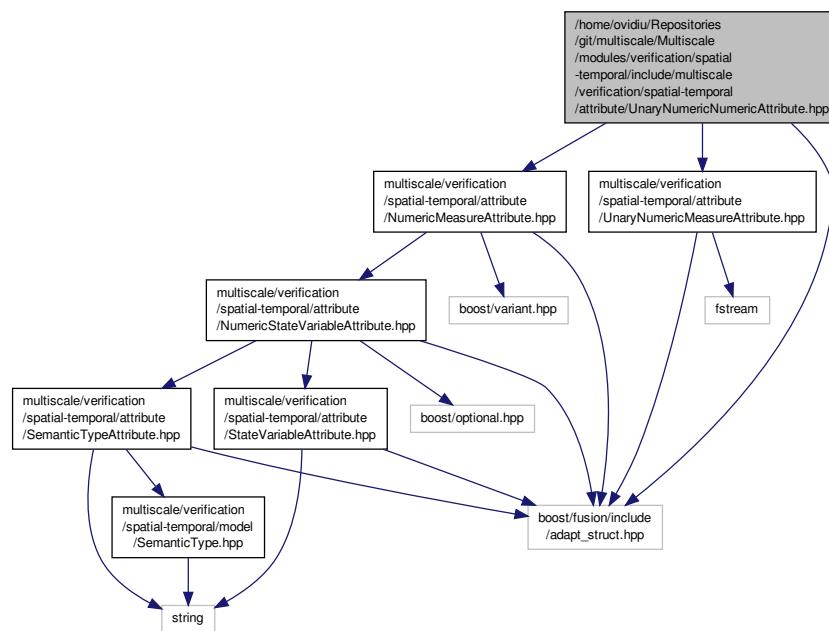
*Enumeration for representing a unary numeric measure type.*

## Functions

- std::ostream & [multiscale::verification::operator<<](#) (std::ostream &out, const [UnaryNumericMeasureType](#) &[UnaryNumericMeasureType](#))  
*Overload the output stream operator for the enumeration.*

## 8.168 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for UnaryNumericAttribute.hpp:
```



### Classes

- class [multiscale::verification::UnaryNumericAttribute](#)  
*Class for representing a unary numeric numeric measure attribute.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

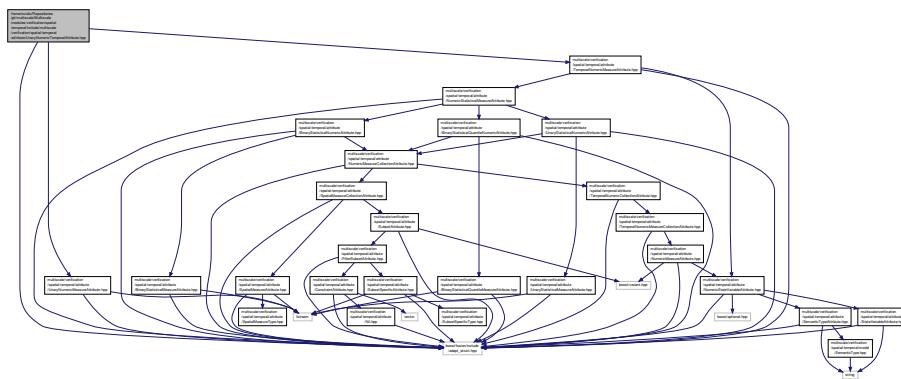
## 8.169 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryNumericTemporalAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumericAttribute.hpp"
```

MeasureAttribute.hpp"

#include <boost/fusion/include/adapt\_struct.hpp>

Include dependency graph for UnaryNumericTemporalAttribute.hpp:



## Classes

- class [multiscale::verification::UnaryNumericTemporalAttribute](#)

*Class for representing a unary numeric temporal measure attribute.*

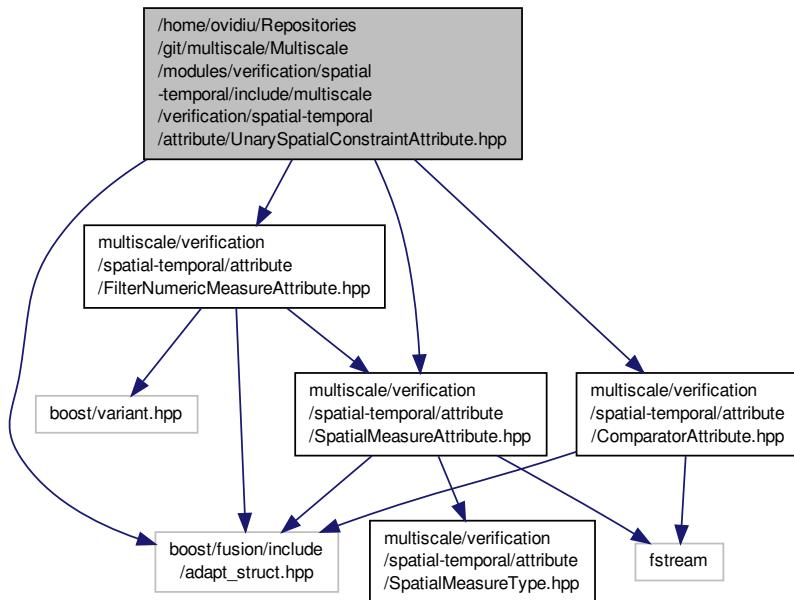
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

8.170 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnarySpatialConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/FilterNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for UnarySpatialConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::UnarySpatialConstraintAttribute](#)

*Class for representing a "unary" spatial constraint attribute.*

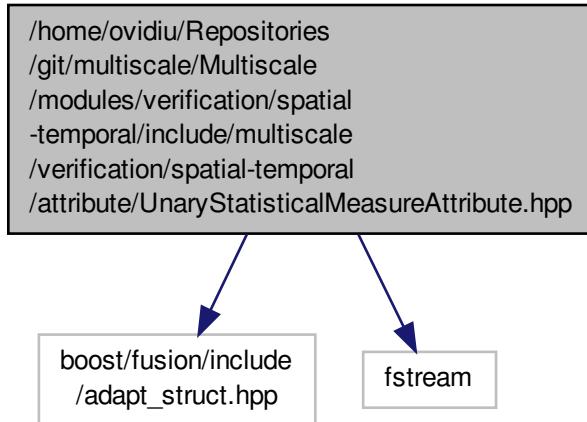
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.171 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp>
#include <fstream>
```

Include dependency graph for UnaryStatisticalMeasureAttribute.hpp:



## Classes

- class `multiscale::verification::UnaryStatisticalMeasureAttribute`  
*Class for representing a unary statistical measure attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## Enumerations

- enum `multiscale::verification::UnaryStatisticalMeasureType` : unsigned int {  
    `multiscale::verification::UnaryStatisticalMeasureType::Avg` = 0,   `multiscale::verification::UnaryStatisticalMeasureType::Count`,   `multiscale::verification::UnaryStatisticalMeasureType::Geomean`,   `multiscale::verification::UnaryStatisticalMeasureType::Harmean`,  
    `multiscale::verification::UnaryStatisticalMeasureType::Kurt`,                  `multiscale::verification::UnaryStatisticalMeasureType::Max`,   `multiscale::verification::UnaryStatisticalMeasureType::Median`,   `multiscale::verification::UnaryStatisticalMeasureType::Min`,  
    `multiscale::verification::UnaryStatisticalMeasureType::Mode`,                  `multiscale::verification::UnaryStatisticalMeasureType::Product`,   `multiscale::verification::UnaryStatisticalMeasureType::Skew`,   `multiscale::verification::UnaryStatisticalMeasureType::Stdev`,  
    `multiscale::verification::UnaryStatisticalMeasureType::Sum`,                  `multiscale::verification::UnaryStatisticalMeasureType::Var` }

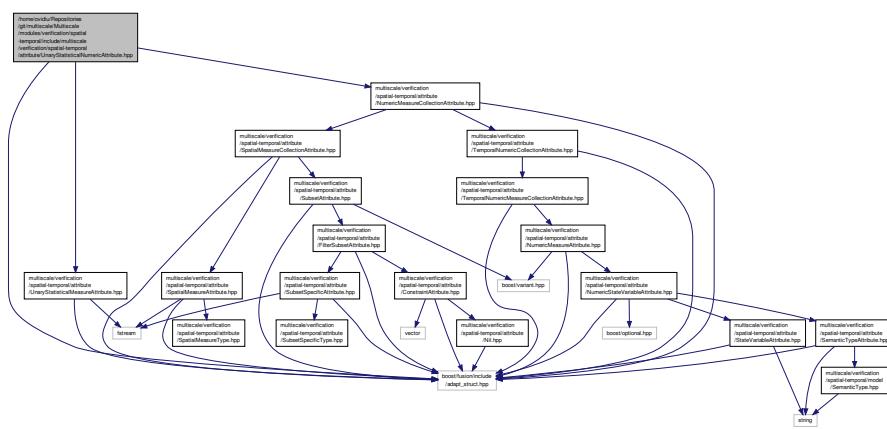
*Enumeration for representing a unary statistical measure type.*

## Functions

- `std::ostream & multiscale::verification::operator<<` (`std::ostream &out, const UnaryStatisticalMeasureType &unaryStatisticalMeasureType)`  
*Overload the output stream operator for the enumeration.*

## 8.172 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
Include dependency graph for UnaryStatisticalNumericAttribute.hpp:
```



### Classes

- class `multiscale::verification::UnaryStatisticalNumericAttribute`

*Class for representing a unary statistical numeric attribute.*

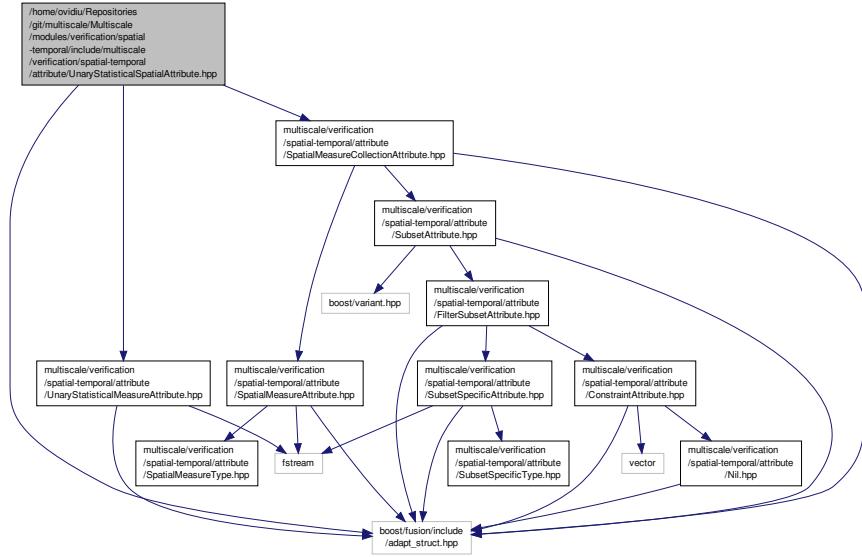
### Namespaces

- `multiscale`
- `multiscale::verification`

## 8.173 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryStatisticalSpatialAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for UnaryStatisticalSpatialAttribute.hpp:



## Classes

- class [multiscale::verification::UnaryStatisticalSpatialAttribute](#)

*Class for representing a unary statistical spatial attribute.*

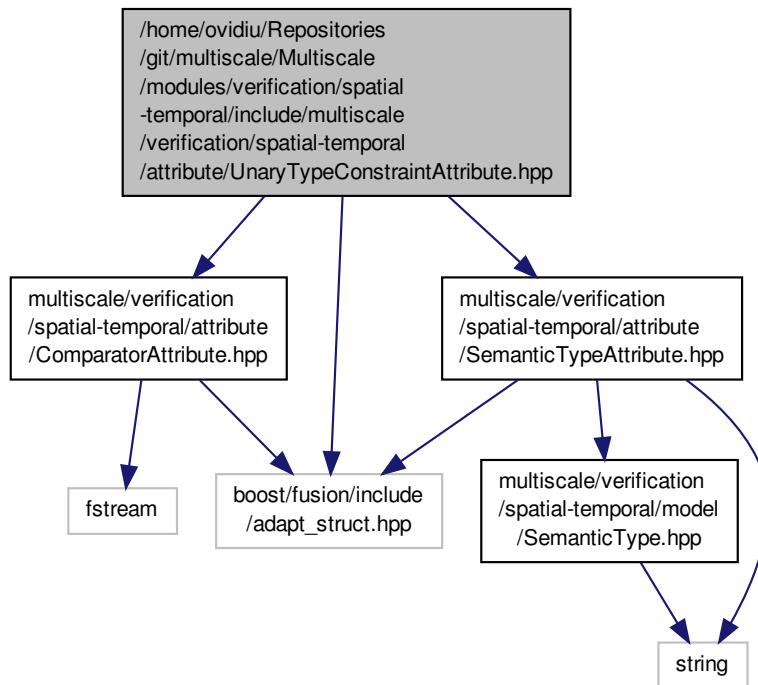
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.174 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UnaryTypeConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```

Include dependency graph for UnaryTypeConstraintAttribute.hpp:



## Classes

- class [multiscale::verification::UnaryTypeConstraintAttribute](#)

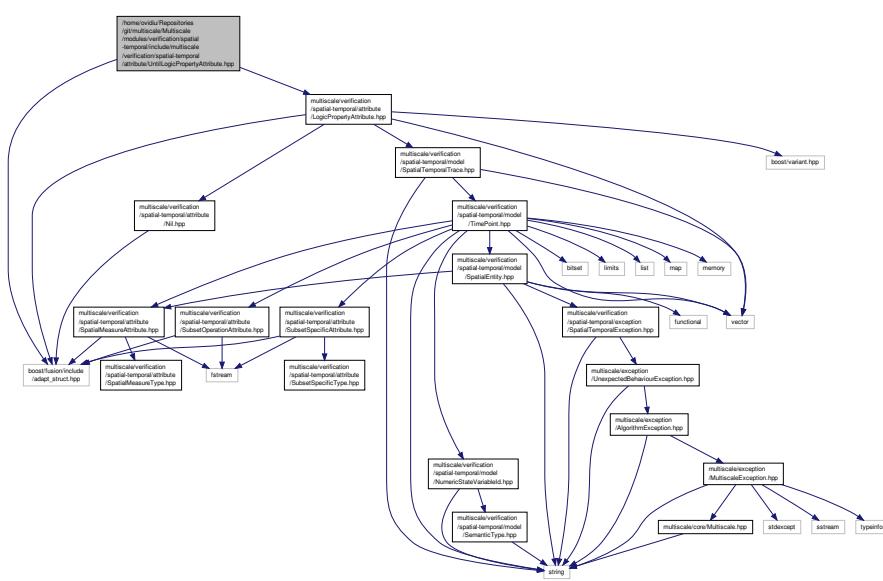
*Class for representing a "unary" type constraint attribute.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.175 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/UntilLogicPropertyAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include <boost/fusion/include/adapt_struct.hpp>
```



## Classes

- class `multiscale::verification::UntilLogicPropertyAttribute`

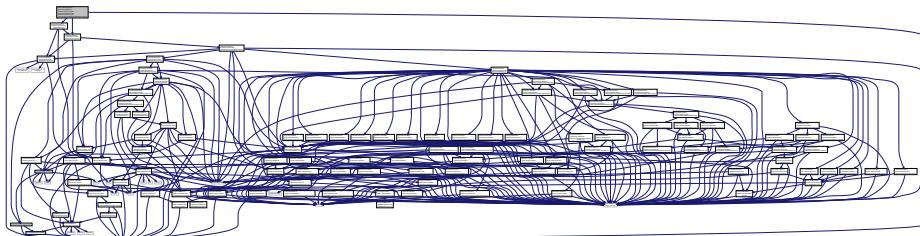
*Class for representing an "until" logic property attribute.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.176 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include <string>
Include dependency graph for ApproximateBayesianModelChecker.hpp:
```



## Classes

- class [multiscale::verification::ApproximateBayesianModelChecker](#)

*Class used to run approximate Bayesian model checking tasks.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

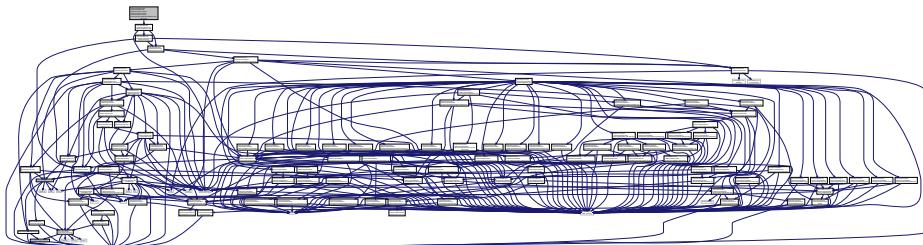
- enum [multiscale::verification::ApproximateBayesianModelCheckingResult](#) : unsigned int { [multiscale::verification::ApproximateBayesianModelCheckingResult::TRUE](#) = 0, [multiscale::verification::ApproximateBayesianModelCheckingResult::FALSE](#), [multiscale::verification::ApproximateBayesianModelCheckingResult::MORE\\_TRACES\\_REQUIRED](#) }

*Enumeration for representing the model checking result.*

## 8.177 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
```

Include dependency graph for ApproximateBayesianModelCheckerFactory.hpp:



## Classes

- class [multiscale::verification::ApproximateBayesianModelCheckerFactory](#)

*Class for creating [ApproximateBayesianModelChecker](#) instances.*

## Namespaces

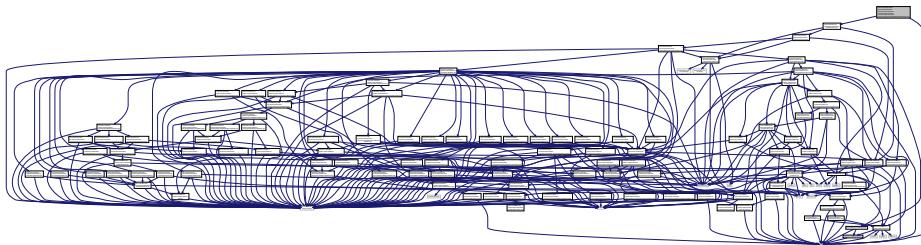
- [multiscale](#)
- [multiscale::verification](#)

8.178 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp File Reference

---

8.178 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp File Reference [229]

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include <string>
Include dependency graph for ApproximateProbabilisticModelChecker.hpp:
```



## Classes

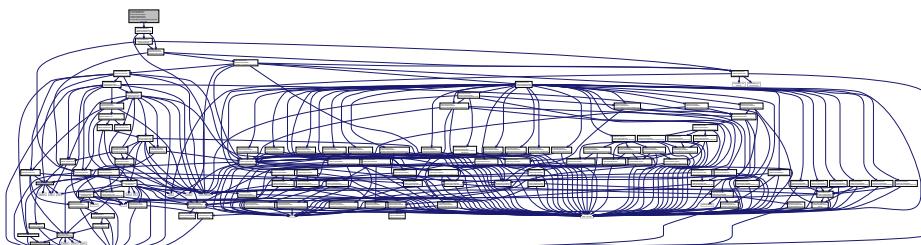
- class [multiscale::verification::ApproximateProbabilisticModelChecker](#)  
*Class used to run approximate probabilistic model checking tasks.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

8.179 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
Include dependency graph for ApproximateProbabilisticModelCheckerFactory.hpp:
```



## Classes

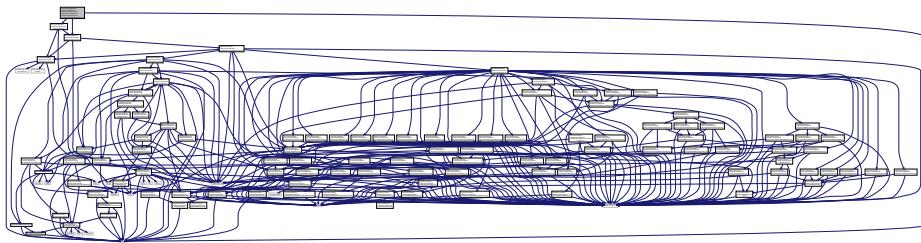
- class [multiscale::verification::ApproximateProbabilisticModelCheckerFactory](#)  
*Class for creating [ApproximateProbabilisticModelChecker](#) instances.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### 8.180 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include <string>
Include dependency graph for BayesianModelChecker.hpp:
```



## Classes

- class [multiscale::verification::BayesianModelChecker](#)  
*Class used to run Bayesian model checking tasks.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## Enumerations

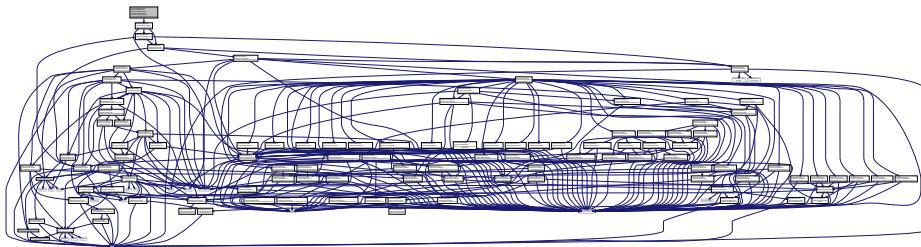
- enum [multiscale::verification::BayesianModelCheckingResult](#) : unsigned int { [multiscale::verification::BayesianModelCheckingResult::TRUE](#) = 0, [multiscale::verification::BayesianModelCheckingResult::FALSE](#), [multiscale::verification::BayesianModelCheckingResult::MORE\\_TRACES\\_REQUIRED](#) }

*Enumeration for representing the model checking result.*

### 8.181 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/BayesianModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
```

Include dependency graph for BayesianModelCheckerFactory.hpp:



## Classes

- class [multiscale::verification::BayesianModelCheckerFactory](#)  
*Class for creating `BayesianModelChecker` instances.*

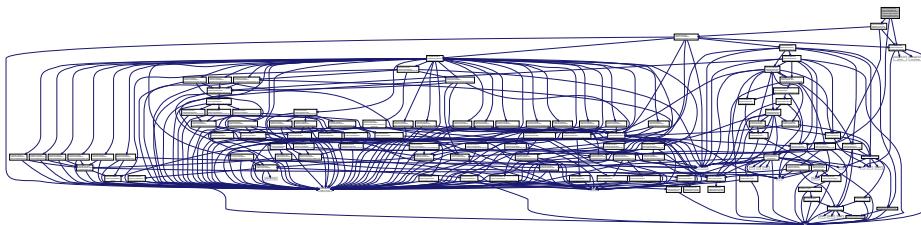
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

# 8.182 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
```

Include dependency graph for ModelChecker.hpp:



## Classes

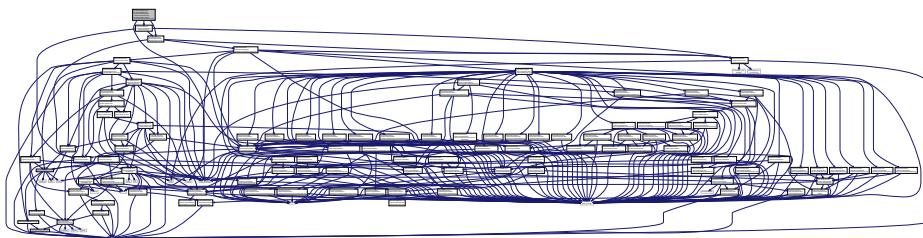
- class [multiscale::verification::ModelChecker](#)  
*Abstract class representing a generic model checker.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.183 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include <memory>
Include dependency graph for ModelCheckerFactory.hpp:
```



### Classes

- class [multiscale::verification::ModelCheckerFactory](#)

*Interface for different model checker factories.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

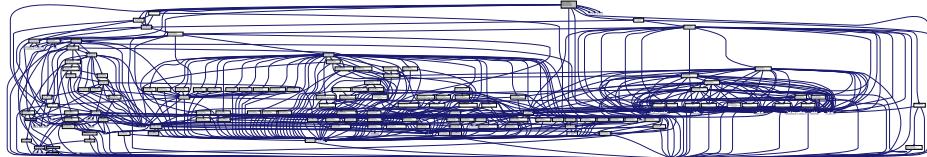
## 8.184 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingManager.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/data/LogicPropertyDataReader.hpp"
#include "multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemantics.hpp"
```

Table.hpp"

```
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <chrono>
#include <ctime>
#include <memory>
#include <string>
#include <thread>
#include <vector>
```

Include dependency graph for ModelCheckingManager.hpp:



## Classes

- class [multiscale::verification::ModelCheckingManager](#)

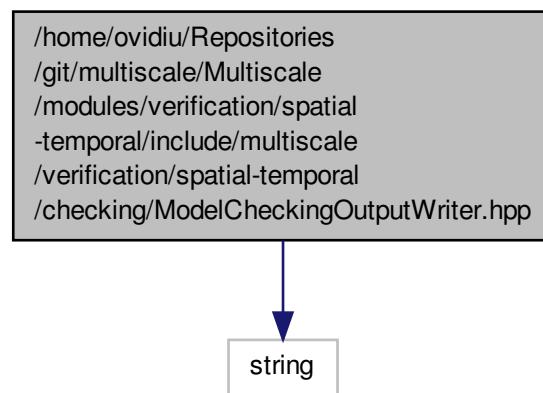
*Class for managing the model checking processes.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.185 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp File Reference

```
#include <string>
Include dependency graph for ModelCheckingOutputWriter.hpp:
```



## Classes

- class [multiscale::verification::ModelCheckingOutputWriter](#)

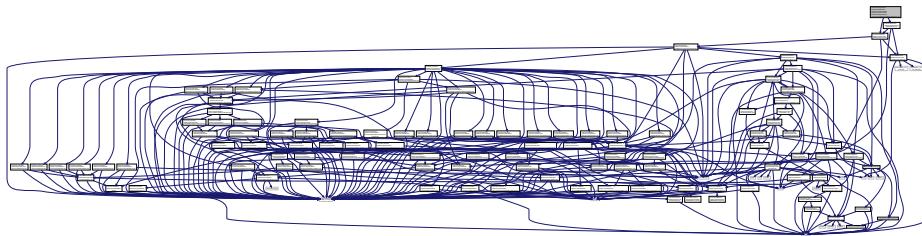
*Class used to output the model checkers progress.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.186 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
Include dependency graph for ProbabilisticBlackBoxModelChecker.hpp:
```



## Classes

- class [multiscale::verification::ProbabilisticBlackBoxModelChecker](#)

*Class used to run probabilistic black-box model checking tasks.*

## Namespaces

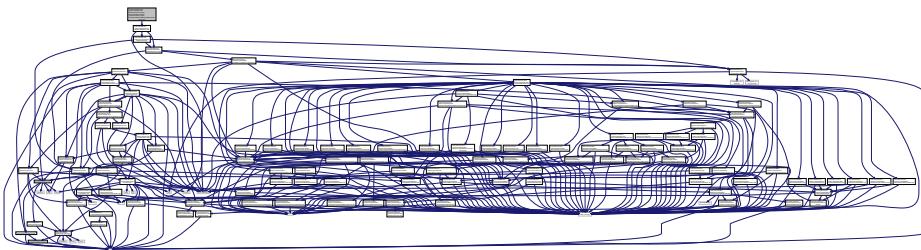
- [multiscale](#)
- [multiscale::verification](#)

## 8.187 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
```

Reference

Include dependency graph for ProbabilisticBlackBoxModelCheckerFactory.hpp:



## Classes

- class multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory  
*Class for creating ProbabilisticBlackBoxModelChecker instances.*

## Namespaces

- multiscale
- multiscale::verification

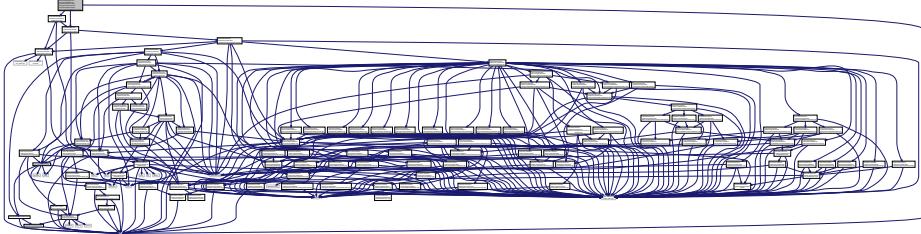
8.188 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/StatisticalModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
```

```
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
```

```
#include <string>
```

Include dependency graph for StatisticalModelChecker.hpp:



## Classes

- class multiscale::verification::StatisticalModelChecker  
*Class used to run statistical model checking tasks.*

## Namespaces

- multiscale
- multiscale::verification

## Enumerations

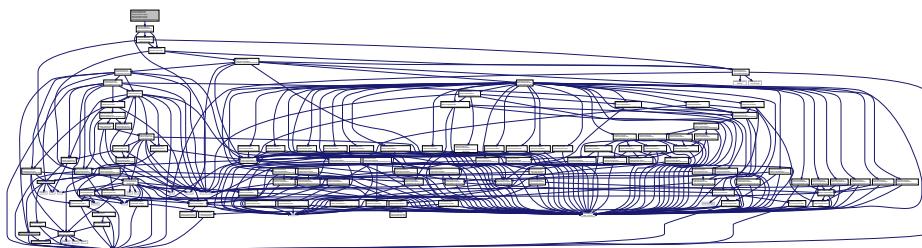
- enum multiscale::verification::StatisticalModelCheckingResult : int { multiscale::verification::StatisticalModelCheckingResult::TRUE = 0, multiscale::verification::StatisticalModelCheckingResult::FALSE, multiscale::verification::StatisticalModelCheckingResult::UNDECIDED, multiscale::verification::StatisticalModelCheckingResult::MORE\_TRACES\_REQUIRED }

*Enumeration for representing the model checking result.*

## 8.189 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/StatisticalModelCheckerFactory.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ModelCheckerFactory.hpp"
```

Include dependency graph for StatisticalModelCheckerFactory.hpp:



## Classes

- class multiscale::verification::StatisticalModelCheckerFactory

*Class for creating StatisticalModelChecker instances.*

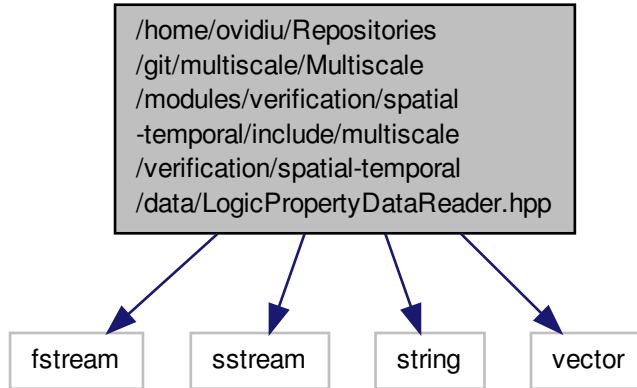
## Namespaces

- multiscale
- multiscale::verification

## 8.190 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/LogicPropertyDataReader.hpp File Reference

```
#include <fstream>
#include <sstream>
#include <string>
#include <vector>
```

Include dependency graph for LogicPropertyDataReader.hpp:



## Classes

- class [multiscale::verification::LogicPropertyDataReader](#)

*Class used to input logic properties.*

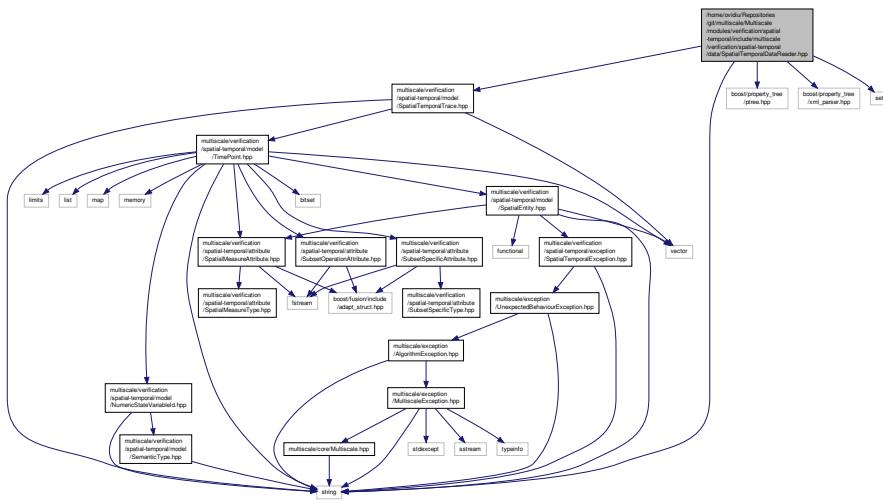
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.191 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include <set>
#include <string>
```

Include dependency graph for SpatialTemporalDataReader.hpp:



## Classes

- class [multiscale::verification::SpatialTemporalDataReader](#)

*Class for reading spatial temporal trace data from input files.*

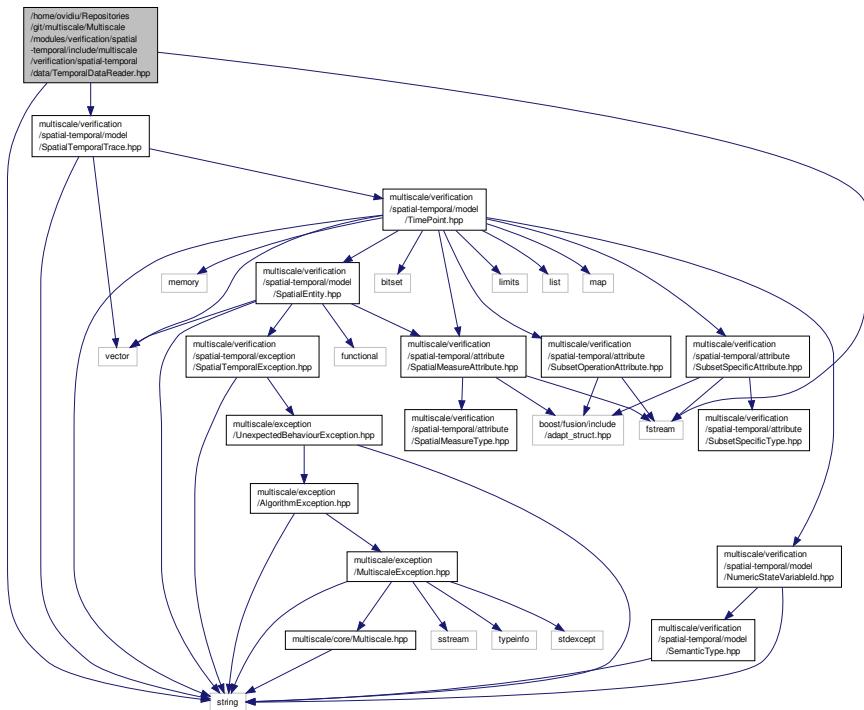
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.192 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/TemporalDataReader.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include <fstream>
#include <string>
```

Include dependency graph for TemporalDataReader.hpp:



## Classes

- class `multiscale::verification::TemporalDataReader`

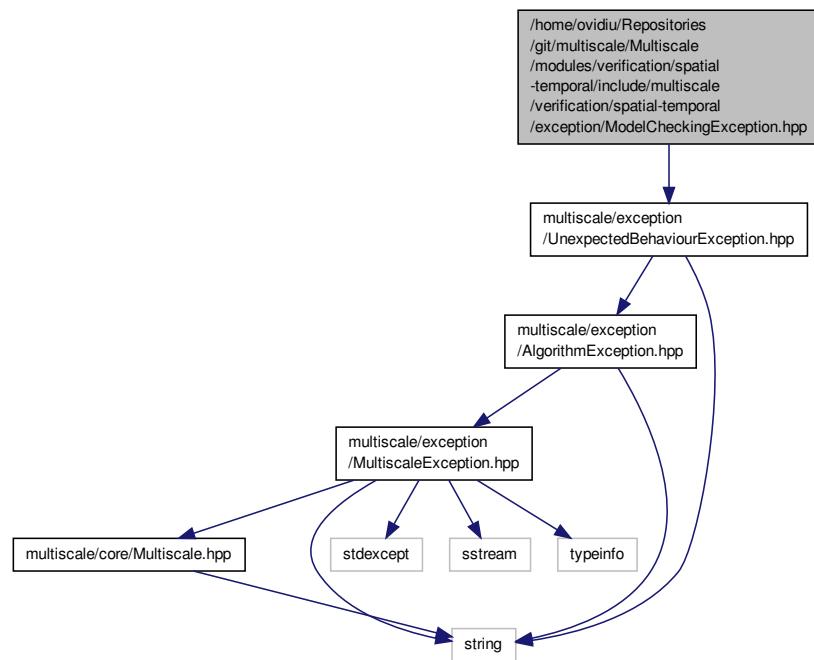
*Class for reading (non-spatial) timeseries data from a .csv file.*

## Namespaces

- multiscale
  - multiscale::verification

```
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
```

Include dependency graph for ModelCheckingException.hpp:



## Classes

- class [multiscale::verification::ModelCheckingException](#)

*Class for representing a model checking exception.*

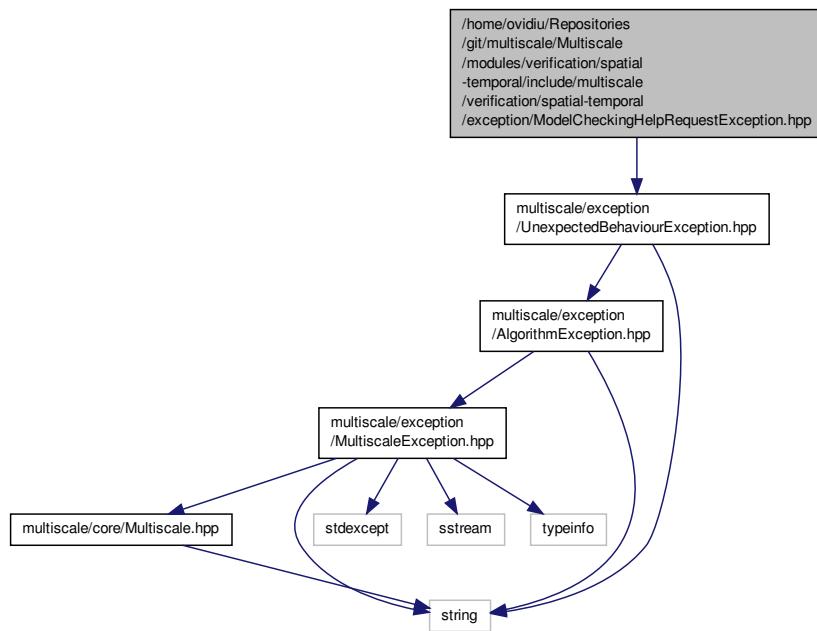
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.194 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingException.hpp File Reference

```
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
```

Include dependency graph for ModelCheckingHelpRequestException.hpp:



## Classes

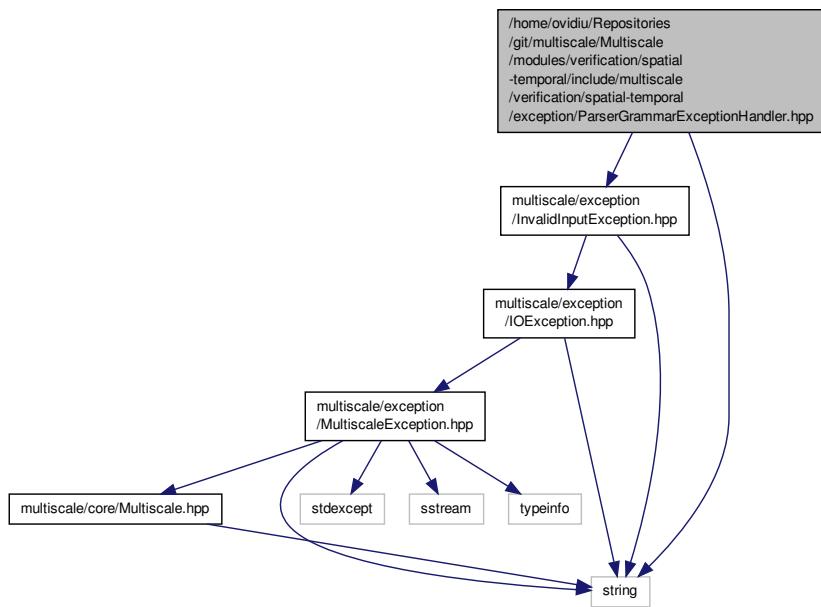
- class multiscale::verification::ModelCheckingHelpRequestException

*Class for representing a model checking help request exception.*

## Namespaces

- multiscale
- multiscale::verification

Include dependency graph for ParserGrammarExceptionHandler.hpp:



## Classes

- class [multiscale::verification::ParserGrammarExceptionHandler](#)

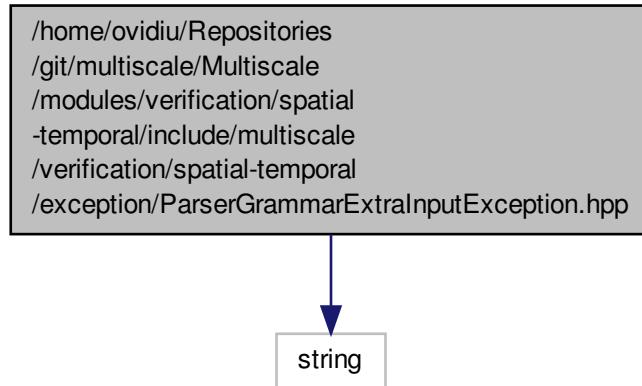
*Class for handling parser grammar exceptions.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.196 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarExtraInputException.hpp File Reference

```
#include <string>
```



## Classes

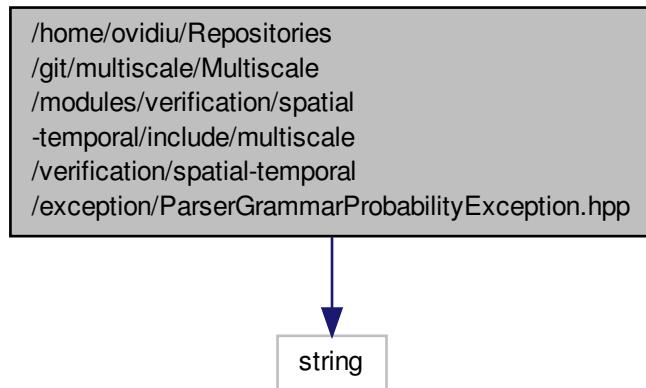
- class [multiscale::verification::ParserGrammarExtraInputException](#)

*Class for representing "extra input" exceptions in the parsing process.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

Include dependency graph for ParserGrammarProbabilityException.hpp:



## Classes

- class [multiscale::verification::ParserGrammarProbabilityException](#)

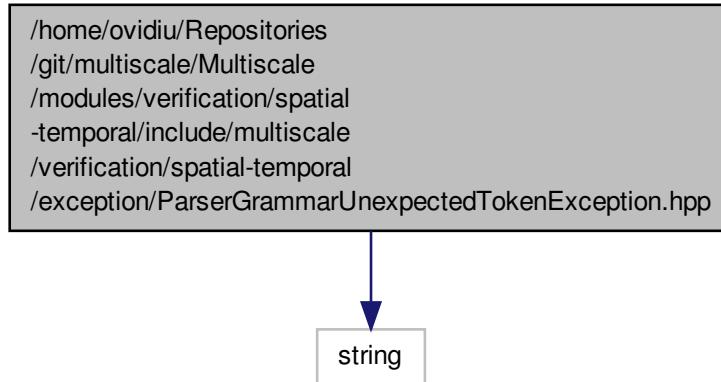
*Class for representing "probability" exceptions in the parsing process.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.198 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarUnexpectedTokenException.hpp File Reference

```
#include <string>
```



## Classes

- class [multiscale::verification::ParserGrammarUnexpectedTokenException](#)

*Class for representing "unexpected token" exceptions in the parsing process.*

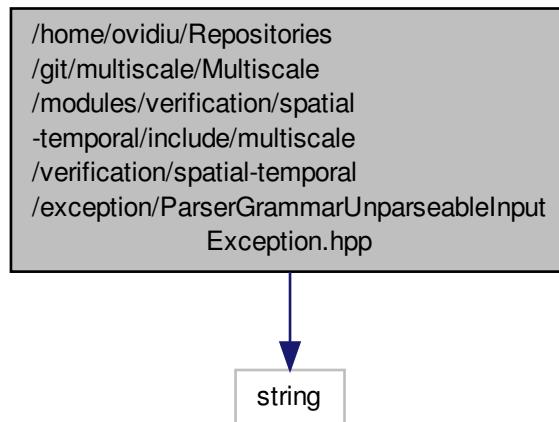
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.199 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserGrammarUnparseableInputException.hpp File Reference

```
#include <string>
```

Include dependency graph for ParserGrammarUnparseableInputException.hpp:



## Classes

- class [multiscale::verification::ParserGrammarUnparseableInputException](#)

*Class for representing "unparseable input" exceptions in the parsing process.*

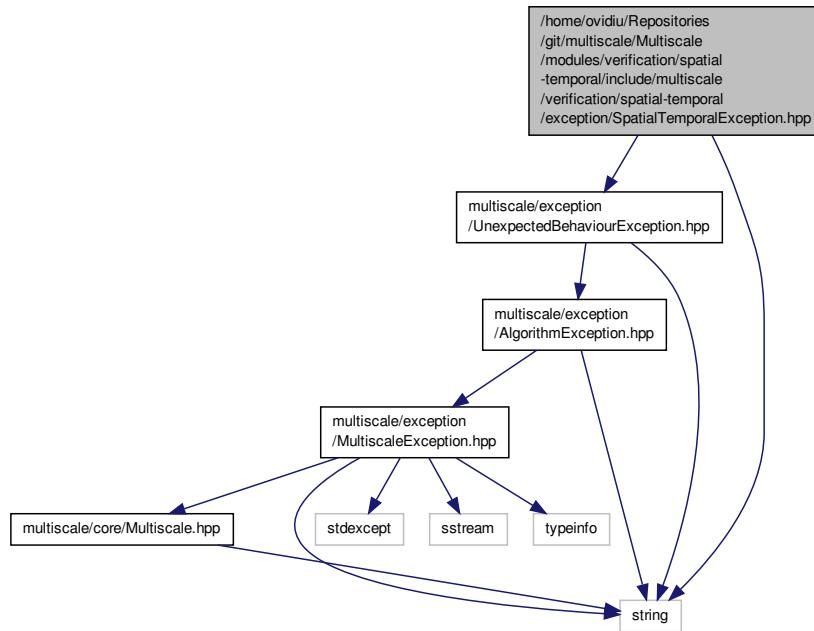
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.200 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp File Reference

```
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include <string>
```

Include dependency graph for SpatialTemporalException.hpp:



## Classes

- class [multiscale::verification::SpatialTemporalException](#)

*Class for representing a spatial temporal exception.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

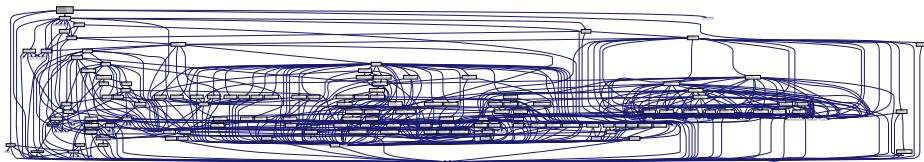
## 8.201 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/CommandLineModelChecking.hpp File Reference

```

#include "multiscale/core/Multiscale.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckingManager.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp"
#include <boost/program_options.hpp>

```

Include dependency graph for CommandLineModelChecking.hpp:



## Classes

- class [multiscale::verification::CommandLineModelChecking](#)

*Class for running model checkers from the command line.*

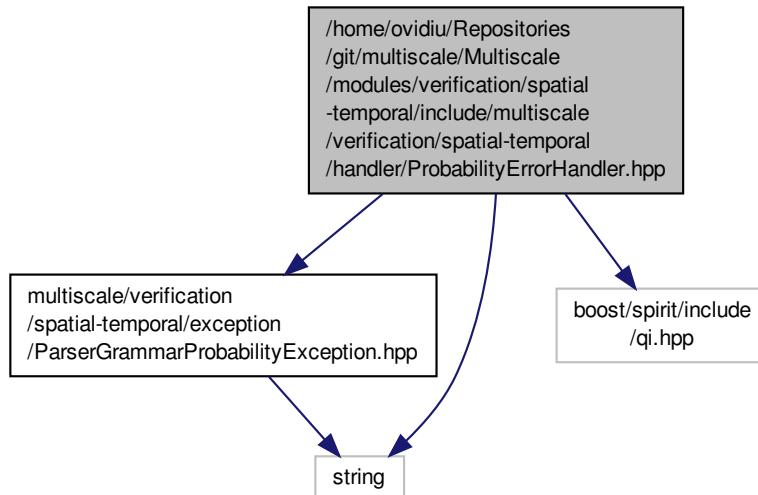
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.202 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/ProbabilityErrorHandler.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarProbabilityException.hpp"
#include <boost/spirit/include/qi.hpp>
#include <string>
```

Include dependency graph for ProbabilityErrorHandler.hpp:



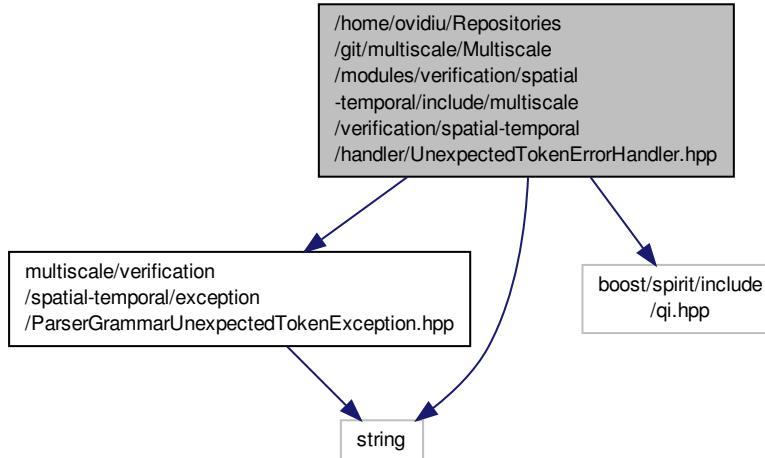
- struct `multiscale::verification::ProbabilityErrorHandler`  
*Structure for defining the error handler for invalid probability errors.*
- struct `multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >`  
*Structure for specifying the type of the result.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.203 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/UnexpectedErrorHandler.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/ParserGrammar.h"
#include "multiscale/verification/spatial-temporal/exception/UnexpectedTokenException.h"
#include <boost/spirit/include/qi.hpp>
#include <string>
Include dependency graph for UnexpectedErrorHandler.hpp:
```



## Classes

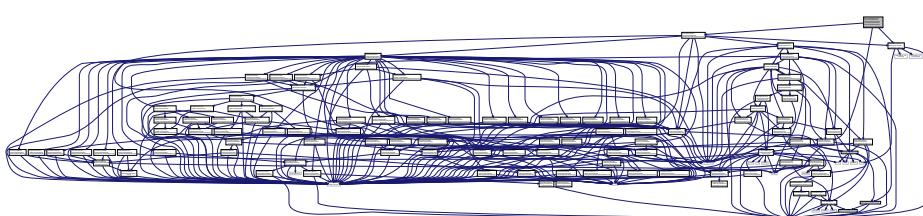
- struct `multiscale::verification::UnexpectedErrorHandler`  
*Structure for defining the error handler for unexpected token errors.*
- struct `multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename >`  
*Structure for specifying the type of the result.*

## Namespaces

- multiscale
- multiscale::verification

### 8.204 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp"
Include dependency graph for AbstractSyntaxTree.hpp:
```



## Classes

- class multiscale::verification::AbstractSyntaxTree

*Class used for representing an abstract syntax tree.*

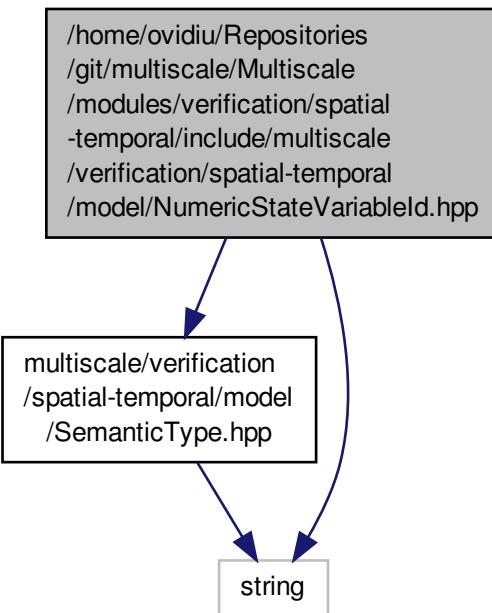
## Namespaces

- multiscale
- multiscale::verification

### 8.205 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/NumericStateVariableId.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include <string>
```

Include dependency graph for NumericStateVariableId.hpp:



## Classes

- class [multiscale::verification::NumericStateVariableId](#)

*Class for representing the identity (name, type) of a numeric state variable.*

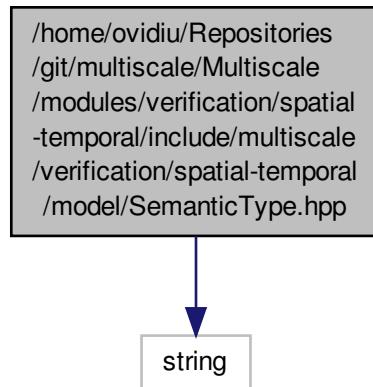
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.206 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SemanticType.hpp File Reference

```
#include <string>
```

Include dependency graph for SemanticType.hpp:



## Classes

- class [multiscale::verification::SemanticType](#)

*Enumeration for defining a semantic type.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.207 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialEntity.hpp

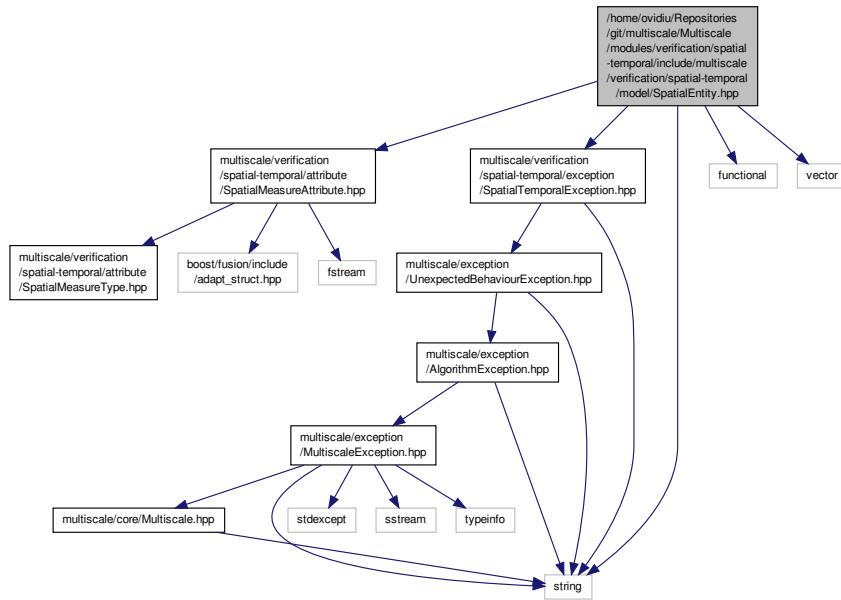
### File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include <functional>
#include <string>
#include <vector>

```

Include dependency graph for SpatialEntity.hpp:



## Classes

- class [multiscale::verification::SpatialEntity](#)

*Class for representing a pseudo-3D spatial entity.*

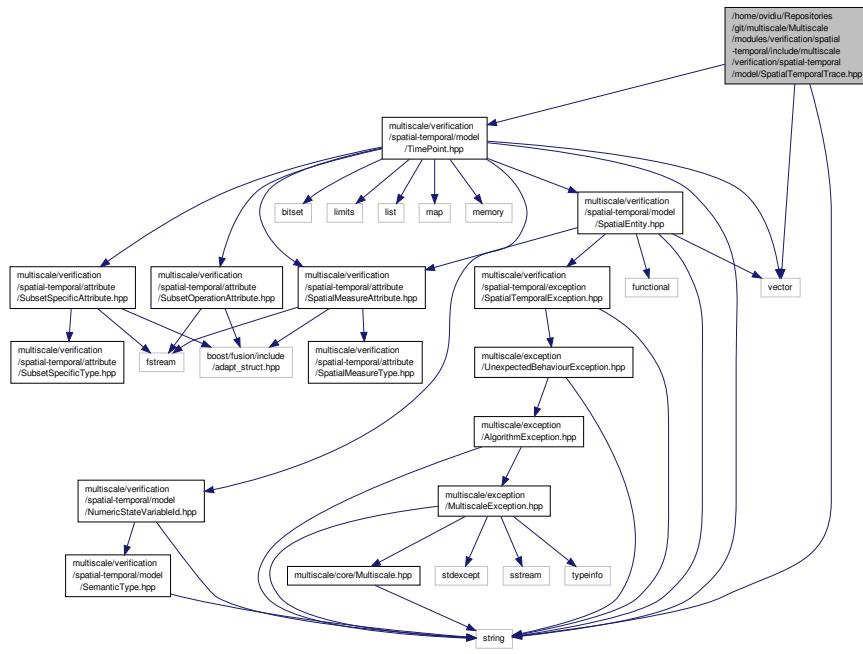
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.208 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/TimePoint.hpp"
#include <string>
#include <vector>
```

Include dependency graph for SpatialTemporalTrace.hpp:



## Classes

- class `multiscale::verification::SpatialTemporalTrace`

*Class for representing a spatial temporal trace.*

## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.209 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TimePoint.hpp File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/NumericStateVariable.hpp"

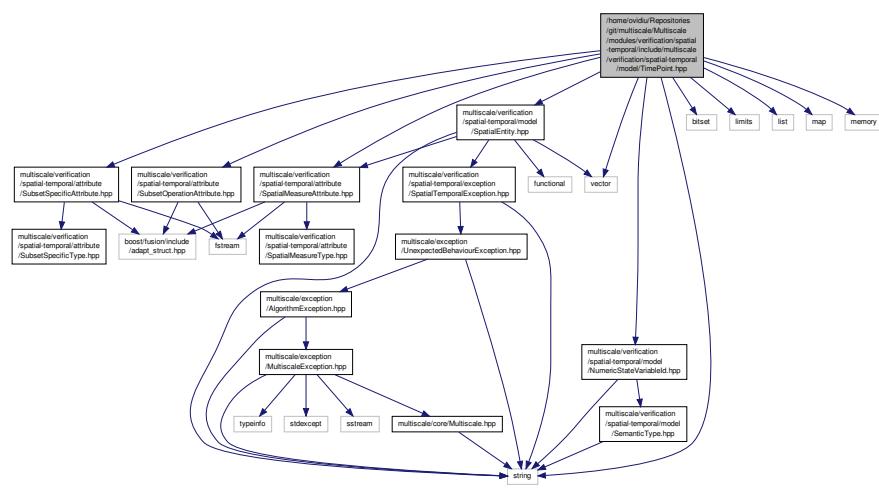
```

## Reference

Id.hpp"

```
#include "multiscale/verification/spatial-temporal/model/SpatialEntity.hpp"
#include <bitset>
#include <limits>
#include <list>
#include <map>
#include <memory>
#include <string>
#include <vector>
```

Include dependency graph for TimePoint.hpp:



## Classes

- class [multiscale::verification::TimePoint](#)

*Class for representing a timepoint.*

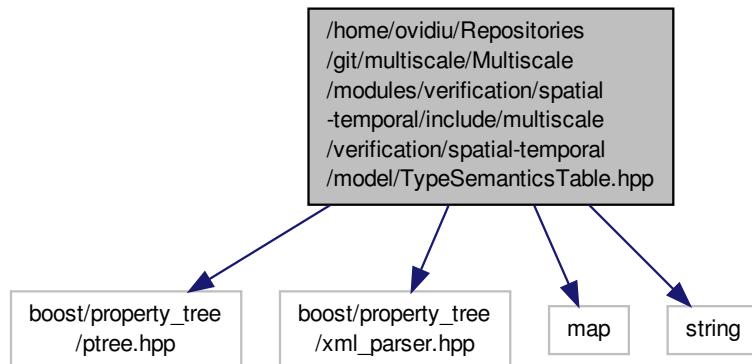
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.210 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp File Reference

```
#include <boost/property_tree/ptree.hpp>
#include <boost/property_tree/xml_parser.hpp>
#include <map>
#include <string>
```

Include dependency graph for TypeSemanticsTable.hpp:



## Classes

- class [multiscale::verification::TypeSemanticsTable](#)

*Class for defining a type semantics table.*

## Namespaces

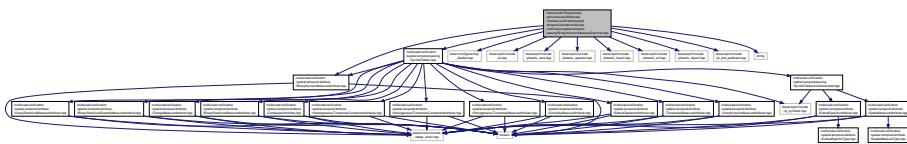
- [multiscale](#)
- [multiscale::verification](#)

## 8.211 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>

```



## Classes

- class multiscale::verification::BinaryNumericMeasureGrammar< Iterator >

*The grammar for parsing binary numeric measure statements.*

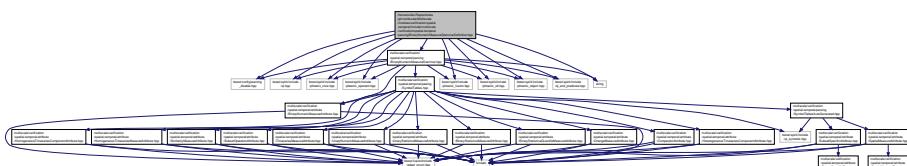
## Namespaces

- multiscale
  - multiscale::verification

8.212 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for BinaryNumericMeasureGrammarDefinition.hpp:



