

Mule

1.0.101

Generated by Doxygen 1.7.6.1

Thu Jul 31 2014 13:50:02

Contents

1 Multiscale	1
1.1 Brief description	1
1.2 Contact	1
2 Namespace Index	3
2.1 Namespace List	3
3 Class 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	37
6.1 multiscale Namespace Reference	37
6.1.1 Enumeration Type Documentation	39
6.1.1.1 ColourCode	39
6.1.1.2 NumberIteratorType	39
6.1.1.3 UnixColourCode	40
6.1.1.4 WindowsColourCode	40
6.1.2 Variable Documentation	41
6.1.2.1 ERR_MSG	41
6.1.2.2 ERR_UNDEFINED_ENUM_VALUE	41
6.1.2.3 EXEC_ERR_CODE	41

6.1.2.4	EXEC_SUCCESS_CODE	41
6.2	multiscale::analysis Namespace Reference	41
6.2.1	Typedef Documentation	42
6.2.1.1	Polygon	42
6.2.2	Enumeration Type Documentation	43
6.2.2.1	Shape2D	43
6.2.2.2	SpatialEntityPseudo3DType	43
6.3	multiscale::verification Namespace Reference	43
6.3.1	Typedef Documentation	52
6.3.1.1	ConstraintAttributeType	52
6.3.1.2	FilterNumericMeasureAttributeType	52
6.3.1.3	LogicPropertyAttributeType	52
6.3.1.4	NumericMeasureAttributeType	53
6.3.1.5	NumericSpatialAttributeType	53
6.3.1.6	PrimaryConstraintAttributeType	53
6.3.1.7	PrimaryLogicPropertyAttributeType	53
6.3.1.8	PrimaryNumericMeasureAttributeType	54
6.3.1.9	SubsetAttributeType	54
6.3.2	Enumeration Type Documentation	54
6.3.2.1	ApproximateBayesianModelCheckingResult	54
6.3.2.2	BayesianModelCheckingResult	54
6.3.2.3	BinaryNumericMeasureType	55
6.3.2.4	BinarySubsetMeasureType	55
6.3.2.5	ComparatorType	55
6.3.2.6	QuaternarySubsetMeasureType	56
6.3.2.7	SetOperationType	56
6.3.2.8	SpatialMeasureType	56
6.3.2.9	StatisticalModelCheckingResult	57
6.3.2.10	SubsetOperationType	57
6.3.2.11	SubsetSpecificType	58
6.3.2.12	TernarySubsetMeasureType	58
6.3.2.13	UnaryNumericMeasureType	58
6.3.2.14	UnarySubsetMeasureType	58
6.3.3	Function Documentation	59

6.3.3.1	operator<<	59
6.3.3.2	operator<<	59
6.3.3.3	operator<<	59
6.3.3.4	operator<<	60
6.3.3.5	operator<<	60
6.3.3.6	operator<<	60
6.3.3.7	operator<<	61
6.3.3.8	operator<<	61
6.3.3.9	operator<<	61
6.3.3.10	operator<<	62
6.3.4	Variable Documentation	62
6.3.4.1	handleProbabilityError	62
6.3.4.2	handleUnexpectedTokenError	62
6.3.4.3	NR_SUBSET_SPECIFIC_TYPES	63
6.3.4.4	WRN_LOGIC_PROPERTY_EVAL_FALSE	63
6.4	multiscale::verification::subsetspecific Namespace Reference	63
6.4.1	Function Documentation	64
6.4.1.1	computeSubsetSpecificType	64
6.4.1.2	computeSubsetSpecificTypeIndex	64
6.4.1.3	validateSubsetSpecificType	65
6.4.1.4	validateSubsetSpecificTypeIndex	65
6.5	multiscale::video Namespace Reference	65
6.6	multiscaletest Namespace Reference	66
6.7	multiscaletest::verification Namespace Reference	67
6.7.1	Function Documentation	67
6.7.1.1	parseInputString	67
7	Class Documentation	69
7.1	multiscale::verification::AbstractSyntaxTree Class Reference	69
7.1.1	Detailed Description	71
7.1.2	Constructor & Destructor Documentation	71
7.1.2.1	AbstractSyntaxTree	71
7.1.2.2	~AbstractSyntaxTree	71
7.1.3	Member Function Documentation	71

7.1.3.1	evaluate	71
7.1.3.2	getComparator	72
7.1.3.3	getProbability	72
7.1.3.4	initialiseTree	72
7.1.4	Member Data Documentation	73
7.1.4.1	ERR_ABSTRACT_SYNTAX_TREE_NOT_INITIALISED	73
7.1.4.2	isInitialised	73
7.1.4.3	probabilisticLogicProperty	73
7.2	multiscale::AdditionOperation Class Reference	73
7.2.1	Detailed Description	74
7.2.2	Member Function Documentation	74
7.2.2.1	operator()	74
7.3	multiscale::AlgorithmException Class Reference	74
7.3.1	Detailed Description	77
7.3.2	Constructor & Destructor Documentation	77
7.3.2.1	AlgorithmException	77
7.3.2.2	AlgorithmException	77
7.3.2.3	AlgorithmException	77
7.4	multiscale::verification::AndConstraintAttribute Class Reference	77
7.4.1	Detailed Description	78
7.4.2	Member Data Documentation	78
7.4.2.1	constraint	78
7.5	multiscale::verification::AndLogicPropertyAttribute Class Reference	78
7.5.1	Detailed Description	78
7.5.2	Member Data Documentation	78
7.5.2.1	logicProperty	78
7.6	multiscale::video::AnnularSector Class Reference	79
7.6.1	Detailed Description	80
7.6.2	Constructor & Destructor Documentation	80
7.6.2.1	AnnularSector	80
7.6.2.2	~AnnularSector	80
7.6.3	Member Function Documentation	80
7.6.3.1	getConcentration	80

CONTENTS	v
7.6.3.2	getEndingAngle 80
7.6.3.3	getEndingRadius 80
7.6.3.4	getStartingAngle 81
7.6.3.5	getStartingRadius 81
7.6.3.6	initialise 81
7.6.3.7	toString 81
7.6.4	Member Data Documentation 81
7.6.4.1	concentration 82
7.6.4.2	endingAngle 82
7.6.4.3	endingRadius 82
7.6.4.4	startingAngle 82
7.6.4.5	startingRadius 82
7.7	multiscale::verification::ApproximateBayesianModelChecker Class Reference 82
7.7.1	Detailed Description 87
7.7.2	Constructor & Destructor Documentation 87
7.7.2.1	ApproximateBayesianModelChecker 87
7.7.2.2	~ApproximateBayesianModelChecker 87
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 90
7.7.3.9	requiresMoreTraces 90
7.7.3.10	updateDerivedModelCheckerForFalseEvaluation 90
7.7.3.11	updateDerivedModelCheckerForTrueEvaluation 90
7.7.3.12	updateMean 91
7.7.3.13	updateMeanAndVariance 91
7.7.3.14	updateModelCheckingResult 91
7.7.3.15	updateModelCheckingResult 91

7.7.3.16	updateModelCheckingResultEnoughTraces	92
7.7.3.17	updateModelCheckingResultNotEnoughTraces	92
7.7.3.18	updateVariance	92
7.7.3.19	validateInput	92
7.7.3.20	validateShapeParameters	93
7.7.3.21	validateVarianceThreshold	93
7.7.4	Member Data Documentation	94
7.7.4.1	alpha	94
7.7.4.2	beta	94
7.7.4.3	ERR_SHAPE_PARAMETERS_BEGIN	94
7.7.4.4	ERR_SHAPE_PARAMETERS_END	94
7.7.4.5	ERR_SHAPE_PARAMETERS_MIDDLE	94
7.7.4.6	ERR_UNEXPECTED_MODEL_CHECKING_RESULT	95
7.7.4.7	ERR_VARIANCE_THRESHOLD_BEGIN	95
7.7.4.8	ERR_VARIANCE_THRESHOLD_END	95
7.7.4.9	mean	95
7.7.4.10	modelCheckingResult	95
7.7.4.11	MSG_OUTPUT_MORE_TRACES_REQUIRED	96
7.7.4.12	MSG_OUTPUT_RESULT_BEGIN	96
7.7.4.13	MSG_OUTPUT_RESULT_END	96
7.7.4.14	MSG_OUTPUT_RESULT_MIDDLE1	96
7.7.4.15	MSG_OUTPUT_RESULT_MIDDLE2	96
7.7.4.16	MSG_OUTPUT_SEPARATOR	96
7.7.4.17	probability	97
7.7.4.18	variance	97
7.7.4.19	varianceThreshold	97
7.8	multiscale::verification::ApproximateBayesianModelCheckerFactory - Class Reference	97
7.8.1	Detailed Description	99
7.8.2	Constructor & Destructor Documentation	100
7.8.2.1	ApproximateBayesianModelCheckerFactory	100
7.8.2.2	~ApproximateBayesianModelCheckerFactory	100
7.8.3	Member Function Documentation	100

7.8.3.1	createInstance	100
7.8.4	Member Data Documentation	100
7.8.4.1	alpha	100
7.8.4.2	beta	100
7.8.4.3	varianceThreshold	101
7.9	multiscaletest::ApproximateBayesianModelCheckerTest Class Reference	101
7.9.1	Detailed Description	104
7.9.2	Constructor & Destructor Documentation	104
7.9.2.1	ApproximateBayesianModelCheckerTest	104
7.9.3	Member Function Documentation	104
7.9.3.1	InitialiseModelChecker	104
7.9.3.2	SetAlphaParamForBetaPrior	104
7.9.3.3	SetBetaParamForBetaPrior	105
7.9.3.4	SetVarianceThreshold	105
7.9.4	Member Data Documentation	105
7.9.4.1	alphaParamForBetaPrior	105
7.9.4.2	betaParamForBetaPrior	105
7.9.4.3	varianceThreshold	106
7.10	multiscale::verification::ApproximateProbabilisticModelChecker Class - Reference	106
7.10.1	Detailed Description	110
7.10.2	Constructor & Destructor Documentation	110
7.10.2.1	ApproximateProbabilisticModelChecker	110
7.10.2.2	~ApproximateProbabilisticModelChecker	110
7.10.3	Member Function Documentation	111
7.10.3.1	acceptsMoreTraces	111
7.10.3.2	doesPropertyHold	111
7.10.3.3	doesPropertyHoldConsideringProbabilityComparator .	111
7.10.3.4	getDetailedResults	111
7.10.3.5	initialise	112
7.10.3.6	initialiseNumberOfRequiredTraces	112
7.10.3.7	isBetweenZeroAndOne	112
7.10.3.8	requiresMoreTraces	112
7.10.3.9	updateDerivedModelCheckerForFalseEvaluation . .	113

7.10.3.10	updateDerivedModelCheckerForTrueEvaluation	113
7.10.3.11	validateInput	113
7.10.4	Member Data Documentation	114
7.10.4.1	delta	114
7.10.4.2	epsilon	114
7.10.4.3	ERR_INVALID_INPUT_BEGIN	114
7.10.4.4	ERR_INVALID_INPUT_END	114
7.10.4.5	ERR_INVALID_INPUT_MIDDLE	114
7.10.4.6	MSG_OUTPUT_MORE_TRACES_REQUIRED	115
7.10.4.7	MSG_OUTPUT_RESULT_BEGIN	115
7.10.4.8	MSG_OUTPUT_RESULT_END	115
7.10.4.9	MSG_OUTPUT_RESULT_MIDDLE1	115
7.10.4.10	MSG_OUTPUT_RESULT_MIDDLE2	115
7.10.4.11	MSG_OUTPUT_SEPARATOR	115
7.10.4.12	nrOfRequiredTraces	116
7.10.4.13	probability	116
7.11	multiscale::verification::ApproximateProbabilisticModelCheckerFactory - Class Reference	116
7.11.1	Detailed Description	118
7.11.2	Constructor & Destructor Documentation	119
7.11.2.1	ApproximateProbabilisticModelCheckerFactory	119
7.11.2.2	~ApproximateProbabilisticModelCheckerFactory	119
7.11.3	Member Function Documentation	119
7.11.3.1	createInstance	119
7.11.4	Member Data Documentation	119
7.11.4.1	delta	119
7.11.4.2	epsilon	120
7.12	multiscaletest::ApproximateProbabilisticModelCheckerTest Class - Reference	120
7.12.1	Detailed Description	123
7.12.2	Constructor & Destructor Documentation	123
7.12.2.1	ApproximateProbabilisticModelCheckerTest	123
7.12.3	Member Function Documentation	123
7.12.3.1	InitialiseModelChecker	123

7.12.3.2	SetDelta	123
7.12.3.3	SetEpsilon	124
7.12.4	Member Data Documentation	124
7.12.4.1	delta	124
7.12.4.2	epsilon	124
7.13	multiscale::verification::BayesianModelChecker Class Reference	124
7.13.1	Detailed Description	129
7.13.2	Constructor & Destructor Documentation	129
7.13.2.1	BayesianModelChecker	130
7.13.2.2	\sim BayesianModelChecker	130
7.13.3	Member Function Documentation	130
7.13.3.1	acceptsMoreTraces	130
7.13.3.2	computeBayesFactorValue	130
7.13.3.3	computeBinomialPDF	131
7.13.3.4	computeMaximumBinomialPDF	131
7.13.3.5	doesPropertyHold	131
7.13.3.6	doesPropertyHoldConsideringProbabilityComparator .	132
7.13.3.7	doesPropertyHoldConsideringResult	132
7.13.3.8	getDetailedResults	132
7.13.3.9	getDetailedUpdatedResults	132
7.13.3.10	indicatorFunction	133
7.13.3.11	initialise	133
7.13.3.12	isValidShapeParameter	133
7.13.3.13	requiresMoreTraces	134
7.13.3.14	updateDerivedModelCheckerForFalseEvaluation	134
7.13.3.15	updateDerivedModelCheckerForTrueEvaluation	134
7.13.3.16	updateModelCheckingResult	134
7.13.3.17	updateModelCheckingResult	135
7.13.3.18	updateModelCheckingResultEnoughTraces	135
7.13.3.19	updateModelCheckingResultNotEnoughTraces	135
7.13.3.20	updateTypeIErrorUpperBound	135
7.13.3.21	validateBayesFactorThreshold	136
7.13.3.22	validateInput	136
7.13.3.23	validateShapeParameters	136

7.13.4 Member Data Documentation	137
7.13.4.1 alpha	137
7.13.4.2 bayesFactorThreshold	137
7.13.4.3 bayesFactorThresholdInverse	137
7.13.4.4 beta	137
7.13.4.5 ERR_BAYES_FACTOR_THRESHOLD_BEGIN . .	138
7.13.4.6 ERR_BAYES_FACTOR_THRESHOLD_END . . .	138
7.13.4.7 ERR_SHAPE_PARAMETERS_BEGIN	138
7.13.4.8 ERR_SHAPE_PARAMETERS_END	138
7.13.4.9 ERR_SHAPE_PARAMETERS_MIDDLE	138
7.13.4.10 ERR_UNEXPECTED_MODEL_CHECKING_RESU- LT	138
7.13.4.11 modelCheckingResult	139
7.13.4.12 MSG_OUTPUT_MORE_TRACES_REQUIRED . .	139
7.13.4.13 MSG_OUTPUT_RESULT_BEGIN	139
7.13.4.14 MSG_OUTPUT_RESULT_END	139
7.13.4.15 MSG_OUTPUT_RESULT_MIDDLE1	139
7.13.4.16 MSG_OUTPUT_RESULT_MIDDLE2	140
7.13.4.17 MSG_OUTPUT_RESULT_MIDDLE3	140
7.13.4.18 MSG_OUTPUT_SEPARATOR	140
7.13.4.19 probability	140
7.13.4.20 typeIErrorUpperBound	140
7.14 multiscale::verification::BayesianModelCheckerFactory Class Reference	141
7.14.1 Detailed Description	143
7.14.2 Constructor & Destructor Documentation	143
7.14.2.1 BayesianModelCheckerFactory	143
7.14.2.2 ~BayesianModelCheckerFactory	143
7.14.3 Member Function Documentation	143
7.14.3.1 createInstance	143
7.14.4 Member Data Documentation	143
7.14.4.1 alpha	143
7.14.4.2 bayesFactorThreshold	143
7.14.4.3 beta	144
7.15 multiscaletest::BayesianModelCheckerTest Class Reference	144

7.15.1	Detailed Description	147
7.15.2	Constructor & Destructor Documentation	147
7.15.2.1	BayesianModelCheckerTest	147
7.15.3	Member Function Documentation	147
7.15.3.1	InitialiseModelChecker	147
7.15.3.2	SetAlphaParamForBetaPrior	147
7.15.3.3	SetBayesFactorThreshold	148
7.15.3.4	SetBetaParamForBetaPrior	148
7.15.4	Member Data Documentation	148
7.15.4.1	alphaParamForBetaPrior	148
7.15.4.2	bayesFactorThreshold	148
7.15.4.3	betaParamForBetaPrior	149
7.16	multiscale::BetaDistribution Class Reference	149
7.16.1	Detailed Description	151
7.16.2	Member Function Documentation	151
7.16.2.1	cdf	151
7.16.2.2	computeCdf	151
7.16.2.3	isValidShapeParameter	152
7.16.2.4	validateShapeParameters	152
7.16.3	Member Data Documentation	152
7.16.3.1	ERR_SHAPE_PARAMETERS_BEGIN	153
7.16.3.2	ERR_SHAPE_PARAMETERS_END	153
7.16.3.3	ERR_SHAPE_PARAMETERS_MIDDLE	153
7.17	multiscale::verification::BinaryNumericFilterAttribute Class Reference	153
7.17.1	Detailed Description	154
7.17.2	Member Data Documentation	154
7.17.2.1	binaryNumericMeasure	154
7.17.2.2	firstFilterNumericMeasure	155
7.17.2.3	secondFilterNumericMeasure	155
7.18	multiscale::verification::BinaryNumericMeasureAttribute Class Reference	155
7.18.1	Detailed Description	155
7.18.2	Member Data Documentation	155
7.18.2.1	binaryNumericMeasureType	156

7.19 multiscale::verification::BinaryNumericMeasureTypeParser Struct - Reference	156
7.19.1 Detailed Description	156
7.19.2 Constructor & Destructor Documentation	156
7.19.2.1 BinaryNumericMeasureTypeParser	156
7.20 multiscale::verification::BinaryNumericNumericAttribute Class Reference	157
7.20.1 Detailed Description	157
7.20.2 Member Data Documentation	158
7.20.2.1 binaryNumericMeasure	158
7.20.2.2 firstNumericMeasure	158
7.20.2.3 secondNumericMeasure	158
7.21 multiscale::verification::BinarySubsetAttribute Class Reference	158
7.21.1 Detailed Description	159
7.21.2 Member Data Documentation	159
7.21.2.1 binarySubsetMeasure	159
7.21.2.2 spatialMeasure	159
7.21.2.3 subset	160
7.22 multiscale::verification::BinarySubsetMeasureAttribute Class Reference	160
7.22.1 Detailed Description	160
7.22.2 Member Data Documentation	160
7.22.2.1 binarySubsetMeasureType	160
7.23 multiscale::verification::BinarySubsetMeasureTypeParser Struct - Reference	161
7.23.1 Detailed Description	161
7.23.2 Constructor & Destructor Documentation	161
7.23.2.1 BinarySubsetMeasureTypeParser	161
7.24 multiscale::BinomialDistribution Class Reference	161
7.24.1 Detailed Description	164
7.24.2 Member Function Documentation	164
7.24.2.1 cdf	164
7.24.2.2 computeCdf	165
7.24.2.3 computePdf	165
7.24.2.4 pdf	165
7.24.2.5 validateInput	166

7.24.2.6	validateNrOfSuccesses	166
7.24.3	Member Data Documentation	167
7.24.3.1	ERR_NR_OF_SUCCESSES_BEGIN	167
7.24.3.2	ERR_NR_OF_SUCCESSES_END	167
7.24.3.3	ERR_NR_OF_SUCCESSES_MIDDLE	167
7.25	multiscale::video::CartesianToConcentrationsConverter Class Reference	167
7.25.1	Detailed Description	170
7.25.2	Constructor & Destructor Documentation	170
7.25.2.1	CartesianToConcentrationsConverter	170
7.25.2.2	~CartesianToConcentrationsConverter	170
7.25.3	Member Function Documentation	170
7.25.3.1	convert	170
7.25.3.2	outputResults	170
7.25.3.3	readConcentrations	170
7.25.3.4	readHeaderLine	171
7.25.3.5	readInputData	171
7.25.4	Member Data Documentation	171
7.25.4.1	concentrations	171
7.25.4.2	ERR_CONC	172
7.25.4.3	ERR_IN_EXTRA_DATA	172
7.25.4.4	ERR_INPUT_OPEN	172
7.25.4.5	ERR_NEG_SIM_TIME	172
7.25.4.6	ERR_NONPOS_DIMENSION	172
7.25.4.7	height	172
7.25.4.8	inputFilepath	173
7.25.4.9	OUTPUT_FILE_EXTENSION	173
7.25.4.10	outputFilepath	173
7.25.4.11	RADIUS_MAX	173
7.25.4.12	RADIUS_MIN	173
7.25.4.13	simulationTime	173
7.25.4.14	width	173
7.26	multiscale::video::CartesianToPolarConverter Class Reference	174
7.26.1	Detailed Description	177
7.26.2	Constructor & Destructor Documentation	177

7.26.2.1	CartesianToPolarConverter	177
7.26.2.2	~CartesianToPolarConverter	177
7.26.3	Member Function Documentation	177
7.26.3.1	convert	177
7.26.3.2	outputResultsAsFile	177
7.26.3.3	outputResultsAsScript	178
7.26.3.4	readConcentrations	178
7.26.3.5	readHeaderLine	178
7.26.3.6	readInputData	179
7.26.3.7	transformToAnnularSectors	179
7.26.4	Member Data Documentation	179
7.26.4.1	annularSectors	179
7.26.4.2	concentrations	179
7.26.4.3	ERR_CONC	179
7.26.4.4	ERR_IN_EXTRA_DATA	180
7.26.4.5	ERR_INPUT_OPEN	180
7.26.4.6	ERR_NEG_SIM_TIME	180
7.26.4.7	ERR_NONPOS_DIMENSION	180
7.26.4.8	inputFilepath	180
7.26.4.9	nrOfConcentricCircles	180
7.26.4.10	nrOfSectors	181
7.26.4.11	OUTPUT_FILE_EXTENSION	181
7.26.4.12	outputFilepath	181
7.26.4.13	RADIUS_MAX	181
7.26.4.14	RADIUS_MIN	181
7.26.4.15	simulationTime	181
7.27	multiscale::analysis::CircularityMeasure Class Reference	182
7.27.1	Detailed Description	182
7.27.2	Member Function Documentation	182
7.27.2.1	compute	182
7.27.2.2	compute	182
7.28	multiscale::analysis::CircularMatFactory Class Reference	183
7.28.1	Detailed Description	186
7.28.2	Constructor & Destructor Documentation	187

7.28.2.1	CircularMatFactory	187
7.28.2.2	~CircularMatFactory	187
7.28.3	Member Function Documentation	187
7.28.3.1	createCircularMask	187
7.28.3.2	createFromViewerImage	187
7.28.3.3	isValidViewerImage	188
7.28.3.4	maxColourBarIntensityFromViewerImage	188
7.28.3.5	processConcentrations	188
7.28.4	Member Data Documentation	189
7.28.4.1	COLOURBAR_MAX_X	189
7.28.4.2	COLOURBAR_MAX_Y	189
7.28.4.3	ERR_UNIMPLEMENTED_METHOD	189
7.28.4.4	INPUT_IMG_HEIGHT	189
7.28.4.5	INPUT_IMG_WIDTH	189
7.28.4.6	INTENSITY_MAX	189
7.28.4.7	ROI_RADIUS	190
7.28.4.8	ROI_START_X	190
7.28.4.9	ROI_START_Y	190
7.29	multiscale::analysis::Cluster Class Reference	190
7.29.1	Detailed Description	194
7.29.2	Constructor & Destructor Documentation	194
7.29.2.1	Cluster	194
7.29.2.2	~Cluster	194
7.29.3	Member Function Documentation	195
7.29.3.1	addEntity	195
7.29.3.2	areValidOriginDependentValues	195
7.29.3.3	getEntities	195
7.29.3.4	getEntitiesCentrePoints	195
7.29.3.5	getEntitiesContourPoints	195
7.29.3.6	getEntitiesConvexHull	196
7.29.3.7	getMinAreaEnclosingCircleCentre	196
7.29.3.8	getMinAreaEnclosingCircleRadius	196
7.29.3.9	getMinAreaEnclosingRect	196
7.29.3.10	getMinAreaEnclosingTriangle	197

7.29.3.11 initialise	197
7.29.3.12 isCircularMeasure	197
7.29.3.13 isRectangularMeasure	197
7.29.3.14 isTriangularMeasure	198
7.29.3.15 setOriginDependentMembers	198
7.29.3.16 type	198
7.29.3.17 updateArea	198
7.29.3.18 updateCentrePoint	199
7.29.3.19 updateClusterednessDegree	199
7.29.3.20 updateDensity	199
7.29.3.21 updatePerimeter	199
7.29.3.22 validateOriginDependentValues	199
7.29.4 Member Data Documentation	200
7.29.4.1 entities	200
7.29.4.2 ERR_ORIGIN_DEPENDENT_VALUES	200
7.29.4.3 ERR_UNDEFINED_SHAPE	200
7.29.4.4 minAreaEnclosingCircleCentre	200
7.29.4.5 minAreaEnclosingCircleRadius	201
7.29.4.6 minAreaEnclosingRect	201
7.29.4.7 minAreaEnclosingTriangle	201
7.30 multiscale::verification::Cluster Class Reference	201
7.30.1 Detailed Description	204
7.31 multiscale::analysis::ClusterDetector Class Reference	204
7.31.1 Detailed Description	209
7.31.2 Constructor & Destructor Documentation	209
7.31.2.1 ClusterDetector	209
7.31.2.2 ~ClusterDetector	209
7.31.3 Member Function Documentation	209
7.31.3.1 addEntitiesToClusters	209
7.31.3.2 analyseClusters	210
7.31.3.3 analyseClustersOriginDependentValues	210
7.31.3.4 clearPreviousDetectionResults	210
7.31.3.5 computeAveragePileUpDegree	211
7.31.3.6 computeClusterednessIndex	211

7.31.3.7 convertEntities	211
7.31.3.8 convertEpsValue	211
7.31.3.9 convertNonPiledUpEntities	212
7.31.3.10 convertPiledUpEntities	212
7.31.3.11 createDetectorSpecificTrackbars	212
7.31.3.12 detectAndAnalyseClusters	212
7.31.3.13 detectClusters	213
7.31.3.14 detectEntitiesInImage	213
7.31.3.15 getClusterConvexHull	214
7.31.3.16 getClusters	214
7.31.3.17 getCollectionOfSpatialEntityPseudo3D	214
7.31.3.18 getDetectorTypeAsString	214
7.31.3.19 getEps	214
7.31.3.20 getMinPoints	215
7.31.3.21 getValidMinPointsValue	215
7.31.3.22 initialiseDetectorSpecificFields	215
7.31.3.23 processImageAndDetect	215
7.31.3.24 setEps	215
7.31.3.25 setMinPoints	216
7.31.3.26 updateClusterOriginDependentValues	216
7.31.4 Member Data Documentation	216
7.31.4.1 clusters	217
7.31.4.2 DETECTOR_TYPE	217
7.31.4.3 entityPileupDegree	217
7.31.4.4 eps	217
7.31.4.5 EPS_MAX	217
7.31.4.6 EPS_MIN	217
7.31.4.7 EPS_REAL_MAX	218
7.31.4.8 EPS_REAL_MIN	218
7.31.4.9 MIN_POINTS_MAX	218
7.31.4.10 MIN_POINTS_MIN	218
7.31.4.11 minPoints	218
7.31.4.12 TRACKBAR_EPS	218
7.31.4.13 TRACKBAR_MINPOINTS	218

7.32 multiscale::verification::CommandLineModelChecking Class Reference	219
7.32.1 Detailed Description	228
7.32.2 Constructor & Destructor Documentation	228
7.32.2.1 CommandLineModelChecking	228
7.32.2.2 ~CommandLineModelChecking	228
7.32.3 Member Function Documentation	228
7.32.3.1 areApproximateBayesianModelCheckingArguments- Present	228
7.32.3.2 areApproximateProbabilisticModelCheckingArguments- Present	229
7.32.3.3 areBayesianModelCheckingArgumentsPresent	229
7.32.3.4 areInvalidExecutionArguments	229
7.32.3.5 areInvalidModelCheckingArguments	230
7.32.3.6 areInvalidModelCheckingArgumentsPresent	230
7.32.3.7 areInvalidModelCheckingTypeSpecificArguments	230
7.32.3.8 areModelCheckingTypeSpecificArgumentsPresent	230
7.32.3.9 areStatisticalModelCheckingArgumentsPresent	231
7.32.3.10 areUnrecognizedArgumentsPresent	231
7.32.3.11 areValidArguments	232
7.32.3.12 areValidArgumentsConsideringConfiguration	232
7.32.3.13 execute	232
7.32.3.14 handleHelpRequest	232
7.32.3.15 initialise	233
7.32.3.16 initialiseAllowedArgumentsConfiguration	233
7.32.3.17 initialiseApproximateBayesianModelChecker	233
7.32.3.18 initialiseApproximateBayesianModelCheckerArguments- Configuration	234
7.32.3.19 initialiseApproximateProbabilisticModelChecker	234
7.32.3.20 initialiseApproximateProbabilisticModelChecker- ArgumentsConfiguration	234
7.32.3.21 initialiseBayesianModelChecker	235
7.32.3.22 initialiseBayesianModelCheckerArgumentsConfiguration	235
7.32.3.23 initialiseClassMembers	235
7.32.3.24 initialiseModelChecker	235
7.32.3.25 initialiseModelCheckerTypeDependentClassMembers	236

7.32.3.26 initialiseModelCheckerTypeSpecificArguments- Configuration	236
7.32.3.27 initialiseModelCheckingManager	236
7.32.3.28 initialiseOptionalArgumentsConfiguration	237
7.32.3.29 initialiseOptionalArgumentsDependentClassMembers	237
7.32.3.30 initialiseProbabilisticBlackBoxModelChecker	237
7.32.3.31 initialiseRequiredArgumentsConfiguration	237
7.32.3.32 initialiseRequiredArgumentsDependentClassMembers	238
7.32.3.33 initialiseStatisticalModelChecker	238
7.32.3.34 initialiseStatisticalModelCheckerArgumentsConfiguration	238
7.32.3.35 isHelpArgumentPresent	239
7.32.3.36 parseAndStoreArgumentsValues	239
7.32.3.37 printHelpClosingMessage	239
7.32.3.38 printHelpContentsMessage	239
7.32.3.39 printHelpIntroMessage	240
7.32.3.40 printHelpMessage	240
7.32.3.41 printModelCheckingInitialisationMessage	240
7.32.3.42 removeApproximateBayesianModelCheckingArguments	240
7.32.3.43 removeApproximateProbabilisticModelChecking- Arguments	241
7.32.3.44 removeBayesianModelCheckingArguments	241
7.32.3.45 removeModelCheckingTypeSpecificArguments	241
7.32.3.46 removeOptionalArguments	242
7.32.3.47 removeRequiredArguments	242
7.32.3.48 removeStatisticalModelCheckingArguments	243
7.32.4 Member Data Documentation	243
7.32.4.1 allowedArguments	243
7.32.4.2 ARG_APPROXIMATE_BAYESIAN_ALPHA_DESC- RIPTION	243
7.32.4.3 ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME- _LONG	243
7.32.4.4 ARG_APPROXIMATE_BAYESIAN_BETA_DESCRI- PTION	244
7.32.4.5 ARG_APPROXIMATE_BAYESIAN_BETA_NAME_- LONG	244

7.32.4.6 ARG_BAYES_FACTOR_THRESHOLD_DESCRIPTION	244
7.32.4.7 ARG_BAYES_FACTOR_THRESHOLD_NAME_LONG	244
7.32.4.8 ARG_BAYESIAN_ALPHA_DESCRIPTION	244
7.32.4.9 ARG_BAYESIAN_ALPHA_NAME_LONG	244
7.32.4.10 ARG_BAYESIAN_BETA_DESCRIPTION	245
7.32.4.11 ARG_BAYESIAN_BETA_NAME_LONG	245
7.32.4.12 ARG_DELTA_DESCRIPTION	245
7.32.4.13 ARG_DELTA_NAME_LONG	245
7.32.4.14 ARG_EPSILON_DESCRIPTION	245
7.32.4.15 ARG_EPSILON_NAME_LONG	246
7.32.4.16 ARG_EXTRA_EVALUATION_PROGRAM_DESCRIPTION	246
7.32.4.17 ARG_EXTRA_EVALUATION_PROGRAM_NAME_BOTH	246
7.32.4.18 ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG	246
7.32.4.19 ARG_EXTRA_EVALUATION_TIME_DESCRIPTION	246
7.32.4.20 ARG_EXTRA_EVALUATION_TIME_NAME_BOTH	247
7.32.4.21 ARG_EXTRA_EVALUATION_TIME_NAME_LONG	247
7.32.4.22 ARG_HELP_DESCRIPTION	247
7.32.4.23 ARG_HELP_NAME_BOTH	247
7.32.4.24 ARG_HELP_NAME_LONG	247
7.32.4.25 ARG_LOGIC_QUERIES_DESCRIPTION	247
7.32.4.26 ARG_LOGIC_QUERIES_NAME_BOTH	248
7.32.4.27 ARG_LOGIC_QUERIES_NAME_LONG	248
7.32.4.28 ARG_MODEL_CHECKER_TYPE_DESCRIPTION	248
7.32.4.29 ARG_MODEL_CHECKER_TYPE_NAME_BOTH	248
7.32.4.30 ARG_MODEL_CHECKER_TYPE_NAME_LONG	248
7.32.4.31 ARG_SPATIAL_TEMPORAL_TRACES_DESCRIPTION	249
7.32.4.32 ARG_SPATIAL_TEMPORAL_TRACES_NAME_BOTH	249
7.32.4.33 ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG	249

7.32.4.34 ARG_TYPE_I_ERROR_DESCRIPTION	249
7.32.4.35 ARG_TYPE_I_ERROR_NAME_LONG	249
7.32.4.36 ARG_TYPE_II_ERROR_DESCRIPTION	249
7.32.4.37 ARG_TYPE_II_ERROR_NAME_LONG	250
7.32.4.38 ARG_VARIANCE_THRESHOLD_DESCRIPTION . .	250
7.32.4.39 ARG_VARIANCE_THRESHOLD_NAME_LONG . .	250
7.32.4.40 ARG_VERBOSE_DESCRIPTION	250
7.32.4.41 ARG_VERBOSE_NAME_BOTH	250
7.32.4.42 ARG_VERBOSE_NAME_LONG	251
7.32.4.43 CONFIG_CAPTION_ALLOWED_ARGUMENTS . . .	251
7.32.4.44 CONFIG_CAPTION_APPROXIMATE_BAYESIAN_- MODEL_CHECKER_ARGUMENTS	251
7.32.4.45 CONFIG_CAPTION_APPROXIMATE_PROBABILI- STIC_MODEL_CHECKER_ARGUMENTS	251
7.32.4.46 CONFIG_CAPTION_BAYESIAN_MODEL_CHECK- ER_ARGUMENTS	251
7.32.4.47 CONFIG_CAPTION_MODEL_CHECKER_TYPE_S- PECIFIC_ARGUMENTS	251
7.32.4.48 CONFIG_CAPTION_OPTIONAL_ARGUMENTS . .	252
7.32.4.49 CONFIG_CAPTION_PROBABILISTIC_BLACK_BO- X_MODEL_CHECKER_ARGUMENTS	252
7.32.4.50 CONFIG_CAPTION_REQUIRED_ARGUMENTS . .	252
7.32.4.51 CONFIG_CAPTION_STATISTICAL_MODEL_CHE- CKER_ARGUMENTS	252
7.32.4.52 ERR_INVALID_COMMAND_LINE_ARGUMENTS . .	252
7.32.4.53 ERR_INVALID_MODEL_CHECKING_ARGUMENTS .	252
7.32.4.54 ERR_INVALID_MODEL_CHECKING_TYPE	253
7.32.4.55 extraEvaluationProgramPath	253
7.32.4.56 extraEvaluationTime	253
7.32.4.57 HELP_AUTHOR_LABEL	253
7.32.4.58 HELP_AUTHOR_MSG	253
7.32.4.59 HELP_COPYRIGHT_LABEL	254
7.32.4.60 HELP_COPYRIGHT_MSG	254
7.32.4.61 HELP_DESCRIPTION_LABEL	254
7.32.4.62 HELP_DESCRIPTION_MSG	254
7.32.4.63 HELP_NAME_LABEL	254

7.32.4.64 HELP_NAME_MSG	254
7.32.4.65 HELP_REPORTING_BUGS_LABEL	255
7.32.4.66 HELP_REPORTING_BUGS_MSG	255
7.32.4.67 HELP_USAGE_LABEL	255
7.32.4.68 HELP_USAGE_MSG	255
7.32.4.69 logicQueriesFilepath	255
7.32.4.70 MODEL_CHECKER_APPROXIMATE_BAYESIAN_- NAME	255
7.32.4.71 MODEL_CHECKER_APPROXIMATE_BAYESIAN_- PARAMETERS_BEGIN	256
7.32.4.72 MODEL_CHECKER_APPROXIMATE_BAYESIAN_- PARAMETERS_END	256
7.32.4.73 MODEL_CHECKER_APPROXIMATE_BAYESIAN_- PARAMETERS_MIDDLE1	256
7.32.4.74 MODEL_CHECKER_APPROXIMATE_BAYESIAN_- PARAMETERS_MIDDLE2	256
7.32.4.75 MODEL_CHECKER_APPROXIMATE_PROBABILI- STIC_NAME	256
7.32.4.76 MODEL_CHECKER_APPROXIMATE_PROBABILI- STIC_PARAMETERS_BEGIN	256
7.32.4.77 MODEL_CHECKER_APPROXIMATE_PROBABILI- STIC_PARAMETERS_END	257
7.32.4.78 MODEL_CHECKER_APPROXIMATE_PROBABILI- STIC_PARAMETERS_MIDDLE	257
7.32.4.79 MODEL_CHECKER_BAYESIAN_NAME	257
7.32.4.80 MODEL_CHECKER_BAYESIAN_PARAMETERS_- BEGIN	257
7.32.4.81 MODEL_CHECKER_BAYESIAN_PARAMETERS_- END	257
7.32.4.82 MODEL_CHECKER_BAYESIAN_PARAMETERS_- MIDDLE1	257
7.32.4.83 MODEL_CHECKER_BAYESIAN_PARAMETERS_- MIDDLE2	258
7.32.4.84 MODEL_CHECKER_PROBABILISTIC_BLACK_BO- X_NAME	258
7.32.4.85 MODEL_CHECKER_PROBABILISTIC_BLACK_BO- X_PARAMETERS	258
7.32.4.86 MODEL_CHECKER_STATISTICAL_NAME	258