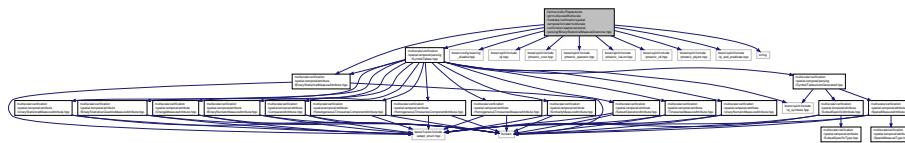
## Namespaces

- multiscale
  - multiscale::verification

## 8.213 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>

Include dependency graph for BinaryStatisticalMeasureGrammar.hpp:
```



### Classes

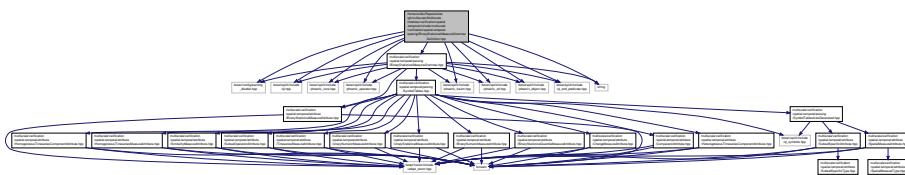
- class [multiscale::verification::BinaryStatisticalMeasureGrammar< Iterator >](#)  
*The grammar for parsing binary statistical measure statements.*

### Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.214 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```



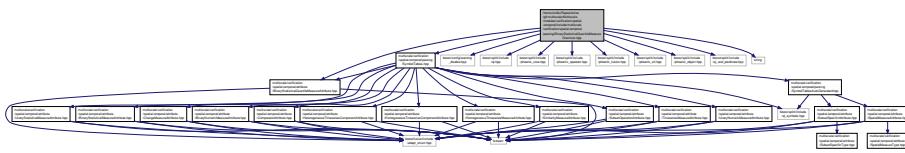
## Namespaces

- multiscale
- multiscale::verification

## 8.215 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for BinaryStatisticalQuantileMeasureGrammar.hpp:



## Classes

- class [multiscale::verification::BinaryStatisticalQuantileMeasureGrammar< Iterator >](#)  
*The grammar for parsing binary statistical quantile measure statements.*

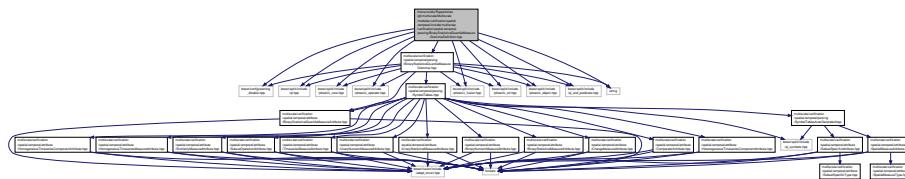
## Namespaces

- multiscale
- multiscale::verification

## 8.216 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for BinaryStatisticalQuantileMeasureGrammarDefinition.hpp:



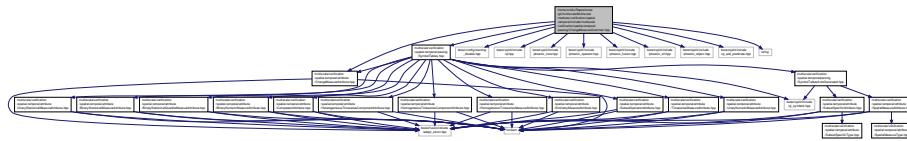
### Namespaces

- multiscale
- multiscale::verification

## 8.217 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for ChangeMeasureGrammar.hpp:



## Classes

- class multiscale::verification::ChangeMeasureGrammar< Iterator >

*The grammar for parsing change measure statements.*

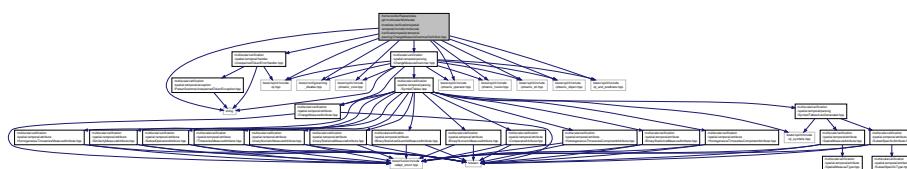
## Namespaces

- multiscale
- multiscale::verification

## 8.218 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for ChangeMeasureGrammarDefinition.hpp:



## Namespaces

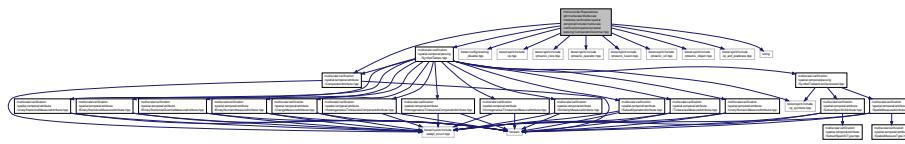
- multiscale
- multiscale::verification

## Variables

- phoenix::function  
 < UnexpectedErrorHandler >  
 const multiscale::verification::handleUnexpectedTokenError = UnexpectedErrorHandler()

## 8.219 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
Include dependency graph for ComparatorGrammar.hpp:
```



## Classes

- class multiscale::verification::ComparatorGrammar< Iterator >  
*The grammar for parsing comparator statements.*

## Namespaces

- multiscale
- multiscale::verification

## 8.220 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ComparatorGrammarDefinition.hpp File Reference

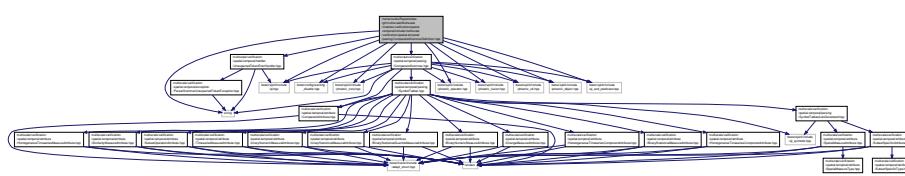
```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.hpp"
```

Reference

hpp"

```
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for ComparatorGrammarDefinition.hpp:



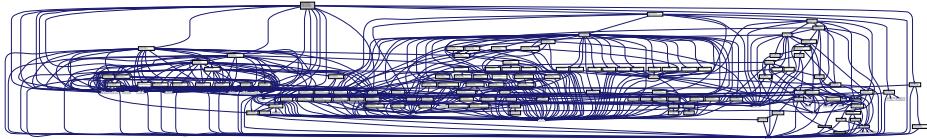
## Namespaces

- multiscale
- multiscale::verification

8.221 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/handler/ProbabilityErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for LogicPropertyGrammar.hpp:



## Classes

- class [multiscale::verification::LogicPropertyGrammar< Iterator >](#)

*The grammar for parsing logic properties.*

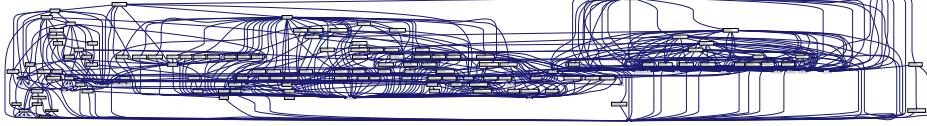
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.222 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for LogicPropertyGrammarDefinition.hpp:



## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

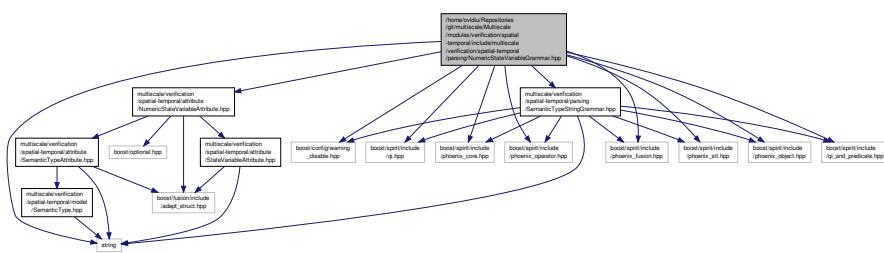
- phoenix::function
 

```
< ProbabilityErrorHandler >
const multiscale::verification::handleProbabilityError = ProbabilityErrorHandler()
```

## 8.223 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericStateVariableAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for NumericStateVariableGrammar.hpp:



## Classes

- class **multiscale::verification::NumericStateVariableGrammar< Iterator >**  
*The grammar for parsing numeric state variable statements.*

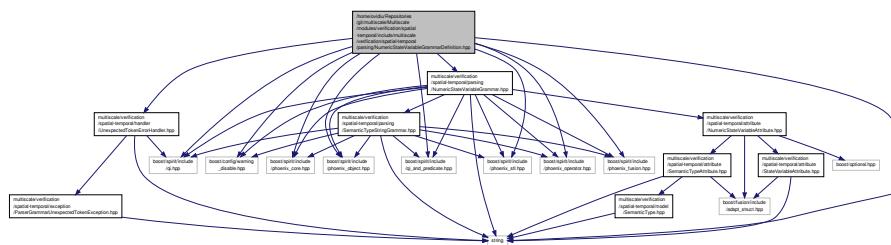
## Namespaces

- multiscale
- multiscale::verification

## 8.224 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedToken.hpp"
```

```
ErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
Include dependency graph for NumericStateVariableGrammarDefinition.hpp:
```

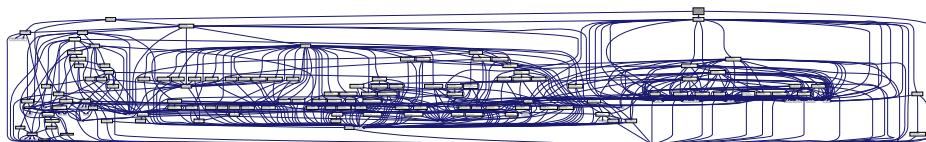


## Namespaces

- multiscale
  - multiscale::verification

8.225 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/Parser.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammar.hpp"
#include <string>
Include dependency graph for Parser.hpp:
```



## Classes

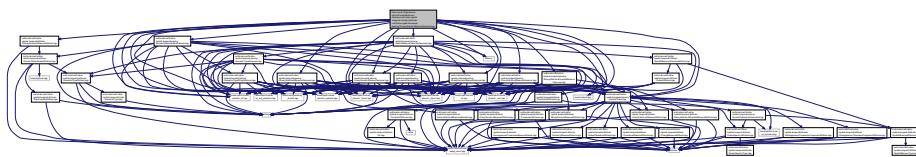
- class multiscale::verification::Parser

*Class used for parsing (P)BLSTL logical queries.*

- multiscale
- multiscale::verification

## 8.226 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/PrimaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <memory>
#include <string>
Include dependency graph for PrimaryNumericMeasureGrammar.hpp:
```



## Classes

- class [multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >](#)  
*The grammar for parsing spatial measure collection statements.*
- singleton [multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >](#)  
*The grammar for parsing primary numeric measure statements.*

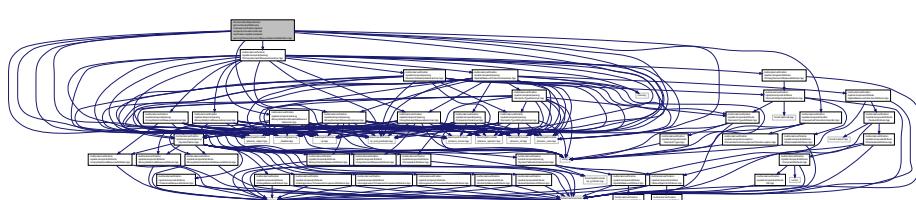
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### 8.227 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for PrimaryNumericMeasureGrammarDefinition.hpp:



## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### 8.228 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SemanticType
```

**8.229 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammarDefinition.hpp**

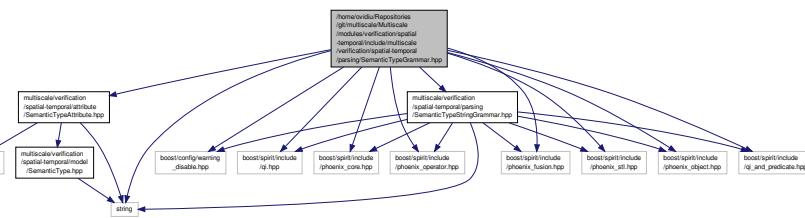
**File Reference**

1269

StringGrammar.hpp"

```
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for SemanticTypeGrammar.hpp:



## Classes

- class [multiscale::verification::SemanticTypeGrammar< Iterator >](#)

*The grammar for parsing semantic type statements.*

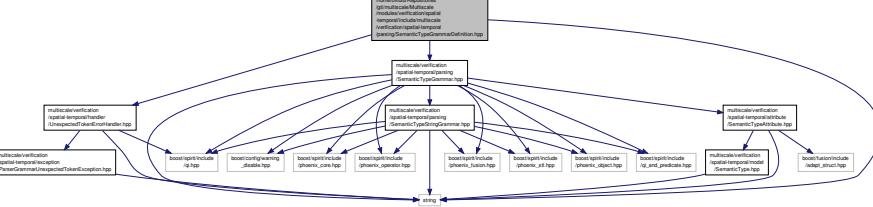
## Namespaces

- multiscale
- multiscale::verification

**8.229 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammarDefinition.hpp File Reference**

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammar.hpp"
#include <string>
```

Include dependency graph for SemanticTypeGrammarDefinition.hpp:



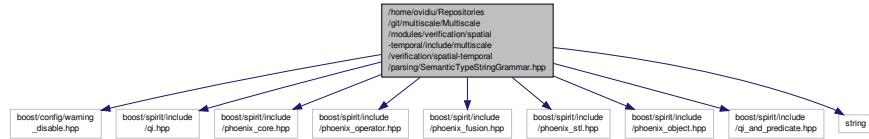
## Namespaces

- multiscale
- multiscale::verification

### 8.230 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammar.hpp File Reference

```
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for SemanticTypeStringGrammar.hpp:



## Classes

- class multiscale::verification::SemanticTypeStringGrammar< Iterator >

*The grammar for parsing semantic type string statements.*

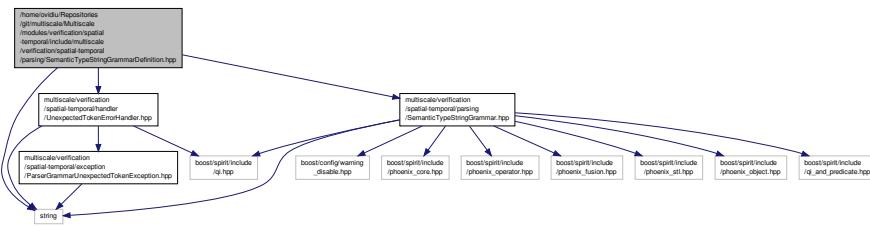
## Namespaces

- multiscale
- multiscale::verification

### 8.231 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammar.hpp"
#include <string>
```

Include dependency graph for SemanticTypeStringGrammarDefinition.hpp:



## Namespaces

- multiscale
- multiscale::verification

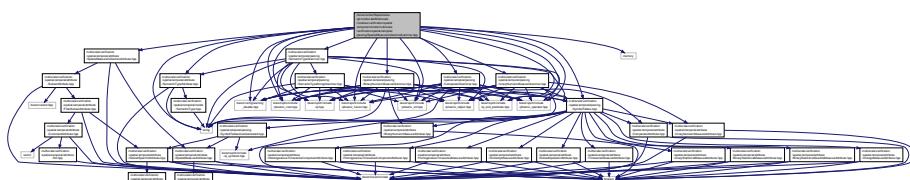
## 8.232 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp File Reference

```

#include "multiscale/verification/spatial-temporal/attribute/SemanticTypeAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureCollectionAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <memory>
#include <string>

```

Include dependency graph for SpatialMeasureCollectionGrammar.hpp:



## Classes

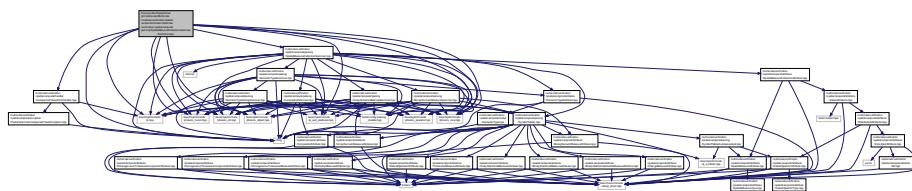
- singleton [multiscale::verification::PrimaryNumericMeasureGrammar< Iterator >](#)  
*The grammar for parsing primary numeric measure statements.*
- class [multiscale::verification::SpatialMeasureCollectionGrammar< Iterator >](#)  
*The grammar for parsing spatial measure collection statements.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### 8.233 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
Include dependency graph for SpatialMeasureCollectionGrammarDefinition.hpp:
```



## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

### 8.234 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalAttribute.hpp"
```

```
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatistical-
QuantileMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasure-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Comparator-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Heterogeneous-
TimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Homogeneous-
TimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Homogeneous-
TimeseriesMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Similarity-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetOperation-
Attribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/Timeseries-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumeric-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatistical-
MeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables-
AutoGenerated.hpp"
#include <boost/spirit/include/qi_symbols.hpp>
```

Include dependency graph for SymbolTables.hpp:



## Classes

- struct [multiscale::verification::BinaryNumericMeasureTypeParser](#)  
*Symbol table and parser for the binary numeric measure type.*
- struct [multiscale::verification::BinaryStatisticalMeasureTypeParser](#)  
*Symbol table and parser for the binary statistical measure type.*
- struct [multiscale::verification::BinaryStatisticalQuantileMeasureTypeParser](#)  
*Symbol table and parser for the binary statistical quantile measure type.*
- struct [multiscale::verification::ChangeMeasureTypeParser](#)  
*Symbol table and parser for the change measure type.*
- struct [multiscale::verification::ComparatorNonEqualTypeParser](#)  
*Symbol table and parser for the comparator type which does not accept the "=" symbol.*
- struct [multiscale::verification::ComparatorTypeParser](#)  
*Symbol table and parser for the comparator type.*
- struct [multiscale::verification::HeterogeneousTimeseriesComponentTypeParser](#)  
*Symbol table and parser for the heterogeneous timeseries component type.*
- struct [multiscale::verification::HomogeneousTimeseriesComponentTypeParser](#)  
*Symbol table and parser for the homogeneous timeseries component type.*
- struct [multiscale::verification::HomogeneousTimeseriesMeasureTypeParser](#)  
*Symbol table and parser for the homogeneous timeseries measure type.*
- struct [multiscale::verification::SimilarityMeasureTypeParser](#)

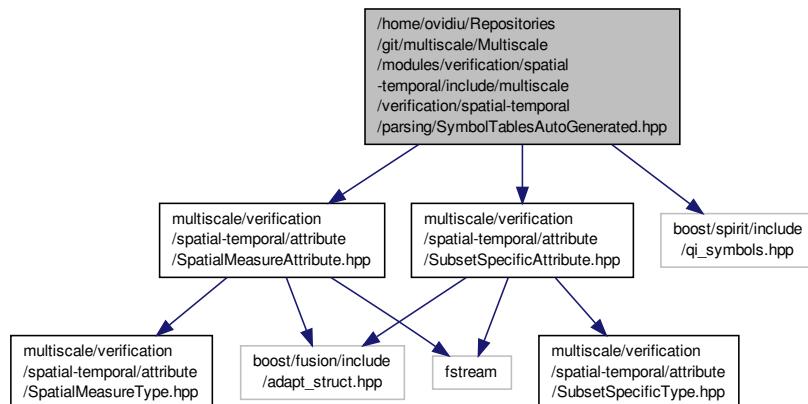
- struct **multiscale::verification::SubsetOperationTypeParser**  
*Symbol table and parser for the subset operation type.*
- struct **multiscale::verification::TimeseriesMeasureTypeParser**  
*Symbol table and parser for the timeseries measure type.*
- struct **multiscale::verification::UnaryNumericMeasureTypeParser**  
*Symbol table and parser for the unary numeric measure type.*
- struct **multiscale::verification::UnaryStatisticalMeasureTypeParser**  
*Symbol table and parser for the unary statistical measure type.*

## Namespaces

- multiscale
- multiscale::verification

## 8.235 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTablesAutoGenerated.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
#include <boost/spirit/include/qi_symbols.hpp>
Include dependency graph for SymbolTablesAutoGenerated.hpp:
```



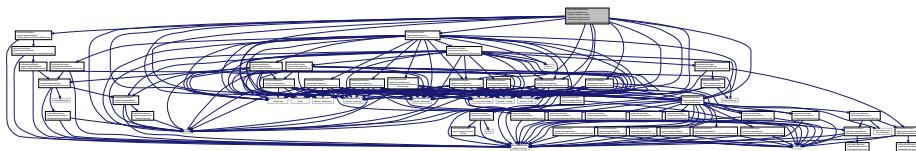
## Classes

- struct **multiscale::verification::SpatialMeasureTypeParser**  
*Symbol table and parser for the spatial measure type.*
- struct **multiscale::verification::SubsetSpecificTypeParser**  
*Symbol table and parser for a specific subset type.*

- multiscale
- multiscale::verification

## 8.236 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericCollectionAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
Include dependency graph for TemporalNumericCollectionGrammar.hpp:
```



## Classes

- class multiscale::verification::TemporalNumericCollectionGrammar< Iterator >

*The grammar for parsing temporal numeric collection statements.*

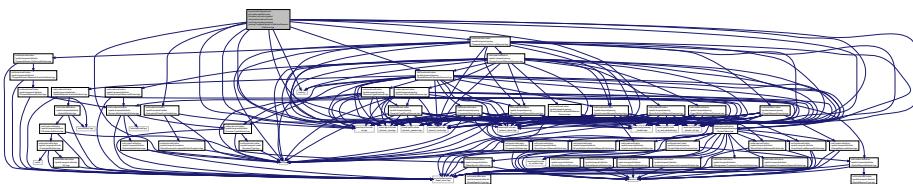
## Namespaces

- multiscale
- multiscale::verification

## 8.237 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <memory>
#include <string>
```

Include dependency graph for TemporalNumericCollectionGrammarDefinition.hpp:



### Namespaces

- multiscale
- multiscale::verification

## 8.238 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumeric
```

## 8.239 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammar

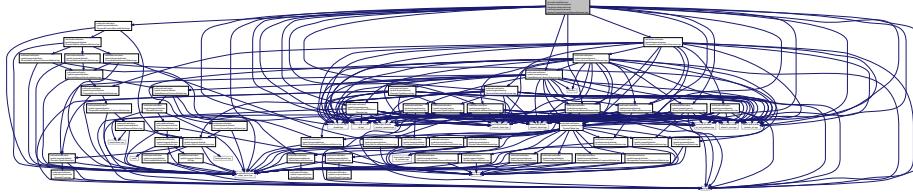
### Definition.hpp File

Reference

1277

```
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for TemporalNumericMeasureGrammar.hpp:



### Classes

- class multiscale::verification::TemporalNumericMeasureGrammar< Iterator >

*The grammar for parsing temporal numeric measure statements.*

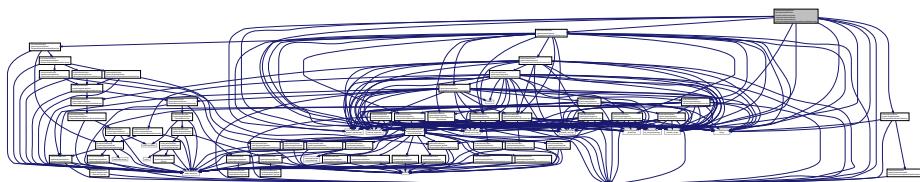
### Namespaces

- multiscale
- multiscale::verification

## 8.239 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for TemporalNumericMeasureGrammarDefinition.hpp:



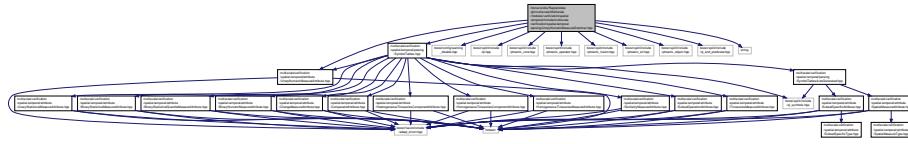
## Namespaces

- multiscale
- multiscale::verification

## 8.240 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for UnaryNumericMeasureGrammar.hpp:



## Classes

- class [multiscale::verification::UnaryNumericMeasureGrammar< Iterator >](#)  
*The grammar for parsing unary numeric measure statements.*

## Namespaces

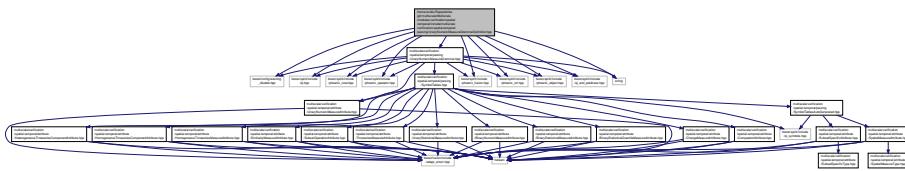
- multiscale
- multiscale::verification

8.241 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar  
Definition.hpp File Reference

8.241 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammarDefinition.hpp File Reference [279]

```
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for UnaryNumericMeasureGrammarDefinition.hpp:



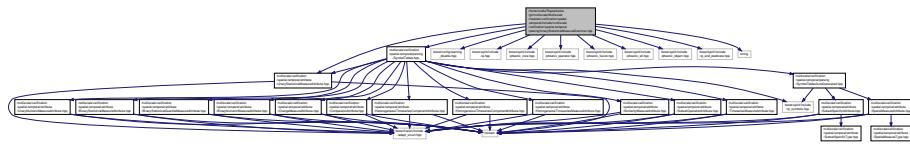
## Namespaces

- multiscale
- multiscale::verification

8.242 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for UnaryStatisticalMeasureGrammar.hpp:



## Classes

- class multiscale::verification::UnaryStatisticalMeasureGrammar< Iterator >

## *The grammar for parsing unary statistical measure statements.*

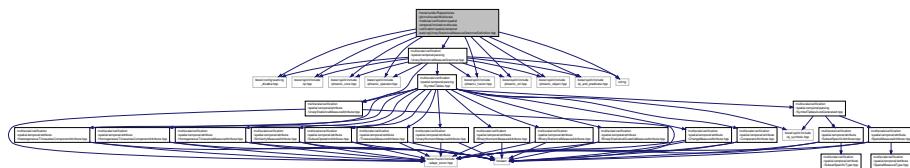
## Namespaces

- multiscale
  - multiscale::verification

8.243 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammarDefinition.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammar.hpp"
#include <boost/config/warning_disable.hpp>
#include <boost/spirit/include/qi.hpp>
#include <boost/spirit/include/phoenix_core.hpp>
#include <boost/spirit/include/phoenix_operator.hpp>
#include <boost/spirit/include/phoenix_fusion.hpp>
#include <boost/spirit/include/phoenix_stl.hpp>
#include <boost/spirit/include/phoenix_object.hpp>
#include <boost/spirit/include/qi_and_predicate.hpp>
#include <string>
```

Include dependency graph for UnaryStatisticalMeasureGrammarDefinition.hpp:

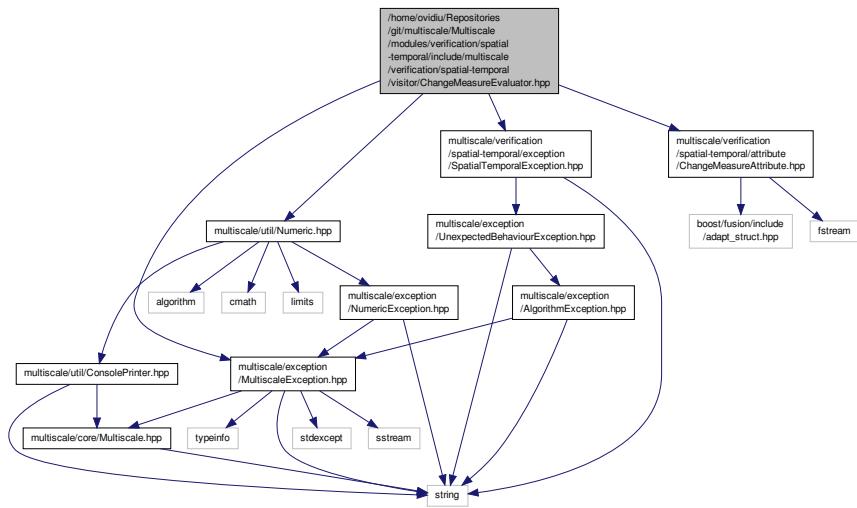


## Namespaces

- multiscale
  - multiscale::verification

## 8.244 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ChangeMeasureEvaluator.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
Include dependency graph for ChangeMeasureEvaluator.hpp:
```



## Classes

- class [multiscale::verification::ChangeMeasureEvaluator](#)

*Class for evaluating change measure expressions.*

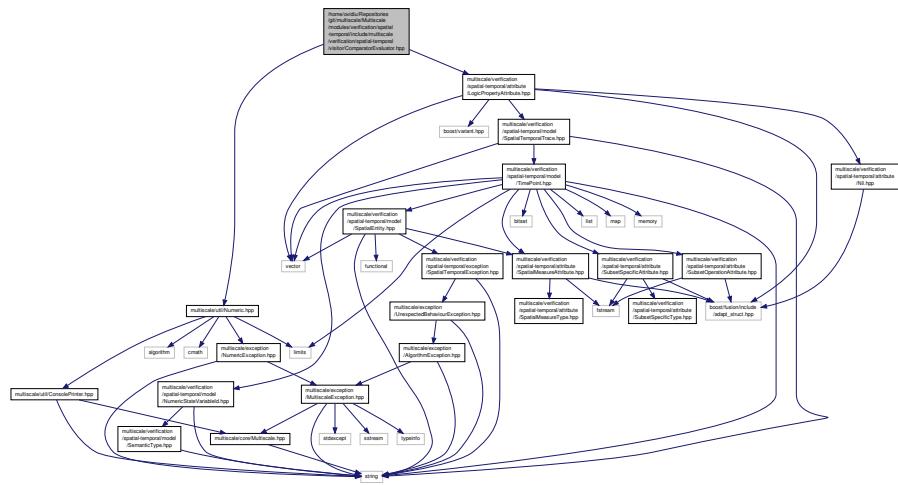
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.245 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp File Reference

```
#include "multiscale/util/Numeric.hpp"
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
```

Include dependency graph for ComparatorEvaluator.hpp:



## Classes

- class `multiscale::verification::ComparatorEvaluator`

*Class for evaluating comparison expressions.*

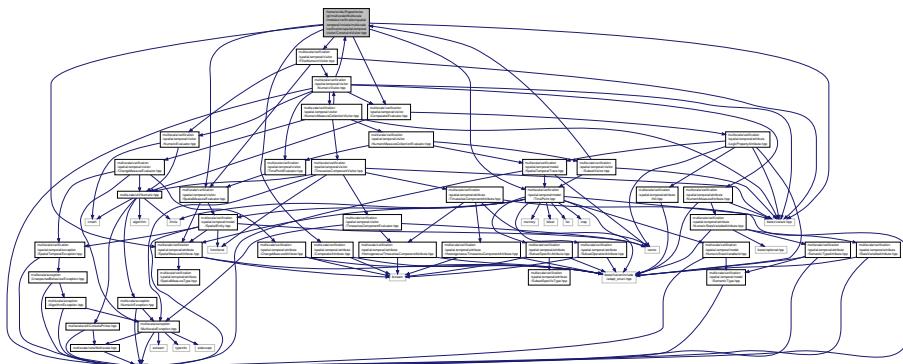
## Namespaces

- multiscale
  - multiscale::verification

**8.246** /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ConstraintVisitor.hpp  
**File Reference**

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/TimePoint.hpp"
#include "multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/FilterNumericVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/SpatialMeasureEvaluator.hpp"
#include <boost/variant.hpp>
```

Include dependency graph for ConstraintVisitor.hpp:



## Classes

- class [multiscale::verification::ConstraintVisitor](#)

*Class used to evaluate constraints.*

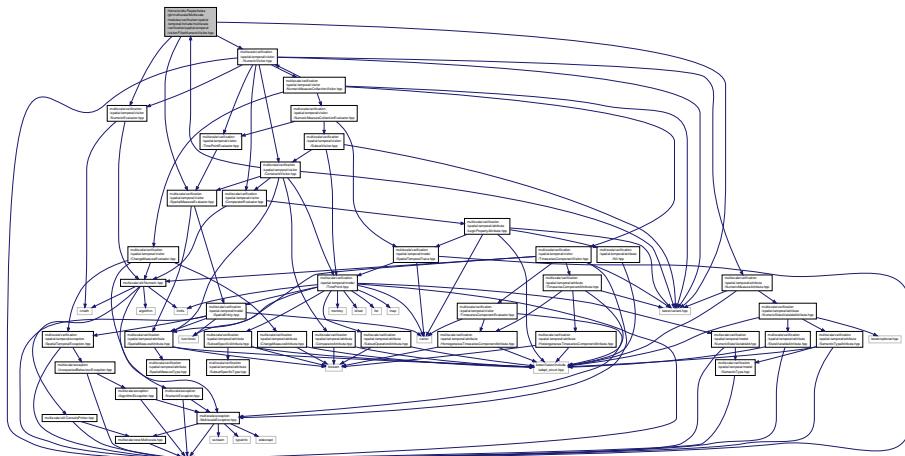
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.247 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/FilterNumericVisitor.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/SpatialMeasureEvaluator.hpp"
#include <boost/variant.hpp>
```

Include dependency graph for FilterNumericVisitor.hpp:



## Classes

- class [multiscale::verification::FilterNumericVisitor](#)

*Class for evaluating filter numeric measures.*

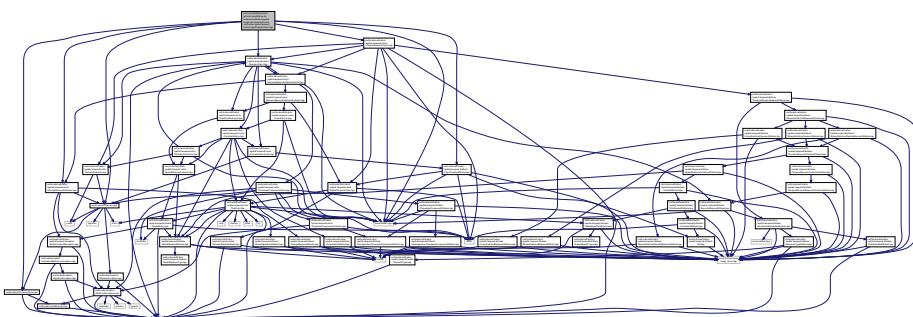
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.248 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/LogicPropertyAttributeVisitor.hpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/verification/spatial-temporal/attribute/LogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/visitor/ChangeMeasureEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/TemporalNumericVisitor.hpp"
#include <boost/variant.hpp>
```

Include dependency graph for LogicPropertyVisitor.hpp:



## Classes

- class [multiscale::verification::LogicPropertyVisitor](#)

*Class used to evaluate logic properties.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

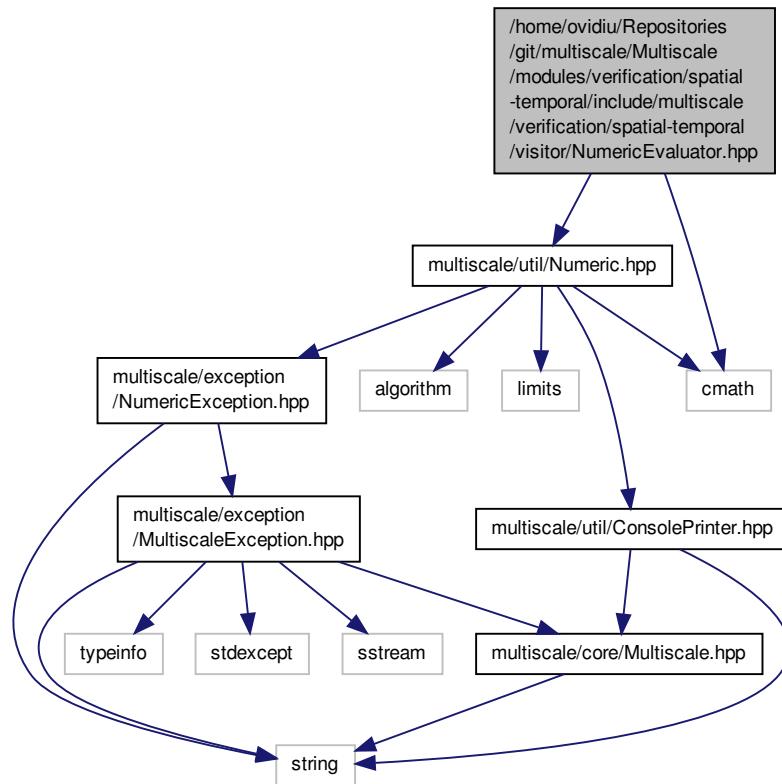
## Variables

- static const std::string [multiscale::verification::WRN\\_LOGIC\\_PROPERTY\\_EVAL\\_FALSE](#) = "The enclosing logic property was evaluated to the default value \"false\"."
- static const std::string [multiscale::verification::WRN\\_OUTPUT\\_SEPARATOR](#) = " "

## 8.249 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp File Reference

```
#include "multiscale/util/Numeric.hpp"
#include <cmath>
```

Include dependency graph for NumericEvaluator.hpp:



## Classes

- class `multiscale::verification::NumericEvaluator`  
*Class for evaluating numeric expressions.*

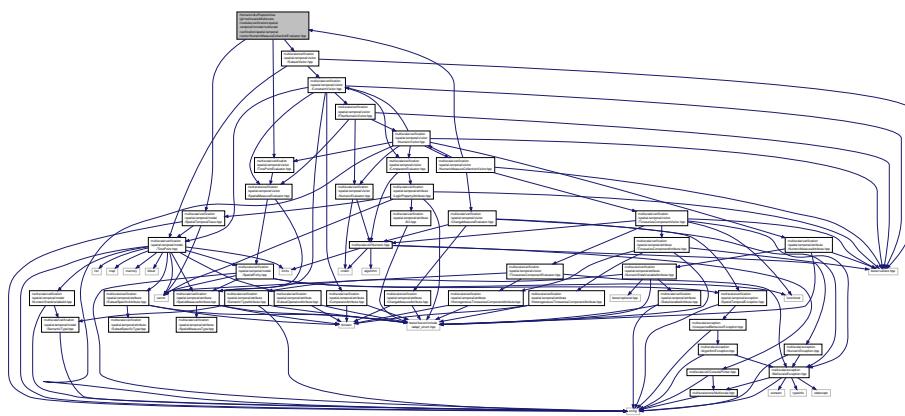
## Namespaces

- `multiscale`
- `multiscale::verification`

## 8.250 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionEvaluator.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include "multiscale/verification/spatial-temporal/visitor/SubsetVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/TimePointEvaluator.hpp"
```

Include dependency graph for NumericMeasureCollectionEvaluator.hpp:



## Classes

- class [multiscale::verification::NumericMeasureCollectionEvaluator](#)

*Class used to evaluate numeric measure collections.*

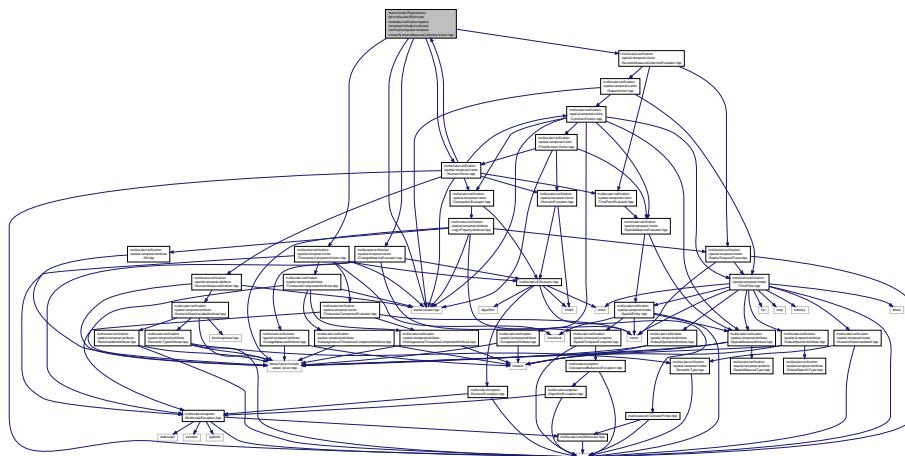
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.251 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionVisitor.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/visitor/ChangeMeasureEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/TimeseriesComponentVisitor.hpp"
#include <boost/variant.hpp>
#include "multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp"
```

Include dependency graph for NumericMeasureCollectionVisitor.hpp:



## Classes

- class [multiscale::verification::NumericMeasureCollectionVisitor](#)

*Class for evaluating numeric measure collections.*

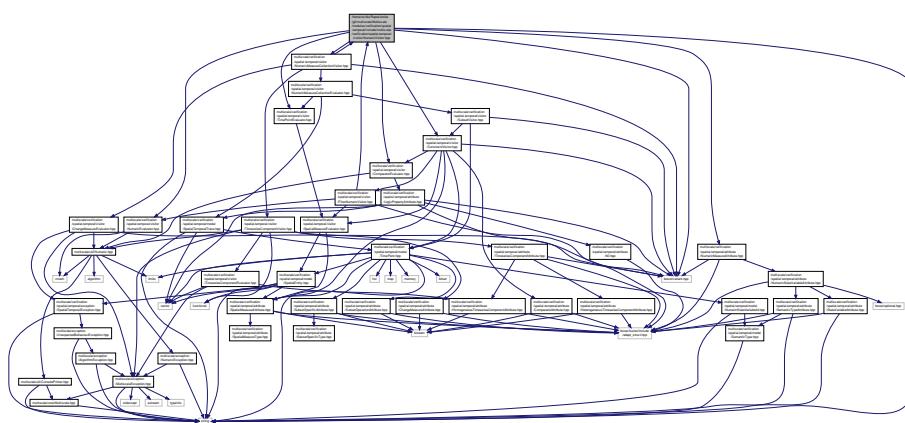
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.252 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/NumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp"
#include "multiscale/verification/spatial-temporal/visitor/TimePointEvaluator.hpp"
#include <boost/variant.hpp>
#include <string>
#include "multiscale/verification/spatial-temporal/visitor/ConstraintVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionVisitor.hpp"
```

Include dependency graph for NumericVisitor.hpp:



## Classes

- class [multiscale::verification::NumericVisitor](#)

*Class for evaluating numeric measures.*

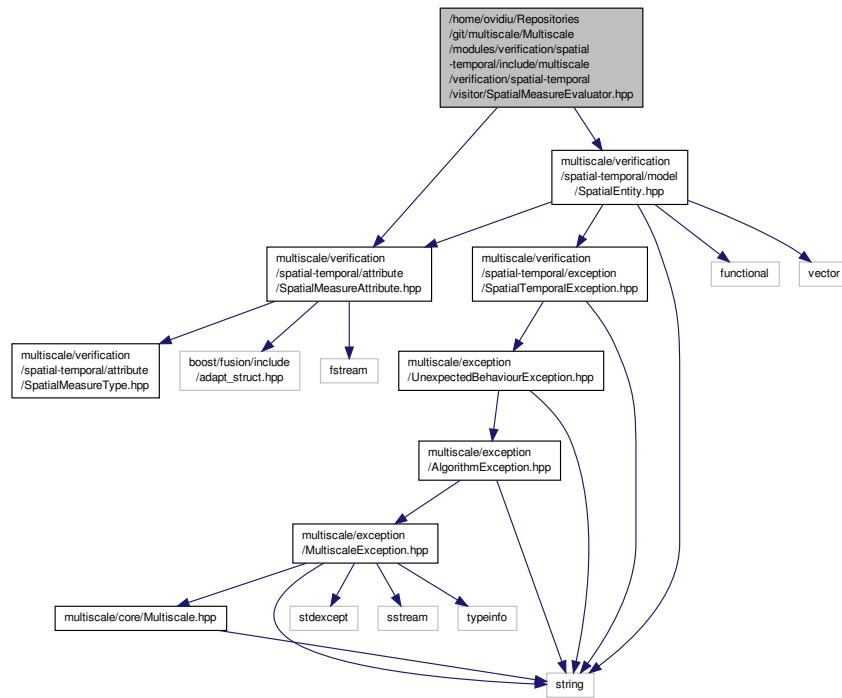
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.253 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SpatialMeasureEvaluator.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialEntity.hpp"
```

Include dependency graph for SpatialMeasureEvaluator.hpp:



## Classes

- class [multiscale::verification::SpatialMeasureEvaluator](#)

*Class for evaluating spatial measures.*

## Namespaces

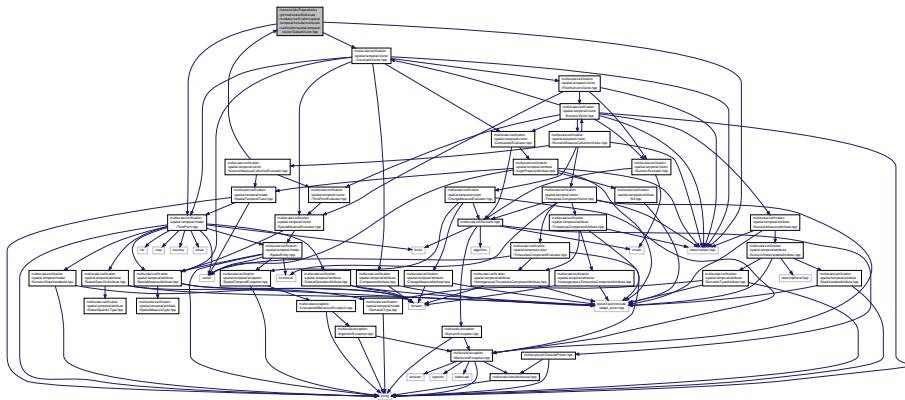
- [multiscale](#)
- [multiscale::verification](#)

## 8.254 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SubsetVisitor.hpp

### File Reference

```
#include "multiscale/verification/spatial-temporal/model/TimePoint.hpp"
#include <boost/variant.hpp>
#include "multiscale/verification/spatial-temporal/visitor/ConstraintVisitor.hpp"
```

Include dependency graph for SubsetVisitor.hpp:



## Classes

- class [multiscale::verification::SubsetVisitor](#)

*Class used to evaluate subsets.*

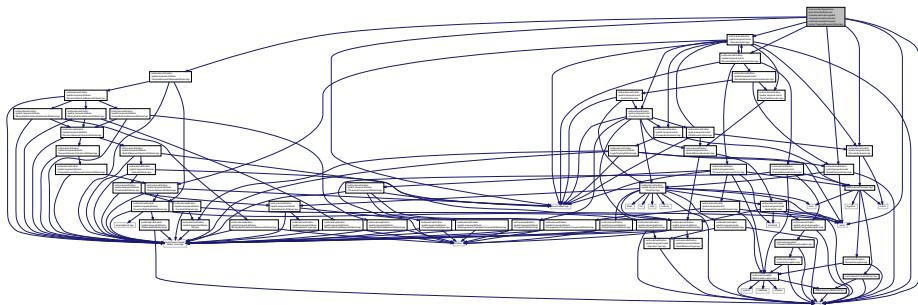
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.255 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TemporalNumericVisitor.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TemporalNumericMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp"
#include <boost/variant.hpp>
#include <string>
#include <vector>
#include "multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp"
#include "multiscale/verification/spatial-temporal/visitor/NumericMeasureCollectionVisitor.hpp"
```

Include dependency graph for TemporalNumericVisitor.hpp:



## Classes

- class [multiscale::verification::TemporalNumericVisitor](#)

*Class for evaluating temporal numeric measures.*

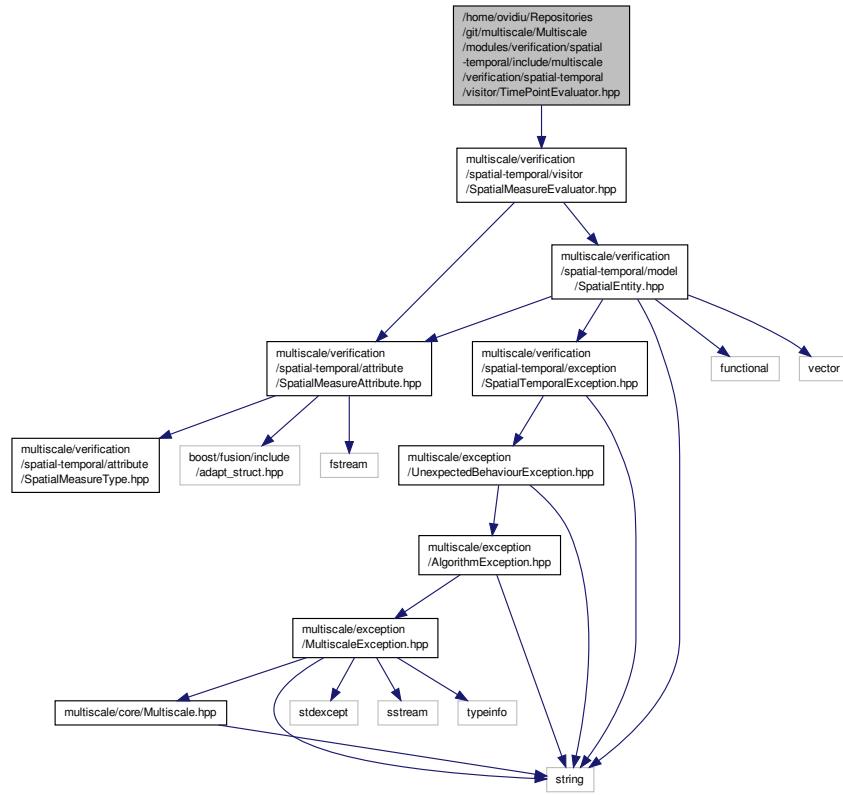
## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

## 8.256 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimePointEvaluator.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/visitor/SpatialMeasureEvaluator.hpp"
```

Include dependency graph for TimePointEvaluator.hpp:



## Classes

- class multiscale::verification::TimePointEvaluator

*Class used to evaluate timepoints.*

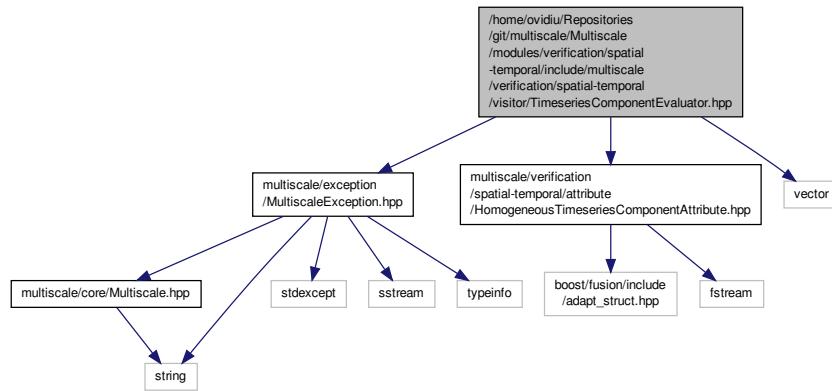
## Namespaces

- multiscale
- multiscale::verification

## 8.257 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp"
#include <vector>
```

Include dependency graph for TimeseriesComponentEvaluator.hpp:



## Classes

- class `multiscale::verification::TimeseriesComponentEvaluator`  
*Class for evaluating timeseries components.*
- class `multiscale::verification::TimeseriesComponentEvaluator::HomogeneousComponentEvaluator< Relation >`
- class `multiscale::verification::TimeseriesComponentEvaluator::UniformHomogeneousComponentEvaluator< Relation >`

## Namespaces

- `multiscale`
- `multiscale::verification`

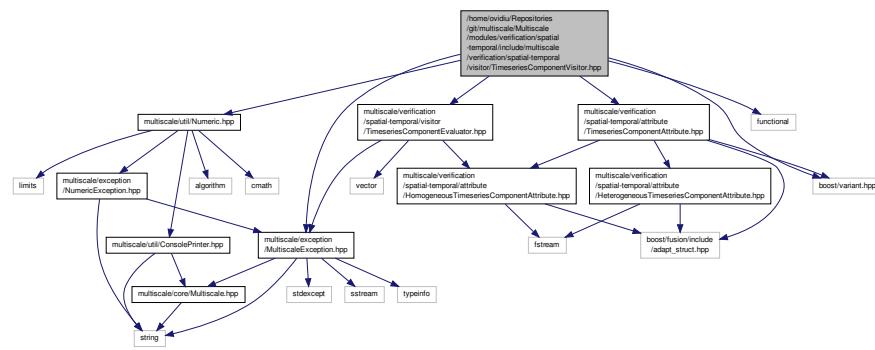
## 8.258 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimeseriesComponentVisitor.hpp File Reference

```

#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/verification/spatial-temporal/attribute/TimeseriesComponentAttribute.hpp"
#include "multiscale/verification/spatial-temporal/visitor/TimeseriesComponentEvaluator.hpp"
#include <boost/variant.hpp>
#include <functional>

```

Include dependency graph for TimeseriesComponentVisitor.hpp:



## Classes

- class [multiscale::verification::TimeseriesComponentVisitor](#)

*Class for evaluating timeseries components.*

## Namespaces

- [multiscale](#)
- [multiscale::verification](#)

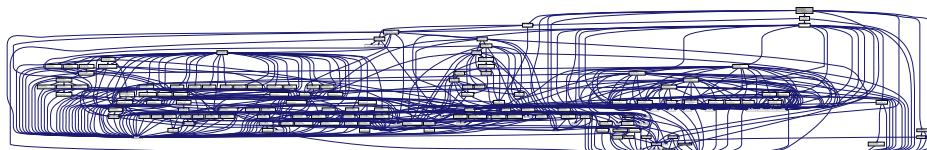
## 8.259 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/LogicPropertyDataReaderSample.cpp File Reference

```

#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/data/LogicPropertyDataReader.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <iostream>
#include <string>
#include <vector>

```

Include dependency graph for LogicPropertyDataReaderSample.cpp:



## Functions

- void [printParsingResult](#) ([Parser](#) &parser, [AbstractSyntaxTree](#) &parsingResult)
- void [printQueries](#) (const std::vector< std::string > &queries)

- void [readQueriesFromFile](#) (const std::string &path)
- int [main](#) (int argc, char \*\*argv)

### 8.259.1 Function Documentation

#### 8.259.1.1 int main ( int *argc*, char \*\* *argv* )

Definition at line 61 of file LogicPropertyDataReaderSample.cpp.

References multiscale::EXEC\_SUCCESS\_CODE, and [readQueriesFromFile\(\)](#).

#### 8.259.1.2 void printParsingResult ( Parser & *parser*, AbstractSyntaxTree & *parsingResult* )

Definition at line 17 of file LogicPropertyDataReaderSample.cpp.

References multiscale::verification::Parser::parse(), and multiscale::ExceptionHandler::printDetailedErrorMessage().

Referenced by [printQueries\(\)](#).

#### 8.259.1.3 void printQueries ( const std::vector< std::string > & *queries* )

Definition at line 30 of file LogicPropertyDataReaderSample.cpp.

References [printParsingResult\(\)](#), and multiscale::verification::Parser::setLogicalQuery().

Referenced by [readQueriesFromFile\(\)](#).

#### 8.259.1.4 void readQueriesFromFile ( const std::string & *path* )

Definition at line 48 of file LogicPropertyDataReaderSample.cpp.

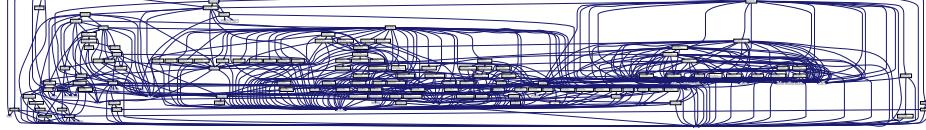
References multiscale::ExceptionHandler::printDetailedErrorMessage(), [printQueries\(\)](#), and multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile().

Referenced by [main\(\)](#).

## 8.260 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserEvaluationSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/model/Cluster.hpp"
#include "multiscale/verification/spatial-temporal/model/Region.hpp"
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <iostream>
```

Include dependency graph for ParserEvaluationSample.cpp:



- void [initialiseTrace \( SpatialTemporalTrace &trace\)](#)
- int [main \(int argc, char \\*\\*argv\)](#)

## 8.260.1 Function Documentation

### 8.260.1.1 void initialiseTrace ( SpatialTemporalTrace & trace )

Definition at line 25 of file ParserEvaluationSample.cpp.

References multiscale::verification::SpatialTemporalTrace::addTimePoint(), multiscale::verification::Angle, multiscale::verification::Area, multiscale::verification::CentroidX, multiscale::verification::CentroidY, multiscale::verification::CircleMeasure, multiscale::verification::SpatialTemporalTrace::clear(), multiscale::verification::Clusteredness, multiscale::verification::Clusters, multiscale::verification::SemanticType::DEFAULT\_VALUE, multiscale::verification::Density, multiscale::verification::DistanceFromOrigin, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, multiscale::verification::Regions, and multiscale::verification::TriangleMeasure.

Referenced by main().

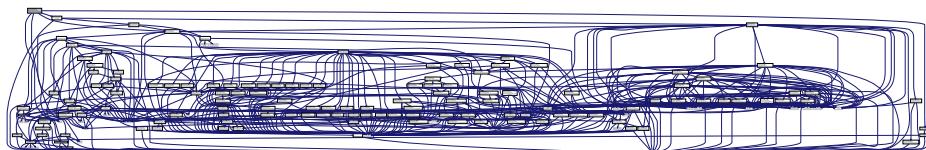
### 8.260.1.2 int main ( int argc, char \*\* argv )

Definition at line 118 of file ParserEvaluationSample.cpp.

References multiscale::verification::AbstractSyntaxTree::evaluate(), multiscale::EXEC\_ERR\_CODE, initialiseTrace(), multiscale::verification::Parser::parse(), and multiscale::ExceptionHandler::printDetailedErrorMessage().

## 8.261 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <iostream>
Include dependency graph for ParserSample.cpp:
```



## Functions

- int [main \(int argc, char \\*\\*argv\)](#)

## 8.261.1 Function Documentation

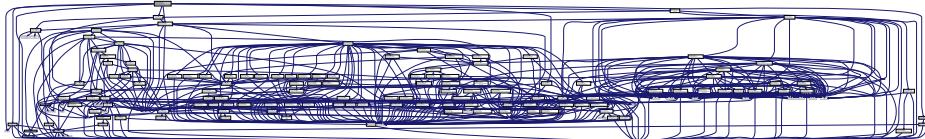
### 8.261.1.1 int main ( int argc, char \*\* argv )

Definition at line 13 of file ParserSample.cpp.

References multiscale::EXEC\_ERR\_CODE, multiscale::verification::Parser::parse(), and multiscale::ExceptionHandler::printDetailedErrorMessage().

## 8.262 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/PatternAnalysisSample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/data/TemporalDataReader.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <iostream>
Include dependency graph for PatternAnalysisSample.cpp:
```



### Functions

- int [analysePatterns](#) (const std::string &inputFilePath)
- int [main](#) (int argc, char \*\*argv)

#### 8.262.1 Function Documentation

##### 8.262.1.1 int analysePatterns ( const std::string & *inputFilePath* )

Definition at line 16 of file PatternAnalysisSample.cpp.

References multiscale::verification::AbstractSyntaxTree::evaluate(), multiscale::EXEC\_ERR\_CODE, multiscale::verification::Parser::parse(), multiscale::ExceptionHandler::printDetailedErrorMessage(), and multiscale::verification::TemporalDataReader::read().

Referenced by main().

##### 8.262.1.2 int main ( int *argc*, char \*\* *argv* )

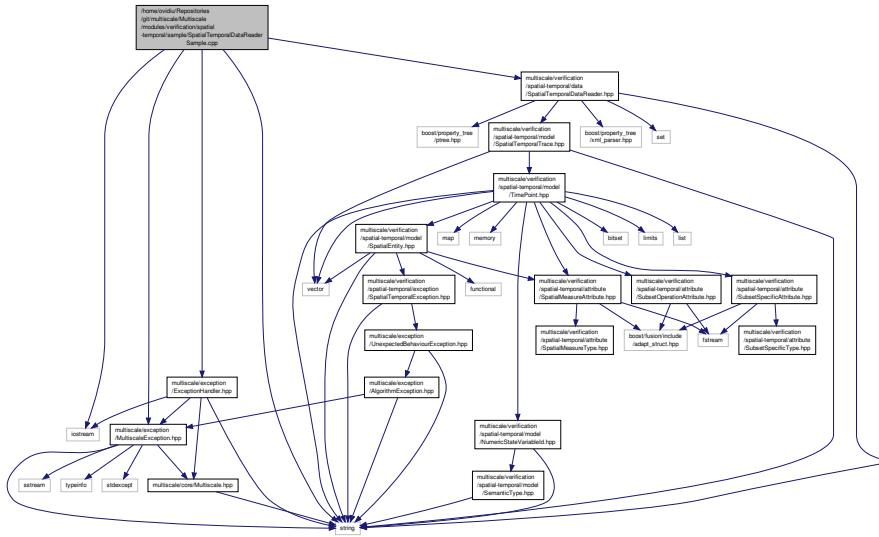
Definition at line 67 of file PatternAnalysisSample.cpp.

References analysePatterns(), multiscale::EXEC\_ERR\_CODE, and multiscale::ExceptionHandler::printDetailedErrorMessage().