7.32.4.87	MODEL_CHECKER_STATISTICAL_PARAMETER-S_BEGIN	258
7.32.4.88	MODEL_CHECKER_STATISTICAL_PARAMETER-S_END	258
7.32.4.89	MODEL_CHECKER_STATISTICAL_PARAMETER-S_MIDDLE	259
7.32.4.90	MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN	259
7.32.4.91	MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC	259
7.32.4.92	MODEL_CHECKER_TYPE_BAYESIAN	259
7.32.4.93	MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX	259
7.32.4.94	MODEL_CHECKER_TYPE_STATISTICAL	260
7.32.4.95	modelCheckerFactory	260
7.32.4.96	modelCheckerParameters	260
7.32.4.97	modelCheckerType	260
7.32.4.98	modelCheckerTypeName	260
7.32.4.99	modelCheckerTypeSpecificArguments	261
7.32.4.100	modelCheckingManager	261
7.32.4.101MSG	MODEL_CHECKING_HELP_REQUESTED	261
7.32.4.102optionalArguments	261	
7.32.4.103requiredArguments	261	
7.32.4.104shouldVerboseDetailedResults	262	
7.32.4.105tracesFolderPath	262	
7.32.4.106variablesMap	262	
7.33	multiscale::verification::ComparatorAttribute Class Reference	262
7.33.1	Detailed Description	263
7.33.2	Member Data Documentation	263
7.33.2.1	comparatorType	263
7.34	multiscale::verification::ComparatorEvaluator Class Reference	263
7.34.1	Detailed Description	264
7.34.2	Member Function Documentation	264
7.34.2.1	evaluate	264
7.35	multiscale::verification::ComparatorNonEqualTypeParser Struct Reference	264

7.35.1	Detailed Description	265
7.35.2	Constructor & Destructor Documentation	265
7.35.2.1	ComparatorNonEqualTypeParser	265
7.36	multiscale::verification::ComparatorTypeParser Struct Reference	265
7.36.1	Detailed Description	265
7.36.2	Constructor & Destructor Documentation	265
7.36.2.1	ComparatorTypeParser	265
7.37	multiscaletest::CompleteTraceTest Class Reference	266
7.37.1	Detailed Description	269
7.37.2	Member Function Documentation	269
7.37.2.1	InitialiseTrace	269
7.38	multiscale::ConsolePrinter Class Reference	269
7.38.1	Detailed Description	272
7.38.2	Member Function Documentation	272
7.38.2.1	getUnixColourCode	272
7.38.2.2	isStdOutTerminalWhichSupportsColour	272
7.38.2.3	printColouredMessage	273
7.38.2.4	printColouredMessageWithColouredTag	273
7.38.2.5	printEmptyLine	274
7.38.2.6	printMessage	274
7.38.2.7	printMessageUsingColour	274
7.38.2.8	printMessageWithColouredTag	275
7.38.2.9	printNewLine	275
7.38.2.10	printNonColouredMessage	276
7.38.2.11	printWarningMessage	276
7.38.2.12	terminalSupportsColour	276
7.38.2.13	terminalSupportsColour	277
7.38.2.14	unixColourCodeToString	277
7.38.3	Member Data Documentation	277
7.38.3.1	CSI_COLOUR_CODE_END_TAG	277
7.38.3.2	CSI_COLOUR_START_VALUE	277
7.38.3.3	CSI_RESET_CODE	278
7.38.3.4	CSI_SEPARATOR	278
7.38.3.5	CSI_START_TAG	278

7.38.3.6	ERR_INVALID_COLOUR_CODE	278
7.38.3.7	SEPARATOR	278
7.38.3.8	TERM_ENV_VARIABLE	278
7.38.3.9	WARNING_TAG	278
7.39	multiscale::verification::ConstraintAttribute Class Reference	279
7.39.1	Detailed Description	281
7.39.2	Member Data Documentation	281
7.39.2.1	firstConstraint	281
7.39.2.2	nextConstraints	281
7.40	multiscale::verification::ConstraintEvaluator Class Reference	281
7.40.1	Detailed Description	282
7.40.2	Member Function Documentation	282
7.40.2.1	evalFilterNumericMeasure	282
7.40.2.2	evalSpatialMeasureConstraint	283
7.40.2.3	evalTypeConstraint	283
7.40.2.4	filterSpatialEntitiesWrtSpatialMeasure	284
7.40.2.5	filterSpatialEntitiesWrtType	284
7.41	multiscale::verification::ConstraintVisitor Class Reference	285
7.41.1	Detailed Description	288
7.41.2	Constructor & Destructor Documentation	288
7.41.2.1	ConstraintVisitor	288
7.41.3	Member Function Documentation	288
7.41.3.1	evaluate	288
7.41.3.2	evaluate	288
7.41.3.3	evaluateNextConstraints	289
7.41.3.4	evaluateNumericMeasure	289
7.41.3.5	evaluateUnarySpatialConstraint	289
7.41.3.6	evaluateUnaryTypeConstraint	290
7.41.3.7	operator()	290
7.41.3.8	operator()	291
7.41.3.9	operator()	291
7.41.3.10	operator()	291
7.41.3.11	operator()	291
7.41.3.12	operator()	292

7.41.3.13 operator()	292
7.41.3.14 operator()	292
7.41.3.15 operator()	293
7.41.3.16 operator()	293
7.41.4 Member Data Documentation	293
7.41.4.1 constraintTimePoint	293
7.41.4.2 initialTimePoint	294
7.42 multiscale::analysis::DataPoint Class Reference	294
7.42.1 Detailed Description	296
7.42.2 Constructor & Destructor Documentation	296
7.42.2.1 ~DataPoint	296
7.42.3 Member Function Documentation	296
7.42.3.1 distanceTo	296
7.43 multiscale::analysis::DBSCAN Class Reference	296
7.43.1 Detailed Description	299
7.43.2 Constructor & Destructor Documentation	299
7.43.2.1 DBSCAN	299
7.43.2.2 ~DBSCAN	299
7.43.3 Member Function Documentation	299
7.43.3.1 addUnclassifiedNodesToSeedsList	299
7.43.3.2 allocateDistanceMatrix	300
7.43.3.3 assignBorderNodesToClusters	300
7.43.3.4 constructDistanceMatrix	300
7.43.3.5 expandCoreCluster	300
7.43.3.6 findClosestCoreDataPoint	301
7.43.3.7 labelUnclassifiedAndNoiseAsBorder	301
7.43.3.8 retrieveNeighbours	302
7.43.3.9 run	302
7.43.3.10 runAlgorithm	302
7.43.4 Member Data Documentation	303
7.43.4.1 CLUSTERING_BORDER	303
7.43.4.2 CLUSTERING_NOISE	303
7.43.4.3 CLUSTERING_UNCLASSIFIED	303
7.43.4.4 distanceMatrix	303

7.43.4.5	eps	304
7.43.4.6	minPoints	304
7.43.4.7	nrOfDataPoints	304
7.44	multiscale::analysis::Detector Class Reference	304
7.44.1	Detailed Description	310
7.44.2	Constructor & Destructor Documentation	311
7.44.2.1	Detector	311
7.44.2.2	~Detector	311
7.44.3	Member Function Documentation	311
7.44.3.1	addAverageMeasuresToPropertyTree	311
7.44.3.2	addNumericStateVariableToPropertyTree	311
7.44.3.3	addSpatialEntitiesToPropertyTree	311
7.44.3.4	addSpatialEntityPropertiesToTree	312
7.44.3.5	addSpatialEntityTypeToPropertyTree	312
7.44.3.6	clearPreviousDetectionResults	312
7.44.3.7	constructPropertyTree	312
7.44.3.8	createDetectorSpecificTrackbars	313
7.44.3.9	createTrackbars	313
7.44.3.10	createTrackbarsWindow	313
7.44.3.11	detect	313
7.44.3.12	detect	313
7.44.3.13	detectInDebugMode	314
7.44.3.14	detectInReleaseMode	314
7.44.3.15	displayImage	314
7.44.3.16	displayResultsInWindow	314
7.44.3.17	findGoodIntersectionPoints	314
7.44.3.18	findGoodPointsForAngle	315
7.44.3.19	getCollectionOfSpatialEntityPseudo3D	315
7.44.3.20	getDetectorTypeAsString	315
7.44.3.21	initialise	315
7.44.3.22	initialiseDetectorSpecificFields	316
7.44.3.23	initialiseDetectorSpecificFieldsIfNotSet	316
7.44.3.24	initialiseDetectorSpecificImageDependentFields	316
7.44.3.25	initialiseImageDependentFields	316

7.44.3.26 initialiseImageOrigin	316
7.44.3.27 isValidInputImage	316
7.44.3.28 minAreaRectCentre	317
7.44.3.29 outputAveragedMeasuresToCsvFile	317
7.44.3.30 outputResults	317
7.44.3.31 outputResultsToCsvFile	317
7.44.3.32 outputResultsToCsvFile	317
7.44.3.33 outputResultsToFile	318
7.44.3.34 outputResultsToImage	318
7.44.3.35 outputResultsToXMLFile	318
7.44.3.36 outputResultsToXMLFile	318
7.44.3.37 outputSpatialEntitiesToCsvFile	318
7.44.3.38 polygonAngle	319
7.44.3.39 polygonAngle	319
7.44.3.40 printOutputErrorMessage	319
7.44.3.41 processImageAndDetect	320
7.44.3.42 processPressedKeyRequest	320
7.44.3.43 setDetectorSpecificFieldsInitialisationFlag	320
7.44.3.44 storeOutputImageOnDisk	320
7.44.4 Member Data Documentation	320
7.44.4.1 avgClusterednessDegree	320
7.44.4.2 avgDensity	321
7.44.4.3 CSV_EXTENSION	321
7.44.4.4 debugMode	321
7.44.4.5 detectMethodCalled	321
7.44.4.6 detectorSpecificFieldsInitialised	321
7.44.4.7 ERR_INVALID_IMAGE	322
7.44.4.8 ERR_OUTPUT_FILE	322
7.44.4.9 ERR_OUTPUT_WITHOUT_DETECT	322
7.44.4.10 image	322
7.44.4.11 IMG_EXTENSION	322
7.44.4.12 KEY_ESC	322
7.44.4.13 KEY_SAVE	322
7.44.4.14 LABEL_ATTRIBUTE	323

7.44.4.15 LABEL_AVG_CLUSTEREDNESS	323
7.44.4.16 LABEL_AVG_DENSITY	323
7.44.4.17 LABEL_COMMENT	323
7.44.4.18 LABEL_COMMENT_CONTENTS	323
7.44.4.19 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE	323
7.44.4.20 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_NAME	323
7.44.4.21 LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_VALUE	323
7.44.4.22 LABEL_EXPERIMENT_TIMEPOINT_SPATIAL_ENTITY	324
7.44.4.23 LABEL_SPATIAL_ENTITY_ANGLE	324
7.44.4.24 LABEL_SPATIAL_ENTITY_AREA	324
7.44.4.25 LABEL_SPATIAL_ENTITY_CENTROID_X	324
7.44.4.26 LABEL_SPATIAL_ENTITY_CENTROID_Y	324
7.44.4.27 LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE	324
7.44.4.28 LABEL_SPATIAL_ENTITY_CLUSTEREDNESS	324
7.44.4.29 LABEL_SPATIAL_ENTITY_DENSITY	324
7.44.4.30 LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN	325
7.44.4.31 LABEL_SPATIAL_ENTITY_PERIMETER	325
7.44.4.32 LABEL_SPATIAL_ENTITY_PSEUDO_3D	325
7.44.4.33 LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE	325
7.44.4.34 LABEL_SPATIAL_ENTITY_SHAPE	325
7.44.4.35 LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE	325
7.44.4.36 LABEL_SPATIAL_ENTITY_TYPE	325
7.44.4.37 origin	325
7.44.4.38 OUTPUT_CLUSTEREDNESS	326
7.44.4.39 OUTPUT_DENSITY	326
7.44.4.40 outputFilepath	326
7.44.4.41 outputImage	326
7.44.4.42 WIN_OUTPUT_IMAGE	326
7.44.4.43 XML_EXTENSION	326
7.45 multiscale::verification::DifferenceAttribute Class Reference	327

7.45.1	Detailed Description	327
7.45.2	Member Data Documentation	328
7.45.2.1	comparator	328
7.45.2.2	IhsNumericMeasure	328
7.45.2.3	rhsNumericMeasure	328
7.46	multiscale::Distribution Class Reference	328
7.46.1	Detailed Description	331
7.46.2	Member Function Documentation	331
7.46.2.1	validateProbability	331
7.46.3	Member Data Documentation	331
7.46.3.1	ERR_PROBABILITY_VALUE_BEGIN	331
7.46.3.2	ERR_PROBABILITY_VALUE_END	331
7.47	multiscale::DivisionOperation Class Reference	332
7.47.1	Detailed Description	332
7.47.2	Member Function Documentation	332
7.47.2.1	operator()	332
7.48	multiscaletest::EmptyTraceTest Class Reference	332
7.48.1	Detailed Description	335
7.48.2	Member Function Documentation	335
7.48.2.1	InitialiseTrace	335
7.49	multiscale::analysis::Entity Class Reference	335
7.49.1	Detailed Description	339
7.49.2	Constructor & Destructor Documentation	339
7.49.2.1	Entity	339
7.49.2.2	Entity	339
7.49.2.3	~Entity	339
7.49.3	Member Function Documentation	339
7.49.3.1	areValid	339
7.49.3.2	distanceTo	340
7.49.3.3	distanceTo	340
7.49.3.4	getArea	340
7.49.3.5	getCentre	340
7.49.3.6	getContourPoints	340
7.49.3.7	getPerimeter	340

7.49.3.8	getPileUpDegree	341
7.49.3.9	toString	341
7.49.3.10	validateInputValues	341
7.49.4	Member Data Documentation	341
7.49.4.1	area	341
7.49.4.2	centre	341
7.49.4.3	contourPoints	342
7.49.4.4	ERR_DISTANCE	342
7.49.4.5	ERR_INPUT	342
7.49.4.6	OUTPUT_SEPARATOR	342
7.49.4.7	perimeter	342
7.49.4.8	pileUpDegree	342
7.50	multiscale::verification::EquivalenceConstraintAttribute Class Reference	343
7.50.1	Detailed Description	343
7.50.2	Member Data Documentation	343
7.50.2.1	constraint	343
7.51	multiscale::verification::EquivalenceLogicPropertyAttribute Class - Reference	343
7.51.1	Detailed Description	344
7.51.2	Member Data Documentation	344
7.51.2.1	logicProperty	344
7.52	EuclideanDataPoint Class Reference	345
7.52.1	Detailed Description	346
7.52.2	Constructor & Destructor Documentation	347
7.52.2.1	EuclideanDataPoint	347
7.52.2.2	EuclideanDataPoint	347
7.52.2.3	~EuclideanDataPoint	347
7.52.3	Member Function Documentation	347
7.52.3.1	distanceTo	347
7.52.4	Member Data Documentation	347
7.52.4.1	x	347
7.52.4.2	y	347
7.53	multiscale::ExceptionHandler Class Reference	348
7.53.1	Detailed Description	348

7.53.2 Member Function Documentation	348
7.53.2.1 printErrorMessage	348
7.54 multiscale::FileOpenException Class Reference	348
7.54.1 Detailed Description	351
7.54.2 Constructor & Destructor Documentation	351
7.54.2.1 FileOpenException	351
7.54.2.2 FileOpenException	351
7.54.2.3 FileOpenException	351
7.55 multiscale::Filesystem Class Reference	351
7.55.1 Detailed Description	353
7.55.2 Member Function Documentation	353
7.55.2.1 getFilesInFolder	353
7.55.2.2 isValidFilePath	353
7.55.2.3 isValidFolderPath	353
7.55.2.4 nativeFormatFilePath	354
7.55.3 Member Data Documentation	354
7.55.3.1 ERR_INVALID_PATH	354
7.56 multiscale::verification::FilterNumericMeasureAttribute Class Reference .	354
7.56.1 Detailed Description	355
7.56.2 Member Data Documentation	355
7.56.2.1 filterNumericMeasure	355
7.57 multiscale::verification::FilterNumericVisitor Class Reference	355
7.57.1 Detailed Description	357
7.57.2 Constructor & Destructor Documentation	357
7.57.2.1 FilterNumericVisitor	357
7.57.3 Member Function Documentation	358
7.57.3.1 evaluate	358
7.57.3.2 evaluate	358
7.57.3.3 operator()	358
7.57.3.4 operator()	359
7.57.3.5 operator()	359
7.57.3.6 operator()	359
7.57.3.7 operator()	360
7.57.4 Member Data Documentation	360

7.57.4.1	spatialEntity	360
7.57.4.2	timePoint	360
7.58	multiscale::verification::FilterSubsetAttribute Class Reference	361
7.58.1	Detailed Description	361
7.58.2	Member Data Documentation	361
7.58.2.1	constraint	362
7.58.2.2	subsetSpecific	362
7.59	multiscale::verification::FutureLogicPropertyAttribute Class Reference	362
7.59.1	Detailed Description	362
7.59.2	Member Data Documentation	362
7.59.2.1	endTimepoint	363
7.59.2.2	logicProperty	363
7.59.2.3	startTimepoint	363
7.60	multiscale::Geometry2D Class Reference	363
7.60.1	Detailed Description	366
7.60.2	Member Function Documentation	366
7.60.2.1	angleBtwPoints	366
7.60.2.2	angleOfLineWrtOxAxis	367
7.60.2.3	areaOfTriangle	367
7.60.2.4	areCollinear	367
7.60.2.5	areEqualPoints	368
7.60.2.6	areIdenticalLines	368
7.60.2.7	areIdenticalLines	368
7.60.2.8	areOnTheSameSideOfLine	369
7.60.2.9	distanceBtwPoints	369
7.60.2.10	distanceBtwPoints	370
7.60.2.11	distanceFromPointToLine	370
7.60.2.12	findPointsOnEdge	370
7.60.2.13	inverseTranslate	371
7.60.2.14	isAngleBetween	371
7.60.2.15	isAngleBetweenNonReflex	371
7.60.2.16	isBetweenCoordinates	372
7.60.2.17	isOppositeAngleBetween	372
7.60.2.18	isOppositeAngleBetweenNonReflex	372

7.60.2.19	isPointOnEdge	373
7.60.2.20	isPointOnLineSegment	373
7.60.2.21	lineCircleIntersection	374
7.60.2.22	lineCircleOneIntersectionPoint	374
7.60.2.23	lineCircleTwoIntersectionPoints	375
7.60.2.24	lineEquationDeterminedByPoints	375
7.60.2.25	lineIntersection	375
7.60.2.26	lineIntersection	376
7.60.2.27	lineIntersection	377
7.60.2.28	lineSegmentCircleIntersection	377
7.60.2.29	lineSegmentIntersection	378
7.60.2.30	middlePoint	378
7.60.2.31	minimumDistancePointIndex	378
7.60.2.32	oppositeAngle	379
7.60.2.33	orthogonalLineToAnotherLineEdgePoints	379
7.60.2.34	slopeOfLine	380
7.60.2.35	translate	380
7.60.3	Member Data Documentation	380
7.60.3.1	MATRIX_START_INDEX	380
7.60.3.2	PI	380
7.61	multiscale::verification::GlobalLogicPropertyAttribute Class Reference	381
7.61.1	Detailed Description	381
7.61.2	Member Data Documentation	381
7.61.2.1	endTimepoint	381
7.61.2.2	logicProperty	381
7.61.2.3	startTimepoint	382
7.62	grammar Class Reference	383
7.63	multiscale::verification::ImplicationConstraintAttribute Class Reference	384
7.63.1	Detailed Description	384
7.63.2	Member Data Documentation	384
7.63.2.1	constraint	384
7.64	multiscale::verification::ImplicationLogicPropertyAttribute Class Reference	384
7.64.1	Detailed Description	385
7.64.2	Member Data Documentation	385

7.64.2.1	logicProperty	385
7.65	multiscale::IndexOutOfBoundsException Class Reference	385
7.65.1	Detailed Description	388
7.65.2	Constructor & Destructor Documentation	388
7.65.2.1	IndexOutOfBoundsException	388
7.65.2.2	IndexOutOfBoundsException	388
7.65.2.3	IndexOutOfBoundsException	388
7.66	multiscale::InvalidInputException Class Reference	388
7.66.1	Detailed Description	391
7.66.2	Constructor & Destructor Documentation	391
7.66.2.1	InvalidInputException	391
7.66.2.2	InvalidInputException	391
7.66.2.3	InvalidInputException	391
7.67	multiscale::IOException Class Reference	391
7.67.1	Detailed Description	394
7.67.2	Constructor & Destructor Documentation	394
7.67.2.1	IOException	394
7.67.2.2	IOException	394
7.67.2.3	IOException	394
7.68	multiscale::LexicographicNumberIterator Class Reference	394
7.68.1	Detailed Description	397
7.68.2	Constructor & Destructor Documentation	397
7.68.2.1	LexicographicNumberIterator	397
7.68.2.2	~LexicographicNumberIterator	398
7.68.3	Member Function Documentation	398
7.68.3.1	digitsToNumber	398
7.68.3.2	hasNextInitialised	398
7.68.3.3	initialise	398
7.68.3.4	isLargerThanUpperBound	398
7.68.3.5	number	399
7.68.3.6	numberToDigits	399
7.68.3.7	padWithZeros	399
7.68.3.8	resetCurrentNumber	400
7.68.3.9	reverseDigits	400

7.68.4 Member Data Documentation	400
7.68.4.1 currentNumberDigits	400
7.68.4.2 upperBoundDigits	400
7.69 multiscale::verification::LogicPropertyAttribute Class Reference	401
7.69.1 Detailed Description	403
7.69.2 Constructor & Destructor Documentation	403
7.69.2.1 LogicPropertyAttribute	403
7.69.2.2 LogicPropertyAttribute	403
7.69.3 Member Data Documentation	403
7.69.3.1 firstLogicProperty	403
7.69.3.2 nextLogicProperties	404
7.70 multiscale::verification::LogicPropertyDataReader Class Reference	404
7.70.1 Detailed Description	406
7.70.2 Member Function Documentation	407
7.70.2.1 appendLineUsingStringBuilder	407
7.70.2.2 createNewLogicProperty	407
7.70.2.3 processLineFromInputFile	407
7.70.2.4 readLogicPropertiesFromFile	408
7.70.2.5 readLogicPropertiesFromOpenStream	408
7.70.2.6 readLogicPropertiesFromValidFilepath	408
7.70.2.7 removeStringBuilderContents	409
7.70.3 Member Data Documentation	409
7.70.3.1 CHAR_START_COMMENT	409
7.70.3.2 CHAR_START_LOGIC_PROPERTY	409
7.70.3.3 ERR_INVALID_INPUT_PATH	409
7.70.3.4 ERR_OPEN_INPUT_FILE	409
7.70.3.5 stringBuilder	409
7.71 multiscale::verification::LogicPropertyVisitor Class Reference	410
7.71.1 Detailed Description	415
7.71.2 Constructor & Destructor Documentation	415
7.71.2.1 LogicPropertyVisitor	415
7.71.3 Member Function Documentation	415
7.71.3.1 constructEvaluationLogicProperty	415
7.71.3.2 evaluate	416

7.71.3.3 evaluate	416
7.71.3.4 evaluateDifference	416
7.71.3.5 evaluateFutureLogicProperty	417
7.71.3.6 evaluateGlobalLogicProperty	417
7.71.3.7 evaluateNextKLogicProperty	418
7.71.3.8 evaluateNextKLogicProperty	418
7.71.3.9 evaluateNextLogicProperties	418
7.71.3.10 evaluateNextLogicProperty	419
7.71.3.11 evaluateNumericMeasure	419
7.71.3.12 evaluateNumericNumericComparison	420
7.71.3.13 evaluateNumericSpatialMeasure	420
7.71.3.14 evaluateNumericSpatialNumericComparison	421
7.71.3.15 evaluatePrecedingLogicProperties	421
7.71.3.16 evaluateUntilLogicProperty	421
7.71.3.17 operator()	422
7.71.3.18 operator()	422
7.71.3.19 operator()	423
7.71.3.20 operator()	423
7.71.3.21 operator()	423
7.71.3.22 operator()	424
7.71.3.23 operator()	424
7.71.3.24 operator()	425
7.71.3.25 operator()	425
7.71.3.26 operator()	425
7.71.3.27 operator()	426
7.71.3.28 operator()	426
7.71.3.29 operator()	426
7.71.3.30 operator()	427
7.71.3.31 operator()	427
7.71.3.32 operator()	428
7.71.3.33 printExceptionMessage	428
7.71.4 Member Data Documentation	428
7.71.4.1 evaluationLogicProperty	428
7.71.4.2 precedingTruthValue	429

7.71.4.3	trace	429
7.72	multiscale::analysis::MatFactory Class Reference	429
7.72.1	Detailed Description	432
7.72.2	Constructor & Destructor Documentation	433
7.72.2.1	MatFactory	433
7.72.2.2	~MatFactory	433
7.72.3	Member Function Documentation	433
7.72.3.1	convertToIntensity	433
7.72.3.2	create	433
7.72.3.3	createFromViewerImage	434
7.72.3.4	initInputFile	434
7.72.3.5	isValidViewerImage	434
7.72.3.6	maxColourBarIntensityFromViewerImage	434
7.72.3.7	processConcentrations	435
7.72.4	Member Data Documentation	435
7.72.4.1	cols	435
7.72.4.2	ERR_IMG_RESOLUTION	435
7.72.4.3	ERR_IN_EXTRA_DATA	435
7.72.4.4	ERR_INPUT_OPEN	436
7.72.4.5	rows	436
7.72.4.6	simulationTime	436
7.73	multiscale::MinEnclosingTriangleFinder Class Reference	436
7.73.1	Detailed Description	441
7.73.2	Constructor & Destructor Documentation	441
7.73.2.1	MinEnclosingTriangleFinder	441
7.73.2.2	~MinEnclosingTriangleFinder	441
7.73.3	Member Function Documentation	442
7.73.3.1	advance	442
7.73.3.2	advanceBToRightChain	442
7.73.3.3	areIdenticalLines	442
7.73.3.4	areIntersectingLines	443
7.73.3.5	find	443
7.73.3.6	findGammaIntersectionPoints	444
7.73.3.7	findMinEnclosingTriangle	444

7.73.3.8	findMinEnclosingTriangle	445
7.73.3.9	findMinTriangle	445
7.73.3.10	findVertexCOnSideB	445
7.73.3.11	gamma	446
7.73.3.12	height	446
7.73.3.13	height	447
7.73.3.14	initialise	447
7.73.3.15	initialiseAlgorithmVariables	447
7.73.3.16	initialiseConvexPolygon	448
7.73.3.17	intersects	448
7.73.3.18	intersectsAbove	448
7.73.3.19	intersectsAboveOrBelow	449
7.73.3.20	intersectsBelow	449
7.73.3.21	isFlushAngleBetweenPredecessorAndSuccessor	450
7.73.3.22	isGammaAngleBetween	450
7.73.3.23	isGammaAngleEqualTo	450
7.73.3.24	isLocalMinimalTriangle	451
7.73.3.25	isNotBTangency	451
7.73.3.26	isValidMinimalTriangle	451
7.73.3.27	lineEquationParameters	452
7.73.3.28	middlePointOfSideB	452
7.73.3.29	moveAlfLowAndBlfHigh	452
7.73.3.30	predecessor	452
7.73.3.31	returnMinEnclosingTriangle	453
7.73.3.32	searchForBTangency	453
7.73.3.33	successor	453
7.73.3.34	updateMinEnclosingTriangle	454
7.73.3.35	updateSideB	454
7.73.3.36	updateSidesBA	454
7.73.3.37	updateSidesCA	455
7.73.4	Member Data Documentation	455
7.73.4.1	a	455
7.73.4.2	area	455
7.73.4.3	b	455

7.73.4.4	c	456
7.73.4.5	CONVEX_HULL_CLOCKWISE	456
7.73.4.6	ERR_MIDPOINT_SIDE_B	456
7.73.4.7	ERR_NR_POINTS	456
7.73.4.8	ERR_SIDE_B_GAMMA	456
7.73.4.9	ERR_TRIANGLE_VERTICES	456
7.73.4.10	ERR_VERTEX_C_ON_SIDE_B	457
7.73.4.11	INTERSECTS_ABOVE	457
7.73.4.12	INTERSECTS_BELOW	457
7.73.4.13	INTERSECTS_CRITICAL	457
7.73.4.14	INTERSECTS_LIMIT	457
7.73.4.15	nrOfPoints	457
7.73.4.16	polygon	457
7.73.4.17	sideAEndVertex	458
7.73.4.18	sideAStartVertex	458
7.73.4.19	sideBEndVertex	458
7.73.4.20	sideBStartVertex	458
7.73.4.21	sideCEndVertex	458
7.73.4.22	sideCStartVertex	459
7.73.4.23	VALIDATION_SIDE_A_TANGENT	459
7.73.4.24	VALIDATION_SIDE_B_TANGENT	459
7.73.4.25	VALIDATION_SIDES_FLUSH	459
7.73.4.26	validationFlag	459
7.73.4.27	vertexA	460
7.73.4.28	vertexB	460
7.73.4.29	vertexC	460
7.74	multiscaletest::MinEnclosingTriangleFinderTest Class Reference	460
7.74.1	Detailed Description	464
7.74.2	Constructor & Destructor Documentation	464
7.74.2.1	MinEnclosingTriangleFinderTest	464
7.74.2.2	~MinEnclosingTriangleFinderTest	464
7.74.3	Member Function Documentation	464
7.74.3.1	ArePointsEnclosed	464
7.74.3.2	GetRandomNrOfExecutions	465

7.74.3.3	GetRandomNrOfPoints	465
7.74.3.4	IsOneEdgeFlush	465
7.74.3.5	IsTriangleTouchingPolygon	465
7.74.3.6	RunTest	465
7.74.3.7	TestMorePoints	465
7.74.3.8	TestMorePointsAndNonEmptyTriangle	466
7.74.3.9	TestNoPoints	466
7.74.3.10	TestOnePoint	466
7.74.3.11	TestPointsWithNegativeCoordinates	466
7.74.3.12	TestPointsWithNegativeXCoordinate	466
7.74.3.13	TestPointsWithNegativeYCoordinate	466
7.74.3.14	TestRandomPoints	466
7.74.3.15	TestThreePoints	467
7.74.3.16	TestTwoPoints	467
7.74.3.17	ValidateTestResults	467
7.74.4	Member Data Documentation	467
7.74.4.1	area	467
7.74.4.2	convexHull	467
7.74.4.3	MAX_NR_EXECUTIONS	467
7.74.4.4	MAX_NR_POINTS	468
7.74.4.5	MIN_NR_EXECUTIONS	468
7.74.4.6	MIN_NR_POINTS	468
7.74.4.7	POINT_IN_TRIANGLE_THRESH	468
7.74.4.8	points	468
7.74.4.9	triangle	468
7.75	multiscale::verification::ModelChecker Class Reference	468
7.75.1	Detailed Description	472
7.75.2	Constructor & Destructor Documentation	472
7.75.2.1	ModelChecker	472
7.75.2.2	~ModelChecker	473
7.75.3	Member Function Documentation	473
7.75.3.1	acceptsMoreTraces	473
7.75.3.2	doesPropertyHold	473
7.75.3.3	doesPropertyHoldUsingPValues	473

7.75.3.4	evaluate	473
7.75.3.5	getDetailedResults	474
7.75.3.6	getDetailedResultsUsingPValues	474
7.75.3.7	isGreaterThanOrEqualToComparator	474
7.75.3.8	requiresMoreTraces	475
7.75.3.9	updateAlternativeHypothesisPValue	475
7.75.3.10	updateDerivedModelCheckerForFalseEvaluation . . .	475
7.75.3.11	updateDerivedModelCheckerForTrueEvaluation . . .	476
7.75.3.12	updateHypothesesPValues	476
7.75.3.13	updateHypothesesPValuesConsideringComparator .	476
7.75.3.14	updateHypothesesPValuesForGreaterThan	477
7.75.3.15	updateHypothesesPValuesForLessThan	477
7.75.3.16	updateModelChecker	477
7.75.3.17	updateModelCheckerForEvaluationResult	478
7.75.3.18	updateModelCheckerForFalseEvaluation	478
7.75.3.19	updateModelCheckerForTrueEvaluation	478
7.75.3.20	updateNullAndAlternativeHypothesesPValues . . .	478
7.75.3.21	updateNullHypothesisPValue	479
7.75.4	Member Data Documentation	479
7.75.4.1	abstractSyntaxTree	479
7.75.4.2	alternativeHypothesisPValue	480
7.75.4.3	arePValuesUpdatedFlag	480
7.75.4.4	MSG_OUTPUT_P_VALUE_BEGIN	480
7.75.4.5	MSG_OUTPUT_P_VALUE_END	480
7.75.4.6	MSG_OUTPUT_P_VALUE_MIDDLE1	480
7.75.4.7	MSG_OUTPUT_P_VALUE_MIDDLE2	480
7.75.4.8	nullHypothesisPValue	481
7.75.4.9	totalNumberOfEvaluations	481
7.75.4.10	totalNumberOfTrueEvaluations	481
7.76	multiscale::verification::ModelCheckerFactory Class Reference	482
7.76.1	Detailed Description	482
7.76.2	Constructor & Destructor Documentation	482
7.76.2.1	ModelCheckerFactory	482
7.76.2.2	~ModelCheckerFactory	482

7.76.3	Member Function Documentation	483
7.76.3.1	createInstance	483
7.77	multiscaletest::ModelCheckerTest Class Reference	483
7.77.1	Detailed Description	486
7.77.2	Member Function Documentation	486
7.77.2.1	Initialise	486
7.77.2.2	InitialiseAbstractSyntaxTree	486
7.77.2.3	InitialiseConstantDecreasingSpatioTemporalTrace	486
7.77.2.4	InitialiseConstantIncreasingSpatioTemporalTrace	486
7.77.2.5	InitialiseConstantSpatioTemporalTrace	487
7.77.2.6	InitialiseDecreasingConstantIncreasingSpatioTemporalTrace	487
7.77.2.7	InitialiseDecreasingConstantSpatioTemporalTrace	487
7.77.2.8	InitialiseDecreasingIncreasingSpatioTemporalTrace	487
7.77.2.9	InitialiseDecreasingSpatioTemporalTrace	487
7.77.2.10	InitialiseIncreasingConstantDecreasingSpatioTemporalTrace	487
7.77.2.11	InitialiseIncreasingConstantIncreasingSpatioTemporalTrace	487
7.77.2.12	InitialiseIncreasingConstantSpatioTemporalTrace	487
7.77.2.13	InitialiseIncreasingDecreasingSpatioTemporalTrace	488
7.77.2.14	InitialiseIncreasingSpatioTemporalTrace	488
7.77.2.15	InitialiseModelChecker	488
7.77.2.16	InitialiseSpatioTemporalTraces	488
7.77.2.17	InitialiseSpatioTemporalTraceWithAreaValues	488
7.77.2.18	RunModelCheckingTest	489
7.77.2.19	RunTest	489
7.77.2.20	ValidateTestResults	489
7.77.3	Member Data Documentation	489
7.77.3.1	abstractSyntaxTree	489
7.77.3.2	evaluationResult	490
7.77.3.3	modelChecker	490
7.77.3.4	traces	490
7.78	multiscale::verification::ModelCheckingException Class Reference	490
7.78.1	Detailed Description	493

7.78.2	Constructor & Destructor Documentation	493
7.78.2.1	ModelCheckingException	493
7.78.2.2	ModelCheckingException	493
7.79	multiscale::verification::ModelCheckingHelpRequestException Class Reference	493
7.79.1	Detailed Description	496
7.79.2	Constructor & Destructor Documentation	496
7.79.2.1	ModelCheckingHelpRequestException	496
7.79.2.2	ModelCheckingHelpRequestException	496
7.80	multiscale::verification::ModelCheckingManager Class Reference	496
7.80.1	Detailed Description	500
7.80.2	Constructor & Destructor Documentation	500
7.80.2.1	ModelCheckingManager	500
7.80.2.2	~ModelCheckingManager	500
7.80.3	Member Function Documentation	500
7.80.3.1	areUnfinishedModelCheckingTasks	500
7.80.3.2	createModelCheckers	500
7.80.3.3	createNewEvaluationResults	501
7.80.3.4	executeExtraEvaluationProgram	501
7.80.3.5	executeExtraEvaluationProgramAndPrintMessage	501
7.80.3.6	getNextSpatialTemporalTrace	502
7.80.3.7	initialise	502
7.80.3.8	initialiseLogicProperties	502
7.80.3.9	isEvaluationTimeRemaining	503
7.80.3.10	isValidLogicProperty	503
7.80.3.11	outputDetailedEvaluationResults	503
7.80.3.12	outputModelCheckerResults	503
7.80.3.13	outputModelCheckersResults	504
7.80.3.14	outputModelCheckersResultsAndPrintMessage	504
7.80.3.15	parseLogicProperties	504
7.80.3.16	parseLogicPropertiesAndPrintMessage	504
7.80.3.17	parseLogicProperty	505
7.80.3.18	parseLogicPropertyAndPrintMessages	505
7.80.3.19	printParsingMessage	505

7.80.3.20	runModelCheckerForTrace	506
7.80.3.21	runModelCheckers	506
7.80.3.22	runModelCheckersAndPrintMessage	506
7.80.3.23	runModelCheckersAndRequestAdditionalTraces . . .	507
7.80.3.24	runModelCheckersForCurrentlyExistingTraces . . .	507
7.80.3.25	runModelCheckersForTrace	507
7.80.3.26	runModelCheckingAndOutputResults	508
7.80.3.27	runModelCheckingTasks	508
7.80.3.28	setExtraEvaluationProgramPath	508
7.80.3.29	setShouldPrintDetailedEvaluation	509
7.80.3.30	storeNewSpatialTemporalTracePath	509
7.80.3.31	updateEvaluationResults	509
7.80.3.32	updateExtraEvaluationStartTime	510
7.80.3.33	updateTraceReader	510
7.80.3.34	waitBeforeRetry	510
7.80.4	Member Data Documentation	510
7.80.4.1	abstractSyntaxTrees	510
7.80.4.2	evaluationResults	510
7.80.4.3	extraEvaluationElapsed Time	511
7.80.4.4	extraEvaluationProgramPath	511
7.80.4.5	extraEvaluationStartTime	511
7.80.4.6	extraEvaluationTime	511
7.80.4.7	logicProperties	511
7.80.4.8	logicPropertyReader	512
7.80.4.9	modelCheckers	512
7.80.4.10	parser	512
7.80.4.11	PARSER_EMPTY_LOGIC_PROPERTY	512
7.80.4.12	shouldPrintDetailedEvaluation	512
7.80.4.13	TRACE_INPUT_REFRESH_TIMEOUT	513
7.80.4.14	traceReader	513
7.80.4.15	tracesPaths	513
7.81	multiscale::verification::ModelCheckingOutputWriter Class Reference .	513
7.81.1	Detailed Description	518
7.81.2	Member Function Documentation	518