## 8.263 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/SpatialTemporalDataReaderSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
```

```
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp"
#include <iostream>
#include <string>
Include dependency graph for SpatialTemporalDataReaderSample.cpp:
```



## Functions

- void [printSpatialEntities](#) (TimePoint &timePoint, const SubsetSpecificType &spatialEntityType)
- void [printTimePoint](#) (TimePoint &timePoint)
- void [printTrace](#) (SpatialTemporalTrace &trace)
- void [readValidXmlFiles](#) (SpatialTemporalDataReader &reader)
- void [readValidXmlFilesFromFolder](#) (const std::string &path)
- int [main](#) (int argc, char \*\*argv)

### 8.263.1 Function Documentation

#### 8.263.1.1 int main ( int argc, char \*\* argv )

Definition at line 67 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, and [readValidXmlFilesFromFolder\(\)](#).

#### 8.263.1.2 void printSpatialEntities ( TimePoint & timePoint, const SubsetSpecificType & spatialEntityType )

Definition at line 13 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), and multiscale::verification::TimePoint::getSpatialEntitiesEndIterator().

Referenced by [printTimePoint\(\)](#).

**8.263.1.3 void printTimePoint ( TimePoint & *timePoint* )**

Definition at line 23 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), multiscale::verification::TimePoint::getValue(), multiscale::verification::NR\_SUBSET\_SPECIFIC\_TYPES, and printSpatialEntities().

Referenced by printTrace().

**8.263.1.4 void printTrace ( SpatialTemporalTrace & *trace* )**

Definition at line 34 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::SpatialTemporalTrace::length(), and printTimePoint().

Referenced by readValidXmlFiles().

**8.263.1.5 void readValidXmlFiles ( SpatialTemporalDataReader & *reader* )**

Definition at line 47 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::SpatialTemporalDataReader::getNextSpatialTemporalTrace(), multiscale::verification::SpatialTemporalDataReader::hasNext(), and printTrace().

Referenced by readValidXmlFilesFromFolder().

**8.263.1.6 void readValidXmlFilesFromFolder ( const std::string & *path* )**

Definition at line 56 of file SpatialTemporalDataReaderSample.cpp.

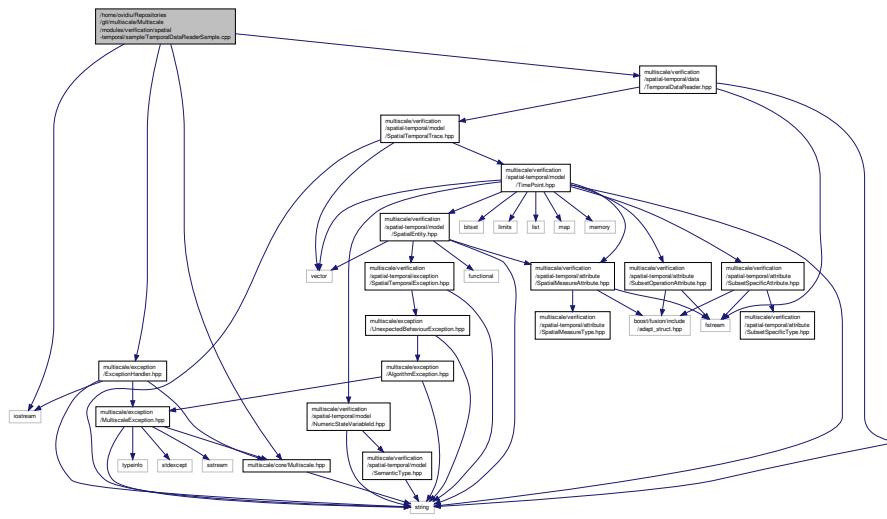
References multiscale::ExceptionHandler::printDetailedErrorMessage(), and readValidXmlFiles().

Referenced by main().

**8.264 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/SpatialTemporalDataReaderSample.cpp File Reference**

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/verification/spatial-temporal/data/TemporalDataReader.hpp"
#include <iostream>
```

Include dependency graph for TemporalDataReaderSample.cpp:



## Functions

- void [printTimePoint \(TimePoint &timePoint\)](#)
- void [printSpatialTemporalTrace \( SpatialTemporalTrace &trace \)](#)
- int [readAndPrintInputFile \(const std::string &filePath\)](#)
- int [main \(int argc, char \\*\\*argv\)](#)

### 8.264.1 Function Documentation

#### 8.264.1.1 int main ( int argc, char \*\* argv )

Definition at line 57 of file TemporalDataReaderSample.cpp.

References multiscale::EXEC\_ERR\_CODE, and [readAndPrintInputFile\(\)](#).

#### 8.264.1.2 void printSpatialTemporalTrace ( SpatialTemporalTrace & trace )

Definition at line 27 of file TemporalDataReaderSample.cpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::SpatialTemporalTrace::length(), and [printTimePoint\(\)](#).

Referenced by [readAndPrintInputFile\(\)](#).

#### 8.264.1.3 void printTimePoint ( TimePoint & timePoint )

Definition at line 12 of file TemporalDataReaderSample.cpp.

References multiscale::verification::TimePoint::getNumericStateVariablesBeginIterator(), multiscale::verification::TimePoint::getNumericStateVariablesEndIterator(), and multiscale::verification::TimePoint::getValue().

Referenced by [printSpatialTemporalTrace\(\)](#).

#### 8.264.1.4 int readAndPrintInputFile ( const std::string & filePath )

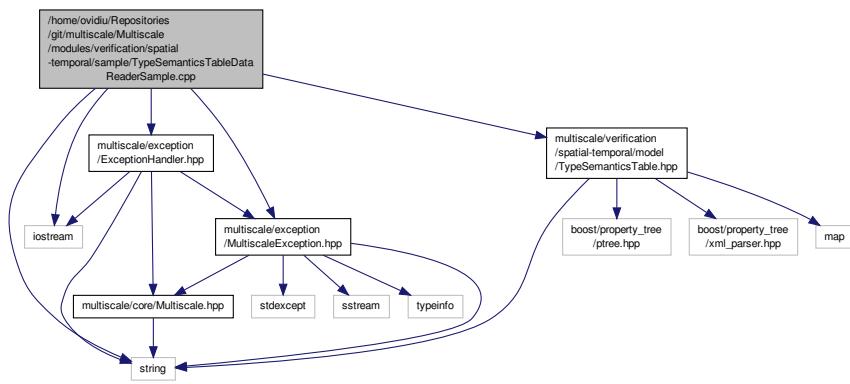
Definition at line 40 of file TemporalDataReaderSample.cpp.

References multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(), printSpatialTemporalTrace(), and multiscale::verification::TemporalDataReader::read().

Referenced by main().

## 8.265 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/TypeSemanticsTableDataReaderSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp"
#include <iostream>
#include <string>
Include dependency graph for TypeSemanticsTableDataReaderSample.cpp:
```



### Functions

- void [printTypeSemanticsTableContents](#) (const [TypeSemanticsTable](#) &typeSemanticsTable)
- void [readTypeSemanticsTable](#) (const std::string &inputFile, [TypeSemanticsTable](#) &typeSemanticsTable)
- void [readAndPrintTypeSemanticsTable](#) (const std::string &inputFile)
- int [main](#) (int argc, char \*\*argv)

#### 8.265.1 Function Documentation

##### 8.265.1.1 int main ( int argc, char \*\* argv )

Definition at line 39 of file TypeSemanticsTableDataReaderSample.cpp.

References multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, and [readAndPrintTypeSemanticsTable\(\)](#).

##### 8.265.1.2 void printTypeSemanticsTableContents ( const [TypeSemanticsTable](#) & typeSemanticsTable )

Definition at line 13 of file TypeSemanticsTableDataReaderSample.cpp.

References multiscale::verification::TypeSemanticsTable::getBeginIterator(), and multiscale::verification::TypeSemanticsTable::getEndIterator().

8.265.1.3 void readAndPrintTypeSemanticsTable ( const std::string & *inputFile* )

Definition at line 27 of file TypeSemanticsTableDataReaderSample.cpp.

References multiscale::ExceptionHandler::printDetailedErrorMessage(), printTypeSemanticsTableContents(), and readTypeSemanticsTable().

Referenced by main().

8.265.1.4 void readTypeSemanticsTable ( const std::string & *inputFile*, TypeSemanticsTable & *typeSemanticsTable* )

Definition at line 22 of file TypeSemanticsTableDataReaderSample.cpp.

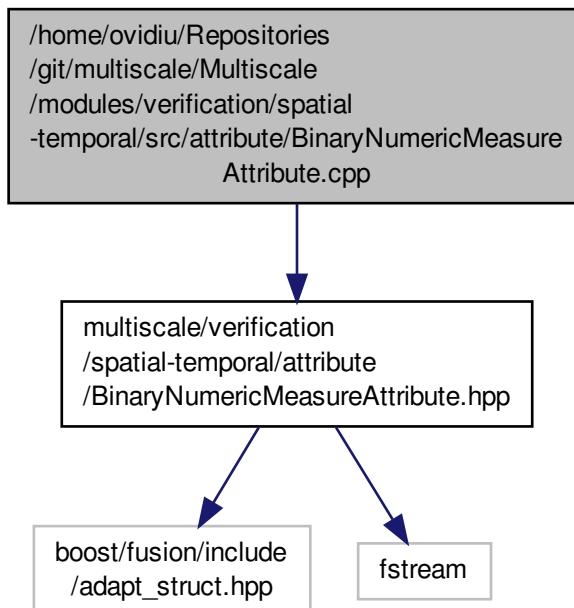
References multiscale::verification::TypeSemanticsTable::readTableFromFile().

Referenced by readAndPrintTypeSemanticsTable().

## 8.266 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinaryNumericMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryNumericMeasureAttribute.hpp"
```

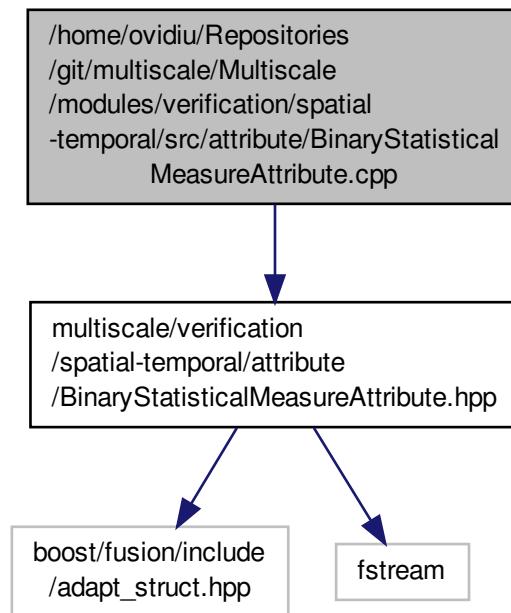
Include dependency graph for BinaryNumericMeasureAttribute.cpp:



## 8.267 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinaryStatisticalMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalMeasureAttribute.hpp"
```

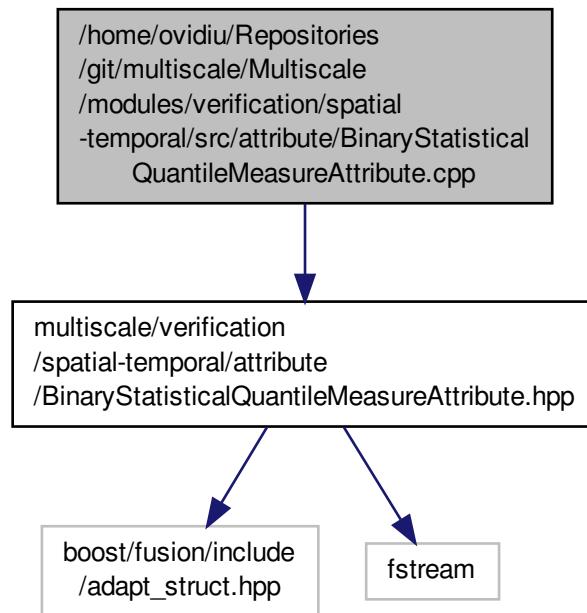
Include dependency graph for BinaryStatisticalMeasureAttribute.cpp:



## 8.268 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinaryStatisticalQuantileMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/BinaryStatisticalQuantileMeasureAttribute.hpp"
```

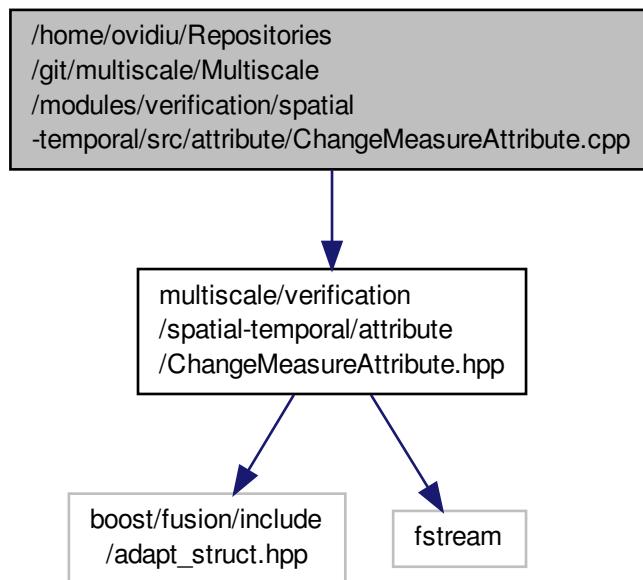
Include dependency graph for BinaryStatisticalQuantileMeasureAttribute.cpp:



## 8.269 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ChangeMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ChangeMeasureAttribute.hpp"
```

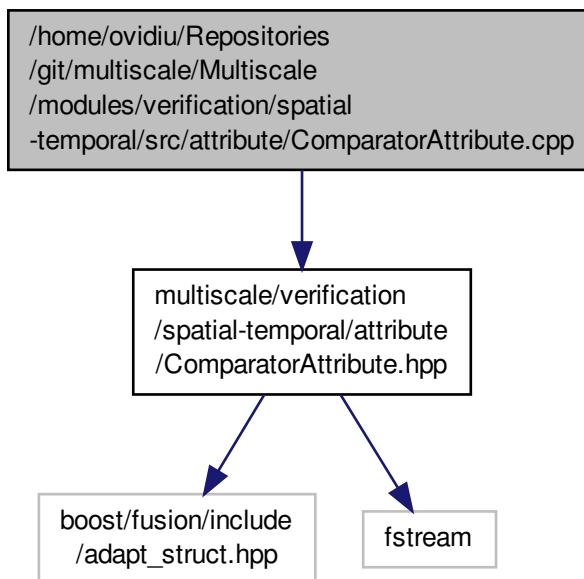
Include dependency graph for ChangeMeasureAttribute.cpp:



## 8.270 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ComparatorAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp"
```

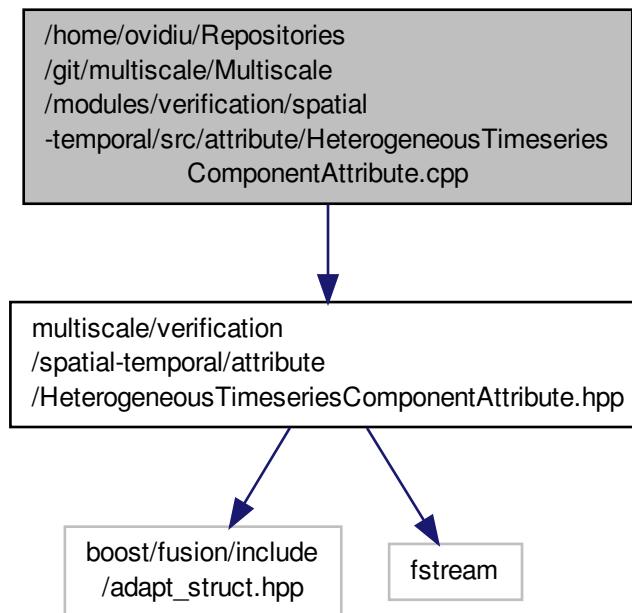
Include dependency graph for ComparatorAttribute.cpp:



8.271 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/HeterogeneousTimeseriesComponentAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/HeterogeneousTimeseriesComponentAttribute.hpp"
```

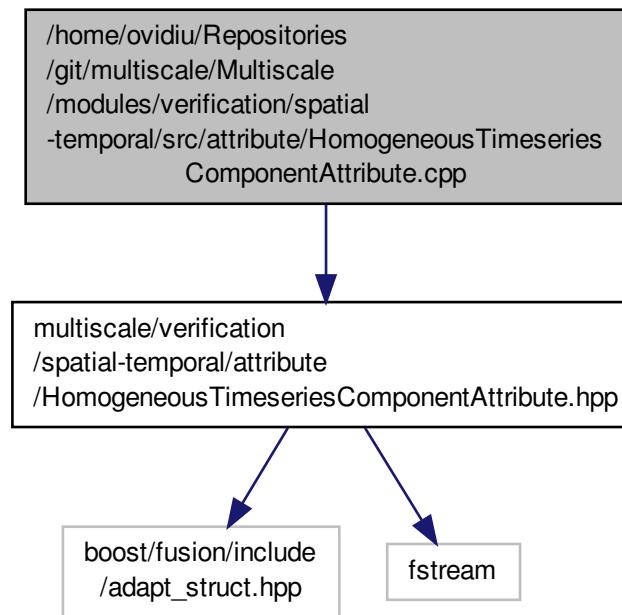
Include dependency graph for HeterogeneousTimeseriesComponentAttribute.cpp:



## 8.272 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/HomogeneousTimeseriesComponentAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesComponentAttribute.hpp"
```

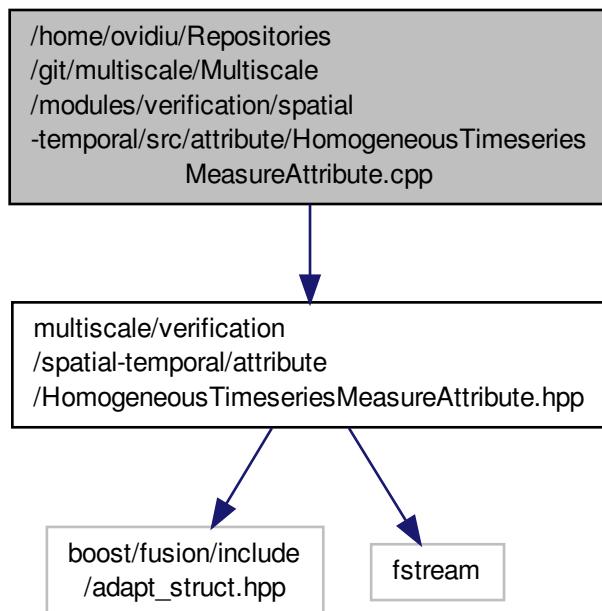
Include dependency graph for HomogeneousTimeseriesComponentAttribute.cpp:



8.273 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/HomogeneousTimeseriesMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/HomogeneousTimeseriesMeasureAttribute.hpp"
```

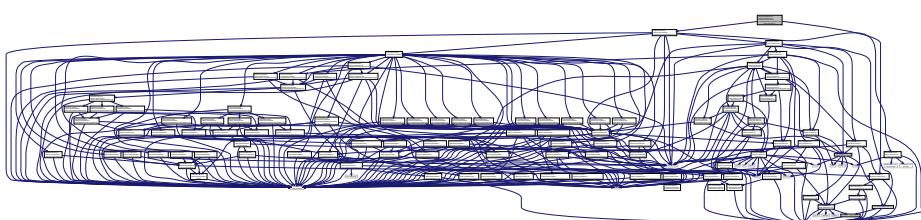
Include dependency graph for HomogeneousTimeseriesMeasureAttribute.cpp:



## 8.274 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ProbabilisticLogicPropertyAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/ProbabilisticLogicPropertyAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
```

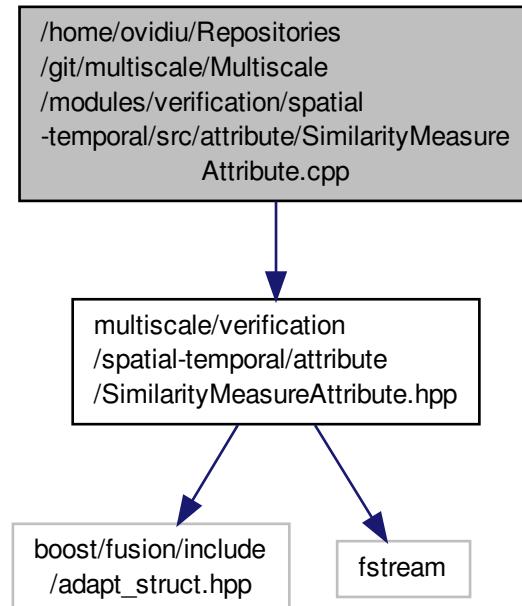
Include dependency graph for ProbabilisticLogicPropertyAttribute.cpp:



## 8.275 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SimilarityMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SimilarityMeasureAttribute.hpp"
```

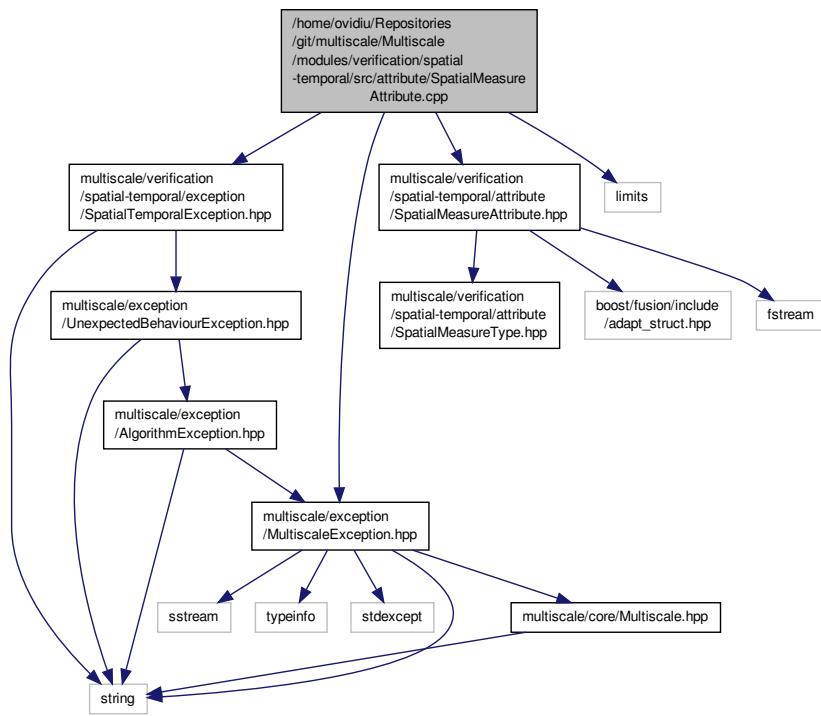
Include dependency graph for SpatialMeasureAttribute.cpp:



## 8.276 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SpatialMeasureAttribute.cpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include <limits>
```

Include dependency graph for SpatialMeasureAttribute.cpp:



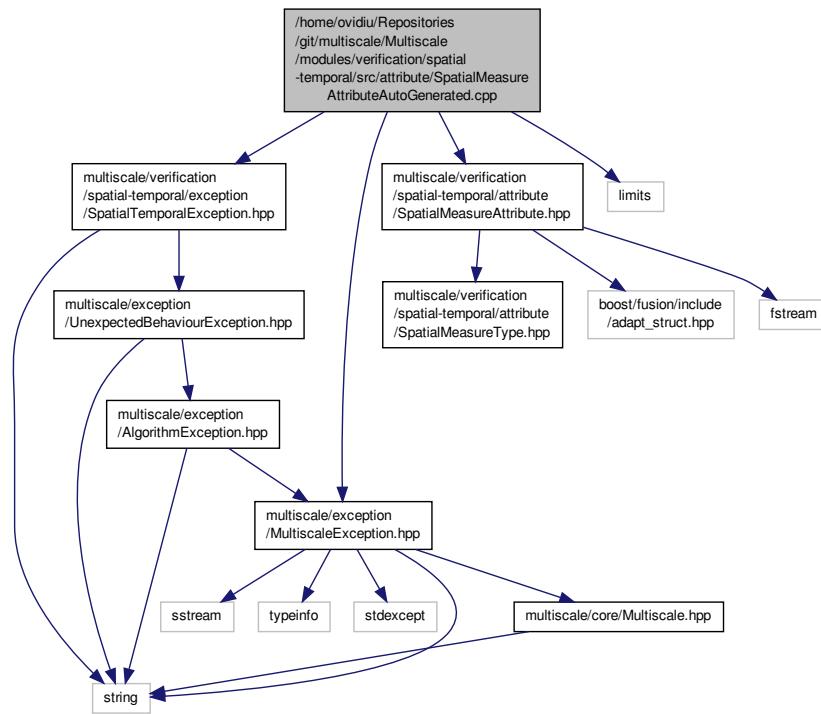
## 8.277 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SpatialMeasureAttributeAutoGenerated.cpp File Reference

```

#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include <limits>

```

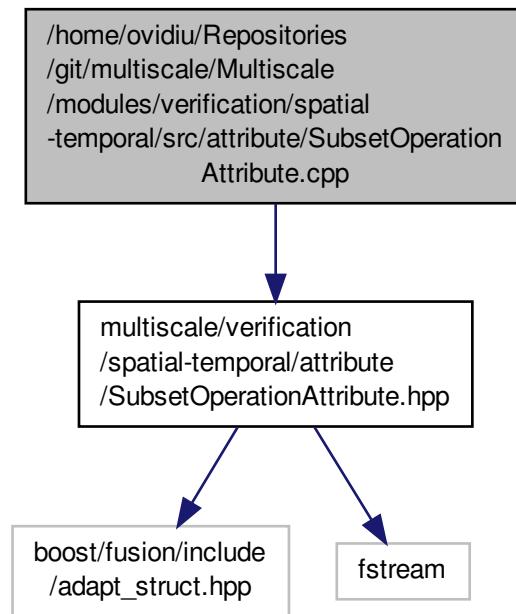
Include dependency graph for SpatialMeasureAttributeAutoGenerated.cpp:



8.278 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SubsetOperationAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp"
```

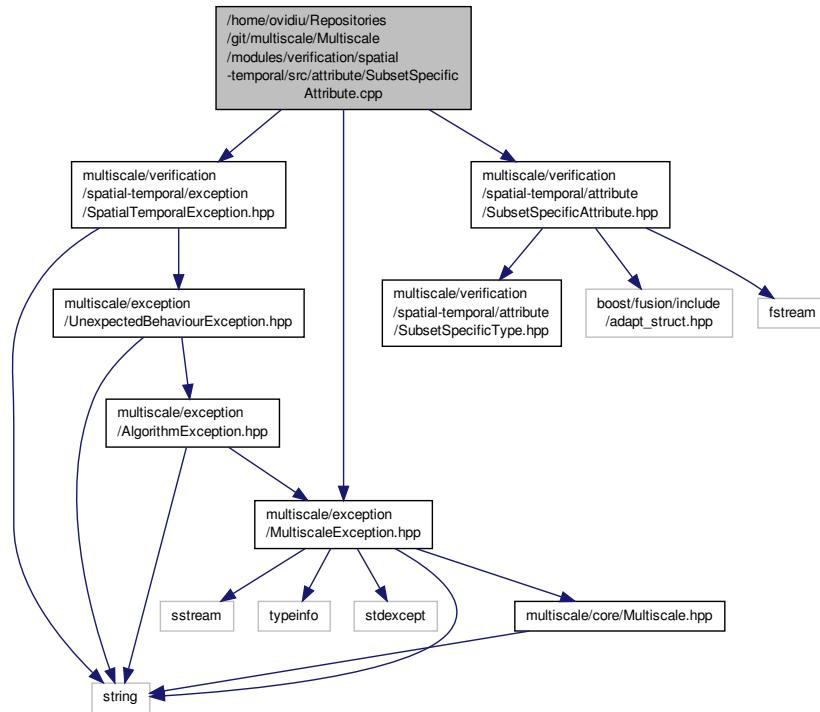
Include dependency graph for SubsetOperationAttribute.cpp:



## 8.279 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SubsetSpecificAttribute.cpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
```

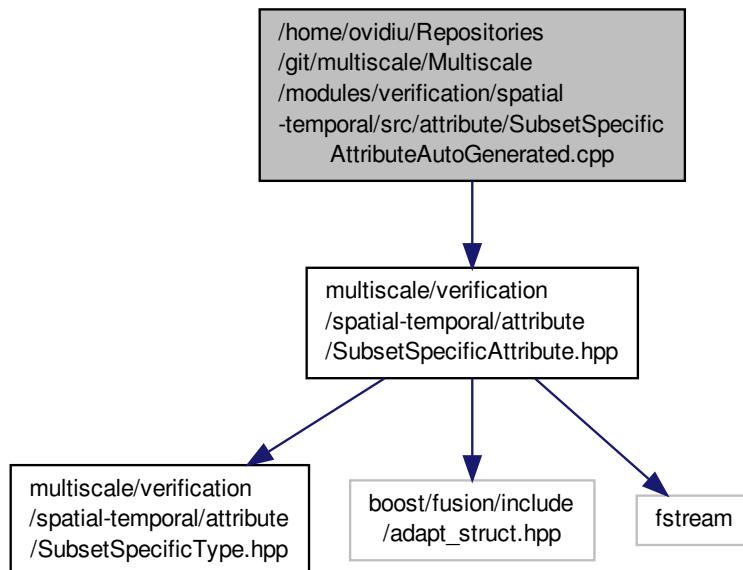
Include dependency graph for SubsetSpecificAttribute.cpp:



8.280 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SubsetSpecificAttributeAutoGenerated.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp"
```

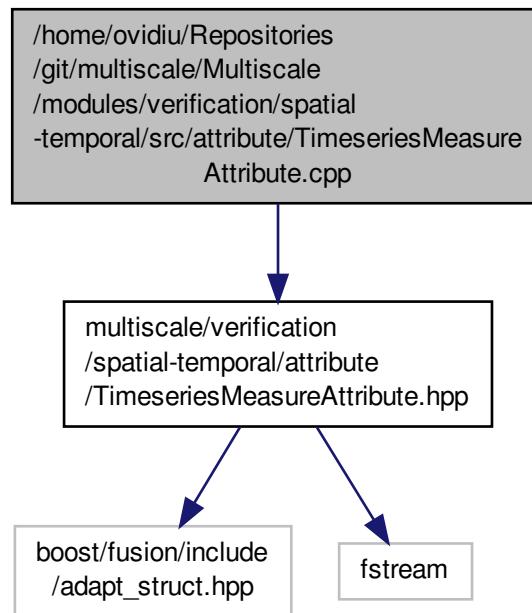
Include dependency graph for SubsetSpecificAttributeAutoGenerated.cpp:



## 8.281 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/TimeseriesMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/TimeseriesMeasureAttribute.hpp"
```

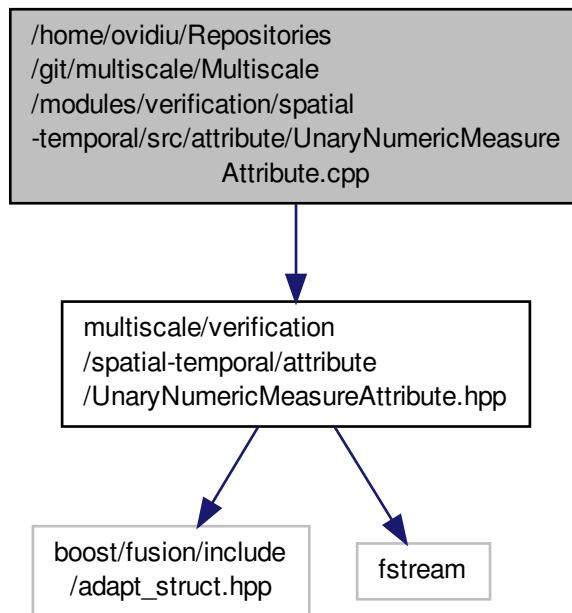
Include dependency graph for TimeseriesMeasureAttribute.cpp:



8.282 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnaryNumericMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp"
```

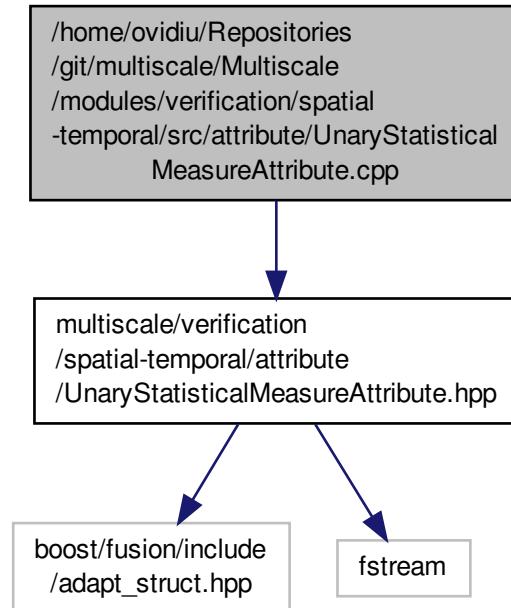
Include dependency graph for UnaryNumericMeasureAttribute.cpp:



### 8.283 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnaryStatisticalMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/UnaryStatisticalMeasureAttribute.hpp"
```

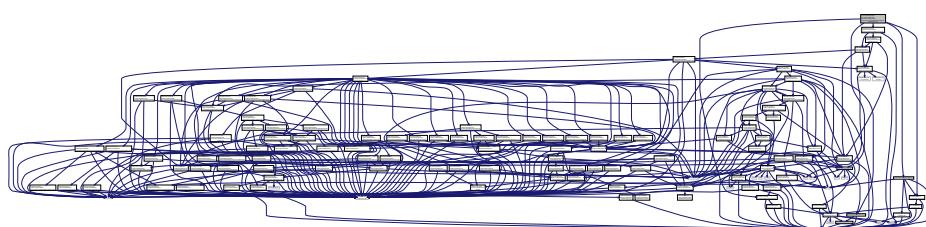
Include dependency graph for UnaryStatisticalMeasureAttribute.cpp:



## 8.284 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelChecker.cpp File Reference

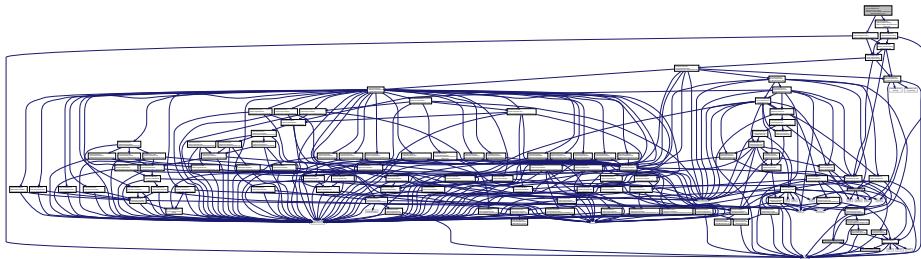
```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp"
```

Include dependency graph for `ApproximateBayesianModelChecker.cpp`:



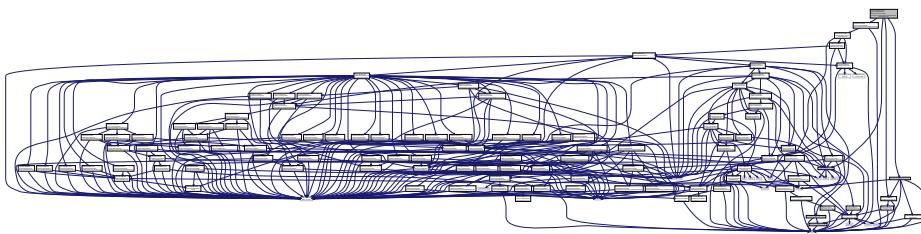
## 8.285 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelCheckerFactory.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelCheckerFactory.hpp"
Include dependency graph for ApproximateBayesianModelCheckerFactory.cpp:
```



## 8.286 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModelChecker.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp"
Include dependency graph for ApproximateProbabilisticModelChecker.cpp:
```

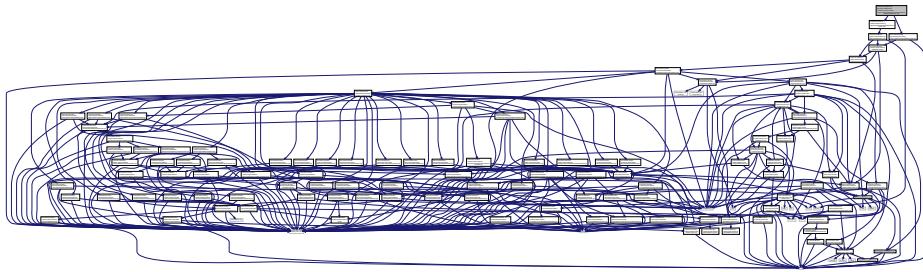


## 8.287 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModelCheckerFactory.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelCheckerFactory.hpp"
```

Reference

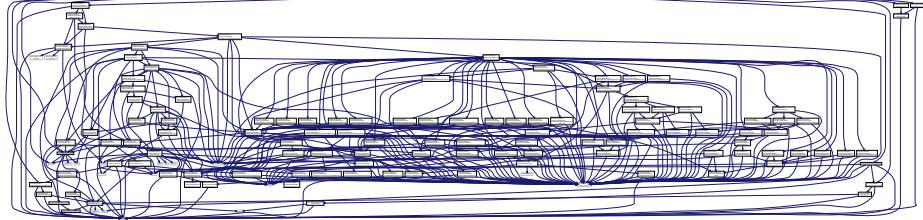
Include dependency graph for ApproximateProbabilisticModelCheckerFactory.cpp:



8.288 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/BayesianModelChecker.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/statistics/BetaDistribution.hpp"
#include "multiscale/util/statistics/BinomialDistribution.hpp"
#include "multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp"
#include <limits>
```

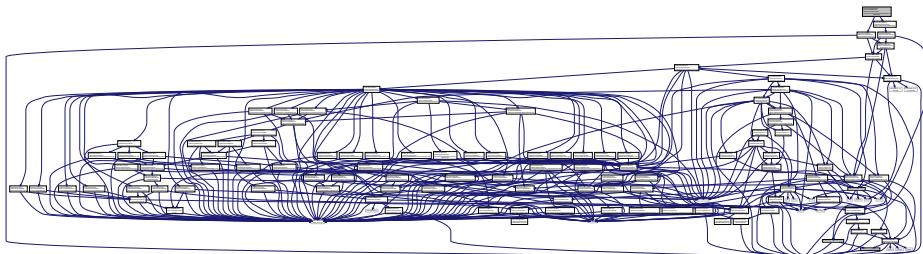
Include dependency graph for BayesianModelChecker.cpp:



8.289 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/BayesianModelCheckerFactory.cpp File Reference

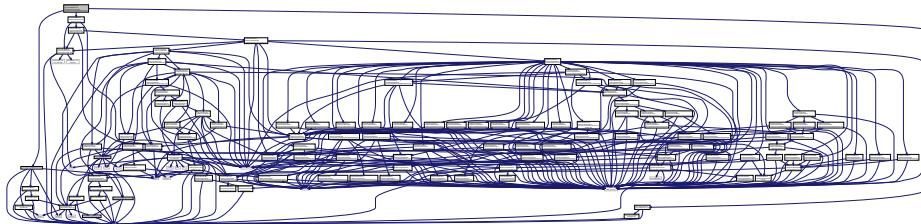
```
#include "multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/BayesianModelCheckerFactory.hpp"
```

Include dependency graph for BayesianModelCheckerFactory.cpp:



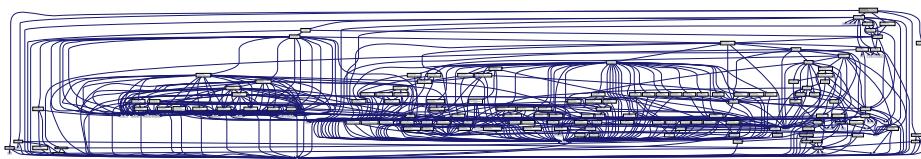
## 8.290 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelChecker.cpp File Reference

```
#include "multiscale/util/statistics/BinomialDistribution.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
Include dependency graph for ModelChecker.cpp:
```



## 8.291 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingManager.cpp File Reference

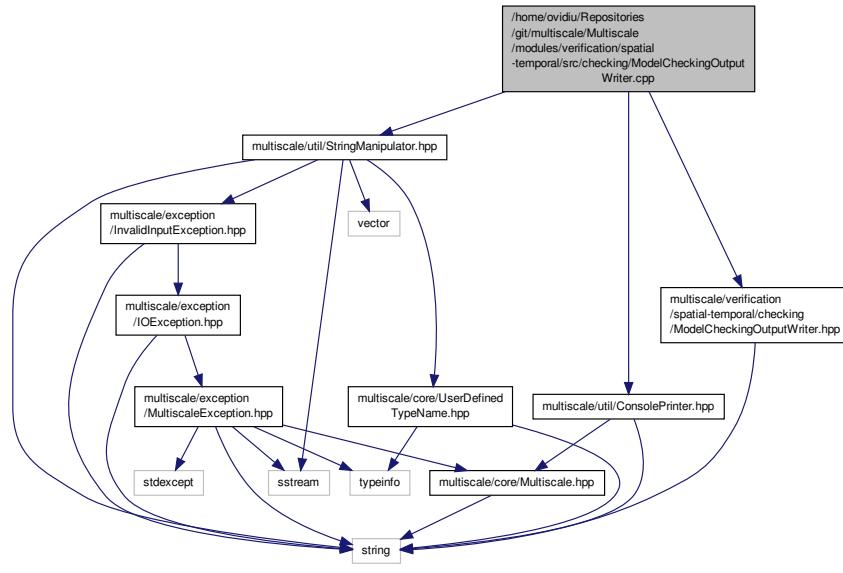
```
#include "multiscale/exception/ExceptionHandler.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/OperatingSystem.hpp"
#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckingManager.hpp"
Include dependency graph for ModelCheckingManager.cpp:
```



## 8.292 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.cpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelCheckingOutputWriter.hpp"
```

Include dependency graph for ModelCheckingOutputWriter.cpp:



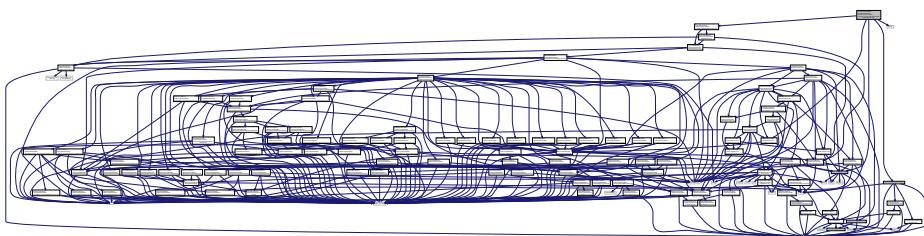
## 8.293 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelChecker.cpp File Reference

```

#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp"
#include <iostream>

```

Include dependency graph for ProbabilisticBlackBoxModelChecker.cpp:



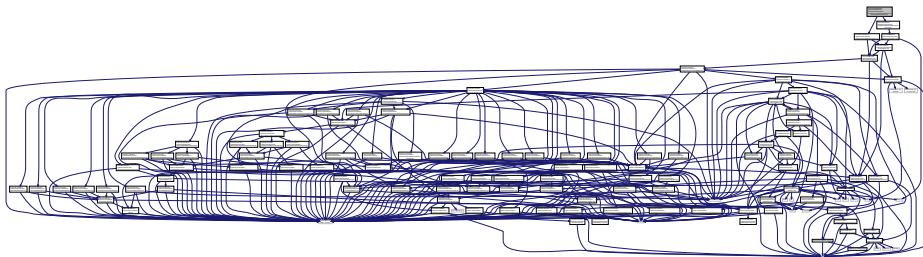
## 8.294 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelCheckerFactory.cpp File Reference

```

#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelCheckerFactory.hpp"

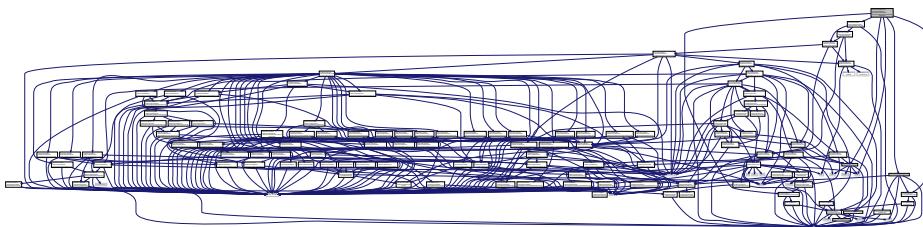
```

`BlackBoxModelCheckerFactory.hpp`  
 Include dependency graph for `ProbabilisticBlackBoxModelCheckerFactory.cpp`:



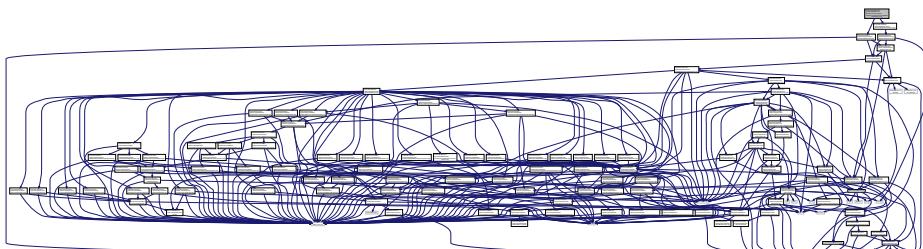
## 8.295 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/StatisticalModelChecker.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/Statistical-
ModelChecker.hpp"
#include <limits>
Include dependency graph for StatisticalModelChecker.cpp:
```



## 8.296 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/StatisticalModelCheckerFactory.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/Statistical-
ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/checking/Statistical-
ModelCheckerFactory.hpp"
Include dependency graph for StatisticalModelCheckerFactory.cpp:
```



8.297 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[LogicPropertyDataReader.cpp File](#)

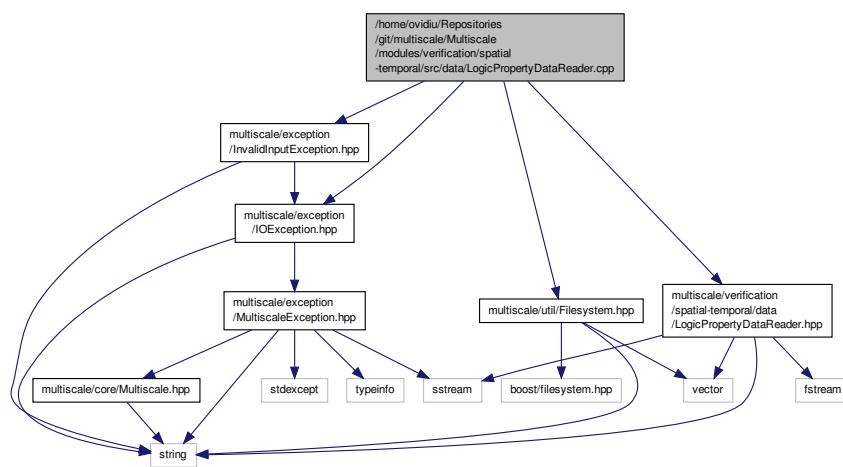
Reference

1325

8.297 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[LogicPropertyDataReader.cpp File Reference](#)

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/IOException.hpp"
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/verification/spatial-temporal/data/LogicPropertyData-
Reader.hpp"
```

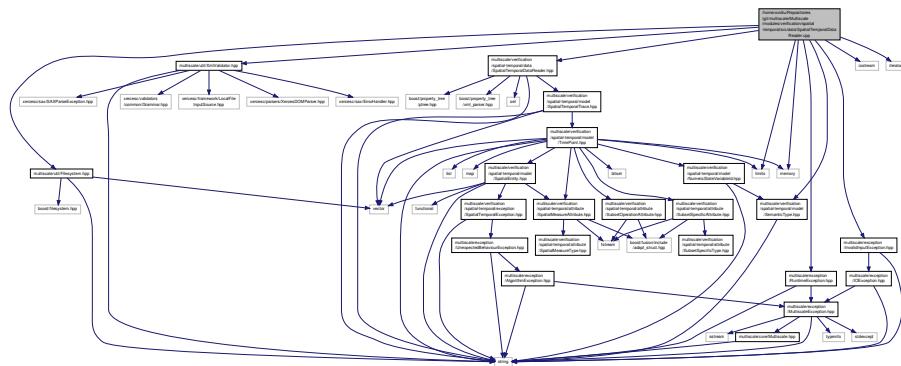
Include dependency graph for LogicPropertyDataReader.cpp:



8.298 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/SpatialTemporalDataReader.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/RuntimeException.hpp"
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/util/XmlValidator.hpp"
#include "multiscale/verification/spatial-temporal/data/SpatialTemporal-
DataReader.hpp"
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include <iostream>
#include <iterator>
#include <limits>
#include <memory>
```

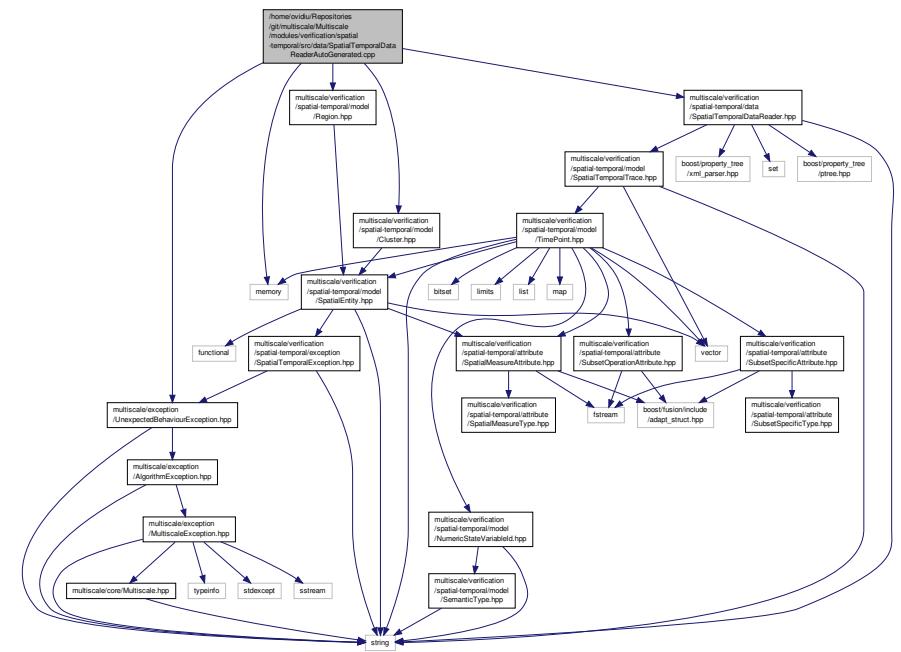
Include dependency graph for SpatialTemporalDataReader.cpp:



8.299 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/SpatialTemporalDataReaderAutoGenerated.cpp File Reference

```
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp"
#include "multiscale/verification/spatial-temporal/model/Cluster.hpp"
#include "multiscale/verification/spatial-temporal/model/Region.hpp"
#include <memory>
```

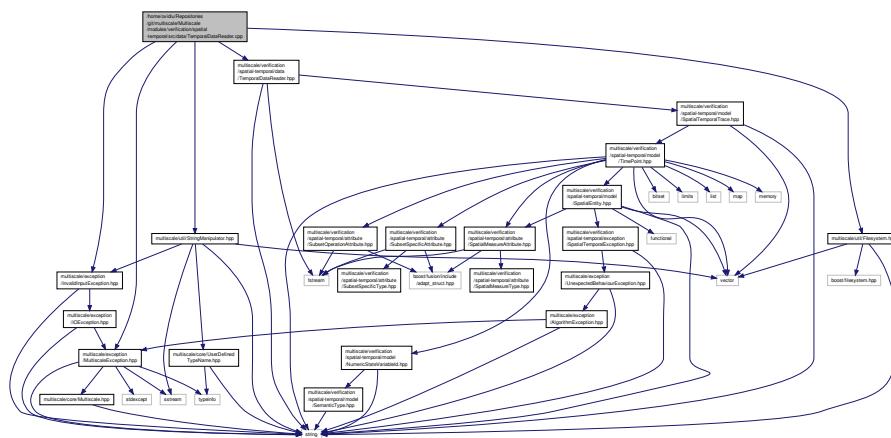
Include dependency graph for SpatialTemporalDataReaderAutoGenerated.cpp:



8.300 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/TemporalDataReader.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/exception/MultiscaleException.hpp"
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/data/TemporalDataReader.hpp"
```

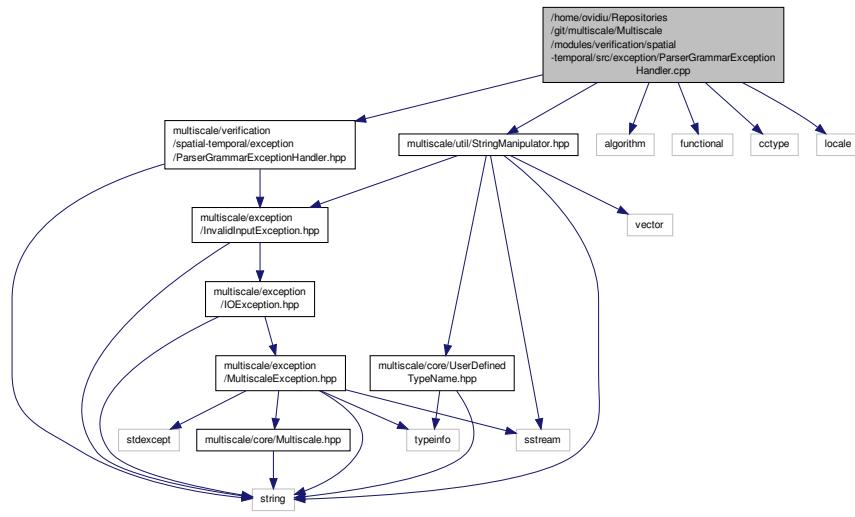
Include dependency graph for TemporalDataReader.cpp:



## 8.301 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ParserGrammarExceptionHandler.cpp File Reference

```
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarExceptionHandler.hpp"
#include <algorithm>
#include <functional>
#include <cctype>
#include <locale>
```

Include dependency graph for ParserGrammarExceptionHandler.cpp:



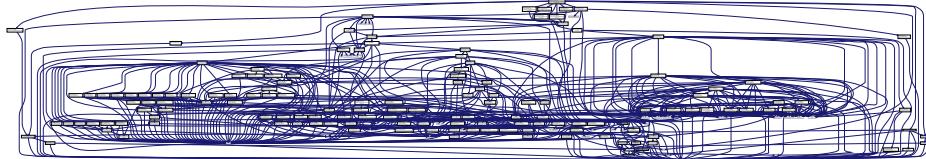
### 8.302 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/CommandLineModelChecking.cpp File Reference

```

#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/checking/BayesianModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/checking/StatisticalModelCheckerFactory.hpp"
#include "multiscale/verification/spatial-temporal/exception/ModelCheckingException.hpp"
#include "multiscale/verification/spatial-temporal/exception/ModelCheckingHelpRequestException.hpp"
#include "multiscale/verification/spatial-temporal/execution/CommandLineModelChecking.hpp"
#include <iostream>

```

Include dependency graph for CommandLineModelChecking.cpp:



8.303 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/AbstractSyntaxTree.cpp File

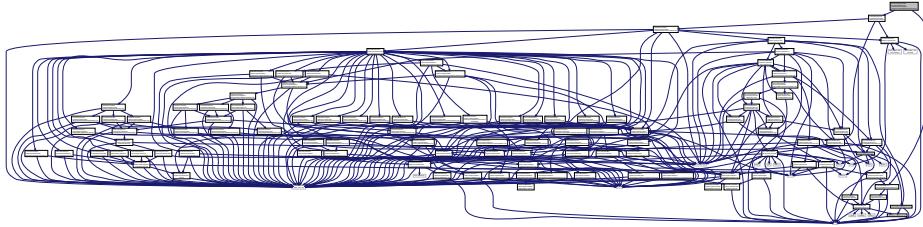
Reference

1329

8.303 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/AbstractSyntaxTree.cpp File Reference

```
#include "multiscale/exception/UnexpectedBehaviourException.hpp"
#include "multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp"
```

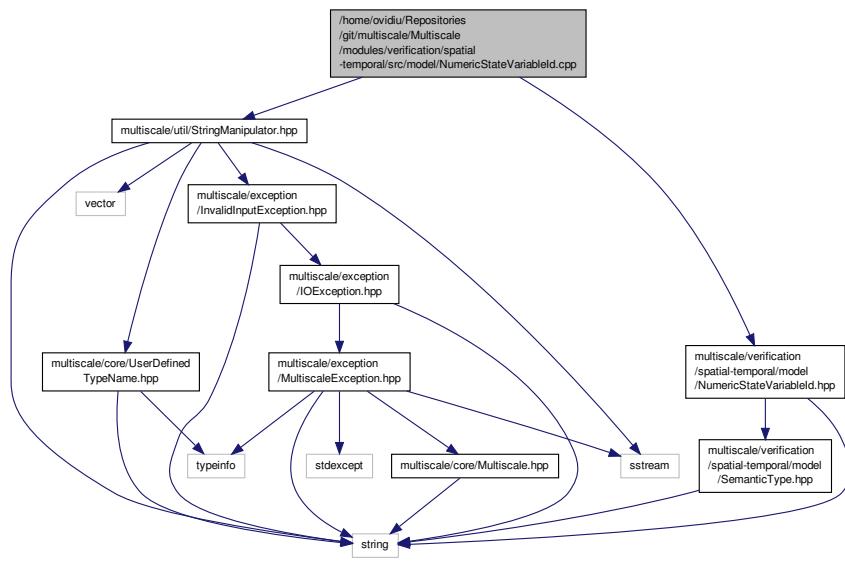
Include dependency graph for AbstractSyntaxTree.cpp:



8.304 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/NumericStateVariableId.cpp File Reference

```
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/model/NumericStateVariableId.hpp"
```

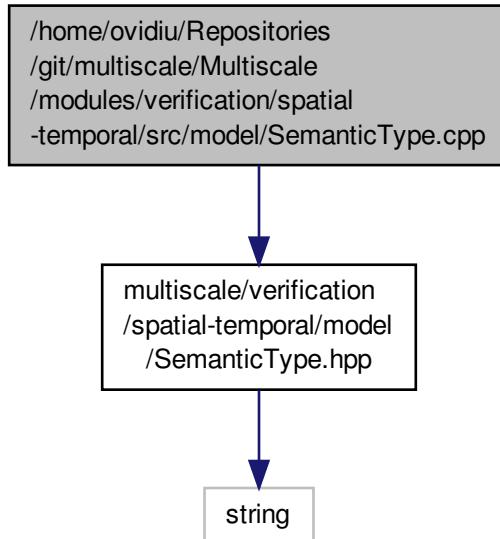
Include dependency graph for NumericStateVariableId.cpp:



8.305 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SemanticType.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
```

Include dependency graph for SemanticType.cpp:



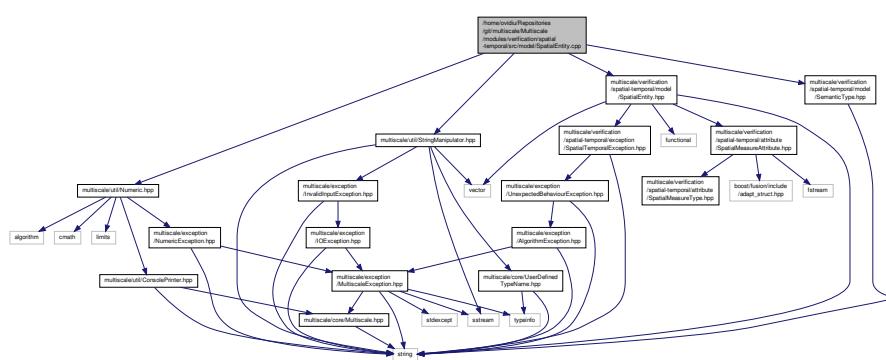
## 8.306 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialEntity.cpp File Reference

```

#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialEntity.hpp"
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"

```

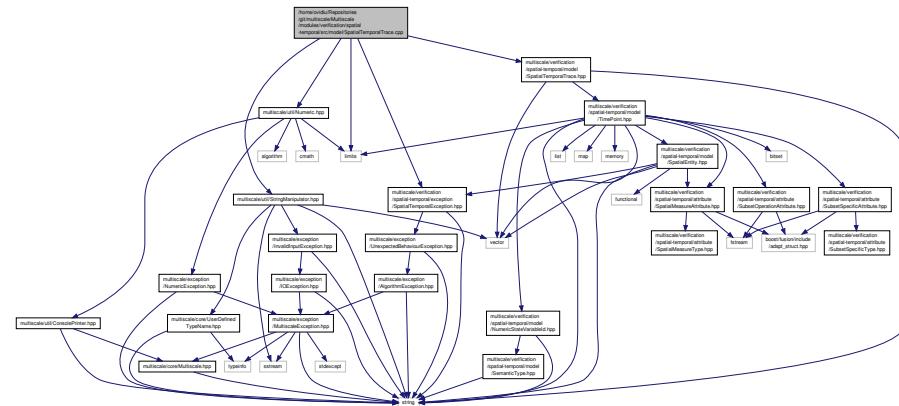
Include dependency graph for SpatialEntity.cpp:



8.307 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialTemporalTrace.cpp File Reference

```
#include "multiscale/util/Numeric.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp"
#include <limits>
Include dependency graph for SpatialTemporalTrace.cpp:
```

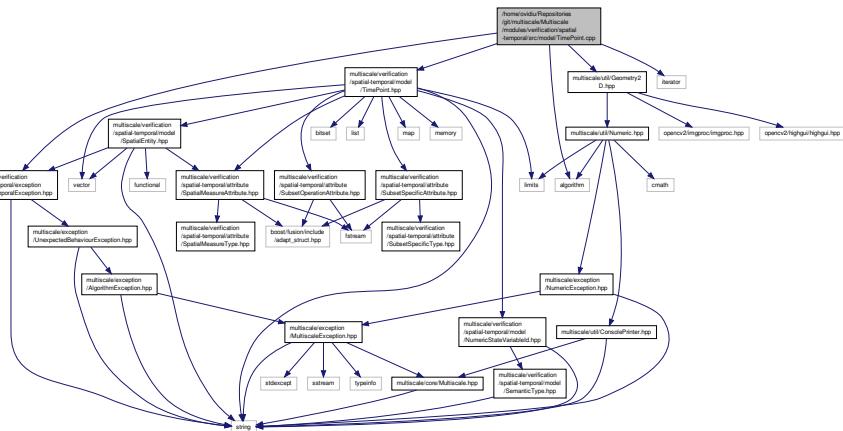
Include dependency graph for SpatialTemporalTrace.cpp:



## 8.308 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/TimePoint.cpp File Reference

```
#include "multiscale/util/Geometry2D.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include "multiscale/verification/spatial-temporal/model/TimePoint.hpp"
#include <algorithm>
#include <iterator>
```

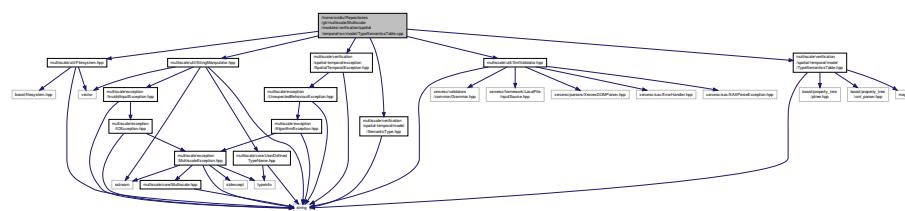
Include dependency graph for TimePoint.cpp?



8.309 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/TypeSemanticsTable.cpp File Reference

```
#include "multiscale/util/Filesystem.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/util/XmlValidator.hpp"
#include "multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp"
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemanticsTable.hpp"
```

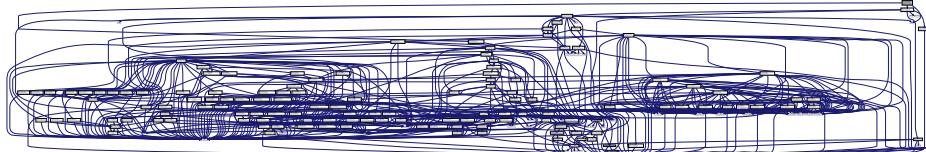
Include dependency graph for TypeSemanticsTable.cpp:



## 8.310 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/Mule.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/ModelCheckingException.hpp"
#include "multiscale/verification/spatial-temporal/execution/CommandLine.hpp"
#include <ctime>
Include dependency graph for Mule.cpp:
```

Include dependency graph for Mule.cpp:



## Functions

- void `runModelCheckingTask` (int argc, char \*\*argv)
  - int `main` (int argc, char \*\*argv)

### 8.310.1 Function Documentation

#### 8.310.1.1 int main ( int *argc*, char \*\* *argv* )

Definition at line 25 of file Mule.cpp.

References multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(), multiscale::ExceptionHandler::printRawErrorMessage(), and runModelCheckingTask().

**8.310.1.2 void runModelCheckingTask( int argc, char \*\* argv )**

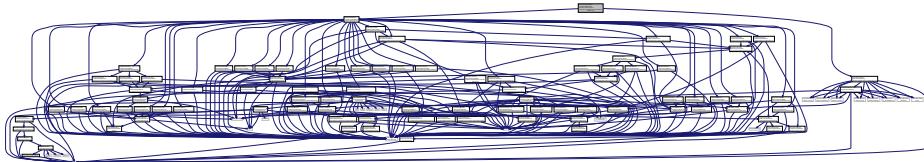
Definition at line 17 of file Mule.cpp.

References multiscale::verification::CommandLineModelChecking::execute(), and multiscale::verification::CommandLineModelChecking::initialise().

Referenced by main().

## **8.311 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/BinaryNumericMeasureGrammar.cpp File Reference**

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryNumericMeasureGrammarDefinition.hpp"
Include dependency graph for BinaryNumericMeasureGrammar.cpp:
```



### **TypeDefs**

- **typedef std::string::const\_iterator iteratorType**

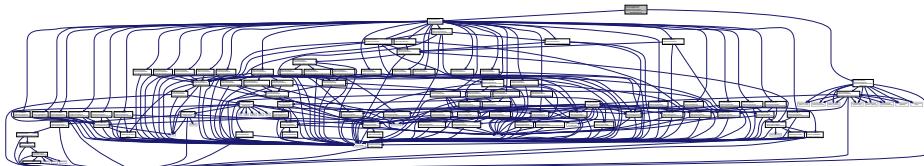
#### **8.311.1 Typedef Documentation**

##### **8.311.1.1 `typedef std::string::const_iterator iteratorType`**

Definition at line 8 of file BinaryNumericMeasureGrammar.cpp.

## **8.312 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/BinaryStatisticalMeasureGrammar.cpp File Reference**

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalMeasureGrammarDefinition.hpp"
Include dependency graph for BinaryStatisticalMeasureGrammar.cpp:
```



## Typedefs

- `typedef std::string::const_iterator iteratorType`

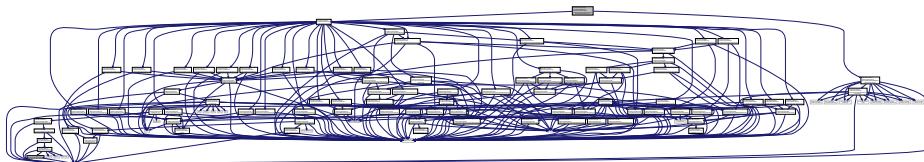
### 8.312.1 Typedef Documentation

#### 8.312.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 8 of file `BinaryStatisticalMeasureGrammar.cpp`.

## 8.313 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/BinaryStatisticalQuantileMeasureGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/BinaryStatisticalQuantileMeasureGrammarDefinition.hpp"
Include dependency graph for BinaryStatisticalQuantileMeasureGrammar.cpp:
```



## Typedefs

- `typedef std::string::const_iterator iteratorType`

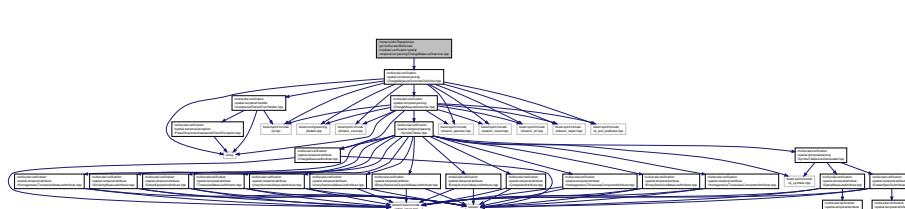
### 8.313.1 Typedef Documentation

#### 8.313.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 8 of file `BinaryStatisticalQuantileMeasureGrammar.cpp`.

## 8.314 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ChangeMeasureGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/ChangeMeasureGrammarDefinition.hpp"
Include dependency graph for ChangeMeasureGrammar.cpp:
```



- `typedef std::string::const_iterator iteratorType`

### 8.314.1 Typedef Documentation

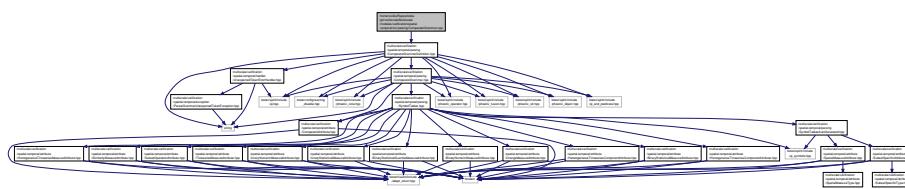
#### 8.314.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file ChangeMeasureGrammar.cpp.

## 8.315 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ComparatorGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/ComparatorGrammar.h"
Definition.hpp"
```

Include dependency graph for ComparatorGrammar.cpp:



### TypeDefs

- `typedef std::string::const_iterator iteratorType`

### 8.315.1 Typedef Documentation

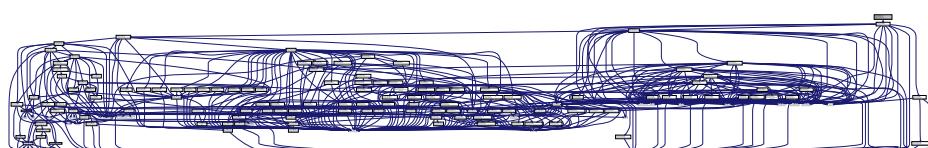
#### 8.315.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file ComparatorGrammar.cpp.

## 8.316 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/LogicPropertyGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/LogicPropertyGrammar.h"
GrammarDefinition.hpp"
```

Include dependency graph for LogicPropertyGrammar.cpp:



## TypeDefs

- `typedef std::string::const_iterator iteratorType`

### 8.316.1 Typedef Documentation

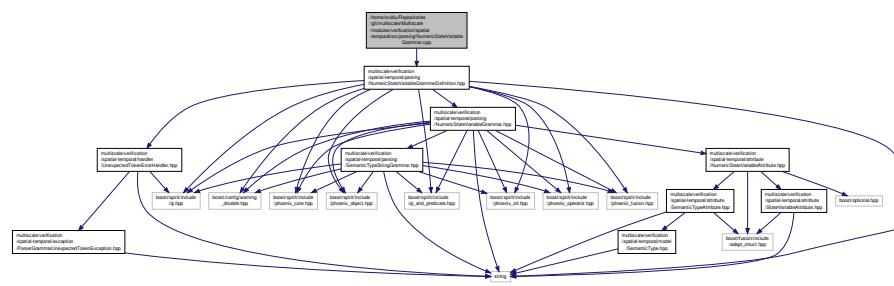
### 8.316.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file LogicPropertyGrammar.cpp.

8.317 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/NumericStateVariableGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/NumericStateVariableGrammarDefinition.hpp"
```

Include dependency graph for NumericStateVariableGrammar.cpp



## TypeDefs

- `typedef std::string::const_iterator iteratorType`

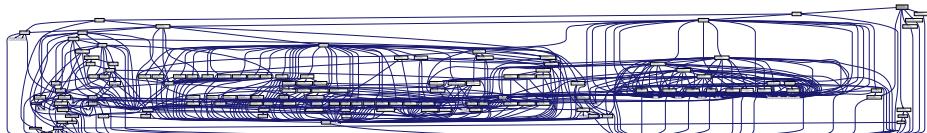
### 8.317.1 Typedef Documentation

### 8.317.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file NumericStateVariableGrammar.cpp.

## 8.318 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/Parser.cpp File Reference

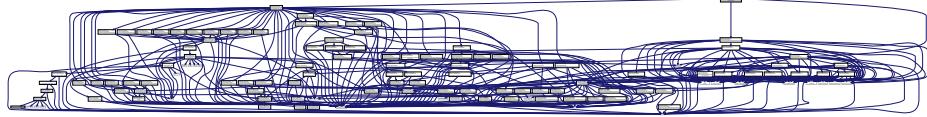
```
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarExceptionHandler.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarExtraInputException.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarProbabilityException.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarUnexpectedTokenException.hpp"
#include "multiscale/verification/spatial-temporal/exception/ParserGrammarUnknownTypeException.hpp"
```



## 8.319 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/PrimaryNumericMeasureGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammarDefinition.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammar.hpp"
```

Include dependency graph for PrimaryNumericMeasureGrammar.cpp:



## Typedefs

- `typedef std::string::const_iterator iteratorType`

### 8.319.1 Typedef Documentation

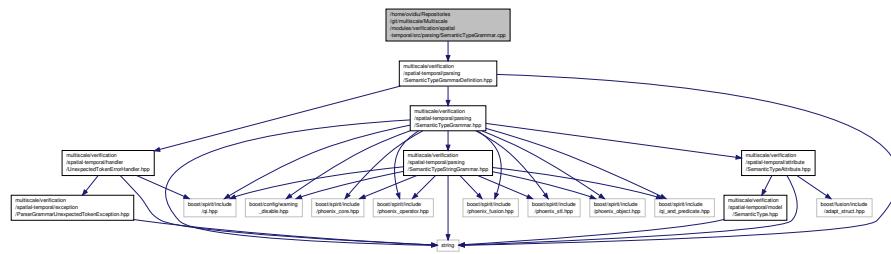
#### 8.319.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 9 of file PrimaryNumericMeasureGrammar.cpp.

## 8.320 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/SemanticTypeGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeGrammarDefinition.hpp"
```

Include dependency graph for SemanticTypeGrammar.cpp:



## TypeDefs

- `typedef std::string::const_iterator iteratorType`

### 8.320.1 Typedef Documentation

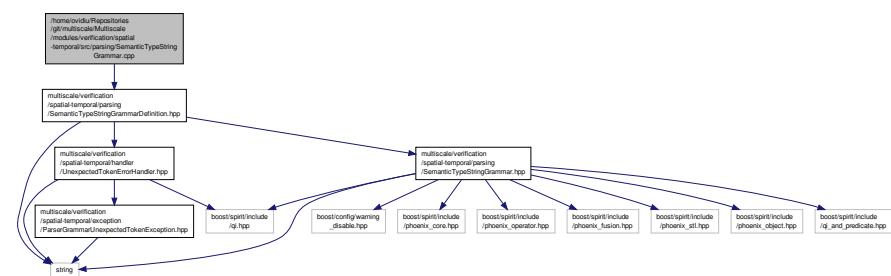
#### 8.320.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file SemanticTypeGrammar.cpp.

## 8.321 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/SemanticTypeStringGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/parsing/SemanticTypeStringGrammarDefinition.hpp"
```

Include dependency graph for SemanticTypeStringGrammar.cpp:



## TypeDefs

- `typedef std::string::const_iterator iteratorType`

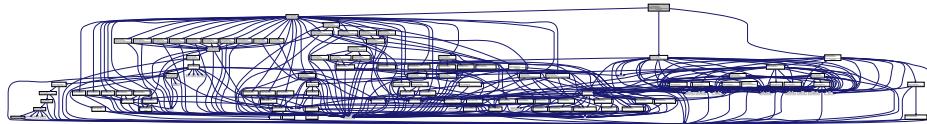
### 8.321.1 Typedef Documentation

#### 8.321.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 7 of file SemanticTypeStringGrammar.cpp.

8.322 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/SpatialMeasureCollectionGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/PrimaryNumericMeasureGrammar.hpp"
#include "multiscale/verification/spatial-temporal/parsing/SpatialMeasureCollectionGrammarDefinition.hpp"
Include dependency graph for SpatialMeasureCollectionGrammar.cpp:
```



## Typedefs

- `typedef std::string::const_iterator iteratorType`

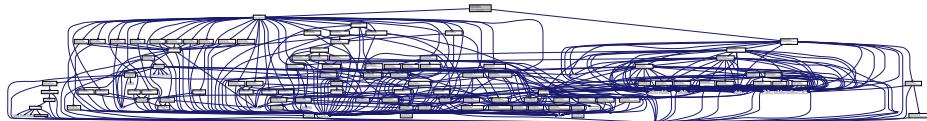
### 8.322.1 Typedef Documentation

#### 8.322.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 9 of file SpatialMeasureCollectionGrammar.cpp.

8.323 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/TemporalNumericCollectionGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericCollectionGrammarDefinition.hpp"
Include dependency graph for TemporalNumericCollectionGrammar.cpp:
```



## Typedefs

- `typedef std::string::const_iterator iteratorType`

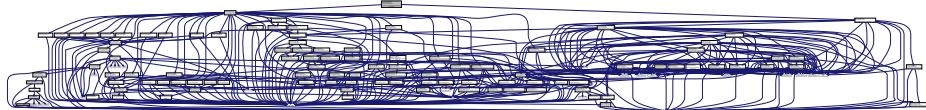
### 8.323.1 Typedef Documentation

#### 8.323.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 8 of file TemporalNumericCollectionGrammar.cpp.

## 8.324 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/TemporalNumericMeasureGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/TemporalNumericMeasureGrammarDefinition.hpp"
Include dependency graph for TemporalNumericMeasureGrammar.cpp:
```



### TypeDefs

- `typedef std::string::const_iterator iteratorType`

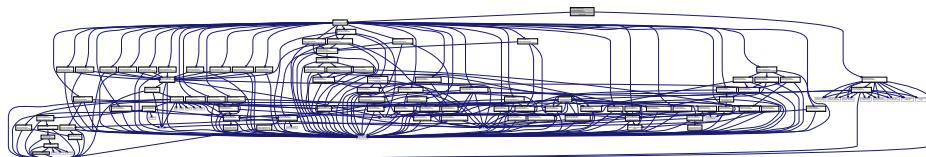
#### 8.324.1 Typedef Documentation

##### 8.324.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 8 of file TemporalNumericMeasureGrammar.cpp.

## 8.325 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/UnaryNumericMeasureGrammar.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryNumericMeasureGrammarDefinition.hpp"
Include dependency graph for UnaryNumericMeasureGrammar.cpp:
```



### TypeDefs

- `typedef std::string::const_iterator iteratorType`

#### 8.325.1 Typedef Documentation

##### 8.325.1.1 `typedef std::string::const_iterator iteratorType`

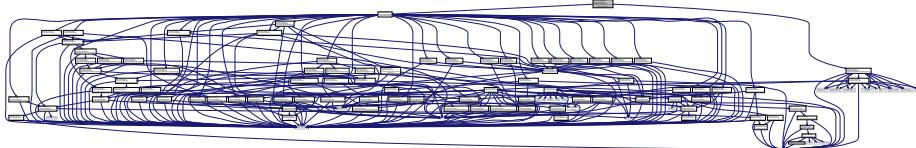
Definition at line 8 of file UnaryNumericMeasureGrammar.cpp.

8.326 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/UnaryStatisticalMeasureGrammar.cpp File

Reference

8.326 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/UnaryStatisticalMeasureGrammar.cpp File Reference 1341

```
#include "multiscale/verification/spatial-temporal/attribute/SynthesizedAttribute.hpp"
#include "multiscale/verification/spatial-temporal/parsing/UnaryStatisticalMeasureGrammarDefinition.hpp"
Include dependency graph for UnaryStatisticalMeasureGrammar.cpp:
```



## TypeDefs

- `typedef std::string::const_iterator iteratorType`

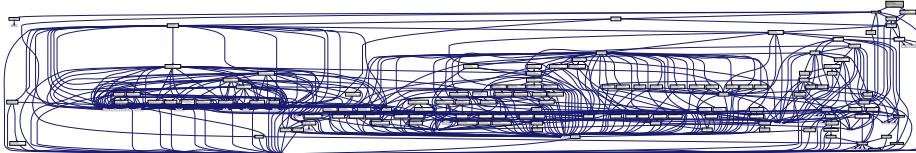
### 8.326.1 Typedef Documentation

#### 8.326.1.1 `typedef std::string::const_iterator iteratorType`

Definition at line 8 of file UnaryStatisticalMeasureGrammar.cpp.

8.327 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateBayesianModelCheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelChecker.hpp"
Include dependency graph for ApproximateBayesianModelCheckerTest.hpp:
```



## Classes

- `class multiscaletest::ApproximateBayesianModelCheckerTest`  
*Class for testing the approximate Bayesian model checker.*

## Namespaces

- `multiscaletest`

## Functions

- [TEST\\_F \(ApproximateBayesianModelCheckerTest, CaseTrue\)](#)
- [TEST\\_F \(ApproximateBayesianModelCheckerTest, CaseFalse\)](#)

### 8.327.1 Function Documentation

#### 8.327.1.1 TEST\_F ( ApproximateBayesianModelCheckerTest , CaseTrue )

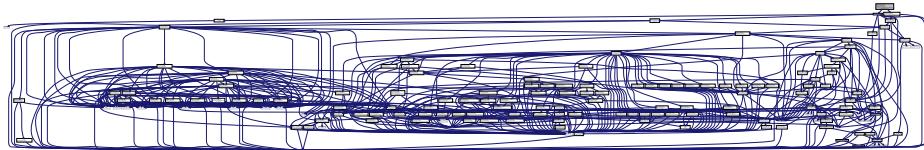
Definition at line 85 of file ApproximateBayesianModelCheckerTest.hpp.

#### 8.327.1.2 TEST\_F ( ApproximateBayesianModelCheckerTest , CaseFalse )

Definition at line 93 of file ApproximateBayesianModelCheckerTest.hpp.

### 8.328 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateProbabilisticModelCheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp"
#include "multiscale/verification/spatial-temporal/checking/ApproximateProbabilisticModelChecker.hpp"
Include dependency graph for ApproximateProbabilisticModelCheckerTest.hpp:
```



## Classes

- class [multiscaletest::ApproximateProbabilisticModelCheckerTest](#)  
*Class for testing the approximate probabilistic model checker.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(ApproximateProbabilisticModelCheckerTest, CaseFalse\)](#)

### 8.328.1 Function Documentation

#### 8.328.1.1 TEST\_F ( ApproximateProbabilisticModelCheckerTest , CaseFalse )

Definition at line 78 of file ApproximateProbabilisticModelCheckerTest.hpp.

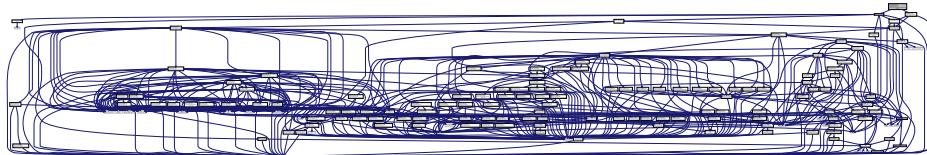
**8.329 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/BayesianModelCheckerTest.hpp File**

Reference

1343

**8.329 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/BayesianModelCheckerTest.hpp File Reference**

```
#include "ModelCheckerTest.hpp"
#include "multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp"
Include dependency graph for BayesianModelCheckerTest.hpp:
```



## Classes

- class [multiscaletest::BayesianModelCheckerTest](#)

*Class for testing the Bayesian model checker.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(BayesianModelCheckerTest, CaseTrue\)](#)
- [TEST\\_F \(BayesianModelCheckerTest, CaseFalse\)](#)

### 8.329.1 Function Documentation

#### 8.329.1.1 TEST\_F( BayesianModelCheckerTest , CaseTrue )

Definition at line 85 of file BayesianModelCheckerTest.hpp.

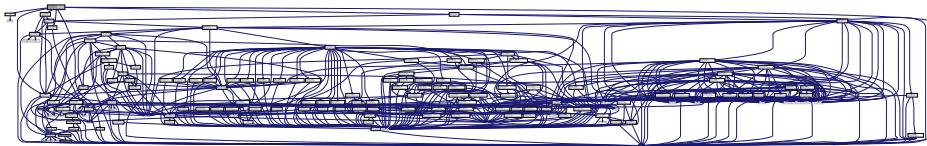
#### 8.329.1.2 TEST\_F( BayesianModelCheckerTest , CaseFalse )

Definition at line 93 of file BayesianModelCheckerTest.hpp.

**8.330 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckerTest.hpp File Reference**

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/exception/TestException.hpp"
#include "multiscale/verification/spatial-temporal/checking/ModelChecker.hpp"
#include "multiscale/verification/spatial-temporal/model/Cluster.hpp"
#include "multiscale/verification/spatial-temporal/model/SpatialTemporal
```

```
Trace.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <string>
Include dependency graph for ModelCheckerTest.hpp:
```



## Classes

- class [multiscaletest::ModelCheckerTest](#)

*Class for testing model checkers.*

## Namespaces

- [multiscaletest](#)

## Variables

- const std::string [INPUT\\_LOGIC\\_PROPERTY](#) = "P > 0.6 [F [0, 3] (avg(clusteredness(clusters)) > 20)]"

### 8.330.1 Variable Documentation

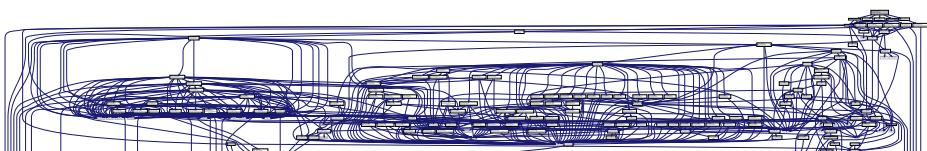
8.330.1.1 const std::string [INPUT\\_LOGIC\\_PROPERTY](#) = "P > 0.6 [F [0, 3] (avg(clusteredness(clusters)) > 20)]"

Definition at line 24 of file ModelCheckerTest.hpp.

Referenced by [multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree\(\)](#).

## 8.331 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckingTest.cpp File Reference

```
#include "ApproximateBayesianModelCheckerTest.hpp"
#include "ApproximateProbabilisticModelCheckerTest.hpp"
#include "BayesianModelCheckerTest.hpp"
#include "ProbabilisticBlackBoxModelCheckerTest.hpp"
#include "StatisticalModelCheckerTest.hpp"
Include dependency graph for ModelCheckingTest.cpp:
```



## Functions

- int [main](#) (int argc, char \*\*argv)

### 8.331.1 Function Documentation

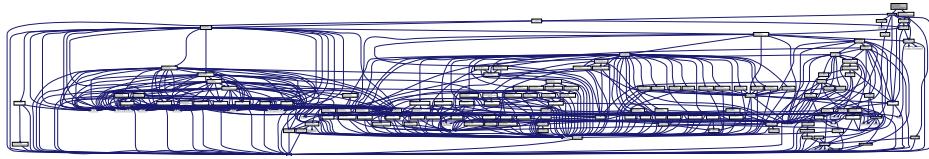
8.331.1.1 int main ( int argc, char \*\* argv )

Definition at line 9 of file ModelCheckingTest.cpp.

## 8.332 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ProbabilisticBlackBoxModelCheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp"
#include "multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelChecker.hpp"
```

Include dependency graph for ProbabilisticBlackBoxModelCheckerTest.hpp:



### Classes

- class [multiscaletest::ProbabilisticBlackBoxModelCheckerTest](#)  
*Class for testing the probabilistic black-box model checker.*

### Namespaces

- [multiscaletest](#)

### Functions

- [TEST\\_F \(ProbabilisticBlackBoxModelCheckerTest, CaseFalse\)](#)

## 8.332.1 Function Documentation

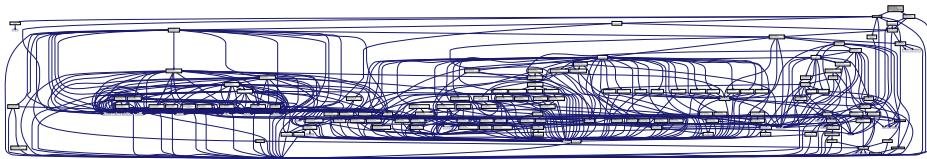
8.332.1.1 TEST\_F ( ProbabilisticBlackBoxModelCheckerTest , CaseFalse )

Definition at line 41 of file ProbabilisticBlackBoxModelCheckerTest.hpp.

## 8.333 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/StatisticalModelCheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp"
#include "multiscale/verification/spatial-temporal/checking/StatisticalModelChecker.hpp"
```

Include dependency graph for StatisticalModelCheckerTest.hpp:



## Classes

- class [multiscaletest::StatisticalModelCheckerTest](#)

*Class for testing the statistical model checker.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(StatisticalModelCheckerTest, CaseTrue\)](#)
- [TEST\\_F \(StatisticalModelCheckerTest, CaseFalse\)](#)

### 8.333.1 Function Documentation

#### 8.333.1.1 TEST\_F( StatisticalModelCheckerTest , CaseTrue )

Definition at line 71 of file StatisticalModelCheckerTest.hpp.

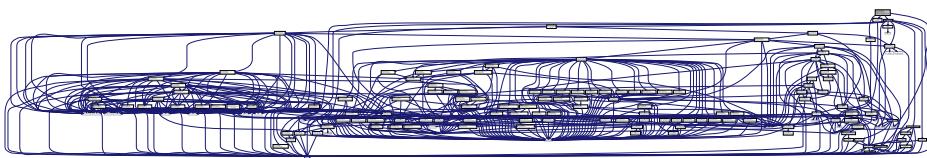
#### 8.333.1.2 TEST\_F( StatisticalModelCheckerTest , CaseFalse )

Definition at line 78 of file StatisticalModelCheckerTest.hpp.

## 8.334 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File Reference

```
#include "TraceEvaluationTest.hpp"
#include "multiscale/verification/spatial-temporal/model/Cluster.hpp"
#include "multiscale/verification/spatial-temporal/model/Region.hpp"
#include <algorithm>
```

Include dependency graph for CompleteTraceTest.hpp:



- class [multiscaletest::CompleteTraceTest](#)

*Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(CompleteTraceTest, BinaryNumericFilter\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureAdd\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureDiv\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureLog\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureMod\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureMultiply\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasurePower\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericMeasureSubtract\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryNumericTemporal\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalQuantileMeasurePercentile\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalQuantileMeasureQuartile\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalQuantileSpatial\)](#)
- [TEST\\_F \(CompleteTraceTest, BinaryStatisticalSpatial\)](#)
- [TEST\\_F \(CompleteTraceTest, ChangeMeasureDifference\)](#)
- [TEST\\_F \(CompleteTraceTest, ChangeMeasureRatio\)](#)
- [TEST\\_F \(CompleteTraceTest, ChangeTemporalNumericCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, ChangeTemporalNumericMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, ComparatorGreaterThan\)](#)
- [TEST\\_F \(CompleteTraceTest, ComparatorLessThan\)](#)
- [TEST\\_F \(CompleteTraceTest, ComparatorGreaterThanOrEqual\)](#)
- [TEST\\_F \(CompleteTraceTest, ComparatorLessThanOrEqual\)](#)
- [TEST\\_F \(CompleteTraceTest, ComparatorEqual\)](#)
- [TEST\\_F \(CompleteTraceTest, CompoundConstraint\)](#)
- [TEST\\_F \(CompleteTraceTest, CompoundConstraintMultiple\)](#)
- [TEST\\_F \(CompleteTraceTest, CompoundLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, CompoundLogicPropertyMultiple\)](#)
- [TEST\\_F \(CompleteTraceTest, ConstraintEnclosedByParentheses\)](#)
- [TEST\\_F \(CompleteTraceTest, ConstraintEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(CompleteTraceTest, ConstraintEnclosedByParenthesesQuadrupled\)](#)
- [TEST\\_F \(CompleteTraceTest, Constraint\)](#)
- [TEST\\_F \(CompleteTraceTest, FilterNumericMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, FilterSubset\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, HeterogeneousTimeseriesComponentPeak\)](#)
- [TEST\\_F \(CompleteTraceTest, HeterogeneousTimeseriesComponentValley\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousHomogeneousTimeseries\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesComponentAscent\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesComponentDescent\)](#)

- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesComponentPlateau\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesComponentUniformAscent\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesComponentUniformDescent\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesMeasureTimeSpan\)](#)
- [TEST\\_F \(CompleteTraceTest, HomogeneousTimeseriesMeasureValue\)](#)
- [TEST\\_F \(CompleteTraceTest, LogicPropertyEnclosedByParentheses\)](#)
- [TEST\\_F \(CompleteTraceTest, LogicPropertyEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(CompleteTraceTest, LogicPropertyEnclosedByParenthesesQuadrupled\)](#)
- [TEST\\_F \(CompleteTraceTest, LogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, MultipleLogicProperties1\)](#)
- [TEST\\_F \(CompleteTraceTest, MultipleLogicProperties2\)](#)
- [TEST\\_F \(CompleteTraceTest, NextKLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, NextLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, NotConstraint\)](#)
- [TEST\\_F \(CompleteTraceTest, NotLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericMeasureCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericSpatialMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWithoutTypes\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableTypeLeft\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableTypeRight\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableBothTypes\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableBothTypesAndDifferentTypeValues\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableOneNumericStateVariable\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWrongRhsType\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWrongName\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWrongLongName\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWrongTypeLhs\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStateVariableWrongTypeLhsLargerValue\)](#)
- [TEST\\_F \(CompleteTraceTest, NumericStatisticalMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, SemanticType\)](#)
- [TEST\\_F \(CompleteTraceTest, SimilarityMeasureAntiSimilar\)](#)
- [TEST\\_F \(CompleteTraceTest, SimilarityMeasureSimilar\)](#)
- [TEST\\_F \(CompleteTraceTest, SimilarityTemporalNumericCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureClusteredness\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureDensity\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureArea\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasurePerimeter\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureDistanceFromOrigin\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureAngle\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureTriangleMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureRectangleMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureCircleMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureCentroidX\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureCentroidY\)](#)
- [TEST\\_F \(CompleteTraceTest, SpatialMeasureCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, Subset\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationDifference\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationDifferenceRegion\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationIntersection\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationIntersectionRegion\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationUnion\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetOperationUnionRegion\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetSpecificClusters\)](#)

Reference

- [TEST\\_F \(CompleteTraceTest, SubsetSpecificRegions\)](#)
- [TEST\\_F \(CompleteTraceTest, SubsetSubsetOperation\)](#)
- [TEST\\_F \(CompleteTraceTest, TemporalNumericCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, TemporalNumericComparison\)](#)
- [TEST\\_F \(CompleteTraceTest, TemporalNumericMeasure\)](#)
- [TEST\\_F \(CompleteTraceTest, TemporalNumericMeasureCollection\)](#)
- [TEST\\_F \(CompleteTraceTest, TimeseriesComponent\)](#)
- [TEST\\_F \(CompleteTraceTest, TimeseriesMeasureEnteringTime\)](#)
- [TEST\\_F \(CompleteTraceTest, TimeseriesMeasureEnteringValue\)](#)
- [TEST\\_F \(CompleteTraceTest, TimeseriesTimeseriesComponent\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericFilter\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureAbs\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureCeil\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureFloor\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureRound\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureSign\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureSqrt\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericMeasureTrunc\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryNumericTemporal\)](#)
- [TEST\\_F \(CompleteTraceTest, UnarySpatialConstraint\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureAvg\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureCount\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureGeomean\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureHarmean\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureKurt\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureMax\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureMedian\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureMin\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureMode\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureProduct\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureSkew\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureStdev\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureSum\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalMeasureVar\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryStatisticalSpatial\)](#)
- [TEST\\_F \(CompleteTraceTest, UnaryTypeConstraint\)](#)
- [TEST\\_F \(CompleteTraceTest, UntilLogicProperty\)](#)
- [TEST\\_F \(CompleteTraceTest, UntilLogicPropertyMultiple\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueReal\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueNumericStateVariable\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueUnaryNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueBinaryNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalConstantValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueReal\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, FutureIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(CompleteTraceTest, GlobalDecreasingValueReal\)](#)

- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, GlobalDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueNumericStateVariable2\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationsValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationsValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, OscillationsValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F\(CompleteTraceTest, EnclosingWithParenthesesDifferently1\)](#)
- [TEST\\_F\(CompleteTraceTest, EnclosingWithParenthesesDifferently2\)](#)
- [TEST\\_F\(CompleteTraceTest, TimeIntervalExceedsTraceEndTime\)](#)
- [TEST\\_F\(CompleteTraceTest, TimeIntervalExceedsTraceStartTime\)](#)
- [TEST\\_F\(CompleteTraceTest, ConstraintsCombinationUnary\)](#)
- [TEST\\_F\(CompleteTraceTest, ConstraintsCombinationBinary\)](#)
- [TEST\\_F\(CompleteTraceTest, ConstraintsCombinationNary\)](#)

### 8.334.1 Function Documentation

#### 8.334.1.1 TEST\_F( CompleteTraceTest , BinaryNumericFilter )

Definition at line 187 of file CompleteTraceTest.hpp.

#### 8.334.1.2 TEST\_F( CompleteTraceTest , BinaryNumericMeasureAdd )

Definition at line 200 of file CompleteTraceTest.hpp.

#### 8.334.1.3 TEST\_F( CompleteTraceTest , BinaryNumericMeasureDiv )

Definition at line 204 of file CompleteTraceTest.hpp.

#### 8.334.1.4 TEST\_F( CompleteTraceTest , BinaryNumericMeasureLog )

Definition at line 208 of file CompleteTraceTest.hpp.

---

8.334.1.5 TEST\_F( CompleteTraceTest , BinaryNumericMeasureMod )

Definition at line 212 of file CompleteTraceTest.hpp.

8.334.1.6 TEST\_F( CompleteTraceTest , BinaryNumericMeasureMultiply )

Definition at line 216 of file CompleteTraceTest.hpp.

8.334.1.7 TEST\_F( CompleteTraceTest , BinaryNumericMeasurePower )

Definition at line 220 of file CompleteTraceTest.hpp.

8.334.1.8 TEST\_F( CompleteTraceTest , BinaryNumericMeasureSubtract )

Definition at line 224 of file CompleteTraceTest.hpp.

8.334.1.9 TEST\_F( CompleteTraceTest , BinaryNumericNumeric )

Definition at line 237 of file CompleteTraceTest.hpp.

8.334.1.10 TEST\_F( CompleteTraceTest , BinaryNumericTemporal )

Definition at line 250 of file CompleteTraceTest.hpp.

8.334.1.11 TEST\_F( CompleteTraceTest , BinaryStatisticalMeasure )

Definition at line 263 of file CompleteTraceTest.hpp.

8.334.1.12 TEST\_F( CompleteTraceTest , BinaryStatisticalNumeric )

Definition at line 276 of file CompleteTraceTest.hpp.

8.334.1.13 TEST\_F( CompleteTraceTest , BinaryStatisticalQuantileMeasurePercentile )

Definition at line 289 of file CompleteTraceTest.hpp.

8.334.1.14 TEST\_F( CompleteTraceTest , BinaryStatisticalQuantileMeasureQuartile )

Definition at line 293 of file CompleteTraceTest.hpp.

8.334.1.15 TEST\_F( CompleteTraceTest , BinaryStatisticalQuantileNumeric )

Definition at line 306 of file CompleteTraceTest.hpp.

8.334.1.16 TEST\_F( CompleteTraceTest , BinaryStatisticalQuantileSpatial )

Definition at line 319 of file CompleteTraceTest.hpp.

**8.334.1.17 TEST\_F( CompleteTraceTest , BinaryStatisticalSpatial )**

Definition at line 332 of file CompleteTraceTest.hpp.

**8.334.1.18 TEST\_F( CompleteTraceTest , ChangeMeasureDifference )**

Definition at line 345 of file CompleteTraceTest.hpp.

**8.334.1.19 TEST\_F( CompleteTraceTest , ChangeMeasureRatio )**

Definition at line 349 of file CompleteTraceTest.hpp.

**8.334.1.20 TEST\_F( CompleteTraceTest , ChangeTemporalNumericCollection )**

Definition at line 362 of file CompleteTraceTest.hpp.

**8.334.1.21 TEST\_F( CompleteTraceTest , ChangeTemporalNumericMeasure )**

Definition at line 375 of file CompleteTraceTest.hpp.

**8.334.1.22 TEST\_F( CompleteTraceTest , ComparatorGreaterThan )**

Definition at line 388 of file CompleteTraceTest.hpp.

**8.334.1.23 TEST\_F( CompleteTraceTest , ComparatorLessThan )**

Definition at line 392 of file CompleteTraceTest.hpp.

**8.334.1.24 TEST\_F( CompleteTraceTest , ComparatorGreaterThanOrEqual )**

Definition at line 396 of file CompleteTraceTest.hpp.

**8.334.1.25 TEST\_F( CompleteTraceTest , ComparatorLessThanOrEqual )**

Definition at line 400 of file CompleteTraceTest.hpp.

**8.334.1.26 TEST\_F( CompleteTraceTest , ComparatorEqual )**

Definition at line 404 of file CompleteTraceTest.hpp.

**8.334.1.27 TEST\_F( CompleteTraceTest , CompoundConstraint )**

Definition at line 417 of file CompleteTraceTest.hpp.

**8.334.1.28 TEST\_F( CompleteTraceTest , CompoundConstraintMultiple )**

Definition at line 424 of file CompleteTraceTest.hpp.

---

8.334.1.29 TEST\_F( CompleteTraceTest , CompoundLogicProperty )

Definition at line 440 of file CompleteTraceTest.hpp.

8.334.1.30 TEST\_F( CompleteTraceTest , CompoundLogicPropertyMultiple )

Definition at line 447 of file CompleteTraceTest.hpp.

8.334.1.31 TEST\_F( CompleteTraceTest , ConstraintEnclosedByParentheses )

Definition at line 463 of file CompleteTraceTest.hpp.

8.334.1.32 TEST\_F( CompleteTraceTest , ConstraintEnclosedByParenthesesDoubled )

Definition at line 467 of file CompleteTraceTest.hpp.

8.334.1.33 TEST\_F( CompleteTraceTest , ConstraintEnclosedByParenthesesQuadrupled )

Definition at line 471 of file CompleteTraceTest.hpp.

8.334.1.34 TEST\_F( CompleteTraceTest , Constraint )

Definition at line 484 of file CompleteTraceTest.hpp.

8.334.1.35 TEST\_F( CompleteTraceTest , FilterNumericMeasure )

Definition at line 497 of file CompleteTraceTest.hpp.

8.334.1.36 TEST\_F( CompleteTraceTest , FilterSubset )

Definition at line 510 of file CompleteTraceTest.hpp.

8.334.1.37 TEST\_F( CompleteTraceTest , FutureLogicProperty )

Definition at line 523 of file CompleteTraceTest.hpp.

8.334.1.38 TEST\_F( CompleteTraceTest , GlobalLogicProperty )

Definition at line 536 of file CompleteTraceTest.hpp.

8.334.1.39 TEST\_F( CompleteTraceTest , HeterogeneousTimeseriesComponentPeak )

Definition at line 549 of file CompleteTraceTest.hpp.

8.334.1.40 TEST\_F( CompleteTraceTest , HeterogeneousTimeseriesComponentValley )

Definition at line 553 of file CompleteTraceTest.hpp.

**8.334.1.41 TEST\_F( CompleteTraceTest , HomogeneousHomogeneousTimeseries )**

Definition at line 566 of file CompleteTraceTest.hpp.

**8.334.1.42 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesComponentAscent )**

Definition at line 579 of file CompleteTraceTest.hpp.

**8.334.1.43 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesComponentDescent )**

Definition at line 583 of file CompleteTraceTest.hpp.

**8.334.1.44 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesComponentPlateau )**

Definition at line 587 of file CompleteTraceTest.hpp.

**8.334.1.45 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesComponentUniformAscent )**

Definition at line 591 of file CompleteTraceTest.hpp.

**8.334.1.46 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesComponentUniformDescent )**

Definition at line 595 of file CompleteTraceTest.hpp.

**8.334.1.47 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesMeasureTimeSpan )**

Definition at line 608 of file CompleteTraceTest.hpp.

**8.334.1.48 TEST\_F( CompleteTraceTest , HomogeneousTimeseriesMeasureValue )**

Definition at line 612 of file CompleteTraceTest.hpp.

**8.334.1.49 TEST\_F( CompleteTraceTest , LogicPropertyEnclosedByParentheses )**

Definition at line 625 of file CompleteTraceTest.hpp.

**8.334.1.50 TEST\_F( CompleteTraceTest , LogicPropertyEnclosedByParenthesesDoubled )**

Definition at line 629 of file CompleteTraceTest.hpp.

**8.334.1.51 TEST\_F( CompleteTraceTest , LogicPropertyEnclosedByParenthesesQuadrupled )**

Definition at line 633 of file CompleteTraceTest.hpp.

**8.334.1.52 TEST\_F( CompleteTraceTest , LogicProperty )**

Definition at line 646 of file CompleteTraceTest.hpp.

---

8.334.1.53 TEST\_F( CompleteTraceTest , MultipleLogicProperties1 )

Definition at line 659 of file CompleteTraceTest.hpp.

8.334.1.54 TEST\_F( CompleteTraceTest , MultipleLogicProperties2 )

Definition at line 663 of file CompleteTraceTest.hpp.

8.334.1.55 TEST\_F( CompleteTraceTest , NextKLogicProperty )

Definition at line 731 of file CompleteTraceTest.hpp.

8.334.1.56 TEST\_F( CompleteTraceTest , NextLogicProperty )

Definition at line 744 of file CompleteTraceTest.hpp.

8.334.1.57 TEST\_F( CompleteTraceTest , NotConstraint )

Definition at line 757 of file CompleteTraceTest.hpp.

8.334.1.58 TEST\_F( CompleteTraceTest , NotLogicProperty )

Definition at line 770 of file CompleteTraceTest.hpp.

8.334.1.59 TEST\_F( CompleteTraceTest , NumericMeasure )

Definition at line 783 of file CompleteTraceTest.hpp.

8.334.1.60 TEST\_F( CompleteTraceTest , NumericMeasureCollection )

Definition at line 796 of file CompleteTraceTest.hpp.

8.334.1.61 TEST\_F( CompleteTraceTest , NumericSpatialMeasure )

Definition at line 809 of file CompleteTraceTest.hpp.

8.334.1.62 TEST\_F( CompleteTraceTest , NumericStateVariableWithoutTypes )

Definition at line 822 of file CompleteTraceTest.hpp.

8.334.1.63 TEST\_F( CompleteTraceTest , NumericStateVariableTypeLeft )

Definition at line 826 of file CompleteTraceTest.hpp.

8.334.1.64 TEST\_F( CompleteTraceTest , NumericStateVariableTypeRight )

Definition at line 830 of file CompleteTraceTest.hpp.

**8.334.1.65 TEST\_F( CompleteTraceTest , NumericStateVariableBothTypes )**

Definition at line 834 of file CompleteTraceTest.hpp.

**8.334.1.66 TEST\_F( CompleteTraceTest , NumericStateVariableBothTypesAndDifferentTypeValues )**

Definition at line 838 of file CompleteTraceTest.hpp.

**8.334.1.67 TEST\_F( CompleteTraceTest , NumericStateVariableOneNumericStateVariable )**

Definition at line 842 of file CompleteTraceTest.hpp.

**8.334.1.68 TEST\_F( CompleteTraceTest , NumericStateVariableWrongRhsType )**

Definition at line 846 of file CompleteTraceTest.hpp.

**8.334.1.69 TEST\_F( CompleteTraceTest , NumericStateVariableWrongName )**

Definition at line 850 of file CompleteTraceTest.hpp.

**8.334.1.70 TEST\_F( CompleteTraceTest , NumericStateVariableWrongLongName )**

Definition at line 854 of file CompleteTraceTest.hpp.

**8.334.1.71 TEST\_F( CompleteTraceTest , NumericStateVariableWrongTypeLhs )**

Definition at line 858 of file CompleteTraceTest.hpp.

**8.334.1.72 TEST\_F( CompleteTraceTest , NumericStateVariableWrongTypeLhsLargerValue )**

Definition at line 862 of file CompleteTraceTest.hpp.

**8.334.1.73 TEST\_F( CompleteTraceTest , NumericStatisticalMeasure )**

Definition at line 875 of file CompleteTraceTest.hpp.

**8.334.1.74 TEST\_F( CompleteTraceTest , ProbabilisticLogicProperty )**

Definition at line 888 of file CompleteTraceTest.hpp.

**8.334.1.75 TEST\_F( CompleteTraceTest , SemanticType )**

Definition at line 901 of file CompleteTraceTest.hpp.

**8.334.1.76 TEST\_F( CompleteTraceTest , SimilarityMeasureAntiSimilar )**

Definition at line 914 of file CompleteTraceTest.hpp.

---

8.334.1.77 TEST\_F( CompleteTraceTest , SimilarityMeasureSimilar )

Definition at line 918 of file CompleteTraceTest.hpp.

8.334.1.78 TEST\_F( CompleteTraceTest , SimilarityTemporalNumericCollection )

Definition at line 931 of file CompleteTraceTest.hpp.

8.334.1.79 TEST\_F( CompleteTraceTest , SpatialMeasureClusteredness )

Definition at line 944 of file CompleteTraceTest.hpp.

8.334.1.80 TEST\_F( CompleteTraceTest , SpatialMeasureDensity )

Definition at line 948 of file CompleteTraceTest.hpp.

8.334.1.81 TEST\_F( CompleteTraceTest , SpatialMeasureArea )

Definition at line 952 of file CompleteTraceTest.hpp.

8.334.1.82 TEST\_F( CompleteTraceTest , SpatialMeasurePerimeter )

Definition at line 956 of file CompleteTraceTest.hpp.

8.334.1.83 TEST\_F( CompleteTraceTest , SpatialMeasureDistanceFromOrigin )

Definition at line 960 of file CompleteTraceTest.hpp.

8.334.1.84 TEST\_F( CompleteTraceTest , SpatialMeasureAngle )

Definition at line 964 of file CompleteTraceTest.hpp.

8.334.1.85 TEST\_F( CompleteTraceTest , SpatialMeasureTriangleMeasure )

Definition at line 968 of file CompleteTraceTest.hpp.

8.334.1.86 TEST\_F( CompleteTraceTest , SpatialMeasureRectangleMeasure )

Definition at line 972 of file CompleteTraceTest.hpp.

8.334.1.87 TEST\_F( CompleteTraceTest , SpatialMeasureCircleMeasure )

Definition at line 976 of file CompleteTraceTest.hpp.

8.334.1.88 TEST\_F( CompleteTraceTest , SpatialMeasureCentroidX )

Definition at line 980 of file CompleteTraceTest.hpp.

**8.334.1.89 TEST\_F( CompleteTraceTest , SpatialMeasureCentroidY )**

Definition at line 984 of file CompleteTraceTest.hpp.

**8.334.1.90 TEST\_F( CompleteTraceTest , SpatialMeasureCollection )**

Definition at line 997 of file CompleteTraceTest.hpp.

**8.334.1.91 TEST\_F( CompleteTraceTest , Subset )**

Definition at line 1010 of file CompleteTraceTest.hpp.

**8.334.1.92 TEST\_F( CompleteTraceTest , SubsetOperationDifference )**

Definition at line 1023 of file CompleteTraceTest.hpp.

**8.334.1.93 TEST\_F( CompleteTraceTest , SubsetOperationDifferenceRegion )**

Definition at line 1027 of file CompleteTraceTest.hpp.

**8.334.1.94 TEST\_F( CompleteTraceTest , SubsetOperationIntersection )**

Definition at line 1031 of file CompleteTraceTest.hpp.

**8.334.1.95 TEST\_F( CompleteTraceTest , SubsetOperationIntersectionRegion )**

Definition at line 1035 of file CompleteTraceTest.hpp.

**8.334.1.96 TEST\_F( CompleteTraceTest , SubsetOperationUnion )**

Definition at line 1039 of file CompleteTraceTest.hpp.

**8.334.1.97 TEST\_F( CompleteTraceTest , SubsetOperationUnionRegion )**

Definition at line 1043 of file CompleteTraceTest.hpp.

**8.334.1.98 TEST\_F( CompleteTraceTest , SubsetSpecificClusters )**

Definition at line 1056 of file CompleteTraceTest.hpp.

**8.334.1.99 TEST\_F( CompleteTraceTest , SubsetSpecificRegions )**

Definition at line 1060 of file CompleteTraceTest.hpp.

**8.334.1.100 TEST\_F( CompleteTraceTest , SubsetSubsetOperation )**

Definition at line 1073 of file CompleteTraceTest.hpp.

---

8.334.1.101 TEST\_F( CompleteTraceTest , TemporalNumericCollection )

Definition at line 1086 of file CompleteTraceTest.hpp.

8.334.1.102 TEST\_F( CompleteTraceTest , TemporalNumericComparison )

Definition at line 1099 of file CompleteTraceTest.hpp.

8.334.1.103 TEST\_F( CompleteTraceTest , TemporalNumericMeasure )

Definition at line 1112 of file CompleteTraceTest.hpp.

8.334.1.104 TEST\_F( CompleteTraceTest , TemporalNumericMeasureCollection )

Definition at line 1125 of file CompleteTraceTest.hpp.

8.334.1.105 TEST\_F( CompleteTraceTest , TimeseriesComponent )

Definition at line 1138 of file CompleteTraceTest.hpp.

8.334.1.106 TEST\_F( CompleteTraceTest , TimeseriesMeasureEnteringTime )

Definition at line 1151 of file CompleteTraceTest.hpp.

8.334.1.107 TEST\_F( CompleteTraceTest , TimeseriesMeasureEnteringValue )

Definition at line 1155 of file CompleteTraceTest.hpp.

8.334.1.108 TEST\_F( CompleteTraceTest , TimeseriesTimeseriesComponent )

Definition at line 1168 of file CompleteTraceTest.hpp.

8.334.1.109 TEST\_F( CompleteTraceTest , UnaryNumericFilter )

Definition at line 1181 of file CompleteTraceTest.hpp.

8.334.1.110 TEST\_F( CompleteTraceTest , UnaryNumericMeasureAbs )

Definition at line 1194 of file CompleteTraceTest.hpp.

8.334.1.111 TEST\_F( CompleteTraceTest , UnaryNumericMeasureCeil )

Definition at line 1198 of file CompleteTraceTest.hpp.

8.334.1.112 TEST\_F( CompleteTraceTest , UnaryNumericMeasureFloor )

Definition at line 1202 of file CompleteTraceTest.hpp.

**8.334.1.113 TEST\_F( CompleteTraceTest , UnaryNumericMeasureRound )**

Definition at line 1206 of file CompleteTraceTest.hpp.

**8.334.1.114 TEST\_F( CompleteTraceTest , UnaryNumericMeasureSign )**

Definition at line 1210 of file CompleteTraceTest.hpp.

**8.334.1.115 TEST\_F( CompleteTraceTest , UnaryNumericMeasureSqrt )**

Definition at line 1214 of file CompleteTraceTest.hpp.

**8.334.1.116 TEST\_F( CompleteTraceTest , UnaryNumericMeasureTrunc )**

Definition at line 1218 of file CompleteTraceTest.hpp.

**8.334.1.117 TEST\_F( CompleteTraceTest , UnaryNumericNumeric )**

Definition at line 1231 of file CompleteTraceTest.hpp.

**8.334.1.118 TEST\_F( CompleteTraceTest , UnaryNumericTemporal )**

Definition at line 1244 of file CompleteTraceTest.hpp.

**8.334.1.119 TEST\_F( CompleteTraceTest , UnarySpatialConstraint )**

Definition at line 1257 of file CompleteTraceTest.hpp.

**8.334.1.120 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureAvg )**

Definition at line 1270 of file CompleteTraceTest.hpp.

**8.334.1.121 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureCount )**

Definition at line 1274 of file CompleteTraceTest.hpp.

**8.334.1.122 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureGeomean )**

Definition at line 1278 of file CompleteTraceTest.hpp.

**8.334.1.123 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureHarmean )**

Definition at line 1282 of file CompleteTraceTest.hpp.

**8.334.1.124 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureKurt )**

Definition at line 1286 of file CompleteTraceTest.hpp.

---

8.334.1.125 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureMax )

Definition at line 1290 of file CompleteTraceTest.hpp.

8.334.1.126 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureMedian )

Definition at line 1294 of file CompleteTraceTest.hpp.

8.334.1.127 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureMin )

Definition at line 1298 of file CompleteTraceTest.hpp.

8.334.1.128 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureMode )

Definition at line 1302 of file CompleteTraceTest.hpp.

8.334.1.129 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureProduct )

Definition at line 1306 of file CompleteTraceTest.hpp.

8.334.1.130 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureSkew )

Definition at line 1310 of file CompleteTraceTest.hpp.

8.334.1.131 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureStdev )

Definition at line 1314 of file CompleteTraceTest.hpp.

8.334.1.132 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureSum )

Definition at line 1318 of file CompleteTraceTest.hpp.

8.334.1.133 TEST\_F( CompleteTraceTest , UnaryStatisticalMeasureVar )

Definition at line 1322 of file CompleteTraceTest.hpp.

8.334.1.134 TEST\_F( CompleteTraceTest , UnaryStatisticalNumeric )

Definition at line 1335 of file CompleteTraceTest.hpp.

8.334.1.135 TEST\_F( CompleteTraceTest , UnaryStatisticalSpatial )

Definition at line 1347 of file CompleteTraceTest.hpp.

8.334.1.136 TEST\_F( CompleteTraceTest , UnaryTypeConstraint )

Definition at line 1360 of file CompleteTraceTest.hpp.

**8.334.1.137 TEST\_F( CompleteTraceTest , UntilLogicProperty )**

Definition at line 1373 of file CompleteTraceTest.hpp.

**8.334.1.138 TEST\_F( CompleteTraceTest , UntilLogicPropertyMultiple )**

Definition at line 1377 of file CompleteTraceTest.hpp.

**8.334.1.139 TEST\_F( CompleteTraceTest , GlobalConstantValueReal )**

Definition at line 1390 of file CompleteTraceTest.hpp.

**8.334.1.140 TEST\_F( CompleteTraceTest , GlobalConstantValueNumericStateVariable )**

Definition at line 1394 of file CompleteTraceTest.hpp.

**8.334.1.141 TEST\_F( CompleteTraceTest , GlobalConstantValueUnaryNumeric )**

Definition at line 1398 of file CompleteTraceTest.hpp.

**8.334.1.142 TEST\_F( CompleteTraceTest , GlobalConstantValueBinaryNumeric )**

Definition at line 1402 of file CompleteTraceTest.hpp.

**8.334.1.143 TEST\_F( CompleteTraceTest , GlobalConstantValueUnaryStatisticalNumeric )**

Definition at line 1406 of file CompleteTraceTest.hpp.

**8.334.1.144 TEST\_F( CompleteTraceTest , GlobalConstantValueBinaryStatisticalNumeric )**

Definition at line 1410 of file CompleteTraceTest.hpp.

**8.334.1.145 TEST\_F( CompleteTraceTest , GlobalConstantValueBinaryStatisticalQuantileNumeric )**

Definition at line 1414 of file CompleteTraceTest.hpp.

**8.334.1.146 TEST\_F( CompleteTraceTest , FutureIncreasingValueReal )**

Definition at line 1427 of file CompleteTraceTest.hpp.

**8.334.1.147 TEST\_F( CompleteTraceTest , FutureIncreasingValueNumericStateVariable )**

Definition at line 1431 of file CompleteTraceTest.hpp.

**8.334.1.148 TEST\_F( CompleteTraceTest , FutureIncreasingValueUnaryNumeric )**

Definition at line 1435 of file CompleteTraceTest.hpp.

8.334.1.149 TEST\_F( CompleteTraceTest , FutureIncreasingValueBinaryNumeric )

Definition at line 1439 of file CompleteTraceTest.hpp.

8.334.1.150 TEST\_F( CompleteTraceTest , FutureIncreasingValueUnaryStatisticalNumeric )

Definition at line 1443 of file CompleteTraceTest.hpp.

8.334.1.151 TEST\_F( CompleteTraceTest , FutureIncreasingValueBinaryStatisticalNumeric )

Definition at line 1447 of file CompleteTraceTest.hpp.

8.334.1.152 TEST\_F( CompleteTraceTest , FutureIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1451 of file CompleteTraceTest.hpp.

8.334.1.153 TEST\_F( CompleteTraceTest , GlobalDecreasingValueReal )

Definition at line 1464 of file CompleteTraceTest.hpp.

8.334.1.154 TEST\_F( CompleteTraceTest , GlobalDecreasingValueNumericStateVariable )

Definition at line 1468 of file CompleteTraceTest.hpp.

8.334.1.155 TEST\_F( CompleteTraceTest , GlobalDecreasingValueUnaryNumeric )

Definition at line 1472 of file CompleteTraceTest.hpp.

8.334.1.156 TEST\_F( CompleteTraceTest , GlobalDecreasingValueBinaryNumeric )

Definition at line 1476 of file CompleteTraceTest.hpp.

8.334.1.157 TEST\_F( CompleteTraceTest , GlobalDecreasingValueUnaryStatisticalNumeric )

Definition at line 1480 of file CompleteTraceTest.hpp.

8.334.1.158 TEST\_F( CompleteTraceTest , GlobalDecreasingValueBinaryStatisticalNumeric )

Definition at line 1484 of file CompleteTraceTest.hpp.

8.334.1.159 TEST\_F( CompleteTraceTest , GlobalDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1488 of file CompleteTraceTest.hpp.

8.334.1.160 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueReal )

Definition at line 1501 of file CompleteTraceTest.hpp.

**8.334.1.161 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueNumericStateVariable )**

Definition at line 1505 of file CompleteTraceTest.hpp.

**8.334.1.162 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueNumericStateVariable2 )**

Definition at line 1509 of file CompleteTraceTest.hpp.

**8.334.1.163 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueUnaryNumeric )**

Definition at line 1513 of file CompleteTraceTest.hpp.

**8.334.1.164 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueBinaryNumeric )**

Definition at line 1517 of file CompleteTraceTest.hpp.

**8.334.1.165 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueUnaryStatisticalNumeric )**

Definition at line 1521 of file CompleteTraceTest.hpp.

**8.334.1.166 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalNumeric )**

Definition at line 1525 of file CompleteTraceTest.hpp.

**8.334.1.167 TEST\_F( CompleteTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric )**

Definition at line 1529 of file CompleteTraceTest.hpp.

**8.334.1.168 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueReal )**

Definition at line 1542 of file CompleteTraceTest.hpp.

**8.334.1.169 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueNumericStateVariable )**

Definition at line 1546 of file CompleteTraceTest.hpp.

**8.334.1.170 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueUnaryNumeric )**

Definition at line 1550 of file CompleteTraceTest.hpp.

**8.334.1.171 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueBinaryNumeric )**

Definition at line 1554 of file CompleteTraceTest.hpp.

**8.334.1.172 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueUnaryStatisticalNumeric )**

Definition at line 1558 of file CompleteTraceTest.hpp.

8.334.1.173 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalNumeric )

Definition at line 1562 of file CompleteTraceTest.hpp.

8.334.1.174 TEST\_F( CompleteTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1566 of file CompleteTraceTest.hpp.

8.334.1.175 TEST\_F( CompleteTraceTest , OscillationValueNumericStateVariable )

Definition at line 1579 of file CompleteTraceTest.hpp.

8.334.1.176 TEST\_F( CompleteTraceTest , OscillationsValueUnaryNumeric )

Definition at line 1583 of file CompleteTraceTest.hpp.

8.334.1.177 TEST\_F( CompleteTraceTest , OscillationsValueBinaryNumeric )

Definition at line 1587 of file CompleteTraceTest.hpp.

8.334.1.178 TEST\_F( CompleteTraceTest , OscillationsValueUnaryStatisticalNumeric )

Definition at line 1591 of file CompleteTraceTest.hpp.

8.334.1.179 TEST\_F( CompleteTraceTest , OscillationsValueBinaryStatisticalNumeric )

Definition at line 1595 of file CompleteTraceTest.hpp.

8.334.1.180 TEST\_F( CompleteTraceTest , OscillationsValueBinaryStatisticalQuantileNumeric )

Definition at line 1599 of file CompleteTraceTest.hpp.

8.334.1.181 TEST\_F( CompleteTraceTest , EnclosingWithParenthesesDifferently1 )

Definition at line 1612 of file CompleteTraceTest.hpp.

8.334.1.182 TEST\_F( CompleteTraceTest , EnclosingWithParenthesesDifferently2 )

Definition at line 1616 of file CompleteTraceTest.hpp.

8.334.1.183 TEST\_F( CompleteTraceTest , TimeIntervalExceedsTraceEndTime )

Definition at line 1629 of file CompleteTraceTest.hpp.

8.334.1.184 TEST\_F( CompleteTraceTest , TimeIntervalExceedsTraceStartTime )

Definition at line 1633 of file CompleteTraceTest.hpp.

### 8.334.1.185 TEST\_F( CompleteTraceTest , ConstraintsCombinationUnary )

Definition at line 1646 of file CompleteTraceTest.hpp.

### 8.334.1.186 TEST\_F( CompleteTraceTest , ConstraintsCombinationBinary )

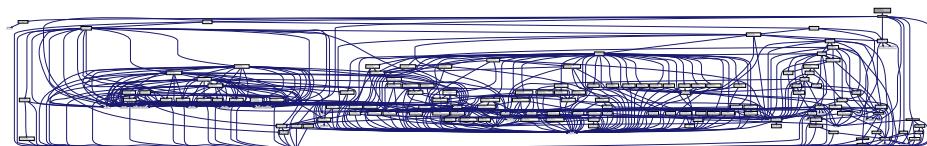
Definition at line 1650 of file CompleteTraceTest.hpp.

### 8.334.1.187 TEST\_F( CompleteTraceTest , ConstraintsCombinationNary )

Definition at line 1654 of file CompleteTraceTest.hpp.

## 8.335 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp File Reference