7.81.2.1	isTraceEvaluatedForLogicProperty	518
7.81.2.2	isTraceEvaluatedTrueForLogicProperty	519
7.81.2.3	printDetailedEvaluationResults	519
7.81.2.4	printDetailedEvaluationResults	520
7.81.2.5	printDetailedEvaluationResultsForLogicProperties . .	520
7.81.2.6	printDetailedEvaluationResultsIntroductionMessage .	521
7.81.2.7	printDetailedTraceEvaluationResult	521
7.81.2.8	printEvaluationResultsSummary	522
7.81.2.9	printEvaluationResultsSummary	522
7.81.2.10	printExecuteExtraEvaluationProgramMessage	523
7.81.2.11	printFailedMessage	523
7.81.2.12	printInitialisationMessage	523
7.81.2.13	printIntroductionMessage	524
7.81.2.14	printLogicPropertyDetailedEvaluationResults	524
7.81.2.15	printLogicPropertyForResult	525
7.81.2.16	printLogicPropertyWithTag	525
7.81.2.17	printModelCheckingDetailedResult	526
7.81.2.18	printModelCheckingResult	526
7.81.2.19	printModelCheckingResultMessage	526
7.81.2.20	printModelCheckingResultsIntroductionMessage . .	527
7.81.2.21	printParsingLogicPropertiesBeginMessage	527
7.81.2.22	printParsingLogicPropertiesEndMessage	527
7.81.2.23	printParsingLogicPropertyMessage	527
7.81.2.24	printResultTag	528
7.81.2.25	printSeparatorTag	528
7.81.2.26	printStartModelCheckingExecutionMessage	528
7.81.2.27	printStartTraceEvaluationMessage	529
7.81.2.28	printSuccessMessage	529
7.81.2.29	printTimeoutMessage	529
7.81.2.30	printTraceEvaluationResult	530
7.81.2.31	printTruthValueDependentMessage	530
7.81.2.32	updateSummaryEvaluationResults	530
7.81.3	Member Data Documentation	531
7.81.3.1	MSG_EVALUATION_RESULTS_INTRODUCTION .	531

7.81.3.2 MSG_EVALUATION_SUMMARY_BEGIN	531
7.81.3.3 MSG_EVALUATION_SUMMARY_END	531
7.81.3.4 MSG_EXECUTION_TIMEOUT_BEGIN	532
7.81.3.5 MSG_EXECUTION_TIMEOUT_END	532
7.81.3.6 MSG_INIT_EXECUTION_PARAMETERS	532
7.81.3.7 MSG_INIT_EXTRA_EVALUATION_TIME	532
7.81.3.8 MSG_INIT_LOGIC_PROPERTIES_PATH	532
7.81.3.9 MSG_INIT_TRACES_FOLDER_PATH	532
7.81.3.10 MSG_INTRO_CONTACT	533
7.81.3.11 MSG_INTRO_COPYRIGHT	533
7.81.3.12 MSG_INTRO_MODEL_CHECKING_PARAMETERS .	533
7.81.3.13 MSG_INTRO_MODEL_CHECKING_TYPE	533
7.81.3.14 MSG_INTRO_NAME	533
7.81.3.15 MSG_LOGIC_PROPERTY_HOLDS	533
7.81.3.16 MSG_LOGIC_PROPERTY_HOLDS_FALSE	534
7.81.3.17 MSG_LOGIC_PROPERTY_HOLDS_TRUE	534
7.81.3.18 MSG_PARSING_INTRODUCTION	534
7.81.3.19 MSG_RESULTS_INTRODUCTION	534
7.81.3.20 MSG_START_EXTRA_EVALUATION_PROGRAM- _EXECUTION	534
7.81.3.21 MSG_START_MODEL_CHECKING_EXECUTION .	534
7.81.3.22 MSG_START_TRACE_EVALUATION	535
7.81.3.23 TAG_DETAILS	535
7.81.3.24 TAG_EXECUTE	535
7.81.3.25 TAG_FAILED	535
7.81.3.26 TAG_FALSE	535
7.81.3.27 TAG_INIT	535
7.81.3.28 TAG_INTRO	536
7.81.3.29 TAG_PARSING	536
7.81.3.30 TAG_RESULT	536
7.81.3.31 TAG_SEPARATOR	536
7.81.3.32 TAG_SUCCESS	536
7.81.3.33 TAG_TIMEOUT	536
7.81.3.34 TAG_TRUE	537

7.82 multiscale::MultiplicationOperation Class Reference	537
7.82.1 Detailed Description	537
7.82.2 Member Function Documentation	537
7.82.2.1 operator()	537
7.83 multiscale::MultiscaleException Class Reference	538
7.83.1 Detailed Description	540
7.83.2 Constructor & Destructor Documentation	540
7.83.2.1 MultiscaleException	540
7.83.2.2 MultiscaleException	540
7.83.2.3 MultiscaleException	540
7.83.3 Member Function Documentation	540
7.83.3.1 constructExplanatoryString	540
7.83.3.2 rawMessage	541
7.83.3.3 what	541
7.83.4 Member Data Documentation	541
7.83.4.1 explanatoryString	541
7.83.4.2 message	541
7.84 multiscaletest::MultiscaleTest Class Reference	542
7.84.1 Detailed Description	542
7.84.2 Constructor & Destructor Documentation	542
7.84.2.1 ~MultiscaleTest	542
7.84.3 Member Function Documentation	543
7.84.3.1 RunTest	543
7.84.3.2 SetUp	543
7.84.3.3 TearDown	543
7.84.3.4 ValidateTestResults	543
7.84.4 Member Data Documentation	543
7.84.4.1 validationFlag	543
7.85 multiscale::verification::NextKLogicPropertyAttribute Class Reference . . .	543
7.85.1 Detailed Description	544
7.85.2 Member Data Documentation	544
7.85.2.1 logicProperty	544
7.85.2.2 nrOfTimepointsAhead	544
7.86 multiscale::verification::NextLogicPropertyAttribute Class Reference . . .	544

7.86.1	Detailed Description	545
7.86.2	Member Data Documentation	545
7.86.2.1	logicProperty	545
7.87	multiscale::verification::Nil Class Reference	545
7.87.1	Detailed Description	545
7.88	multiscale::verification::NotConstraintAttribute Class Reference	546
7.88.1	Detailed Description	546
7.88.2	Member Data Documentation	546
7.88.2.1	constraint	546
7.89	multiscale::verification::NotLogicPropertyAttribute Class Reference	546
7.89.1	Detailed Description	547
7.89.2	Member Data Documentation	547
7.89.2.1	logicProperty	547
7.90	multiscale::NumberIterator Class Reference	547
7.90.1	Detailed Description	549
7.90.2	Constructor & Destructor Documentation	549
7.90.2.1	NumberIterator	549
7.90.2.2	~NumberIterator	549
7.90.3	Member Function Documentation	549
7.90.3.1	hasNext	549
7.90.3.2	hasNextInitialised	550
7.90.3.3	hasNextNotInitialised	550
7.90.3.4	init	550
7.90.3.5	initialise	550
7.90.3.6	number	550
7.90.3.7	reset	551
7.90.3.8	resetCurrentNumber	551
7.90.4	Member Data Documentation	551
7.90.4.1	isInitialised	551
7.90.4.2	upperBound	551
7.91	multiscale::Numeric Class Reference	552
7.91.1	Detailed Description	558
7.91.2	Member Function Documentation	558
7.91.2.1	almostEqual	558

7.91.2.2	applyOperation	559
7.91.2.3	areOverflowUnderflowFlagsSet	559
7.91.2.4	average	559
7.91.2.5	average	560
7.91.2.6	combinations	560
7.91.2.7	computeCombinations	560
7.91.2.8	computeKurtosisFirstTerm	561
7.91.2.9	computeKurtosisLastTerm	561
7.91.2.10	computeKurtosisMiddleTerm	561
7.91.2.11	computeMode	562
7.91.2.12	computeQuartileValue	562
7.91.2.13	computeSkewFirstTerm	562
7.91.2.14	computeSkewLastTerm	562
7.91.2.15	covariance	563
7.91.2.16	covariance	563
7.91.2.17	factorial	563
7.91.2.18	geometricMean	564
7.91.2.19	geometricMean	564
7.91.2.20	greaterOrEqual	564
7.91.2.21	harmonicMean	565
7.91.2.22	harmonicMean	565
7.91.2.23	isPositive	565
7.91.2.24	kurtosis	566
7.91.2.25	kurtosis	566
7.91.2.26	lessOrEqual	566
7.91.2.27	log	567
7.91.2.28	maximum	567
7.91.2.29	maximum	568
7.91.2.30	maximum	568
7.91.2.31	median	568
7.91.2.32	median	569
7.91.2.33	minimum	569
7.91.2.34	minimum	569
7.91.2.35	mode	569

7.91.2.36 mode	570
7.91.2.37 numberInverse	570
7.91.2.38 percentile	570
7.91.2.39 percentile	571
7.91.2.40 printNoValuesWarningMessage	571
7.91.2.41 product	571
7.91.2.42 product	572
7.91.2.43 quartile	572
7.91.2.44 quartile	572
7.91.2.45 resetOverflowUnderflowFlags	573
7.91.2.46 sign	573
7.91.2.47 skew	573
7.91.2.48 skew	573
7.91.2.49 standardDeviation	574
7.91.2.50 standardDeviation	574
7.91.2.51 sum	574
7.91.2.52 sum	574
7.91.2.53 validateLogBase	575
7.91.2.54 validateLogNumber	575
7.91.2.55 validateLogNumberAndBase	575
7.91.2.56 validatePercentile	576
7.91.2.57 validateQuartile	576
7.91.2.58 variance	576
7.91.2.59 variance	577
7.91.3 Member Data Documentation	577
7.91.3.1 epsilon	577
7.91.3.2 ERR_COMBINATIONS_END	577
7.91.3.3 ERR_COMBINATIONS_MIDDLE	577
7.91.3.4 ERR_COMBINATIONS_START	577
7.91.3.5 ERR_LOG_BASE_END	578
7.91.3.6 ERR_LOG_BASE_START	578
7.91.3.7 ERR_LOG_NUMBER_END	578
7.91.3.8 ERR_LOG_NUMBER_START	578
7.91.3.9 ERR_OVERFLOW_UNDERFLOW	578

7.91.3.10 ERR_PERCENTILE_VALUE_END	578
7.91.3.11 ERR_PERCENTILE_VALUE_START	578
7.91.3.12 ERR_QUARTILE_VALUE_END	579
7.91.3.13 ERR_QUARTILE_VALUE_START	579
7.91.3.14 WRN_AVERAGE_FUNCTION_NAME	579
7.91.3.15 WRN_COVARIANCE_FUNCTION_NAME	579
7.91.3.16 WRN_GEOMETRIC_MEAN_FUNCTION_NAME . .	579
7.91.3.17 WRN_HARMONIC_MEAN_FUNCTION_NAME . .	579
7.91.3.18 WRN_KURTOSIS_FUNCTION_NAME	579
7.91.3.19 WRN_MAXIMUM_FUNCTION_NAME	580
7.91.3.20 WRN_MEDIAN_FUNCTION_NAME	580
7.91.3.21 WRN_MINIMUM_FUNCTION_NAME	580
7.91.3.22 WRN_MODE_FUNCTION_NAME	580
7.91.3.23 WRN_NOT_ENOUGH_VALUES_END	580
7.91.3.24 WRN_NOT_ENOUGH_VALUES_START	580
7.91.3.25 WRN_NUMBER_INVERSE	580
7.91.3.26 WRN_PERCENTILE_FUNCTION_NAME	581
7.91.3.27 WRN_PRODUCT_FUNCTION_NAME	581
7.91.3.28 WRN_QUARTILE_FUNCTION_NAME	581
7.91.3.29 WRN_SKEW_FUNCTION_NAME	581
7.91.3.30 WRN_STANDARD_DEVIATION_FUNCTION_NAME	581
7.91.3.31 WRN_SUM_FUNCTION_NAME	581
7.91.3.32 WRN_VARIANCE_FUNCTION_NAME	581
7.92 multiscale::verification::NumericEvaluator Class Reference	582
7.92.1 Detailed Description	582
7.92.2 Member Function Documentation	583
7.92.2.1 evaluate	583
7.92.2.2 evaluate	583
7.92.2.3 evaluate	583
7.92.2.4 evaluate	584
7.92.2.5 evaluate	584
7.92.2.6 evaluate	585
7.93 multiscale::NumericException Class Reference	585
7.93.1 Detailed Description	588

7.93.2 Constructor & Destructor Documentation	588
7.93.2.1 NumericException	588
7.93.2.2 NumericException	588
7.93.2.3 NumericException	588
7.94 multiscale::verification::NumericMeasureAttribute Class Reference	588
7.94.1 Detailed Description	589
7.94.2 Member Data Documentation	589
7.94.2.1 numericMeasure	589
7.95 multiscale::verification::NumericNumericComparisonAttribute Class Reference	589
7.95.1 Detailed Description	590
7.95.2 Member Data Documentation	591
7.95.2.1 comparator	591
7.95.2.2 numericMeasure	591
7.95.2.3 numericStateVariable	591
7.96 multiscale::NumericRangeManipulator Class Reference	591
7.96.1 Detailed Description	592
7.96.2 Member Function Documentation	592
7.96.2.1 convertFromRange	592
7.97 multiscale::verification::NumericSpatialAttribute Class Reference	592
7.97.1 Detailed Description	592
7.97.2 Member Data Documentation	593
7.97.2.1 numericSpatialMeasure	593
7.98 multiscale::verification::NumericSpatialNumericComparisonAttribute Class Reference	593
7.98.1 Detailed Description	594
7.98.2 Member Data Documentation	594
7.98.2.1 comparator	594
7.98.2.2 numericMeasure	595
7.98.2.3 numericSpatialMeasure	595
7.99 multiscale::verification::NumericStateVariableAttribute Class Reference	595
7.99.1 Detailed Description	597
7.99.2 Member Data Documentation	597
7.99.2.1 stateVariable	597

7.100multiscaletest::NumericStateVariableTraceTest Class Reference	597
7.100.1 Detailed Description	600
7.100.2 Member Function Documentation	600
7.100.2.1 InitialiseTrace	600
7.101multiscale::verification::NumericVisitor Class Reference	600
7.101.1 Detailed Description	602
7.101.2 Constructor & Destructor Documentation	602
7.101.2.1 NumericVisitor	602
7.101.3 Member Function Documentation	603
7.101.3.1 evaluate	603
7.101.3.2 evaluateNumericSpatialMeasure	603
7.101.3.3 evaluatePrimaryNumericMeasure	603
7.101.3.4 operator()	604
7.101.3.5 operator()	604
7.101.3.6 operator()	604
7.101.3.7 operator()	605
7.101.3.8 operator()	605
7.101.3.9 operator()	605
7.101.3.10operator()	606
7.101.3.11operator()	606
7.101.3.12operator()	606
7.101.3.13operator()	607
7.101.3.14operator()	607
7.101.4 Member Data Documentation	607
7.101.4.1 timePoint	607
7.102multiscale::OperatingSystem Class Reference	608
7.102.1 Detailed Description	610
7.102.2 Member Function Documentation	610
7.102.2.1 executeProgram	610
7.102.2.2 executeProgramAndVerifyPath	611
7.102.2.3 executeProgramOSSpecific	611
7.102.2.4 getEnvironmentVariable	611
7.102.3 Member Data Documentation	611
7.102.3.1 ERR_EXECUTE_PROGRAM	611

7.102.3.2 ERR_INVALID_PROGRAM_PATH	612
7.102.3.3 TIMEOUT_MAX_NR_SECONDS	612
7.102.3.4 TIMEOUT_NR_SECONDS	612
7.103 multiscale::verification::OrConstraintAttribute Class Reference	612
7.103.1 Detailed Description	612
7.103.2 Member Data Documentation	612
7.103.2.1 constraint	613
7.104 multiscale::verification::OrLogicPropertyAttribute Class Reference	613
7.104.1 Detailed Description	613
7.104.2 Member Data Documentation	613
7.104.2.1 logicProperty	613
7.105 multiscale::verification::Parser Class Reference	614
7.105.1 Detailed Description	615
7.105.2 Constructor & Destructor Documentation	615
7.105.2.1 Parser	615
7.105.2.2 ~Parser	615
7.105.3 Member Function Documentation	615
7.105.3.1 checkIfErrorCase	615
7.105.3.2 initialise	615
7.105.3.3 isStringParsedCompletely	615
7.105.3.4 parse	616
7.105.3.5 parseLogicalQuery	616
7.105.3.6 parseLogicalQuery	616
7.105.3.7 setLogicalQuery	617
7.105.4 Member Data Documentation	617
7.105.4.1 grammar	617
7.105.4.2 logicalQuery	617
7.105.4.3 logicalQueryEnd	617
7.105.4.4 logicalQueryIterator	617
7.106 multiscale::verification::ParserGrammar< Iterator > Class Template - Reference	618
7.106.1 Detailed Description	626
7.106.2 Constructor & Destructor Documentation	626
7.106.2.1 ParserGrammar	626

7.106.3 Member Function Documentation	626
7.106.3.1 assignNamesToComparatorRules	626
7.106.3.2 assignNamesToComposedConstraintRules	626
7.106.3.3 assignNamesToComposedLogicPropertyRules	627
7.106.3.4 assignNamesToConstraintRules	627
7.106.3.5 assignNamesToConstraintsRules	627
7.106.3.6 assignNamesToFilterNumericMeasureRules	627
7.106.3.7 assignNamesToLogicPropertiesRules	627
7.106.3.8 assignNamesToLogicPropertyRules	628
7.106.3.9 assignNamesToNaryNumericMeasureRules	628
7.106.3.10 assignNamesToNumericMeasureRules	628
7.106.3.11 assignNamesToNumericSpatialMeasureRules	628
7.106.3.12 assignNamesToNumericSpatialSubsetMeasureRules	629
7.106.3.13 assignNamesToNumericStateVariableRules	629
7.106.3.14 assignNamesToPrimaryConstraintRules	629
7.106.3.15 assignNamesToPrimaryLogicPropertyRules	629
7.106.3.16 assignNamesToProbabilisticLogicPropertyRules	629
7.106.3.17 assignNamesToRules	630
7.106.3.18 assignNamesToSpatialMeasureRules	630
7.106.3.19 assignNamesToSubsetRules	630
7.106.3.20 initialise	630
7.106.3.21 initialiseComparatorRuleDebugging	631
7.106.3.22 initialiseComparatorRules	631
7.106.3.23 initialiseComposedConstraintErrorHandlingSupport	631
7.106.3.24 initialiseComposedConstraintRule	631
7.106.3.25 initialiseComposedConstraintRuleDebugging	631
7.106.3.26 initialiseComposedLogicPropertyErrorHandlingSupport	632
7.106.3.27 initialiseComposedLogicPropertyRule	632
7.106.3.28 initialiseComposedLogicPropertyRuleDebugging	632
7.106.3.29 initialiseConstraintRule	632
7.106.3.30 initialiseConstraintRuleDebugging	633
7.106.3.31 initialiseConstraintsErrorHandlingSupport	633
7.106.3.32 initialiseConstraintsRules	633
7.106.3.33 initialiseConstraintsRulesDebugging	633

7.106.3.34initialiseDebugSupport	633
7.106.3.35initialiseErrorHandlerSupport	634
7.106.3.36initialiseFilterNumericMeasureErrorHandlerSupport . .	634
7.106.3.37initialiseFilterNumericMeasureRule	634
7.106.3.38initialiseFilterNumericMeasureRuleDebugging	634
7.106.3.39initialiseGrammar	635
7.106.3.40initialiseLogicPropertiesErrorHandlerSupport	635
7.106.3.41initialiseLogicPropertiesRules	635
7.106.3.42initialiseLogicPropertiesRulesDebugging	635
7.106.3.43initialiseLogicPropertyRule	635
7.106.3.44initialiseLogicPropertyRuleDebugging	636
7.106.3.45initialiseNaryNumericMeasureRule	636
7.106.3.46initialiseNaryNumericMeasureRuleDebugging	636
7.106.3.47initialiseNumericMeasureErrorHandlerSupport	636
7.106.3.48initialiseNumericMeasureRule	637
7.106.3.49initialiseNumericMeasureRuleDebugging	637
7.106.3.50initialiseNumericSpatialMeasureErrorHandlerSupport .	637
7.106.3.51initialiseNumericSpatialMeasureRule	637
7.106.3.52initialiseNumericSpatialMeasureRuleDebugging	637
7.106.3.53initialiseNumericSpatialSubsetMeasureRule	638
7.106.3.54initialiseNumericStateVariableRule	638
7.106.3.55initialiseNumericStateVariableRuleDebugging	638
7.106.3.56initialisePrimaryConstraintErrorHandlerSupport	638
7.106.3.57initialisePrimaryConstraintRule	639
7.106.3.58initialisePrimaryConstraintRuleDebugging	639
7.106.3.59initialisePrimaryLogicPropertyErrorHandlerSupport .	639
7.106.3.60initialisePrimaryLogicPropertyRule	639
7.106.3.61initialisePrimaryLogicPropertyRuleDebugging	639
7.106.3.62initialiseProbabilisticLogicPropertyErrorHandlerSupport	640
7.106.3.63initialiseProbabilisticLogicPropertyRule	640
7.106.3.64initialiseProbabilisticLogicPropertyRuleDebugging	640
7.106.3.65initialiseRulesDebugging	640
7.106.3.66initialiseSpatialMeasureRule	641

7.106.3.67initialiseSpatialMeasureRuleDebugging	641
7.106.3.68initialiseSpatialSubsetMeasureRuleDebugging	641
7.106.3.69initialiseStateVariableErrorHandlingSupport	641
7.106.3.70initialiseSubsetErrorHandlingSupport	641
7.106.3.71initialiseSubsetRule	642
7.106.3.72initialiseSubsetRuleDebugging	642
7.106.4 Member Data Documentation	642
7.106.4.1 andConstraintRule	642
7.106.4.2 andLogicPropertyRule	642
7.106.4.3 binaryNumericFilterRule	643
7.106.4.4 binaryNumericMeasureRule	643
7.106.4.5 binaryNumericMeasureTypeParser	643
7.106.4.6 binaryNumericNumericRule	644
7.106.4.7 binarySubsetMeasureRule	644
7.106.4.8 binarySubsetMeasureTypeParser	644
7.106.4.9 binarySubsetRule	644
7.106.4.10comparatorNonEqualTypeParser	645
7.106.4.11comparatorRule	645
7.106.4.12comparatorTypeParser	645
7.106.4.13constraintRule	646
7.106.4.14differenceRule	646
7.106.4.15equivalenceConstraintRule	646
7.106.4.16equivalenceLogicPropertyRule	647
7.106.4.17filterNumericMeasureRule	647
7.106.4.18filterSubsetRule	647
7.106.4.19futureLogicPropertyRule	648
7.106.4.20globalLogicPropertyRule	648
7.106.4.21implicationConstraintRule	648
7.106.4.22implicationLogicPropertyRule	649
7.106.4.23logicPropertyRule	649
7.106.4.24nextKLogicPropertyRule	649
7.106.4.25nextLogicPropertyRule	650
7.106.4.26notConstraintRule	650
7.106.4.27notLogicPropertyRule	650

7.106.4.28numericMeasureRule	650
7.106.4.29numericNumericComparisonRule	651
7.106.4.30numericSpatialNumericComparisonRule	651
7.106.4.31numericSpatialRule	651
7.106.4.32numericStateVariableRule	652
7.106.4.33orConstraintRule	652
7.106.4.34orLogicPropertyRule	652
7.106.4.35primaryConstraintRule	653
7.106.4.36primaryLogicPropertyRule	653
7.106.4.37primaryNumericMeasureRule	653
7.106.4.38probabilisticLogicPropertyComparatorRule	654
7.106.4.39probabilisticLogicPropertyRule	654
7.106.4.40probabilityRule	654
7.106.4.41quaternarySubsetMeasureRule	655
7.106.4.42quaternarySubsetMeasureTypeParser	655
7.106.4.43quaternarySubsetRule	655
7.106.4.44spatialMeasureRule	655
7.106.4.45spatialMeasureTypeParser	656
7.106.4.46stateVariableNameRule	656
7.106.4.47stateVariableRule	656
7.106.4.48subsetOperationTypeParser	657
7.106.4.49subsetRule	657
7.106.4.50subsetSpecificRule	657
7.106.4.51subsetSpecificTypeParser	657
7.106.4.52subsetSubsetOperationRule	658
7.106.4.53ternarySubsetMeasureRule	658
7.106.4.54ternarySubsetMeasureTypeParser	658
7.106.4.55ternarySubsetRule	658
7.106.4.56unaryNumericFilterRule	659
7.106.4.57unaryNumericMeasureRule	659
7.106.4.58unaryNumericMeasureTypeParser	659
7.106.4.59unaryNumericNumericRule	660
7.106.4.60unarySpatialConstraintRule	660
7.106.4.61unarySubsetMeasureRule	660

7.106.4.62unarySubsetMeasureTypeParser	660
7.106.4.63unarySubsetRule	661
7.106.4.64unaryTypeConstraintRule	661
7.106.4.65untilLogicPropertyRule	661
7.107multiscale::verification::ParserGrammarExceptionHandler Class - Reference	662
7.107.1 Detailed Description	663
7.107.2 Member Function Documentation	663
7.107.2.1 getIntroductoryErrorMessage	663
7.107.2.2 handleExpectedTokenAtEndOfString	663
7.107.2.3 handleExtraInputException	663
7.107.2.4 handleProbabilityException	664
7.107.2.5 handleUnexpectedTokenException	664
7.107.2.6 handleUnexpectedTokenInString	664
7.107.2.7 handleUnparseableInputException	665
7.107.2.8 trimRight	665
7.108multiscale::verification::ParserGrammarExtraInputException Class - Reference	665
7.108.1 Detailed Description	666
7.108.2 Constructor & Destructor Documentation	667
7.108.2.1 ParserGrammarExtraInputException	667
7.108.3 Member Function Documentation	667
7.108.3.1 getErrorString	667
7.108.4 Member Data Documentation	667
7.108.4.1 errorString	667
7.109multiscale::verification::ParserGrammarProbabilityException Class - Reference	667
7.109.1 Detailed Description	669
7.109.2 Constructor & Destructor Documentation	669
7.109.2.1 ParserGrammarProbabilityException	669
7.109.3 Member Function Documentation	669
7.109.3.1 getErrorString	669
7.109.3.2 getExpectedToken	669
7.109.4 Member Data Documentation	669
7.109.4.1 errorString	670

7.109.4.2 expectedToken	670
7.110 multiscale::verification::ParserGrammarUnexpectedTokenException - Class Reference	670
7.110.1 Detailed Description	672
7.110.2 Constructor & Destructor Documentation	672
7.110.2.1 ParserGrammarUnexpectedTokenException	672
7.110.3 Member Function Documentation	672
7.110.3.1 getErrorString	672
7.110.3.2 getExpectedToken	672
7.110.4 Member Data Documentation	672
7.110.4.1 errorString	672
7.110.4.2 expectedToken	673
7.111 multiscale::verification::ParserGrammarUnparseableInputException - Class Reference	673
7.111.1 Detailed Description	674
7.111.2 Constructor & Destructor Documentation	675
7.111.2.1 ParserGrammarUnparseableInputException	675
7.111.3 Member Function Documentation	675
7.111.3.1 getErrorString	675
7.111.4 Member Data Documentation	675
7.111.4.1 errorString	675
7.112 multiscale::video::PolarCsvToInputFilesConverter Class Reference	675
7.112.1 Detailed Description	679
7.112.2 Constructor & Destructor Documentation	679
7.112.2.1 PolarCsvToInputFilesConverter	679
7.112.2.2 ~PolarCsvToInputFilesConverter	679
7.112.3 Member Function Documentation	679
7.112.3.1 computeConcentration	679
7.112.3.2 computeConcentrationWrtArea	680
7.112.3.3 computeNextPositionConcentration	680
7.112.3.4 computeNonScaledConcentration	680
7.112.3.5 computeNormalisedConcentration	681
7.112.3.6 computeScaledConcentration	681
7.112.3.7 computeSimulationTime	681

7.112.3.8 convert	681
7.112.3.9 initInputFile	682
7.112.3.10initIterators	682
7.112.3.11initMaximumConcentration	682
7.112.3.12initOutputFile	682
7.112.3.13processInputFile	683
7.112.3.14processLine	683
7.112.3.15splitFirstPartInConcentrations	683
7.112.3.16splitLineInConcentrations	684
7.112.3.17splitOtherPartsInConcentrations	684
7.112.3.18updateMaximumConcentration	684
7.112.3.19validateInput	685
7.112.3.20validateInputLine	685
7.112.3.21validateSelectedConcentrationIndex	685
7.112.4 Member Data Documentation	685
7.112.4.1 circlesIterator	685
7.112.4.2 concentrationsIndex	686
7.112.4.3 ERR_INPUT_OPEN	686
7.112.4.4 ERR_INVALID_VALUE_LINE	686
7.112.4.5 ERR_INVALID_VALUE_TOKEN	686
7.112.4.6 ERR_NEG_CONCENTRATION	686
7.112.4.7 ERR_NEG_SIM_TIME	686
7.112.4.8 ERR_NR_CONCENTRATIONS	686
7.112.4.9 ERR_SELECTED_CONCENTRATION_INDEX	686
7.112.4.10INPUT_FILE_SEPARATOR	687
7.112.4.11inputFilepath	687
7.112.4.12maximumConcentration	687
7.112.4.13nrOfConcentrationsForPosition	687
7.112.4.14nrOfConcentricCircles	687
7.112.4.15nrOfSectors	687
7.112.4.16OUTPUT_EXTENSION	687
7.112.4.17OUTPUT_FILE_SEPARATOR	688
7.112.4.18OUTPUT_SEPARATOR	688
7.112.4.19outputFilepath	688

7.112.4.20RADIUS_MIN	688
7.112.4.21sectorsIterator	688
7.112.4.22selectedConcentrationIndex	688
7.112.4.23useLogScaling	688
7.113multiscale::video::PolarGnuplotScriptGenerator Class Reference	689
7.113.1 Detailed Description	692
7.113.2 Member Function Documentation	692
7.113.2.1 generateBody	692
7.113.2.2 generateFooter	692
7.113.2.3 generateHeader	692
7.113.2.4 generateScript	693
7.113.2.5 outputContent	693
7.113.2.6 outputFooter	693
7.113.2.7 outputHeader	694
7.113.2.8 readContentTemplate	694
7.113.3 Member Data Documentation	694
7.113.3.1 CONTENT_IN	694
7.113.3.2 FOOTER_IN	694
7.113.3.3 GNUPLOT_EXTENSION	695
7.113.3.4 HEADER_IN	695
7.113.3.5 REPLACE_CONTENT_CONCENTRATION	695
7.113.3.6 REPLACE_CONTENT_END_ANGLE	695
7.113.3.7 REPLACE_CONTENT_INDEX	695
7.113.3.8 REPLACE_CONTENT_RADIUS	695
7.113.3.9 REPLACE_CONTENT_START_ANGLE	695
7.113.3.10REPLACE_HEADER_FILENAME	695
7.113.3.11REPLACE_HEADER_SIM_TIME	696
7.114multiscale::verification::PrimaryConstraintAttribute Class Reference	696
7.114.1 Detailed Description	696
7.114.2 Member Data Documentation	696
7.114.2.1 primaryConstraint	696
7.115multiscale::verification::PrimaryLogicPropertyAttribute Class Reference	697
7.115.1 Detailed Description	697
7.115.2 Member Data Documentation	697

7.115.2.1 primaryLogicProperty	697
7.116 multiscale::verification::PrimaryNumericMeasureAttribute Class - Reference	697
7.116.1 Detailed Description	698
7.116.2 Member Data Documentation	698
7.116.2.1 primaryNumericMeasure	698
7.117 multiscale::verification::ProbabilisticBlackBoxModelChecker Class - Reference	698
7.117.1 Detailed Description	701
7.117.2 Constructor & Destructor Documentation	701
7.117.2.1 ProbabilisticBlackBoxModelChecker	701
7.117.2.2 ~ProbabilisticBlackBoxModelChecker	702
7.117.3 Member Function Documentation	702
7.117.3.1 acceptsMoreTraces	702
7.117.3.2 doesPropertyHold	702
7.117.3.3 getDetailedResults	702
7.117.3.4 requiresMoreTraces	702
7.117.3.5 updateDerivedModelCheckerForFalseEvaluation	703
7.117.3.6 updateDerivedModelCheckerForTrueEvaluation	703
7.118 multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory - Class Reference	703
7.118.1 Detailed Description	705
7.118.2 Constructor & Destructor Documentation	705
7.118.2.1 ProbabilisticBlackBoxModelCheckerFactory	705
7.118.2.2 ~ProbabilisticBlackBoxModelCheckerFactory	706
7.118.3 Member Function Documentation	706
7.118.3.1 createInstance	706
7.119 multiscaletest::ProbabilisticBlackBoxModelCheckerTest Class Reference	706
7.119.1 Detailed Description	708
7.119.2 Member Function Documentation	709
7.119.2.1 InitialiseModelChecker	709
7.120 multiscale::verification::ProbabilisticLogicPropertyAttribute Class - Reference	709
7.120.1 Detailed Description	711
7.120.2 Member Function Documentation	711

7.120.2.1 evaluate	711
7.120.2.2 getComparator	711
7.120.2.3 getProbability	711
7.120.3 Member Data Documentation	712
7.120.3.1 comparator	712
7.120.3.2 ERR_TRACE_LENGTH_ZERO	712
7.120.3.3 evaluationLogicProperty	712
7.120.3.4 logicProperty	712
7.120.3.5 probability	712
7.121 multiscale::verification::ProbabilityErrorHandler Struct Reference	713
7.121.1 Detailed Description	713
7.121.2 Member Function Documentation	713
7.121.2.1 getExpectedTokenAsString	714
7.121.2.2 operator()	714
7.122 multiscale::verification::QuaternarySubsetAttribute Class Reference	714
7.122.1 Detailed Description	715
7.122.2 Member Data Documentation	715
7.122.2.1 firstSpatialMeasure	715
7.122.2.2 firstSubset	716
7.122.2.3 quaternarySubsetMeasure	716
7.122.2.4 secondSpatialMeasure	716
7.122.2.5 secondSubset	716
7.123 multiscale::verification::QuaternarySubsetMeasureAttribute Class - Reference	716
7.123.1 Detailed Description	717
7.123.2 Member Data Documentation	717
7.123.2.1 quaternarySubsetMeasureType	717
7.124 multiscale::verification::QuaternarySubsetMeasureTypeParser Struct Reference	717
7.124.1 Detailed Description	717
7.124.2 Constructor & Destructor Documentation	718
7.124.2.1 QuaternarySubsetMeasureTypeParser	718
7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference	718
7.125.1 Detailed Description	722

7.125.2 Constructor & Destructor Documentation	722
7.125.2.1 <code>RectangularCsvToInputFilesConverter</code>	722
7.125.2.2 <code>~RectangularCsvToInputFilesConverter</code>	722
7.125.3 Member Function Documentation	722
7.125.3.1 <code>computeConcentration</code>	722
7.125.3.2 <code>computeNextPositionConcentration</code>	722
7.125.3.3 <code>computeNonScaledConcentration</code>	723
7.125.3.4 <code>computeNormalisedConcentration</code>	723
7.125.3.5 <code>computeScaledConcentration</code>	723
7.125.3.6 <code>computeSimulationTime</code>	723
7.125.3.7 <code>convert</code>	724
7.125.3.8 <code>initInputFile</code>	724
7.125.3.9 <code>initIterators</code>	724
7.125.3.10 <code>initMaximumConcentration</code>	724
7.125.3.11 <code>initOutputFile</code>	725
7.125.3.12 <code>processInputFile</code>	725
7.125.3.13 <code>processLine</code>	725
7.125.3.14 <code>splitLineInConcentrations</code>	726
7.125.3.15 <code>splitLineInConcentrations</code>	726
7.125.3.16 <code>updateMaximumConcentration</code>	726
7.125.3.17 <code>validateInput</code>	727
7.125.3.18 <code>validateInputLine</code>	727
7.125.3.19 <code>validateSelectedConcentrationIndex</code>	727
7.125.4 Member Data Documentation	727
7.125.4.1 <code>columnsIterator</code>	727
7.125.4.2 <code>concentrationsIndex</code>	728
7.125.4.3 <code>ERR_INPUT_OPEN</code>	728
7.125.4.4 <code>ERR_INVALID_VALUE_LINE</code>	728
7.125.4.5 <code>ERR_INVALID_VALUE_TOKEN</code>	728
7.125.4.6 <code>ERR_NEG_CONCENTRATION</code>	728
7.125.4.7 <code>ERR_NEG_SIM_TIME</code>	728
7.125.4.8 <code>ERR_NR_CONCENTRATIONS</code>	728
7.125.4.9 <code>ERR_SELECTED_CONCENTRATION_INDEX</code>	729
7.125.4.10 <code>height</code>	729

7.125.4.11	INPUT_FILE_SEPARATOR	729
7.125.4.12	inputFilepath	729
7.125.4.13	maximumConcentration	729
7.125.4.14	nrOfConcentrationsForPosition	729
7.125.4.15	OUTPUT_EXTENSION	729
7.125.4.16	OUTPUT_FILE_SEPARATOR	730
7.125.4.17	OUTPUT_SEPARATOR	730
7.125.4.18	outputFilepath	730
7.125.4.19	rowsIterator	730
7.125.4.20	selectedConcentrationIndex	730
7.125.4.21	useLogScaling	730
7.125.4.22	width	730
7.126	multiscale::video::RectangularEntityCsvToInputFilesConverter Class Reference	731
7.126.1	Detailed Description	734
7.126.2	Constructor & Destructor Documentation	734
7.126.2.1	RectangularEntityCsvToInputFilesConverter	734
7.126.2.2	~RectangularEntityCsvToInputFilesConverter	735
7.126.3	Member Function Documentation	735
7.126.3.1	computeCoordinate	735
7.126.3.2	computeSimulationTime	735
7.126.3.3	convert	735
7.126.3.4	initInputFile	735
7.126.3.5	initIterators	736
7.126.3.6	initOutputFile	736
7.126.3.7	processInputFile	736
7.126.3.8	processLine	737
7.126.3.9	splitLineInCoordinates	737
7.126.3.10	validateCoordinate	737
7.126.3.11	validateEntitiesGrid	738
7.126.3.12	validateInput	738
7.126.3.13	validateInputLine	738
7.126.3.14	validateMaxNrOfEntitiesPerPosition	738
7.126.3.15	validateSimulationTime	739