```
#include "TraceEvaluationTest.hpp"
Include dependency graph for EmptyTraceTest.hpp:
```



### Classes

- class [multiscaletest::EmptyTraceTest](#)  
*Class for testing evaluation of empty traces.*

### Namespaces

- [multiscaletest](#)

### Functions

- [TEST\\_F\(EmptyTraceTest, BinaryNumericFilter\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureAdd\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureDiv\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureLog\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureMod\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureMultiply\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasurePower\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericMeasureSubtract\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericNumeric\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryNumericTemporal\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryStatisticalMeasure\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryStatisticalNumeric\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryStatisticalQuantileMeasurePercentile\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryStatisticalQuantileMeasureQuartile\)](#)
- [TEST\\_F\(EmptyTraceTest, BinaryStatisticalQuantileNumeric\)](#)

- [TEST\\_F \(EmptyTraceTest, BinaryStatisticalQuantileSpatial\)](#)
- [TEST\\_F \(EmptyTraceTest, BinaryStatisticalSpatial\)](#)
- [TEST\\_F \(EmptyTraceTest, ChangeMeasureDifference\)](#)
- [TEST\\_F \(EmptyTraceTest, ChangeMeasureRatio\)](#)
- [TEST\\_F \(EmptyTraceTest, ChangeTemporalNumericCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, ChangeTemporalNumericMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, ComparatorGreaterThan\)](#)
- [TEST\\_F \(EmptyTraceTest, ComparatorLessThan\)](#)
- [TEST\\_F \(EmptyTraceTest, ComparatorGreaterThanOrEqualTo\)](#)
- [TEST\\_F \(EmptyTraceTest, ComparatorLessThanOrEqualTo\)](#)
- [TEST\\_F \(EmptyTraceTest, ComparatorEqual\)](#)
- [TEST\\_F \(EmptyTraceTest, CompoundConstraint\)](#)
- [TEST\\_F \(EmptyTraceTest, CompoundConstraintMultiple\)](#)
- [TEST\\_F \(EmptyTraceTest, CompoundLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, CompoundLogicPropertyMultiple\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintEnclosedByParentheses\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintEnclosedByParenthesesQuadrupled\)](#)
- [TEST\\_F \(EmptyTraceTest, Constraint\)](#)
- [TEST\\_F \(EmptyTraceTest, FilterNumericMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, FilterSubset\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, HeterogeneousTimeseriesComponentPeak\)](#)
- [TEST\\_F \(EmptyTraceTest, HeterogeneousTimeseriesComponentValley\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousHomogeneousTimeseries\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesComponentAscent\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesComponentDescent\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesComponentPlateau\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesComponentUniformAscent\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesComponentUniformDescent\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesMeasureTimeSpan\)](#)
- [TEST\\_F \(EmptyTraceTest, HomogeneousTimeseriesMeasureValue\)](#)
- [TEST\\_F \(EmptyTraceTest, LogicPropertyEnclosedByParentheses\)](#)
- [TEST\\_F \(EmptyTraceTest, LogicPropertyEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(EmptyTraceTest, LogicPropertyEnclosedByParenthesesQuadrupled\)](#)
- [TEST\\_F \(EmptyTraceTest, LogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, MultipleLogicProperties1\)](#)
- [TEST\\_F \(EmptyTraceTest, MultipleLogicProperties2\)](#)
- [TEST\\_F \(EmptyTraceTest, NextKLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, NextLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, NotConstraint\)](#)
- [TEST\\_F \(EmptyTraceTest, NotLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericMeasureCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericSpatialMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWithoutTypes\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableTypeLeft\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableTypeRight\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableBothTypes\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableBothTypesAndDifferentTypeValues\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableOneNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWrongRhsType\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWrongName\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWrongLongName\)](#)

- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWrongTypeLhs\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStateVariableWrongTypeLhsLargerValue\)](#)
- [TEST\\_F \(EmptyTraceTest, NumericStatisticalMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, SemanticType\)](#)
- [TEST\\_F \(EmptyTraceTest, SimilarityMeasureAntiSimilar\)](#)
- [TEST\\_F \(EmptyTraceTest, SimilarityMeasureSimilar\)](#)
- [TEST\\_F \(EmptyTraceTest, SimilarityTemporalNumericCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureClusteredness\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureDensity\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureArea\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasurePerimeter\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureDistanceFromOrigin\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureAngle\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureTriangleMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureRectangleMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureCircleMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureCentroidX\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureCentroidY\)](#)
- [TEST\\_F \(EmptyTraceTest, SpatialMeasureCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, Subset\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetOperationDifference\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetOperationIntersection\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetOperationUnion\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetSpecificClusters\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetSpecificRegions\)](#)
- [TEST\\_F \(EmptyTraceTest, SubsetSubsetOperation\)](#)
- [TEST\\_F \(EmptyTraceTest, TemporalNumericCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, TemporalNumericComparison\)](#)
- [TEST\\_F \(EmptyTraceTest, TemporalNumericMeasure\)](#)
- [TEST\\_F \(EmptyTraceTest, TemporalNumericMeasureCollection\)](#)
- [TEST\\_F \(EmptyTraceTest, TimeseriesComponent\)](#)
- [TEST\\_F \(EmptyTraceTest, TimeseriesMeasureEnteringTime\)](#)
- [TEST\\_F \(EmptyTraceTest, TimeseriesMeasureEnteringValue\)](#)
- [TEST\\_F \(EmptyTraceTest, TimeseriesTimeseriesComponent\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericFilter\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureAbs\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureCeil\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureFloor\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureRound\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureSign\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureSqrt\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericMeasureTrunc\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryNumericTemporal\)](#)
- [TEST\\_F \(EmptyTraceTest, UnarySpatialConstraint\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureAvg\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureCount\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureGeomean\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureHarmean\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureKurt\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureMax\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureMedian\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureMin\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureMode\)](#)

- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureProduct\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureSkew\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureStdDev\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureSum\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalMeasureVar\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryStatisticalSpatial\)](#)
- [TEST\\_F \(EmptyTraceTest, UnaryTypeConstraint\)](#)
- [TEST\\_F \(EmptyTraceTest, UntilLogicProperty\)](#)
- [TEST\\_F \(EmptyTraceTest, UntilLogicPropertyMultiple\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueReal\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalConstantValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueReal\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, FutureIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, GlobalDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueNumericStateVariable2\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationsValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationsValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, OscillationsValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(EmptyTraceTest, EnclosingWithParenthesesDifferently1\)](#)
- [TEST\\_F \(EmptyTraceTest, EnclosingWithParenthesesDifferently2\)](#)
- [TEST\\_F \(EmptyTraceTest, TimeIntervalExceedsTraceEndTime\)](#)

- [TEST\\_F \(EmptyTraceTest, TimeIntervalExceedsTraceStartTime\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintsCombinationUnary\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintsCombinationBinary\)](#)
- [TEST\\_F \(EmptyTraceTest, ConstraintsCombinationNary\)](#)

### 8.335.1 Function Documentation

#### 8.335.1.1 TEST\_F( EmptyTraceTest , BinaryNumericFilter )

Definition at line 46 of file EmptyTraceTest.hpp.

#### 8.335.1.2 TEST\_F( EmptyTraceTest , BinaryNumericMeasureAdd )

Definition at line 59 of file EmptyTraceTest.hpp.

#### 8.335.1.3 TEST\_F( EmptyTraceTest , BinaryNumericMeasureDiv )

Definition at line 63 of file EmptyTraceTest.hpp.

#### 8.335.1.4 TEST\_F( EmptyTraceTest , BinaryNumericMeasureLog )

Definition at line 67 of file EmptyTraceTest.hpp.

#### 8.335.1.5 TEST\_F( EmptyTraceTest , BinaryNumericMeasureMod )

Definition at line 71 of file EmptyTraceTest.hpp.

#### 8.335.1.6 TEST\_F( EmptyTraceTest , BinaryNumericMeasureMultiply )

Definition at line 75 of file EmptyTraceTest.hpp.

#### 8.335.1.7 TEST\_F( EmptyTraceTest , BinaryNumericMeasurePower )

Definition at line 79 of file EmptyTraceTest.hpp.

#### 8.335.1.8 TEST\_F( EmptyTraceTest , BinaryNumericMeasureSubtract )

Definition at line 83 of file EmptyTraceTest.hpp.

#### 8.335.1.9 TEST\_F( EmptyTraceTest , BinaryNumericNumeric )

Definition at line 96 of file EmptyTraceTest.hpp.

#### 8.335.1.10 TEST\_F( EmptyTraceTest , BinaryNumericTemporal )

Definition at line 109 of file EmptyTraceTest.hpp.

#### 8.335.1.11 TEST\_F( EmptyTraceTest , BinaryStatisticalMeasure )

Definition at line 122 of file EmptyTraceTest.hpp.

---

8.335.1.12 TEST\_F( EmptyTraceTest , BinaryStatisticalNumeric )

Definition at line 135 of file EmptyTraceTest.hpp.

8.335.1.13 TEST\_F( EmptyTraceTest , BinaryStatisticalQuantileMeasurePercentile )

Definition at line 148 of file EmptyTraceTest.hpp.

8.335.1.14 TEST\_F( EmptyTraceTest , BinaryStatisticalQuantileMeasureQuartile )

Definition at line 152 of file EmptyTraceTest.hpp.

8.335.1.15 TEST\_F( EmptyTraceTest , BinaryStatisticalQuantileNumeric )

Definition at line 165 of file EmptyTraceTest.hpp.

8.335.1.16 TEST\_F( EmptyTraceTest , BinaryStatisticalQuantileSpatial )

Definition at line 178 of file EmptyTraceTest.hpp.

8.335.1.17 TEST\_F( EmptyTraceTest , BinaryStatisticalSpatial )

Definition at line 191 of file EmptyTraceTest.hpp.

8.335.1.18 TEST\_F( EmptyTraceTest , ChangeMeasureDifference )

Definition at line 204 of file EmptyTraceTest.hpp.

8.335.1.19 TEST\_F( EmptyTraceTest , ChangeMeasureRatio )

Definition at line 208 of file EmptyTraceTest.hpp.

8.335.1.20 TEST\_F( EmptyTraceTest , ChangeTemporalNumericCollection )

Definition at line 221 of file EmptyTraceTest.hpp.

8.335.1.21 TEST\_F( EmptyTraceTest , ChangeTemporalNumericMeasure )

Definition at line 234 of file EmptyTraceTest.hpp.

8.335.1.22 TEST\_F( EmptyTraceTest , ComparatorGreaterThan )

Definition at line 247 of file EmptyTraceTest.hpp.

8.335.1.23 TEST\_F( EmptyTraceTest , ComparatorLessThan )

Definition at line 251 of file EmptyTraceTest.hpp.

**8.335.1.24 TEST\_F( EmptyTraceTest , ComparatorGreaterThanOrEqualTo )**

Definition at line 255 of file EmptyTraceTest.hpp.

**8.335.1.25 TEST\_F( EmptyTraceTest , ComparatorLessThanOrEqualTo )**

Definition at line 259 of file EmptyTraceTest.hpp.

**8.335.1.26 TEST\_F( EmptyTraceTest , ComparatorEqual )**

Definition at line 263 of file EmptyTraceTest.hpp.

**8.335.1.27 TEST\_F( EmptyTraceTest , CompoundConstraint )**

Definition at line 276 of file EmptyTraceTest.hpp.

References CONSTRAINTS\_BINARY\_OPERATORS.

**8.335.1.28 TEST\_F( EmptyTraceTest , CompoundConstraintMultiple )**

Definition at line 284 of file EmptyTraceTest.hpp.

References CONSTRAINTS\_BINARY\_OPERATORS.

**8.335.1.29 TEST\_F( EmptyTraceTest , CompoundLogicProperty )**

Definition at line 301 of file EmptyTraceTest.hpp.

References LOGIC\_PROPERTIES\_BINARY\_OPERATORS.

**8.335.1.30 TEST\_F( EmptyTraceTest , CompoundLogicPropertyMultiple )**

Definition at line 309 of file EmptyTraceTest.hpp.

References LOGIC\_PROPERTIES\_BINARY\_OPERATORS.

**8.335.1.31 TEST\_F( EmptyTraceTest , ConstraintEnclosedByParentheses )**

Definition at line 326 of file EmptyTraceTest.hpp.

**8.335.1.32 TEST\_F( EmptyTraceTest , ConstraintEnclosedByParenthesesDoubled )**

Definition at line 330 of file EmptyTraceTest.hpp.

**8.335.1.33 TEST\_F( EmptyTraceTest , ConstraintEnclosedByParenthesesQuadrupled )**

Definition at line 334 of file EmptyTraceTest.hpp.

**8.335.1.34 TEST\_F( EmptyTraceTest , Constraint )**

Definition at line 347 of file EmptyTraceTest.hpp.

---

8.335.1.35 TEST\_F( EmptyTraceTest , FilterNumericMeasure )

Definition at line 360 of file EmptyTraceTest.hpp.

8.335.1.36 TEST\_F( EmptyTraceTest , FilterSubset )

Definition at line 373 of file EmptyTraceTest.hpp.

8.335.1.37 TEST\_F( EmptyTraceTest , FutureLogicProperty )

Definition at line 386 of file EmptyTraceTest.hpp.

8.335.1.38 TEST\_F( EmptyTraceTest , GlobalLogicProperty )

Definition at line 399 of file EmptyTraceTest.hpp.

8.335.1.39 TEST\_F( EmptyTraceTest , HeterogeneousTimeseriesComponentPeak )

Definition at line 412 of file EmptyTraceTest.hpp.

8.335.1.40 TEST\_F( EmptyTraceTest , HeterogeneousTimeseriesComponentValley )

Definition at line 416 of file EmptyTraceTest.hpp.

8.335.1.41 TEST\_F( EmptyTraceTest , HomogeneousHomogeneousTimeseries )

Definition at line 429 of file EmptyTraceTest.hpp.

8.335.1.42 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesComponentAscent )

Definition at line 442 of file EmptyTraceTest.hpp.

8.335.1.43 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesComponentDescent )

Definition at line 446 of file EmptyTraceTest.hpp.

8.335.1.44 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesComponentPlateau )

Definition at line 450 of file EmptyTraceTest.hpp.

8.335.1.45 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesComponentUniformAscent )

Definition at line 454 of file EmptyTraceTest.hpp.

8.335.1.46 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesComponentUniformDescent )

Definition at line 458 of file EmptyTraceTest.hpp.

**8.335.1.47 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesMeasureTimeSpan )**

Definition at line 471 of file EmptyTraceTest.hpp.

**8.335.1.48 TEST\_F( EmptyTraceTest , HomogeneousTimeseriesMeasureValue )**

Definition at line 475 of file EmptyTraceTest.hpp.

**8.335.1.49 TEST\_F( EmptyTraceTest , LogicPropertyEnclosedByParentheses )**

Definition at line 488 of file EmptyTraceTest.hpp.

**8.335.1.50 TEST\_F( EmptyTraceTest , LogicPropertyEnclosedByParenthesesDoubled )**

Definition at line 492 of file EmptyTraceTest.hpp.

**8.335.1.51 TEST\_F( EmptyTraceTest , LogicPropertyEnclosedByParenthesesQuadrupled )**

Definition at line 496 of file EmptyTraceTest.hpp.

**8.335.1.52 TEST\_F( EmptyTraceTest , LogicProperty )**

Definition at line 509 of file EmptyTraceTest.hpp.

**8.335.1.53 TEST\_F( EmptyTraceTest , MultipleLogicProperties1 )**

Definition at line 522 of file EmptyTraceTest.hpp.

**8.335.1.54 TEST\_F( EmptyTraceTest , MultipleLogicProperties2 )**

Definition at line 526 of file EmptyTraceTest.hpp.

**8.335.1.55 TEST\_F( EmptyTraceTest , NextKLogicProperty )**

Definition at line 595 of file EmptyTraceTest.hpp.

**8.335.1.56 TEST\_F( EmptyTraceTest , NextLogicProperty )**

Definition at line 608 of file EmptyTraceTest.hpp.

**8.335.1.57 TEST\_F( EmptyTraceTest , NotConstraint )**

Definition at line 621 of file EmptyTraceTest.hpp.

**8.335.1.58 TEST\_F( EmptyTraceTest , NotLogicProperty )**

Definition at line 634 of file EmptyTraceTest.hpp.

---

8.335.1.59 TEST\_F( EmptyTraceTest , NumericMeasure )

Definition at line 647 of file EmptyTraceTest.hpp.

8.335.1.60 TEST\_F( EmptyTraceTest , NumericMeasureCollection )

Definition at line 660 of file EmptyTraceTest.hpp.

8.335.1.61 TEST\_F( EmptyTraceTest , NumericSpatialMeasure )

Definition at line 673 of file EmptyTraceTest.hpp.

8.335.1.62 TEST\_F( EmptyTraceTest , NumericStateVariableWithoutTypes )

Definition at line 686 of file EmptyTraceTest.hpp.

8.335.1.63 TEST\_F( EmptyTraceTest , NumericStateVariableTypeLeft )

Definition at line 690 of file EmptyTraceTest.hpp.

8.335.1.64 TEST\_F( EmptyTraceTest , NumericStateVariableTypeRight )

Definition at line 694 of file EmptyTraceTest.hpp.

8.335.1.65 TEST\_F( EmptyTraceTest , NumericStateVariableBothTypes )

Definition at line 698 of file EmptyTraceTest.hpp.

8.335.1.66 TEST\_F( EmptyTraceTest , NumericStateVariableBothTypesAndDifferentTypeValues )

Definition at line 702 of file EmptyTraceTest.hpp.

8.335.1.67 TEST\_F( EmptyTraceTest , NumericStateVariableOneNumericStateVariable )

Definition at line 706 of file EmptyTraceTest.hpp.

8.335.1.68 TEST\_F( EmptyTraceTest , NumericStateVariableWrongRhsType )

Definition at line 710 of file EmptyTraceTest.hpp.

8.335.1.69 TEST\_F( EmptyTraceTest , NumericStateVariableWrongName )

Definition at line 714 of file EmptyTraceTest.hpp.

8.335.1.70 TEST\_F( EmptyTraceTest , NumericStateVariableWrongLongName )

Definition at line 718 of file EmptyTraceTest.hpp.

**8.335.1.71 TEST\_F( EmptyTraceTest , NumericStateVariableWrongTypeLhs )**

Definition at line 722 of file EmptyTraceTest.hpp.

**8.335.1.72 TEST\_F( EmptyTraceTest , NumericStateVariableWrongTypeLhsLargerValue )**

Definition at line 726 of file EmptyTraceTest.hpp.

**8.335.1.73 TEST\_F( EmptyTraceTest , NumericStatisticalMeasure )**

Definition at line 739 of file EmptyTraceTest.hpp.

**8.335.1.74 TEST\_F( EmptyTraceTest , ProbabilisticLogicProperty )**

Definition at line 752 of file EmptyTraceTest.hpp.

**8.335.1.75 TEST\_F( EmptyTraceTest , SemanticType )**

Definition at line 765 of file EmptyTraceTest.hpp.

**8.335.1.76 TEST\_F( EmptyTraceTest , SimilarityMeasureAntiSimilar )**

Definition at line 778 of file EmptyTraceTest.hpp.

**8.335.1.77 TEST\_F( EmptyTraceTest , SimilarityMeasureSimilar )**

Definition at line 782 of file EmptyTraceTest.hpp.

**8.335.1.78 TEST\_F( EmptyTraceTest , SimilarityTemporalNumericCollection )**

Definition at line 795 of file EmptyTraceTest.hpp.

**8.335.1.79 TEST\_F( EmptyTraceTest , SpatialMeasureClusteredness )**

Definition at line 808 of file EmptyTraceTest.hpp.

**8.335.1.80 TEST\_F( EmptyTraceTest , SpatialMeasureDensity )**

Definition at line 812 of file EmptyTraceTest.hpp.

**8.335.1.81 TEST\_F( EmptyTraceTest , SpatialMeasureArea )**

Definition at line 816 of file EmptyTraceTest.hpp.

**8.335.1.82 TEST\_F( EmptyTraceTest , SpatialMeasurePerimeter )**

Definition at line 820 of file EmptyTraceTest.hpp.

---

8.335.1.83 TEST\_F( EmptyTraceTest , SpatialMeasureDistanceFromOrigin )

Definition at line 824 of file EmptyTraceTest.hpp.

8.335.1.84 TEST\_F( EmptyTraceTest , SpatialMeasureAngle )

Definition at line 828 of file EmptyTraceTest.hpp.

8.335.1.85 TEST\_F( EmptyTraceTest , SpatialMeasureTriangleMeasure )

Definition at line 832 of file EmptyTraceTest.hpp.

8.335.1.86 TEST\_F( EmptyTraceTest , SpatialMeasureRectangleMeasure )

Definition at line 836 of file EmptyTraceTest.hpp.

8.335.1.87 TEST\_F( EmptyTraceTest , SpatialMeasureCircleMeasure )

Definition at line 840 of file EmptyTraceTest.hpp.

8.335.1.88 TEST\_F( EmptyTraceTest , SpatialMeasureCentroidX )

Definition at line 844 of file EmptyTraceTest.hpp.

8.335.1.89 TEST\_F( EmptyTraceTest , SpatialMeasureCentroidY )

Definition at line 848 of file EmptyTraceTest.hpp.

8.335.1.90 TEST\_F( EmptyTraceTest , SpatialMeasureCollection )

Definition at line 861 of file EmptyTraceTest.hpp.

8.335.1.91 TEST\_F( EmptyTraceTest , Subset )

Definition at line 874 of file EmptyTraceTest.hpp.

8.335.1.92 TEST\_F( EmptyTraceTest , SubsetOperationDifference )

Definition at line 887 of file EmptyTraceTest.hpp.

8.335.1.93 TEST\_F( EmptyTraceTest , SubsetOperationIntersection )

Definition at line 891 of file EmptyTraceTest.hpp.

8.335.1.94 TEST\_F( EmptyTraceTest , SubsetOperationUnion )

Definition at line 895 of file EmptyTraceTest.hpp.

**8.335.1.95 TEST\_F( EmptyTraceTest , SubsetSpecificClusters )**

Definition at line 908 of file EmptyTraceTest.hpp.

**8.335.1.96 TEST\_F( EmptyTraceTest , SubsetSpecificRegions )**

Definition at line 912 of file EmptyTraceTest.hpp.

**8.335.1.97 TEST\_F( EmptyTraceTest , SubsetSubsetOperation )**

Definition at line 925 of file EmptyTraceTest.hpp.

**8.335.1.98 TEST\_F( EmptyTraceTest , TemporalNumericCollection )**

Definition at line 938 of file EmptyTraceTest.hpp.

**8.335.1.99 TEST\_F( EmptyTraceTest , TemporalNumericComparison )**

Definition at line 951 of file EmptyTraceTest.hpp.

**8.335.1.100 TEST\_F( EmptyTraceTest , TemporalNumericMeasure )**

Definition at line 964 of file EmptyTraceTest.hpp.

**8.335.1.101 TEST\_F( EmptyTraceTest , TemporalNumericMeasureCollection )**

Definition at line 977 of file EmptyTraceTest.hpp.

**8.335.1.102 TEST\_F( EmptyTraceTest , TimeseriesComponent )**

Definition at line 990 of file EmptyTraceTest.hpp.

**8.335.1.103 TEST\_F( EmptyTraceTest , TimeseriesMeasureEnteringTime )**

Definition at line 1003 of file EmptyTraceTest.hpp.

**8.335.1.104 TEST\_F( EmptyTraceTest , TimeseriesMeasureEnteringValue )**

Definition at line 1007 of file EmptyTraceTest.hpp.

**8.335.1.105 TEST\_F( EmptyTraceTest , TimeseriesTimeseriesComponent )**

Definition at line 1020 of file EmptyTraceTest.hpp.

**8.335.1.106 TEST\_F( EmptyTraceTest , UnaryNumericFilter )**

Definition at line 1033 of file EmptyTraceTest.hpp.

---

8.335.1.107 TEST\_F( EmptyTraceTest , UnaryNumericMeasureAbs )

Definition at line 1046 of file EmptyTraceTest.hpp.

8.335.1.108 TEST\_F( EmptyTraceTest , UnaryNumericMeasureCeil )

Definition at line 1050 of file EmptyTraceTest.hpp.

8.335.1.109 TEST\_F( EmptyTraceTest , UnaryNumericMeasureFloor )

Definition at line 1054 of file EmptyTraceTest.hpp.

8.335.1.110 TEST\_F( EmptyTraceTest , UnaryNumericMeasureRound )

Definition at line 1058 of file EmptyTraceTest.hpp.

8.335.1.111 TEST\_F( EmptyTraceTest , UnaryNumericMeasureSign )

Definition at line 1062 of file EmptyTraceTest.hpp.

8.335.1.112 TEST\_F( EmptyTraceTest , UnaryNumericMeasureSqrt )

Definition at line 1066 of file EmptyTraceTest.hpp.

8.335.1.113 TEST\_F( EmptyTraceTest , UnaryNumericMeasureTrunc )

Definition at line 1070 of file EmptyTraceTest.hpp.

8.335.1.114 TEST\_F( EmptyTraceTest , UnaryNumericNumeric )

Definition at line 1083 of file EmptyTraceTest.hpp.

8.335.1.115 TEST\_F( EmptyTraceTest , UnaryNumericTemporal )

Definition at line 1096 of file EmptyTraceTest.hpp.

8.335.1.116 TEST\_F( EmptyTraceTest , UnarySpatialConstraint )

Definition at line 1109 of file EmptyTraceTest.hpp.

8.335.1.117 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureAvg )

Definition at line 1122 of file EmptyTraceTest.hpp.

8.335.1.118 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureCount )

Definition at line 1126 of file EmptyTraceTest.hpp.

8.335.1.119 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureGeomean )

Definition at line 1130 of file EmptyTraceTest.hpp.

8.335.1.120 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureHarmean )

Definition at line 1134 of file EmptyTraceTest.hpp.

8.335.1.121 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureKurt )

Definition at line 1138 of file EmptyTraceTest.hpp.

8.335.1.122 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureMax )

Definition at line 1142 of file EmptyTraceTest.hpp.

8.335.1.123 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureMedian )

Definition at line 1146 of file EmptyTraceTest.hpp.

8.335.1.124 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureMin )

Definition at line 1150 of file EmptyTraceTest.hpp.

8.335.1.125 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureMode )

Definition at line 1154 of file EmptyTraceTest.hpp.

8.335.1.126 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureProduct )

Definition at line 1158 of file EmptyTraceTest.hpp.

8.335.1.127 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureSkew )

Definition at line 1162 of file EmptyTraceTest.hpp.

8.335.1.128 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureStdev )

Definition at line 1166 of file EmptyTraceTest.hpp.

8.335.1.129 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureSum )

Definition at line 1170 of file EmptyTraceTest.hpp.

8.335.1.130 TEST\_F( EmptyTraceTest , UnaryStatisticalMeasureVar )

Definition at line 1174 of file EmptyTraceTest.hpp.

---

8.335.1.131 TEST\_F( EmptyTraceTest , UnaryStatisticalNumeric )

Definition at line 1187 of file EmptyTraceTest.hpp.

8.335.1.132 TEST\_F( EmptyTraceTest , UnaryStatisticalSpatial )

Definition at line 1199 of file EmptyTraceTest.hpp.

8.335.1.133 TEST\_F( EmptyTraceTest , UnaryTypeConstraint )

Definition at line 1212 of file EmptyTraceTest.hpp.

8.335.1.134 TEST\_F( EmptyTraceTest , UntilLogicProperty )

Definition at line 1225 of file EmptyTraceTest.hpp.

8.335.1.135 TEST\_F( EmptyTraceTest , UntilLogicPropertyMultiple )

Definition at line 1229 of file EmptyTraceTest.hpp.

8.335.1.136 TEST\_F( EmptyTraceTest , GlobalConstantValueReal )

Definition at line 1242 of file EmptyTraceTest.hpp.

8.335.1.137 TEST\_F( EmptyTraceTest , GlobalConstantValueNumericStateVariable )

Definition at line 1246 of file EmptyTraceTest.hpp.

8.335.1.138 TEST\_F( EmptyTraceTest , GlobalConstantValueUnaryNumeric )

Definition at line 1250 of file EmptyTraceTest.hpp.

8.335.1.139 TEST\_F( EmptyTraceTest , GlobalConstantValueBinaryNumeric )

Definition at line 1254 of file EmptyTraceTest.hpp.

8.335.1.140 TEST\_F( EmptyTraceTest , GlobalConstantValueUnaryStatisticalNumeric )

Definition at line 1258 of file EmptyTraceTest.hpp.

8.335.1.141 TEST\_F( EmptyTraceTest , GlobalConstantValueBinaryStatisticalNumeric )

Definition at line 1262 of file EmptyTraceTest.hpp.

8.335.1.142 TEST\_F( EmptyTraceTest , GlobalConstantValueBinaryStatisticalQuantileNumeric )

Definition at line 1266 of file EmptyTraceTest.hpp.

8.335.1.143 TEST\_F( EmptyTraceTest , FutureIncreasingValueReal )

Definition at line 1279 of file EmptyTraceTest.hpp.

8.335.1.144 TEST\_F( EmptyTraceTest , FutureIncreasingValueNumericStateVariable )

Definition at line 1283 of file EmptyTraceTest.hpp.

8.335.1.145 TEST\_F( EmptyTraceTest , FutureIncreasingValueUnaryNumeric )

Definition at line 1287 of file EmptyTraceTest.hpp.

8.335.1.146 TEST\_F( EmptyTraceTest , FutureIncreasingValueBinaryNumeric )

Definition at line 1291 of file EmptyTraceTest.hpp.

8.335.1.147 TEST\_F( EmptyTraceTest , FutureIncreasingValueUnaryStatisticalNumeric )

Definition at line 1295 of file EmptyTraceTest.hpp.

8.335.1.148 TEST\_F( EmptyTraceTest , FutureIncreasingValueBinaryStatisticalNumeric )

Definition at line 1299 of file EmptyTraceTest.hpp.

8.335.1.149 TEST\_F( EmptyTraceTest , FutureIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1303 of file EmptyTraceTest.hpp.

8.335.1.150 TEST\_F( EmptyTraceTest , GlobalDecreasingValueReal )

Definition at line 1316 of file EmptyTraceTest.hpp.

8.335.1.151 TEST\_F( EmptyTraceTest , GlobalDecreasingValueNumericStateVariable )

Definition at line 1320 of file EmptyTraceTest.hpp.

8.335.1.152 TEST\_F( EmptyTraceTest , GlobalDecreasingValueUnaryNumeric )

Definition at line 1324 of file EmptyTraceTest.hpp.

8.335.1.153 TEST\_F( EmptyTraceTest , GlobalDecreasingValueBinaryNumeric )

Definition at line 1328 of file EmptyTraceTest.hpp.

8.335.1.154 TEST\_F( EmptyTraceTest , GlobalDecreasingValueUnaryStatisticalNumeric )

Definition at line 1332 of file EmptyTraceTest.hpp.

8.335.1.155 TEST\_F( EmptyTraceTest , GlobalDecreasingValueBinaryStatisticalNumeric )

Definition at line 1336 of file EmptyTraceTest.hpp.

8.335.1.156 TEST\_F( EmptyTraceTest , GlobalDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1340 of file EmptyTraceTest.hpp.

8.335.1.157 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueReal )

Definition at line 1353 of file EmptyTraceTest.hpp.

8.335.1.158 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueNumericStateVariable )

Definition at line 1357 of file EmptyTraceTest.hpp.

8.335.1.159 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueNumericStateVariable2 )

Definition at line 1361 of file EmptyTraceTest.hpp.

8.335.1.160 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueUnaryNumeric )

Definition at line 1365 of file EmptyTraceTest.hpp.

8.335.1.161 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueBinaryNumeric )

Definition at line 1369 of file EmptyTraceTest.hpp.

8.335.1.162 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueUnaryStatisticalNumeric )

Definition at line 1373 of file EmptyTraceTest.hpp.

8.335.1.163 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalNumeric )

Definition at line 1377 of file EmptyTraceTest.hpp.

8.335.1.164 TEST\_F( EmptyTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1381 of file EmptyTraceTest.hpp.

8.335.1.165 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueReal )

Definition at line 1394 of file EmptyTraceTest.hpp.

8.335.1.166 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueNumericStateVariable )

Definition at line 1398 of file EmptyTraceTest.hpp.

8.335.1.167 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueUnaryNumeric )

Definition at line 1402 of file EmptyTraceTest.hpp.

8.335.1.168 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueBinaryNumeric )

Definition at line 1406 of file EmptyTraceTest.hpp.

8.335.1.169 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueUnaryStatisticalNumeric )

Definition at line 1410 of file EmptyTraceTest.hpp.

8.335.1.170 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalNumeric )

Definition at line 1414 of file EmptyTraceTest.hpp.

8.335.1.171 TEST\_F( EmptyTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1418 of file EmptyTraceTest.hpp.

8.335.1.172 TEST\_F( EmptyTraceTest , OscillationValueNumericStateVariable )

Definition at line 1431 of file EmptyTraceTest.hpp.

8.335.1.173 TEST\_F( EmptyTraceTest , OscillationsValueUnaryNumeric )

Definition at line 1435 of file EmptyTraceTest.hpp.

8.335.1.174 TEST\_F( EmptyTraceTest , OscillationsValueBinaryNumeric )

Definition at line 1439 of file EmptyTraceTest.hpp.

8.335.1.175 TEST\_F( EmptyTraceTest , OscillationsValueUnaryStatisticalNumeric )

Definition at line 1443 of file EmptyTraceTest.hpp.

8.335.1.176 TEST\_F( EmptyTraceTest , OscillationsValueBinaryStatisticalNumeric )

Definition at line 1447 of file EmptyTraceTest.hpp.

8.335.1.177 TEST\_F( EmptyTraceTest , OscillationsValueBinaryStatisticalQuantileNumeric )

Definition at line 1451 of file EmptyTraceTest.hpp.

8.335.1.178 TEST\_F( EmptyTraceTest , EnclosingWithParenthesesDifferently1 )

Definition at line 1464 of file EmptyTraceTest.hpp.

8.335.1.179 TEST\_F( EmptyTraceTest , EnclosingWithParenthesesDifferently2 )

Definition at line 1468 of file EmptyTraceTest.hpp.

8.335.1.180 TEST\_F( EmptyTraceTest , TimeIntervalExceedsTraceEndTime )

Definition at line 1481 of file EmptyTraceTest.hpp.

8.335.1.181 TEST\_F( EmptyTraceTest , TimeIntervalExceedsTraceStartTime )

Definition at line 1485 of file EmptyTraceTest.hpp.

8.335.1.182 TEST\_F( EmptyTraceTest , ConstraintsCombinationUnary )

Definition at line 1498 of file EmptyTraceTest.hpp.

8.335.1.183 TEST\_F( EmptyTraceTest , ConstraintsCombinationBinary )

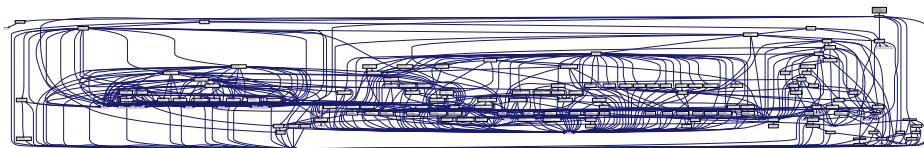
Definition at line 1502 of file EmptyTraceTest.hpp.

8.335.1.184 TEST\_F( EmptyTraceTest , ConstraintsCombinationNary )

Definition at line 1506 of file EmptyTraceTest.hpp.

## 8.336 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File Reference

```
#include "TraceEvaluationTest.hpp"
Include dependency graph for NumericStateVariableTraceTest.hpp:
```



### Classes

- class [multiscaletest::NumericStateVariableTraceTest](#)

*Class for testing evaluation of numeric state variable-only traces.*

### Namespaces

- [multiscaletest](#)

## Functions

- `TEST_F (NumericStateVariableTraceTest, BinaryNumericFilter)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureAdd)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureDiv)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureLog)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureMod)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureMultiply)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasurePower)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureSubtract)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericNumeric)`
- `TEST_F (NumericStateVariableTraceTest, BinaryNumericTemporal)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalMeasure)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalNumeric)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalQuantileMeasurePercentile)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalQuantileMeasureQuartile)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalQuantileNumeric)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalQuantileSpatial)`
- `TEST_F (NumericStateVariableTraceTest, BinaryStatisticalSpatial)`
- `TEST_F (NumericStateVariableTraceTest, ChangeMeasureDifference)`
- `TEST_F (NumericStateVariableTraceTest, ChangeMeasureRatio)`
- `TEST_F (NumericStateVariableTraceTest, ChangeTemporalNumericCollection)`
- `TEST_F (NumericStateVariableTraceTest, ChangeTemporalNumericMeasure)`
- `TEST_F (NumericStateVariableTraceTest, ComparatorGreaterThan)`
- `TEST_F (NumericStateVariableTraceTest, ComparatorLessThan)`
- `TEST_F (NumericStateVariableTraceTest, ComparatorGreaterThanOrEqualTo)`
- `TEST_F (NumericStateVariableTraceTest, ComparatorLessThanOrEqualTo)`
- `TEST_F (NumericStateVariableTraceTest, ComparatorEqual)`
- `TEST_F (NumericStateVariableTraceTest, CompoundConstraint)`
- `TEST_F (NumericStateVariableTraceTest, CompoundConstraintMultiple)`
- `TEST_F (NumericStateVariableTraceTest, CompoundLogicProperty)`
- `TEST_F (NumericStateVariableTraceTest, CompoundLogicPropertyMultiple)`
- `TEST_F (NumericStateVariableTraceTest, ConstraintEnclosedByParentheses)`
- `TEST_F (NumericStateVariableTraceTest, ConstraintEnclosedByParenthesesDoubled)`
- `TEST_F (NumericStateVariableTraceTest, ConstraintEnclosedByParenthesesQuadrupled)`
- `TEST_F (NumericStateVariableTraceTest, Constraint)`
- `TEST_F (NumericStateVariableTraceTest, FilterNumericMeasure)`
- `TEST_F (NumericStateVariableTraceTest, FilterSubset)`
- `TEST_F (NumericStateVariableTraceTest, FutureLogicProperty)`
- `TEST_F (NumericStateVariableTraceTest, GlobalLogicProperty)`
- `TEST_F (NumericStateVariableTraceTest, HeterogeneousTimeseriesComponentPeak)`
- `TEST_F (NumericStateVariableTraceTest, HeterogeneousTimeseriesComponentValley)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousHomogeneousTimeseries)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesComponentAscent)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesComponentDescent)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesComponentPlateau)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesComponentUniformAscent)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesComponentUniformDescent)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesMeasureTimeSpan)`
- `TEST_F (NumericStateVariableTraceTest, HomogeneousTimeseriesMeasureValue)`
- `TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParentheses)`
- `TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParenthesesDoubled)`
- `TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParenthesesQuadrupled)`
- `TEST_F (NumericStateVariableTraceTest, LogicProperty)`
- `TEST_F (NumericStateVariableTraceTest, MultipleLogicProperties1)`

Reference

- [TEST\\_F \(NumericStateVariableTraceTest, MultipleLogicProperties2\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NextKLogicProperty\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NextLogicProperty\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NotConstraint\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NotLogicProperty\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericMeasureCollection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericSpatialMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWithoutTypes\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableTypeLeft\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableTypeRight\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableBothTypes\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableBothTypesAndDifferentTypeValues\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableOneNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWrongRhsType\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWrongName\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWrongLongName\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWrongTypeLhs\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStateVariableWrongTypeLhsLargerValue\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, NumericStatisticalMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SemanticType\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SimilarityMeasureAntiSimilar\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SimilarityMeasureSimilar\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SimilarityTemporalNumericCollection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureClusteredness\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureDensity\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureArea\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasurePerimeter\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureDistanceFromOrigin\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureAngle\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureTriangleMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureRectangleMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureCircleMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureCentroidX\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureCentroidY\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SpatialMeasureCollection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, Subset\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationDifference\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationDifferenceRegion\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationIntersection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationIntersectionRegion\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationUnion\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetOperationUnionRegion\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetSpecificClusters\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetSpecificRegions\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, SubsetSubsetOperation\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TemporalNumericCollection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TemporalNumericComparison\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TemporalNumericMeasure\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TemporalNumericMeasureCollection\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeseriesComponent\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeseriesMeasureEnteringTime\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeseriesMeasureEnteringValue\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeseriesTimeseriesComponent\)](#)

- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericFilter\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureAbs\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureCeil\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureFloor\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureRound\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureSign\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureSqrt\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericMeasureTrunc\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryNumericTemporal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnarySpatialConstraint\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureAvg\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureCount\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureGeomean\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureHarmean\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureKurt\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureMax\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureMedian\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureMin\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureMode\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureProduct\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureSkew\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureStdev\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureSum\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalMeasureVar\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UnaryStatisticalSpatial\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UntilLogicProperty\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, UntilLogicPropertyMultiple\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueReal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueUnaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalConstantValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueReal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, FutureIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, GlobalDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueNumericStateVariable2\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)

- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationsValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationsValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, OscillationsValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, EnclosingWithParenthesesDifferently1\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, EnclosingWithParenthesesDifferently2\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeIntervalExceedsTraceEndTime\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, TimeIntervalExceedsTraceStartTime\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, ConstraintsCombinationUnary\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, ConstraintsCombinationBinary\)](#)
- [TEST\\_F \(NumericStateVariableTraceTest, ConstraintsCombinationNary\)](#)

### 8.336.1 Function Documentation

#### 8.336.1.1 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericFilter )

Definition at line 115 of file NumericStateVariableTraceTest.hpp.

#### 8.336.1.2 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericMeasureAdd )

Definition at line 128 of file NumericStateVariableTraceTest.hpp.

#### 8.336.1.3 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericMeasureDiv )

Definition at line 132 of file NumericStateVariableTraceTest.hpp.

#### 8.336.1.4 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericMeasureLog )

Definition at line 136 of file NumericStateVariableTraceTest.hpp.

#### 8.336.1.5 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericMeasureMod )

Definition at line 140 of file NumericStateVariableTraceTest.hpp.

#### 8.336.1.6 TEST\_F ( NumericStateVariableTraceTest , BinaryNumericMeasureMultiply )

Definition at line 144 of file NumericStateVariableTraceTest.hpp.

**8.336.1.7 TEST\_F( NumericStateVariableTraceTest , BinaryNumericMeasurePower )**

Definition at line 148 of file NumericStateVariableTraceTest.hpp.

**8.336.1.8 TEST\_F( NumericStateVariableTraceTest , BinaryNumericMeasureSubtract )**

Definition at line 152 of file NumericStateVariableTraceTest.hpp.

**8.336.1.9 TEST\_F( NumericStateVariableTraceTest , BinaryNumericNumeric )**

Definition at line 165 of file NumericStateVariableTraceTest.hpp.

**8.336.1.10 TEST\_F( NumericStateVariableTraceTest , BinaryNumericTemporal )**

Definition at line 178 of file NumericStateVariableTraceTest.hpp.

**8.336.1.11 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalMeasure )**

Definition at line 191 of file NumericStateVariableTraceTest.hpp.

**8.336.1.12 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalNumeric )**

Definition at line 204 of file NumericStateVariableTraceTest.hpp.

**8.336.1.13 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalQuantileMeasurePercentile )**

Definition at line 217 of file NumericStateVariableTraceTest.hpp.

**8.336.1.14 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalQuantileMeasureQuartile )**

Definition at line 221 of file NumericStateVariableTraceTest.hpp.

**8.336.1.15 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalQuantileNumeric )**

Definition at line 234 of file NumericStateVariableTraceTest.hpp.

**8.336.1.16 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalQuantileSpatial )**

Definition at line 247 of file NumericStateVariableTraceTest.hpp.

**8.336.1.17 TEST\_F( NumericStateVariableTraceTest , BinaryStatisticalSpatial )**

Definition at line 260 of file NumericStateVariableTraceTest.hpp.

**8.336.1.18 TEST\_F( NumericStateVariableTraceTest , ChangeMeasureDifference )**

Definition at line 273 of file NumericStateVariableTraceTest.hpp.

---

8.336.1.19 TEST\_F( NumericStateVariableTraceTest , ChangeMeasureRatio )

Definition at line 277 of file NumericStateVariableTraceTest.hpp.

8.336.1.20 TEST\_F( NumericStateVariableTraceTest , ChangeTemporalNumericCollection )

Definition at line 290 of file NumericStateVariableTraceTest.hpp.

8.336.1.21 TEST\_F( NumericStateVariableTraceTest , ChangeTemporalNumericMeasure )

Definition at line 303 of file NumericStateVariableTraceTest.hpp.

8.336.1.22 TEST\_F( NumericStateVariableTraceTest , ComparatorGreaterThan )

Definition at line 316 of file NumericStateVariableTraceTest.hpp.

8.336.1.23 TEST\_F( NumericStateVariableTraceTest , ComparatorLessThan )

Definition at line 320 of file NumericStateVariableTraceTest.hpp.

8.336.1.24 TEST\_F( NumericStateVariableTraceTest , ComparatorGreaterThanOrEqual )

Definition at line 324 of file NumericStateVariableTraceTest.hpp.

8.336.1.25 TEST\_F( NumericStateVariableTraceTest , ComparatorLessThanOrEqual )

Definition at line 328 of file NumericStateVariableTraceTest.hpp.

8.336.1.26 TEST\_F( NumericStateVariableTraceTest , ComparatorEqual )

Definition at line 332 of file NumericStateVariableTraceTest.hpp.

8.336.1.27 TEST\_F( NumericStateVariableTraceTest , CompoundConstraint )

Definition at line 345 of file NumericStateVariableTraceTest.hpp.

8.336.1.28 TEST\_F( NumericStateVariableTraceTest , CompoundConstraintMultiple )

Definition at line 352 of file NumericStateVariableTraceTest.hpp.

8.336.1.29 TEST\_F( NumericStateVariableTraceTest , CompoundLogicProperty )

Definition at line 368 of file NumericStateVariableTraceTest.hpp.

8.336.1.30 TEST\_F( NumericStateVariableTraceTest , CompoundLogicPropertyMultiple )

Definition at line 375 of file NumericStateVariableTraceTest.hpp.

**8.336.1.31 TEST\_F( NumericStateVariableTraceTest , ConstraintEnclosedByParentheses )**

Definition at line 391 of file NumericStateVariableTraceTest.hpp.

**8.336.1.32 TEST\_F( NumericStateVariableTraceTest , ConstraintEnclosedByParenthesesDoubled )**

Definition at line 395 of file NumericStateVariableTraceTest.hpp.

**8.336.1.33 TEST\_F( NumericStateVariableTraceTest , ConstraintEnclosedByParenthesesQuadrupled )**

Definition at line 399 of file NumericStateVariableTraceTest.hpp.

**8.336.1.34 TEST\_F( NumericStateVariableTraceTest , Constraint )**

Definition at line 412 of file NumericStateVariableTraceTest.hpp.

**8.336.1.35 TEST\_F( NumericStateVariableTraceTest , FilterNumericMeasure )**

Definition at line 425 of file NumericStateVariableTraceTest.hpp.

**8.336.1.36 TEST\_F( NumericStateVariableTraceTest , FilterSubset )**

Definition at line 438 of file NumericStateVariableTraceTest.hpp.

**8.336.1.37 TEST\_F( NumericStateVariableTraceTest , FutureLogicProperty )**

Definition at line 451 of file NumericStateVariableTraceTest.hpp.

**8.336.1.38 TEST\_F( NumericStateVariableTraceTest , GlobalLogicProperty )**

Definition at line 464 of file NumericStateVariableTraceTest.hpp.

**8.336.1.39 TEST\_F( NumericStateVariableTraceTest , HeterogeneousTimeseriesComponentPeak )**

Definition at line 477 of file NumericStateVariableTraceTest.hpp.

**8.336.1.40 TEST\_F( NumericStateVariableTraceTest , HeterogeneousTimeseriesComponentValley )**

Definition at line 481 of file NumericStateVariableTraceTest.hpp.

**8.336.1.41 TEST\_F( NumericStateVariableTraceTest , HomogeneousHomogeneousTimeseries )**

Definition at line 494 of file NumericStateVariableTraceTest.hpp.

**8.336.1.42 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesComponentAscent )**

Definition at line 507 of file NumericStateVariableTraceTest.hpp.

---

8.336.1.43 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesComponentDescent )

Definition at line 511 of file NumericStateVariableTraceTest.hpp.

8.336.1.44 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesComponentPlateau )

Definition at line 515 of file NumericStateVariableTraceTest.hpp.

8.336.1.45 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesComponentUniformAscent )

Definition at line 519 of file NumericStateVariableTraceTest.hpp.

8.336.1.46 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesComponentUniformDescent )

Definition at line 523 of file NumericStateVariableTraceTest.hpp.

8.336.1.47 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesMeasureTimeSpan )

Definition at line 536 of file NumericStateVariableTraceTest.hpp.

8.336.1.48 TEST\_F( NumericStateVariableTraceTest , HomogeneousTimeseriesMeasureValue )

Definition at line 540 of file NumericStateVariableTraceTest.hpp.

8.336.1.49 TEST\_F( NumericStateVariableTraceTest , LogicPropertyEnclosedByParentheses )

Definition at line 553 of file NumericStateVariableTraceTest.hpp.

8.336.1.50 TEST\_F( NumericStateVariableTraceTest , LogicPropertyEnclosedByParenthesesDoubled )

Definition at line 557 of file NumericStateVariableTraceTest.hpp.

8.336.1.51 TEST\_F( NumericStateVariableTraceTest , LogicPropertyEnclosedByParenthesesQuadrupled )

Definition at line 561 of file NumericStateVariableTraceTest.hpp.

8.336.1.52 TEST\_F( NumericStateVariableTraceTest , LogicProperty )

Definition at line 574 of file NumericStateVariableTraceTest.hpp.

8.336.1.53 TEST\_F( NumericStateVariableTraceTest , MultipleLogicProperties1 )

Definition at line 587 of file NumericStateVariableTraceTest.hpp.

8.336.1.54 TEST\_F( NumericStateVariableTraceTest , MultipleLogicProperties2 )

Definition at line 591 of file NumericStateVariableTraceTest.hpp.

**8.336.1.55 TEST\_F( NumericStateVariableTraceTest , NextKLogicProperty )**

Definition at line 659 of file NumericStateVariableTraceTest.hpp.

**8.336.1.56 TEST\_F( NumericStateVariableTraceTest , NextLogicProperty )**

Definition at line 672 of file NumericStateVariableTraceTest.hpp.

**8.336.1.57 TEST\_F( NumericStateVariableTraceTest , NotConstraint )**

Definition at line 685 of file NumericStateVariableTraceTest.hpp.

**8.336.1.58 TEST\_F( NumericStateVariableTraceTest , NotLogicProperty )**

Definition at line 698 of file NumericStateVariableTraceTest.hpp.

**8.336.1.59 TEST\_F( NumericStateVariableTraceTest , NumericMeasure )**

Definition at line 711 of file NumericStateVariableTraceTest.hpp.

**8.336.1.60 TEST\_F( NumericStateVariableTraceTest , NumericMeasureCollection )**

Definition at line 724 of file NumericStateVariableTraceTest.hpp.

**8.336.1.61 TEST\_F( NumericStateVariableTraceTest , NumericSpatialMeasure )**

Definition at line 737 of file NumericStateVariableTraceTest.hpp.

**8.336.1.62 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWithoutTypes )**

Definition at line 750 of file NumericStateVariableTraceTest.hpp.

**8.336.1.63 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableTypeLeft )**

Definition at line 754 of file NumericStateVariableTraceTest.hpp.

**8.336.1.64 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableTypeRight )**

Definition at line 758 of file NumericStateVariableTraceTest.hpp.

**8.336.1.65 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableBothTypes )**

Definition at line 762 of file NumericStateVariableTraceTest.hpp.

**8.336.1.66 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableBothTypesAndDifferentTypeValues )**

Definition at line 766 of file NumericStateVariableTraceTest.hpp.

---

8.336.1.67 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableOneNumericStateVariable )

Definition at line 770 of file NumericStateVariableTraceTest.hpp.

8.336.1.68 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWrongRhsType )

Definition at line 774 of file NumericStateVariableTraceTest.hpp.

8.336.1.69 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWrongName )

Definition at line 778 of file NumericStateVariableTraceTest.hpp.

8.336.1.70 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWrongLongName )

Definition at line 782 of file NumericStateVariableTraceTest.hpp.

8.336.1.71 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWrongTypeLhs )

Definition at line 786 of file NumericStateVariableTraceTest.hpp.

8.336.1.72 TEST\_F( NumericStateVariableTraceTest , NumericStateVariableWrongTypeLhsLargerValue )

Definition at line 790 of file NumericStateVariableTraceTest.hpp.

8.336.1.73 TEST\_F( NumericStateVariableTraceTest , NumericStatisticalMeasure )

Definition at line 803 of file NumericStateVariableTraceTest.hpp.

8.336.1.74 TEST\_F( NumericStateVariableTraceTest , ProbabilisticLogicProperty )

Definition at line 816 of file NumericStateVariableTraceTest.hpp.

8.336.1.75 TEST\_F( NumericStateVariableTraceTest , SemanticType )

Definition at line 829 of file NumericStateVariableTraceTest.hpp.

8.336.1.76 TEST\_F( NumericStateVariableTraceTest , SimilarityMeasureAntiSimilar )

Definition at line 842 of file NumericStateVariableTraceTest.hpp.

8.336.1.77 TEST\_F( NumericStateVariableTraceTest , SimilarityMeasureSimilar )

Definition at line 846 of file NumericStateVariableTraceTest.hpp.

8.336.1.78 TEST\_F( NumericStateVariableTraceTest , SimilarityTemporalNumericCollection )

Definition at line 859 of file NumericStateVariableTraceTest.hpp.

**8.336.1.79 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureClusteredness )**

Definition at line 872 of file NumericStateVariableTraceTest.hpp.

**8.336.1.80 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureDensity )**

Definition at line 876 of file NumericStateVariableTraceTest.hpp.

**8.336.1.81 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureArea )**

Definition at line 880 of file NumericStateVariableTraceTest.hpp.

**8.336.1.82 TEST\_F( NumericStateVariableTraceTest , SpatialMeasurePerimeter )**

Definition at line 884 of file NumericStateVariableTraceTest.hpp.

**8.336.1.83 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureDistanceFromOrigin )**

Definition at line 888 of file NumericStateVariableTraceTest.hpp.

**8.336.1.84 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureAngle )**

Definition at line 892 of file NumericStateVariableTraceTest.hpp.

**8.336.1.85 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureTriangleMeasure )**

Definition at line 896 of file NumericStateVariableTraceTest.hpp.

**8.336.1.86 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureRectangleMeasure )**

Definition at line 900 of file NumericStateVariableTraceTest.hpp.

**8.336.1.87 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureCircleMeasure )**

Definition at line 904 of file NumericStateVariableTraceTest.hpp.

**8.336.1.88 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureCentroidX )**

Definition at line 908 of file NumericStateVariableTraceTest.hpp.

**8.336.1.89 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureCentroidY )**

Definition at line 912 of file NumericStateVariableTraceTest.hpp.

**8.336.1.90 TEST\_F( NumericStateVariableTraceTest , SpatialMeasureCollection )**

Definition at line 926 of file NumericStateVariableTraceTest.hpp.

---

8.336.1.91 TEST\_F( NumericStateVariableTraceTest , Subset )

Definition at line 939 of file NumericStateVariableTraceTest.hpp.

8.336.1.92 TEST\_F( NumericStateVariableTraceTest , SubsetOperationDifference )

Definition at line 952 of file NumericStateVariableTraceTest.hpp.

8.336.1.93 TEST\_F( NumericStateVariableTraceTest , SubsetOperationDifferenceRegion )

Definition at line 956 of file NumericStateVariableTraceTest.hpp.

8.336.1.94 TEST\_F( NumericStateVariableTraceTest , SubsetOperationIntersection )

Definition at line 960 of file NumericStateVariableTraceTest.hpp.

8.336.1.95 TEST\_F( NumericStateVariableTraceTest , SubsetOperationIntersectionRegion )

Definition at line 964 of file NumericStateVariableTraceTest.hpp.

8.336.1.96 TEST\_F( NumericStateVariableTraceTest , SubsetOperationUnion )

Definition at line 968 of file NumericStateVariableTraceTest.hpp.

8.336.1.97 TEST\_F( NumericStateVariableTraceTest , SubsetOperationUnionRegion )

Definition at line 972 of file NumericStateVariableTraceTest.hpp.

8.336.1.98 TEST\_F( NumericStateVariableTraceTest , SubsetSpecificClusters )

Definition at line 985 of file NumericStateVariableTraceTest.hpp.

8.336.1.99 TEST\_F( NumericStateVariableTraceTest , SubsetSpecificRegions )

Definition at line 989 of file NumericStateVariableTraceTest.hpp.

8.336.1.100 TEST\_F( NumericStateVariableTraceTest , SubsetSubsetOperation )

Definition at line 1002 of file NumericStateVariableTraceTest.hpp.

8.336.1.101 TEST\_F( NumericStateVariableTraceTest , TemporalNumericCollection )

Definition at line 1015 of file NumericStateVariableTraceTest.hpp.

8.336.1.102 TEST\_F( NumericStateVariableTraceTest , TemporalNumericComparison )

Definition at line 1028 of file NumericStateVariableTraceTest.hpp.

**8.336.1.103 TEST\_F( NumericStateVariableTraceTest , TemporalNumericMeasure )**

Definition at line 1041 of file NumericStateVariableTraceTest.hpp.

**8.336.1.104 TEST\_F( NumericStateVariableTraceTest , TemporalNumericMeasureCollection )**

Definition at line 1054 of file NumericStateVariableTraceTest.hpp.

**8.336.1.105 TEST\_F( NumericStateVariableTraceTest , TimeseriesComponent )**

Definition at line 1067 of file NumericStateVariableTraceTest.hpp.

**8.336.1.106 TEST\_F( NumericStateVariableTraceTest , TimeseriesMeasureEnteringTime )**

Definition at line 1080 of file NumericStateVariableTraceTest.hpp.

**8.336.1.107 TEST\_F( NumericStateVariableTraceTest , TimeseriesMeasureEnteringValue )**

Definition at line 1084 of file NumericStateVariableTraceTest.hpp.

**8.336.1.108 TEST\_F( NumericStateVariableTraceTest , TimeseriesTimeseriesComponent )**

Definition at line 1097 of file NumericStateVariableTraceTest.hpp.

**8.336.1.109 TEST\_F( NumericStateVariableTraceTest , UnaryNumericFilter )**

Definition at line 1110 of file NumericStateVariableTraceTest.hpp.

**8.336.1.110 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureAbs )**

Definition at line 1123 of file NumericStateVariableTraceTest.hpp.

**8.336.1.111 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureCeil )**

Definition at line 1127 of file NumericStateVariableTraceTest.hpp.

**8.336.1.112 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureFloor )**

Definition at line 1131 of file NumericStateVariableTraceTest.hpp.

**8.336.1.113 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureRound )**

Definition at line 1135 of file NumericStateVariableTraceTest.hpp.

**8.336.1.114 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureSign )**

Definition at line 1139 of file NumericStateVariableTraceTest.hpp.

---

8.336.1.115 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureSqrt )

Definition at line 1143 of file NumericStateVariableTraceTest.hpp.

8.336.1.116 TEST\_F( NumericStateVariableTraceTest , UnaryNumericMeasureTrunc )

Definition at line 1147 of file NumericStateVariableTraceTest.hpp.

8.336.1.117 TEST\_F( NumericStateVariableTraceTest , UnaryNumericNumeric )

Definition at line 1160 of file NumericStateVariableTraceTest.hpp.

8.336.1.118 TEST\_F( NumericStateVariableTraceTest , UnaryNumericTemporal )

Definition at line 1173 of file NumericStateVariableTraceTest.hpp.

8.336.1.119 TEST\_F( NumericStateVariableTraceTest , UnarySpatialConstraint )

Definition at line 1186 of file NumericStateVariableTraceTest.hpp.

8.336.1.120 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureAvg )

Definition at line 1199 of file NumericStateVariableTraceTest.hpp.

8.336.1.121 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureCount )

Definition at line 1203 of file NumericStateVariableTraceTest.hpp.

8.336.1.122 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureGeomean )

Definition at line 1207 of file NumericStateVariableTraceTest.hpp.

8.336.1.123 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureHarmean )

Definition at line 1211 of file NumericStateVariableTraceTest.hpp.

8.336.1.124 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureKurt )

Definition at line 1215 of file NumericStateVariableTraceTest.hpp.

8.336.1.125 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureMax )

Definition at line 1219 of file NumericStateVariableTraceTest.hpp.

8.336.1.126 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureMedian )

Definition at line 1223 of file NumericStateVariableTraceTest.hpp.

**8.336.1.127 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureMin )**

Definition at line 1227 of file NumericStateVariableTraceTest.hpp.

**8.336.1.128 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureMode )**

Definition at line 1231 of file NumericStateVariableTraceTest.hpp.

**8.336.1.129 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureProduct )**

Definition at line 1235 of file NumericStateVariableTraceTest.hpp.

**8.336.1.130 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureSkew )**

Definition at line 1239 of file NumericStateVariableTraceTest.hpp.

**8.336.1.131 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureStdev )**

Definition at line 1243 of file NumericStateVariableTraceTest.hpp.

**8.336.1.132 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureSum )**

Definition at line 1247 of file NumericStateVariableTraceTest.hpp.

**8.336.1.133 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalMeasureVar )**

Definition at line 1251 of file NumericStateVariableTraceTest.hpp.

**8.336.1.134 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalNumeric )**

Definition at line 1264 of file NumericStateVariableTraceTest.hpp.

**8.336.1.135 TEST\_F( NumericStateVariableTraceTest , UnaryStatisticalSpatial )**

Definition at line 1277 of file NumericStateVariableTraceTest.hpp.

**8.336.1.136 TEST\_F( NumericStateVariableTraceTest , UnaryTypeConstraint )**

Definition at line 1290 of file NumericStateVariableTraceTest.hpp.

**8.336.1.137 TEST\_F( NumericStateVariableTraceTest , UntilLogicProperty )**

Definition at line 1303 of file NumericStateVariableTraceTest.hpp.

**8.336.1.138 TEST\_F( NumericStateVariableTraceTest , UntilLogicPropertyMultiple )**

Definition at line 1307 of file NumericStateVariableTraceTest.hpp.

8.336.1.139 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueReal )

Definition at line 1320 of file NumericStateVariableTraceTest.hpp.

8.336.1.140 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueNumericStateVariable )

Definition at line 1324 of file NumericStateVariableTraceTest.hpp.

8.336.1.141 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueUnaryNumeric )

Definition at line 1328 of file NumericStateVariableTraceTest.hpp.

8.336.1.142 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueBinaryNumeric )

Definition at line 1332 of file NumericStateVariableTraceTest.hpp.

8.336.1.143 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueUnaryStatisticalNumeric )

Definition at line 1336 of file NumericStateVariableTraceTest.hpp.

8.336.1.144 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueBinaryStatisticalNumeric )

Definition at line 1340 of file NumericStateVariableTraceTest.hpp.

8.336.1.145 TEST\_F( NumericStateVariableTraceTest , GlobalConstantValueBinaryStatisticalQuantileNumeric )

Definition at line 1344 of file NumericStateVariableTraceTest.hpp.

8.336.1.146 TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueReal )

Definition at line 1357 of file NumericStateVariableTraceTest.hpp.

8.336.1.147 TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueNumericStateVariable )

Definition at line 1361 of file NumericStateVariableTraceTest.hpp.

8.336.1.148 TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueUnaryNumeric )

Definition at line 1365 of file NumericStateVariableTraceTest.hpp.

8.336.1.149 TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueBinaryNumeric )

Definition at line 1369 of file NumericStateVariableTraceTest.hpp.

8.336.1.150 TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueUnaryStatisticalNumeric )

Definition at line 1373 of file NumericStateVariableTraceTest.hpp.

8.336.1.151 **TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueBinaryStatisticalNumeric )**

Definition at line 1377 of file NumericStateVariableTraceTest.hpp.

8.336.1.152 **TEST\_F( NumericStateVariableTraceTest , FutureIncreasingValueBinaryStatisticalQuantileNumeric )**

Definition at line 1381 of file NumericStateVariableTraceTest.hpp.

8.336.1.153 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueReal )**

Definition at line 1394 of file NumericStateVariableTraceTest.hpp.

8.336.1.154 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueNumericStateVariable )**

Definition at line 1398 of file NumericStateVariableTraceTest.hpp.

8.336.1.155 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueUnaryNumeric )**

Definition at line 1402 of file NumericStateVariableTraceTest.hpp.

8.336.1.156 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueBinaryNumeric )**

Definition at line 1406 of file NumericStateVariableTraceTest.hpp.

8.336.1.157 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueUnaryStatisticalNumeric )**

Definition at line 1410 of file NumericStateVariableTraceTest.hpp.

8.336.1.158 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueBinaryStatisticalNumeric )**

Definition at line 1414 of file NumericStateVariableTraceTest.hpp.

8.336.1.159 **TEST\_F( NumericStateVariableTraceTest , GlobalDecreasingValueBinaryStatisticalQuantileNumeric )**

Definition at line 1418 of file NumericStateVariableTraceTest.hpp.

8.336.1.160 **TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueReal )**

Definition at line 1431 of file NumericStateVariableTraceTest.hpp.

8.336.1.161 **TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueNumericStateVariable )**

Definition at line 1435 of file NumericStateVariableTraceTest.hpp.

8.336.1.162 **TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueNumericStateVariable2 )**

Definition at line 1439 of file NumericStateVariableTraceTest.hpp.

8.336.1.163 TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueUnaryNumeric )

Definition at line 1443 of file NumericStateVariableTraceTest.hpp.

8.336.1.164 TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueBinaryNumeric )

Definition at line 1447 of file NumericStateVariableTraceTest.hpp.

8.336.1.165 TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueUnaryStatisticalNumeric )

Definition at line 1451 of file NumericStateVariableTraceTest.hpp.

8.336.1.166 TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalNumeric )

Definition at line 1455 of file NumericStateVariableTraceTest.hpp.

8.336.1.167 TEST\_F( NumericStateVariableTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1459 of file NumericStateVariableTraceTest.hpp.

8.336.1.168 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueReal )

Definition at line 1472 of file NumericStateVariableTraceTest.hpp.

8.336.1.169 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueNumericStateVariable )

Definition at line 1476 of file NumericStateVariableTraceTest.hpp.

8.336.1.170 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueUnaryNumeric )

Definition at line 1480 of file NumericStateVariableTraceTest.hpp.

8.336.1.171 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueBinaryNumeric )

Definition at line 1484 of file NumericStateVariableTraceTest.hpp.

8.336.1.172 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueUnaryStatisticalNumeric )

Definition at line 1488 of file NumericStateVariableTraceTest.hpp.

8.336.1.173 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalNumeric )

Definition at line 1492 of file NumericStateVariableTraceTest.hpp.

8.336.1.174 TEST\_F( NumericStateVariableTraceTest , DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1496 of file NumericStateVariableTraceTest.hpp.

**8.336.1.175 TEST\_F( NumericStateVariableTraceTest , OscillationValueNumericStateVariable )**

Definition at line 1509 of file NumericStateVariableTraceTest.hpp.

**8.336.1.176 TEST\_F( NumericStateVariableTraceTest , OscillationsValueUnaryNumeric )**

Definition at line 1513 of file NumericStateVariableTraceTest.hpp.

**8.336.1.177 TEST\_F( NumericStateVariableTraceTest , OscillationsValueBinaryNumeric )**

Definition at line 1517 of file NumericStateVariableTraceTest.hpp.

**8.336.1.178 TEST\_F( NumericStateVariableTraceTest , OscillationsValueUnaryStatisticalNumeric )**

Definition at line 1521 of file NumericStateVariableTraceTest.hpp.

**8.336.1.179 TEST\_F( NumericStateVariableTraceTest , OscillationsValueBinaryStatisticalNumeric )**

Definition at line 1525 of file NumericStateVariableTraceTest.hpp.

**8.336.1.180 TEST\_F( NumericStateVariableTraceTest , OscillationsValueBinaryStatisticalQuantileNumeric )**

Definition at line 1529 of file NumericStateVariableTraceTest.hpp.

**8.336.1.181 TEST\_F( NumericStateVariableTraceTest , EnclosingWithParenthesesDifferently1 )**

Definition at line 1542 of file NumericStateVariableTraceTest.hpp.

**8.336.1.182 TEST\_F( NumericStateVariableTraceTest , EnclosingWithParenthesesDifferently2 )**

Definition at line 1546 of file NumericStateVariableTraceTest.hpp.

**8.336.1.183 TEST\_F( NumericStateVariableTraceTest , TimeIntervalExceedsTraceEndTime )**

Definition at line 1559 of file NumericStateVariableTraceTest.hpp.

**8.336.1.184 TEST\_F( NumericStateVariableTraceTest , TimeIntervalExceedsTraceStartTime )**

Definition at line 1563 of file NumericStateVariableTraceTest.hpp.

**8.336.1.185 TEST\_F( NumericStateVariableTraceTest , ConstraintsCombinationUnary )**

Definition at line 1576 of file NumericStateVariableTraceTest.hpp.

**8.336.1.186 TEST\_F( NumericStateVariableTraceTest , ConstraintsCombinationBinary )**

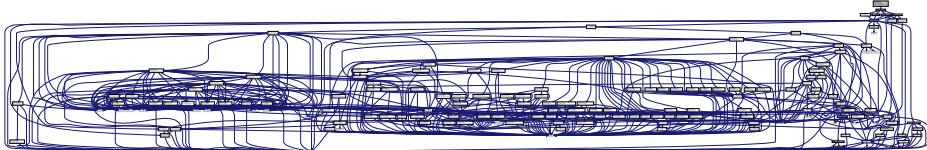
Definition at line 1580 of file NumericStateVariableTraceTest.hpp.

Definition at line 1584 of file NumericStateVariableTraceTest.hpp.

## 8.337 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.cpp File Reference