7.126.4 Member Data Documentation	739
7.126.4.1 entitiesIterator	739
7.126.4.2 ERR_INPUT_OPEN	739
7.126.4.3 ERR_INVALID_NR_ENTITIES	739
7.126.4.4 ERR_INVALID_OX_COORDINATE	739
7.126.4.5 ERR_INVALID_OY_COORDINATE	740
7.126.4.6 ERR_INVALID_VALUE_LINE	740
7.126.4.7 ERR_INVALID_VALUE_TOKEN	740
7.126.4.8 ERR_MAX_NR_ENTITIES	740
7.126.4.9 ERR_NEG_SIM_TIME	740
7.126.4.10 ERR_NR_COORDINATES	740
7.126.4.11 height	740
7.126.4.12 INPUT_FILE_SEPARATOR	741
7.126.4.13 inputfilepath	741
7.126.4.14 maxNrOfEntitiesPerPosition	741
7.126.4.15 nrOfEntities	741
7.126.4.16 OUTPUT_EXTENSION	741
7.126.4.17 OUTPUT_FILE_SEPARATOR	741
7.126.4.18 OUTPUT_SEPARATOR	741
7.126.4.19 outputFilepath	742
7.126.4.20 width	742
7.127 multiscale::video::RectangularGnuplotScriptGenerator Class Reference .	742
7.127.1 Detailed Description	744
7.127.2 Member Function Documentation	745
7.127.2.1 generateBody	745
7.127.2.2 generateFooter	745
7.127.2.3 generateHeader	745
7.127.2.4 generateScript	746
7.127.2.5 outputContent	746
7.127.2.6 outputFooter	746
7.127.2.7 outputHeader	747
7.127.3 Member Data Documentation	747
7.127.3.1 CONTENT_IN	747
7.127.3.2 FOOTER_IN	747

7.127.3.3 GNUMPLOT_EXTENSION	747
7.127.3.4 HEADER_IN	747
7.127.3.5 OUTPUT_SEPARATOR	748
7.127.3.6 REPLACE_DIMENSION_EXTRA	748
7.127.3.7 REPLACE_HEADER_FILENAME	748
7.127.3.8 REPLACE_HEADER_HEIGHT	748
7.127.3.9 REPLACE_HEADER_SIM_TIME	748
7.127.3.10 REPLACE_HEADER_WIDTH	748
7.128 multiscale::analysis::RectangularMatFactory Class Reference	749
7.128.1 Detailed Description	752
7.128.2 Constructor & Destructor Documentation	752
7.128.2.1 RectangularMatFactory	752
7.128.2.2 ~RectangularMatFactory	753
7.128.3 Member Function Documentation	753
7.128.3.1 createFromViewerImage	753
7.128.3.2 isValidViewerImage	753
7.128.3.3 maxColourBarIntensityFromViewerImage	753
7.128.3.4 processConcentrations	754
7.128.4 Member Data Documentation	754
7.128.4.1 COLOURBAR_MAX_X	754
7.128.4.2 COLOURBAR_MAX_Y	754
7.128.4.3 ERR_CONC	755
7.128.4.4 INPUT_IMG_HEIGHT	755
7.128.4.5 INPUT_IMG_WIDTH	755
7.128.4.6 ROI_HEIGHT	755
7.128.4.7 ROI_START_X	755
7.128.4.8 ROI_START_Y	755
7.128.4.9 ROI_WIDTH	755
7.129 multiscale::analysis::Region Class Reference	756
7.129.1 Detailed Description	760
7.129.2 Constructor & Destructor Documentation	760
7.129.2.1 Region	760
7.129.2.2 ~Region	760
7.129.3 Member Function Documentation	761

7.129.3.1 areValidInputPolygons	761
7.129.3.2 areValidInputPolygons	761
7.129.3.3 areValidInputValues	761
7.129.3.4 computeArealfOuterBoderDefined	762
7.129.3.5 computeClusterednessDegreelfOuterBorderDefined	762
7.129.3.6 getInnerBorderPolygons	762
7.129.3.7 getOuterBorderPolygon	763
7.129.3.8 isCircularMeasure	763
7.129.3.9 isRectangularMeasure	763
7.129.3.10isTriangularMeasure	763
7.129.3.11isValidInputPolygon	763
7.129.3.12type	764
7.129.3.13updateArea	764
7.129.3.14updateCentrePoint	764
7.129.3.15updateClusterednessDegree	764
7.129.3.16updateDensity	765
7.129.3.17updatePerimeter	765
7.129.3.18validateInputValues	765
7.129.4 Member Data Documentation	766
7.129.4.1 CONTOUR_CLOSED	766
7.129.4.2 CONTOUR_ORIENTED	766
7.129.4.3 innerBorderPolygons	766
7.129.4.4 outerBorderPolygon	766
7.130multiscale::verification::Region Class Reference	766
7.130.1 Detailed Description	770
7.131multiscale::analysis::RegionDetector Class Reference	770
7.131.1 Detailed Description	777
7.131.2 Constructor & Destructor Documentation	777
7.131.2.1 RegionDetector	777
7.131.2.2 ~RegionDetector	777
7.131.3 Member Function Documentation	777
7.131.3.1 approximatePolygonOuterBorder	777
7.131.3.2 changeContrastAndBrightness	777
7.131.3.3 clearPreviousDetectionResults	778

7.131.3.4 computeAverageClusterednessDegree	778
7.131.3.5 computeAverageDensity	778
7.131.3.6 computeAverageMeasures	779
7.131.3.7 convertAlpha	779
7.131.3.8 convertBeta	779
7.131.3.9 createDetectorSpecificTrackbars	779
7.131.3.10 createPolygon	780
7.131.3.11 createPolygons	780
7.131.3.12 createPolygonsFromContours	781
7.131.3.13 createRegionFromPolygon	781
7.131.3.14 existContours	781
7.131.3.15 findPolygonsInImage	782
7.131.3.16 findRegions	782
7.131.3.17 getAlpha	782
7.131.3.18 getBeta	782
7.131.3.19 getBlurKernelSize	783
7.131.3.20 getCollectionOfSpatialEntityPseudo3D	783
7.131.3.21 getDetectorTypeAsString	783
7.131.3.22 getEpsilon	783
7.131.3.23 getMorphologicalCloselterations	783
7.131.3.24 getOriginXCoordinate	784
7.131.3.25 getOriginYCoordinate	784
7.131.3.26 getRegionAreaThresh	784
7.131.3.27 getRegions	784
7.131.3.28 getThresholdValue	784
7.131.3.29 initialiseDetectorSpecificFields	784
7.131.3.30 initialiseDetectorSpecificImageDependentFields	785
7.131.3.31 isValidContour	785
7.131.3.32 isValideHole	785
7.131.3.33 morphologicalClose	785
7.131.3.34 outputRegionInnerBordersToImage	786
7.131.3.35 outputRegionOuterBorderToImage	786
7.131.3.36 outputRegionToImage	786
7.131.3.37 outputResultsToImage	787

7.131.3.38processImageAndDetect	787
7.131.3.39regionDensity	787
7.131.3.40setAlpha	788
7.131.3.41setBeta	788
7.131.3.42setBlurKernelSize	788
7.131.3.43setEpsilon	789
7.131.3.44setMorphologicalCloselterations	789
7.131.3.45setOriginXCoordinate	789
7.131.3.46setOriginYCoordinate	790
7.131.3.47setPolygonInnerContours	790
7.131.3.48setPolygonOuterContour	790
7.131.3.49setRegionAreaThresh	791
7.131.3.50setThresholdValue	791
7.131.3.51smoothImage	791
7.131.3.52sumOfAverageCentroidDistances	792
7.131.3.53thresholdImage	792
7.131.4 Member Data Documentation	792
7.131.4.1 alpha	792
7.131.4.2 ALPHA_MAX	792
7.131.4.3 ALPHA_REAL_MAX	793
7.131.4.4 ALPHA_REAL_MIN	793
7.131.4.5 beta	793
7.131.4.6 BETA_MAX	793
7.131.4.7 BETA_REAL_MAX	793
7.131.4.8 BETA_REAL_MIN	793
7.131.4.9 blurKernelSize	793
7.131.4.10CANNY_THRESH_MAX	794
7.131.4.11CONTOUR_AREA_ORIENTED	794
7.131.4.12DETECTOR_TYPE	794
7.131.4.13DISPLAY_LINE_THICKNESS	794
7.131.4.14epsilon	794
7.131.4.15EPSILON_MAX	794
7.131.4.16HIERARCHY_FIRST_CHILD_INDEX	795
7.131.4.17HIERARCHY_NEXT_INDEX	795

7.131.4.18	HIERARCHY_PARENT_INDEX	795
7.131.4.19	HIERARCHY_PREV_INDEX	795
7.131.4.20	INTENSITY_MAX	795
7.131.4.21	KERNEL_MAX	795
7.131.4.22	MORPH_ITER_MAX	795
7.131.4.23	morphologicalCloselterations	796
7.131.4.24	POLYGON_CLOSED	796
7.131.4.25	REGION_AREA_THRESH_MAX	796
7.131.4.26	regionAreaThresh	796
7.131.4.27	regions	796
7.131.4.28	THRESHOLD_CLUSTEREDNESS	797
7.131.4.29	THRESHOLD_HOLE_AREA	797
7.131.4.30	THRESHOLD_MAX	797
7.131.4.31	thresholdValue	797
7.131.4.32	TRACKBAR_ALPHA	797
7.131.4.33	TRACKBAR_BETA	797
7.131.4.34	TRACKBAR_CANNY	797
7.131.4.35	TRACKBAR_EPSILON	798
7.131.4.36	TRACKBAR_KERNEL	798
7.131.4.37	TRACKBAR_MORPH	798
7.131.4.38	TRACKBAR_REGION_AREA_THRESH	798
7.131.4.39	TRACKBAR_THRESHOLD	798
7.131.4.40	USE_CANNY_L2	798
7.132	multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename > Struct Template Reference	799
7.132.1	Detailed Description	799
7.132.2	Member Typedef Documentation	799
7.132.2.1	type	799
7.133	multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename > Struct Template Reference	799
7.133.1	Detailed Description	800
7.133.2	Member Typedef Documentation	800
7.133.2.1	type	800
7.134	multiscale::RGBColourGenerator Class Reference	800

7.134.1 Detailed Description	801
7.134.2 Member Function Documentation	801
7.134.2.1 computeRGBValues	801
7.134.2.2 convertHSVToRGB	802
7.134.2.3 convertRGBToString	802
7.134.2.4 generate	802
7.134.2.5 generate	802
7.134.3 Member Data Documentation	803
7.134.3.1 blue	803
7.134.3.2 green	803
7.134.3.3 HUE_MAX	803
7.134.3.4 HUE_MIN	803
7.134.3.5 red	803
7.134.3.6 SATURATION	803
7.134.3.7 VALUE	803
7.135 multiscale::RuntimeException Class Reference	804
7.135.1 Detailed Description	807
7.135.2 Constructor & Destructor Documentation	807
7.135.2.1 RuntimeException	807
7.135.2.2 RuntimeException	807
7.135.2.3 RuntimeException	807
7.136 multiscale::analysis::Silhouette Class Reference	807
7.136.1 Detailed Description	808
7.136.2 Member Function Documentation	808
7.136.2.1 computeAverageDissimilarityBtwEntityAndCluster	808
7.136.2.2 computeAverageDissimilarityToOtherClusters	809
7.136.2.3 computeAverageDissimilarityWithinCluster	809
7.136.2.4 computeAverageMeasure	809
7.136.2.5 computeMeasure	810
7.136.2.6 computeOverallAverageMeasure	810
7.137 multiscale::analysis::SimulationClusterDetector Class Reference	810
7.137.1 Detailed Description	814
7.137.2 Constructor & Destructor Documentation	814
7.137.2.1 SimulationClusterDetector	814

7.137.2.2 ~SimulationClusterDetector	814
7.137.3 Member Function Documentation	814
7.137.3.1 computePileUpDegreeAtPosition	814
7.137.3.2 detectEntitiesInImage	815
7.137.3.3 getEntityCentrePoint	815
7.137.3.4 getEntityContourPoints	815
7.137.3.5 initialiseDetectorSpecificImageDependentFields . . .	816
7.137.3.6 initialiseThresholdedImage	816
7.137.3.7 isEntityAtPosition	816
7.137.3.8 outputClusterCircularShape	817
7.137.3.9 outputClusterRectangularShape	817
7.137.3.10outputClusterShape	817
7.137.3.11outputClusterTolImage	818
7.137.3.12outputClusterTriangularShape	818
7.137.3.13outputResultsTolImage	818
7.137.4 Member Data Documentation	819
7.137.4.1 DATAPOINT_THICKNESS	819
7.137.4.2 DATAPOINT_WIDTH	819
7.137.4.3 ENTITY_THRESH	819
7.137.4.4 entityHeight	819
7.137.4.5 entityWidth	819
7.137.4.6 height	820
7.137.4.7 THRESHOLD	820
7.137.4.8 THRESHOLD_MAX	820
7.137.4.9 thresholdedImage	820
7.137.4.10width	820
7.138multiscaletest::SpatialEntitiesTraceTest Class Reference	821
7.138.1 Detailed Description	823
7.138.2 Member Function Documentation	823
7.138.2.1 InitialiseTrace	823
7.139multiscale::verification::SpatialEntity Class Reference	823
7.139.1 Detailed Description	828
7.139.2 Constructor & Destructor Documentation	828
7.139.2.1 SpatialEntity	828

7.139.2.2 ~SpatialEntity	828
7.139.3 Member Function Documentation	828
7.139.3.1 getAngle	828
7.139.3.2 getArea	828
7.139.3.3 getCentroidX	828
7.139.3.4 getCentroidY	829
7.139.3.5 getCircleMeasure	829
7.139.3.6 getClusteredness	829
7.139.3.7 getDensity	829
7.139.3.8 getDistanceFromOrigin	829
7.139.3.9 getPerimeter	830
7.139.3.10 getRectangleMeasure	830
7.139.3.11 getTriangleMeasure	830
7.139.3.12 setType	830
7.139.3.13 operator<	830
7.139.3.14 setAngle	831
7.139.3.15 setArea	831
7.139.3.16 setCentroidX	831
7.139.3.17 setCentroidY	831
7.139.3.18 setCircleMeasure	831
7.139.3.19 setClusteredness	832
7.139.3.20 setDensity	832
7.139.3.21 setDistanceFromOrigin	832
7.139.3.22 setPerimeter	832
7.139.3.23 setRectangleMeasure	832
7.139.3.24 setTriangleMeasure	833
7.139.3.25 setType	833
7.139.3.26 validateAngleValue	833
7.139.3.27 validateRealNonNegativeValue	833
7.139.3.28 validateRealValueBtwZeroAndOne	834
7.139.4 Member Data Documentation	834
7.139.4.1 angle	834
7.139.4.2 area	834
7.139.4.3 centroidX	834

7.139.4.4 centroidY	835
7.139.4.5 circleMeasure	835
7.139.4.6 clusteredness	835
7.139.4.7 density	835
7.139.4.8 distanceFromOrigin	835
7.139.4.9 ERR_ANGLE_VALUE	835
7.139.4.10ERR_REAL_BTW_ZERO_AND_ONE	836
7.139.4.11ERR_REAL_NON_NEGATIVE_VALUE	836
7.139.4.12ERR_SUFFIX	836
7.139.4.13perimeter	836
7.139.4.14rectangleMeasure	836
7.139.4.15triangleMeasure	836
7.139.4.16type	837
7.140multiscale::verification::SpatialEntityEvaluator Class Reference	837
7.140.1 Detailed Description	837
7.140.2 Member Function Documentation	837
7.140.2.1 evaluate	837
7.141multiscale::analysis::SpatialEntityPseudo3D Class Reference	838
7.141.1 Detailed Description	843
7.141.2 Constructor & Destructor Documentation	843
7.141.2.1 SpatialEntityPseudo3D	843
7.141.2.2 ~SpatialEntityPseudo3D	843
7.141.3 Member Function Documentation	844
7.141.3.1 convertPoints	844
7.141.3.2 fieldNamesToString	844
7.141.3.3 fieldValuesToString	844
7.141.3.4 getAngle	844
7.141.3.5 getArea	844
7.141.3.6 getCentre	845
7.141.3.7 getCircularMeasure	845
7.141.3.8 getClusterednessDegree	845
7.141.3.9 getDensity	845
7.141.3.10getDistanceFromOrigin	845
7.141.3.11getPerimeter	846

7.141.3.12getRectangularMeasure	846
7.141.3.13getShape	846
7.141.3.14getShapeAsString	846
7.141.3.15getTriangularMeasure	846
7.141.3.16initialise	846
7.141.3.17isCircularMeasure	847
7.141.3.18isRectangularMeasure	847
7.141.3.19isTriangularMeasure	847
7.141.3.20normalisedShapeMeasure	847
7.141.3.21shapeAsString	848
7.141.3.22toString	848
7.141.3.23type	848
7.141.3.24typeAsString	848
7.141.3.25updateArea	848
7.141.3.26updateCentrePoint	849
7.141.3.27updateClusterednessDegree	849
7.141.3.28updateDensity	849
7.141.3.29updateMeasures	849
7.141.3.30updateMeasuresIfRequired	849
7.141.3.31updatePerimeter	850
7.141.3.32updateShape	850
7.141.4 Member Data Documentation	850
7.141.4.1 angle	850
7.141.4.2 area	850
7.141.4.3 centre	851
7.141.4.4 circularMeasure	851
7.141.4.5 clusterednessDegree	851
7.141.4.6 CONVEX_HULL_CLOCKWISE	851
7.141.4.7 density	851
7.141.4.8 distanceFromOrigin	852
7.141.4.9 ERR_INPUT	852
7.141.4.10ERR_UNDEFINED_TYPE	852
7.141.4.11OUTPUT_SEPARATOR	852
7.141.4.12perimeter	852

7.141.4.13rectangularMeasure	853
7.141.4.14shape	853
7.141.4.15STR_CIRCLE	853
7.141.4.16STR_CLUSTER	853
7.141.4.17STR_RECTANGLE	853
7.141.4.18STR_REGION	853
7.141.4.19STR_TRIANGLE	854
7.141.4.20STR_UNDEFINED	854
7.141.4.21triangularMeasure	854
7.141.4.22updateFlag	854
7.142multiscale::verification::SpatialMeasureAttribute Class Reference . . .	854
7.142.1 Detailed Description	855
7.142.2 Member Data Documentation	855
7.142.2.1 spatialMeasureType	855
7.143multiscale::verification::SpatialMeasureEvaluator Class Reference . . .	855
7.143.1 Detailed Description	855
7.143.2 Member Function Documentation	856
7.143.2.1 evaluate	856
7.144multiscale::verification::SpatialMeasureTypeParser Struct Reference . .	856
7.144.1 Detailed Description	857
7.144.2 Constructor & Destructor Documentation	857
7.144.2.1 SpatialMeasureTypeParser	857
7.145multiscale::verification::SpatialNumericComparisonAttribute Class - Reference	857
7.145.1 Detailed Description	858
7.145.2 Member Data Documentation	858
7.145.2.1 comparator	858
7.145.2.2 numericMeasure	858
7.145.2.3 spatialMeasure	858
7.146multiscale::verification::SpatialTemporalDataReader Class Reference . .	858
7.146.1 Detailed Description	862
7.146.2 Constructor & Destructor Documentation	863
7.146.2.1 SpatialTemporalDataReader	863
7.146.2.2 ~SpatialTemporalDataReader	863

7.146.3 Member Function Documentation	863
7.146.3.1 addEntitiesToTimePoint	863
7.146.3.2 addNumericStateVariableToTimePoint	863
7.146.3.3 addSpatialEntityToTimePoint	864
7.146.3.4 addTimePointToTrace	864
7.146.3.5 clearInputFilesSets	864
7.146.3.6 constructSpatialTemporalTrace	865
7.146.3.7 constructSpatialTemporalTrace	865
7.146.3.8 convertTimePointPropertyTreeToTrace	865
7.146.3.9 createDerivedSpatialEntity	866
7.146.3.10 generateSpatialTemporalTrace	866
7.146.3.11 generateSpatialTemporalTrace	866
7.146.3.12 getFilesInFolder	867
7.146.3.13 getFirstValidUnprocessedInputFilepath	867
7.146.3.14 getNextSpatialTemporalTrace	867
7.146.3.15 getNextSpatialTemporalTrace	867
7.146.3.16 getRandomValidUnprocessedInputFilepath	868
7.146.3.17 hasNext	868
7.146.3.18 hasValidNext	868
7.146.3.19 initialise	868
7.146.3.20 initialise	869
7.146.3.21 isValidInputFile	869
7.146.3.22 refresh	869
7.146.3.23 setSpatialEntityValues	870
7.146.3.24 setTimePointValue	870
7.146.3.25 timePointHasValue	870
7.146.3.26 updateInputFilesSets	871
7.146.3.27 validateFolderPath	871
7.146.4 Member Data Documentation	871
7.146.4.1 ERR_INVALID_FOLDER_PATH	871
7.146.4.2 ERR_NO_VALID_INPUT_FILES_REMAINING	871
7.146.4.3 ERR_UNDEFINED_SPATIAL_ENTITY_TYPE	872
7.146.4.4 folderPath	872
7.146.4.5 INPUT_FILES_EXTENSION	872

7.146.4.6 INPUT_FILES_SCHEMA_PATH	872
7.146.4.7 LABEL_EXPERIMENT	872
7.146.4.8 LABEL_NUMERIC_STATE_VARIABLE	872
7.146.4.9 LABEL_NUMERIC_STATE_VARIABLE_NAME	873
7.146.4.10LABEL_NUMERIC_STATE_VARIABLE_VALUE	873
7.146.4.11LABEL_SPATIAL_ENTITY	873
7.146.4.12LABEL_SPATIAL_ENTITY_ANGLE	873
7.146.4.13LABEL_SPATIAL_ENTITY_AREA	873
7.146.4.14LABEL_SPATIAL_ENTITY_CENTROID_X	873
7.146.4.15LABEL_SPATIAL_ENTITY_CENTROID_Y	874
7.146.4.16LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE	874
7.146.4.17LABEL_SPATIAL_ENTITY_CLUSTEREDNESS	874
7.146.4.18LABEL_SPATIAL_ENTITY_DENSITY	874
7.146.4.19LABEL_SPATIAL_ENTITY_DISTANCE_FROM_OR- IGIN	874
7.146.4.20LABEL_SPATIAL_ENTITY_PERIMETER	874
7.146.4.21LABEL_SPATIAL_ENTITY_PSEUDO3D_TYPE	875
7.146.4.22LABEL_SPATIAL_ENTITY_RECTANGLE_MEASU- RE	875
7.146.4.23LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE . .	875
7.146.4.24LABEL_TIMEPOINT_VALUE	875
7.146.4.25processedInputFiles	875
7.146.4.26PSEUDO3D_SPATIAL_ENTITY_TYPE_CLUSTER . .	875
7.146.4.27PSEUDO3D_SPATIAL_ENTITY_TYPE_REGION . .	876
7.146.4.28unprocessedInputFiles	876
7.147multiscale::verification::SpatialTemporalException Class Reference	876
7.147.1 Detailed Description	879
7.147.2 Constructor & Destructor Documentation	879
7.147.2.1 SpatialTemporalException	879
7.147.2.2 SpatialTemporalException	879
7.148multiscale::verification::SpatialTemporalTrace Class Reference	879
7.148.1 Detailed Description	882
7.148.2 Constructor & Destructor Documentation	882
7.148.2.1 SpatialTemporalTrace	882

7.148.2.2 SpatialTemporalTrace	882
7.148.2.3 ~SpatialTemporalTrace	882
7.148.3 Member Function Documentation	883
7.148.3.1 addTimePoint	883
7.148.3.2 addTimePointsToSubTrace	883
7.148.3.3 clear	883
7.148.3.4 getTimePoint	883
7.148.3.5 indexOfFirstTimePointGreaterOrEqualToValue	884
7.148.3.6 initialise	884
7.148.3.7 length	884
7.148.3.8 resetSubTraceStartIndex	885
7.148.3.9 setSubTrace	885
7.148.3.10 setSubTraceIndex	885
7.148.3.11 subTrace	885
7.148.3.12 updateLastTimePointValue	886
7.148.3.13 validateIndex	886
7.148.3.14 validateTimePointValue	886
7.148.3.15 validateTimePointValue	887
7.148.3.16 validateValue	887
7.148.4 Member Data Documentation	887
7.148.4.1 beginIndex	887
7.148.4.2 ERR_ITERATOR_NEXT	888
7.148.4.3 ERR_TIMEPOINT_END_END	888
7.148.4.4 ERR_TIMEPOINT_END_MIDDLE	888
7.148.4.5 ERR_TIMEPOINT_END_START	888
7.148.4.6 ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_END	888
7.148.4.7 ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_START	888
7.148.4.8 ERR_TIMEPOINT_VALUE_INVALID_END	888
7.148.4.9 ERR_TIMEPOINT_VALUE_INVALID_MIDDLE	889
7.148.4.10 ERR_TIMEPOINT_VALUE_INVALID_START	889
7.148.4.11 ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_END	889
7.148.4.12 ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_START	889

7.148.4.13sLastTimePointValueInitialised	889
7.148.4.14lastTimePointValue	889
7.148.4.15TIMEPOINT_INDEX_NOT_FOUND	890
7.148.4.16timePoints	890
7.149multiscale::StandardNumberIterator Class Reference	890
7.149.1 Detailed Description	893
7.149.2 Constructor & Destructor Documentation	893
7.149.2.1 StandardNumberIterator	893
7.149.2.2 ~StandardNumberIterator	893
7.149.3 Member Function Documentation	893
7.149.3.1 hasNextInitialised	893
7.149.3.2 initialise	894
7.149.3.3 number	894
7.149.3.4 resetCurrentNumber	894
7.149.4 Member Data Documentation	894
7.149.4.1 currentNumber	894
7.150multiscale::verification::StateVariableAttribute Class Reference	895
7.150.1 Detailed Description	895
7.150.2 Member Data Documentation	896
7.150.2.1 name	896
7.151multiscale::verification::StatisticalModelChecker Class Reference	896
7.151.1 Detailed Description	901
7.151.2 Constructor & Destructor Documentation	901
7.151.2.1 StatisticalModelChecker	901
7.151.2.2 ~StatisticalModelChecker	902
7.151.3 Member Function Documentation	902
7.151.3.1 acceptsMoreTraces	902
7.151.3.2 computeFPrimeValue	902
7.151.3.3 computeFPrimeValueFirstTerm	902
7.151.3.4 computeFPrimeValueSecondTerm	903
7.151.3.5 computeFValue	903
7.151.3.6 computeFValueFirstTerm	903
7.151.3.7 computeFValueSecondTerm	903
7.151.3.8 computeIndifferenceIntervalHalf	904

7.151.3.9 doesPropertyHold	904
7.151.3.10doesPropertyHoldConsideringProbabilityComparator	904
7.151.3.11doesPropertyHoldConsideringResult	905
7.151.3.12getDetailedResults	905
7.151.3.13getDetailedUpdatedResults	905
7.151.3.14initialise	905
7.151.3.15isValidTypeError	906
7.151.3.16requiresMoreTraces	906
7.151.3.17updateDerivedModelCheckerForFalseEvaluation	906
7.151.3.18updateDerivedModelCheckerForTrueEvaluation	906
7.151.3.19updateInitialisedModelCheckingResult	907
7.151.3.20updateModelCheckingResult	907
7.151.3.21updateModelCheckingResult	907
7.151.3.22updateModelCheckingResultEnoughTraces	908
7.151.3.23updateModelCheckingResultNotEnoughTraces	908
7.151.3.24validateTypesErrors	908
7.151.4 Member Data Documentation	909
7.151.4.1 a1FromPaper	909
7.151.4.2 a2FromPaper	909
7.151.4.3 b1FromPaper	909
7.151.4.4 b2FromPaper	909
7.151.4.5 ERR_TYPES_ERROR_VALUES_BEGIN	909
7.151.4.6 ERR_TYPES_ERROR_VALUES_END	910
7.151.4.7 ERR_TYPES_ERROR_VALUES_MIDDLE	910
7.151.4.8 ERR_UNEXPECTED_MODEL_CHECKING_RESULT	910
7.151.4.9 INDIFFERENCE_INTERVAL_HALF_K	910
7.151.4.10indifferenceIntervalHalf	910
7.151.4.11LOGARITHM_ZERO_VALUE	910
7.151.4.12minTypesError	911
7.151.4.13modelCheckingResult	911
7.151.4.14MSG_OUTPUT_MORE_TRACES_REQUIRED	911
7.151.4.15MSG_OUTPUT_RESULT_BEGIN	911
7.151.4.16MSG_OUTPUT_RESULT_END	912

7.151.4.17MSG_OUTPUT_RESULT_MIDDLE	912
7.151.4.18MSG_OUTPUT_SEPARATOR	912
7.151.4.19probability	912
7.151.4.20typeIError	912
7.151.4.21typeIIError	912
7.152multiscale::verification::StatisticalModelCheckerFactory Class Reference	913
7.152.1 Detailed Description	914
7.152.2 Constructor & Destructor Documentation	915
7.152.2.1 StatisticalModelCheckerFactory	915
7.152.2.2 ~StatisticalModelCheckerFactory	915
7.152.3 Member Function Documentation	915
7.152.3.1 createInstance	915
7.152.4 Member Data Documentation	915
7.152.4.1 typeIError	915
7.152.4.2 typeIIError	915
7.153multiscaletest::StatisticalModelCheckerTest Class Reference	916
7.153.1 Detailed Description	919
7.153.2 Constructor & Destructor Documentation	919
7.153.2.1 StatisticalModelCheckerTest	919
7.153.3 Member Function Documentation	919
7.153.3.1 InitialiseModelChecker	919
7.153.3.2 SetTypeIError	919
7.153.3.3 SetTypeIIError	920
7.153.4 Member Data Documentation	920
7.153.4.1 typeIError	920
7.153.4.2 typeIIError	920
7.154multiscale::StringManipulator Class Reference	920
7.154.1 Detailed Description	921
7.154.2 Member Function Documentation	921
7.154.2.1 filenameFromPath	921
7.154.2.2 replace	921
7.154.2.3 split	922
7.154.2.4 toString	922
7.154.2.5 trimRight	923

7.154.2.6 trimRight	923
7.154.3 Member Data Documentation	923
7.154.3.1 DIR_SEPARATOR	923
7.155 multiscale::verification::SubsetAttribute Class Reference	924
7.155.1 Detailed Description	924
7.155.2 Member Data Documentation	924
7.155.2.1 subset	924
7.156 multiscale::verification::SubsetOperationAttribute Class Reference	924
7.156.1 Detailed Description	925
7.156.2 Member Data Documentation	925
7.156.2.1 subsetOperationType	925
7.157 multiscale::verification::SubsetOperationTypeParser Struct Reference	925
7.157.1 Detailed Description	925
7.157.2 Constructor & Destructor Documentation	926
7.157.2.1 SubsetOperationTypeParser	926
7.158 multiscale::verification::SubsetSpecificAttribute Class Reference	926
7.158.1 Detailed Description	926
7.158.2 Member Data Documentation	926
7.158.2.1 subsetSpecificType	926
7.159 multiscale::verification::SubsetSpecificTypeParser Struct Reference	927
7.159.1 Detailed Description	927
7.159.2 Constructor & Destructor Documentation	927
7.159.2.1 SubsetSpecificTypeParser	927
7.160 multiscale::verification::SubsetSubsetOperationAttribute Class Reference	927
7.160.1 Detailed Description	928
7.160.2 Member Data Documentation	928
7.160.2.1 firstSubset	928
7.160.2.2 secondSubset	929
7.160.2.3 subsetOperation	929
7.161 multiscale::verification::SubsetVisitor Class Reference	929
7.161.1 Detailed Description	931
7.161.2 Constructor & Destructor Documentation	931
7.161.2.1 SubsetVisitor	931
7.161.3 Member Function Documentation	931

7.161.3.1 evaluate	931
7.161.3.2 evaluateSubsetOperation	932
7.161.3.3 filterTimePoint	932
7.161.3.4 operator()	932
7.161.3.5 operator()	933
7.161.3.6 operator()	933
7.161.3.7 operator()	933
7.161.3.8 setTimePointConsideredSpatialEntityType	934
7.161.4 Member Data Documentation	934
7.161.4.1 timePoint	934
7.162 multiscale::SubtractionOperation Class Reference	934
7.162.1 Detailed Description	935
7.162.2 Member Function Documentation	935
7.162.2.1 operator()	935
7.163 multiscale::verification::TernarySubsetAttribute Class Reference	935
7.163.1 Detailed Description	936
7.163.2 Member Data Documentation	936
7.163.2.1 parameter	936
7.163.2.2 spatialMeasure	936
7.163.2.3 subset	937
7.163.2.4 ternarySubsetMeasure	937
7.164 multiscale::verification::TernarySubsetMeasureAttribute Class Reference	937
7.164.1 Detailed Description	937
7.164.2 Member Data Documentation	937
7.164.2.1 ternarySubsetMeasureType	938
7.165 multiscale::verification::TernarySubsetMeasureTypeParser Struct - Reference	938
7.165.1 Detailed Description	938
7.165.2 Constructor & Destructor Documentation	938
7.165.2.1 TernarySubsetMeasureTypeParser	938
7.166 multiscale::TestException Class Reference	939
7.166.1 Detailed Description	942
7.166.2 Constructor & Destructor Documentation	942
7.166.2.1 TestException	942

7.166.2.2 TestException	942
7.166.2.3 TestException	942
7.167 multiscale::verification::TimePoint Class Reference	942
7.167.1 Detailed Description	945
7.167.2 Constructor & Destructor Documentation	945
7.167.2.1 TimePoint	946
7.167.2.2 TimePoint	946
7.167.2.3 ~TimePoint	946
7.167.3 Member Function Documentation	946
7.167.3.1 addNumericStateVariable	946
7.167.3.2 addSpatialEntity	946
7.167.3.3 avgClusteredness	947
7.167.3.4 avgDensity	947
7.167.3.5 avgDensity	947
7.167.3.6 avgDistanceBetweenCentroids	947
7.167.3.7 existsNumericStateVariable	948
7.167.3.8 getConsideredSpatialEntities	948
7.167.3.9 getConsideredSpatialEntityTypes	948
7.167.3.10 getNumericStateVariable	949
7.167.3.11 getSpatialEntitiesBeginIterator	949
7.167.3.12 getSpatialEntitiesBeginIterator	949
7.167.3.13 getSpatialEntitiesEndIterator	950
7.167.3.14 getSpatialEntitiesEndIterator	950
7.167.3.15 getValue	951
7.167.3.16 numberOfSpatialEntities	951
7.167.3.17 removeSpatialEntity	951
7.167.3.18 setConsideredSpatialEntityType	951
7.167.3.19 setValue	952
7.167.3.20 spatialEntitiesSetOperation	952
7.167.3.21 timePointDifference	953
7.167.3.22 timePointIntersection	953
7.167.3.23 timePointSetOperation	954
7.167.3.24 timePointUnion	954
7.167.3.25 updateConsideredSpatialEntityTypes	954

7.167.3.26updateSpatialEntities	955
7.167.4 Member Data Documentation	955
7.167.4.1 consideredSpatialEntityTypes	955
7.167.4.2 ERR_GET_NUMERIC_STATE_VARIABLE_PREFIX	955
7.167.4.3 ERR_GET_NUMERIC_STATE_VARIABLE_SUFFIX	956
7.167.4.4 numericStateVariables	956
7.167.4.5 spatialEntities	956
7.167.4.6 value	956
7.168multiscale::verification::TimePointEvaluator Class Reference	956
7.168.1 Detailed Description	957
7.168.2 Member Function Documentation	957
7.168.2.1 getSpatialMeasureValues	957
7.168.2.2 getSpatialMeasureValues	957
7.169multiscaletest::TraceEvaluationTest Class Reference	958
7.169.1 Detailed Description	960
7.169.2 Member Function Documentation	960
7.169.2.1 InitialiseQuery	960
7.169.2.2 InitialiseTrace	960
7.169.2.3 RunEvaluationTest	961
7.169.2.4 RunTest	961
7.169.2.5 ValidateTestResults	961
7.169.3 Member Data Documentation	961
7.169.3.1 evaluationResult	961
7.169.3.2 query	962
7.169.3.3 trace	962
7.170multiscale::verification::UnaryNumericFilterAttribute Class Reference	962
7.170.1 Detailed Description	963
7.170.2 Member Data Documentation	963
7.170.2.1 filterNumericMeasure	963
7.170.2.2 unaryNumericMeasure	964
7.171multiscale::verification::UnaryNumericMeasureAttribute Class Reference	964
7.171.1 Detailed Description	964
7.171.2 Member Data Documentation	964
7.171.2.1 unaryNumericMeasureType	964

7.172multiscale::verification::UnaryNumericMeasureTypeParser Struct - Reference	965
7.172.1 Detailed Description	965
7.172.2 Constructor & Destructor Documentation	965
7.172.2.1 UnaryNumericMeasureTypeParser	965
7.173multiscale::verification::UnaryNumericNumericAttribute Class Reference	965
7.173.1 Detailed Description	966
7.173.2 Member Data Documentation	966
7.173.2.1 numericMeasure	966
7.173.2.2 unaryNumericMeasure	967
7.174multiscale::verification::UnarySpatialConstraintAttribute Class Reference	967
7.174.1 Detailed Description	968
7.174.2 Member Data Documentation	968
7.174.2.1 comparator	968
7.174.2.2 filterNumericMeasure	968
7.174.2.3 spatialMeasure	968
7.175multiscale::verification::UnarySubsetAttribute Class Reference	968
7.175.1 Detailed Description	969
7.175.2 Member Data Documentation	969
7.175.2.1 subset	969
7.175.2.2 unarySubsetMeasure	970
7.176multiscale::verification::UnarySubsetMeasureAttribute Class Reference	970
7.176.1 Detailed Description	970
7.176.2 Member Data Documentation	970
7.176.2.1 unarySubsetMeasureType	970
7.177multiscale::verification::UnarySubsetMeasureTypeParser Struct - Reference	971
7.177.1 Detailed Description	971
7.177.2 Constructor & Destructor Documentation	971
7.177.2.1 UnarySubsetMeasureTypeParser	971
7.178multiscale::verification::UnaryTypeConstraintAttribute Class Reference	971
7.178.1 Detailed Description	972
7.178.2 Member Data Documentation	972
7.178.2.1 comparator	972

7.178.2.2 filterNumericMeasure	973
7.179 multiscale::UnexpectedBehaviourException Class Reference	973
7.179.1 Detailed Description	975
7.179.2 Constructor & Destructor Documentation	975
7.179.2.1 UnexpectedBehaviourException	975
7.179.2.2 UnexpectedBehaviourException	975
7.179.2.3 UnexpectedBehaviourException	975
7.180 multiscale::verification::UnexpectedErrorHandler Struct Reference	975
7.180.1 Detailed Description	976
7.180.2 Member Function Documentation	976
7.180.2.1 getExpectedTokenAsString	976
7.180.2.2 operator()	976
7.181 multiscale::UnimplementedMethodException Class Reference	977
7.181.1 Detailed Description	980
7.181.2 Constructor & Destructor Documentation	980
7.181.2.1 UnimplementedMethodException	980
7.181.2.2 UnimplementedMethodException	980
7.181.2.3 UnimplementedMethodException	980
7.182 multiscale::verification::UntilLogicPropertyAttribute Class Reference	980
7.182.1 Detailed Description	981
7.182.2 Member Data Documentation	981
7.182.2.1 endTimepoint	981
7.182.2.2 logicProperty	981
7.182.2.3 startTimepoint	981
7.183 multiscale::XmlValidator::XmlValidationErrorHandler Class Reference	982
7.183.1 Detailed Description	984
7.183.2 Member Function Documentation	984
7.183.2.1 constructExceptionMessage	984
7.183.2.2 error	985
7.183.2.3 fatalError	985
7.183.2.4 handleValidationException	985
7.183.2.5 resetErrors	985
7.183.2.6 warning	986
7.183.3 Member Data Documentation	986