```
#include "ParserEvaluationTest.hpp"
```

Include dependency graph for ParserEvaluationTest.cpp:



### Functions

- int [main](#) (int argc, char \*\*argv)

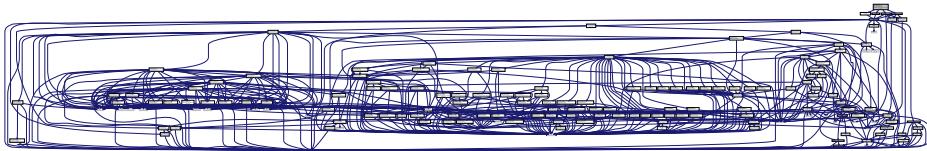
#### 8.337.1 Function Documentation

##### 8.337.1.1 int main ( int argc, char \*\* argv )

Definition at line 4 of file ParserEvaluationTest.cpp.

## 8.338 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.hpp File Reference

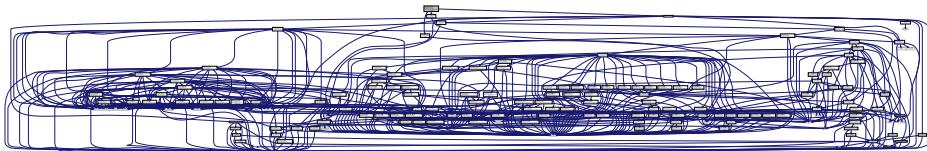
```
#include "CompleteTraceTest.hpp"
#include "EmptyTraceTest.hpp"
#include "NumericStateVariableTraceTest.hpp"
#include "SpatialEntitiesTraceTest.hpp"
Include dependency graph for ParserEvaluationTest.hpp:
```



## 8.339 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File Reference

```
#include "TraceEvaluationTest.hpp"
#include "multiscale/verification/spatial-temporal/model/Cluster.hpp"
#include "multiscale/verification/spatial-temporal/model/Region.hpp"
```

Include dependency graph for SpatialEntitiesTraceTest.hpp:



## Classes

- class [multiscaletest::SpatialEntitiesTraceTest](#)  
*Class for testing evaluation of spatial entities-only traces.*

## Namespaces

- [multiscaletest](#)

## Functions

- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericFilter\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureAdd\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureDiv\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureLog\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureMod\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureMultiply\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasurePower\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericMeasureSubtract\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryNumericTemporal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalQuantileMeasurePercentile\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalQuantileMeasureQuartile\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalQuantileSpatial\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, BinaryStatisticalSpatial\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ChangeMeasureDifference\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ChangeMeasureRatio\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ChangeTemporalNumericCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ChangeTemporalNumericMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ComparatorGreaterThan\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ComparatorLessThan\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ComparatorGreaterThanOrEqualTo\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ComparatorLessThanOrEqualTo\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ComparatorEqual\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, CompoundConstraint\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, CompoundConstraintMultiple\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, CompoundLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, CompoundLogicPropertyMultiple\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintEnclosedByParentheses\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintEnclosedByParenthesesQuadrupled\)](#)

Reference

- [TEST\\_F \(SpatialEntitiesTraceTest, Constraint\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FilterNumericMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FilterSubset\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HeterogeneousTimeseriesComponentPeak\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HeterogeneousTimeseriesComponentValley\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousHomogeneousTimeseries\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesComponentAscent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesComponentDescent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesComponentPlateau\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesComponentUniformAscent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesComponentUniformDescent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesMeasureTimeSpan\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, HomogeneousTimeseriesMeasureValue\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, LogicPropertyEnclosedByParentheses\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, LogicPropertyEnclosedByParenthesesDoubled\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, LogicPropertyEnclosedByParenthesesQuadrupled\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, LogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, MultipleLogicProperties1\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, MultipleLogicProperties2\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NextKLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NextLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NotConstraint\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NotLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericMeasureCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericSpatialMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWithoutTypes\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableTypeLeft\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableTypeRight\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableBothTypes\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableBothTypesAndDifferentTypeValues\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableOneNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWrongRhsType\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWrongName\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWrongLongName\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWrongTypeLhs\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStateVariableWrongTypeLhsLargerValue\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, NumericStatisticalMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SemanticType\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SimilarityMeasureAntiSimilar\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SimilarityMeasureSimilar\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SimilarityTemporalNumericCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureClusteredness\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureDensity\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureArea\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasurePerimeter\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureDistanceFromOrigin\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureAngle\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureTriangleMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureRectangleMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureCircleMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureCentroidX\)](#)

- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureCentroidY\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SpatialMeasureCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, Subset\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationDifference\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationDifferenceRegion\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationIntersection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationIntersectionRegion\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationUnion\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetOperationUnionRegion\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetSpecificClusters\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetSpecificRegions\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, SubsetSubsetOperation\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TemporalNumericCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TemporalNumericComparison\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TemporalNumericMeasure\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TemporalNumericMeasureCollection\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeseriesComponent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeseriesMeasureEnteringTime\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeseriesMeasureEnteringValue\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeseriesTimeseriesComponent\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericFilter\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureAbs\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureCeil\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureFloor\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureRound\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureSign\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureSqrt\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericMeasureTrunc\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryNumericTemporal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnarySpatialConstraint\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureAvg\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureCount\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureGeomean\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureHarmean\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureKurt\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureMax\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureMedian\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureMin\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureMode\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureProduct\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureSkew\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureStdev\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureSum\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalMeasureVar\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryStatisticalSpatial\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UnaryTypeConstraint\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UntilLogicProperty\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, UntilLogicPropertyMultiple\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueReal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueUnaryStatisticalNumeric\)](#)

- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalConstantValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueReal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, FutureIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueNumericStateVariable2\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationsValueUnaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationsValueBinaryStatisticalNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, OscillationsValueBinaryStatisticalQuantileNumeric\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, EnclosingWithParenthesesDifferently1\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, EnclosingWithParenthesesDifferently2\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeIntervalExceedsTraceEndTime\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, TimeIntervalExceedsTraceStartTime\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintsCombinationUnary\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintsCombinationBinary\)](#)
- [TEST\\_F \(SpatialEntitiesTraceTest, ConstraintsCombinationNary\)](#)

## 8.339.1 Function Documentation

### 8.339.1.1 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericFilter )

Definition at line 138 of file SpatialEntitiesTraceTest.hpp.

### 8.339.1.2 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureAdd )

Definition at line 151 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.3 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureDiv )**

Definition at line 155 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.4 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureLog )**

Definition at line 159 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.5 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureMod )**

Definition at line 163 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.6 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureMultiply )**

Definition at line 167 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.7 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasurePower )**

Definition at line 171 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.8 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericMeasureSubtract )**

Definition at line 175 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.9 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericNumeric )**

Definition at line 188 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.10 TEST\_F( SpatialEntitiesTraceTest , BinaryNumericTemporal )**

Definition at line 201 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.11 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalMeasure )**

Definition at line 214 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.12 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalNumeric )**

Definition at line 227 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.13 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalQuantileMeasurePercentile )**

Definition at line 240 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.14 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalQuantileMeasureQuartile )**

Definition at line 244 of file SpatialEntitiesTraceTest.hpp.

8.339.1.15 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalQuantileNumeric )

Definition at line 257 of file SpatialEntitiesTraceTest.hpp.

8.339.1.16 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalQuantileSpatial )

Definition at line 270 of file SpatialEntitiesTraceTest.hpp.

8.339.1.17 TEST\_F( SpatialEntitiesTraceTest , BinaryStatisticalSpatial )

Definition at line 283 of file SpatialEntitiesTraceTest.hpp.

8.339.1.18 TEST\_F( SpatialEntitiesTraceTest , ChangeMeasureDifference )

Definition at line 296 of file SpatialEntitiesTraceTest.hpp.

8.339.1.19 TEST\_F( SpatialEntitiesTraceTest , ChangeMeasureRatio )

Definition at line 300 of file SpatialEntitiesTraceTest.hpp.

8.339.1.20 TEST\_F( SpatialEntitiesTraceTest , ChangeTemporalNumericCollection )

Definition at line 313 of file SpatialEntitiesTraceTest.hpp.

8.339.1.21 TEST\_F( SpatialEntitiesTraceTest , ChangeTemporalNumericMeasure )

Definition at line 326 of file SpatialEntitiesTraceTest.hpp.

8.339.1.22 TEST\_F( SpatialEntitiesTraceTest , ComparatorGreaterThan )

Definition at line 339 of file SpatialEntitiesTraceTest.hpp.

8.339.1.23 TEST\_F( SpatialEntitiesTraceTest , ComparatorLessThan )

Definition at line 343 of file SpatialEntitiesTraceTest.hpp.

8.339.1.24 TEST\_F( SpatialEntitiesTraceTest , ComparatorGreaterThanOrEqual )

Definition at line 347 of file SpatialEntitiesTraceTest.hpp.

8.339.1.25 TEST\_F( SpatialEntitiesTraceTest , ComparatorLessThanOrEqual )

Definition at line 351 of file SpatialEntitiesTraceTest.hpp.

8.339.1.26 TEST\_F( SpatialEntitiesTraceTest , ComparatorEqual )

Definition at line 355 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.27 TEST\_F( SpatialEntitiesTraceTest , CompoundConstraint )**

Definition at line 368 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.28 TEST\_F( SpatialEntitiesTraceTest , CompoundConstraintMultiple )**

Definition at line 375 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.29 TEST\_F( SpatialEntitiesTraceTest , CompoundLogicProperty )**

Definition at line 391 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.30 TEST\_F( SpatialEntitiesTraceTest , CompoundLogicPropertyMultiple )**

Definition at line 398 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.31 TEST\_F( SpatialEntitiesTraceTest , ConstraintEnclosedByParentheses )**

Definition at line 414 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.32 TEST\_F( SpatialEntitiesTraceTest , ConstraintEnclosedByParenthesesDoubled )**

Definition at line 418 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.33 TEST\_F( SpatialEntitiesTraceTest , ConstraintEnclosedByParenthesesQuadrupled )**

Definition at line 422 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.34 TEST\_F( SpatialEntitiesTraceTest , Constraint )**

Definition at line 435 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.35 TEST\_F( SpatialEntitiesTraceTest , FilterNumericMeasure )**

Definition at line 448 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.36 TEST\_F( SpatialEntitiesTraceTest , FilterSubset )**

Definition at line 461 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.37 TEST\_F( SpatialEntitiesTraceTest , FutureLogicProperty )**

Definition at line 474 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.38 TEST\_F( SpatialEntitiesTraceTest , GlobalLogicProperty )**

Definition at line 487 of file SpatialEntitiesTraceTest.hpp.

---

8.339.1.39 TEST\_F( SpatialEntitiesTraceTest , HeterogeneousTimeseriesComponentPeak )

Definition at line 500 of file SpatialEntitiesTraceTest.hpp.

8.339.1.40 TEST\_F( SpatialEntitiesTraceTest , HeterogeneousTimeseriesComponentValley )

Definition at line 504 of file SpatialEntitiesTraceTest.hpp.

8.339.1.41 TEST\_F( SpatialEntitiesTraceTest , HomogeneousHomogeneousTimeseries )

Definition at line 517 of file SpatialEntitiesTraceTest.hpp.

8.339.1.42 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesComponentAscent )

Definition at line 530 of file SpatialEntitiesTraceTest.hpp.

8.339.1.43 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesComponentDescent )

Definition at line 534 of file SpatialEntitiesTraceTest.hpp.

8.339.1.44 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesComponentPlateau )

Definition at line 538 of file SpatialEntitiesTraceTest.hpp.

8.339.1.45 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesComponentUniformAscent )

Definition at line 542 of file SpatialEntitiesTraceTest.hpp.

8.339.1.46 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesComponentUniformDescent )

Definition at line 546 of file SpatialEntitiesTraceTest.hpp.

8.339.1.47 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesMeasureTimeSpan )

Definition at line 559 of file SpatialEntitiesTraceTest.hpp.

8.339.1.48 TEST\_F( SpatialEntitiesTraceTest , HomogeneousTimeseriesMeasureValue )

Definition at line 563 of file SpatialEntitiesTraceTest.hpp.

8.339.1.49 TEST\_F( SpatialEntitiesTraceTest , LogicPropertyEnclosedByParentheses )

Definition at line 576 of file SpatialEntitiesTraceTest.hpp.

8.339.1.50 TEST\_F( SpatialEntitiesTraceTest , LogicPropertyEnclosedByParenthesesDoubled )

Definition at line 580 of file SpatialEntitiesTraceTest.hpp.

8.339.1.51 TEST\_F( **SpatialEntitiesTraceTest** , LogicPropertyEnclosedByParenthesesQuadrupled )

Definition at line 584 of file SpatialEntitiesTraceTest.hpp.

8.339.1.52 TEST\_F( **SpatialEntitiesTraceTest** , LogicProperty )

Definition at line 597 of file SpatialEntitiesTraceTest.hpp.

8.339.1.53 TEST\_F( **SpatialEntitiesTraceTest** , MultipleLogicProperties1 )

Definition at line 610 of file SpatialEntitiesTraceTest.hpp.

8.339.1.54 TEST\_F( **SpatialEntitiesTraceTest** , MultipleLogicProperties2 )

Definition at line 614 of file SpatialEntitiesTraceTest.hpp.

8.339.1.55 TEST\_F( **SpatialEntitiesTraceTest** , NextKLogicProperty )

Definition at line 682 of file SpatialEntitiesTraceTest.hpp.

8.339.1.56 TEST\_F( **SpatialEntitiesTraceTest** , NextLogicProperty )

Definition at line 695 of file SpatialEntitiesTraceTest.hpp.

8.339.1.57 TEST\_F( **SpatialEntitiesTraceTest** , NotConstraint )

Definition at line 708 of file SpatialEntitiesTraceTest.hpp.

8.339.1.58 TEST\_F( **SpatialEntitiesTraceTest** , NotLogicProperty )

Definition at line 721 of file SpatialEntitiesTraceTest.hpp.

8.339.1.59 TEST\_F( **SpatialEntitiesTraceTest** , NumericMeasure )

Definition at line 734 of file SpatialEntitiesTraceTest.hpp.

8.339.1.60 TEST\_F( **SpatialEntitiesTraceTest** , NumericMeasureCollection )

Definition at line 747 of file SpatialEntitiesTraceTest.hpp.

8.339.1.61 TEST\_F( **SpatialEntitiesTraceTest** , NumericSpatialMeasure )

Definition at line 760 of file SpatialEntitiesTraceTest.hpp.

8.339.1.62 TEST\_F( **SpatialEntitiesTraceTest** , NumericStateVariableWithoutTypes )

Definition at line 773 of file SpatialEntitiesTraceTest.hpp.

---

8.339.1.63 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableTypeLeft )

Definition at line 777 of file SpatialEntitiesTraceTest.hpp.

8.339.1.64 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableTypeRight )

Definition at line 781 of file SpatialEntitiesTraceTest.hpp.

8.339.1.65 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableBothTypes )

Definition at line 785 of file SpatialEntitiesTraceTest.hpp.

8.339.1.66 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableBothTypesAndDifferentTypeValues )

Definition at line 789 of file SpatialEntitiesTraceTest.hpp.

8.339.1.67 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableOneNumericStateVariable )

Definition at line 793 of file SpatialEntitiesTraceTest.hpp.

8.339.1.68 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableWrongRhsType )

Definition at line 797 of file SpatialEntitiesTraceTest.hpp.

8.339.1.69 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableWrongName )

Definition at line 801 of file SpatialEntitiesTraceTest.hpp.

8.339.1.70 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableWrongLongName )

Definition at line 805 of file SpatialEntitiesTraceTest.hpp.

8.339.1.71 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableWrongTypeLhs )

Definition at line 809 of file SpatialEntitiesTraceTest.hpp.

8.339.1.72 TEST\_F( SpatialEntitiesTraceTest , NumericStateVariableWrongTypeLhsLargerValue )

Definition at line 813 of file SpatialEntitiesTraceTest.hpp.

8.339.1.73 TEST\_F( SpatialEntitiesTraceTest , NumericStatisticalMeasure )

Definition at line 826 of file SpatialEntitiesTraceTest.hpp.

8.339.1.74 TEST\_F( SpatialEntitiesTraceTest , ProbabilisticLogicProperty )

Definition at line 839 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.75 TEST\_F( SpatialEntitiesTraceTest , SemanticType )**

Definition at line 852 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.76 TEST\_F( SpatialEntitiesTraceTest , SimilarityMeasureAntiSimilar )**

Definition at line 865 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.77 TEST\_F( SpatialEntitiesTraceTest , SimilarityMeasureSimilar )**

Definition at line 869 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.78 TEST\_F( SpatialEntitiesTraceTest , SimilarityTemporalNumericCollection )**

Definition at line 882 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.79 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureClusteredness )**

Definition at line 895 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.80 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureDensity )**

Definition at line 899 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.81 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureArea )**

Definition at line 903 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.82 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasurePerimeter )**

Definition at line 907 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.83 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureDistanceFromOrigin )**

Definition at line 911 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.84 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureAngle )**

Definition at line 915 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.85 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureTriangleMeasure )**

Definition at line 919 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.86 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureRectangleMeasure )**

Definition at line 923 of file SpatialEntitiesTraceTest.hpp.

---

8.339.1.87 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureCircleMeasure )

Definition at line 927 of file SpatialEntitiesTraceTest.hpp.

8.339.1.88 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureCentroidX )

Definition at line 931 of file SpatialEntitiesTraceTest.hpp.

8.339.1.89 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureCentroidY )

Definition at line 935 of file SpatialEntitiesTraceTest.hpp.

8.339.1.90 TEST\_F( SpatialEntitiesTraceTest , SpatialMeasureCollection )

Definition at line 948 of file SpatialEntitiesTraceTest.hpp.

8.339.1.91 TEST\_F( SpatialEntitiesTraceTest , Subset )

Definition at line 961 of file SpatialEntitiesTraceTest.hpp.

8.339.1.92 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationDifference )

Definition at line 974 of file SpatialEntitiesTraceTest.hpp.

8.339.1.93 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationDifferenceRegion )

Definition at line 978 of file SpatialEntitiesTraceTest.hpp.

8.339.1.94 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationIntersection )

Definition at line 982 of file SpatialEntitiesTraceTest.hpp.

8.339.1.95 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationIntersectionRegion )

Definition at line 986 of file SpatialEntitiesTraceTest.hpp.

8.339.1.96 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationUnion )

Definition at line 990 of file SpatialEntitiesTraceTest.hpp.

8.339.1.97 TEST\_F( SpatialEntitiesTraceTest , SubsetOperationUnionRegion )

Definition at line 994 of file SpatialEntitiesTraceTest.hpp.

8.339.1.98 TEST\_F( SpatialEntitiesTraceTest , SubsetSpecificClusters )

Definition at line 1007 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.99 TEST\_F( SpatialEntitiesTraceTest , SubsetSpecificRegions )**

Definition at line 1011 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.100 TEST\_F( SpatialEntitiesTraceTest , SubsetSubsetOperation )**

Definition at line 1025 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.101 TEST\_F( SpatialEntitiesTraceTest , TemporalNumericCollection )**

Definition at line 1038 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.102 TEST\_F( SpatialEntitiesTraceTest , TemporalNumericComparison )**

Definition at line 1051 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.103 TEST\_F( SpatialEntitiesTraceTest , TemporalNumericMeasure )**

Definition at line 1064 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.104 TEST\_F( SpatialEntitiesTraceTest , TemporalNumericMeasureCollection )**

Definition at line 1077 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.105 TEST\_F( SpatialEntitiesTraceTest , TimeseriesComponent )**

Definition at line 1090 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.106 TEST\_F( SpatialEntitiesTraceTest , TimeseriesMeasureEnteringTime )**

Definition at line 1103 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.107 TEST\_F( SpatialEntitiesTraceTest , TimeseriesMeasureEnteringValue )**

Definition at line 1107 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.108 TEST\_F( SpatialEntitiesTraceTest , TimeseriesTimeseriesComponent )**

Definition at line 1120 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.109 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericFilter )**

Definition at line 1133 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.110 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureAbs )**

Definition at line 1146 of file SpatialEntitiesTraceTest.hpp.

---

8.339.1.111 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureCeil )

Definition at line 1150 of file SpatialEntitiesTraceTest.hpp.

8.339.1.112 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureFloor )

Definition at line 1154 of file SpatialEntitiesTraceTest.hpp.

8.339.1.113 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureRound )

Definition at line 1158 of file SpatialEntitiesTraceTest.hpp.

8.339.1.114 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureSign )

Definition at line 1162 of file SpatialEntitiesTraceTest.hpp.

8.339.1.115 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureSqrt )

Definition at line 1166 of file SpatialEntitiesTraceTest.hpp.

8.339.1.116 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericMeasureTrunc )

Definition at line 1170 of file SpatialEntitiesTraceTest.hpp.

8.339.1.117 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericNumeric )

Definition at line 1183 of file SpatialEntitiesTraceTest.hpp.

8.339.1.118 TEST\_F( SpatialEntitiesTraceTest , UnaryNumericTemporal )

Definition at line 1196 of file SpatialEntitiesTraceTest.hpp.

8.339.1.119 TEST\_F( SpatialEntitiesTraceTest , UnarySpatialConstraint )

Definition at line 1209 of file SpatialEntitiesTraceTest.hpp.

8.339.1.120 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureAvg )

Definition at line 1222 of file SpatialEntitiesTraceTest.hpp.

8.339.1.121 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureCount )

Definition at line 1226 of file SpatialEntitiesTraceTest.hpp.

8.339.1.122 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureGeomean )

Definition at line 1230 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.123 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureHarmean )**

Definition at line 1234 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.124 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureKurt )**

Definition at line 1238 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.125 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureMax )**

Definition at line 1242 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.126 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureMedian )**

Definition at line 1246 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.127 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureMin )**

Definition at line 1250 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.128 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureMode )**

Definition at line 1254 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.129 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureProduct )**

Definition at line 1258 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.130 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureSkew )**

Definition at line 1262 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.131 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureStdev )**

Definition at line 1266 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.132 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureSum )**

Definition at line 1270 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.133 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalMeasureVar )**

Definition at line 1274 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.134 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalNumeric )**

Definition at line 1287 of file SpatialEntitiesTraceTest.hpp.

---

8.339.1.135 TEST\_F( SpatialEntitiesTraceTest , UnaryStatisticalSpatial )

Definition at line 1300 of file SpatialEntitiesTraceTest.hpp.

8.339.1.136 TEST\_F( SpatialEntitiesTraceTest , UnaryTypeConstraint )

Definition at line 1313 of file SpatialEntitiesTraceTest.hpp.

8.339.1.137 TEST\_F( SpatialEntitiesTraceTest , UntilLogicProperty )

Definition at line 1326 of file SpatialEntitiesTraceTest.hpp.

8.339.1.138 TEST\_F( SpatialEntitiesTraceTest , UntilLogicPropertyMultiple )

Definition at line 1330 of file SpatialEntitiesTraceTest.hpp.

8.339.1.139 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueReal )

Definition at line 1343 of file SpatialEntitiesTraceTest.hpp.

8.339.1.140 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueNumericStateVariable )

Definition at line 1347 of file SpatialEntitiesTraceTest.hpp.

8.339.1.141 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueUnaryNumeric )

Definition at line 1351 of file SpatialEntitiesTraceTest.hpp.

8.339.1.142 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueBinaryNumeric )

Definition at line 1355 of file SpatialEntitiesTraceTest.hpp.

8.339.1.143 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueUnaryStatisticalNumeric )

Definition at line 1359 of file SpatialEntitiesTraceTest.hpp.

8.339.1.144 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueBinaryStatisticalNumeric )

Definition at line 1363 of file SpatialEntitiesTraceTest.hpp.

8.339.1.145 TEST\_F( SpatialEntitiesTraceTest , GlobalConstantValueBinaryStatisticalQuantileNumeric )

Definition at line 1367 of file SpatialEntitiesTraceTest.hpp.

8.339.1.146 TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueReal )

Definition at line 1380 of file SpatialEntitiesTraceTest.hpp.

8.339.1.147 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueNumericStateVariable )**

Definition at line 1384 of file SpatialEntitiesTraceTest.hpp.

8.339.1.148 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueUnaryNumeric )**

Definition at line 1388 of file SpatialEntitiesTraceTest.hpp.

8.339.1.149 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueBinaryNumeric )**

Definition at line 1392 of file SpatialEntitiesTraceTest.hpp.

8.339.1.150 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueUnaryStatisticalNumeric )**

Definition at line 1396 of file SpatialEntitiesTraceTest.hpp.

8.339.1.151 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueBinaryStatisticalNumeric )**

Definition at line 1400 of file SpatialEntitiesTraceTest.hpp.

8.339.1.152 **TEST\_F( SpatialEntitiesTraceTest , FutureIncreasingValueBinaryStatisticalQuantileNumeric )**

Definition at line 1404 of file SpatialEntitiesTraceTest.hpp.

8.339.1.153 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueReal )**

Definition at line 1417 of file SpatialEntitiesTraceTest.hpp.

8.339.1.154 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueNumericStateVariable )**

Definition at line 1421 of file SpatialEntitiesTraceTest.hpp.

8.339.1.155 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueUnaryNumeric )**

Definition at line 1425 of file SpatialEntitiesTraceTest.hpp.

8.339.1.156 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueBinaryNumeric )**

Definition at line 1429 of file SpatialEntitiesTraceTest.hpp.

8.339.1.157 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueUnaryStatisticalNumeric )**

Definition at line 1433 of file SpatialEntitiesTraceTest.hpp.

8.339.1.158 **TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueBinaryStatisticalNumeric )**

Definition at line 1437 of file SpatialEntitiesTraceTest.hpp.

8.339.1.159 TEST\_F( SpatialEntitiesTraceTest , GlobalDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1441 of file SpatialEntitiesTraceTest.hpp.

8.339.1.160 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueReal )

Definition at line 1454 of file SpatialEntitiesTraceTest.hpp.

8.339.1.161 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueNumericStateVariable )

Definition at line 1458 of file SpatialEntitiesTraceTest.hpp.

8.339.1.162 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueNumericStateVariable2 )

Definition at line 1462 of file SpatialEntitiesTraceTest.hpp.

8.339.1.163 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueUnaryNumeric )

Definition at line 1466 of file SpatialEntitiesTraceTest.hpp.

8.339.1.164 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueBinaryNumeric )

Definition at line 1470 of file SpatialEntitiesTraceTest.hpp.

8.339.1.165 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueUnaryStatisticalNumeric )

Definition at line 1474 of file SpatialEntitiesTraceTest.hpp.

8.339.1.166 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalNumeric )

Definition at line 1478 of file SpatialEntitiesTraceTest.hpp.

8.339.1.167 TEST\_F( SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1482 of file SpatialEntitiesTraceTest.hpp.

8.339.1.168 TEST\_F( SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueReal )

Definition at line 1495 of file SpatialEntitiesTraceTest.hpp.

8.339.1.169 TEST\_F( SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueNumericStateVariable )

Definition at line 1499 of file SpatialEntitiesTraceTest.hpp.

8.339.1.170 TEST\_F( SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueUnaryNumeric )

Definition at line 1503 of file SpatialEntitiesTraceTest.hpp.

8.339.1.171 TEST\_F( **SpatialEntitiesTraceTest** , DecreasingUntilIncreasingValueBinaryNumeric )

Definition at line 1507 of file SpatialEntitiesTraceTest.hpp.

8.339.1.172 TEST\_F( **SpatialEntitiesTraceTest** , DecreasingUntilIncreasingValueUnaryStatisticalNumeric )

Definition at line 1511 of file SpatialEntitiesTraceTest.hpp.

8.339.1.173 TEST\_F( **SpatialEntitiesTraceTest** , DecreasingUntilIncreasingValueBinaryStatisticalNumeric )

Definition at line 1515 of file SpatialEntitiesTraceTest.hpp.

8.339.1.174 TEST\_F( **SpatialEntitiesTraceTest** , DecreasingUntilIncreasingValueBinaryStatisticalQuantileNumeric )

Definition at line 1519 of file SpatialEntitiesTraceTest.hpp.

8.339.1.175 TEST\_F( **SpatialEntitiesTraceTest** , OscillationValueNumericStateVariable )

Definition at line 1532 of file SpatialEntitiesTraceTest.hpp.

8.339.1.176 TEST\_F( **SpatialEntitiesTraceTest** , OscillationsValueUnaryNumeric )

Definition at line 1536 of file SpatialEntitiesTraceTest.hpp.

8.339.1.177 TEST\_F( **SpatialEntitiesTraceTest** , OscillationsValueBinaryNumeric )

Definition at line 1540 of file SpatialEntitiesTraceTest.hpp.

8.339.1.178 TEST\_F( **SpatialEntitiesTraceTest** , OscillationsValueUnaryStatisticalNumeric )

Definition at line 1544 of file SpatialEntitiesTraceTest.hpp.

8.339.1.179 TEST\_F( **SpatialEntitiesTraceTest** , OscillationsValueBinaryStatisticalNumeric )

Definition at line 1548 of file SpatialEntitiesTraceTest.hpp.

8.339.1.180 TEST\_F( **SpatialEntitiesTraceTest** , OscillationsValueBinaryStatisticalQuantileNumeric )

Definition at line 1552 of file SpatialEntitiesTraceTest.hpp.

8.339.1.181 TEST\_F( **SpatialEntitiesTraceTest** , EnclosingWithParenthesesDifferently1 )

Definition at line 1565 of file SpatialEntitiesTraceTest.hpp.

8.339.1.182 TEST\_F( **SpatialEntitiesTraceTest** , EnclosingWithParenthesesDifferently2 )

Definition at line 1569 of file SpatialEntitiesTraceTest.hpp.

### **8.340 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.cpp File**

**Reference**

**1425**

**8.339.1.183 TEST\_F( SpatialEntitiesTraceTest , TimeIntervalExceedsTraceEndTime )**

Definition at line 1582 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.184 TEST\_F( SpatialEntitiesTraceTest , TimeIntervalExceedsTraceStartTime )**

Definition at line 1586 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.185 TEST\_F( SpatialEntitiesTraceTest , ConstraintsCombinationUnary )**

Definition at line 1599 of file SpatialEntitiesTraceTest.hpp.

**8.339.1.186 TEST\_F( SpatialEntitiesTraceTest , ConstraintsCombinationBinary )**

Definition at line 1603 of file SpatialEntitiesTraceTest.hpp.

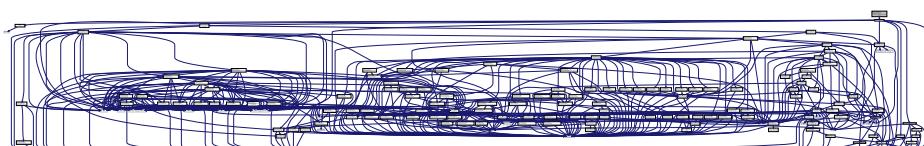
**8.339.1.187 TEST\_F( SpatialEntitiesTraceTest , ConstraintsCombinationNary )**

Definition at line 1607 of file SpatialEntitiesTraceTest.hpp.

### **8.340 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.cpp File Reference**

```
#include "TraceEvaluationTest.hpp"
```

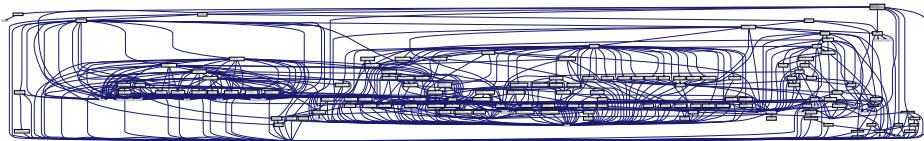
Include dependency graph for TraceEvaluationTest.cpp:



### **8.341 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.hpp File Reference**

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/exception/TestException.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include "multiscale/verification/spatial-temporal/model/NumericStateVariable<Id.hpp>
#include "multiscale/verification/spatial-temporal/model/SemanticType.hpp"
#include "multiscale/verification/spatial-temporal/model/TypeSemantics<Table.hpp>
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <limits>
#include <string>
```

Include dependency graph for TraceEvaluationTest.hpp:



## Classes

- class [multiscaletest::TraceEvaluationTest](#)

*Class for testing evaluation of traces.*

## Namespaces

- [multiscaletest](#)

## Variables

- static const std::string [ERR\\_MSG\\_TEST](#) = "The given input string could not be successfully parsed."

### 8.341.1 Variable Documentation

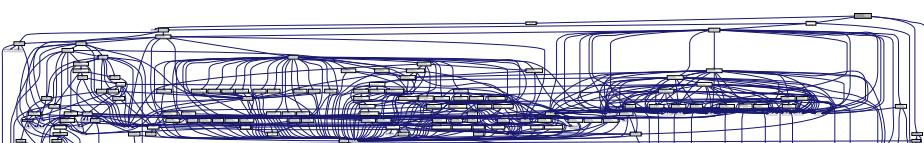
8.341.1.1 `const std::string ERR_MSG_TEST = "The given input string could not be successfully parsed." [static]`

Definition at line 17 of file TraceEvaluationTest.hpp.

Referenced by [multiscaletest::TraceEvaluationTest::RunTest\(\)](#).

## 8.342 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/InputStringParser.hpp File Reference

```
#include "multiscale/core/MultiscaleTest.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/verification/spatial-temporal/parsing/Parser.hpp"
#include <string>
Include dependency graph for InputStringParser.hpp:
```



## Namespaces

- [multiscaletest](#)
- [multiscaletest::verification](#)

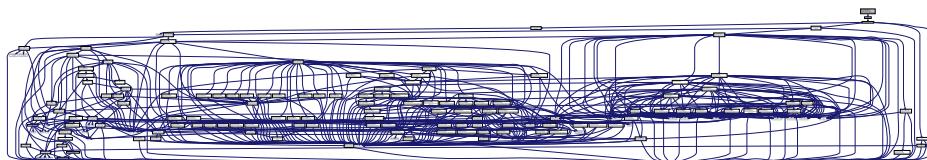
- bool [multiscaletest::verification::parseInputString](#) (const std::string &inputString)

*Parse the input string and return the result of the parsing.*

## 8.343 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.cpp File Reference

```
#include "ParserTest.hpp"
```

Include dependency graph for ParserTest.cpp:



### Functions

- int [main](#) (int argc, char \*\*argv)

#### 8.343.1 Function Documentation

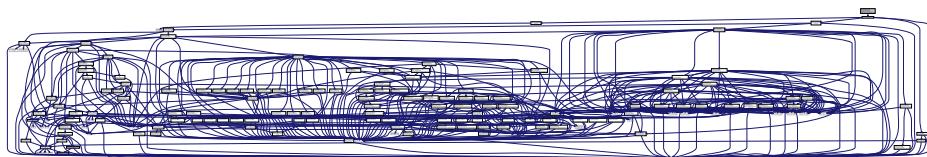
##### 8.343.1.1 int main ( int argc, char \*\* argv )

Definition at line 6 of file ParserTest.cpp.

## 8.344 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.hpp File Reference

```
#include "parsing/InputStreamParser.hpp"
```

Include dependency graph for ParserTest.hpp:



### Functions

- [TEST](#) (BinaryNumericFilter, IncorrectInputMissingParameterOne)
- [TEST](#) (BinaryNumericFilter, IncorrectInputMissingParameterTwo)
- [TEST](#) (BinaryNumericFilter, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (BinaryNumericFilter, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinaryNumericFilter, IncorrectInputAfterStartBracket)
- [TEST](#) (BinaryNumericFilter, InvalidFirstParameter)
- [TEST](#) (BinaryNumericFilter, MissingParametersComma)

- [TEST](#) (BinaryNumericFilter, InvalidSecondParameter)
- [TEST](#) (BinaryNumericFilter, IncorrectInputBeforeEndBracket)
- [TEST](#) (BinaryNumericFilter, IncorrectInputAfterEndBracket)
- [TEST](#) (BinaryNumericFilter, Correct)
- [TEST](#) (BinaryNumericMeasure, IncorrectBinaryNumericMeasure)
- [TEST](#) (BinaryNumericMeasure, CorrectAdd)
- [TEST](#) (BinaryNumericMeasure, CorrectDiv)
- [TEST](#) (BinaryNumericMeasure, CorrectLog)
- [TEST](#) (BinaryNumericMeasure, CorrectMod)
- [TEST](#) (BinaryNumericMeasure, CorrectMultiply)
- [TEST](#) (BinaryNumericMeasure, CorrectPower)
- [TEST](#) (BinaryNumericMeasure, CorrectSubtract)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputMissingParameterOne)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputMissingParameterTwo)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputAfterStartBracket)
- [TEST](#) (BinaryNumericNumeric, InvalidFirstParameter)
- [TEST](#) (BinaryNumericNumeric, MissingParametersComma)
- [TEST](#) (BinaryNumericNumeric, InvalidSecondParameter)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputBeforeEndBracket)
- [TEST](#) (BinaryNumericNumeric, IncorrectInputAfterEndBracket)
- [TEST](#) (BinaryNumericNumeric, Correct)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputMissingParameterOne)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputMissingParameterTwo)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputAfterStartBracket)
- [TEST](#) (BinaryNumericTemporal, InvalidFirstParameter)
- [TEST](#) (BinaryNumericTemporal, MissingParametersComma)
- [TEST](#) (BinaryNumericTemporal, InvalidSecondParameter)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputBeforeEndBracket)
- [TEST](#) (BinaryNumericTemporal, IncorrectInputAfterEndBracket)
- [TEST](#) (BinaryNumericTemporal, Correct)
- [TEST](#) (BinaryStatisticalMeasure, IncorrectQuaternarySubsetMeasure)
- [TEST](#) (BinaryStatisticalMeasure, CorrectCovar)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputMissingParameterOne)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputMissingParameterTwo)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputAfterStartBracket)
- [TEST](#) (BinaryStatisticalNumeric, MissingComma)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputBeforeEndBracket)
- [TEST](#) (BinaryStatisticalNumeric, IncorrectInputAfterEndBracket)
- [TEST](#) (BinaryStatisticalNumeric, Correct)
- [TEST](#) (BinaryStatisticalQuantileMeasure, IncorrectBinaryStatisticalQuantileMeasure)
- [TEST](#) (BinaryStatisticalQuantileMeasure, CorrectPercentile)
- [TEST](#) (BinaryStatisticalQuantileMeasure, CorrectQuartile)
- [TEST](#) (BinaryStatisticalQuantileNumeric, IncorrectInputMissingParameterOne)
- [TEST](#) (BinaryStatisticalQuantileNumeric, IncorrectInputMissingParameterTwo)
- [TEST](#) (BinaryStatisticalQuantileNumeric, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (BinaryStatisticalQuantileNumeric, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinaryStatisticalQuantileNumeric, IncorrectInputAfterStartBracket)
- [TEST](#) (BinaryStatisticalQuantileNumeric, MissingComma)
- [TEST](#) (BinaryStatisticalQuantileNumeric, InvalidSpatialMeasureCollection)

Reference

- `TEST (BinaryStatisticalQuantileNumeric, IncorrectInputBeforeEndBracket)`
- `TEST (BinaryStatisticalQuantileNumeric, IncorrectInputAfterEndBracket)`
- `TEST (BinaryStatisticalQuantileNumeric, Correct)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputMissingParameterOne)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputMissingParameterTwo)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputMissingParametersOneTwo)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputBeforeStartBracket)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputAfterStartBracket)`
- `TEST (BinaryStatisticalQuantileSpatial, MissingComma)`
- `TEST (BinaryStatisticalQuantileSpatial, InvalidSpatialMeasureCollection)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputBeforeEndBracket)`
- `TEST (BinaryStatisticalQuantileSpatial, IncorrectInputAfterEndBracket)`
- `TEST (BinaryStatisticalQuantileSpatial, Correct)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputMissingParameterOne)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputMissingParameterTwo)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputMissingParametersOneTwo)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputBeforeStartBracket)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputAfterStartBracket)`
- `TEST (BinaryStatisticalSpatial, MissingComma)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputBeforeEndBracket)`
- `TEST (BinaryStatisticalSpatial, IncorrectInputAfterEndBracket)`
- `TEST (BinaryStatisticalSpatial, Correct)`
- `TEST (ChangeMeasure, IncorrectChangeMeasure)`
- `TEST (ChangeMeasure, CorrectDifference)`
- `TEST (ChangeMeasure, CorrectRatio)`
- `TEST (ChangeTemporalNumericCollection, IncorrectChangeMeasureDf)`
- `TEST (ChangeTemporalNumericCollection, IncorrectChangeMeasureDiff)`
- `TEST (ChangeTemporalNumericCollection, IncorrectChangeMeasureRatio)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputBeforeChangeMeasure)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputBeforeStartParanthesis)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputAfterStartParanthesis)`
- `TEST (ChangeTemporalNumericCollection, MissingParameter)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputMissingParameterAndBrackets)`
- `TEST (ChangeTemporalNumericCollection, IncorrectOpeningBracket)`
- `TEST (ChangeTemporalNumericCollection, IncorrectClosingBracket)`
- `TEST (ChangeTemporalNumericCollection, IncorrectBrackets)`
- `TEST (ChangeTemporalNumericCollection, IncorrectBracketsInverted)`
- `TEST (ChangeTemporalNumericCollection, IncorrectBracketsDoubled)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputBeforeEndParanthesis)`
- `TEST (ChangeTemporalNumericCollection, IncorrectInputAfterEndParanthesis)`
- `TEST (ChangeTemporalNumericCollection, IncorrectTemporalNumericCollectionRealNumber)`
- `TEST (ChangeTemporalNumericCollection, IncorrectTemporalNumericCollectionNumericStateVariable)`
- `TEST (ChangeTemporalNumericCollection, IncorrectTemporalNumericCollectionNumericSpatialMeasure)`
- `TEST (ChangeTemporalNumericCollection, Correct)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectChangeMeasure)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectInputBeforeChangeMeasure)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectInputBeforeStartParanthesis)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectInputAfterStartParanthesis)`
- `TEST (ChangeTemporalNumericMeasure, MissingParameter)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectInputMissingParameterAndBrackets)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectOpeningBracket)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectClosingBracket)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectBrackets)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectBracketsInverted)`
- `TEST (ChangeTemporalNumericMeasure, IncorrectBracketsDoubled)`

- [TEST](#) (ChangeTemporalNumericMeasure, IncorrectInputBeforeEndParanthesis)
- [TEST](#) (ChangeTemporalNumericMeasure, IncorrectInputAfterEndParanthesis)
- [TEST](#) (ChangeTemporalNumericMeasure, MissingComparator)
- [TEST](#) (ChangeTemporalNumericMeasure, IncorrectEndOperand)
- [TEST](#) (ChangeTemporalNumericMeasure, Correct)
- [TEST](#) (Comparator, IncorrectEqual)
- [TEST](#) (Comparator, IncorrectDifferent1)
- [TEST](#) (Comparator, IncorrectDifferent2)
- [TEST](#) (Comparator, CorrectGreaterThan)
- [TEST](#) (Comparator, CorrectLessThan)
- [TEST](#) (Comparator, CorrectGreaterThanOrEqual)
- [TEST](#) (Comparator, CorrectLessThanOrEqual)
- [TEST](#) (Comparator, CorrectEqual)
- [TEST](#) (CompoundConstraint, MissingBinaryOperator)
- [TEST](#) (CompoundConstraint, MissingConstraints)
- [TEST](#) (CompoundConstraint, MissingFirstConstraint)
- [TEST](#) (CompoundConstraint, MissingSecondConstraint)
- [TEST](#) (CompoundConstraint, BinaryOperatorAsUnaryBefore)
- [TEST](#) (CompoundConstraint, BinaryOperatorAsUnaryAfter)
- [TEST](#) (CompoundConstraint, TemporalNumericComparisonBeforeBinaryOperator)
- [TEST](#) (CompoundConstraint, UnaryNumericMeasureAfterBinaryOperator)
- [TEST](#) (CompoundConstraint, AdditionalOperatorBeforeBinaryOperator)
- [TEST](#) (CompoundConstraint, AdditionalOperatorAfterBinaryOperator)
- [TEST](#) (CompoundConstraint, Correct)
- [TEST](#) (CompoundConstraint, MultipleCorrect)
- [TEST](#) (CompoundLogicProperty, MissingBinaryOperator)
- [TEST](#) (CompoundLogicProperty, MissingLogicProperties)
- [TEST](#) (CompoundLogicProperty, MissingFirstLogicProperty)
- [TEST](#) (CompoundLogicProperty, MissingSecondLogicProperty)
- [TEST](#) (CompoundLogicProperty, BinaryOperatorAsUnaryBefore)
- [TEST](#) (CompoundLogicProperty, BinaryOperatorAsUnaryAfter)
- [TEST](#) (CompoundLogicProperty, UnaryStatisticalMeasureBeforeBinaryOperator)
- [TEST](#) (CompoundLogicProperty, UnaryNumericMeasureAfterBinaryOperator)
- [TEST](#) (CompoundLogicProperty, AdditionalOperatorBeforeBinaryOperator)
- [TEST](#) (CompoundLogicProperty, AdditionalOperatorAfterBinaryOperator)
- [TEST](#) (CompoundLogicProperty, Correct)
- [TEST](#) (CompoundLogicProperty, MultipleCorrect)
- [TEST](#) (ConstraintEnclosedByParentheses, MissingParenthesisRight)
- [TEST](#) (ConstraintEnclosedByParentheses, MissingParenthesisLeft)
- [TEST](#) (ConstraintEnclosedByParentheses, ExtraParenthesisLeft)
- [TEST](#) (ConstraintEnclosedByParentheses, ExtraParenthesisRight)
- [TEST](#) (ConstraintEnclosedByParentheses, InvertedParentheses)
- [TEST](#) (ConstraintEnclosedByParentheses, ExtraParenthesesBothSides)
- [TEST](#) (ConstraintEnclosedByParentheses, ParenthesesInWrongOrder)
- [TEST](#) (ConstraintEnclosedByParentheses, Correct)
- [TEST](#) (ConstraintEnclosedByParentheses, CorrectDoubled)
- [TEST](#) (ConstraintEnclosedByParentheses, CorrectQuadrupled)
- [TEST](#) (Constraint, ExtraInputBeforeConstraint)
- [TEST](#) (Constraint, ExtraInputAfterConstraint)
- [TEST](#) (Constraint, Correct)
- [TEST](#) (FilterSubset, IncorrectAlternative)
- [TEST](#) (FilterSubset, CorrectSpatialMeasureRealValue)
- [TEST](#) (FilterSubset, CorrectSpatialMeasures)
- [TEST](#) (FilterSubset, CorrectMultiple)
- [TEST](#) (FilterSubset, CorrectMultipleComplex)

Reference

- [TEST \(FilterSubset, IncorrectInputMisspelledFilter\)](#)
- [TEST \(FilterSubset, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(FilterSubset, IncorrectInputAfterStartBracket\)](#)
- [TEST \(FilterSubset, IncorrectInputMissingComma\)](#)
- [TEST \(FilterSubset, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(FilterSubset, IncorrectInputAfterEndBracket\)](#)
- [TEST \(FilterSubset, Correct\)](#)
- [TEST \(FutureLogicProperty, WrongInputMissingStartTimepoint\)](#)
- [TEST \(FutureLogicProperty, WrongInputMissingEndTimepoint\)](#)
- [TEST \(FutureLogicProperty, WrongInputMissingTimepoints\)](#)
- [TEST \(FutureLogicProperty, WrongInputMissingTimepointsAndBrackets\)](#)
- [TEST \(FutureLogicProperty, WrongInputBeforeStartParanthesis\)](#)
- [TEST \(FutureLogicProperty, WrongInputAfterStartParanthesis\)](#)
- [TEST \(FutureLogicProperty, MissingTimepointComma\)](#)
- [TEST \(FutureLogicProperty, InvalidStartTimepoint\)](#)
- [TEST \(FutureLogicProperty, InvalidEndTimepoint\)](#)
- [TEST \(FutureLogicProperty, InvalidTimepoints\)](#)
- [TEST \(FutureLogicProperty, WrongInputBeforeEndParanthesis\)](#)
- [TEST \(FutureLogicProperty, WrongInputAfterEndParanthesis\)](#)
- [TEST \(FutureLogicProperty, Correct\)](#)
- [TEST \(GlobalLogicProperty, WrongInputMissingStartTimepoint\)](#)
- [TEST \(GlobalLogicProperty, WrongInputMissingEndTimepoint\)](#)
- [TEST \(GlobalLogicProperty, WrongInputMissingTimepoints\)](#)
- [TEST \(GlobalLogicProperty, WrongInputMissingTimepointsAndBrackets\)](#)
- [TEST \(GlobalLogicProperty, WrongInputBeforeStartParanthesis\)](#)
- [TEST \(GlobalLogicProperty, WrongInputAfterStartParanthesis\)](#)
- [TEST \(GlobalLogicProperty, MissingTimepointComma\)](#)
- [TEST \(GlobalLogicProperty, InvalidStartTimepoint\)](#)
- [TEST \(GlobalLogicProperty, InvalidEndTimepoint\)](#)
- [TEST \(GlobalLogicProperty, InvalidTimepoints\)](#)
- [TEST \(GlobalLogicProperty, WrongInputBeforeEndParanthesis\)](#)
- [TEST \(GlobalLogicProperty, WrongInputAfterEndParanthesis\)](#)
- [TEST \(GlobalLogicProperty, Correct\)](#)
- [TEST \(HeterogeneousTimeseriesComponent, WrongAlternativeSpike\)](#)
- [TEST \(HeterogeneousTimeseriesComponent, WrongAlternativeSink\)](#)
- [TEST \(HeterogeneousTimeseriesComponent, WrongAlternativeTrough\)](#)
- [TEST \(HeterogeneousTimeseriesComponent, CorrectPeak\)](#)
- [TEST \(HeterogeneousTimeseriesComponent, CorrectValley\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputBeforeHomogeneousTimeseriesMeasure\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputWrongHomogeneousTimeseriesMeasure\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputBeforeStartParanthesis\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputAfterStartParanthesis\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputInvalidFirstParameter\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputMissingFirstParameter\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputMissingCommaBetweenParameters\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputMissingSecondParameter\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputInvalidSecondParameter\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputBeforeEndParanthesis\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputAfterEndParanthesis\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputMissingParantheses\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputInvertedParantheses\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputDoubledParantheses\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, IncorrectInputWrongParanthesesType\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, CorrectNumericStateVariables\)](#)
- [TEST \(HomogeneousHomogeneousTimeseries, CorrectSpatialEntities\)](#)