7.183.3.1 ERR_EXCEPTION_BEGIN_MSG	986
7.183.3.2 ERR_EXCEPTION_COLUMN_MSG	986
7.183.3.3 ERR_EXCEPTION_END_MSG	986
7.183.3.4 ERR_EXCEPTION_LINE_MSG	986
7.183.3.5 ERR_EXCEPTION_MIDDLE_MSG	986
7.184 multiscale::XmlValidator Class Reference	987
7.184.1 Detailed Description	989
7.184.2 Member Function Documentation	990
7.184.2.1 checkIfValidXmlFile	990
7.184.2.2 configureParser	990
7.184.2.3 isValidXmlFile	990
7.184.2.4 isValidXmlPathAndFile	991
7.184.2.5 loadParserSchema	991
7.184.2.6 validateXmlFilepath	991
7.184.2.7 validateXmlSchemaPath	992
7.184.2.8 verifyIfValidXmlFile	992
7.184.3 Member Data Documentation	992
7.184.3.1 ERR_INVALID_SCHEMA_FILEPATH	992
7.184.3.2 ERR_INVALID_XML_FILEPATH	992
7.184.3.3 ERR_SCHEMA_CONTENTS	993
8 File Documentation	995
8.1 config/mainpage.dox File Reference	995
8.2 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/- Multiscale.hpp File Reference	995
8.3 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/- MultiscaleTest.hpp File Reference	996
8.4 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- AlgorithmException.hpp File Reference	996
8.5 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- ExceptionHandler.hpp File Reference	997
8.6 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- FileOpenException.hpp File Reference	997
8.7 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- IndexOutOfBoundsException.hpp File Reference	998

8.8	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- InvalidInputException.hpp File Reference	999
8.9	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- IOException.hpp File Reference	1000
8.10	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- MultiscaleException.hpp File Reference	1001
8.10.1	Define Documentation	1002
8.10.1.1	MS_throw	1002
8.10.1.2	MS_throw_detailed	1004
8.11	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- NumericException.hpp File Reference	1004
8.12	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- RuntimeException.hpp File Reference	1005
8.13	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- TestException.hpp File Reference	1006
8.14	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- UnexpectedBehaviourException.hpp File Reference	1006
8.15	/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- UnimplementedMethodException.hpp File Reference	1007
8.16	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- CircularityMeasure.hpp File Reference	1008
8.17	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- Cluster.hpp File Reference	1009
8.18	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Cluster.hpp File Reference	1010
8.19	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/clust- SimulationClusterDetector.hpp File Reference	1010
8.20	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- ClusterDetector.hpp File Reference	1011
8.21	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- DataPoint.hpp File Reference	1012
8.22	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- DBSCAN.hpp File Reference	1012
8.23	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- Detector.hpp File Reference	1013
8.24	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/- Entity.hpp File Reference	1014
8.25	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/factor- CircularMatFactory.hpp File Reference	1014

8.26 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/RectangularMatFactory.hpp File Reference	1015
8.27 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/MatFactory.hpp File Reference	1016
8.28 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/Region.hpp File Reference	1016
8.29 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Region.hpp File Reference	1017
8.30 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/RegionDetector.hpp File Reference	1018
8.31 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/Shape2D.hpp File Reference	1019
8.32 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/Silhouette.hpp File Reference	1019
8.33 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/SpatialEntityPseudo3D.hpp File Reference	1020
8.34 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/SpatialEntityPseudo3DType.hpp File Reference	1020
8.35 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/-CircularMatFactorySample.cpp File Reference	1021
8.35.1 Function Documentation	1021
8.35.1.1 main	1021
8.36 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/-LexicographicIteratorSample.cpp File Reference	1021
8.36.1 Function Documentation	1022
8.36.1.1 main	1022
8.37 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/-RectangularMatFactorySample.cpp File Reference	1022
8.37.1 Function Documentation	1023
8.37.1.1 main	1023
8.38 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-CircularDetectRegions.cpp File Reference	1023
8.38.1 Function Documentation	1024
8.38.1.1 areValidParameters	1024
8.38.1.2 initArgumentsConfig	1024
8.38.1.3 loadDetectorParameterValues	1024
8.38.1.4 loadDetectorParameterValues	1025

8.38.1.5	main	1025
8.38.1.6	printHelpInformation	1025
8.38.1.7	printWrongParameters	1025
8.38.1.8	saveDetectorParameterValues	1025
8.38.1.9	saveDetectorParameterValues	1026
8.38.2	Variable Documentation	1026
8.38.2.1	CONFIG_FILE	1026
8.38.2.2	LABEL_ALPHA	1026
8.38.2.3	LABEL_BETA	1026
8.38.2.4	LABEL_BLUR_KERNEL_SIZE	1026
8.38.2.5	LABEL_EPSILON	1026
8.38.2.6	LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS .	1027
8.38.2.7	LABEL_REGION_AREA_THRESH	1027
8.38.2.8	LABEL_ROOT_COMMENT	1027
8.38.2.9	LABEL_THRESHOLD_VALUE	1027
8.38.2.10	ROOT_COMMENT	1027
8.39	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-CircularDetectRegions.in.cpp File Reference	1027
8.39.1	Function Documentation	1029
8.39.1.1	areValidParameters	1029
8.39.1.2	initArgumentsConfig	1029
8.39.1.3	loadDetectorParameterValues	1029
8.39.1.4	loadDetectorParameterValues	1029
8.39.1.5	main	1029
8.39.1.6	printHelpInformation	1029
8.39.1.7	printWrongParameters	1030
8.39.1.8	saveDetectorParameterValues	1030
8.39.1.9	saveDetectorParameterValues	1030
8.39.2	Variable Documentation	1030
8.39.2.1	CONFIG_FILE	1030
8.39.2.2	LABEL_ALPHA	1030
8.39.2.3	LABEL_BETA	1030
8.39.2.4	LABEL_BLUR_KERNEL_SIZE	1030
8.39.2.5	LABEL_EPSILON	1031

8.39.2.6	LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS	1031
8.39.2.7	LABEL_REGION_AREA_THRESH	1031
8.39.2.8	LABEL_ROOT_COMMENT	1031
8.39.2.9	LABEL_THRESHOLD_VALUE	1031
8.39.2.10	ROOT_COMMENT	1031
8.40	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-CircularityMeasure.cpp File Reference	1031
8.41	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Cluster.cpp File Reference	1032
8.42	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/-SimulationClusterDetector.cpp File Reference	1032
8.43	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-ClusterDetector.cpp File Reference	1033
8.44	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-DBSCAN.cpp File Reference	1034
8.45	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Detector.cpp File Reference	1034
8.46	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Entity.cpp File Reference	1034
8.47	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-CircularMatFactory.cpp File Reference	1035
8.48	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-RectangularMatFactory.cpp File Reference	1035
8.49	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-MatFactory.cpp File Reference	1036
8.50	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-RectangularDetectRegions.cpp File Reference	1037
8.50.1	Function Documentation	1038
8.50.1.1	areValidParameters	1038
8.50.1.2	initArgumentsConfig	1038
8.50.1.3	loadDetectorParameterValues	1038
8.50.1.4	loadDetectorParameterValues	1038
8.50.1.5	main	1038
8.50.1.6	printHelpInformation	1039
8.50.1.7	printWrongParameters	1039
8.50.1.8	saveDetectorParameterValues	1039
8.50.1.9	saveDetectorParameterValues	1039

8.50.2 Variable Documentation	1039
8.50.2.1 CONFIG_FILE	1039
8.50.2.2 LABEL_ALPHA	1039
8.50.2.3 LABEL_BETA	1039
8.50.2.4 LABEL_BLUR_KERNEL_SIZE	1040
8.50.2.5 LABEL_EPSILON	1040
8.50.2.6 LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS .	1040
8.50.2.7 LABEL_REGION_AREA_THRESH	1040
8.50.2.8 LABEL_ROOT_COMMENT	1040
8.50.2.9 LABEL_THRESHOLD_VALUE	1040
8.50.2.10 ROOT_COMMENT	1040
8.51 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/RectangularDetectRegions.in.cpp File Reference	1040
8.51.1 Function Documentation	1042
8.51.1.1 areValidParameters	1042
8.51.1.2 initArgumentsConfig	1042
8.51.1.3 loadDetectorParameterValues	1042
8.51.1.4 loadDetectorParameterValues	1042
8.51.1.5 main	1042
8.51.1.6 printHelpInformation	1042
8.51.1.7 printWrongParameters	1043
8.51.1.8 saveDetectorParameterValues	1043
8.51.1.9 saveDetectorParameterValues	1043
8.51.2 Variable Documentation	1043
8.51.2.1 CONFIG_FILE	1043
8.51.2.2 LABEL_ALPHA	1043
8.51.2.3 LABEL_BETA	1043
8.51.2.4 LABEL_BLUR_KERNEL_SIZE	1043
8.51.2.5 LABEL_EPSILON	1044
8.51.2.6 LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS .	1044
8.51.2.7 LABEL_REGION_AREA_THRESH	1044
8.51.2.8 LABEL_ROOT_COMMENT	1044
8.51.2.9 LABEL_THRESHOLD_VALUE	1044
8.51.2.10 ROOT_COMMENT	1044

8.52 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/Region.cpp File Reference	1044
8.53 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/RegionDetector.cpp File Reference	1045
8.54 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/Silhouette.cpp File Reference	1045
8.55 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/SimulationDetectClusters.cpp File Reference	1046
8.55.1 Function Documentation	1047
8.55.1.1 areValidParameters	1047
8.55.1.2 initArgumentsConfig	1047
8.55.1.3 loadDetectorParameterValues	1047
8.55.1.4 loadDetectorParameterValues	1047
8.55.1.5 main	1048
8.55.1.6 printHelpInformation	1048
8.55.1.7 printWrongParameters	1048
8.55.1.8 saveDetectorParameterValues	1048
8.55.1.9 saveDetectorParameterValues	1048
8.55.2 Variable Documentation	1048
8.55.2.1 CONFIG_FILE	1048
8.55.2.2 LABEL_EPS	1048
8.55.2.3 LABEL_MINPOINTS	1049
8.55.2.4 LABEL_ROOT_COMMENT	1049
8.55.2.5 ROOT_COMMENT	1049
8.56 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-/SimulationDetectClusters.in.cpp File Reference	1049
8.56.1 Function Documentation	1050
8.56.1.1 areValidParameters	1050
8.56.1.2 initArgumentsConfig	1050
8.56.1.3 loadDetectorParameterValues	1050
8.56.1.4 loadDetectorParameterValues	1051
8.56.1.5 main	1051
8.56.1.6 printHelpInformation	1051
8.56.1.7 printWrongParameters	1051
8.56.1.8 saveDetectorParameterValues	1051

8.56.1.9	saveDetectorParameterValues	1051
8.56.2	Variable Documentation	1051
8.56.2.1	CONFIG_FILE	1052
8.56.2.2	LABEL_EPS	1052
8.56.2.3	LABEL_MINPOINTS	1052
8.56.2.4	LABEL_ROOT_COMMENT	1052
8.56.2.5	ROOT_COMMENT	1052
8.57	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/ SpatialEntityPseudo3D.cpp File Reference	1052
8.58	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/ DBSCANTest.cpp File Reference	1053
8.58.1	Function Documentation	1053
8.58.1.1	convertPoints	1053
8.58.1.2	main	1054
8.58.1.3	printResults	1054
8.58.1.4	runTest	1054
8.58.1.5	runTest1	1054
8.58.1.6	runTest2	1054
8.58.1.7	runTest3	1054
8.58.1.8	runTest4	1054
8.58.1.9	runTest5	1055
8.58.1.10	runTests	1055
8.59	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- ConsolePrinter.hpp File Reference	1055
8.60	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- Filesystem.hpp File Reference	1056
8.61	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- Geometry2D.hpp File Reference	1057
8.62	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- LexicographicNumberIterator.hpp File Reference	1057
8.63	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- NumberIteratorType.hpp File Reference	1058
8.64	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- StandardNumberIterator.hpp File Reference	1058
8.65	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- MinEnclosingTriangleFinder.hpp File Reference	1059

8.66 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-NumberIterator.hpp File Reference	1059
8.67 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-Numeric.hpp File Reference	1060
8.68 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-NumericRangeManipulator.hpp File Reference	1061
8.69 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-OperatingSystem.hpp File Reference	1061
8.70 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-RGBColourGenerator.hpp File Reference	1062
8.71 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/-BetaDistribution.hpp File Reference	1062
8.72 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/-BinomialDistribution.hpp File Reference	1063
8.73 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/-Distribution.hpp File Reference	1064
8.74 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-StringManipulator.hpp File Reference	1064
8.75 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-XmlValidator.hpp File Reference	1065
8.76 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-ConsolePrinterSample.cpp File Reference	1065
8.76.1 Function Documentation	1066
8.76.1.1 main	1066
8.76.2 Variable Documentation	1066
8.76.2.1 SAMPLE_MSG	1066
8.76.2.2 SAMPLE_TAG	1067
8.77 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-ExecuteProgramSample.cpp File Reference	1067
8.77.1 Function Documentation	1067
8.77.1.1 main	1067
8.78 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-LineCircleIntersectionSample.cpp File Reference	1068
8.78.1 Function Documentation	1068
8.78.1.1 main	1068
8.78.1.2 printPoints	1068
8.79 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-MinEnclosingTriangleFinderSample.cpp File Reference	1068

8.79.1	Function Documentation	1070
8.79.1.1	arePointsEnclosed	1070
8.79.1.2	generateRandomSetOf2DPoints	1070
8.79.1.3	isOneEdgeFlush	1070
8.79.1.4	isTriangleTouchingPolygon	1070
8.79.1.5	isValidTriangle	1070
8.79.1.6	main	1071
8.79.1.7	outputMinEnclosingTriangleFinderResults	1071
8.79.1.8	printPolygon	1071
8.79.1.9	runMinEnclosingTriangleFinder	1071
8.79.1.10	runMinEnclosingTriangleFinder	1071
8.79.1.11	runMinEnclosingTriangleFinderUsingRandomPolygons	1071
8.79.2	Variable Documentation	1071
8.79.2.1	KEY_ESC	1072
8.79.2.2	LINE_THICKNESS	1072
8.79.2.3	MAX_POLYGON_POINTS	1072
8.79.2.4	NR_RAND_POLYGONS	1072
8.79.2.5	POINT_IN_TRIANGLE_THRESH	1072
8.79.2.6	POLYGON_POINT_X_MAX	1072
8.79.2.7	POLYGON_POINT_Y_MAX	1072
8.79.2.8	RADIUS	1072
8.79.2.9	WIN_MIN_AREA_TRIANGLE	1073
8.80	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/- RGBColourGeneratorSample.cpp File Reference	1073
8.80.1	Function Documentation	1073
8.80.1.1	main	1073
8.81	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/- ConsolePrinter.cpp File Reference	1074
8.82	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/- Filesystem.cpp File Reference	1074
8.83	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/- Geometry2D.cpp File Reference	1075
8.84	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/- LexicographicNumberIterator.cpp File Reference	1076
8.85	/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/- StandardNumberIterator.cpp File Reference	1076

8.86 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/-MinEnclosingTriangleFinder.cpp File Reference	1076
8.87 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/-NumberIterator.cpp File Reference	1077
8.88 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/-Numeric.cpp File Reference	1077
8.89 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/-OperatingSystem.cpp File Reference	1078
8.90 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/R-GBColourGenerator.cpp File Reference	1078
8.91 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-BetaDistribution.cpp File Reference	1079
8.92 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-BinomialDistribution.cpp File Reference	1079
8.93 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-Distribution.cpp File Reference	1080
8.94 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/StringManipulator.cpp File Reference	1080
8.95 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/XmlValidator.cpp File Reference	1081
8.96 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/-Geometry2DTest.cpp File Reference	1081
8.96.1 Function Documentation	1082
8.96.1.1 main	1082
8.96.1.2 TEST	1082
8.96.1.3 TEST	1082
8.96.2 Variable Documentation	1083
8.96.2.1 DOUBLE_COMP_ERROR	1083
8.97 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/-MinEnclosingTriangleFinderTest.cpp File Reference	1083
8.97.1 Function Documentation	1084
8.97.1.1 main	1084
8.97.1.2 TEST_F	1084
8.97.1.3 TEST_F	1084
8.97.1.4 TEST_F	1084
8.97.1.5 TEST_F	1084
8.98 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/-NumericTest.cpp File Reference	1084

8.98.1 Function Documentation	1086
8.98.1.1 main	1086
8.98.1.2 TEST	1086
8.98.1.3 TEST	1086
8.98.1.4 TEST	1086
8.98.1.5 TEST	1086
8.98.1.6 TEST	1086
8.98.1.7 TEST	1086
8.98.1.8 TEST	1086
8.98.1.9 TEST	1086
8.98.1.10 TEST	1087
8.98.1.11 TEST	1087
8.98.1.12 TEST	1087
8.98.1.13 TEST	1087
8.98.1.14 TEST	1087
8.98.1.15 TEST	1087
8.98.1.16 TEST	1087
8.98.1.17 TEST	1087
8.98.1.18 TEST	1088
8.98.1.19 TEST	1088
8.98.1.20 TEST	1088
8.98.1.21 TEST	1088
8.98.1.22 TEST	1088
8.98.2 Variable Documentation	1088
8.98.2.1 DOUBLE_COMP_ERROR	1088
8.99 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/- StatisticsTest.cpp File Reference	1088
8.99.1 Function Documentation	1089
8.99.1.1 TEST	1089
8.99.1.2 TEST	1089
8.99.1.3 TEST	1089
8.100/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/And- ConstraintAttribute.hpp File Reference	1090

8.101/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/AndLogicPropertyAttribute.hpp File Reference	1090
8.102/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-BinaryNumericFilterAttribute.hpp File Reference	1091
8.103/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-BinaryNumericMeasureAttribute.hpp File Reference	1092
8.104/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-BinaryNumericNumericAttribute.hpp File Reference	1093
8.105/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-BinarySubsetAttribute.hpp File Reference	1093
8.106/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-BinarySubsetMeasureAttribute.hpp File Reference	1094
8.107/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-ComparatorAttribute.hpp File Reference	1095
8.108/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-ConstraintAttribute.hpp File Reference	1096
8.109/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-DifferenceAttribute.hpp File Reference	1097
8.110/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-EquivalenceConstraintAttribute.hpp File Reference	1098
8.111/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-EquivalenceLogicPropertyAttribute.hpp File Reference	1098
8.112/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Filter-NumericMeasureAttribute.hpp File Reference	1099
8.113/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Filter-SubsetAttribute.hpp File Reference	1100
8.114/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-FutureLogicPropertyAttribute.hpp File Reference	1101

8.115/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-GlobalLogicPropertyAttribute.hpp File Reference	1101
8.116/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-ImplicationConstraintAttribute.hpp File Reference	1102
8.117/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-ImplicationLogicPropertyAttribute.hpp File Reference	1103
8.118/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Logic-PropertyAttribute.hpp File Reference	1104
8.119/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Next-KLogicPropertyAttribute.hpp File Reference	1105
8.120/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Next-LogicPropertyAttribute.hpp File Reference	1105
8.121/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-Nil.hpp File Reference	1106
8.121.1 Function Documentation	1107
8.121.1.1 BOOST_FUSION_ADAPT_STRUCT	1107
8.122/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Not-ConstraintAttribute.hpp File Reference	1107
8.123/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Not-LogicPropertyAttribute.hpp File Reference	1108
8.124/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-NumericMeasureAttribute.hpp File Reference	1108
8.125/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-NumericNumericComparisonAttribute.hpp File Reference	1109
8.126/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-NumericSpatialAttribute.hpp File Reference	1110
8.127/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-NumericSpatialNumericComparisonAttribute.hpp File Reference	1111

8.128/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericStateVariableAttribute.hpp File Reference	1112
8.129/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Or- ConstraintAttribute.hpp File Reference	1112
8.130/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Or- LogicPropertyAttribute.hpp File Reference	1113
8.131/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryConstraintAttribute.hpp File Reference	1114
8.132/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryLogicPropertyAttribute.hpp File Reference	1115
8.132.1 Function Documentation	1116
8.132.1.1 BOOST_FUSION_ADAPT_STRUCT	1116
8.133/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryNumericMeasureAttribute.hpp File Reference	1116
8.134/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- ProbabilisticLogicPropertyAttribute.hpp File Reference	1117
8.134.1 Function Documentation	1118
8.134.1.1 BOOST_FUSION_ADAPT_STRUCT	1118
8.135/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- QuaternarySubsetAttribute.hpp File Reference	1118
8.136/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- QuaternarySubsetMeasureAttribute.hpp File Reference	1118
8.137/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SpatialMeasureAttribute.hpp File Reference	1119
8.138/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SpatialNumericComparisonAttribute.hpp File Reference	1120
8.139/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/State- VariableAttribute.hpp File Reference	1121

8.140/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-SubsetAttribute.hpp File Reference	1122
8.141/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-SubsetOperationAttribute.hpp File Reference	1123
8.142/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-SubsetSpecificAttribute.hpp File Reference	1123
8.143/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-SubsetSubsetOperationAttribute.hpp File Reference	1125
8.144/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-SynthesizedAttribute.hpp File Reference	1125
8.145/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-TernarySubsetAttribute.hpp File Reference	1126
8.146/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-TernarySubsetMeasureAttribute.hpp File Reference	1127
8.147/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnaryNumericFilterAttribute.hpp File Reference	1128
8.148/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnaryNumericMeasureAttribute.hpp File Reference	1129
8.149/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnaryNumericNumericAttribute.hpp File Reference	1130
8.150/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnarySpatialConstraintAttribute.hpp File Reference	1130
8.151/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnarySubsetAttribute.hpp File Reference	1131
8.152/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnarySubsetMeasureAttribute.hpp File Reference	1132
8.153/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnaryTypeConstraintAttribute.hpp File Reference	1133

8.154/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Until-LogicPropertyAttribute.hpp File Reference	1133
8.155/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ApproximateBayesianModelChecker.hpp File Reference	1134
8.156/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ApproximateBayesianModelCheckerFactory.hpp File Reference	1135
8.157/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ApproximateProbabilisticModelChecker.hpp File Reference	1135
8.158/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ApproximateProbabilisticModelCheckerFactory.hpp File Reference	1136
8.159/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/BayesianModelChecker.hpp File Reference	1137
8.160/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/BayesianModelCheckerFactory.hpp File Reference	1137
8.161/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ModelChecker.hpp File Reference	1138
8.162/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ModelCheckerFactory.hpp File Reference	1138
8.163/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ModelCheckingManager.hpp File Reference	1139
8.164/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ModelCheckingOutputWriter.hpp File Reference	1140
8.165/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ProbabilisticBlackBoxModelChecker.hpp File Reference	1140
8.166/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/ProbabilisticBlackBoxModelCheckerFactory.hpp File Reference	1141
8.167/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking-/StatisticalModelChecker.hpp File Reference	1141

8.168/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/-StatisticalModelCheckerFactory.hpp File Reference	1142
8.169/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/LogicPropertyDataReader.hpp File Reference	1143
8.170/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/SpatialTemporalDataReader.hpp File Reference	1143
8.171/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ModelCheckingException.hpp File Reference	1144
8.172/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ModelCheckingHelpRequestException.hpp File Reference	1145
8.173/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ParserGrammarExceptionHandler.hpp File Reference	1146
8.174/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ParserGrammarExtraInputException.hpp File Reference	1146
8.175/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ParserGrammarProbabilityException.hpp File Reference	1147
8.176/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ParserGrammarUnexpectedTokenException.hpp File Reference	1148
8.177/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ParserGrammarUnparseableInputException.hpp File Reference	1148
8.178/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-SpatialTemporalException.hpp File Reference	1149
8.179/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/-CommandLineModelChecking.hpp File Reference	1150
8.180/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/-ProbabilityErrorHandler.hpp File Reference	1150
8.181/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/-UnexpectedTokenErrorHandler.hpp File Reference	1151

8.182/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/AbstractSyntaxTree.hpp File Reference	1152
8.183/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialEntity.hpp File Reference	1152
8.184/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp File Reference	1153
8.185/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/TimePoint.hpp File Reference	1154
8.186/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/Parser.hpp File Reference	1155
8.187/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/ParserGrammar.hpp File Reference	1156
8.188/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp File Reference	1156
8.189/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ComparatorEvaluator.hpp File Reference	1158
8.190/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ConstraintEvaluator.hpp File Reference	1158
8.191/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ConstraintVisitor.hpp File Reference	1159
8.192/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/FilterNumericVisitor.hpp File Reference	1160
8.193/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/LogicPropertyVisitor.hpp File Reference	1161
8.194/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericEvaluator.hpp File Reference	1161
8.195/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/NumericVisitor.hpp File Reference	1162

8.196/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/Spatial-EntityEvaluator.hpp File Reference	1163
8.197/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/Spatial-MeasureEvaluator.hpp File Reference	1163
8.198/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/Subset-Visitor.hpp File Reference	1164
8.199/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/Time-PointEvaluator.hpp File Reference	1165
8.200/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/LogicPropertyDataReaderSample.cpp File Reference .	1166
8.200.1 Function Documentation	1166
8.200.1.1 main	1166
8.200.1.2 printParsingResult	1166
8.200.1.3 printQueries	1167
8.200.1.4 readQueriesFromFile	1167
8.201/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserEvaluationSample.cpp File Reference	1167
8.201.1 Function Documentation	1167
8.201.1.1 initialiseTrace	1167
8.201.1.2 main	1168
8.202/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserSample.cpp File Reference	1168
8.202.1 Function Documentation	1168
8.202.1.1 main	1168
8.203/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/SpatialTemporalDataReaderSample.cpp File Reference	1169
8.203.1 Function Documentation	1169
8.203.1.1 main	1169
8.203.1.2 printSpatialEntities	1169
8.203.1.3 printTimePoint	1170
8.203.1.4 printTrace	1170
8.203.1.5 readValidXmlFiles	1170
8.203.1.6 readValidXmlFilesFromFolder	1170

8.204/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinaryNumericMeasureAttribute.cpp File Reference	1170
8.205/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinarySubsetMeasureAttribute.cpp File Reference	1171
8.206/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ComparatorAttribute.cpp File Reference	1171
8.207/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ProbabilisticLogicPropertyAttribute.cpp File - Reference	1172
8.208/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/QuaternarySubsetMeasureAttribute.cpp File - Reference	1172
8.209/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SpatialMeasureAttribute.cpp File Reference	1173
8.210/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SubsetSpecificAttribute.cpp File Reference	1173
8.211/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/TernarySubsetMeasureAttribute.cpp File Reference	1174
8.212/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnaryNumericMeasureAttribute.cpp File Reference	1174
8.213/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnarySubsetMeasureAttribute.cpp File Reference	1175
8.214/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelChecker.cpp File Reference	1175
8.215/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelCheckerFactory.cpp File Reference	1176
8.216/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModelChecker.cpp File Reference	1176
8.217/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModelCheckerFactory.cpp File Reference	1176
8.218/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/BayesianModelChecker.cpp File Reference	1177
8.219/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/BayesianModelCheckerFactory.cpp File Reference	1177
8.220/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelChecker.cpp File Reference	1178

8.221/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingManager.cpp File Reference . . .	1178
8.222/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.cpp File Reference . . .	1178
8.223/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.in.cpp File Reference	1179
8.224/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelChecker.cpp File - Reference	1179
8.225/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelCheckerFactory.cpp File Reference	1180
8.226/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/StatisticalModelChecker.cpp File Reference	1180
8.227/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/StatisticalModelCheckerFactory.cpp File Reference	1181
8.228/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/LogicPropertyDataReader.cpp File Reference	1181
8.229/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/SpatialTemporalDataReader.cpp File Reference	1181
8.230/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/SpatialTemporalDataReader.in.cpp File Reference . . .	1182
8.231/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ParserGrammarExceptionHandler.cpp File - Reference	1183
8.232/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/CommandLineModelChecking.cpp File Reference	1183
8.233/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/AbstractSyntaxTree.cpp File Reference	1184
8.234/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialEntity.cpp File Reference	1184
8.235/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialTemporalTrace.cpp File Reference	1185
8.236/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/TimePoint.cpp File Reference	1185
8.237/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/Mule.cpp File Reference	1185
8.237.1 Function Documentation	1186
8.237.1.1 main	1186
8.237.1.2 runModelCheckingTask	1186

8.238/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/Parser.cpp File Reference	1186
8.239/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericFilterTest.hpp File Reference	1187
8.239.1 Function Documentation	1187
8.239.1.1 TEST	1187
8.239.1.2 TEST	1187
8.239.1.3 TEST	1187
8.239.1.4 TEST	1188
8.239.1.5 TEST	1188
8.239.1.6 TEST	1188
8.239.1.7 TEST	1188
8.239.1.8 TEST	1188
8.239.1.9 TEST	1188
8.239.1.10TEST	1188
8.239.1.11TEST	1188
8.240/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericMeasureTest.hpp File Reference	1189
8.240.1 Function Documentation	1189
8.240.1.1 TEST	1189
8.240.1.2 TEST	1189
8.240.1.3 TEST	1189
8.240.1.4 TEST	1190
8.240.1.5 TEST	1190
8.240.1.6 TEST	1190
8.240.1.7 TEST	1190
8.240.1.8 TEST	1190
8.241/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericNumericTest.hpp File Reference	1190
8.241.1 Function Documentation	1191
8.241.1.1 TEST	1191
8.241.1.2 TEST	1191
8.241.1.3 TEST	1191
8.241.1.4 TEST	1191
8.241.1.5 TEST	1191

8.241.1.6 TEST	1191
8.241.1.7 TEST	1192
8.241.1.8 TEST	1192
8.241.1.9 TEST	1192
8.241.1.10TEST	1192
8.241.1.11TEST	1192
8.242/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetMeasureTest.hpp File Reference . . .	1192
8.242.1 Function Documentation	1193
8.242.1.1 TEST	1193
8.242.1.2 TEST	1193
8.242.1.3 TEST	1193
8.242.1.4 TEST	1193
8.242.1.5 TEST	1193
8.242.1.6 TEST	1194
8.242.1.7 TEST	1194
8.242.1.8 TEST	1194
8.242.1.9 TEST	1194
8.242.1.10TEST	1194
8.242.1.11TEST	1194
8.242.1.12TEST	1194
8.242.1.13TEST	1194
8.242.1.14TEST	1195
8.243/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetTest.hpp File Reference	1195
8.243.1 Function Documentation	1195
8.243.1.1 TEST	1195
8.243.1.2 TEST	1195
8.243.1.3 TEST	1196
8.243.1.4 TEST	1196
8.243.1.5 TEST	1196
8.243.1.6 TEST	1196
8.243.1.7 TEST	1196
8.243.1.8 TEST	1196

8.243.1.9 TEST	1196
8.243.1.10 TEST	1196
8.244/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ComparatorTest.hpp File Reference	1197
8.244.1 Function Documentation	1197
8.244.1.1 TEST	1197
8.244.1.2 TEST	1197
8.244.1.3 TEST	1197
8.244.1.4 TEST	1197
8.244.1.5 TEST	1198
8.244.1.6 TEST	1198
8.244.1.7 TEST	1198
8.244.1.8 TEST	1198
8.245/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundConstraintTest.hpp File Reference	1198
8.245.1 Function Documentation	1199
8.245.1.1 TEST	1199
8.245.1.2 TEST	1199
8.245.1.3 TEST	1199
8.245.1.4 TEST	1199
8.245.1.5 TEST	1199
8.245.1.6 TEST	1200
8.245.1.7 TEST	1200
8.245.1.8 TEST	1200
8.245.1.9 TEST	1200
8.245.1.10 TEST	1200
8.245.1.11 TEST	1200
8.245.1.12 TEST	1200
8.245.2 Variable Documentation	1201
8.245.2.1 CONSTRAINTS_BINARY_OPERATORS	1201
8.246/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundLogicPropertyTest.hpp File Reference	1201
8.246.1 Function Documentation	1202
8.246.1.1 TEST	1202
8.246.1.2 TEST	1202

8.246.1.3 TEST	1202
8.246.1.4 TEST	1202
8.246.1.5 TEST	1202
8.246.1.6 TEST	1202
8.246.1.7 TEST	1202
8.246.1.8 TEST	1203
8.246.1.9 TEST	1203
8.246.1.10TEST	1203
8.246.1.11TEST	1203
8.246.1.12TEST	1203
8.246.2 Variable Documentation	1203
8.246.2.1 LOGIC_PROPERTIES_BINARY_OPERATORS	1203
8.247/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ConstraintParenthesesTest.hpp File Reference	1204
8.247.1 Function Documentation	1204
8.247.1.1 TEST	1204
8.247.1.2 TEST	1204
8.247.1.3 TEST	1204
8.247.1.4 TEST	1205
8.247.1.5 TEST	1205
8.247.1.6 TEST	1205
8.247.1.7 TEST	1205
8.247.1.8 TEST	1205
8.247.1.9 TEST	1205
8.247.1.10TEST	1205
8.248/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ConstraintTest.hpp File Reference	1205
8.248.1 Function Documentation	1206
8.248.1.1 TEST	1206
8.248.1.2 TEST	1206
8.248.1.3 TEST	1206
8.249/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/DifferenceTest.hpp File Reference	1206
8.249.1 Function Documentation	1207
8.249.1.1 TEST	1207

8.249.1.2 TEST	1207
8.249.1.3 TEST	1207
8.249.1.4 TEST	1207
8.249.1.5 TEST	1207
8.249.1.6 TEST	1208
8.249.1.7 TEST	1208
8.249.1.8 TEST	1208
8.249.1.9 TEST	1208
8.250/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FilterNumericMeasureTest.hpp File Reference . . .	1208
8.250.1 Function Documentation	1209
8.250.1.1 TEST	1209
8.250.1.2 TEST	1209
8.250.1.3 TEST	1209
8.250.1.4 TEST	1209
8.250.1.5 TEST	1209
8.251/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FilterSubsetTest.hpp File Reference	1209
8.251.1 Function Documentation	1210
8.251.1.1 TEST	1210
8.251.1.2 TEST	1210
8.251.1.3 TEST	1210
8.251.1.4 TEST	1210
8.251.1.5 TEST	1210
8.251.1.6 TEST	1210
8.251.1.7 TEST	1211
8.252/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FutureLogicPropertyTest.hpp File Reference . . .	1211
8.252.1 Function Documentation	1211
8.252.1.1 TEST	1211
8.252.1.2 TEST	1212
8.252.1.3 TEST	1212
8.252.1.4 TEST	1212
8.252.1.5 TEST	1212
8.252.1.6 TEST	1212

CONTENTS	cxix
8.252.1.7 TEST	1212
8.252.1.8 TEST	1212
8.252.1.9 TEST	1212
8.252.1.10TEST	1213
8.252.1.11TEST	1213
8.252.1.12TEST	1213
8.252.1.13TEST	1213
8.253/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/GlobalLogicPropertyTest.hpp File Reference . . .	1213
8.253.1 Function Documentation	1214
8.253.1.1 TEST	1214
8.253.1.2 TEST	1214
8.253.1.3 TEST	1214
8.253.1.4 TEST	1214
8.253.1.5 TEST	1214
8.253.1.6 TEST	1214
8.253.1.7 TEST	1215
8.253.1.8 TEST	1215
8.253.1.9 TEST	1215
8.253.1.10TEST	1215
8.253.1.11TEST	1215
8.253.1.12TEST	1215
8.253.1.13TEST	1215
8.254/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/LogicPropertyParenthesesTest.hpp File Reference	1216
8.254.1 Function Documentation	1216
8.254.1.1 TEST	1216
8.254.1.2 TEST	1216
8.254.1.3 TEST	1216
8.254.1.4 TEST	1217
8.254.1.5 TEST	1217
8.254.1.6 TEST	1217
8.254.1.7 TEST	1217
8.254.1.8 TEST	1217

8.254.1.9 TEST	1217
8.254.1.10 TEST	1217
8.255/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/LogicPropertyTest.hpp File Reference	1217
8.255.1 Function Documentation	1218
8.255.1.1 TEST	1218
8.255.1.2 TEST	1218
8.255.1.3 TEST	1218
8.255.1.4 TEST	1218
8.256/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/MultipleLogicPropertiesTest.hpp File Reference	1218
8.256.1 Function Documentation	1219
8.256.1.1 TEST	1219
8.256.1.2 TEST	1219
8.257/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextKLogicPropertyTest.hpp File Reference	1219
8.257.1 Function Documentation	1220
8.257.1.1 TEST	1220
8.257.1.2 TEST	1220
8.257.1.3 TEST	1220
8.257.1.4 TEST	1220
8.257.1.5 TEST	1220
8.257.1.6 TEST	1220
8.258/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextLogicPropertyTest.hpp File Reference	1221
8.258.1 Function Documentation	1221
8.258.1.1 TEST	1221
8.258.1.2 TEST	1221
8.259/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotConstraintTest.hpp File Reference	1221
8.259.1 Function Documentation	1222
8.259.1.1 TEST	1222
8.259.1.2 TEST	1222
8.259.1.3 TEST	1222
8.259.1.4 TEST	1222

8.259.1.5 TEST	1222
8.260/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotLogicPropertyTest.hpp File Reference	1222
8.260.1 Function Documentation	1223
8.260.1.1 TEST	1223
8.260.1.2 TEST	1223
8.260.1.3 TEST	1223
8.260.1.4 TEST	1223
8.261/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericMeasureTest.hpp File Reference	1223
8.261.1 Function Documentation	1224
8.261.1.1 TEST	1224
8.261.1.2 TEST	1224
8.262/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericNumericComparisonTest.hpp File - Reference	1224
8.262.1 Function Documentation	1225
8.262.1.1 TEST	1225
8.262.1.2 TEST	1225
8.262.1.3 TEST	1225
8.262.1.4 TEST	1225
8.262.1.5 TEST	1225
8.262.1.6 TEST	1225
8.263/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialMeasureTest.hpp File Reference	1226
8.263.1 Function Documentation	1226
8.263.1.1 TEST	1226
8.263.1.2 TEST	1226
8.264/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialNumericComparisonTest.hpp File Reference	1226
8.264.1 Function Documentation	1227
8.264.1.1 TEST	1227
8.264.1.2 TEST	1227
8.264.1.3 TEST	1227
8.264.1.4 TEST	1227