- `TEST (HomogeneousTimeseriesComponent, IncorrectAlternativeHeterogeneousTimeseriesComponent)`
- `TEST (HomogeneousTimeseriesComponent, CorrectAscent)`
- `TEST (HomogeneousTimeseriesComponent, CorrectDescent)`
- `TEST (HomogeneousTimeseriesComponent, CorrectPlateau)`
- `TEST (HomogeneousTimeseriesComponent, CorrectUniformAscent)`
- `TEST (HomogeneousTimeseriesComponent, CorrectUniformDescent)`
- `TEST (HomogeneousTimeseriesMeasure, IncorrectAlternativeTimeseriesMeasure)`
- `TEST (HomogeneousTimeseriesMeasure, CorrectTimeSpan)`
- `TEST (HomogeneousTimeseriesMeasure, CorrectValue)`
- `TEST (LogicPropertyEnclosedByParentheses, MissingParenthesisRight)`
- `TEST (LogicPropertyEnclosedByParentheses, MissingParenthesisLeft)`
- `TEST (LogicPropertyEnclosedByParentheses, ExtraParenthesisLeft)`
- `TEST (LogicPropertyEnclosedByParentheses, ExtraParenthesisRight)`
- `TEST (LogicPropertyEnclosedByParentheses, InvertedParentheses)`
- `TEST (LogicPropertyEnclosedByParentheses, ExtraParenthesesBothSides)`
- `TEST (LogicPropertyEnclosedByParentheses, ParenthesesInWrongOrder)`
- `TEST (LogicPropertyEnclosedByParentheses, Correct)`
- `TEST (LogicPropertyEnclosedByParentheses, CorrectDoubled)`
- `TEST (LogicPropertyEnclosedByParentheses, CorrectQuadrupled)`
- `TEST (LogicProperty, ExtraInputBeforeLogicProperty)`
- `TEST (LogicProperty, ExtraInputInsideLogicProperty)`
- `TEST (LogicProperty, ExtraInputAfterLogicProperty)`
- `TEST (LogicProperty, Correct)`
- `TEST (MultipleLogicProperties, Correct1)`
- `TEST (MultipleLogicProperties, Correct2)`
- `TEST (NextKLogicProperty, IncorrectInputMissingTimepoint)`
- `TEST (NextKLogicProperty, IncorrectInputAfterNextSymbol)`
- `TEST (NextKLogicProperty, IncorrectValueForNextTimepoints)`
- `TEST (NextKLogicProperty, RealValueForNextTimepoints)`
- `TEST (NextKLogicProperty, IncorrectInputBeforeLogicProperty)`
- `TEST (NextKLogicProperty, Correct)`
- `TEST (NextLogicProperty, IncorrectNextSymbol)`
- `TEST (NextLogicProperty, IncorrectInputAfterNextSymbol)`
- `TEST (NextLogicProperty, Correct)`
- `TEST (NotConstraint, IncorrectOperator)`
- `TEST (NotConstraint, OperatorAfterConstraint)`
- `TEST (NotConstraint, OperatorAfterConstraintAndExtraConstraint)`
- `TEST (NotConstraint, OperatorBeforeConstraintAndExtraConstraint)`
- `TEST (NotConstraint, Correct)`
- `TEST (NotLogicProperty, OperatorAfterLogicProperty)`
- `TEST (NotLogicProperty, OperatorAfterLogicPropertyAndExtraLogicProperty)`
- `TEST (NotLogicProperty, OperatorBeforeLogicPropertyAndExtraLogicProperty)`
- `TEST (NotLogicProperty, Correct)`
- `TEST (NumericMeasure, WrongAlternative)`
- `TEST (NumericMeasure, Correct)`
- `TEST (NumericMeasureCollection, WrongAlternative)`
- `TEST (NumericMeasureCollection, Correct)`
- `TEST (NumericSpatialMeasure, IncorrectAlternative)`
- `TEST (NumericSpatialMeasure, Correct)`
- `TEST (NumericStateVariable, MissingLeftCurlyBrace)`
- `TEST (NumericStateVariable, MissingRightCurlyBrace)`
- `TEST (NumericStateVariable, ExtraLeftCurlyBrace)`
- `TEST (NumericStateVariable, ExtraRightCurlyBrace)`
- `TEST (NumericStateVariable, InvertedCurlyBraces)`
- `TEST (NumericStateVariable, DoubleCurlyBraces)`

Reference

- [TEST \(NumericStateVariable, TripleCurlyBraces\)](#)
- [TEST \(NumericStateVariable, IncorrectSquareBrackets\)](#)
- [TEST \(NumericStateVariable, IncorrectRoundBrackets\)](#)
- [TEST \(NumericStateVariable, MissingLeftParanthesisForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingRightParanthesisForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingBothParanthesesForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingTypeKeywordForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingEqualComparatorForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingTypeValueForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingTypeKeywordAndEqualComparatorForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingAllForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectLeftParanthesisForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectRightParanthesisForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectParanthesesForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeSpecifierForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeSpecifierForTypeConstraint2\)](#)
- [TEST \(NumericStateVariable, IncorrectComparatorForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeInvalidCharacterForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeMissingValueBeforeDotForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeMissingValueAfterDotForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, IncorrectTypeMissingValueBetweenDotsForTypeConstraint\)](#)
- [TEST \(NumericStateVariable, MissingNumericStateVariableName\)](#)
- [TEST \(NumericStateVariable, MissingNumericStateVariableNameAndBraces\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableSimple1\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableSimple2\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableSimple3\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableWithTypeConstraint4\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableWithTypeConstraint5\)](#)
- [TEST \(NumericStateVariable, CorrectNumericStateVariableWithTypeConstraint6\)](#)
- [TEST \(NumericStatisticalMeasure, IncorrectAlternative\)](#)
- [TEST \(NumericStatisticalMeasure, Correct\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectProbabilitySymbol\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectProbabilitySymbol2\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectComparator\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectEqualComparator\)](#)
- [TEST \(ProbabilisticLogicProperty, InvalidProbabilityValueTooLow\)](#)
- [TEST \(ProbabilisticLogicProperty, InvalidProbabilityValueTooLowMinor\)](#)
- [TEST \(ProbabilisticLogicProperty, InvalidProbabilityValueTooHigh\)](#)
- [TEST \(ProbabilisticLogicProperty, InvalidProbabilityValueTooHighMinor\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectLogicProperty\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesLeftMissing\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesRightMissing\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesLeftExtra\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesRightExtra\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesLeftRightExtra\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesInverted\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParanthesesClosing\)](#)
- [TEST \(ProbabilisticLogicProperty, IncorrectlyEnclosingParantheses\)](#)
- [TEST \(ProbabilisticLogicProperty, Correct\)](#)
- [TEST \(ProbabilisticLogicProperty, ProbabilityMin\)](#)
- [TEST \(ProbabilisticLogicProperty, ProbabilityMax\)](#)
- [TEST \(ProbabilisticLogicProperty, ProbabilityLow\)](#)
- [TEST \(ProbabilisticLogicProperty, ProbabilityHigh\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeInvalidCharacter\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeMissingValuesAfterAndBeforeDot\)](#)

- [TEST \(SemanticType, IncorrectSemanticTypeMissingValueBeforeDot\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeMissingValueAfterDot\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeMissingValueBetweenDots\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeManyDots\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeInvalidValuesBeforeAndAfterDots\)](#)
- [TEST \(SemanticType, IncorrectSemanticTypeInvalidValuesBeforeAndAfterDots2\)](#)
- [TEST \(SemanticType, Correct\)](#)
- [TEST \(SimilarityMeasure, IncorrectSimilarityMeasure\)](#)
- [TEST \(SimilarityMeasure, CorrectAntiSimilar\)](#)
- [TEST \(SimilarityMeasure, CorrectSimilar\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputBeforeStartParanthesis\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputAfterStartParanthesis\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectFirstParameterType\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingFirstParameter\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectSecondParameterType\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingSecondParameter\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingSecondParameterExcludingComma\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingFirstAndSecondParameter\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputInvalidThirdParameterType\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputBeforeEndParanthesis\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputAfterEndParanthesis\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputExtraParameter\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputBetweenFirstAndSecondParameters\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, IncorrectInputBetweenSecondAndThirdParameters\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingParameters\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingParametersAndParantheses\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, MissingSimilarityMeasure\)](#)
- [TEST \(SimilarityTemporalNumericCollectionAttribute, Correct\)](#)
- [TEST \(SpatialMeasure, IncorrectSpatialMeasure\)](#)
- [TEST \(SpatialMeasure, CorrectClusteredness\)](#)
- [TEST \(SpatialMeasure, CorrectDensity\)](#)
- [TEST \(SpatialMeasure, CorrectArea\)](#)
- [TEST \(SpatialMeasure, CorrectPerimeter\)](#)
- [TEST \(SpatialMeasure, CorrectDistanceFromOrigin\)](#)
- [TEST \(SpatialMeasure, CorrectAngle\)](#)
- [TEST \(SpatialMeasure, CorrectTriangleMeasure\)](#)
- [TEST \(SpatialMeasure, CorrectRectangleMeasure\)](#)
- [TEST \(SpatialMeasure, CorrectCircleMeasure\)](#)
- [TEST \(SpatialMeasure, CorrectCentroidX\)](#)
- [TEST \(SpatialMeasure, CorrectCentroidY\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputBeforeSpatialMeasure\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectSpatialMeasure\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputAfterSpatialMeasure\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputMissingFirstParanthesis\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputMissingSecondParanthesis\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputInvalidSubset\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputMissingSubset\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputMissingSubsetAndParantheses\)](#)
- [TEST \(SpatialMeasureCollection, IncorrectInputMissingAll\)](#)
- [TEST \(SpatialMeasureCollection, Correct\)](#)
- [TEST \(Subset, IncorrectInputWrongSubsetAlternativeRegion\)](#)
- [TEST \(Subset, IncorrectInputWrongSubsetAlternativeCluster\)](#)
- [TEST \(Subset, Correct\)](#)
- [TEST \(SubsetOperation, IncorrectInputWrongSubsetOperationAlternative\)](#)
- [TEST \(SubsetOperation, CorrectDifference\)](#)

Reference

- [TEST](#) (SubsetOperation, CorrectIntersection)
- [TEST](#) (SubsetOperation, CorrectUnion)
- [TEST](#) (SubsetSpecific, IncorrectInputWrongSubsetAlternative)
- [TEST](#) (SubsetSpecific, CorrectClusters)
- [TEST](#) (SubsetSpecific, CorrectRegions)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputWrongAlternative)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputBeforeStartParanthesis)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputAfterStartParanthesis)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputMissingFirstArgument)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputMissingSeparatorComma)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputMissingCommaAndArgument)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputMissingSecondArgument)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputBeforeEndParanthesis)
- [TEST](#) (SubsetSubsetOperation, IncorrectInputAfterEndParanthesis)
- [TEST](#) (SubsetSubsetOperation, Correct)
- [TEST](#) (TemporalNumericCollection, IncorrectInputWrongAlternative)
- [TEST](#) (TemporalNumericCollection, Correct)
- [TEST](#) (TemporalNumericComparison, NumericMeasureFirst1)
- [TEST](#) (TemporalNumericComparison, NumericMeasureFirst2)
- [TEST](#) (TemporalNumericComparison, ComparatorFirst1)
- [TEST](#) (TemporalNumericComparison, ComparatorFirst2)
- [TEST](#) (TemporalNumericComparison, IncorrectOrder)
- [TEST](#) (TemporalNumericComparison, IncorrectInputMissingFirstOperand)
- [TEST](#) (TemporalNumericComparison, IncorrectInputMissingComparator)
- [TEST](#) (TemporalNumericComparison, IncorrectInputMissingSecondOperand)
- [TEST](#) (TemporalNumericComparison, IncorrectInputMissingBothOperands)
- [TEST](#) (TemporalNumericComparison, IncorrectInputMissingBothOperandsAndComparator)
- [TEST](#) (TemporalNumericComparison, IncorrectInputAfterNumericMeasure)
- [TEST](#) (TemporalNumericComparison, IncorrectInputAfterComparator)
- [TEST](#) (TemporalNumericComparison, Correct)
- [TEST](#) (TemporalNumericMeasure, WrongAlternative)
- [TEST](#) (TemporalNumericMeasure, Correct)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputBeforeBeginParanthesis)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingBeginParanthesis)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingBeginTimepoint)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingComma)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingEndTimepoint)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingEndTimepointAndComma)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputExtraTimepoint)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputMissingEndParanthesis)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputExtraSurroundingParantheses)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputBeforeNumericMeasure)
- [TEST](#) (TemporalNumericMeasureCollection, IncorrectInputAfterNumericMeasure)
- [TEST](#) (TemporalNumericMeasureCollection, Correct)
- [TEST](#) (TimeseriesComponent, IncorrectInputBeforeTimeseriesComponent)
- [TEST](#) (TimeseriesComponent, IncorrectParanthesisBeforeTimeseriesComponent)
- [TEST](#) (TimeseriesComponent, IncorrectInputAfterTimeseriesComponent)
- [TEST](#) (TimeseriesComponent, IncorrectParanthesisAfterTimeseriesComponent)
- [TEST](#) (TimeseriesComponent, IncorrectInputInvalidAlternative)
- [TEST](#) (TimeseriesComponent, Correct)
- [TEST](#) (TimeseriesMeasure, IncorrectInputWrongAlternative)
- [TEST](#) (TimeseriesMeasure, CorrectEnteringTime)
- [TEST](#) (TimeseriesMeasure, CorrectEnteringValue)
- [TEST](#) (TimeseriesTimeseriesComponent, IncorrectInputBeforeTimeseriesMeasure)
- [TEST](#) (TimeseriesTimeseriesComponent, IncorrectInputWrongTimeseriesMeasure)

- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputBeforeStartParanthesis\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputAfterStartParanthesis\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputInvalidFirstParameter\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputMissingFirstParameter\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputMissingCommaBetweenParameters\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputMissingSecondParameter\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputInvalidSecondParameter\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputBeforeEndParanthesis\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputAfterEndParanthesis\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputMissingParantheses\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputInvertedParantheses\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputDoubledParantheses\)](#)
- [TEST \(TimeseriesTimeseriesComponent, IncorrectInputWrongParanthesesType\)](#)
- [TEST \(TimeseriesTimeseriesComponent, Correct\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputMissingParameter\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputAfterStartBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputAfterEndBracket\)](#)
- [TEST \(UnaryNumericFilter, Correct\)](#)
- [TEST \(UnaryNumericMeasure, IncorrectUnaryNumericMeasure\)](#)
- [TEST \(UnaryNumericMeasure, CorrectAbs\)](#)
- [TEST \(UnaryNumericMeasure, CorrectCeil\)](#)
- [TEST \(UnaryNumericMeasure, CorrectFloor\)](#)
- [TEST \(UnaryNumericMeasure, CorrectRound\)](#)
- [TEST \(UnaryNumericMeasure, CorrectSign\)](#)
- [TEST \(UnaryNumericMeasure, CorrectSqrt\)](#)
- [TEST \(UnaryNumericMeasure, CorrectTrunc\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputMissingParameter\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputAfterStartBracket\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputAfterEndBracket\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputDoubleBrackets\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputMissingComparator\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputMissingFirstOperand\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputMissingSecondOperand\)](#)
- [TEST \(UnaryNumericNumeric, IncorrectInputMissingBothOperands\)](#)
- [TEST \(UnaryNumericNumeric, Correct\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputMissingParameter\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputAfterStartBracket\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputAfterEndBracket\)](#)
- [TEST \(UnaryNumericTemporal, IncorrectInputDoubleBrackets\)](#)
- [TEST \(UnaryNumericTemporal, Correct\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectSpatialMeasureBeforeConstraint\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputSpatialEntityInsteadOfSpatialMeasure\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingSpatialMeasure\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingComparator\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingNumericMeasure\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingComparatorNumericMeasure\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingSpatialMeasureNumericMeasure\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputMissingSpatialMeasureComparator\)](#)
- [TEST \(UnarySpatialConstraint, IncorrectInputEmptyConstraint\)](#)

Reference

- [TEST](#) (UnarySpatialConstraint, Correct)
- [TEST](#) (UnaryStatisticalMeasure, IncorrectUnaryStatisticalMeasure)
- [TEST](#) (UnaryStatisticalMeasure, CorrectAvg)
- [TEST](#) (UnaryStatisticalMeasure, CorrectCount)
- [TEST](#) (UnaryStatisticalMeasure, CorrectGeomean)
- [TEST](#) (UnaryStatisticalMeasure, CorrectHarmean)
- [TEST](#) (UnaryStatisticalMeasure, CorrectKurt)
- [TEST](#) (UnaryStatisticalMeasure, CorrectMax)
- [TEST](#) (UnaryStatisticalMeasure, CorrectMedian)
- [TEST](#) (UnaryStatisticalMeasure, CorrectMin)
- [TEST](#) (UnaryStatisticalMeasure, CorrectMode)
- [TEST](#) (UnaryStatisticalMeasure, CorrectProduct)
- [TEST](#) (UnaryStatisticalMeasure, CorrectSkew)
- [TEST](#) (UnaryStatisticalMeasure, CorrectStdev)
- [TEST](#) (UnaryStatisticalMeasure, CorrectSum)
- [TEST](#) (UnaryStatisticalMeasure, CorrectVar)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputNoSubset)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputBeforeStartBracket)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputAfterStartBracket)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputBeforeEndBracket)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputAfterEndBracket)
- [TEST](#) (UnaryStatisticalNumeric, IncorrectInputDoubleBrackets)
- [TEST](#) (UnaryStatisticalNumeric, Correct)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputNoSubset)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputBeforeStartBracket)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputAfterStartBracket)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputBeforeEndBracket)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputAfterEndBracket)
- [TEST](#) (UnaryStatisticalSpatial, IncorrectInputDoubleBrackets)
- [TEST](#) (UnaryStatisticalSpatial, Correct)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputWrongTypeKeywordExtraLetterAfter)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputWrongTypeKeywordExtraLetterBefore)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputBeforeTypeKeyword)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputAfterTypeKeyword)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputAfterComparator)
- [TEST](#) (UnaryTypeConstraint, IncorrectInputAfterFilterNumericMeasure)
- [TEST](#) (UnaryTypeConstraint, Correct)
- [TEST](#) (UntilLogicProperty, IncorrectInputMissingStartTimepoint)
- [TEST](#) (UntilLogicProperty, IncorrectInputMissingEndTimepoint)
- [TEST](#) (UntilLogicProperty, IncorrectInputMissingTimepoints)
- [TEST](#) (UntilLogicProperty, IncorrectInputMissingTimepointsAndBrackets)
- [TEST](#) (UntilLogicProperty, UntilOperatorAsUnaryBefore)
- [TEST](#) (UntilLogicProperty, UntilOperatorAsUnaryAfter)
- [TEST](#) (UntilLogicProperty, IncorrectInputBeforeUntilOperator)
- [TEST](#) (UntilLogicProperty, AdditionalOperatorBeforeUntilOperator)
- [TEST](#) (UntilLogicProperty, IncorrectInputAfterUntilOperator)
- [TEST](#) (UntilLogicProperty, AdditionalOperatorAfterUntilOperator)
- [TEST](#) (UntilLogicProperty, WrongInputBeforeStartParenthesis)
- [TEST](#) (UntilLogicProperty, WrongInputAfterStartParenthesis)
- [TEST](#) (UntilLogicProperty, MissingTimepointsComma)
- [TEST](#) (UntilLogicProperty, StartTimepointInvalid)
- [TEST](#) (UntilLogicProperty, StartTimepointRealNumber)
- [TEST](#) (UntilLogicProperty, EndTimepointInvalid)
- [TEST](#) (UntilLogicProperty, EndTimepointRealNumber)
- [TEST](#) (UntilLogicProperty, TimepointsInvalid)

- [TEST \(UntilLogicProperty, TimepointsRealNumber\)](#)
- [TEST \(UntilLogicProperty, WrongInputBeforeEndParenthesis\)](#)
- [TEST \(UntilLogicProperty, WrongInputAfterEndParenthesis\)](#)
- [TEST \(UntilLogicProperty, Correct\)](#)
- [TEST \(UntilLogicProperty, MultipleCorrect\)](#)
- [TEST \(Input, IncorrectTrueInput\)](#)
- [TEST \(Input, IncorrectTInput\)](#)
- [TEST \(Input, IncorrectFalseInput\)](#)
- [TEST \(Input, IncorrectFInput\)](#)

## Variables

- static const std::vector<std::string> [CONSTRAINTS\\_BINARY\\_OPERATORS](#) = std::vector<std::string>({" $\wedge$ ", "V", " $=>$ ", " $<=>$ "})
- static const std::vector<std::string> [LOGIC\\_PROPERTIES\\_BINARY\\_OPERATORS](#) = std::vector<std::string>({" $\wedge$ ", "V", " $=>$ ", " $<=>$ "})

### 8.344.1 Function Documentation

#### 8.344.1.1 TEST ( BinaryNumericFilter , IncorrectInputMissingParameterOne )

Definition at line 29 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

#### 8.344.1.2 TEST ( BinaryNumericFilter , IncorrectInputMissingParameterTwo )

Definition at line 33 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

#### 8.344.1.3 TEST ( BinaryNumericFilter , IncorrectInputMissingParametersOneTwo )

Definition at line 37 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

#### 8.344.1.4 TEST ( BinaryNumericFilter , IncorrectInputBeforeStartBracket )

Definition at line 41 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

#### 8.344.1.5 TEST ( BinaryNumericFilter , IncorrectInputAfterStartBracket )

Definition at line 45 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

#### 8.344.1.6 TEST ( BinaryNumericFilter , InvalidFirstParameter )

Definition at line 49 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.7 TEST ( BinaryNumericFilter , MissingParametersComma )

Definition at line 53 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.8 TEST ( BinaryNumericFilter , InvalidSecondParameter )

Definition at line 57 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.9 TEST ( BinaryNumericFilter , IncorrectInputBeforeEndBracket )

Definition at line 61 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.10 TEST ( BinaryNumericFilter , IncorrectInputAfterEndBracket )

Definition at line 65 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.11 TEST ( BinaryNumericFilter , Correct )

Definition at line 69 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.12 TEST ( BinaryNumericMeasure , IncorrectBinaryNumericMeasure )

Definition at line 83 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.13 TEST ( BinaryNumericMeasure , CorrectAdd )

Definition at line 87 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.14 TEST ( BinaryNumericMeasure , CorrectDiv )

Definition at line 91 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.15 TEST ( BinaryNumericMeasure , CorrectLog )

Definition at line 95 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.16 TEST( BinaryNumericMeasure , CorrectMod )**

Definition at line 99 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.17 TEST( BinaryNumericMeasure , CorrectMultiply )**

Definition at line 103 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.18 TEST( BinaryNumericMeasure , CorrectPower )**

Definition at line 107 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.19 TEST( BinaryNumericMeasure , CorrectSubtract )**

Definition at line 111 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.20 TEST( BinaryNumericNumeric , IncorrectInputMissingParameterOne )**

Definition at line 124 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.21 TEST( BinaryNumericNumeric , IncorrectInputMissingParameterTwo )**

Definition at line 128 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.22 TEST( BinaryNumericNumeric , IncorrectInputMissingParametersOneTwo )**

Definition at line 132 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.23 TEST( BinaryNumericNumeric , IncorrectInputBeforeStartBracket )**

Definition at line 136 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.24 TEST( BinaryNumericNumeric , IncorrectInputAfterStartBracket )**

Definition at line 140 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.25 TEST( BinaryNumericNumeric , InvalidFirstParameter )

Definition at line 144 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.26 TEST( BinaryNumericNumeric , MissingParametersComma )

Definition at line 148 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.27 TEST( BinaryNumericNumeric , InvalidSecondParameter )

Definition at line 152 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.28 TEST( BinaryNumericNumeric , IncorrectInputBeforeEndBracket )

Definition at line 156 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.29 TEST( BinaryNumericNumeric , IncorrectInputAfterEndBracket )

Definition at line 160 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.30 TEST( BinaryNumericNumeric , Correct )

Definition at line 164 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.31 TEST( BinaryNumericTemporal , IncorrectInputMissingParameterOne )

Definition at line 177 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.32 TEST( BinaryNumericTemporal , IncorrectInputMissingParameterTwo )

Definition at line 181 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.33 TEST( BinaryNumericTemporal , IncorrectInputMissingParametersOneTwo )

Definition at line 185 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.34 TEST ( BinaryNumericTemporal , IncorrectInputBeforeStartBracket )**

Definition at line 189 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.35 TEST ( BinaryNumericTemporal , IncorrectInputAfterStartBracket )**

Definition at line 193 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.36 TEST ( BinaryNumericTemporal , InvalidFirstParameter )**

Definition at line 197 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.37 TEST ( BinaryNumericTemporal , MissingParametersComma )**

Definition at line 201 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.38 TEST ( BinaryNumericTemporal , InvalidSecondParameter )**

Definition at line 205 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.39 TEST ( BinaryNumericTemporal , IncorrectInputBeforeEndBracket )**

Definition at line 209 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.40 TEST ( BinaryNumericTemporal , IncorrectInputAfterEndBracket )**

Definition at line 213 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.41 TEST ( BinaryNumericTemporal , Correct )**

Definition at line 217 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.42 TEST ( BinaryStatisticalMeasure , IncorrectQuaternarySubsetMeasure )**

Definition at line 230 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.43 TEST( BinaryStatisticalMeasure , CorrectCovar )

Definition at line 234 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.44 TEST( BinaryStatisticalNumeric , IncorrectInputMissingParameterOne )

Definition at line 247 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.45 TEST( BinaryStatisticalNumeric , IncorrectInputMissingParameterTwo )

Definition at line 251 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.46 TEST( BinaryStatisticalNumeric , IncorrectInputMissingParametersOneTwo )

Definition at line 255 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.47 TEST( BinaryStatisticalNumeric , IncorrectInputBeforeStartBracket )

Definition at line 259 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.48 TEST( BinaryStatisticalNumeric , IncorrectInputAfterStartBracket )

Definition at line 263 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.49 TEST( BinaryStatisticalNumeric , MissingComma )

Definition at line 267 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.50 TEST( BinaryStatisticalNumeric , IncorrectInputBeforeEndBracket )

Definition at line 271 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.51 TEST( BinaryStatisticalNumeric , IncorrectInputAfterEndBracket )

Definition at line 275 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.52 TEST( BinaryStatisticalNumeric , Correct )**

Definition at line 279 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.53 TEST( BinaryStatisticalQuantileMeasure , IncorrectBinaryStatisticalQuantileMeasure )**

Definition at line 292 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.54 TEST( BinaryStatisticalQuantileMeasure , CorrectPercentile )**

Definition at line 296 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.55 TEST( BinaryStatisticalQuantileMeasure , CorrectQuartile )**

Definition at line 300 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.56 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputMissingParameterOne )**

Definition at line 313 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.57 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputMissingParameterTwo )**

Definition at line 317 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.58 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputMissingParametersOneTwo )**

Definition at line 321 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.59 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputBeforeStartBracket )**

Definition at line 325 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.60 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputAfterStartBracket )**

Definition at line 329 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.61 TEST( BinaryStatisticalQuantileNumeric , MissingComma )

Definition at line 333 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.62 TEST( BinaryStatisticalQuantileNumeric , InvalidSpatialMeasureCollection )

Definition at line 337 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.63 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputBeforeEndBracket )

Definition at line 341 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.64 TEST( BinaryStatisticalQuantileNumeric , IncorrectInputAfterEndBracket )

Definition at line 345 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.65 TEST( BinaryStatisticalQuantileNumeric , Correct )

Definition at line 349 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.66 TEST( BinaryStatisticalQuantileSpatial , IncorrectInputMissingParameterOne )

Definition at line 362 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.67 TEST( BinaryStatisticalQuantileSpatial , IncorrectInputMissingParameterTwo )

Definition at line 366 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.68 TEST( BinaryStatisticalQuantileSpatial , IncorrectInputMissingParametersOneTwo )

Definition at line 370 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.69 TEST( BinaryStatisticalQuantileSpatial , IncorrectInputBeforeStartBracket )

Definition at line 374 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.70 TEST ( BinaryStatisticalQuantileSpatial , IncorrectInputAfterStartBracket )**

Definition at line 378 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.71 TEST ( BinaryStatisticalQuantileSpatial , MissingComma )**

Definition at line 382 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.72 TEST ( BinaryStatisticalQuantileSpatial , InvalidSpatialMeasureCollection )**

Definition at line 386 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.73 TEST ( BinaryStatisticalQuantileSpatial , IncorrectInputBeforeEndBracket )**

Definition at line 390 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.74 TEST ( BinaryStatisticalQuantileSpatial , IncorrectInputAfterEndBracket )**

Definition at line 394 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.75 TEST ( BinaryStatisticalQuantileSpatial , Correct )**

Definition at line 398 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.76 TEST ( BinaryStatisticalSpatial , IncorrectInputMissingParameterOne )**

Definition at line 411 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.77 TEST ( BinaryStatisticalSpatial , IncorrectInputMissingParameterTwo )**

Definition at line 415 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.78 TEST ( BinaryStatisticalSpatial , IncorrectInputMissingParametersOneTwo )**

Definition at line 419 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.79 TEST( BinaryStatisticalSpatial , IncorrectInputBeforeStartBracket )

Definition at line 423 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.80 TEST( BinaryStatisticalSpatial , IncorrectInputAfterStartBracket )

Definition at line 427 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.81 TEST( BinaryStatisticalSpatial , MissingComma )

Definition at line 431 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.82 TEST( BinaryStatisticalSpatial , IncorrectInputBeforeEndBracket )

Definition at line 435 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.83 TEST( BinaryStatisticalSpatial , IncorrectInputAfterEndBracket )

Definition at line 439 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.84 TEST( BinaryStatisticalSpatial , Correct )

Definition at line 443 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.85 TEST( ChangeMeasure , IncorrectChangeMeasure )

Definition at line 456 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.86 TEST( ChangeMeasure , CorrectDifference )

Definition at line 460 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.87 TEST( ChangeMeasure , CorrectRatio )

Definition at line 464 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.88 TEST ( ChangeTemporalNumericCollection , IncorrectChangeMeasureDf )**

Definition at line 477 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.89 TEST ( ChangeTemporalNumericCollection , IncorrectChangeMeasureDiff )**

Definition at line 481 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.90 TEST ( ChangeTemporalNumericCollection , IncorrectChangeMeasureRatio )**

Definition at line 485 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.91 TEST ( ChangeTemporalNumericCollection , IncorrectInputBeforeChangeMeasure )**

Definition at line 489 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.92 TEST ( ChangeTemporalNumericCollection , IncorrectInputBeforeStartParanthesis )**

Definition at line 493 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.93 TEST ( ChangeTemporalNumericCollection , IncorrectInputAfterStartParanthesis )**

Definition at line 497 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.94 TEST ( ChangeTemporalNumericCollection , MissingParameter )**

Definition at line 501 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.95 TEST ( ChangeTemporalNumericCollection , IncorrectInputMissingParameterAndBrackets )**

Definition at line 505 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.96 TEST ( ChangeTemporalNumericCollection , IncorrectOpeningBracket )**

Definition at line 509 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.97 TEST( ChangeTemporalNumericCollection , IncorrectClosingBracket )

Definition at line 513 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.98 TEST( ChangeTemporalNumericCollection , IncorrectBrackets )

Definition at line 517 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.99 TEST( ChangeTemporalNumericCollection , IncorrectBracketsInverted )

Definition at line 521 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.100 TEST( ChangeTemporalNumericCollection , IncorrectBracketsDoubled )

Definition at line 525 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.101 TEST( ChangeTemporalNumericCollection , IncorrectInputBeforeEndParanthesis )

Definition at line 529 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.102 TEST( ChangeTemporalNumericCollection , IncorrectInputAfterEndParanthesis )

Definition at line 533 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.103 TEST( ChangeTemporalNumericCollection , IncorrectTemporalNumericCollectionRealNumber )

Definition at line 537 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.104 TEST( ChangeTemporalNumericCollection , IncorrectTemporalNumericCollectionNumericStateVariable )

Definition at line 541 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.105 TEST( ChangeTemporalNumericCollection , IncorrectTemporalNumericCollectionNumericSpatialMeasure )

Definition at line 545 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.106 TEST ( ChangeTemporalNumericCollection , Correct )**

Definition at line 549 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.107 TEST ( ChangeTemporalNumericMeasure , IncorrectChangeMeasure )**

Definition at line 562 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.108 TEST ( ChangeTemporalNumericMeasure , IncorrectInputBeforeChangeMeasure )**

Definition at line 566 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.109 TEST ( ChangeTemporalNumericMeasure , IncorrectInputBeforeStartParanthesis )**

Definition at line 570 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.110 TEST ( ChangeTemporalNumericMeasure , IncorrectInputAfterStartParanthesis )**

Definition at line 574 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.111 TEST ( ChangeTemporalNumericMeasure , MissingParameter )**

Definition at line 578 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.112 TEST ( ChangeTemporalNumericMeasure , IncorrectInputMissingParameterAndBrackets )**

Definition at line 582 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.113 TEST ( ChangeTemporalNumericMeasure , IncorrectOpeningBracket )**

Definition at line 586 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.114 TEST ( ChangeTemporalNumericMeasure , IncorrectClosingBracket )**

Definition at line 590 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.115 TEST ( ChangeTemporalNumericMeasure , IncorrectBrackets )

Definition at line 594 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.116 TEST ( ChangeTemporalNumericMeasure , IncorrectBracketsInverted )

Definition at line 598 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.117 TEST ( ChangeTemporalNumericMeasure , IncorrectBracketsDoubled )

Definition at line 602 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.118 TEST ( ChangeTemporalNumericMeasure , IncorrectInputBeforeEndParanthesis )

Definition at line 606 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.119 TEST ( ChangeTemporalNumericMeasure , IncorrectInputAfterEndParanthesis )

Definition at line 610 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.120 TEST ( ChangeTemporalNumericMeasure , MissingComparator )

Definition at line 614 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.121 TEST ( ChangeTemporalNumericMeasure , IncorrectEndOperand )

Definition at line 618 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.122 TEST ( ChangeTemporalNumericMeasure , Correct )

Definition at line 622 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.123 TEST ( Comparator , IncorrectEqual )

Definition at line 635 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.124 TEST ( Comparator , IncorrectDifferent1 )**

Definition at line 639 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.125 TEST ( Comparator , IncorrectDifferent2 )**

Definition at line 643 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.126 TEST ( Comparator , CorrectGreaterThan )**

Definition at line 647 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.127 TEST ( Comparator , CorrectLessThan )**

Definition at line 651 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.128 TEST ( Comparator , CorrectGreaterThanOrEqualTo )**

Definition at line 655 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.129 TEST ( Comparator , CorrectLessThanOrEqualTo )**

Definition at line 659 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.130 TEST ( Comparator , CorrectEqual )**

Definition at line 663 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.131 TEST ( CompoundConstraint , MissingBinaryOperator )**

Definition at line 683 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.132 TEST ( CompoundConstraint , MissingConstraints )**

Definition at line 687 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.133 TEST ( CompoundConstraint , MissingFirstConstraint )

Definition at line 693 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.134 TEST ( CompoundConstraint , MissingSecondConstraint )

Definition at line 699 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.135 TEST ( CompoundConstraint , BinaryOperatorAsUnaryBefore )

Definition at line 705 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.136 TEST ( CompoundConstraint , BinaryOperatorAsUnaryAfter )

Definition at line 711 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.137 TEST ( CompoundConstraint , TemporalNumericComparisonBeforeBinaryOperator )

Definition at line 717 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.138 TEST ( CompoundConstraint , UnaryNumericMeasureAfterBinaryOperator )

Definition at line 723 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.139 TEST ( CompoundConstraint , AdditionalOperatorBeforeBinaryOperator )

Definition at line 729 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.140 TEST ( CompoundConstraint , AdditionalOperatorAfterBinaryOperator )

Definition at line 735 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.141 TEST ( CompoundConstraint , Correct )

Definition at line 741 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.142 TEST ( CompoundConstraint , MultipleCorrect )**

Definition at line 747 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.143 TEST ( CompoundLogicProperty , MissingBinaryOperator )**

Definition at line 769 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.144 TEST ( CompoundLogicProperty , MissingLogicProperties )**

Definition at line 773 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.145 TEST ( CompoundLogicProperty , MissingFirstLogicProperty )**

Definition at line 779 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.146 TEST ( CompoundLogicProperty , MissingSecondLogicProperty )**

Definition at line 785 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.147 TEST ( CompoundLogicProperty , BinaryOperatorAsUnaryBefore )**

Definition at line 791 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.148 TEST ( CompoundLogicProperty , BinaryOperatorAsUnaryAfter )**

Definition at line 797 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.149 TEST ( CompoundLogicProperty , UnaryStatisticalMeasureBeforeBinaryOperator )**

Definition at line 803 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.150 TEST ( CompoundLogicProperty , UnaryNumericMeasureAfterBinaryOperator )**

Definition at line 809 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.151 TEST ( CompoundLogicProperty , AdditionalOperatorBeforeBinaryOperator )

Definition at line 815 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.152 TEST ( CompoundLogicProperty , AdditionalOperatorAfterBinaryOperator )

Definition at line 821 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.153 TEST ( CompoundLogicProperty , Correct )

Definition at line 827 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.154 TEST ( CompoundLogicProperty , MultipleCorrect )

Definition at line 833 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.155 TEST ( ConstraintEnclosedByParentheses , MissingParenthesisRight )

Definition at line 848 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.156 TEST ( ConstraintEnclosedByParentheses , MissingParenthesisLeft )

Definition at line 852 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.157 TEST ( ConstraintEnclosedByParentheses , ExtraParenthesisLeft )

Definition at line 856 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.158 TEST ( ConstraintEnclosedByParentheses , ExtraParenthesisRight )

Definition at line 860 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.159 TEST ( ConstraintEnclosedByParentheses , InvertedParentheses )

Definition at line 864 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.160 TEST ( ConstraintEnclosedByParentheses , ExtraParenthesesBothSides )**

Definition at line 868 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.161 TEST ( ConstraintEnclosedByParentheses , ParenthesesInWrongOrder )**

Definition at line 872 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.162 TEST ( ConstraintEnclosedByParentheses , Correct )**

Definition at line 876 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.163 TEST ( ConstraintEnclosedByParentheses , CorrectDoubled )**

Definition at line 880 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.164 TEST ( ConstraintEnclosedByParentheses , CorrectQuadrupled )**

Definition at line 884 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.165 TEST ( Constraint , ExtraInputBeforeConstraint )**

Definition at line 897 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.166 TEST ( Constraint , ExtraInputAfterConstraint )**

Definition at line 901 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.167 TEST ( Constraint , Correct )**

Definition at line 905 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.168 TEST ( FilterSubset , IncorrectAlternative )**

Definition at line 918 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.169 TEST ( FilterSubset , CorrectSpatialMeasureRealValue )

Definition at line 922 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.170 TEST ( FilterSubset , CorrectSpatialMeasures )

Definition at line 926 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.171 TEST ( FilterSubset , CorrectMultiple )

Definition at line 930 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.172 TEST ( FilterSubset , CorrectMultipleComplex )

Definition at line 934 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.173 TEST ( FilterSubset , IncorrectInputMisspelledFilter )

Definition at line 947 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.174 TEST ( FilterSubset , IncorrectInputBeforeStartBracket )

Definition at line 951 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.175 TEST ( FilterSubset , IncorrectInputAfterStartBracket )

Definition at line 955 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.176 TEST ( FilterSubset , IncorrectInputMissingComma )

Definition at line 959 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.177 TEST ( FilterSubset , IncorrectInputBeforeEndBracket )

Definition at line 963 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.178 TEST ( FilterSubset , IncorrectInputAfterEndBracket )**

Definition at line 967 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.179 TEST ( FilterSubset , Correct )**

Definition at line 971 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.180 TEST ( FutureLogicProperty , WrongInputMissingStartTimepoint )**

Definition at line 984 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.181 TEST ( FutureLogicProperty , WrongInputMissingEndTimepoint )**

Definition at line 988 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.182 TEST ( FutureLogicProperty , WrongInputMissingTimepoints )**

Definition at line 992 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.183 TEST ( FutureLogicProperty , WrongInputMissingTimepointsAndBrackets )**

Definition at line 996 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.184 TEST ( FutureLogicProperty , WrongInputBeforeStartParanthesis )**

Definition at line 1000 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.185 TEST ( FutureLogicProperty , WrongInputAfterStartParanthesis )**

Definition at line 1004 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.186 TEST ( FutureLogicProperty , MissingTimepointComma )**

Definition at line 1008 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.187 TEST ( FutureLogicProperty , InvalidStartTimepoint )

Definition at line 1012 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.188 TEST ( FutureLogicProperty , InvalidEndTimepoint )

Definition at line 1016 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.189 TEST ( FutureLogicProperty , InvalidTimepoints )

Definition at line 1020 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.190 TEST ( FutureLogicProperty , WrongInputBeforeEndParanthesis )

Definition at line 1024 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.191 TEST ( FutureLogicProperty , WrongInputAfterEndParanthesis )

Definition at line 1028 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.192 TEST ( FutureLogicProperty , Correct )

Definition at line 1032 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.193 TEST ( GlobalLogicProperty , WrongInputMissingStartTimepoint )

Definition at line 1045 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.194 TEST ( GlobalLogicProperty , WrongInputMissingEndTimepoint )

Definition at line 1049 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.195 TEST ( GlobalLogicProperty , WrongInputMissingTimepoints )

Definition at line 1053 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.196 TEST ( GlobalLogicProperty , WrongInputMissingTimepointsAndBrackets )**

Definition at line 1057 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.197 TEST ( GlobalLogicProperty , WrongInputBeforeStartParanthesis )**

Definition at line 1061 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.198 TEST ( GlobalLogicProperty , WrongInputAfterStartParanthesis )**

Definition at line 1065 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.199 TEST ( GlobalLogicProperty , MissingTimepointComma )**

Definition at line 1069 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.200 TEST ( GlobalLogicProperty , InvalidStartTimepoint )**

Definition at line 1073 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.201 TEST ( GlobalLogicProperty , InvalidEndTimepoint )**

Definition at line 1077 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.202 TEST ( GlobalLogicProperty , InvalidTimepoints )**

Definition at line 1081 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.203 TEST ( GlobalLogicProperty , WrongInputBeforeEndParanthesis )**

Definition at line 1085 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.204 TEST ( GlobalLogicProperty , WrongInputAfterEndParanthesis )**

Definition at line 1089 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.205 TEST ( GlobalLogicProperty , Correct )**

Definition at line 1093 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.206 TEST ( HeterogeneousTimeseriesComponent , WrongAlternativeSpike )**

Definition at line 1106 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.207 TEST ( HeterogeneousTimeseriesComponent , WrongAlternativeSink )**

Definition at line 1110 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.208 TEST ( HeterogeneousTimeseriesComponent , WrongAlternativeTrough )**

Definition at line 1114 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.209 TEST ( HeterogeneousTimeseriesComponent , CorrectPeak )**

Definition at line 1118 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.210 TEST ( HeterogeneousTimeseriesComponent , CorrectValley )**

Definition at line 1122 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.211 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputBeforeHomogeneousTimeseriesMeasure )**

Definition at line 1135 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.212 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputWrongHomogeneousTimeseriesMeasure )**

Definition at line 1139 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.213 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputBeforeStartParanthesis )**

Definition at line 1143 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.214 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputAfterStartParanthesis )**

Definition at line 1147 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.215 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputInvalidFirstParameter )**

Definition at line 1151 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.216 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputMissingFirstParameter )**

Definition at line 1155 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.217 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputMissingCommaBetweenParameters )**

Definition at line 1159 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.218 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputMissingSecondParameter )**

Definition at line 1163 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.219 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputInvalidSecondParameter )**

Definition at line 1167 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.220 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputBeforeEndParanthesis )**

Definition at line 1171 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.221 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputAfterEndParanthesis )**

Definition at line 1175 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.222 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputMissingParantheses )**

Definition at line 1179 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.223 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputInvertedParentheses )

Definition at line 1183 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.224 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputDoubledParentheses )

Definition at line 1187 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.225 TEST ( HomogeneousHomogeneousTimeseries , IncorrectInputWrongParenthesesType )

Definition at line 1191 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.226 TEST ( HomogeneousHomogeneousTimeseries , CorrectNumericStateVariables )

Definition at line 1195 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.227 TEST ( HomogeneousHomogeneousTimeseries , CorrectSpatialEntities )

Definition at line 1199 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.228 TEST ( HomogeneousTimeseriesComponent , IncorrectAlternativeHeterogeneousTimeseriesComponent )

Definition at line 1212 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.229 TEST ( HomogeneousTimeseriesComponent , CorrectAscent )

Definition at line 1216 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.230 TEST ( HomogeneousTimeseriesComponent , CorrectDescent )

Definition at line 1220 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.231 TEST ( HomogeneousTimeseriesComponent , CorrectPlateau )

Definition at line 1224 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.232 TEST ( HomogeneousTimeseriesComponent , CorrectUniformAscent )**

Definition at line 1228 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.233 TEST ( HomogeneousTimeseriesComponent , CorrectUniformDescent )**

Definition at line 1232 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.234 TEST ( HomogeneousTimeseriesMeasure , IncorrectAlternativeTimeseriesMeasure )**

Definition at line 1245 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.235 TEST ( HomogeneousTimeseriesMeasure , CorrectTimeSpan )**

Definition at line 1249 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.236 TEST ( HomogeneousTimeseriesMeasure , CorrectValue )**

Definition at line 1253 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.237 TEST ( LogicPropertyEnclosedByParentheses , MissingParenthesisRight )**

Definition at line 1266 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.238 TEST ( LogicPropertyEnclosedByParentheses , MissingParenthesisLeft )**

Definition at line 1270 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.239 TEST ( LogicPropertyEnclosedByParentheses , ExtraParenthesisLeft )**

Definition at line 1274 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.240 TEST ( LogicPropertyEnclosedByParentheses , ExtraParenthesisRight )**

Definition at line 1278 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.241 TEST ( LogicPropertyEnclosedByParentheses , InvertedParentheses )

Definition at line 1282 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.242 TEST ( LogicPropertyEnclosedByParentheses , ExtraParenthesesBothSides )

Definition at line 1286 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.243 TEST ( LogicPropertyEnclosedByParentheses , ParenthesesInWrongOrder )

Definition at line 1290 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.244 TEST ( LogicPropertyEnclosedByParentheses , Correct )

Definition at line 1294 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.245 TEST ( LogicPropertyEnclosedByParentheses , CorrectDoubled )

Definition at line 1298 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.246 TEST ( LogicPropertyEnclosedByParentheses , CorrectQuadrupled )

Definition at line 1302 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.247 TEST ( LogicProperty , ExtraInputBeforeLogicProperty )

Definition at line 1315 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.248 TEST ( LogicProperty , ExtraInputInsideLogicProperty )

Definition at line 1319 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.249 TEST ( LogicProperty , ExtraInputAfterLogicProperty )

Definition at line 1323 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.250 TEST ( LogicProperty , Correct )**

Definition at line 1327 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.251 TEST ( MultipleLogicProperties , Correct1 )**

Definition at line 1340 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.252 TEST ( MultipleLogicProperties , Correct2 )**

Definition at line 1344 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.253 TEST ( NextKLogicProperty , IncorrectInputMissingTimepoint )**

Definition at line 1357 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.254 TEST ( NextKLogicProperty , IncorrectInputAfterNextSymbol )**

Definition at line 1361 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.255 TEST ( NextKLogicProperty , IncorrectValueForNextTimepoints )**

Definition at line 1365 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.256 TEST ( NextKLogicProperty , RealValueForNextTimepoints )**

Definition at line 1369 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.257 TEST ( NextKLogicProperty , IncorrectInputBeforeLogicProperty )**

Definition at line 1373 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.258 TEST ( NextKLogicProperty , Correct )**

Definition at line 1377 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.259 TEST ( NextLogicProperty , IncorrectNextSymbol )

Definition at line 1390 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.260 TEST ( NextLogicProperty , IncorrectInputAfterNextSymbol )

Definition at line 1394 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.261 TEST ( NextLogicProperty , Correct )

Definition at line 1398 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.262 TEST ( NotConstraint , IncorrectOperator )

Definition at line 1411 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.263 TEST ( NotConstraint , OperatorAfterConstraint )

Definition at line 1415 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.264 TEST ( NotConstraint , OperatorAfterConstraintAndExtraConstraint )

Definition at line 1419 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.265 TEST ( NotConstraint , OperatorBeforeConstraintAndExtraConstraint )

Definition at line 1423 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.266 TEST ( NotConstraint , Correct )

Definition at line 1427 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.267 TEST ( NotLogicProperty , OperatorAfterLogicProperty )

Definition at line 1440 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.268 TEST ( NotLogicProperty , OperatorAfterLogicPropertyAndExtraLogicProperty )**

Definition at line 1444 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.269 TEST ( NotLogicProperty , OperatorBeforeLogicPropertyAndExtraLogicProperty )**

Definition at line 1448 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.270 TEST ( NotLogicProperty , Correct )**

Definition at line 1452 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.271 TEST ( NumericMeasure , WrongAlternative )**

Definition at line 1465 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.272 TEST ( NumericMeasure , Correct )**

Definition at line 1469 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.273 TEST ( NumericMeasureCollection , WrongAlternative )**

Definition at line 1482 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.274 TEST ( NumericMeasureCollection , Correct )**

Definition at line 1486 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.275 TEST ( NumericSpatialMeasure , IncorrectAlternative )**

Definition at line 1499 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.276 TEST ( NumericSpatialMeasure , Correct )**

Definition at line 1503 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.277 TEST ( NumericStateVariable , MissingLeftCurlyBrace )

Definition at line 1516 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.278 TEST ( NumericStateVariable , MissingRightCurlyBrace )

Definition at line 1520 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.279 TEST ( NumericStateVariable , ExtraLeftCurlyBrace )

Definition at line 1524 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.280 TEST ( NumericStateVariable , ExtraRightCurlyBrace )

Definition at line 1528 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.281 TEST ( NumericStateVariable , InvertedCurlyBraces )

Definition at line 1532 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.282 TEST ( NumericStateVariable , DoubleCurlyBraces )

Definition at line 1536 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.283 TEST ( NumericStateVariable , TripleCurlyBraces )

Definition at line 1540 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.284 TEST ( NumericStateVariable , IncorrectSquareBrackets )

Definition at line 1544 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.285 TEST ( NumericStateVariable , IncorrectRoundBrackets )

Definition at line 1548 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.286 TEST ( NumericStateVariable , MissingLeftParanthesisForTypeConstraint )**

Definition at line 1552 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.287 TEST ( NumericStateVariable , MissingRightParanthesisForTypeConstraint )**

Definition at line 1556 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.288 TEST ( NumericStateVariable , MissingBothParanthesesForTypeConstraint )**

Definition at line 1560 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.289 TEST ( NumericStateVariable , MissingTypeKeywordForTypeConstraint )**

Definition at line 1564 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.290 TEST ( NumericStateVariable , MissingEqualComparatorForTypeConstraint )**

Definition at line 1568 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.291 TEST ( NumericStateVariable , MissingTypeValueForTypeConstraint )**

Definition at line 1572 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.292 TEST ( NumericStateVariable , MissingTypeKeywordAndEqualComparatorForTypeConstraint )**

Definition at line 1576 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.293 TEST ( NumericStateVariable , MissingAllForTypeConstraint )**

Definition at line 1580 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.294 TEST ( NumericStateVariable , IncorrectLeftParanthesisForTypeConstraint )**

Definition at line 1584 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.295 TEST ( NumericStateVariable , IncorrectRightParanthesisForTypeConstraint )

Definition at line 1588 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.296 TEST ( NumericStateVariable , IncorrectParanthesesForTypeConstraint )

Definition at line 1592 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.297 TEST ( NumericStateVariable , IncorrectTypeSpecifierForTypeConstraint )

Definition at line 1596 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.298 TEST ( NumericStateVariable , IncorrectTypeSpecifierForTypeConstraint2 )

Definition at line 1600 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.299 TEST ( NumericStateVariable , IncorrectComparatorForTypeConstraint )

Definition at line 1604 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.300 TEST ( NumericStateVariable , IncorrectTypeInvalidCharacterForTypeConstraint )

Definition at line 1608 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.301 TEST ( NumericStateVariable , IncorrectTypeMissingValueBeforeDotForTypeConstraint )

Definition at line 1612 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.302 TEST ( NumericStateVariable , IncorrectTypeMissingValueAfterDotForTypeConstraint )

Definition at line 1616 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.303 TEST ( NumericStateVariable , IncorrectTypeMissingValueBetweenDotsForTypeConstraint )

Definition at line 1620 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.304 TEST ( NumericStateVariable , MissingNumericStateVariableName )**

Definition at line 1624 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.305 TEST ( NumericStateVariable , MissingNumericStateVariableNameAndBraces )**

Definition at line 1628 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.306 TEST ( NumericStateVariable , CorrectNumericStateVariableSimple1 )**

Definition at line 1632 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.307 TEST ( NumericStateVariable , CorrectNumericStateVariableSimple2 )**

Definition at line 1636 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.308 TEST ( NumericStateVariable , CorrectNumericStateVariableSimple3 )**

Definition at line 1640 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.309 TEST ( NumericStateVariable , CorrectNumericStateVariableWithTypeConstraint4 )**

Definition at line 1644 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.310 TEST ( NumericStateVariable , CorrectNumericStateVariableWithTypeConstraint5 )**

Definition at line 1648 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.311 TEST ( NumericStateVariable , CorrectNumericStateVariableWithTypeConstraint6 )**

Definition at line 1652 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.312 TEST ( NumericStatisticalMeasure , IncorrectAlternative )**

Definition at line 1665 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.313 TEST ( NumericStatisticalMeasure , Correct )

Definition at line 1669 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.314 TEST ( ProbabilisticLogicProperty , IncorrectProbabilitySymbol )

Definition at line 1682 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.315 TEST ( ProbabilisticLogicProperty , IncorrectProbabilitySymbol2 )

Definition at line 1686 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.316 TEST ( ProbabilisticLogicProperty , IncorrectComparator )

Definition at line 1690 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.317 TEST ( ProbabilisticLogicProperty , IncorrectEqualComparator )

Definition at line 1694 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.318 TEST ( ProbabilisticLogicProperty , InvalidProbabilityValueTooLow )

Definition at line 1698 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.319 TEST ( ProbabilisticLogicProperty , InvalidProbabilityValueTooLowMinor )

Definition at line 1702 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.320 TEST ( ProbabilisticLogicProperty , InvalidProbabilityValueTooHigh )

Definition at line 1706 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.321 TEST ( ProbabilisticLogicProperty , InvalidProbabilityValueTooHighMinor )

Definition at line 1710 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.322 TEST ( ProbabilisticLogicProperty , IncorrectLogicProperty )**

Definition at line 1714 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.323 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftMissing )**

Definition at line 1718 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.324 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesRightMissing )**

Definition at line 1722 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.325 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftExtra )**

Definition at line 1726 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.326 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesRightExtra )**

Definition at line 1730 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.327 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftRightExtra )**

Definition at line 1734 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.328 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesInverted )**

Definition at line 1738 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.329 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesClosing )**

Definition at line 1742 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.330 TEST ( ProbabilisticLogicProperty , IncorrectlyEnclosingParantheses )**

Definition at line 1746 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.331 TEST ( ProbabilisticLogicProperty , Correct )

Definition at line 1750 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.332 TEST ( ProbabilisticLogicProperty , ProbabilityMin )

Definition at line 1754 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.333 TEST ( ProbabilisticLogicProperty , ProbabilityMax )

Definition at line 1758 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.334 TEST ( ProbabilisticLogicProperty , ProbabilityLow )

Definition at line 1762 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.335 TEST ( ProbabilisticLogicProperty , ProbabilityHigh )

Definition at line 1766 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.336 TEST ( SemanticType , IncorrectSemanticTypeInvalidCharacter )

Definition at line 1779 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.337 TEST ( SemanticType , IncorrectSemanticTypeMissingValuesAfterAndBeforeDot )

Definition at line 1783 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.338 TEST ( SemanticType , IncorrectSemanticTypeMissingValueBeforeDot )

Definition at line 1787 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.339 TEST ( SemanticType , IncorrectSemanticTypeMissingValueAfterDot )

Definition at line 1791 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.340 TEST ( SemanticType , IncorrectSemanticTypeMissingValueBetweenDots )**

Definition at line 1795 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.341 TEST ( SemanticType , IncorrectSemanticTypeManyDots )**

Definition at line 1799 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.342 TEST ( SemanticType , IncorrectSemanticTypeInvalidValuesBeforeAndAfterDots )**

Definition at line 1803 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.343 TEST ( SemanticType , IncorrectSemanticTypeInvalidValuesBeforeAndAfterDots2 )**

Definition at line 1807 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.344 TEST ( SemanticType , Correct )**

Definition at line 1811 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.345 TEST ( SimilarityMeasure , IncorrectSimilarityMeasure )**

Definition at line 1824 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.346 TEST ( SimilarityMeasure , CorrectAntiSimilar )**

Definition at line 1828 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.347 TEST ( SimilarityMeasure , CorrectSimilar )**

Definition at line 1832 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.348 TEST ( SimilarityTemporalNumericCollectionAttribute , IncorrectInputBeforeStartParanthesis )**

Definition at line 1845 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.349 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **IncorrectInputAfterStartParanthesis** )

Definition at line 1849 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.350 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **IncorrectFirstParameterType** )

Definition at line 1853 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.351 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **MissingFirstParameter** )

Definition at line 1857 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.352 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **IncorrectSecondParameterType** )

Definition at line 1861 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.353 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **MissingSecondParameter** )

Definition at line 1865 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.354 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **MissingSecondParameterExcludingComma** )

Definition at line 1869 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.355 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **MissingFirstAndSecondParameter** )

Definition at line 1873 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.356 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **IncorrectInputInvalidThirdParameterType** )

Definition at line 1877 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.357 TEST ( **SimilarityTemporalNumericCollectionAttribute** , **IncorrectInputBeforeEndParanthesis** )

Definition at line 1881 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.358 TEST ( SimilarityTemporalNumericCollectionAttribute , IncorrectInputAfterEndParanthesis )**

Definition at line 1885 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.359 TEST ( SimilarityTemporalNumericCollectionAttribute , IncorrectInputExtraParameter )**

Definition at line 1889 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.360 TEST ( SimilarityTemporalNumericCollectionAttribute , IncorrectInputBetweenFirstAndSecondParameters )**

Definition at line 1893 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.361 TEST ( SimilarityTemporalNumericCollectionAttribute , IncorrectInputBetweenSecondAndThirdParameters )**

Definition at line 1897 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.362 TEST ( SimilarityTemporalNumericCollectionAttribute , MissingParameters )**

Definition at line 1901 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.363 TEST ( SimilarityTemporalNumericCollectionAttribute , MissingParametersAndParantheses )**

Definition at line 1905 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.364 TEST ( SimilarityTemporalNumericCollectionAttribute , MissingSimilarityMeasure )**

Definition at line 1909 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.365 TEST ( SimilarityTemporalNumericCollectionAttribute , Correct )**

Definition at line 1913 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.366 TEST ( SpatialMeasure , IncorrectSpatialMeasure )**

Definition at line 1926 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.367 TEST ( SpatialMeasure , CorrectClusteredness )

Definition at line 1930 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.368 TEST ( SpatialMeasure , CorrectDensity )

Definition at line 1934 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.369 TEST ( SpatialMeasure , CorrectArea )

Definition at line 1938 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.370 TEST ( SpatialMeasure , CorrectPerimeter )

Definition at line 1942 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.371 TEST ( SpatialMeasure , CorrectDistanceFromOrigin )

Definition at line 1946 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.372 TEST ( SpatialMeasure , CorrectAngle )

Definition at line 1950 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.373 TEST ( SpatialMeasure , CorrectTriangleMeasure )

Definition at line 1954 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.374 TEST ( SpatialMeasure , CorrectRectangleMeasure )

Definition at line 1958 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.375 TEST ( SpatialMeasure , CorrectCircleMeasure )

Definition at line 1962 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.376 TEST ( SpatialMeasure , CorrectCentroidX )**

Definition at line 1966 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.377 TEST ( SpatialMeasure , CorrectCentroidY )**

Definition at line 1970 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.378 TEST ( SpatialMeasureCollection , IncorrectInputBeforeSpatialMeasure )**

Definition at line 1983 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.379 TEST ( SpatialMeasureCollection , IncorrectSpatialMeasure )**

Definition at line 1987 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.380 TEST ( SpatialMeasureCollection , IncorrectInputAfterSpatialMeasure )**

Definition at line 1991 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.381 TEST ( SpatialMeasureCollection , IncorrectInputMissingFirstParanthesis )**

Definition at line 1995 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.382 TEST ( SpatialMeasureCollection , IncorrectInputMissingSecondParanthesis )**

Definition at line 1999 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.383 TEST ( SpatialMeasureCollection , IncorrectInputInvalidSubset )**

Definition at line 2003 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.384 TEST ( SpatialMeasureCollection , IncorrectInputMissingSubset )**

Definition at line 2007 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.385 TEST ( SpatialMeasureCollection , IncorrectInputMissingSubsetAndParantheses )

Definition at line 2011 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.386 TEST ( SpatialMeasureCollection , IncorrectInputMissingAll )

Definition at line 2015 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.387 TEST ( SpatialMeasureCollection , Correct )

Definition at line 2019 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.388 TEST ( Subset , IncorrectInputWrongSubsetAlternativeRegion )

Definition at line 2032 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.389 TEST ( Subset , IncorrectInputWrongSubsetAlternativeCluster )

Definition at line 2036 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.390 TEST ( Subset , Correct )