8.264.1.5 TEST	1227
8.264.1.6 TEST	1227
8.265/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericStateVariableTest.hpp File Reference . .	1228
8.265.1 Function Documentation	1228
8.265.1.1 TEST	1228
8.265.1.2 TEST	1228
8.265.1.3 TEST	1228
8.265.1.4 TEST	1229
8.265.1.5 TEST	1229
8.265.1.6 TEST	1229
8.265.1.7 TEST	1229
8.265.1.8 TEST	1229
8.265.1.9 TEST	1229
8.265.1.10TEST	1229
8.266/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ProbabilisticLogicPropertyTest.hpp File Reference	1230
8.266.1 Function Documentation	1230
8.266.1.1 TEST	1230
8.266.1.2 TEST	1231
8.266.1.3 TEST	1231
8.266.1.4 TEST	1231
8.266.1.5 TEST	1231
8.266.1.6 TEST	1231
8.266.1.7 TEST	1231
8.266.1.8 TEST	1231
8.266.1.9 TEST	1231
8.266.1.10TEST	1232
8.266.1.11TEST	1232
8.266.1.12TEST	1232
8.266.1.13TEST	1232
8.266.1.14TEST	1232
8.266.1.15TEST	1232
8.266.1.16TEST	1232

8.266.1.17TEST	1232
8.266.1.18TEST	1233
8.266.1.19TEST	1233
8.266.1.20TEST	1233
8.266.1.21TEST	1233
8.267/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetMeasureTest.hpp File Reference	1233
8.267.1 Function Documentation	1233
8.267.1.1 TEST	1234
8.267.1.2 TEST	1234
8.268/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetTest.hpp File Reference	1234
8.268.1 Function Documentation	1235
8.268.1.1 TEST	1235
8.268.1.2 TEST	1235
8.268.1.3 TEST	1235
8.268.1.4 TEST	1235
8.268.1.5 TEST	1235
8.268.1.6 TEST	1235
8.268.1.7 TEST	1235
8.268.1.8 TEST	1236
8.268.1.9 TEST	1236
8.268.1.10TEST	1236
8.268.1.11TEST	1236
8.268.1.12TEST	1236
8.268.1.13TEST	1236
8.268.1.14TEST	1236
8.268.1.15TEST	1236
8.268.1.16TEST	1237
8.268.1.17TEST	1237
8.268.1.18TEST	1237
8.268.1.19TEST	1237
8.268.1.20TEST	1237
8.268.1.21TEST	1237

8.268.1.22TEST	1237
8.269/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SpatialMeasureTest.hpp File Reference	1237
8.269.1 Function Documentation	1238
8.269.1.1 TEST	1238
8.269.1.2 TEST	1238
8.269.1.3 TEST	1238
8.269.1.4 TEST	1238
8.269.1.5 TEST	1239
8.269.1.6 TEST	1239
8.269.1.7 TEST	1239
8.269.1.8 TEST	1239
8.269.1.9 TEST	1239
8.269.1.10TEST	1239
8.269.1.11TEST	1239
8.269.1.12TEST	1239
8.270/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetOperationTest.hpp File Reference	1240
8.270.1 Function Documentation	1240
8.270.1.1 TEST	1240
8.270.1.2 TEST	1240
8.270.1.3 TEST	1240
8.270.1.4 TEST	1240
8.271/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSpecificTest.hpp File Reference	1241
8.271.1 Function Documentation	1241
8.271.1.1 TEST	1241
8.271.1.2 TEST	1241
8.271.1.3 TEST	1241
8.272/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSubsetOperationTest.hpp File Reference	1241
8.272.1 Function Documentation	1242
8.272.1.1 TEST	1242
8.272.1.2 TEST	1242
8.272.1.3 TEST	1242

8.272.1.4 TEST	1242
8.272.1.5 TEST	1243
8.272.1.6 TEST	1243
8.272.1.7 TEST	1243
8.272.1.8 TEST	1243
8.272.1.9 TEST	1243
8.272.1.10TEST	1243
8.273/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetTest.hpp File Reference	1243
8.273.1 Function Documentation	1244
8.273.1.1 TEST	1244
8.273.1.2 TEST	1244
8.273.1.3 TEST	1244
8.274/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetMeasureTest.hpp File Reference	1244
8.274.1 Function Documentation	1245
8.274.1.1 TEST	1245
8.274.1.2 TEST	1245
8.274.1.3 TEST	1245
8.275/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetTest.hpp File Reference	1245
8.275.1 Function Documentation	1246
8.275.1.1 TEST	1246
8.275.1.2 TEST	1246
8.275.1.3 TEST	1246
8.275.1.4 TEST	1246
8.275.1.5 TEST	1246
8.275.1.6 TEST	1246
8.275.1.7 TEST	1246
8.275.1.8 TEST	1247
8.275.1.9 TEST	1247
8.275.1.10TEST	1247
8.275.1.11TEST	1247
8.275.1.12TEST	1247
8.275.1.13TEST	1247

8.275.1.14 TEST	1247
8.275.1.15 TEST	1247
8.276/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericFilterTest.hpp File Reference . . .	1248
8.276.1 Function Documentation	1248
8.276.1.1 TEST	1248
8.276.1.2 TEST	1248
8.276.1.3 TEST	1248
8.276.1.4 TEST	1248
8.276.1.5 TEST	1249
8.276.1.6 TEST	1249
8.277/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericMeasureTest.hpp File Reference . .	1249
8.277.1 Function Documentation	1249
8.277.1.1 TEST	1249
8.277.1.2 TEST	1250
8.277.1.3 TEST	1250
8.277.1.4 TEST	1250
8.277.1.5 TEST	1250
8.277.1.6 TEST	1250
8.277.1.7 TEST	1250
8.277.1.8 TEST	1250
8.278/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericNumericTest.hpp File Reference . .	1251
8.278.1 Function Documentation	1251
8.278.1.1 TEST	1251
8.278.1.2 TEST	1251
8.278.1.3 TEST	1251
8.278.1.4 TEST	1251
8.278.1.5 TEST	1252
8.278.1.6 TEST	1252
8.279/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySpatialConstraintTest.hpp File Reference . .	1252
8.279.1 Function Documentation	1252
8.279.1.1 TEST	1252

8.279.1.2 TEST	1253
8.279.1.3 TEST	1253
8.279.1.4 TEST	1253
8.279.1.5 TEST	1253
8.279.1.6 TEST	1253
8.279.1.7 TEST	1253
8.279.1.8 TEST	1253
8.279.1.9 TEST	1253
8.280/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetMeasureTest.hpp File Reference . . .	1254
8.280.1 Function Documentation	1254
8.280.1.1 TEST	1254
8.280.1.2 TEST	1254
8.280.1.3 TEST	1254
8.280.1.4 TEST	1254
8.281/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetTest.hpp File Reference	1255
8.281.1 Function Documentation	1255
8.281.1.1 TEST	1255
8.281.1.2 TEST	1255
8.281.1.3 TEST	1255
8.281.1.4 TEST	1255
8.281.1.5 TEST	1256
8.281.1.6 TEST	1256
8.282/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryTypeConstraintTest.hpp File Reference . . .	1256
8.282.1 Function Documentation	1256
8.282.1.1 TEST	1256
8.282.1.2 TEST	1257
8.282.1.3 TEST	1257
8.282.1.4 TEST	1257
8.282.1.5 TEST	1257
8.282.1.6 TEST	1257
8.282.1.7 TEST	1257

8.283/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UntilLogicPropertyTest.hpp File Reference	1257
8.283.1 Function Documentation	1258
8.283.1.1 TEST	1258
8.283.1.2 TEST	1258
8.283.1.3 TEST	1259
8.283.1.4 TEST	1259
8.283.1.5 TEST	1259
8.283.1.6 TEST	1259
8.283.1.7 TEST	1259
8.283.1.8 TEST	1259
8.283.1.9 TEST	1259
8.283.1.10TEST	1259
8.283.1.11TEST	1260
8.283.1.12TEST	1260
8.283.1.13TEST	1260
8.283.1.14TEST	1260
8.283.1.15TEST	1260
8.283.1.16TEST	1260
8.283.1.17TEST	1260
8.283.1.18TEST	1260
8.283.1.19TEST	1261
8.283.1.20TEST	1261
8.283.1.21TEST	1261
8.283.1.22TEST	1261
8.283.1.23TEST	1261
8.284/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateBayesianModelCheckerTest.hpp File Reference	1261
8.284.1 Function Documentation	1262
8.284.1.1 TEST_F	1262
8.284.1.2 TEST_F	1262
8.285/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateProbabilisticModelCheckerTest.hpp File Reference	1262

8.285.1 Function Documentation	1263
8.285.1.1 TEST_F	1263
8.286/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/BayesianModelCheckerTest.hpp File Reference	1263
8.286.1 Function Documentation	1264
8.286.1.1 TEST_F	1264
8.286.1.2 TEST_F	1264
8.287/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckerTest.hpp File Reference	1264
8.287.1 Variable Documentation	1264
8.287.1.1 INPUT_LOGIC_PROPERTY	1265
8.288/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckingTest.cpp File Reference	1265
8.288.1 Function Documentation	1265
8.288.1.1 main	1265
8.289/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ProbabilisticBlackBoxModelCheckerTest.hpp File Reference	1265
8.289.1 Function Documentation	1266
8.289.1.1 TEST_F	1266
8.290/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/StatisticalModelCheckerTest.hpp File Reference	1266
8.290.1 Function Documentation	1267
8.290.1.1 TEST_F	1267
8.290.1.2 TEST_F	1267
8.291/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File Reference	1267
8.291.1 Function Documentation	1271
8.291.1.1 TEST_F	1271
8.291.1.2 TEST_F	1271
8.291.1.3 TEST_F	1271
8.291.1.4 TEST_F	1271
8.291.1.5 TEST_F	1271
8.291.1.6 TEST_F	1271
8.291.1.7 TEST_F	1272
8.291.1.8 TEST_F	1272

8.291.1.9 TEST_F	1272
8.291.1.10TEST_F	1272
8.291.1.11TEST_F	1272
8.291.1.12TEST_F	1272
8.291.1.13TEST_F	1272
8.291.1.14TEST_F	1272
8.291.1.15TEST_F	1272
8.291.1.16TEST_F	1272
8.291.1.17TEST_F	1273
8.291.1.18TEST_F	1273
8.291.1.19TEST_F	1273
8.291.1.20TEST_F	1273
8.291.1.21TEST_F	1273
8.291.1.22TEST_F	1273
8.291.1.23TEST_F	1273
8.291.1.24TEST_F	1273
8.291.1.25TEST_F	1273
8.291.1.26TEST_F	1273
8.291.1.27TEST_F	1274
8.291.1.28TEST_F	1274
8.291.1.29TEST_F	1274
8.291.1.30TEST_F	1274
8.291.1.31TEST_F	1274
8.291.1.32TEST_F	1274
8.291.1.33TEST_F	1274
8.291.1.34TEST_F	1274
8.291.1.35TEST_F	1274
8.291.1.36TEST_F	1274
8.291.1.37TEST_F	1275
8.291.1.38TEST_F	1275
8.291.1.39TEST_F	1275
8.291.1.40TEST_F	1275
8.291.1.41TEST_F	1275
8.291.1.42TEST_F	1275

CONTENTS**cxxxii**

8.291.1.43TEST_F	1275
8.291.1.44TEST_F	1275
8.291.1.45TEST_F	1275
8.291.1.46TEST_F	1275
8.291.1.47TEST_F	1276
8.291.1.48TEST_F	1276
8.291.1.49TEST_F	1276
8.291.1.50TEST_F	1276
8.291.1.51TEST_F	1276
8.291.1.52TEST_F	1276
8.291.1.53TEST_F	1276
8.291.1.54TEST_F	1276
8.291.1.55TEST_F	1276
8.291.1.56TEST_F	1276
8.291.1.57TEST_F	1277
8.291.1.58TEST_F	1277
8.291.1.59TEST_F	1277
8.291.1.60TEST_F	1277
8.291.1.61TEST_F	1277
8.291.1.62TEST_F	1277
8.291.1.63TEST_F	1277
8.291.1.64TEST_F	1277
8.291.1.65TEST_F	1277
8.291.1.66TEST_F	1277
8.291.1.67TEST_F	1278
8.291.1.68TEST_F	1278
8.291.1.69TEST_F	1278
8.291.1.70TEST_F	1278
8.291.1.71TEST_F	1278
8.291.1.72TEST_F	1278
8.291.1.73TEST_F	1278
8.291.1.74TEST_F	1278
8.291.1.75TEST_F	1278
8.291.1.76TEST_F	1278

8.291.1.77TEST_F	1279
8.291.1.78TEST_F	1279
8.291.1.79TEST_F	1279
8.291.1.80TEST_F	1279
8.291.1.81TEST_F	1279
8.291.1.82TEST_F	1279
8.291.1.83TEST_F	1279
8.291.1.84TEST_F	1279
8.291.1.85TEST_F	1279
8.291.1.86TEST_F	1279
8.291.1.87TEST_F	1280
8.291.1.88TEST_F	1280
8.291.1.89TEST_F	1280
8.291.1.90TEST_F	1280
8.291.1.91TEST_F	1280
8.291.1.92TEST_F	1280
8.291.1.93TEST_F	1280
8.291.1.94TEST_F	1280
8.291.1.95TEST_F	1280
8.291.1.96TEST_F	1280
8.291.1.97TEST_F	1281
8.291.1.98TEST_F	1281
8.291.1.99TEST_F	1281
8.291.1.100TEST_F	1281
8.291.1.101TEST_F	1281
8.291.1.102TEST_F	1281
8.291.1.103TEST_F	1281
8.291.1.104TEST_F	1281
8.291.1.105TEST_F	1282
8.291.1.106TEST_F	1282
8.291.1.107TEST_F	1282
8.291.1.108TEST_F	1282
8.291.1.109TEST_F	1282
8.291.1.110TEST_F	1282

8.291.1.11 T EST_F	1282
8.291.1.11 P EST_F	1282
8.291.1.11 B EST_F	1282
8.291.1.11 M EST_F	1282
8.291.1.11 E EST_F	1282
8.291.1.11 T EST_F	1282
8.291.1.11 V EST_F	1283
8.291.1.11 B EST_F	1283
8.291.1.11 D EST_F	1283
8.291.1.12 T EST_F	1283
8.291.1.12 P EST_F	1283
8.291.1.12 B EST_F	1283
8.291.1.12 M EST_F	1283
8.291.1.12 E EST_F	1283
8.291.1.12 V EST_F	1284
8.291.1.12 B EST_F	1284
8.291.1.13 D EST_F	1284
8.291.1.13 T EST_F	1284
8.291.1.13 P EST_F	1284
8.291.1.13 B EST_F	1284
8.291.1.13 M EST_F	1284
8.291.1.13 E EST_F	1285
8.291.1.13 V EST_F	1285
8.291.1.13 B EST_F	1285
8.291.1.14 D EST_F	1285
8.291.1.14 T EST_F	1285
8.291.1.14 P EST_F	1285
8.291.1.14 B EST_F	1285
8.291.1.14 M EST_F	1285

8.291.1.14 TEST_F	1286
8.291.1.15 TEST_F	1286
8.292/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp File Reference	1287
8.292.1 Function Documentation	1290
8.292.1.1 TEST_F	1290
8.292.1.2 TEST_F	1291
8.292.1.3 TEST_F	1291
8.292.1.4 TEST_F	1291
8.292.1.5 TEST_F	1291
8.292.1.6 TEST_F	1291
8.292.1.7 TEST_F	1291
8.292.1.8 TEST_F	1291
8.292.1.9 TEST_F	1291
8.292.1.10 TEST_F	1291
8.292.1.11 TEST_F	1291
8.292.1.12 TEST_F	1292
8.292.1.13 TEST_F	1292
8.292.1.14 TEST_F	1292
8.292.1.15 TEST_F	1292
8.292.1.16 TEST_F	1292
8.292.1.17 TEST_F	1292
8.292.1.18 TEST_F	1292
8.292.1.19 TEST_F	1292
8.292.1.20 TEST_F	1292
8.292.1.21 TEST_F	1292

CONTENTS**cxxxv**

8.292.1.22TEST_F	1293
8.292.1.23TEST_F	1293
8.292.1.24TEST_F	1293
8.292.1.25TEST_F	1293
8.292.1.26TEST_F	1293
8.292.1.27TEST_F	1293
8.292.1.28TEST_F	1293
8.292.1.29TEST_F	1293
8.292.1.30TEST_F	1293
8.292.1.31TEST_F	1294
8.292.1.32TEST_F	1294
8.292.1.33TEST_F	1294
8.292.1.34TEST_F	1294
8.292.1.35TEST_F	1294
8.292.1.36TEST_F	1294
8.292.1.37TEST_F	1294
8.292.1.38TEST_F	1294
8.292.1.39TEST_F	1294
8.292.1.40TEST_F	1294
8.292.1.41TEST_F	1295
8.292.1.42TEST_F	1295
8.292.1.43TEST_F	1295
8.292.1.44TEST_F	1295
8.292.1.45TEST_F	1295
8.292.1.46TEST_F	1295
8.292.1.47TEST_F	1295
8.292.1.48TEST_F	1295
8.292.1.49TEST_F	1295
8.292.1.50TEST_F	1295
8.292.1.51TEST_F	1296
8.292.1.52TEST_F	1296
8.292.1.53TEST_F	1296
8.292.1.54TEST_F	1296
8.292.1.55TEST_F	1296

8.292.1.56TEST_F	1296
8.292.1.57TEST_F	1296
8.292.1.58TEST_F	1296
8.292.1.59TEST_F	1296
8.292.1.60TEST_F	1296
8.292.1.61TEST_F	1297
8.292.1.62TEST_F	1297
8.292.1.63TEST_F	1297
8.292.1.64TEST_F	1297
8.292.1.65TEST_F	1297
8.292.1.66TEST_F	1297
8.292.1.67TEST_F	1297
8.292.1.68TEST_F	1297
8.292.1.69TEST_F	1297
8.292.1.70TEST_F	1297
8.292.1.71TEST_F	1298
8.292.1.72TEST_F	1298
8.292.1.73TEST_F	1298
8.292.1.74TEST_F	1298
8.292.1.75TEST_F	1298
8.292.1.76TEST_F	1298
8.292.1.77TEST_F	1298
8.292.1.78TEST_F	1298
8.292.1.79TEST_F	1298
8.292.1.80TEST_F	1298
8.292.1.81TEST_F	1299
8.292.1.82TEST_F	1299
8.292.1.83TEST_F	1299
8.292.1.84TEST_F	1299
8.292.1.85TEST_F	1299
8.292.1.86TEST_F	1299
8.292.1.87TEST_F	1299
8.292.1.88TEST_F	1299
8.292.1.89TEST_F	1299

CONTENTS**cxxxvii**

8.292.1.90TEST_F	1299
8.292.1.91TEST_F	1300
8.292.1.92TEST_F	1300
8.292.1.93TEST_F	1300
8.292.1.94TEST_F	1300
8.292.1.95TEST_F	1300
8.292.1.96TEST_F	1300
8.292.1.97TEST_F	1300
8.292.1.98TEST_F	1300
8.292.1.99TEST_F	1300
8.292.1.100TEST_F	1300
8.292.1.101TEST_F	1301
8.292.1.102TEST_F	1301
8.292.1.103TEST_F	1301
8.292.1.104TEST_F	1301
8.292.1.105TEST_F	1301
8.292.1.106TEST_F	1301
8.292.1.107TEST_F	1301
8.292.1.108TEST_F	1301
8.292.1.109TEST_F	1301
8.292.1.110TEST_F	1301
8.292.1.111TEST_F	1302
8.292.1.112TEST_F	1302
8.292.1.113TEST_F	1302
8.292.1.114TEST_F	1302
8.292.1.115TEST_F	1302
8.292.1.116TEST_F	1302
8.292.1.117TEST_F	1302
8.292.1.118TEST_F	1302
8.292.1.119TEST_F	1302
8.292.1.120TEST_F	1302
8.292.1.121TEST_F	1303
8.292.1.122TEST_F	1303
8.292.1.123TEST_F	1303

8.292.1.12 T EST_F	1303
8.292.1.12 E EST_F	1303
8.292.1.12 S EST_F	1303
8.292.1.12 V EST_F	1303
8.292.1.12 B EST_F	1303
8.292.1.12 D EST_F	1303
8.292.1.13 T EST_F	1304
8.292.1.13 E EST_F	1304
8.292.1.13 S EST_F	1304
8.292.1.13 V EST_F	1304
8.292.1.13 B EST_F	1304
8.292.1.13 D EST_F	1304
8.292.1.14 T EST_F	1304
8.292.1.14 E EST_F	1305
8.292.1.14 S EST_F	1305
8.292.1.14 V EST_F	1305
8.292.1.14 B EST_F	1305
8.292.1.14 D EST_F	1305
8.292.1.15 T EST_F	1306
8.292.1.15 E EST_F	1306
8.292.1.15 S EST_F	1306
8.293/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File - Reference	1306
8.293.1 Function Documentation	1310

8.293.1.1 TEST_F	1310
8.293.1.2 TEST_F	1310
8.293.1.3 TEST_F	1311
8.293.1.4 TEST_F	1311
8.293.1.5 TEST_F	1311
8.293.1.6 TEST_F	1311
8.293.1.7 TEST_F	1311
8.293.1.8 TEST_F	1311
8.293.1.9 TEST_F	1311
8.293.1.10TEST_F	1311
8.293.1.11TEST_F	1311
8.293.1.12TEST_F	1311
8.293.1.13TEST_F	1312
8.293.1.14TEST_F	1312
8.293.1.15TEST_F	1312
8.293.1.16TEST_F	1312
8.293.1.17TEST_F	1312
8.293.1.18TEST_F	1312
8.293.1.19TEST_F	1312
8.293.1.20TEST_F	1312
8.293.1.21TEST_F	1312
8.293.1.22TEST_F	1312
8.293.1.23TEST_F	1313
8.293.1.24TEST_F	1313
8.293.1.25TEST_F	1313
8.293.1.26TEST_F	1313
8.293.1.27TEST_F	1313
8.293.1.28TEST_F	1313
8.293.1.29TEST_F	1313
8.293.1.30TEST_F	1313
8.293.1.31TEST_F	1313
8.293.1.32TEST_F	1313
8.293.1.33TEST_F	1314
8.293.1.34TEST_F	1314

8.293.1.35TEST_F	1314
8.293.1.36TEST_F	1314
8.293.1.37TEST_F	1314
8.293.1.38TEST_F	1314
8.293.1.39TEST_F	1314
8.293.1.40TEST_F	1314
8.293.1.41TEST_F	1314
8.293.1.42TEST_F	1314
8.293.1.43TEST_F	1315
8.293.1.44TEST_F	1315
8.293.1.45TEST_F	1315
8.293.1.46TEST_F	1315
8.293.1.47TEST_F	1315
8.293.1.48TEST_F	1315
8.293.1.49TEST_F	1315
8.293.1.50TEST_F	1315
8.293.1.51TEST_F	1315
8.293.1.52TEST_F	1315
8.293.1.53TEST_F	1316
8.293.1.54TEST_F	1316
8.293.1.55TEST_F	1316
8.293.1.56TEST_F	1316
8.293.1.57TEST_F	1316
8.293.1.58TEST_F	1316
8.293.1.59TEST_F	1316
8.293.1.60TEST_F	1316
8.293.1.61TEST_F	1316
8.293.1.62TEST_F	1316
8.293.1.63TEST_F	1317
8.293.1.64TEST_F	1317
8.293.1.65TEST_F	1317
8.293.1.66TEST_F	1317
8.293.1.67TEST_F	1317
8.293.1.68TEST_F	1317

8.293.1.69TEST_F	1317
8.293.1.70TEST_F	1317
8.293.1.71TEST_F	1317
8.293.1.72TEST_F	1317
8.293.1.73TEST_F	1318
8.293.1.74TEST_F	1318
8.293.1.75TEST_F	1318
8.293.1.76TEST_F	1318
8.293.1.77TEST_F	1318
8.293.1.78TEST_F	1318
8.293.1.79TEST_F	1318
8.293.1.80TEST_F	1318
8.293.1.81TEST_F	1318
8.293.1.82TEST_F	1318
8.293.1.83TEST_F	1319
8.293.1.84TEST_F	1319
8.293.1.85TEST_F	1319
8.293.1.86TEST_F	1319
8.293.1.87TEST_F	1319
8.293.1.88TEST_F	1319
8.293.1.89TEST_F	1319
8.293.1.90TEST_F	1319
8.293.1.91TEST_F	1319
8.293.1.92TEST_F	1319
8.293.1.93TEST_F	1320
8.293.1.94TEST_F	1320
8.293.1.95TEST_F	1320
8.293.1.96TEST_F	1320
8.293.1.97TEST_F	1320
8.293.1.98TEST_F	1320
8.293.1.99TEST_F	1320
8.293.1.100TEST_F	1320
8.293.1.101TEST_F	1320
8.293.1.102TEST_F	1320

8.293.1.10 BEST _F	1321
8.293.1.10 TEST _F	1321
8.293.1.10 EST _F	1321
8.293.1.10 EEST _F	1321
8.293.1.10 TTEST _F	1321
8.293.1.10 TEST _F	1321
8.293.1.10 TEST _F	1321
8.293.1.11 TEST _F	1321
8.293.1.11 TEST _F	1322
8.293.1.11 TEST _F	1322
8.293.1.11 EST _F	1322
8.293.1.11 EEST _F	1322
8.293.1.11 TTEST _F	1322
8.293.1.11 TEST _F	1322
8.293.1.11 TEST _F	1322
8.293.1.12 TEST _F	1322
8.293.1.12 TEST _F	1323
8.293.1.12 TEST _F	1323
8.293.1.12 EST _F	1323
8.293.1.12 EEST _F	1323
8.293.1.12 TTEST _F	1323
8.293.1.12 TEST _F	1323
8.293.1.12 TEST _F	1323
8.293.1.13 TEST _F	1324
8.293.1.13 TEST _F	1324
8.293.1.13 EST _F	1324
8.293.1.13 EEST _F	1324
8.293.1.13 TTEST _F	1324
8.293.1.13 TEST _F	1324

8.293.1.13 T EST_F	1324
8.293.1.13 B EST_F	1324
8.293.1.13 D EST_F	1325
8.293.1.14 D EST_F	1325
8.293.1.14 T EST_F	1325
8.293.1.14 P EST_F	1325
8.293.1.14 B EST_F	1325
8.293.1.14 M EST_F	1325
8.293.1.14 E EST_F	1325
8.293.1.14 G EST_F	1325
8.293.1.14 T EST_F	1325
8.293.1.14 R EST_F	1326
8.293.1.15 D EST_F	1326
8.293.1.15 T EST_F	1326
8.293.1.15 P EST_F	1326
8.293.1.15 B EST_F	1326
8.294/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.cpp File Reference	1326
8.294.1 Function Documentation	1327
8.294.1.1 main	1327
8.295/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ParserEvaluationTest.hpp File Reference	1327
8.296/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File Reference	1327
8.296.1 Function Documentation	1331
8.296.1.1 TEST_F	1331
8.296.1.2 TEST_F	1331
8.296.1.3 TEST_F	1332
8.296.1.4 TEST_F	1332
8.296.1.5 TEST_F	1332
8.296.1.6 TEST_F	1332
8.296.1.7 TEST_F	1332
8.296.1.8 TEST_F	1332
8.296.1.9 TEST_F	1332

8.296.1.10TEST_F	1332
8.296.1.11TEST_F	1332
8.296.1.12TEST_F	1332
8.296.1.13TEST_F	1333
8.296.1.14TEST_F	1333
8.296.1.15TEST_F	1333
8.296.1.16TEST_F	1333
8.296.1.17TEST_F	1333
8.296.1.18TEST_F	1333
8.296.1.19TEST_F	1333
8.296.1.20TEST_F	1333
8.296.1.21TEST_F	1333
8.296.1.22TEST_F	1333
8.296.1.23TEST_F	1334
8.296.1.24TEST_F	1334
8.296.1.25TEST_F	1334
8.296.1.26TEST_F	1334
8.296.1.27TEST_F	1334
8.296.1.28TEST_F	1334
8.296.1.29TEST_F	1334
8.296.1.30TEST_F	1334
8.296.1.31TEST_F	1334
8.296.1.32TEST_F	1334
8.296.1.33TEST_F	1335
8.296.1.34TEST_F	1335
8.296.1.35TEST_F	1335
8.296.1.36TEST_F	1335
8.296.1.37TEST_F	1335
8.296.1.38TEST_F	1335
8.296.1.39TEST_F	1335
8.296.1.40TEST_F	1335
8.296.1.41TEST_F	1335
8.296.1.42TEST_F	1335
8.296.1.43TEST_F	1336

8.296.1.44TEST_F	1336
8.296.1.45TEST_F	1336
8.296.1.46TEST_F	1336
8.296.1.47TEST_F	1336
8.296.1.48TEST_F	1336
8.296.1.49TEST_F	1336
8.296.1.50TEST_F	1336
8.296.1.51TEST_F	1336
8.296.1.52TEST_F	1336
8.296.1.53TEST_F	1337
8.296.1.54TEST_F	1337
8.296.1.55TEST_F	1337
8.296.1.56TEST_F	1337
8.296.1.57TEST_F	1337
8.296.1.58TEST_F	1337
8.296.1.59TEST_F	1337
8.296.1.60TEST_F	1337
8.296.1.61TEST_F	1337
8.296.1.62TEST_F	1337
8.296.1.63TEST_F	1338
8.296.1.64TEST_F	1338
8.296.1.65TEST_F	1338
8.296.1.66TEST_F	1338
8.296.1.67TEST_F	1338
8.296.1.68TEST_F	1338
8.296.1.69TEST_F	1338
8.296.1.70TEST_F	1338
8.296.1.71TEST_F	1338
8.296.1.72TEST_F	1338
8.296.1.73TEST_F	1339
8.296.1.74TEST_F	1339
8.296.1.75TEST_F	1339
8.296.1.76TEST_F	1339
8.296.1.77TEST_F	1339

8.296.1.78TEST_F	1339
8.296.1.79TEST_F	1339
8.296.1.80TEST_F	1339
8.296.1.81TEST_F	1339
8.296.1.82TEST_F	1339
8.296.1.83TEST_F	1340
8.296.1.84TEST_F	1340
8.296.1.85TEST_F	1340
8.296.1.86TEST_F	1340
8.296.1.87TEST_F	1340
8.296.1.88TEST_F	1340
8.296.1.89TEST_F	1340
8.296.1.90TEST_F	1340
8.296.1.91TEST_F	1340
8.296.1.92TEST_F	1340
8.296.1.93TEST_F	1341
8.296.1.94TEST_F	1341
8.296.1.95TEST_F	1341
8.296.1.96TEST_F	1341
8.296.1.97TEST_F	1341
8.296.1.98TEST_F	1341
8.296.1.99TEST_F	1341
8.296.1.100TEST_F	1341
8.296.1.101TEST_F	1341
8.296.1.102TEST_F	1341
8.296.1.103TEST_F	1342
8.296.1.104TEST_F	1342
8.296.1.105TEST_F	1342
8.296.1.106TEST_F	1342
8.296.1.107TEST_F	1342
8.296.1.108TEST_F	1342
8.296.1.109TEST_F	1342
8.296.1.110TEST_F	1342
8.296.1.111TEST_F	1342

8.296.1.11 TEST_F	1342
8.296.1.11 BEST_F	1343
8.296.1.11 TTEST_F	1343
8.296.1.11 TEST_F	1343
8.296.1.11 TEST_F	1343
8.296.1.11 TTEST_F	1343
8.296.1.11 BEST_F	1343
8.296.1.11 TTEST_F	1343
8.296.1.12 TEST_F	1343
8.296.1.12 TTEST_F	1343
8.296.1.12 TEST_F	1343
8.296.1.12 BEST_F	1344
8.296.1.12 TTEST_F	1344
8.296.1.12 TEST_F	1344
8.296.1.12 BEST_F	1344
8.296.1.12 TTEST_F	1344
8.296.1.12 TEST_F	1344
8.296.1.12 BEST_F	1344
8.296.1.12 TTEST_F	1344
8.296.1.13 TEST_F	1344
8.296.1.13 PTEST_F	1345
8.296.1.13 BEST_F	1345
8.296.1.13 TTEST_F	1345
8.296.1.13 TEST_F	1345
8.296.1.13 BEST_F	1345
8.296.1.13 TTEST_F	1345
8.296.1.13 TEST_F	1345
8.296.1.13 BEST_F	1345
8.296.1.13 TTEST_F	1345
8.296.1.14 TEST_F	1346
8.296.1.14 PTEST_F	1346
8.296.1.14 BEST_F	1346
8.296.1.14 TTEST_F	1346
8.296.1.14 TEST_F	1346

8.296.1.14 TEST_F	1346
8.296.1.15 TEST_F	1346
8.296.1.15 TEST_F	1347
8.296.1.15 TEST_F	1347
8.296.1.15 TEST_F	1347
8.297/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.hpp File Reference	1347
8.297.1 Variable Documentation	1348
8.297.1.1 ERR_MSG_TEST	1348
8.298/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/InputStringParser.hpp File Reference	1348
8.299/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.cpp File Reference	1348
8.299.1 Function Documentation	1349
8.299.1.1 main	1349
8.300/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.hpp File Reference	1349
8.300.1 Function Documentation	1350
8.300.1.1 TEST	1350
8.300.1.2 TEST	1350
8.300.1.3 TEST	1350
8.300.1.4 TEST	1350
8.301/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/AnnularSector.hpp File Reference	1351
8.302/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/CartesianToPolarConverter.hpp File Reference	1351
8.303/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarCsvToInputFilesConverter.hpp File Reference	1352
8.304/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarGnuplotScriptGenerator.hpp File Reference	1352
8.305/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src-/AnnularSector.cpp File Reference	1353
8.305.1 Variable Documentation	1354
8.305.1.1 SEPARATOR	1354

8.306/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- CartesianToPolarConverter.cpp File Reference	1354
8.307/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- MapCartesianToPolarScript.cpp File Reference	1354
8.307.1 Function Documentation	1355
8.307.1.1 areValidParameters	1355
8.307.1.2 initArgumentsConfig	1355
8.307.1.3 isValidOutputType	1355
8.307.1.4 main	1356
8.307.1.5 printHelpInformation	1356
8.307.1.6 printWrongParameters	1356
8.308/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- PolarCsvToInputFilesConverter.cpp File Reference	1356
8.309/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- PolarGnuplotScriptGenerator.cpp File Reference	1357
8.310/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- PolarGnuplotScriptGenerator.in.cpp File Reference	1357
8.311/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- PolarMapCsvToInputFiles.cpp File Reference	1357
8.311.1 Function Documentation	1358
8.311.1.1 areValidParameters	1358
8.311.1.2 initArgumentsConfig	1359
8.311.1.3 isValidNrOfConcentrationsForPosition	1359
8.311.1.4 main	1359
8.311.1.5 printHelpInformation	1359
8.311.1.6 printWrongParameters	1359
8.311.1.7 setLogScaling	1359
8.311.1.8 setNumberIteratorType	1359
8.311.1.9 setSelectedConcentrationIndex	1360
8.312/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/- CartesianToConcentrationsConverter.hpp File Reference	1360
8.313/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/- RectangularCsvToInputFilesConverter.hpp File Reference	1360
8.314/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/- RectangularEntityCsvToInputFilesConverter.hpp File Reference	1361
8.315/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/- RectangularGnuplotScriptGenerator.hpp File Reference	1362

8.316/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- CartesianToConcentrationsConverter.cpp File Reference	1362
8.317/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- MapCartesianToScript.cpp File Reference	1363
8.317.1 Function Documentation	1363
8.317.1.1 areValidParameters	1364
8.317.1.2 initArgumentsConfig	1364
8.317.1.3 main	1364
8.317.1.4 printHelpInformation	1364
8.317.1.5 printWrongParameters	1364
8.318/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularCsvToInputFilesConverter.cpp File Reference	1364
8.319/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularEntityCsvToInputFilesConverter.cpp File Reference	1365
8.320/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularGnuplotScriptGenerator.cpp File Reference	1365
8.321/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularGnuplotScriptGenerator.in.cpp File Reference	1366
8.322/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularMapCsvToInputFiles.cpp File Reference	1366
8.322.1 Function Documentation	1367
8.322.1.1 areValidParameters	1367
8.322.1.2 initArgumentsConfig	1368
8.322.1.3 isValidNrOfConcentrationsForPosition	1368
8.322.1.4 main	1368
8.322.1.5 printHelpInformation	1368
8.322.1.6 printWrongParameters	1368
8.322.1.7 setLogScaling	1368
8.322.1.8 setNumberIteratorType	1368
8.322.1.9 setSelectedConcentrationIndex	1368
8.323/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/- RectangularMapEntityCsvToInputFiles.cpp File Reference	1369
8.323.1 Function Documentation	1369
8.323.1.1 areValidParameters	1369
8.323.1.2 initArgumentsConfig	1370
8.323.1.3 main	1370

CONTENTS**cli**

8.323.1.4 printHelpInformation	1370
8.323.1.5 printWrongParameters	1370
8.323.1.6 setNumberIteratorType	1370

Chapter 1

Multiscale

1.1 Brief description

The "Multiscale" software is a multiscale model checker implemented during the Ph-D 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:

<code>multiplescale</code>	37
<code>multiplescale::analysis</code>	41
<code>multiplescale::verification</code>	43
<code>multiplescale::verification::subsetspecific</code>	63
<code>multiplescale::video</code>	65
<code>multiplescaletest</code>	66
<code>multiplescaletest::verification</code>	67

Chapter 3

Class Index

3.1 Class Hierarchy

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

multiscale::verification::AbstractSyntaxTree	69
multiscale::AdditionOperation	73
multiscale::verification::AndConstraintAttribute	77
multiscale::verification::AndLogicPropertyAttribute	78
multiscale::video::AnnularSector	79
multiscale::verification::BinaryNumericFilterAttribute	153
multiscale::verification::BinaryNumericMeasureAttribute	155
multiscale::verification::BinaryNumericMeasureTypeParser	156
multiscale::verification::BinaryNumericNumericAttribute	157
multiscale::verification::BinarySubsetAttribute	158
multiscale::verification::BinarySubsetMeasureAttribute	160
multiscale::verification::BinarySubsetMeasureTypeParser	161
multiscale::video::CartesianToConcentrationsConverter	167
multiscale::video::CartesianToPolarConverter	174
multiscale::analysis::CircularityMeasure	182
multiscale::verification::CommandLineModelChecking	219
multiscale::verification::ComparatorAttribute	262
multiscale::verification::ComparatorEvaluator	263
multiscale::verification::ComparatorNonEqualTypeParser	264
multiscale::verification::ComparatorTypeParser	265
multiscale::ConsolePrinter	269
multiscale::verification::ConstraintAttribute	279
multiscale::verification::ConstraintEvaluator	281
multiscale::verification::ConstraintVisitor	285
multiscale::analysis::DataPoint	294
EuclideanDataPoint	345
multiscale::analysis::Entity	335
multiscale::analysis::DBSCAN	296
multiscale::analysis::Detector	304

multiscale::analysis::ClusterDetector	204
multiscale::analysis::SimulationClusterDetector	810
multiscale::analysis::RegionDetector	770
multiscale::verification::DifferenceAttribute	327
multiscale::Distribution	328
multiscale::BetaDistribution	149
multiscale::BinomialDistribution	161
multiscale::DivisionOperation	332
multiscale::verification::EquivalenceConstraintAttribute	343
multiscale::verification::EquivalenceLogicPropertyAttribute	343
multiscale::ExceptionHandler	348
multiscale::Filesystem	351
multiscale::verification::FilterNumericMeasureAttribute	354
multiscale::verification::FilterNumericVisitor	355
multiscale::verification::FilterSubsetAttribute	361
multiscale::verification::FutureLogicPropertyAttribute	362
multiscale::Geometry2D	363
multiscale::verification::GlobalLogicPropertyAttribute	381
grammar	383
multiscale::verification::ParserGrammar< Iterator >	618
multiscale::verification::ImplicationConstraintAttribute	384
multiscale::verification::ImplicationLogicPropertyAttribute	384
multiscale::verification::LogicPropertyAttribute	401
multiscale::verification::LogicPropertyDataReader	404
multiscale::verification::LogicPropertyVisitor	410
multiscale::analysis::MatFactory	429
multiscale::analysis::CircularMatFactory	183
multiscale::analysis::RectangularMatFactory	749
multiscale::MinEnclosingTriangleFinder	436
multiscale::verification::ModelChecker	468
multiscale::verification::ApproximateBayesianModelChecker	82
multiscale::verification::ApproximateProbabilisticModelChecker	106
multiscale::verification::BayesianModelChecker	124
multiscale::verification::ProbabilisticBlackBoxModelChecker	698
multiscale::verification::StatisticalModelChecker	896
multiscale::verification::ModelCheckerFactory	482
multiscale::verification::ApproximateBayesianModelCheckerFactory	97
multiscale::verification::ApproximateProbabilisticModelCheckerFactory	116
multiscale::verification::BayesianModelCheckerFactory	141
multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory	703
multiscale::verification::StatisticalModelCheckerFactory	913
multiscale::verification::ModelCheckingManager	496
multiscale::verification::ModelCheckingOutputWriter	513
multiscale::MultiplicationOperation	537
multiscale::MultiscaleException	538
multiscale::AlgorithmException	74
multiscale::UnexpectedBehaviourException	973
multiscale::verification::ModelCheckingException	490

multiscale::verification::ModelCheckingHelpRequestException	493
multiscale::verification::SpatialTemporalException	876
multiscale::UnimplementedMethodException	977
multiscale::IOException	391
multiscale::FileOpenException	348
multiscale::InvalidInputException	388
multiscale::NumericException	585
multiscale::RuntimeException	804
multiscale::IndexOutOfBoundsException	385
multiscale::TestException	939
multiscaletest::MultiscaleTest	542
multiscaletest::MinEnclosingTriangleFinderTest	460
multiscaletest::ModelCheckerTest	483
multiscaletest::ApproximateBayesianModelCheckerTest	101
multiscaletest::ApproximateProbabilisticModelCheckerTest	120
multiscaletest::BayesianModelCheckerTest	144
multiscaletest::ProbabilisticBlackBoxModelCheckerTest	706
multiscaletest::StatisticalModelCheckerTest	916
multiscaletest::TraceEvaluationTest	958
multiscaletest::CompleteTraceTest	266
multiscaletest::EmptyTraceTest	332
multiscaletest::NumericStateVariableTraceTest	597
multiscaletest::SpatialEntitiesTraceTest	821
multiscale::verification::NextKLogicPropertyAttribute	543
multiscale::verification::NextLogicPropertyAttribute	544
multiscale::verification::Nil	545
multiscale::verification::NotConstraintAttribute	546
multiscale::verification::NotLogicPropertyAttribute	546
multiscale::NumberIterator	547
multiscale::LexicographicNumberIterator	394
multiscale::StandardNumberIterator	890
multiscale::Numeric	552
multiscale::verification::NumericEvaluator	582
multiscale::verification::NumericMeasureAttribute	588
multiscale::verification::NumericNumericComparisonAttribute	589
multiscale::NumericRangeManipulator	591
multiscale::verification::NumericSpatialAttribute	592
multiscale::verification::NumericSpatialNumericComparisonAttribute	593
multiscale::verification::NumericStateVariableAttribute	595
multiscale::verification::NumericVisitor	600
multiscale::OperatingSystem	608
multiscale::verification::OrConstraintAttribute	612
multiscale::verification::OrLogicPropertyAttribute	613
multiscale::verification::Parser	614
multiscale::verification::ParserGrammarExceptionHandler	662
multiscale::verification::ParserGrammarExtraInputException	665
multiscale::verification::ParserGrammarProbabilityException	667
multiscale::verification::ParserGrammarUnexpectedTokenException	670

multiscale::verification::ParserGrammarUnparseableInputException	673
multiscale::video::PolarCsvToInputFilesConverter	675
multiscale::video::PolarGnuplotScriptGenerator	689
multiscale::verification::PrimaryConstraintAttribute	696
multiscale::verification::PrimaryLogicPropertyAttribute	697
multiscale::verification::PrimaryNumericMeasureAttribute	697
multiscale::verification::ProbabilisticLogicPropertyAttribute	709
multiscale::verification::ProbabilityErrorHandler	713
multiscale::verification::QuaternarySubsetAttribute	714
multiscale::verification::QuaternarySubsetMeasureAttribute	716
multiscale::verification::QuaternarySubsetMeasureTypeParser	717
multiscale::video::RectangularCsvToInputFilesConverter	718
multiscale::video::RectangularEntityCsvToInputFilesConverter	731
multiscale::video::RectangularGnuplotScriptGenerator	742
multiscale::verification::ProbabilityErrorHandler::result< typename, type- name, typename >	799
multiscale::verification::UnexpectedTokenErrorHandler::result< typename, typename, typename >	799
multiscale::RGBColourGenerator	800
multiscale::analysis::Silhouette	807
multiscale::verification::SpatialEntity	823
multiscale::verification::Cluster	201
multiscale::verification::Region	766
multiscale::verification::SpatialEntityEvaluator	837
multiscale::analysis::SpatialEntityPseudo3D	838
multiscale::analysis::Cluster	190
multiscale::analysis::Region	756
multiscale::verification::SpatialMeasureAttribute	854
multiscale::verification::SpatialMeasureEvaluator	855
multiscale::verification::SpatialMeasureTypeParser	856
multiscale::verification::SpatialNumericComparisonAttribute	857
multiscale::verification::SpatialTemporalDataReader	858
multiscale::verification::SpatialTemporalTrace	879
multiscale::verification::StateVariableAttribute	895
multiscale::StringManipulator	920
multiscale::verification::SubsetAttribute	924
multiscale::verification::SubsetOperationAttribute	924
multiscale::verification::SubsetOperationTypeParser	925
multiscale::verification::SubsetSpecificAttribute	926
multiscale::verification::SubsetSpecificTypeParser	927
multiscale::verification::SubsetSubsetOperationAttribute	927
multiscale::verification::SubsetVisitor	929
multiscale::SubtractionOperation	934
multiscale::verification::TernarySubsetAttribute	935
multiscale::verification::TernarySubsetMeasureAttribute	937
multiscale::verification::TernarySubsetMeasureTypeParser	938
multiscale::verification::TimePoint	942
multiscale::verification::TimePointEvaluator	956
multiscale::verification::UnaryNumericFilterAttribute	962

multiscale::verification::UnaryNumericMeasureAttribute	964
multiscale::verification::UnaryNumericMeasureTypeParser	965
multiscale::verification::UnaryNumericNumericAttribute	965
multiscale::verification::UnarySpatialConstraintAttribute	967
multiscale::verification::UnarySubsetAttribute	968
multiscale::verification::UnarySubsetMeasureAttribute	970
multiscale::verification::UnarySubsetMeasureTypeParser	971
multiscale::verification::UnaryTypeConstraintAttribute	971
multiscale::verification::UnexpectedTokenErrorHandler	975
multiscale::verification::UntilLogicPropertyAttribute	980
multiscale::XmlValidator::XmlValidationErrorHandler	982
multiscale::XmlValidator	987

Chapter 4

Class Index

4.1 Class List

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

<code>multiscale::verification::AbstractSyntaxTree</code>	Class used for representing an abstract syntax tree	69
<code>multiscale::AdditionOperation</code>	Functor representing an addition operation	73
<code>multiscale::AlgorithmException</code>	Class for representing algorithm exceptions	74
<code>multiscale::verification::AndConstraintAttribute</code>	Class for representing an "and" constraint attribute	77
<code>multiscale::verification::AndLogicPropertyAttribute</code>	Class for representing an "and" logic property attribute	78
<code>multiscale::video::AnnularSector</code>	An annular sector is the basic element in the considered circular geometry	79
<code>multiscale::verification::ApproximateBayesianModelChecker</code>	Class used to run approximate Bayesian model checking tasks	82
<code>multiscale::verification::ApproximateBayesianModelCheckerFactory</code>	Class for creating <code>ApproximateBayesianModelChecker</code> instances	97
<code>multiscaletest::ApproximateBayesianModelCheckerTest</code>	Class for testing the approximate Bayesian model checker	101
<code>multiscale::verification::ApproximateProbabilisticModelChecker</code>	Class used to run approximate probabilistic model checking tasks	106
<code>multiscale::verification::ApproximateProbabilisticModelCheckerFactory</code>	Class for creating <code>ApproximateProbabilisticModelChecker</code> instances	116
<code>multiscaletest::ApproximateProbabilisticModelCheckerTest</code>	Class for testing the approximate probabilistic model checker	120
<code>multiscale::verification::BayesianModelChecker</code>	Class used to run Bayesian model checking tasks	124
<code>multiscale::verification::BayesianModelCheckerFactory</code>	Class for creating <code>BayesianModelChecker</code> instances	141

multiscaletest::BayesianModelCheckerTest	Class for testing the Bayesian model checker	144
multiscale::BetaDistribution	Class for analysing Beta distributed data	149
multiscale::verification::BinaryNumericFilterAttribute	Class for representing a binary numeric filter attribute	153
multiscale::verification::BinaryNumericMeasureAttribute	Class for representing a binary numeric measure attribute	155
multiscale::verification::BinaryNumericMeasureTypeParser	Symbol table and parser for the binary numeric measure type	156
multiscale::verification::BinaryNumericNumericAttribute	Class for representing a binary numeric numeric measure attribute	157
multiscale::verification::BinarySubsetAttribute	Class for representing a binary subset attribute	158
multiscale::verification::BinarySubsetMeasureAttribute	Class for representing a binary subset measure attribute	160
multiscale::verification::BinarySubsetMeasureTypeParser	Symbol table and parser for the binary subset measure type	161
multiscale::BinomialDistribution	Class for analysing Binomial distributed data	161
multiscale::video::CartesianToConcentrationsConverter	Scale the values of the rectangular geometry grid cells	167
multiscale::video::CartesianToPolarConverter	Converter from the rectangular geometry grid cells to annular sectors	174
multiscale::analysis::CircularityMeasure	Class for computing the circularity measure for the given collection of points	182
multiscale::analysis::CircularMatFactory	Class for creating a Mat object considering a circular grid	183
multiscale::analysis::Cluster	Class for representing a cluster of entities in an image	190
multiscale::verification::Cluster	Class for representing a cluster	201
multiscale::analysis::ClusterDetector	Class for detecting clusters in 2D images	204
multiscale::verification::CommandLineModelChecking	Class for running model checkers from the command line	219
multiscale::verification::ComparatorAttribute	Class for representing a comparator attribute	262
multiscale::verification::ComparatorEvaluator	Class for evaluating comparison expressions	263
multiscale::verification::ComparatorNonEqualTypeParser	Symbol table and parser for the comparator type which does not accept the "=" symbol	264
multiscale::verification::ComparatorTypeParser	Symbol table and parser for the comparator type	265
multiscaletest::CompleteTraceTest	Class for testing evaluation of complete traces containing both nu- meric state variables and spatial entities	266

multiscale::ConsolePrinter	Class used to print (coloured) messages to the console	269
multiscale::verification::ConstraintAttribute	Class for representing a constraint attribute	279
multiscale::verification::ConstraintEvaluator	Class for evaluating constraint expressions	281
multiscale::verification::ConstraintVisitor	Class used to evaluate constraints	285
multiscale::analysis::DataPoint	Class for representing a data point	294
multiscale::analysis::DBSCAN	Class which implements an improved version of the DBSCAN algorithm	296
multiscale::analysis::Detector	Abstract class for detecting entities of interest in images	304
multiscale::verification::DifferenceAttribute	Class for representing a difference attribute	327
multiscale::Distribution		328
multiscale::DivisionOperation	Functor representing a division operation	332
multiscaletest::EmptyTraceTest	Class for testing evaluation of empty traces	332
multiscale::analysis::Entity	Class for representing an entity in an image (e.g. cell, organism etc.)	335
multiscale::verification::EquivalenceConstraintAttribute	Class for representing an "equivalence" constraint attribute	343
multiscale::verification::EquivalenceLogicPropertyAttribute	Class for representing an "equivalence" logic property attribute	343
EuclideanDataPoint		345
multiscale::ExceptionHandler	Exception handler class	348
multiscale::FileOpenException	Class for representing exceptions when opening a file	348
multiscale::Filesystem	Class containing methods for interacting with the filesystem	351
multiscale::verification::FilterNumericMeasureAttribute	Class for representing a filter numeric measure	354
multiscale::verification::FilterNumericVisitor	Class for evaluating filter numeric measures	355
multiscale::verification::FilterSubsetAttribute	Class for representing a filter subset attribute	361
multiscale::verification::FutureLogicPropertyAttribute	Class for representing a "future" logic property attribute	362
multiscale::Geometry2D	Two-dimensional geometric operations	363
multiscale::verification::GlobalLogicPropertyAttribute	Class for representing a "globally" logic property attribute	381
grammar		383
multiscale::verification::ImplicationConstraintAttribute	Class for representing an "implication" constraint attribute	384

multiscale::verification::ImplicationLogicPropertyAttribute	
Class for representing an "implication" logic property attribute	384
multiscale::IndexOutOfBoundsException	
Class for representing an index out of bounds exception	385
multiscale::InvalidInputException	
Class for representing invalid input exceptions	388
multiscale::IOException	
Class for representing input and output exceptions	391
multiscale::LexicographicNumberIterator	
Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "_"	394
multiscale::verification::LogicPropertyAttribute	
Class for representing a logic property attribute	401
multiscale::verification::LogicPropertyDataReader	
Class used to input logic properties	404
multiscale::verification::LogicPropertyVisitor	
Class used to evaluate logic properties	410
multiscale::analysis::MatFactory	
Class for creating a Mat object	429
multiscale::MinEnclosingTriangleFinder	
Class for computing the minimum area enclosing triangle for a given polygon	436
multiscaletest::MinEnclosingTriangleFinderTest	
Class for testing the minimum enclosing triangle algorithm	460
multiscale::verification::ModelChecker	
Abstract class representing a generic model checker	468
multiscale::verification::ModelCheckerFactory	
Interface for different model checker factories	482
multiscaletest::ModelCheckerTest	
Class for testing model checkers	483
multiscale::verification::ModelCheckingException	
Class for representing a model checking exception	490
multiscale::verification::ModelCheckingHelpRequestException	
Class for representing a model checking help request exception	493
multiscale::verification::ModelCheckingManager	
Class for managing the model checking processes	496
multiscale::verification::ModelCheckingOutputWriter	
Class used to output the model checkers progress	513
multiscale::MultiplicationOperation	
Functor representing a multiplication operation	537
multiscale::MultiscaleException	
Parent exception class for the project	538
multiscaletest::MultiscaleTest	
.	542
multiscale::verification::NextKLogicPropertyAttribute	
Class for representing a "next K" logic property attribute	543
multiscale::verification::NextLogicPropertyAttribute	
Class for representing a "next" logic property attribute	544
multiscale::verification::Nil	
A class used to avoid run-time errors when defining a variant type	545

multiscale::verification::NotConstraintAttribute	Class for representing a "not" constraint attribute	546
multiscale::verification::NotLogicPropertyAttribute	Class for representing a "not" logic property attribute	546
multiscale::NumberIterator	Abstract class representing a number iterator	547
multiscale::Numeric	Class for processing numeric (shorts, ints, floats, doubles etc.) expressions	552
multiscale::verification::NumericEvaluator	Class for evaluating numeric expressions	582
multiscale::NumericException	Class for representing algorithm exceptions	585
multiscale::verification::NumericMeasureAttribute	Class for representing a numeric measure attribute	588
multiscale::verification::NumericNumericComparisonAttribute	Class for representing a numeric numeric comparison attribute	589
multiscale::NumericRangeManipulator	Operations for ranges of numeric values	591
multiscale::verification::NumericSpatialAttribute	Class for representing a numeric spatial attribute	592
multiscale::verification::NumericSpatialNumericComparisonAttribute	Class for representing a numeric spatial numeric comparison attribute	593
multiscale::verification::NumericStateVariableAttribute	Class for representing a numeric state variable attribute	595
multiscaletest::NumericStateVariableTraceTest	Class for testing evaluation of numeric state variable-only traces	597
multiscale::verification::NumericVisitor	Class for evaluating numeric measures	600
multiscale::OperatingSystem	Class for executing operating system related functions	608
multiscale::verification::OrConstraintAttribute	Class for representing an "or" constraint attribute	612
multiscale::verification::OrLogicPropertyAttribute	Class for representing an "or" logic property attribute	613
multiscale::verification::Parser	Class used for parsing (P)BLSTL logical queries	614
multiscale::verification::ParserGrammar< Iterator >	The grammar for parsing (P)BLSTL spatial-temporal logical queries	618
multiscale::verification::ParserGrammarExceptionHandler	Class for handling parser grammar exceptions	662
multiscale::verification::ParserGrammarExtraInputException	Class for representing "extra input" exceptions in the parsing process	665
multiscale::verification::ParserGrammarProbabilityException	Class for representing "probability" exceptions in the parsing process	667
multiscale::verification::ParserGrammarUnexpectedTokenException	Class for representing "unexpected token" exceptions in the parsing process	670

multiscale::verification::ParserGrammarUnparseableInputException	
Class for representing "unparseable input" exceptions in the parsing process	673
multiscale::video::PolarCsvToInputFilesConverter	
Csv file to input file converter considering polar coordinates	675
multiscale::video::PolarGnuplotScriptGenerator	
Gnuplot script generator from the provided annular sectors	689
multiscale::verification::PrimaryConstraintAttribute	
Class for representing a primary constraint attribute	696
multiscale::verification::PrimaryLogicPropertyAttribute	
Class for representing a primary logic property attribute	697
multiscale::verification::PrimaryNumericMeasureAttribute	
Class for representing a primary numeric measure attribute	697
multiscale::verification::ProbabilisticBlackBoxModelChecker	
Class used to run probabilistic black-box model checking tasks	698
multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory	
Class for creating ProbabilisticBlackBoxModelChecker instances	703
multiscaletest::ProbabilisticBlackBoxModelCheckerTest	
Class for testing the probabilistic black-box model checker	706
multiscale::verification::ProbabilisticLogicPropertyAttribute	
Class for representing a probabilistic logic property attribute	709
multiscale::verification::ProbabilityErrorHandler	
Structure for defining the error handler for invalid probability errors	713
multiscale::verification::QuaternarySubsetAttribute	
Class for representing a quaternary subset attribute	714
multiscale::verification::QuaternarySubsetMeasureAttribute	
Class for representing a quaternary subset measure attribute	716
multiscale::verification::QuaternarySubsetMeasureTypeParser	
Symbol table and parser for the quaternary subset measure type	717
multiscale::video::RectangularCsvToInputFilesConverter	
Csv file to input file converter considering cartesian coordinates	718
multiscale::video::RectangularEntityCsvToInputFilesConverter	
Csv entity file to input file converter considering cartesian coordinates	731
multiscale::video::RectangularGnuplotScriptGenerator	
Gnuplot script generator from the provided concentrations considering a rectangular geometry	742
multiscale::analysis::RectangularMatFactory	
Class for creating a Mat object considering a rectangular grid	749
multiscale::analysis::Region	
Class for representing a region	756
multiscale::verification::Region	
Class for representing a region	766
multiscale::analysis::RegionDetector	
Class for detecting regions of high intensity in grayscale images	770
multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >	
Structure for specifying the type of the result	799

multiscale::verification::UnexpectedErrorHandler::result< typename, typename, typename >	Structure for specifying the type of the result	799
multiscale::RGBColourGenerator	Generate a RGB colour	800
multiscale::RuntimeException	Class for representing runtime exceptions	804
multiscale::analysis::Silhouette	Class for computing the "Silhouette" clustering index	807
multiscale::analysis::SimulationClusterDetector	Class for detecting clusters in 2D images obtained from simulations	810
multiscaletest::SpatialEntitiesTraceTest	Class for testing evaluation of spatial entities-only traces	821
multiscale::verification::SpatialEntity	Class for representing a pseudo-3D spatial entity	823
multiscale::verification::SpatialEntityEvaluator	Class used to evaluate spatial entities	837
multiscale::analysis::SpatialEntityPseudo3D	Class for representing a pseudo-3D (explicit 2D + implicit height) ob- ject	838
multiscale::verification::SpatialMeasureAttribute	Class for representing a spatial measure attribute	854
multiscale::verification::SpatialMeasureEvaluator	Class for evaluating spatial measures	855
multiscale::verification::SpatialMeasureTypeParser	Symbol table and parser for the spatial measure type	856
multiscale::verification::SpatialNumericComparisonAttribute	Class for representing a spatial numeric comparison attribute	857
multiscale::verification::SpatialTemporalDataReader	Class for reading spatial temporal trace data from input files	858
multiscale::verification::SpatialTemporalException	Class for representing a spatial temporal exception	876
multiscale::verification::SpatialTemporalTrace	Class for representing a spatial temporal trace	879
multiscale::StandardNumberIterator	Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached	890
multiscale::verification::StateVariableAttribute	Class for representing a state variable attribute	895
multiscale::verification::StatisticalModelChecker	Class used to run statistical model checking tasks	896
multiscale::verification::StatisticalModelCheckerFactory	Class for creating StatisticalModelChecker instances	913
multiscaletest::StatisticalModelCheckerTest	Class for testing the statistical model checker	916
multiscale::StringManipulator	Class for manipulating strings	920
multiscale::verification::SubsetAttribute	Class for representing a subset attribute	924

multiscale::verification::SubsetOperationAttribute	Class for representing a subset operation attribute	924
multiscale::verification::SubsetOperationTypeParser	Symbol table and parser for the subset operation type	925
multiscale::verification::SubsetSpecificAttribute	Class for representing a subset specific attribute	926
multiscale::verification::SubsetSpecificTypeParser	Symbol table and parser for a specific subset type	927
multiscale::verification::SubsetSubsetOperationAttribute	Class for representing a subset subset operation attribute	927
multiscale::verification::SubsetVisitor	Class used to evaluate subsets	929
multiscale::SubtractionOperation	Functor representing a subtraction operation	934
multiscale::verification::TernarySubsetAttribute	Class for representing a ternary subset attribute	935
multiscale::verification::TernarySubsetMeasureAttribute	Class for representing a ternary subset measure attribute	937
multiscale::verification::TernarySubsetMeasureTypeParser	Symbol table and parser for the ternary subset measure type	938
multiscale::TestException	Class for representing testing exceptions	939
multiscale::verification::TimePoint	Class for representing a timepoint	942
multiscale::verification::TimePointEvaluator	Class used to evaluate timepoints	956
multiscaletest::TraceEvaluationTest	Class for testing evaluation of traces	958
multiscale::verification::UnaryNumericFilterAttribute	Class for representing a unary numeric filter attribute	962
multiscale::verification::UnaryNumericMeasureAttribute	Class for representing a unary numeric measure attribute	964
multiscale::verification::UnaryNumericMeasureTypeParser	Symbol table and parser for the unary numeric measure type	965
multiscale::verification::UnaryNumericNumericAttribute	Class for representing a unary numeric numeric measure attribute	965
multiscale::verification::UnarySpatialConstraintAttribute	Class for representing a "unary" spatial constraint attribute	967
multiscale::verification::UnarySubsetAttribute	Class for representing a unary subset attribute	968
multiscale::verification::UnarySubsetMeasureAttribute	Class for representing a unary subset measure attribute	970
multiscale::verification::UnarySubsetMeasureTypeParser	Symbol table and parser for the unary subset measure type	971
multiscale::verification::UnaryTypeConstraintAttribute	Class for representing a "unary" type constraint attribute	971
multiscale::UnexpectedBehaviourException	Class for representing unexpected behaviour exceptions	973
multiscale::verification::UnexpectedErrorHandler	Structure for defining the error handler for unexpected token errors .	975

multiscale::UnimplementedMethodException	Class for representing unimplemented method exceptions	977
multiscale::verification::UntilLogicPropertyAttribute	Class for representing an "until" logic property attribute	980
multiscale::XmlValidator::XmlValidationErrorHandler	Class used for handling errors during the xml file validation process	982
multiscale::XmlValidator	Class used to validate xml files	987

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	995
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/core/-	
MultiscaleTest.hpp	996
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
AlgorithmException.hpp	996
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
ExceptionHandler.hpp	997
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
FileOpenException.hpp	997
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
IndexOutOfBoundsException.hpp	998
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
InvalidInputException.hpp	999
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
IOException.hpp	1000
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
MultiscaleException.hpp	1001
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
NumericException.hpp	1004
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
RuntimeException.hpp	1005
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
TestException.hpp	1006
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
UnexpectedBehaviourException.hpp	1006
/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-	
UnimplementedMethodException.hpp	1007

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ CircularityMeasure.hpp	1008
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Cluster.hpp	1009
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ ClusterDetector.hpp	1011
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ DataPoint.hpp	1012
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ DBSCAN.hpp	1012
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Detector.hpp	1013
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Entity.hpp	1014
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ MatFactory.hpp	1016
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Region.hpp	1016
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ RegionDetector.hpp	1018
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Shape2D.hpp	1019
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ Silhouette.hpp	1019
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ SpatialEntityPseudo3D.hpp	1020
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ SpatialEntityPseudo3DType.hpp	1020
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ SimulationClusterDetector.hpp	1010
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ CircularMatFactory.hpp	1014
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/ RectangularMatFactory.hpp	1015
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/- CircularMatFactorySample.cpp	1021
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/- LexicographicIteratorSample.cpp	1021
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/- RectangularMatFactorySample.cpp	1022
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- CircularDetectRegions.cpp	1023
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- CircularDetectRegions.in.cpp	1027
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- CircularityMeasure.cpp	1031
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- Cluster.cpp	1032
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- ClusterDetector.cpp	1033

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
DBSCAN.cpp	1034
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
Detector.cpp	1034
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
Entity.cpp	1034
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
MatFactory.cpp	1036
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
RectangularDetectRegions.cpp	1037
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
RectangularDetectRegions.in.cpp	1040
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
Region.cpp	1044
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
RegionDetector.cpp	1045
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
Silhouette.cpp	1045
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
SimulationDetectClusters.cpp	1046
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
SimulationDetectClusters.in.cpp	1049
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-	
SpatialEntityPseudo3D.cpp	1052
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/-	
SimulationClusterDetector.cpp	1032
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-	
CircularMatFactory.cpp	1035
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-	
RectangularMatFactory.cpp	1035
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/-	
DBSCANTest.cpp	1053
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
ConsolePrinter.hpp	1055
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
Filesystem.hpp	1056
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
Geometry2D.hpp	1057
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
MinEnclosingTriangleFinder.hpp	1059
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
NumberIterator.hpp	1059
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
Numeric.hpp	1060
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
NumericRangeManipulator.hpp	1061
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
OperatingSystem.hpp	1061
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-	
RGBColourGenerator.hpp	1062

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- StringManipulator.hpp	1064
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- XmlValidator.hpp	1065
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- LexicographicNumberIterator.hpp	1057
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- NumberIteratorType.hpp	1058
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/- StandardNumberIterator.hpp	1058
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/- BetaDistribution.hpp	1062
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/- BinomialDistribution.hpp	1063
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/- Distribution.hpp	1064
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/- ConsolePrinterSample.cpp	1065
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/- ExecuteProgramSample.cpp	1067
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/ Line- CircleIntersectionSample.cpp	1068
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/ Min- EnclosingTriangleFinderSample.cpp	1068
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/ RGB- ColourGeneratorSample.cpp	1073
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Console- Printer.cpp	1074
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Filesystem.- cpp	1074
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Geometry2- D.cpp	1075
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Min- EnclosingTriangleFinder.cpp	1076
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Number- Iterator.cpp	1077
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Numeric.- cpp	1077
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Operating- System.cpp	1078
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ RGB- ColourGenerator.cpp	1078
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ String- Manipulator.cpp	1080
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/ Xml- Validator.cpp	1081
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/- LexicographicNumberIterator.cpp	1076
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/- StandardNumberIterator.cpp	1076

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-	
BetaDistribution.cpp	1079
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-	
BinomialDistribution.cpp	1079
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-	
Distribution.cpp	1080
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/Geometry2-	
DTest.cpp	1081
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/Min-	
EnclosingTriangleFinderTest.cpp	1083
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/Numeric-	
Test.cpp	1084
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/test/Statistics-	
Test.cpp	1088
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
AndConstraintAttribute.hpp	1090
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
AndLogicPropertyAttribute.hpp	1090
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
BinaryNumericFilterAttribute.hpp	1091
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
BinaryNumericMeasureAttribute.hpp	1092
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
BinaryNumericNumericAttribute.hpp	1093
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
BinarySubsetAttribute.hpp	1093
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
BinarySubsetMeasureAttribute.hpp	1094
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
ComparatorAttribute.hpp	1095
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
ConstraintAttribute.hpp	1096
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
DifferenceAttribute.hpp	1097
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
EquivalenceConstraintAttribute.hpp	1098
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-	
temporal/include/multiscale/verification/spatial-temporal/attribute/-	
EquivalenceLogicPropertyAttribute.hpp	1098

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- FilterNumericMeasureAttribute.hpp	1099
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- FilterSubsetAttribute.hpp	1100
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- FutureLogicPropertyAttribute.hpp	1101
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- GlobalLogicPropertyAttribute.hpp	1101
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- ImplicationConstraintAttribute.hpp	1102
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- ImplicationLogicPropertyAttribute.hpp	1103
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- LogicPropertyAttribute.hpp	1104
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NextKLogicPropertyAttribute.hpp	1105
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NextLogicPropertyAttribute.hpp	1105
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- Nil.hpp	1106
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NotConstraintAttribute.hpp	1107
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NotLogicPropertyAttribute.hpp	1108
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericMeasureAttribute.hpp	1108
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericNumericComparisonAttribute.hpp	1109
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericSpatialAttribute.hpp	1110
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericSpatialNumericComparisonAttribute.hpp	1111

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericStateVariableAttribute.hpp	1112
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- OrConstraintAttribute.hpp	1112
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- OrLogicPropertyAttribute.hpp	1113
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryConstraintAttribute.hpp	1114
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryLogicPropertyAttribute.hpp	1115
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- PrimaryNumericMeasureAttribute.hpp	1116
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- ProbabilisticLogicPropertyAttribute.hpp	1117
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- QuaternarySubsetAttribute.hpp	1118
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- QuaternarySubsetMeasureAttribute.hpp	1118
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SpatialMeasureAttribute.hpp	1119
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SpatialNumericComparisonAttribute.hpp	1120
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- StateVariableAttribute.hpp	1121
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SubsetAttribute.hpp	1122
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SubsetOperationAttribute.hpp	1123
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SubsetSpecificAttribute.hpp	1123
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SubsetSubsetOperationAttribute.hpp	1125

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SynthesizedAttribute.hpp	1125
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- TernarySubsetAttribute.hpp	1126
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- TernarySubsetMeasureAttribute.hpp	1127
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnaryNumericFilterAttribute.hpp	1128
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnaryNumericMeasureAttribute.hpp	1129
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnaryNumericNumericAttribute.hpp	1130
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnarySpatialConstraintAttribute.hpp	1130
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnarySubsetAttribute.hpp	1131
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnarySubsetMeasureAttribute.hpp	1132
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UnaryTypeConstraintAttribute.hpp	1133
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- UntilLogicPropertyAttribute.hpp	1133
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ApproximateBayesianModelChecker.hpp	1134
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ApproximateBayesianModelCheckerFactory.hpp	1135
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ApproximateProbabilisticModelChecker.hpp	1135
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ApproximateProbabilisticModelCheckerFactory.hpp	1136
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- BayesianModelChecker.hpp	1137

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- BayesianModelCheckerFactory.hpp	1137
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ModelChecker.hpp	1138
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ModelCheckerFactory.hpp	1138
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ModelCheckingManager.hpp	1139
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ModelCheckingOutputWriter.hpp	1140
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ProbabilisticBlackBoxModelChecker.hpp	1140
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ProbabilisticBlackBoxModelCheckerFactory.hpp	1141
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- StatisticalModelChecker.hpp	1141
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- StatisticalModelCheckerFactory.hpp	1142
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/ Logic-PropertyDataReader.hpp	1143
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/- SpatialTemporalDataReader.hpp	1143
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ModelCheckingException.hpp	1144
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ModelCheckingHelpRequestException.hpp	1145
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ParserGrammarExceptionHandler.hpp	1146
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ParserGrammarExtraInputException.hpp	1146
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ParserGrammarProbabilityException.hpp	1147

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ParserGrammarUnexpectedTokenException.hpp	1148
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- ParserGrammarUnparseableInputException.hpp	1148
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/- SpatialTemporalException.hpp	1149
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/execution/- CommandLineModelChecking.hpp	1150
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/- ProbabilityErrorHandler.hpp	1150
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/- UnexpectedTokenErrorHandler.hpp	1151
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- AbstractSyntaxTree.hpp	1152
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- Cluster.hpp	1010
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- Region.hpp	1017
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- SpatialEntity.hpp	1152
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- SpatialTemporalTrace.hpp	1153
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/- TimePoint.hpp	1154
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/- Parser.hpp	1155
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/- ParserGrammar.hpp	1156
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/- SymbolTables.hpp	1156
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- ComparatorEvaluator.hpp	1158

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- ConstraintEvaluator.hpp	1158
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- ConstraintVisitor.hpp	1159
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- FilterNumericVisitor.hpp	1160
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- LogicPropertyVisitor.hpp	1161
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- NumericEvaluator.hpp	1161
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- NumericVisitor.hpp	1162
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- SpatialEntityEvaluator.hpp	1163
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- SpatialMeasureEvaluator.hpp	1163
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- SubsetVisitor.hpp	1164
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- TimePointEvaluator.hpp	1165
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ LogicPropertyDataReaderSample.cpp	1166
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ ParserEvaluationSample.cpp	1167
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ ParserSample.cpp	1168
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ SpatialTemporalDataReaderSample.cpp	1169
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/ Mule.cpp	1185
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ BinaryNumericMeasureAttribute.cpp	1170
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ BinarySubsetMeasureAttribute.cpp	1171
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ ComparatorAttribute.cpp	1171
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ ProbabilisticLogicPropertyAttribute.cpp	1172
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ QuaternarySubsetMeasureAttribute.cpp	1172

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SpatialMeasureAttribute.cpp	1173
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ SubsetSpecificAttribute.cpp	1173
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ TernarySubsetMeasureAttribute.cpp	1174
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ UnaryNumericMeasureAttribute.cpp	1174
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ UnarySubsetMeasureAttribute.cpp	1175
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateBayesianModelChecker.cpp	1175
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateBayesianModelCheckerFactory.cpp	1176
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateProbabilisticModelChecker.cpp	1176
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ApproximateProbabilisticModelChecker-Factory.cpp	1176
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ BayesianModelChecker.cpp	1177
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ BayesianModelCheckerFactory.cpp	1177
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelChecker.cpp	1178
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelCheckingManager.cpp	1178
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelCheckingOutputWriter.cpp	1178
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ModelCheckingOutputWriter.in.cpp	1179
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ProbabilisticBlackBoxModelChecker.cpp	1179
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ ProbabilisticBlackBoxModelCheckerFactory.cpp	1180
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelChecker.cpp	1180
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ StatisticalModelCheckerFactory.cpp	1181
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ LogicPropertyDataReader.cpp	1181
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReader.cpp	1181
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/ SpatialTemporalDataReader.in.cpp	1182
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ ParserGrammarExceptionHandler.cpp	1183

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/execution/ CommandLineModelChecking.cpp	1183
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ AbstractSyntaxTree.cpp	1184
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialEntity.cpp	1184
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ SpatialTemporalTrace.cpp	1185
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/ TimePoint.cpp	1185
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/ Parser.cpp	1186
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ BinaryNumericFilterTest.hpp	1187
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ BinaryNumericMeasureTest.hpp	1189
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ BinaryNumericNumericTest.hpp	1190
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ BinarySubsetMeasureTest.hpp	1192
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ BinarySubsetTest.hpp	1195
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ ComparatorTest.hpp	1197
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ CompoundConstraintTest.hpp	1198
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ CompoundLogicPropertyTest.hpp	1201
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ ConstraintParenthesesTest.hpp	1204
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ ConstraintTest.hpp	1205
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ DifferenceTest.hpp	1206
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ FilterNumericMeasureTest.hpp	1208
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ FilterSubsetTest.hpp	1209
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ FutureLogicPropertyTest.hpp	1211
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ GlobalLogicPropertyTest.hpp	1213
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ LogicPropertyParenthesesTest.hpp	1216
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ LogicPropertyTest.hpp	1217
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ MultipleLogicPropertiesTest.hpp	1218
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ NextKLogicPropertyTest.hpp	1219

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextLogicPropertyTest.hpp	1221
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotConstraintTest.hpp	1221
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotLogicPropertyTest.hpp	1222
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericMeasureTest.hpp	1223
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericNumericComparisonTest.hpp . . .	1224
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialMeasureTest.hpp	1226
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialNumericComparisonTest.hpp .	1226
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericStateVariableTest.hpp	1228
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ProbabilisticLogicPropertyTest.hpp	1230
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetMeasureTest.hpp	1233
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetTest.hpp	1234
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SpatialMeasureTest.hpp	1237
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetOperationTest.hpp	1240
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSpecificTest.hpp	1241
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSubsetOperationTest.hpp	1241
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetTest.hpp	1243
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetMeasureTest.hpp	1244
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetTest.hpp	1245
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericFilterTest.hpp	1248
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericMeasureTest.hpp	1249
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericNumericTest.hpp	1251
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySpatialConstraintTest.hpp	1252
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetMeasureTest.hpp	1254
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetTest.hpp	1255
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryTypeConstraintTest.hpp	1256

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ UntilLogicPropertyTest.hpp	1257
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ApproximateBayesianModelCheckerTest.hpp	1261
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ApproximateProbabilisticModelCheckerTest.hpp	1262
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ BayesianModelCheckerTest.hpp	1263
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ModelCheckerTest.hpp	1264
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ModelCheckingTest.cpp	1265
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ ProbabilisticBlackBoxModelCheckerTest.hpp	1265
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ StatisticalModelCheckerTest.hpp	1266
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ CompleteTraceTest.hpp	1267
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ EmptyTraceTest.hpp	1287
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ NumericStateVariableTraceTest.hpp	1306
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ ParserEvaluationTest.cpp	1326
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ ParserEvaluationTest.hpp	1327
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ SpatialEntitiesTraceTest.hpp	1327
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/ TraceEvaluationTest.hpp	1347
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ InputStringParser.hpp	1348
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ ParserTest.cpp	1348
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ ParserTest.hpp	1349
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/- AnnularSector.hpp	1351
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/- CartesianToPolarConverter.hpp	1351
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/- PolarCsvToInputFilesConverter.hpp	1352
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/- PolarGnuplotScriptGenerator.hpp	1352
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/- AnnularSector.cpp	1353
/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/- CartesianToPolarConverter.cpp	1354

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-	
MapCartesianToPolarScript.cpp	1354
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-	
PolarCsvToInputFilesConverter.cpp	1356
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-	
PolarGnuplotScriptGenerator.cpp	1357
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-	
PolarGnuplotScriptGenerator.in.cpp	1357
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-	
PolarMapCsvToInputFiles.cpp	1357
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectan-	
CartesianToConcentrationsConverter.hpp	1360
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectan-	
RectangularCsvToInputFilesConverter.hpp	1360
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectan-	
RectangularEntityCsvToInputFilesConverter.hpp	1361
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectan-	
RectangularGnuplotScriptGenerator.hpp	1362
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
CartesianToConcentrationsConverter.cpp	1362
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
MapCartesianToScript.cpp	1363
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularCsvToInputFilesConverter.cpp	1364
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularEntityCsvToInputFilesConverter.cpp	1365
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularGnuplotScriptGenerator.cpp	1365
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularGnuplotScriptGenerator.in.cpp	1366
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularMapCsvToInputFiles.cpp	1366
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-	
RectangularMapEntityCsvToInputFiles.cpp	1369

Chapter 6

Namespace Documentation

6.1 multiscale Namespace Reference

Namespaces

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

Classes

- class [AlgorithmException](#)
Class for representing algorithm exceptions.
- class [ExceptionHandler](#)
Exception handler class.
- class [FileOpenException](#)
Class for representing exceptions when opening a file.
- class [IndexOutOfBoundsException](#)
Class for representing an index out of bounds exception.
- class [InvalidInputException](#)
Class for representing invalid input exceptions.
- class [IOException](#)
Class for representing input and output exceptions.
- class [MultiscaleException](#)
Parent exception class for the project.
- class [NumericException](#)
Class for representing algorithm exceptions.
- class [RuntimeException](#)
Class for representing runtime exceptions.
- class [TestException](#)

- class [UnexpectedBehaviourException](#)

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

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

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

Class used to print (coloured) messages to the console.
- class [Geometry2D](#)

Class containing methods for interacting with the filesystem.
- class [LexicographicNumberIterator](#)

Two-dimensional geometric operations.
- class [StandardNumberIterator](#)

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

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

Class for computing the minimum area enclosing triangle for a given polygon.
- class [AdditionOperation](#)

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

Functor representing an addition operation.
- class [MultiplicationOperation](#)

Functor representing a division operation.
- class [SubtractionOperation](#)

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

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

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

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

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

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

Class for analysing Beta distributed data.
- class [Distribution](#)

Class for analysing Binomial distributed data.
- class [StringManipulator](#)

Class for manipulating strings.
- class [XmlValidator](#)

Class used to validate xml files.