Definition at line 2040 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.391 TEST ( SubsetOperation , IncorrectInputWrongSubsetOperationAlternative )

Definition at line 2053 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.392 TEST ( SubsetOperation , CorrectDifference )

Definition at line 2057 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.393 TEST ( SubsetOperation , CorrectIntersection )

Definition at line 2061 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.394 TEST ( SubsetOperation , CorrectUnion )**

Definition at line 2065 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.395 TEST ( SubsetSpecific , IncorrectInputWrongSubsetAlternative )**

Definition at line 2078 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.396 TEST ( SubsetSpecific , CorrectClusters )**

Definition at line 2082 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.397 TEST ( SubsetSpecific , CorrectRegions )**

Definition at line 2086 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.398 TEST ( SubsetSubsetOperation , IncorrectInputWrongAlternative )**

Definition at line 2099 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.399 TEST ( SubsetSubsetOperation , IncorrectInputBeforeStartParanthesis )**

Definition at line 2103 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.400 TEST ( SubsetSubsetOperation , IncorrectInputAfterStartParanthesis )**

Definition at line 2107 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.401 TEST ( SubsetSubsetOperation , IncorrectInputMissingFirstArgument )**

Definition at line 2111 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.402 TEST ( SubsetSubsetOperation , IncorrectInputMissingSeparatorComma )**

Definition at line 2115 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.403 TEST ( SubsetSubsetOperation , IncorrectInputMissingCommaAndArgument )

Definition at line 2119 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.404 TEST ( SubsetSubsetOperation , IncorrectInputMissingSecondArgument )

Definition at line 2123 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.405 TEST ( SubsetSubsetOperation , IncorrectInputBeforeEndParanthesis )

Definition at line 2127 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.406 TEST ( SubsetSubsetOperation , IncorrectInputAfterEndParanthesis )

Definition at line 2131 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.407 TEST ( SubsetSubsetOperation , Correct )

Definition at line 2135 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.408 TEST ( TemporalNumericCollection , IncorrectInputWrongAlternative )

Definition at line 2148 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.409 TEST ( TemporalNumericCollection , Correct )

Definition at line 2152 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.410 TEST ( TemporalNumericComparison , NumericMeasureFirst1 )

Definition at line 2165 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.411 TEST ( TemporalNumericComparison , NumericMeasureFirst2 )

Definition at line 2169 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.412 TEST ( TemporalNumericComparison , ComparatorFirst1 )**

Definition at line 2173 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.413 TEST ( TemporalNumericComparison , ComparatorFirst2 )**

Definition at line 2177 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.414 TEST ( TemporalNumericComparison , IncorrectOrder )**

Definition at line 2181 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.415 TEST ( TemporalNumericComparison , IncorrectInputMissingFirstOperand )**

Definition at line 2185 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.416 TEST ( TemporalNumericComparison , IncorrectInputMissingComparator )**

Definition at line 2189 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.417 TEST ( TemporalNumericComparison , IncorrectInputMissingSecondOperand )**

Definition at line 2193 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.418 TEST ( TemporalNumericComparison , IncorrectInputMissingBothOperands )**

Definition at line 2197 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.419 TEST ( TemporalNumericComparison , IncorrectInputMissingBothOperandsAndComparator )**

Definition at line 2201 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.420 TEST ( TemporalNumericComparison , IncorrectInputAfterNumericMeasure )**

Definition at line 2205 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.421 TEST ( TemporalNumericComparison , IncorrectInputAfterComparator )

Definition at line 2209 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.422 TEST ( TemporalNumericComparison , Correct )

Definition at line 2213 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.423 TEST ( TemporalNumericMeasure , WrongAlternative )

Definition at line 2226 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.424 TEST ( TemporalNumericMeasure , Correct )

Definition at line 2230 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.425 TEST ( TemporalNumericMeasureCollection , IncorrectInputBeforeBeginParanthesis )

Definition at line 2243 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.426 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingBeginParanthesis )

Definition at line 2247 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.427 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingBeginTimepoint )

Definition at line 2251 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.428 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingComma )

Definition at line 2255 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.429 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingEndTimepoint )

Definition at line 2259 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.430 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingEndTimepointAndComma )**

Definition at line 2263 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.431 TEST ( TemporalNumericMeasureCollection , IncorrectInputExtraTimepoint )**

Definition at line 2267 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.432 TEST ( TemporalNumericMeasureCollection , IncorrectInputMissingEndParanthesis )**

Definition at line 2271 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.433 TEST ( TemporalNumericMeasureCollection , IncorrectInputExtraSurroundingParantheses )**

Definition at line 2275 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.434 TEST ( TemporalNumericMeasureCollection , IncorrectInputBeforeNumericMeasure )**

Definition at line 2279 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.435 TEST ( TemporalNumericMeasureCollection , IncorrectInputAfterNumericMeasure )**

Definition at line 2283 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.436 TEST ( TemporalNumericMeasureCollection , Correct )**

Definition at line 2287 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.437 TEST ( TimeseriesComponent , IncorrectInputBeforeTimeseriesComponent )**

Definition at line 2300 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.438 TEST ( TimeseriesComponent , IncorrectParanthesisBeforeTimeseriesComponent )**

Definition at line 2304 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.439 TEST ( TimeseriesComponent , IncorrectInputAfterTimeseriesComponent )

Definition at line 2308 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.440 TEST ( TimeseriesComponent , IncorrectParanthesisAfterTimeseriesComponent )

Definition at line 2312 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.441 TEST ( TimeseriesComponent , IncorrectInputInvalidAlternative )

Definition at line 2316 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.442 TEST ( TimeseriesComponent , Correct )

Definition at line 2320 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.443 TEST ( TimeseriesMeasure , IncorrectInputWrongAlternative )

Definition at line 2333 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.444 TEST ( TimeseriesMeasure , CorrectEnteringTime )

Definition at line 2337 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.445 TEST ( TimeseriesMeasure , CorrectEnteringValue )

Definition at line 2341 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.446 TEST ( TimeseriesTimeseriesComponent , IncorrectInputBeforeTimeseriesMeasure )

Definition at line 2354 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.447 TEST ( TimeseriesTimeseriesComponent , IncorrectInputWrongTimeseriesMeasure )

Definition at line 2358 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.448 TEST ( TimeseriesTimeseriesComponent , IncorrectInputBeforeStartParanthesis )**

Definition at line 2362 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.449 TEST ( TimeseriesTimeseriesComponent , IncorrectInputAfterStartParanthesis )**

Definition at line 2366 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.450 TEST ( TimeseriesTimeseriesComponent , IncorrectInputInvalidFirstParameter )**

Definition at line 2370 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.451 TEST ( TimeseriesTimeseriesComponent , IncorrectInputMissingFirstParameter )**

Definition at line 2374 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.452 TEST ( TimeseriesTimeseriesComponent , IncorrectInputMissingCommaBetweenParameters )**

Definition at line 2378 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.453 TEST ( TimeseriesTimeseriesComponent , IncorrectInputMissingSecondParameter )**

Definition at line 2382 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.454 TEST ( TimeseriesTimeseriesComponent , IncorrectInputInvalidSecondParameter )**

Definition at line 2386 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.455 TEST ( TimeseriesTimeseriesComponent , IncorrectInputBeforeEndParanthesis )**

Definition at line 2390 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.456 TEST ( TimeseriesTimeseriesComponent , IncorrectInputAfterEndParanthesis )**

Definition at line 2394 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

**8.344.1.457 TEST ( TimeseriesTimeseriesComponent , IncorrectInputMissingParantheses )**

Definition at line 2398 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.458 TEST ( TimeseriesTimeseriesComponent , IncorrectInputInvertedParantheses )**

Definition at line 2402 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.459 TEST ( TimeseriesTimeseriesComponent , IncorrectInputDoubledParantheses )**

Definition at line 2406 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.460 TEST ( TimeseriesTimeseriesComponent , IncorrectInputWrongParanthesesType )**

Definition at line 2410 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.461 TEST ( TimeseriesTimeseriesComponent , Correct )**

Definition at line 2414 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.462 TEST ( UnaryNumericFilter , IncorrectInputMissingParameter )**

Definition at line 2427 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.463 TEST ( UnaryNumericFilter , IncorrectInputBeforeStartBracket )**

Definition at line 2431 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.464 TEST ( UnaryNumericFilter , IncorrectInputAfterStartBracket )**

Definition at line 2435 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.465 TEST ( UnaryNumericFilter , IncorrectInputBeforeEndBracket )**

Definition at line 2439 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.466 TEST ( UnaryNumericFilter , IncorrectInputAfterEndBracket )**

Definition at line 2443 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.467 TEST ( UnaryNumericFilter , Correct )**

Definition at line 2447 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.468 TEST ( UnaryNumericMeasure , IncorrectUnaryNumericMeasure )**

Definition at line 2460 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.469 TEST ( UnaryNumericMeasure , CorrectAbs )**

Definition at line 2464 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.470 TEST ( UnaryNumericMeasure , CorrectCeil )**

Definition at line 2468 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.471 TEST ( UnaryNumericMeasure , CorrectFloor )**

Definition at line 2472 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.472 TEST ( UnaryNumericMeasure , CorrectRound )**

Definition at line 2476 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.473 TEST ( UnaryNumericMeasure , CorrectSign )**

Definition at line 2480 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.474 TEST ( UnaryNumericMeasure , CorrectSqrt )**

Definition at line 2484 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.475 TEST ( UnaryNumericMeasure , CorrectTrunc )

Definition at line 2488 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.476 TEST ( UnaryNumericNumeric , IncorrectInputMissingParameter )

Definition at line 2501 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.477 TEST ( UnaryNumericNumeric , IncorrectInputBeforeStartBracket )

Definition at line 2505 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.478 TEST ( UnaryNumericNumeric , IncorrectInputAfterStartBracket )

Definition at line 2509 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.479 TEST ( UnaryNumericNumeric , IncorrectInputBeforeEndBracket )

Definition at line 2513 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.480 TEST ( UnaryNumericNumeric , IncorrectInputAfterEndBracket )

Definition at line 2517 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.481 TEST ( UnaryNumericNumeric , IncorrectInputDoubleBrackets )

Definition at line 2521 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.482 TEST ( UnaryNumericNumeric , IncorrectInputMissingComparator )

Definition at line 2525 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.483 TEST ( UnaryNumericNumeric , IncorrectInputMissingFirstOperand )

Definition at line 2529 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.484 TEST ( UnaryNumericNumeric , IncorrectInputMissingSecondOperand )**

Definition at line 2533 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.485 TEST ( UnaryNumericNumeric , IncorrectInputMissingBothOperands )**

Definition at line 2537 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.486 TEST ( UnaryNumericNumeric , Correct )**

Definition at line 2541 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.487 TEST ( UnaryNumericTemporal , IncorrectInputMissingParameter )**

Definition at line 2554 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.488 TEST ( UnaryNumericTemporal , IncorrectInputBeforeStartBracket )**

Definition at line 2558 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.489 TEST ( UnaryNumericTemporal , IncorrectInputAfterStartBracket )**

Definition at line 2562 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.490 TEST ( UnaryNumericTemporal , IncorrectInputBeforeEndBracket )**

Definition at line 2566 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.491 TEST ( UnaryNumericTemporal , IncorrectInputAfterEndBracket )**

Definition at line 2570 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.492 TEST ( UnaryNumericTemporal , IncorrectInputDoubleBrackets )**

Definition at line 2574 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.493 TEST ( UnaryNumericTemporal , Correct )

Definition at line 2578 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.494 TEST ( UnarySpatialConstraint , IncorrectSpatialMeasureBeforeConstraint )

Definition at line 2591 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.495 TEST ( UnarySpatialConstraint , IncorrectInputSpatialEntityInsteadOfSpatialMeasure )

Definition at line 2595 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.496 TEST ( UnarySpatialConstraint , IncorrectInputMissingSpatialMeasure )

Definition at line 2599 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.497 TEST ( UnarySpatialConstraint , IncorrectInputMissingComparator )

Definition at line 2603 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.498 TEST ( UnarySpatialConstraint , IncorrectInputMissingNumericMeasure )

Definition at line 2607 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.499 TEST ( UnarySpatialConstraint , IncorrectInputMissingComparatorNumericMeasure )

Definition at line 2611 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.500 TEST ( UnarySpatialConstraint , IncorrectInputMissingSpatialMeasureNumericMeasure )

Definition at line 2615 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.501 TEST ( UnarySpatialConstraint , IncorrectInputMissingSpatialMeasureComparator )

Definition at line 2619 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.502 TEST ( UnarySpatialConstraint , IncorrectInputEmptyConstraint )**

Definition at line 2623 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.503 TEST ( UnarySpatialConstraint , Correct )**

Definition at line 2627 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.504 TEST ( UnaryStatisticalMeasure , IncorrectUnaryStatisticalMeasure )**

Definition at line 2640 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.505 TEST ( UnaryStatisticalMeasure , CorrectAvg )**

Definition at line 2644 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.506 TEST ( UnaryStatisticalMeasure , CorrectCount )**

Definition at line 2648 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.507 TEST ( UnaryStatisticalMeasure , CorrectGeomean )**

Definition at line 2652 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.508 TEST ( UnaryStatisticalMeasure , CorrectHarmean )**

Definition at line 2656 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.509 TEST ( UnaryStatisticalMeasure , CorrectKurt )**

Definition at line 2660 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.510 TEST ( UnaryStatisticalMeasure , CorrectMax )**

Definition at line 2664 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

Reference

8.344.1.511 TEST ( UnaryStatisticalMeasure , CorrectMedian )

Definition at line 2668 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.512 TEST ( UnaryStatisticalMeasure , CorrectMin )

Definition at line 2672 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.513 TEST ( UnaryStatisticalMeasure , CorrectMode )

Definition at line 2676 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.514 TEST ( UnaryStatisticalMeasure , CorrectProduct )

Definition at line 2680 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.515 TEST ( UnaryStatisticalMeasure , CorrectSkew )

Definition at line 2684 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.516 TEST ( UnaryStatisticalMeasure , CorrectStdev )

Definition at line 2688 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.517 TEST ( UnaryStatisticalMeasure , CorrectSum )

Definition at line 2692 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.518 TEST ( UnaryStatisticalMeasure , CorrectVar )

Definition at line 2696 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.519 TEST ( UnaryStatisticalNumeric , IncorrectInputNoSubset )

Definition at line 2709 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.520 TEST ( UnaryStatisticalNumeric , IncorrectInputBeforeStartBracket )**

Definition at line 2713 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.521 TEST ( UnaryStatisticalNumeric , IncorrectInputAfterStartBracket )**

Definition at line 2717 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.522 TEST ( UnaryStatisticalNumeric , IncorrectInputBeforeEndBracket )**

Definition at line 2721 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.523 TEST ( UnaryStatisticalNumeric , IncorrectInputAfterEndBracket )**

Definition at line 2725 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.524 TEST ( UnaryStatisticalNumeric , IncorrectInputDoubleBrackets )**

Definition at line 2729 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.525 TEST ( UnaryStatisticalNumeric , Correct )**

Definition at line 2733 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.526 TEST ( UnaryStatisticalSpatial , IncorrectInputNoSubset )**

Definition at line 2746 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.527 TEST ( UnaryStatisticalSpatial , IncorrectInputBeforeStartBracket )**

Definition at line 2750 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.528 TEST ( UnaryStatisticalSpatial , IncorrectInputAfterStartBracket )**

Definition at line 2754 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.529 TEST ( UnaryStatisticalSpatial , IncorrectInputBeforeEndBracket )

Definition at line 2758 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.530 TEST ( UnaryStatisticalSpatial , IncorrectInputAfterEndBracket )

Definition at line 2762 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.531 TEST ( UnaryStatisticalSpatial , IncorrectInputDoubleBrackets )

Definition at line 2766 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.532 TEST ( UnaryStatisticalSpatial , Correct )

Definition at line 2770 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.533 TEST ( UnaryTypeConstraint , IncorrectInputWrongTypeKeywordExtraLetterAfter )

Definition at line 2783 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.534 TEST ( UnaryTypeConstraint , IncorrectInputWrongTypeKeywordExtraLetterBefore )

Definition at line 2787 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.535 TEST ( UnaryTypeConstraint , IncorrectInputBeforeTypeKeyword )

Definition at line 2791 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.536 TEST ( UnaryTypeConstraint , IncorrectInputAfterTypeKeyword )

Definition at line 2795 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.537 TEST ( UnaryTypeConstraint , IncorrectInputAfterComparator )

Definition at line 2799 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.538 TEST ( UnaryTypeConstraint , IncorrectInputAfterFilterNumericMeasure )**

Definition at line 2803 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.539 TEST ( UnaryTypeConstraint , Correct )**

Definition at line 2807 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.540 TEST ( UntilLogicProperty , IncorrectInputMissingStartTimepoint )**

Definition at line 2820 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.541 TEST ( UntilLogicProperty , IncorrectInputMissingEndTimepoint )**

Definition at line 2824 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.542 TEST ( UntilLogicProperty , IncorrectInputMissingTimepoints )**

Definition at line 2828 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.543 TEST ( UntilLogicProperty , IncorrectInputMissingTimepointsAndBrackets )**

Definition at line 2832 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.544 TEST ( UntilLogicProperty , UntilOperatorAsUnaryBefore )**

Definition at line 2836 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.545 TEST ( UntilLogicProperty , UntilOperatorAsUnaryAfter )**

Definition at line 2840 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.546 TEST ( UntilLogicProperty , IncorrectInputBeforeUntilOperator )**

Definition at line 2844 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

---

8.344.1.547 TEST ( UntilLogicProperty , AdditionalOperatorBeforeUntilOperator )

Definition at line 2848 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.548 TEST ( UntilLogicProperty , IncorrectInputAfterUntilOperator )

Definition at line 2852 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.549 TEST ( UntilLogicProperty , AdditionalOperatorAfterUntilOperator )

Definition at line 2856 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.550 TEST ( UntilLogicProperty , WrongInputBeforeStartParenthesis )

Definition at line 2860 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.551 TEST ( UntilLogicProperty , WrongInputAfterStartParenthesis )

Definition at line 2864 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.552 TEST ( UntilLogicProperty , MissingTimepointsComma )

Definition at line 2868 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.553 TEST ( UntilLogicProperty , StartTimepointInvalid )

Definition at line 2872 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.554 TEST ( UntilLogicProperty , StartTimepointRealNumber )

Definition at line 2876 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.555 TEST ( UntilLogicProperty , EndTimepointInvalid )

Definition at line 2880 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.556 TEST ( UntilLogicProperty , EndTimepointRealNumber )**

Definition at line 2884 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.557 TEST ( UntilLogicProperty , TimepointsInvalid )**

Definition at line 2888 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.558 TEST ( UntilLogicProperty , TimepointsRealNumber )**

Definition at line 2892 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.559 TEST ( UntilLogicProperty , WrongInputBeforeEndParenthesis )**

Definition at line 2896 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.560 TEST ( UntilLogicProperty , WrongInputAfterEndParenthesis )**

Definition at line 2900 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.561 TEST ( UntilLogicProperty , Correct )**

Definition at line 2904 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.562 TEST ( UntilLogicProperty , MultipleCorrect )**

Definition at line 2908 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.563 TEST ( Input , IncorrectTrueInput )**

Definition at line 2921 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

**8.344.1.564 TEST ( Input , IncorrectTInput )**

Definition at line 2925 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.565 TEST ( Input , IncorrectFalseInput )

Definition at line 2929 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.344.1.566 TEST ( Input , IncorrectFlInput )

Definition at line 2933 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

## 8.344.2 Variable Documentation

8.344.2.1 const std::vector<std::string> CONSTRAINTS\_BINARY\_OPERATORS = std::vector<std::string>({"^", "V", "=>", "<=>"}); [static]

Definition at line 678 of file ParserTest.hpp.

Referenced by TEST\_F().

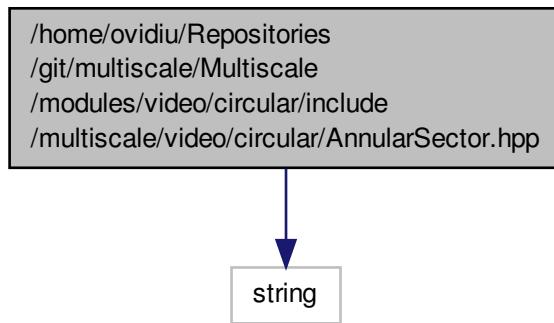
8.344.2.2 const std::vector<std::string> LOGIC\_PROPERTIES\_BINARY\_OPERATORS = std::vector<std::string>({"^", "V", "=>", "<=>"}); [static]

Definition at line 764 of file ParserTest.hpp.

Referenced by TEST\_F().

## 8.345 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/vic AnnularSector.hpp File Reference

#include <string>  
Include dependency graph for AnnularSector.hpp:



## Classes

- class [multiscale::video::AnnularSector](#)

*An annular sector is the basic element in the considered circular geometry.*

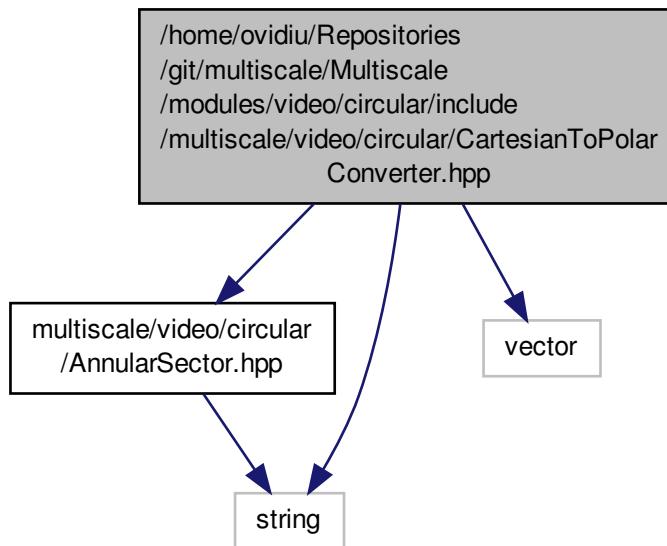
## Namespaces

- [multiscale](#)
- [multiscale::video](#)

## 8.346 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/CartesianToPolarConverter.hpp File Reference

```
#include "multiscale/video/circular/AnnularSector.hpp"
#include <string>
#include <vector>
```

Include dependency graph for CartesianToPolarConverter.hpp:



## Classes

- class [multiscale::video::CartesianToPolarConverter](#)

*Converter from the rectangular geometry grid cells to annular sectors.*

## Namespaces

- [multiscale](#)
- [multiscale::video](#)

8.347 /home/ovidiu.Repositories/git/multiscale/

Multiscale/modules/video/circular/include/multiscale/video/circular/PolarCsvToInputFilesConverter.hpp

File Reference

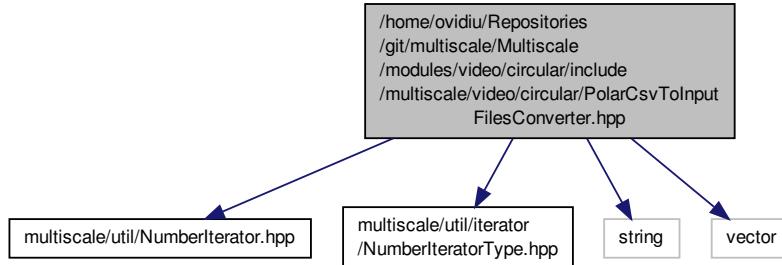
8.347 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/vic

1503

## PolarCsvToInputFilesConverter.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include <string>
#include <vector>
```

Include dependency graph for PolarCsvToInputFilesConverter.hpp:



## Classes

- class [multiscale::video::PolarCsvToInputFilesConverter](#)

*Csv file to input file converter considering polar coordinates.*

## Namespaces

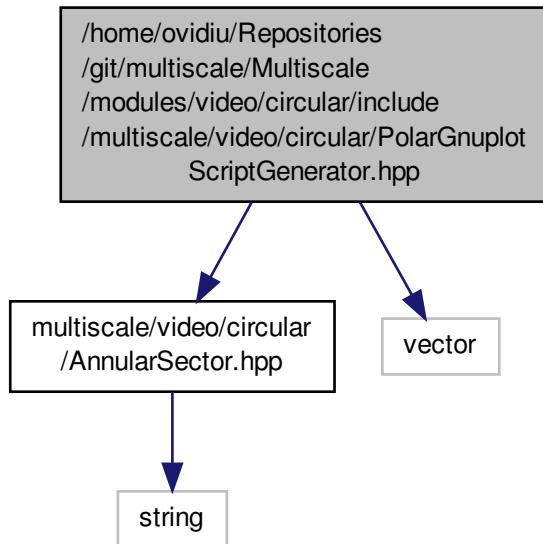
- [multiscale](#)
- [multiscale::video](#)

8.348 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/vic

## PolarGnuplotScriptGenerator.hpp File Reference

```
#include "multiscale/video/circular/AnnularSector.hpp"
#include <vector>
```

Include dependency graph for PolarGnuplotScriptGenerator.hpp:



## Classes

- class [multiscale::video::PolarGnuplotScriptGenerator](#)

*Gnuplot script generator from the provided annular sectors.*

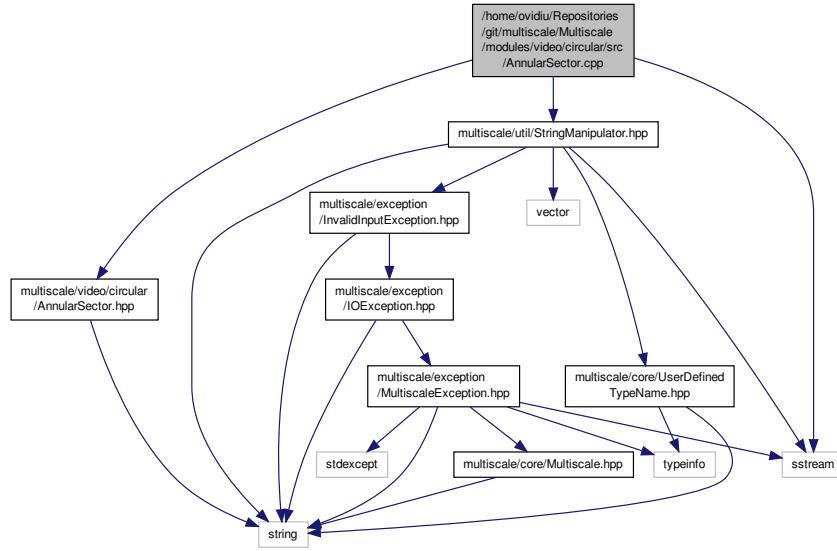
## Namespaces

- [multiscale](#)
- [multiscale::video](#)

## 8.349 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/← AnnularSector.cpp File Reference

```
#include "multiscale/video/circular/AnnularSector.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <sstream>
```

Include dependency graph for AnnularSector.cpp:



## Variables

- const std::string **SEPARATOR** = " "

### 8.349.1 Variable Documentation

#### 8.349.1.1 const std::string SEPARATOR = " "

Definition at line 6 of file `AnnularSector.cpp`.

Referenced by `multiscale::video::AnnularSector::toString()`.

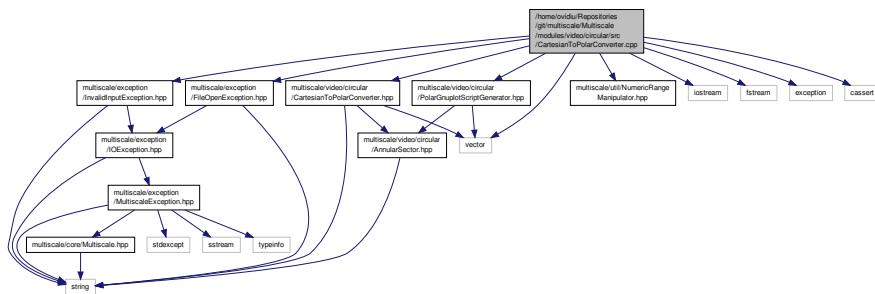
## 8.350 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/CartesianToPolarConverter.cpp File Reference

```

#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/video/circular/CartesianToPolarConverter.hpp"
#include "multiscale/video/circular/PolarGnuplotScriptGenerator.hpp"
#include "multiscale/util/NumericRangeManipulator.hpp"
#include <iostream>
#include <fstream>
#include <exception>
#include <cassert>
#include <vector>

```

Include dependency graph for `CartesianToPolarConverter.cpp`:



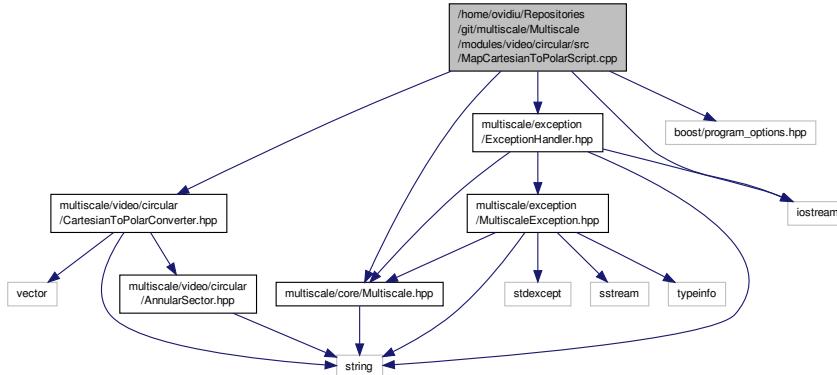
## 8.351 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/MapCartesianToPolarScript.cpp File Reference

```

#include "multiscale/core/Multiscale.hpp"
#include "multiscale/video/circular/CartesianToPolarConverter.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/program_options.hpp>
#include <iostream>

```

Include dependency graph for `MapCartesianToPolarScript.cpp`:



## Functions

- `po::variables_map initArgumentsConfig (po::options_description &usageDescription, int argc, char **argv)`
- `void printHelpInformation (const po::variables_map &vm, const po::options_description &usageDescription)`
- `void printWrongParameters ()`
- `bool isValidOutputType (const po::variables_map &vm, bool &isScript)`
- `bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, bool &isScript, int argc, char **argv)`
- `int main (int argc, char **argv)`

### 8.351.1 Function Documentation

## 8.352 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarCsvToInputFilesConverter.cpp File

### Reference

1507

8.351.1.1 bool areValidParameters ( std::string & *inputFilepath*, std::string & *outputfilename*, bool & *isScript*, int *argc*, char \*\* *argv* )

Definition at line 99 of file MapCartesianToPolarScript.cpp.

References initArgumentsConfig(), isValidOutputType(), and printHelpInformation().

Referenced by main().

8.351.1.2 po::variables\_map initArgumentsConfig ( po::options\_description & *usageDescription*, int *argc*, char \*\* *argv* )

Definition at line 51 of file MapCartesianToPolarScript.cpp.

Referenced by areValidParameters().

8.351.1.3 bool isValidOutputType ( const po::variables\_map & *vm*, bool & *isScript* )

Definition at line 78 of file MapCartesianToPolarScript.cpp.

References multiscale::ERR\_MSG.

Referenced by areValidParameters().

8.351.1.4 int main ( int *argc*, char \*\* *argv* )

Definition at line 125 of file MapCartesianToPolarScript.cpp.

References areValidParameters(), multiscale::video::CartesianToPolarConverter::convert(), multiscale::EXEC\_→  
ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(),  
and printWrongParameters().

8.351.1.5 void printHelpInformation ( const po::variables\_map & *vm*, const po::options\_description & *usageDescription* )

Definition at line 67 of file MapCartesianToPolarScript.cpp.

Referenced by areValidParameters().

8.351.1.6 void printWrongParameters ( )

Definition at line 72 of file MapCartesianToPolarScript.cpp.

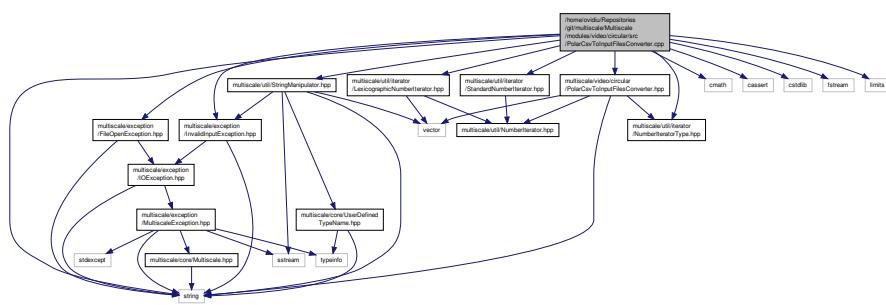
References multiscale::ERR\_MSG.

Referenced by main().

## 8.352 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/PolarCsvToInputFilesConverter.cpp File Reference

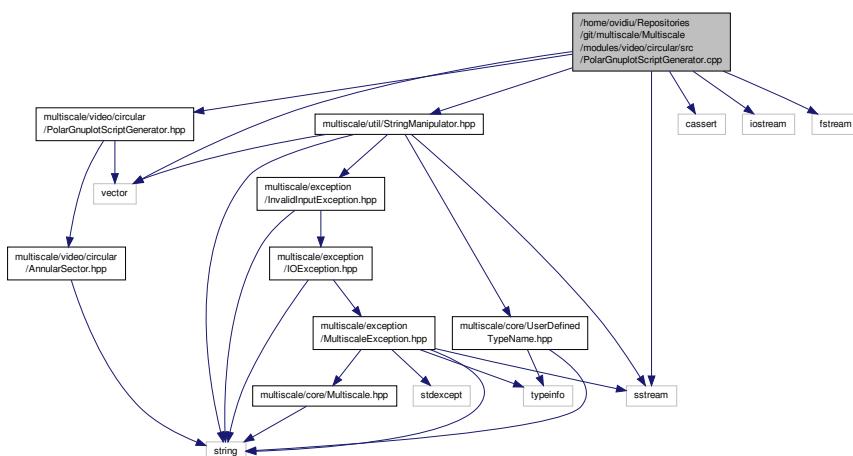
```
#include "multiscale/exception/FileOpenException.hpp"
```

```
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/video/circular/PolarCsvToInputFilesConverter.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/util/iterator/LexicographicNumberIterator.hpp"
#include "multiscale/util/iterator/StandardNumberIterator.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <cmath>
#include <cassert>
#include <cstdlib>
#include <fstream>
#include <limits>
#include <string>
Include dependency graph for PolarCsvToInputFilesConverter.cpp:
```



### 8.353 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/← PolarGnuplotScriptGenerator.cpp File Reference

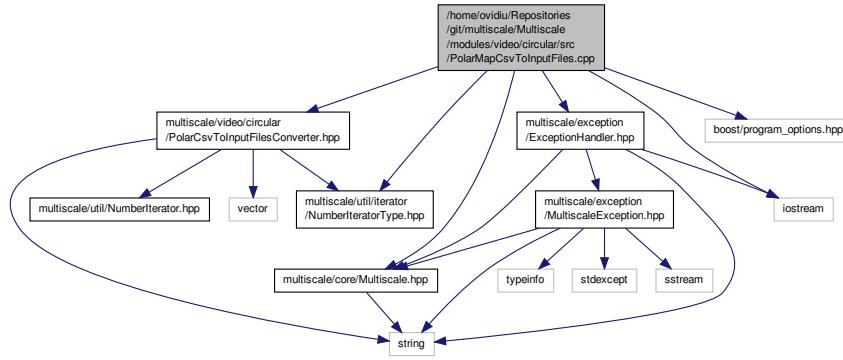
```
#include "multiscale/video/circular/PolarGnuplotScriptGenerator.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <cassert>
#include <iostream>
#include <vector>
#include <sstream>
#include <fstream>
Include dependency graph for PolarGnuplotScriptGenerator.cpp:
```



## PolarMapCsvToInputFiles.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/video/circular/PolarCsvToInputFilesConverter.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for PolarMapCsvToInputFiles.cpp:



## Functions

- po::variables\_map initArgumentsConfig (po::options\_description &usageDescription, int argc, char \*\*argv)
- void printHelpInformation (const po::variables\_map &vm, const po::options\_description &usageDescription)
- void printWrongParameters ()
- void setNumberIteratorType (const po::variables\_map &vm, NumberIteratorType &numberIteratorType)
- void setSelectedConcentrationIndex (const po::variables\_map &vm, unsigned int &selectedConcentrationIndex)
- void setLogScaling (const po::variables\_map &vm, bool &useLogScaling)
- bool isValidNrOfConcentrationsForPosition (const po::variables\_map &vm, unsigned int &nrOfConcentrationsForPosition)
- bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, unsigned int &nrOfConcentricCircles, unsigned int &nrOfSectors, unsigned int &nrOfConcentrationsForPosition, unsigned int &selectedConcentrationIndex, bool &useLogScaling, NumberIteratorType &numberIteratorType, int argc, char \*\*argv)
- int main (int argc, char \*\*argv)

### 8.354.1 Function Documentation

8.354.1.1 bool areValidParameters ( std::string & inputFilepath, std::string & outputFilename, unsigned int & nrOfConcentricCircles, unsigned int & nrOfSectors, unsigned int & nrOfConcentrationsForPosition, unsigned int & selectedConcentrationIndex, bool & useLogScaling, NumberIteratorType & numberIteratorType, int argc, char \*\* argv )

Definition at line 90 of file PolarMapCsvToInputFiles.cpp.

References initArgumentsConfig(), isValidNrOfConcentrationsForPosition(), printHelpInformation(), setLogScaling(), setNumberIteratorType(), and setSelectedConcentrationIndex().

Referenced by main().

8.354.1.2 `po::variables_map initArgumentsConfig ( po::options_description & usageDescription, int argc, char ** argv )`

Definition at line 31 of file PolarMapCsvToInputFiles.cpp.

Referenced by `isValidParameters()`.

8.354.1.3 `bool isValidNrOfConcentrationsForPosition ( const po::variables_map & vm, unsigned int & nrOfConcentrationsForPosition )`

Definition at line 75 of file PolarMapCsvToInputFiles.cpp.

References `multiscale::ERR_MSG`.

Referenced by `isValidParameters()`.

8.354.1.4 `int main ( int argc, char ** argv )`

Definition at line 137 of file PolarMapCsvToInputFiles.cpp.

References `isValidParameters()`, `multiscale::video::PolarCsvToInputFilesConverter::convert()`, `multiscale::EXEC_<_ERR_CODE`, `multiscale::EXEC_SUCCESS_CODE`, `multiscale::ExceptionHandler::printDetailedErrorMessage()`, `printWrongParameters()`, and `multiscale::STANDARD`.

8.354.1.5 `void printHelpInformation ( const po::variables_map & vm, const po::options_description & usageDescription )`

Definition at line 49 of file PolarMapCsvToInputFiles.cpp.

Referenced by `isValidParameters()`.

8.354.1.6 `void printWrongParameters ( )`

Definition at line 54 of file PolarMapCsvToInputFiles.cpp.

References `multiscale::ERR_MSG`.

Referenced by `main()`.

8.354.1.7 `void setLogScaling ( const po::variables_map & vm, bool & useLogScaling )`

Definition at line 70 of file PolarMapCsvToInputFiles.cpp.

Referenced by `isValidParameters()`.

8.354.1.8 `void setNumberIteratorType ( const po::variables_map & vm, NumberIteratorType & numberIteratorType )`

Definition at line 60 of file PolarMapCsvToInputFiles.cpp.

References `multiscale::LEXICOGRAPHIC`.

Referenced by `isValidParameters()`.

8.354.1.9 `void setSelectedConcentrationIndex ( const po::variables_map & vm, unsigned int & selectedConcentrationIndex )`

Definition at line 65 of file PolarMapCsvToInputFiles.cpp.

Referenced by `isValidParameters()`.

8.355 /home/ovidiu.Repositories/git/multiscale/

Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/CartesianToConcentrations

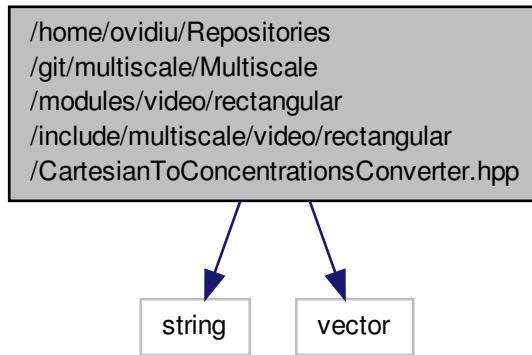
Converter.hpp File

Reference

## CartesianToConcentrationsConverter.hpp File Reference

```
#include <string>
#include <vector>
```

Include dependency graph for CartesianToConcentrationsConverter.hpp:



## Classes

- class [multiscale::video::CartesianToConcentrationsConverter](#)

*Scale the values of the rectangular geometry grid cells.*

## Namespaces

- [multiscale](#)
- [multiscale::video](#)

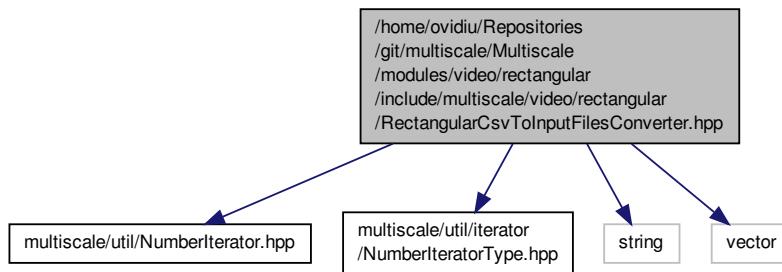
8.356

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale

## RectangularCsvToInputFilesConverter.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include <string>
#include <vector>
```

Include dependency graph for `RectangularCsvToInputFilesConverter.hpp`:



## Classes

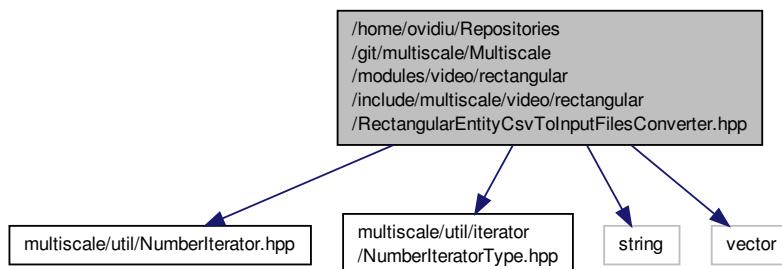
- class [multiscale::video::RectangularCsvToInputFilesConverter](#)  
*Csv file to input file converter considering cartesian coordinates.*

## Namespaces

- [multiscale](#)
- [multiscale::video](#)

## 8.357 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale RectangularEntityCsvToInputFilesConverter.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include <string>
#include <vector>
Include dependency graph for RectangularEntityCsvToInputFilesConverter.hpp:
```



## Classes

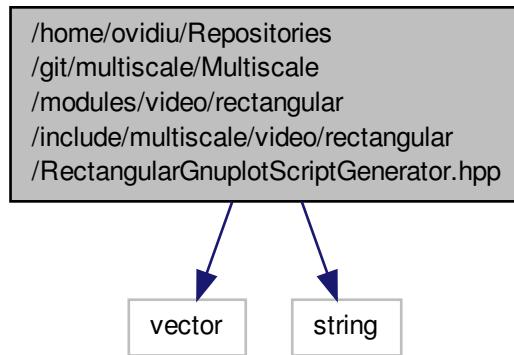
- class [multiscale::video::RectangularEntityCsvToInputFilesConverter](#)

## Namespaces

- multiscale
- multiscale::video

## 8.358 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/RectangularGnuplotScriptGenerator.hpp File Reference

```
#include <vector>
#include <string>
Include dependency graph for RectangularGnuplotScriptGenerator.hpp:
```



## Classes

- class [multiscale::video::RectangularGnuplotScriptGenerator](#)  
*Gnuplot script generator from the provided concentrations considering a rectangular geometry.*

## Namespaces

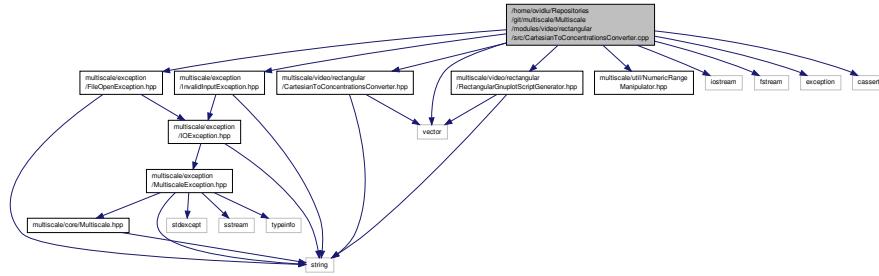
- multiscale
- multiscale::video

## 8.359 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/ ↵ CartesianToConcentrationsConverter.cpp File Reference

```
#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/video/rectangular/CartesianToConcentrationsConverter.hpp"
#include "multiscale/video/rectangular/RectangularGnuplotScriptGenerator.hpp"
```

```
hpp"
#include "multiscale/util/NumericRangeManipulator.hpp"
#include <iostream>
#include <fstream>
#include <exception>
#include <cassert>
#include <vector>
```

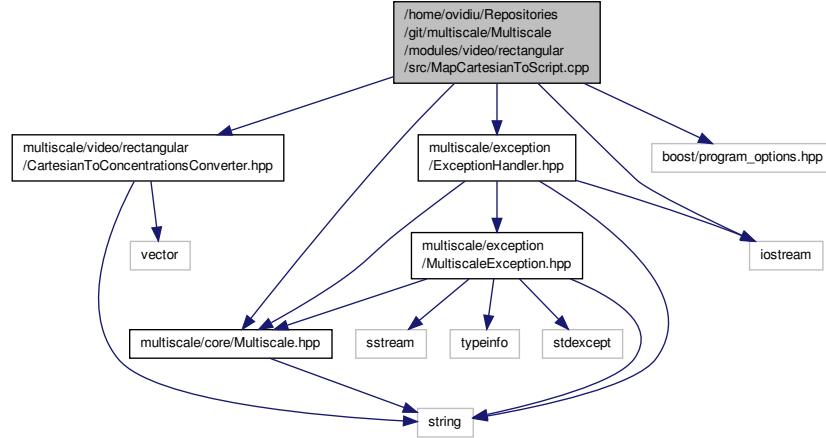
Include dependency graph for `CartesianToConcentrationsConverter.cpp`:



## 8.360 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/MapCartesianToScript.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/video/rectangular/CartesianToConcentrationsConverter.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for `MapCartesianToScript.cpp`:



## Functions

- `po::variables_map initArgumentsConfig (po::options_description &usageDescription, int argc, char **argv)`
- `void printHelpInformation (const po::variables_map &vm, const po::options_description &usageDescription)`

Reference

- void [printWrongParameters\(\)](#)
- bool [isValidParameters\(std::string &inputfilepath, std::string &outputfilename, int argc, char \\*\\*argv\)](#)
- int [main\(int argc, char \\*\\*argv\)](#)

### 8.360.1 Function Documentation

8.360.1.1 bool [isValidParameters\( std::string & inputfilepath, std::string & outputfilename, int argc, char \\*\\* argv \)](#)

Definition at line 59 of file MapCartesianToScript.cpp.

References [initArgumentsConfig\(\)](#), and [printHelpInformation\(\)](#).

Referenced by [main\(\)](#).

8.360.1.2 po::variables\_map [initArgumentsConfig\( po::options\\_description & usageDescription, int argc, char \\*\\* argv \)](#)

Definition at line 36 of file MapCartesianToScript.cpp.

Referenced by [isValidParameters\(\)](#).

8.360.1.3 int [main\( int argc, char \\*\\* argv \)](#)

Definition at line 83 of file MapCartesianToScript.cpp.

References [isValidParameters\(\)](#), [multiscale::video::CartesianToConcentrationsConverter::convert\(\)](#), [multiscale::EXEC\\_ERR\\_CODE](#), [multiscale::EXEC\\_SUCCESS\\_CODE](#), [multiscale::ExceptionHandler::printDetailedErrorMessage\(\)](#), and [printWrongParameters\(\)](#).

8.360.1.4 void [printHelpInformation\( const po::variables\\_map & vm, const po::options\\_description & usageDescription \)](#)

Definition at line 48 of file MapCartesianToScript.cpp.

Referenced by [isValidParameters\(\)](#).

8.360.1.5 void [printWrongParameters\( \)](#)

Definition at line 53 of file MapCartesianToScript.cpp.

References [multiscale::ERR\\_MSG](#).

Referenced by [main\(\)](#).

## 8.361 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularCsvToInputFilesConverter.cpp File Reference

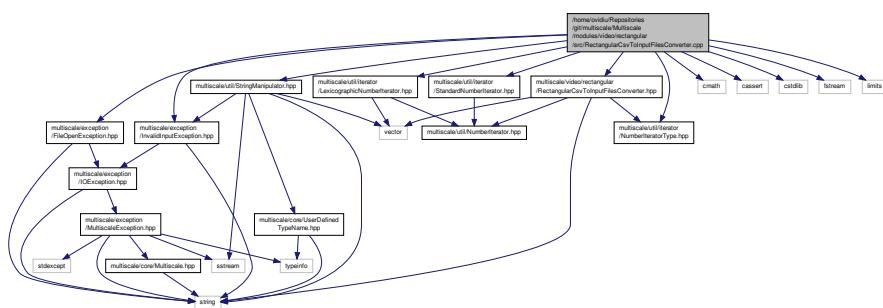
```
#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/video/rectangular/RectangularCsvToInputFilesConverter.h"
```

```

.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/util/iterator/LexicographicNumberIterator.hpp"
#include "multiscale/util/iterator/StandardNumberIterator.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <cmath>
#include <cassert>
#include <cstdlib>
#include <fstream>
#include <limits>

```

Include dependency graph for RectangularCsvToInputFilesConverter.cpp:



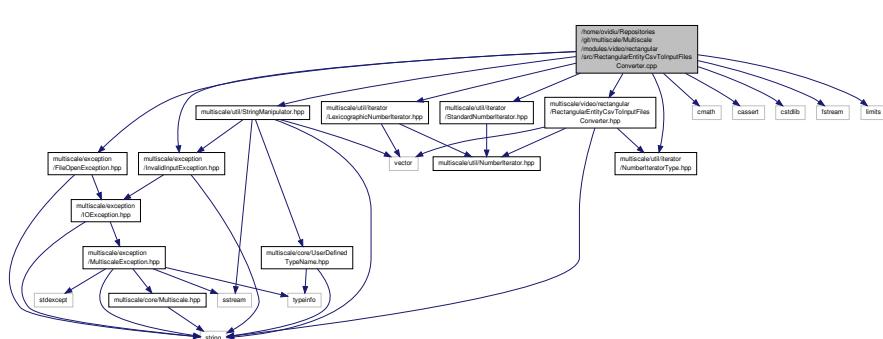
## 8.362 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularEntityCsvToInputFilesConverter.cpp File Reference

```

#include "multiscale/exception/FileOpenException.hpp"
#include "multiscale/exception/InvalidInputException.hpp"
#include "multiscale/video/rectangular/RectangularEntityCsvToInputFilesConverter.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/util/iterator/LexicographicNumberIterator.hpp"
#include "multiscale/util/iterator/StandardNumberIterator.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <cmath>
#include <cassert>
#include <cstdlib>
#include <fstream>
#include <limits>

```

Include dependency graph for RectangularEntityCsvToInputFilesConverter.cpp:



8.363 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularGnuplotScriptGenerator.cpp File

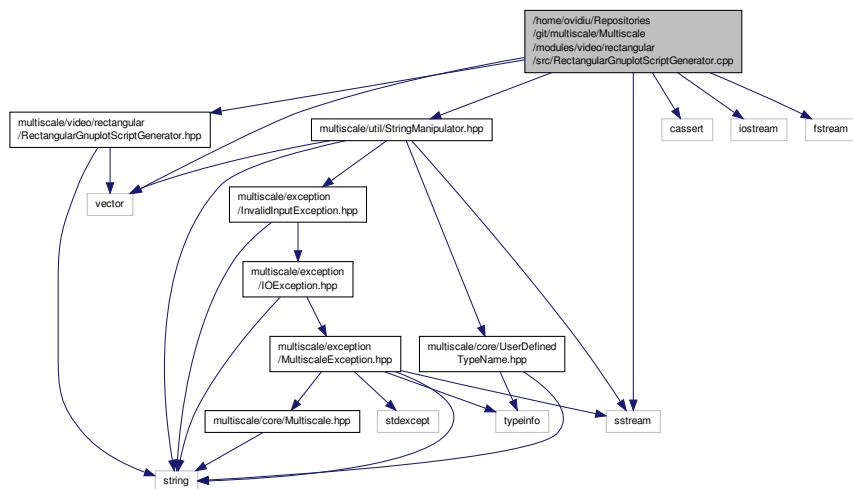
Reference

1517

8.363 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularGnuplotScriptGenerator.cpp File Reference

```
#include "multiscale/video/rectangular/RectangularGnuplotScriptGenerator.hpp"
#include "multiscale/util/StringManipulator.hpp"
#include <cassert>
#include <iostream>
#include <vector>
#include <sstream>
#include <fstream>
```

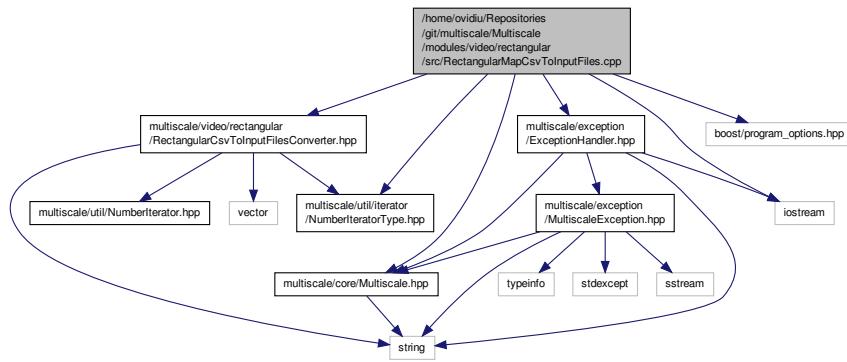
Include dependency graph for RectangularGnuplotScriptGenerator.cpp:



8.364 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularMapCsvToInputFiles.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/video/rectangular/RectangularCsvToInputFilesConverter.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for RectangularMapCsvToInputFiles.cpp:



## Functions

- `po::variables_map initArgumentsConfig (po::options_description &usageDescription, int argc, char **argv)`
- `void printHelpInformation (const po::variables_map &vm, const po::options_description &usageDescription)`
- `void printWrongParameters ()`
- `void setNumberIteratorType (const po::variables_map &vm, NumberIteratorType &numberIteratorType)`
- `void setSelectedConcentrationIndex (const po::variables_map &vm, unsigned int &selectedConcentrationIndex)`
- `void setLogScaling (const po::variables_map &vm, bool &useLogScaling)`
- `bool isValidNrOfConcentrationsForPosition (const po::variables_map &vm, unsigned int &nrOfConcentrationsForPosition)`
- `bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, unsigned int &height, unsigned int &width, unsigned int &nrOfConcentrationsForPosition, unsigned int &selectedConcentrationIndex, bool &useLogScaling, NumberIteratorType &numberIteratorType, int argc, char **argv)`
- `int main (int argc, char **argv)`

### 8.364.1 Function Documentation

**8.364.1.1** `bool areValidParameters ( std::string & inputfilepath, std::string & outputfilename, unsigned int & height, unsigned int & width, unsigned int & nrOfConcentrationsForPosition, unsigned int & selectedConcentrationIndex, bool & useLogScaling, NumberIteratorType & numberIteratorType, int argc, char ** argv )`

Definition at line 90 of file RectangularMapCsvToInputFiles.cpp.

References `initArgumentsConfig()`, `isValidNrOfConcentrationsForPosition()`, `printHelpInformation()`, `setLogScaling()`, `setNumberIteratorType()`, and `setSelectedConcentrationIndex()`.

Referenced by `main()`.

**8.364.1.2** `po::variables_map initArgumentsConfig ( po::options_description & usageDescription, int argc, char ** argv )`

Definition at line 31 of file RectangularMapCsvToInputFiles.cpp.

Referenced by `areValidParameters()`.

**8.364.1.3** `bool isValidNrOfConcentrationsForPosition ( const po::variables_map & vm, unsigned int & nrOfConcentrationsForPosition )`

Definition at line 75 of file RectangularMapCsvToInputFiles.cpp.

Reference

References multiscale::ERR\_MSG.

Referenced by areValidParameters().

8.364.1.4 int main ( int argc, char \*\* argv )

Definition at line 137 of file RectangularMapCsvToInputFiles.cpp.

References areValidParameters(), multiscale::video::RectangularCsvToInputFilesConverter::convert(), multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(), printWrongParameters(), and multiscale::STANDARD.

8.364.1.5 void printHelpInformation ( const po::variables\_map & vm, const po::options\_description & usageDescription )

Definition at line 49 of file RectangularMapCsvToInputFiles.cpp.

Referenced by areValidParameters().

8.364.1.6 void printWrongParameters ( )

Definition at line 54 of file RectangularMapCsvToInputFiles.cpp.

References multiscale::ERR\_MSG.

Referenced by main().

8.364.1.7 void setLogScaling ( const po::variables\_map & vm, bool & useLogScaling )

Definition at line 70 of file RectangularMapCsvToInputFiles.cpp.

Referenced by areValidParameters().

8.364.1.8 void setNumberIteratorType ( const po::variables\_map & vm, NumberIteratorType & numberIteratorType )

Definition at line 60 of file RectangularMapCsvToInputFiles.cpp.

References multiscale::LEXICOGRAPHIC.

Referenced by areValidParameters().

8.364.1.9 void setSelectedConcentrationIndex ( const po::variables\_map & vm, unsigned int & selectedConcentrationIndex )

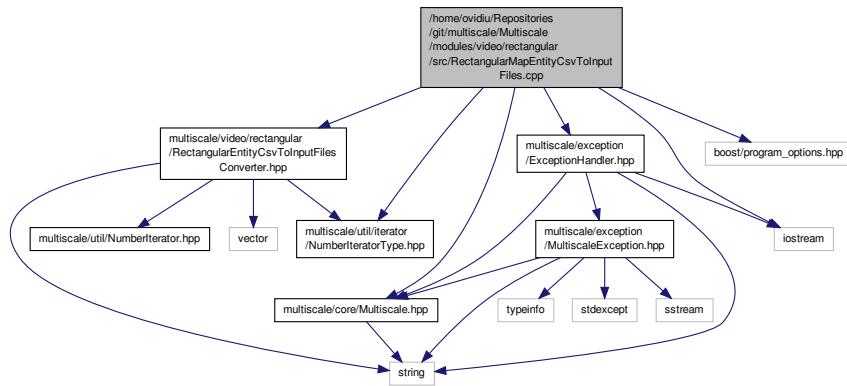
Definition at line 65 of file RectangularMapCsvToInputFiles.cpp.

Referenced by areValidParameters().

**8.365 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/RectangularMapEntityCsvToInputFiles.cpp File Reference**

```
#include "multiscale/core/Multiscale.hpp"
#include "multiscale/video/rectangular/RectangularEntityCsvToInputFilesConverter.hpp"
#include "multiscale/util/iterator/NumberIteratorType.hpp"
#include "multiscale/exception/ExceptionHandler.hpp"
#include <boost/program_options.hpp>
#include <iostream>
```

Include dependency graph for RectangularMapEntityCsvToInputFiles.cpp:



## Functions

- po::variables\_map initArgumentsConfig (po::options\_description &usageDescription, int argc, char \*\*argv)
- void printHelpInformation (const po::variables\_map &vm, const po::options\_description &usageDescription)
- void printWrongParameters ()
- void setNumberIteratorType (const po::variables\_map &vm, NumberIteratorType &numberIteratorType)
- bool areValidParameters (std::string &inputFilepath, std::string &outputFilename, unsigned int &height, unsigned int &width, unsigned int &nrOfEntities, unsigned int &maxPileup, NumberIteratorType &numberIteratorType, int argc, char \*\*argv)
- int main (int argc, char \*\*argv)

### 8.365.1 Function Documentation

**8.365.1.1 bool areValidParameters ( std::string & *inputFilepath*, std::string & *outputFilename*, unsigned int & *height*, unsigned int & *width*, unsigned int & *nrOfEntities*, unsigned int & *maxPileup*, NumberIteratorType & *numberIteratorType*, int *argc*, char \*\* *argv* )**

Definition at line 64 of file RectangularMapEntityCsvToInputFiles.cpp.

References initArgumentsConfig(), printHelpInformation(), and setNumberIteratorType().

Referenced by main().

**8.365.1.2 po::variables\_map initArgumentsConfig ( po::options\_description & *usageDescription*, int *argc*, char \*\* *argv* )**

Definition at line 31 of file RectangularMapEntityCsvToInputFiles.cpp.

Referenced by areValidParameters().

**8.365.1.3 int main ( int *argc*, char \*\* *argv* )**

Definition at line 102 of file RectangularMapEntityCsvToInputFiles.cpp.

References areValidParameters(), multiscale::video::RectangularEntityCsvToInputFilesConverter::convert(), multiscale::EXEC\_ERR\_CODE, multiscale::EXEC\_SUCCESS\_CODE, multiscale::ExceptionHandler::printDetailedErrorMessage(), printWrongParameters(), and multiscale::STANDARD.

Reference

8.365.1.4 void printHelpInformation ( const po::variables\_map & vm, const po::options\_description & usageDescription )

Definition at line 48 of file RectangularMapEntityCsvToInputFiles.cpp.

Referenced by areValidParameters().

8.365.1.5 void printWrongParameters ( )

Definition at line 53 of file RectangularMapEntityCsvToInputFiles.cpp.

References multiscale::ERR\_MSG.

Referenced by main().

8.365.1.6 void setNumberIteratorType ( const po::variables\_map & vm, NumberIteratorType & numberIteratorType )

Definition at line 59 of file RectangularMapEntityCsvToInputFiles.cpp.

References multiscale::LEXICOGRAPHIC.

Referenced by areValidParameters().