Enumerations

- enum `UnixColourCode` { `BLACK` = 0, `RED` = 1, `GREEN` = 2, `YELLOW` = 3, `BLUE` = 4, `MAGENTA` = 5, `CYAN` = 6, `WHITE` = 7 }
- enum `WindowsColourCode` { `BLACK` = 0, `DARK_BLUE` = 1, `DARK_GREEN` = 2, `DARK_CYAN` = 3, `DARK_RED` = 4, `DARK_MAGENTA` = 5, `DARK_YELLOW` = 6, `DARK_WHITE` = 7, `GRAY` = 8, `BLUE` = 4, `GREEN` = 2, `CYAN` = 6, `RED` = 1, `MAGENTA` = 5, `YELLOW` = 3, `WHITE` = 7 }
- enum `ColourCode` { `BLACK` = 0, `RED` = 1, `GREEN` = 2, `YELLOW` = 3, `BLUE` = 4, `MAGENTA` = 5, `CYAN` = 6, `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."

6.1.1 Enumeration Type Documentation

6.1.1.1 enum multiscale::ColourCode

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
 Faint 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

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
 - Faint white non-colour

Definition at line 12 of file ConsolePrinter.hpp.

6.1.1.4 enum multiscale::WindowsColourCode

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** White non-colour
 - 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_MSG = "An error occurred: "`

Definition at line 25 of file Multiscale.hpp.

Referenced by `isValidNrOfConcentrationsForPosition()`, `isValidOutputType()`, `multiscale::ExceptionHandler::printErrorMessage()`, and `printWrongParameters()`.

6.1.2.2 `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 70 of file MultiscaleException.hpp.

Referenced by `multiscale::verification::SpatialEntityEvaluator::evaluate()`, `multiscale::verification::SpatialMeasureEvaluator::evaluate()`, `multiscale::verification::NumericEvaluator::evaluate()`, `multiscale::verification::ComparatorEvaluator::evaluate()`, `multiscale::verification::SubsetVisitor::evaluateSubsetOperation()`, `multiscale::verification::subsetspecific::validateSubsetSpecificType()`, and `multiscale::verification::subsetspecific::validateSubsetSpecificTypeIndex()`.

6.1.2.3 `const int multiscale::EXEC_ERR_CODE = 1`

Definition at line 22 of file Multiscale.hpp.

Referenced by `main()`.

6.1.2.4 `const int multiscale::EXEC_SUCCESS_CODE = 0`

Definition at line 21 of file Multiscale.hpp.

Referenced by `main()`.

6.2 multiscale::analysis Namespace Reference

Classes

- class [CircularityMeasure](#)

Class for computing the circularity measure for the given collection of points.

- class [SimulationClusterDetector](#)

Class for detecting clusters in 2D images obtained from simulations.

- 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 which implements an improved version of the DBSCAN algorithm.
- class **Detector**
Abstract class for detecting entities of interest in images.
- class **Entity**
Class for representing an entity in an image (e.g. cell, organism etc.)
- class **CircularMatFactory**
Class for creating a Mat object considering a circular grid.
- class **RectangularMatFactory**
Class for creating a Mat object considering a rectangular grid.
- class **MatFactory**
Class for creating a Mat object.
- 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 **SpatialEntityPseudo3D**
Class for representing a pseudo-3D (explicit 2D + implicit height) object.

TypeDefs

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

Enumerations

- enum **Shape2D** { **Triangle** = 1, **Rectangle** = 2, **Circle** = 3, **Undefined** = 4 }
Enumeration for determining the type of a 2D shape.
- enum **SpatialEntityPseudo3DType** { **Cluster** = 1, **Region** = 2 }
Enumeration for determining the type of a pseudo 3D entity.

6.2.1 Typedef Documentation

6.2.1.1 **typedef std::pair<std::vector<Point>, std::vector<std::vector<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 23 of file RegionDetector.hpp.

6.2.2 Enumeration Type Documentation

6.2.2.1 enum multiscale::analysis::Shape2D

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

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

- namespace [subset specific](#)

Classes

- class [AndConstraintAttribute](#)
Class for representing an "and" constraint attribute.
- class [AndLogicPropertyAttribute](#)
Class for representing an "and" logic property attribute.
- class [BinaryNumericFilterAttribute](#)
Class for representing a binary numeric filter attribute.
- class [BinaryNumericMeasureAttribute](#)
Class for representing a binary numeric measure attribute.
- class [BinaryNumericNumericAttribute](#)
Class for representing a binary numeric numeric measure attribute.

- class [BinarySubsetAttribute](#)
Class for representing a binary subset attribute.
- class [BinarySubsetMeasureAttribute](#)
Class for representing a binary subset measure attribute.
- class [ComparatorAttribute](#)
Class for representing a comparator attribute.
- class [ConstraintAttribute](#)
Class for representing a constraint attribute.
- class [DifferenceAttribute](#)
Class for representing a difference attribute.
- 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 [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 [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 [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 [NumericMeasureAttribute](#)
Class for representing a numeric measure attribute.
- class [NumericNumericComparisonAttribute](#)
Class for representing a numeric numeric comparison attribute.
- class [NumericSpatialAttribute](#)

- class [NumericSpatialNumericComparisonAttribute](#)
 - Class for representing a numeric spatial attribute.*
- class [NumericStateVariableAttribute](#)
 - Class for representing a numeric spatial numeric comparison attribute.*
- class [OrConstraintAttribute](#)
 - Class for representing a numeric state variable attribute.*
- class [OrLogicPropertyAttribute](#)
 - Class for representing an "or" constraint attribute.*
- class [PrimaryConstraintAttribute](#)
 - Class for representing an "or" logic property attribute.*
- class [PrimaryLogicPropertyAttribute](#)
 - Class for representing a primary constraint attribute.*
- class [PrimaryNumericMeasureAttribute](#)
 - Class for representing a primary logic property attribute.*
- class [ProbabilisticLogicPropertyAttribute](#)
 - Class for representing a primary numeric measure attribute.*
- class [QuaternarySubsetAttribute](#)
 - Class for representing a quaternary subset attribute.*
- class [QuaternarySubsetMeasureAttribute](#)
 - Class for representing a quaternary subset measure attribute.*
- class [SpatialMeasureAttribute](#)
 - Class for representing a spatial measure attribute.*
- class [SpatialNumericComparisonAttribute](#)
 - Class for representing a spatial numeric comparison attribute.*
- class [StateVariableAttribute](#)
 - Class for representing a state variable attribute.*
- class [SubsetAttribute](#)
 - Class for representing a subset attribute.*
- class [SubsetOperationAttribute](#)
 - Class for representing a subset operation attribute.*
- class [SubsetSpecificAttribute](#)
 - Class for representing a subset specific attribute.*
- class [SubsetSubsetOperationAttribute](#)
 - Class for representing a subset subset operation attribute.*
- class [TernarySubsetAttribute](#)
 - Class for representing a ternary subset attribute.*
- class [TernarySubsetMeasureAttribute](#)
 - Class for representing a ternary subset measure attribute.*
- class [UnaryNumericFilterAttribute](#)
 - Class for representing a unary numeric filter attribute.*
- class [UnaryNumericMeasureAttribute](#)
 - Class for representing a unary numeric measure attribute.*

- class [UnaryNumericNumericAttribute](#)
Class for representing a unary numeric numeric measure attribute.
- class [UnarySpatialConstraintAttribute](#)
Class for representing a "unary" spatial constraint attribute.
- class [UnarySubsetAttribute](#)
Class for representing a unary subset attribute.
- class [UnarySubsetMeasureAttribute](#)
Class for representing a unary subset measure attribute.
- class [UnaryTypeConstraintAttribute](#)
Class for representing a "unary" type constraint attribute.
- class [UntilLogicPropertyAttribute](#)
Class for representing an "until" 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 [ModelChecker](#)
Abstract class representing a generic model checker.
- class [ModelCheckerFactory](#)
Interface for different model checker factories.
- class [ModelCheckingManager](#)
Class for managing the model checking processes.
- class [ModelCheckingOutputWriter](#)
Class used to output the model checkers progress.
- class [ProbabilisticBlackBoxModelChecker](#)
Class used to run probabilistic black-box model checking tasks.
- class [ProbabilisticBlackBoxModelCheckerFactory](#)
Class for creating `ProbabilisticBlackBoxModelChecker` instances.
- class [StatisticalModelChecker](#)
Class used to run statistical model checking tasks.
- class [StatisticalModelCheckerFactory](#)
Class for creating `StatisticalModelChecker` instances.
- class [LogicPropertyDataReader](#)
Class used to input logic properties.
- class [SpatialTemporalDataReader](#)

- class [ModelCheckingException](#)
Class for reading spatial temporal trace data from input files.
- class [ModelCheckingHelpRequestException](#)
Class for representing a model checking exception.
- class [ParserGrammarExceptionHandler](#)
Class for representing a model checking help request exception.
- 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 "probability" exceptions in the parsing process.
- class [ParserGrammarUnparseableInputException](#)
Class for representing "unexpected token" exceptions in the parsing process.
- class [SpatialTemporalException](#)
Class for representing "unparseable input" exceptions in the parsing process.
- class [CommandLineModelChecking](#)
Class for representing a spatial temporal exception.
- struct [ProbabilityErrorHandler](#)
Structure for running model checkers from the command line.
- struct [UnexpectedTokenErrorHandler](#)
Structure for defining the error handler for invalid probability errors.
- class [AbstractSyntaxTree](#)
Structure for defining the error handler for unexpected token errors.
- class [Cluster](#)
Class used for representing an abstract syntax tree.
- class [Region](#)
Class for representing a cluster.
- class [SpatialEntity](#)
Class for representing a region.
- class [SpatialTemporalTrace](#)
Class for representing a pseudo-3D spatial entity.
- class [TimePoint](#)
Class for representing a spatial temporal trace.
- class [Parser](#)
Class for representing a timepoint.
- class [ParserGrammar](#)
Class used for parsing (P)BLSTL logical queries.
- struct [ComparatorTypeParser](#)
The grammar for parsing (P)BLSTL spatial-temporal logical queries.
- struct [ComparatorNonEqualTypeParser](#)
Symbol table and parser for the comparator type.
- struct [ComparatorEqualTypeParser](#)
Symbol table and parser for the comparator type which does not accept the "=" symbol.

- struct [SpatialMeasureTypeParser](#)
Symbol table and parser for the spatial measure type.
- struct [SubsetSpecificTypeParser](#)
Symbol table and parser for a specific subset type.
- struct [SubsetOperationTypeParser](#)
Symbol table and parser for the subset operation type.
- struct [BinaryNumericMeasureTypeParser](#)
Symbol table and parser for the binary numeric measure type.
- struct [UnaryNumericMeasureTypeParser](#)
Symbol table and parser for the unary numeric measure type.
- struct [QuaternarySubsetMeasureTypeParser](#)
Symbol table and parser for the quaternary subset measure type.
- struct [TernarySubsetMeasureTypeParser](#)
Symbol table and parser for the ternary subset measure type.
- struct [BinarySubsetMeasureTypeParser](#)
Symbol table and parser for the binary subset measure type.
- struct [UnarySubsetMeasureTypeParser](#)
Symbol table and parser for the unary subset measure type.
- class [ComparatorEvaluator](#)
Class for evaluating comparison expressions.
- class [ConstraintEvaluator](#)
Class for evaluating constraint expressions.
- class [ConstraintVisitor](#)
Class used to evaluate constraints.
- class [FilterNumericVisitor](#)
Class for evaluating filter numeric measures.
- class [LogicPropertyVisitor](#)
Class used to evaluate logic properties.
- class [NumericEvaluator](#)
Class for evaluating numeric expressions.
- class [NumericVisitor](#)
Class for evaluating numeric measures.
- class [SpatialEntityEvaluator](#)
Class used to evaluate spatial entities.
- class [SpatialMeasureEvaluator](#)
Class for evaluating spatial measures.
- class [SubsetVisitor](#)
Class used to evaluate subsets.
- class [TimePointEvaluator](#)
Class used to evaluate timepoints.

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 < NumericSpatialAttribute > , boost::recursive_wrapper < PrimaryNumericMeasureAttribute > , boost::recursive_wrapper < UnaryNumericNumericAttribute > , boost::recursive_wrapper < BinaryNumericNumericAttribute > , boost::recursive_wrapper < NumericMeasureAttribute > > > NumericMeasureAttributeType`

Variant for the numeric measure attribute.

- `typedef boost::variant < UnarySubsetAttribute, BinarySubsetAttribute, TernarySubsetAttribute, QuaternarySubsetAttribute, boost::recursive_wrapper < - NumericSpatialAttribute > > NumericSpatialAttributeType`

Variant for a numeric spatial 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 < DifferenceAttribute, NumericSpatialNumericComparisonAttribute, NumericNumericComparisonAttribute, 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 < NumericSpatialAttribute >, 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.

Enumerations

- `enum BinaryNumericMeasureType { Add = 1, Div = 2, Log = 3, Mod = 4, Multiply = 5, Power = 6, Subtract = 7 }`
Enumeration for representing a binary numeric measure type.
- `enum BinarySubsetMeasureType { Avg = 1, Geomean = 2, Harmean = 3, Kurt = 4, Max = 5, Median = 6, Min = 7, Mode = 8, Product = 9, Skew = 10, Stdev = 11, Sum = 12, Var = 13 }`
Enumeration for representing a binary subset measure type.
- `enum ComparatorType { GreaterThan = 1, GreaterThanOrEqual = 2, LessThan = 3, LessThanOrEqual = 4, Equal = 5 }`
Enumeration for representing a comparator type.
- `enum QuaternarySubsetMeasureType { Covar = 1 }`
Enumeration for representing a quaternary subset measure type.
- `enum SpatialMeasureType { Clusteredness = 1, Density = 2, Area = 3, Perimeter = 4, DistanceFromOrigin = 5, Angle = 6, TriangleMeasure = 7, RectangleMeasure = 8, CircleMeasure = 9, CentroidX = 10, CentroidY = 11 }`
Enumeration for representing the types of spatial measures.
- `enum SubsetOperationType { Difference = 1, Intersection = 2, Union = 3 }`
Enumeration for representing the types of subset operations.
- `enum SubsetSpecificType { Clusters = 0, Regions, NrOfSubsetSpecificTypeEntries }`
Enumeration for representing a specific subset type.
- `enum TernarySubsetMeasureType { Percentile = 1, Quartile = 2 }`
Enumeration for representing a ternary subset measure type.
- `enum UnaryNumericMeasureType { Abs = 1, Ceil = 2, Floor = 3, Round = 4, Sign = 5, Sqrt = 6, Trunc = 7 }`
Enumeration for representing a unary numeric measure type.
- `enum UnarySubsetMeasureType { Count = 1, Clusteredness = 1, Density = 2 }`
Enumeration for representing a unary subset measure type.
- `enum ApproximateBayesianModelCheckingResult { TRUE, FALSE, MORE_TRACES_REQUIRED }`
Enumeration for representing the model checking result.
- `enum BayesianModelCheckingResult { TRUE, FALSE, MORE_TRACES_REQUIRED }`
Enumeration for representing the model checking result.

- enum `StatisticalModelCheckingResult` { `TRUE`, `FALSE`, `UNDECIDED`, `MORE_T-RACES_REQUIRED` }
- Enumeration for representing the model checking result.*
- enum `SetOperationType` { `Difference` = 1, `Intersection` = 2, `Union` = 3 }
- Enumeration for representing the set operation type(s)*

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 BinarySubsetMeasureType &binarySubsetMeasureType)`
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 QuaternarySubsetMeasureType &quaternarySubsetMeasureType)`
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 TernarySubsetMeasureType &ternarySubsetMeasureType)`
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 UnarySubsetMeasureType &unarySubsetMeasureType)`
Overload the output stream operator for the enumeration.

Variables

- 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 handleUnexpectedTokenError = 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\"."

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< double, NumericStateVariableAttribute,
boost::recursive_wrapper<NumericSpatialAttribute>,
boost::recursive_wrapper<PrimaryNumericMeasureAttribute>,
boost::recursive_wrapper<UnaryNumericNumericAttribute>,
boost::recursive_wrapper<BinaryNumericNumericAttribute>,
boost::recursive_wrapper<NumericMeasureAttribute> >
multiscale::verification::NumericMeasureAttributeType
```

Variant for the numeric measure attribute.

Definition at line 18 of file NumericMeasureAttribute.hpp.

```
6.3.1.5 typedef boost::variant< UnarySubsetAttribute, BinarySubsetAttribute,
TernarySubsetAttribute, QuaternarySubsetAttribute,
boost::recursive_wrapper<NumericSpatialAttribute> >
multiscale::verification::NumericSpatialAttributeType
```

Variant for a numeric spatial attribute.

Definition at line 19 of file NumericSpatialAttribute.hpp.

```
6.3.1.6 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.

```
6.3.1.7 typedef boost::variant< DifferenceAttribute, NumericSpatialNumeric-
ComparisonAttribute, NumericNumericComparisonAttribute,
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.

```
6.3.1.8 typedef boost::variant< double, NumericStateVariableAttribute,
boost::recursive_wrapper<NumericSpatialAttribute>,
boost::recursive_wrapper<PrimaryNumericMeasureAttribute> >
multiscale::verification::PrimaryNumericMeasureAttributeType
```

Variant for the primary numeric measure attribute.

Definition at line 15 of file PrimaryNumericMeasureAttribute.hpp.

```
6.3.1.9 typedef boost::variant< SubsetSpecificAttribute, FilterSubsetAttribute,
boost::recursive_wrapper<SubsetSubsetOperation-
Attribute>, boost::recursive_wrapper<SubsetAttribute> >
multiscale::verification::SubsetAttributeType
```

Variant for a subset attribute.

Definition at line 16 of file SubsetAttribute.hpp.

6.3.2 Enumeration Type Documentation

```
6.3.2.1 enum multiscale::verification::ApproximateBayesianModelCheckingResult
```

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
```

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

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::BinarySubsetMeasureType

Enumeration for representing a binary subset measure type.

Enumerator:

- Avg** The average (arithmetic mean)
- 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 BinarySubsetMeasureAttribute.hpp.

6.3.2.5 enum multiscale::verification::ComparatorType

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.

6.3.2.6 enum multiscale::verification::QuaternarySubsetMeasureType

Enumeration for representing a quaternary subset measure type.

Enumerator:

- Covar** Covariance

Definition at line 13 of file QuaternarySubsetMeasureAttribute.hpp.

6.3.2.7 enum multiscale::verification::SetOperationType

Enumeration for representing the set operation type(s)

Enumerator:

- Difference** Difference of two subsets
 - Set difference
- Intersection** Intersection of two subsets
 - Set intersection
- Union** Union of two subsets
 - Set union

Definition at line 22 of file TimePoint.hpp.

6.3.2.8 enum multiscale::verification::SpatialMeasureType

Enumeration for representing the types of spatial measures.

Enumerator:

- Clusteredness** The clusteredness of the spatial entity
 - The overall clusteredness of the entities
- Density** The density of the spatial entity
 - The overall density of the entities

Area The area of the spatial entity

Perimeter The perimeter of the spatial entity

DistanceFromOrigin The distance of the spatial entity from the origin (centre of the discretised space)

Angle The angle determined by the spatial entity wrt the origin (centre of the discretised space) in degrees

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 centroid of the spatial entity

CentroidY The y coordinate of the centroid of the spatial entity

Definition at line 13 of file SpatialMeasureAttribute.hpp.

6.3.2.9 enum multiscale::verification::StatisticalModelCheckingResult

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.

6.3.2.10 enum multiscale::verification::SubsetOperationType

Enumeration for representing the types of subset operations.

Enumerator:

Difference Difference of two subsets

Set difference

Intersection Intersection of two subsets

Set intersection

Union Union of two subsets

Set union

Definition at line 13 of file SubsetOperationAttribute.hpp.

6.3.2.11 enum multiscale::verification::SubsetSpecificType

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 13 of file SubsetSpecificAttribute.hpp.

6.3.2.12 enum multiscale::verification::TernarySubsetMeasureType

Enumeration for representing a ternary subset measure type.

Enumerator:

Percentile The percentile

Quartile The quartile

Definition at line 13 of file TernarySubsetMeasureAttribute.hpp.

6.3.2.13 enum multiscale::verification::UnaryNumericMeasureType

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.

6.3.2.14 enum multiscale::verification::UnarySubsetMeasureType

Enumeration for representing a unary subset measure type.

Enumerator:

Count Number of spatial entities

Clusteredness The clusteredness of the spatial entity

The overall clusteredness of the entities

Density The density of the spatial entity

The overall density of the entities

Definition at line 13 of file UnarySubsetMeasureAttribute.hpp.

6.3.3 Function Documentation

6.3.3.1 `std::ostream & multiscale::verification::operator<< (std::ostream & out, const QuaternarySubsetMeasureType & quaternarySubsetMeasureType)`

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>quaternary- Subset- Measure- Type</i>	The quaternary subset measure type to be printed out

Definition at line 7 of file QuaternarySubsetMeasureAttribute.cpp.

References Covar.

6.3.3.2 `std::ostream & multiscale::verification::operator<< (std::ostream & out, const TernarySubsetMeasureType & ternarySubsetMeasureType)`

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>ternary- Subset- Measure- Type</i>	The ternary subset measure type to be printed out

Definition at line 7 of file TernarySubsetMeasureAttribute.cpp.

References Percentile, and Quartile.

6.3.3.3 `std::ostream & multiscale::verification::operator<< (std::ostream & out, const UnarySubsetMeasureType & unarySubsetMeasureType)`

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>unary-Subset-Measure-Type</i>	The unary subset measure type to be printed out

Definition at line 7 of file UnarySubsetMeasureAttribute.cpp.

References Clusteredness, Count, and Density.

6.3.3.4 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-Operation-Type</i>	The subset operation type to be printed out

6.3.3.5 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>comparator-Type</i>	The comparator type to be printed out

Definition at line 7 of file ComparatorAttribute.cpp.

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

6.3.3.6 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>unary-Numeric-Measure-Type</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.

6.3.3.7 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>binary-Numeric-Measure-Type</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.

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

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>spatial-Measure-Type</i>	The spatial measure type to be printed out

Definition at line 7 of file SpatialMeasureAttribute.cpp.

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

6.3.3.9 std::ostream & multiscale::verification::operator<< (std::ostream & *out*, const BinarySubsetMeasureType & *binarySubsetMeasureType*)

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>binary-Subset-Measure-Type</i>	The binary subset measure type to be printed out

Definition at line 7 of file BinarySubsetMeasureAttribute.cpp.

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

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

Overload the output stream operator for the enumeration.

Parameters

<i>out</i>	Output stream
<i>subset-SpecificType</i>	The specific subset type to be printed out

Definition at line 39 of file SubsetSpecificAttribute.cpp.

References Clusters, and Regions.

6.3.4 Variable Documentation

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

Definition at line 32 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyErrorHandlingSupport().

6.3.4.2 phoenix::function<UnexpectedTokenErrorHandler> const multiscale::verification::handleUnexpectedTokenError = UnexpectedTokenErrorHandler()

Definition at line 31 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator

```
>::initialiseFilterNumericMeasureErrorHandler(),      multiscale::verification-
::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureError-
HandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator
>::initialiseNumericSpatialMeasureErrorHandler(),   multiscale::verification-
::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintError-
HandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator
>::initialisePrimaryLogicPropertyErrorHandler(),    multiscale::verification::
ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicProperty-
ErrorHandler(), multiscale::verification::ParserGrammar< std::string::const_-
iterator >::initialiseStateVariableErrorHandler(), and multiscale::verification::
ParserGrammar< std::string::const_iterator >::initialiseSubsetErrorHandler().
```

6.3.4.3 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 22 of file SubsetSpecificAttribute.hpp.

Referenced by multiscale::verification::ConstraintEvaluator::evalSpatialMeasure-
Constraint(), multiscale::verification::ConstraintEvaluator::evalTypeConstraint(),
multiscale::verification::TimePoint::getConsideredSpatialEntities(), multiscale::verification-
::TimePoint::numberOfSpatialEntities(), multiscale::verification::TimePoint::TimePoint(),
multiscale::verification::TimePoint::updateSpatialEntities(), and multiscale::verification-
::subsetsspecific::validateSubsetSpecificTypeIndex().

6.3.4.4 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 16 of file LogicPropertyVisitor.hpp.

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

6.4 multiscale::verification::subsetsspecific Namespace Reference

Functions

- void [validateSubsetSpecificType](#) (const [SubsetSpecificType](#) &subsetSpecific-
Type)

Check if the given subset specific type is valid.
- void [validateSubsetSpecificTypeIndex](#) (const std::size_t &subsetSpecificType-
Index)

Check if the given subset specific type index is valid.
- size_t [computeSubsetSpecificTypeIndex](#) (const [SubsetSpecificType](#) &subset-
SpecificType)

Compute the index of the subset specific type.

- **SubsetSpecificType computeSubsetSpecificType (const std::size_t &subsetSpecificTypeIndex)**

Compute the subset specific type from the given index.

6.4.1 Function Documentation

- 6.4.1.1 **SubsetSpecificType multiscale::verification::subsetspecific::computeSubsetSpecificType (const std::size_t & subsetSpecificTypeIndex)**

Compute the subset specific type from the given index.

Parameters

<i>subset-Specific-TypeIndex</i>	The given subset specific type index
----------------------------------	--------------------------------------

Definition at line 31 of file SubsetSpecificAttribute.cpp.

References validateSubsetSpecificTypeIndex().

Referenced by multiscale::verification::ConstraintEvaluator::evalSpatialMeasureConstraint(), multiscale::verification::ConstraintEvaluator::evalTypeConstraint(), and multiscale::verification::TimePoint::updateSpatialEntities().

- 6.4.1.2 **size_t multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex (const SubsetSpecificType & subsetSpecificType)**

Compute the index of the subset specific type.

Parameters

<i>subset-SpecificType</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().

6.4.1.3 void multiscale::verification::subsetsspecific::validateSubsetSpecificType (const SubsetSpecificType & subsetSpecificType)

Check if the given subset specific type is valid.

Parameters

<i>subset-SpecificType</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().

6.4.1.4 void multiscale::verification::subsetsspecific::validateSubsetSpecificTypeIndex (const std::size_t & subsetSpecificTypeIndex)

Check if the given subset specific type index is valid.

Parameters

<i>subset-Specific-TypeIndex</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().

6.5 multiscale::video Namespace Reference

Classes

- class [AnnularSector](#)

An annular sector is the basic element in the considered circular geometry.
- class [CartesianToPolarConverter](#)

Converter from the rectangular geometry grid cells to annular sectors.
- class [PolarCsvToInputFilesConverter](#)

- class [PolarGnuplotScriptGenerator](#)

Csv file to input file converter considering polar coordinates.
- class [CartesianToConcentrationsConverter](#)

Gnuplot script generator from the provided annular sectors.
- class [RectangularCsvToInputFilesConverter](#)

Scale the values of the rectangular geometry grid cells.
- class [RectangularEntityCsvToInputFilesConverter](#)

Csv file to input file converter considering cartesian coordinates.
- class [RectangularGnuplotScriptGenerator](#)

Csv entity file to input file converter considering cartesian coordinates.
- class [RectangularGnuplotScriptGenerator](#)

Gnuplot script generator from the provided concentrations considering a rectangular geometry.

6.6 multiscaletest Namespace Reference

Namespaces

- namespace [verification](#)

Classes

- class [MultiscaleTest](#)
- class [MinEnclosingTriangleFinderTest](#)

Class for testing the minimum enclosing triangle algorithm.
- 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 [ModelCheckerTest](#)

Class for testing model checkers.
- class [ProbabilisticBlackBoxModelCheckerTest](#)

Class for testing the probabilistic black-box model checker.
- class [StatisticalModelCheckerTest](#)

Class for testing the statistical 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 [NumericStateVariableTraceTest](#)

Class for testing evaluation of numeric state variable-only traces.

- class [SpatialEntitiesTraceTest](#)
Class for testing evaluation of spatial entities-only traces.
- class [TraceEvaluationTest](#)
Class for testing evaluation of traces.

6.7 multiscaletest::verification Namespace Reference

Functions

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

6.7.1 Function Documentation

6.7.1.1 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)`

Evaluate the abstract syntax tree considering the given trace.

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 13 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*)

Evaluate the abstract syntax tree considering the given trace.

Parameters

<code>spatial-Temporal-Trace</code>	The given spatial temporal trace
-------------------------------------	----------------------------------

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 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>probabilistic- Logic- Property- Attribute</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 46 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 17 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 18 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>
```

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 17 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

<code><i>operand1</i></code>	The first operand
<code><i>operand2</i></code>	The second operand

Definition at line 27 of file Numeric.hpp.

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

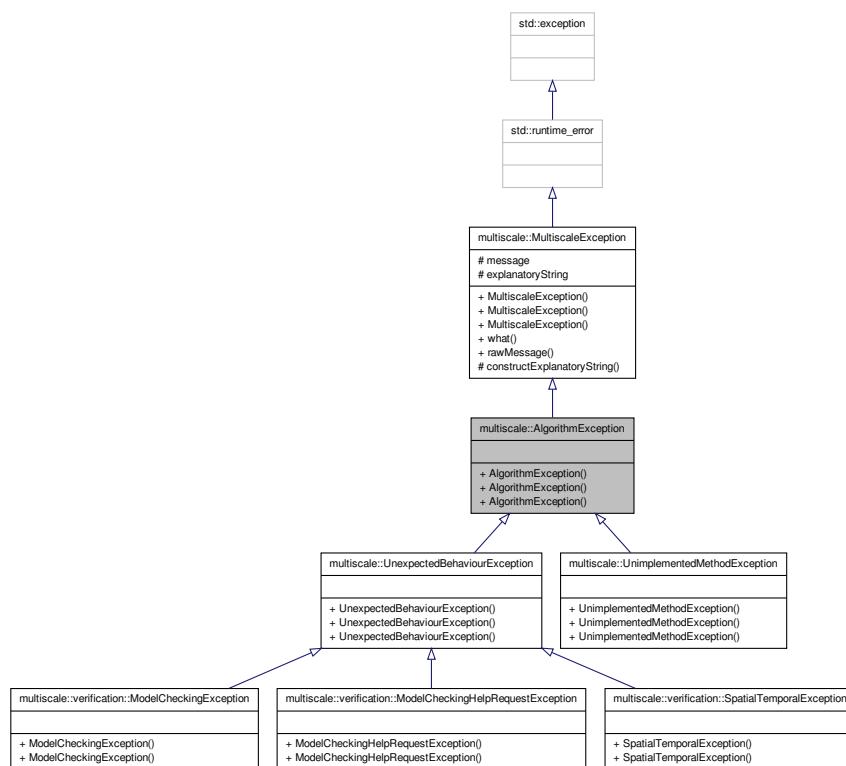
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util-/Numeric.hpp](#)

7.3 multiscale::AlgorithmException Class Reference

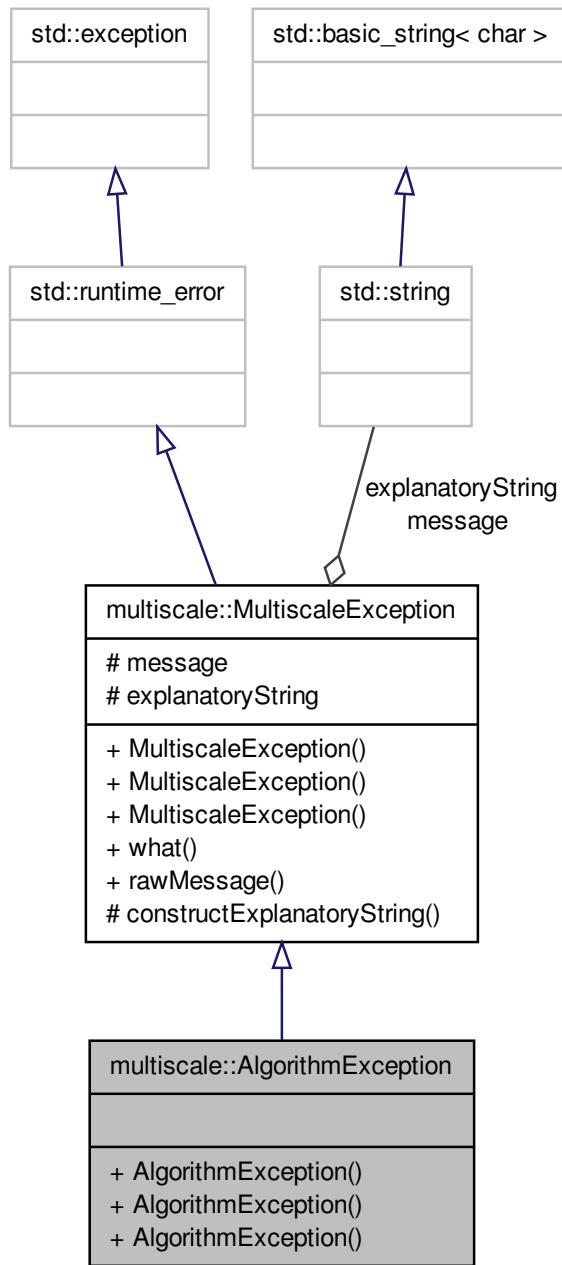
Class for representing algorithm exceptions.

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

Inheritance diagram for multiscale::AlgorithmException:



Collaboration diagram for multiscale::AlgorithmException:



Public Member Functions

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

7.3.1 Detailed Description

Class for representing algorithm exceptions.

Definition at line 14 of file AlgorithmException.hpp.

7.3.2 Constructor & Destructor Documentation

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

Definition at line 18 of file AlgorithmException.hpp.

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

Definition at line 20 of file AlgorithmException.hpp.

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

Definition at line 24 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>
```

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>
```

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>
```

Public Member Functions

- [AnnularSector \(\)](#)
Initialise the members of the class.
- [~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.
- string [toString \(\)](#)
Get the 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 16 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>starting-Radius</i>	Starting radius
<i>ending-Radius</i>	Ending radius
<i>starting-Angle</i>	Starting angle
<i>endingAngle</i>	Ending angle
<i>concentra-tion</i>	Concentration

Definition at line 21 of file AnnularSector.cpp.

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

7.6.3.7 string AnnularSector::toString()

Get the 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 24 of file AnnularSector.hpp.

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

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

Definition at line 23 of file AnnularSector.hpp.

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

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

Definition at line 21 of file AnnularSector.hpp.

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

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

Definition at line 22 of file AnnularSector.hpp.

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

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

Definition at line 20 of file AnnularSector.hpp.

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

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

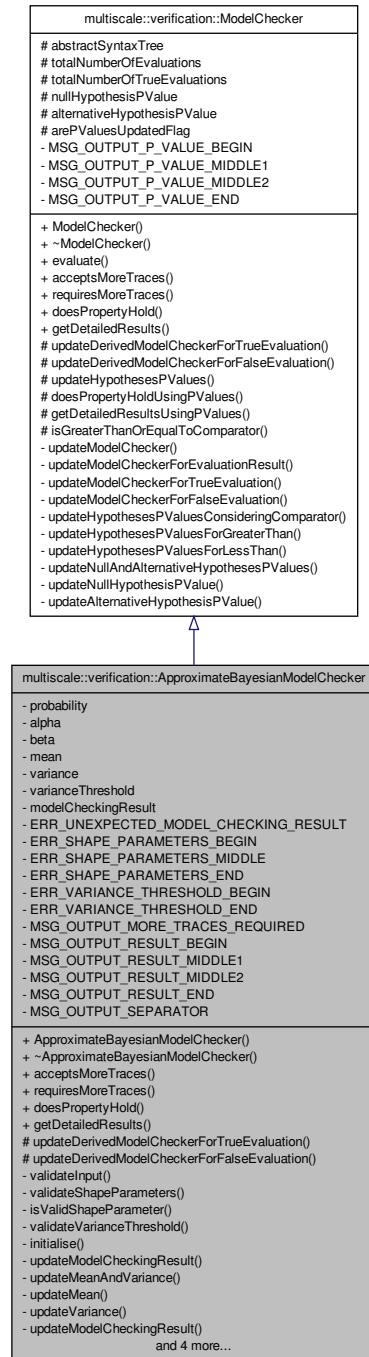
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/[AnnularSector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/[AnnularSector.cpp](#)

7.7 multiscale::verification::ApproximateBayesianModelChecker Class Reference

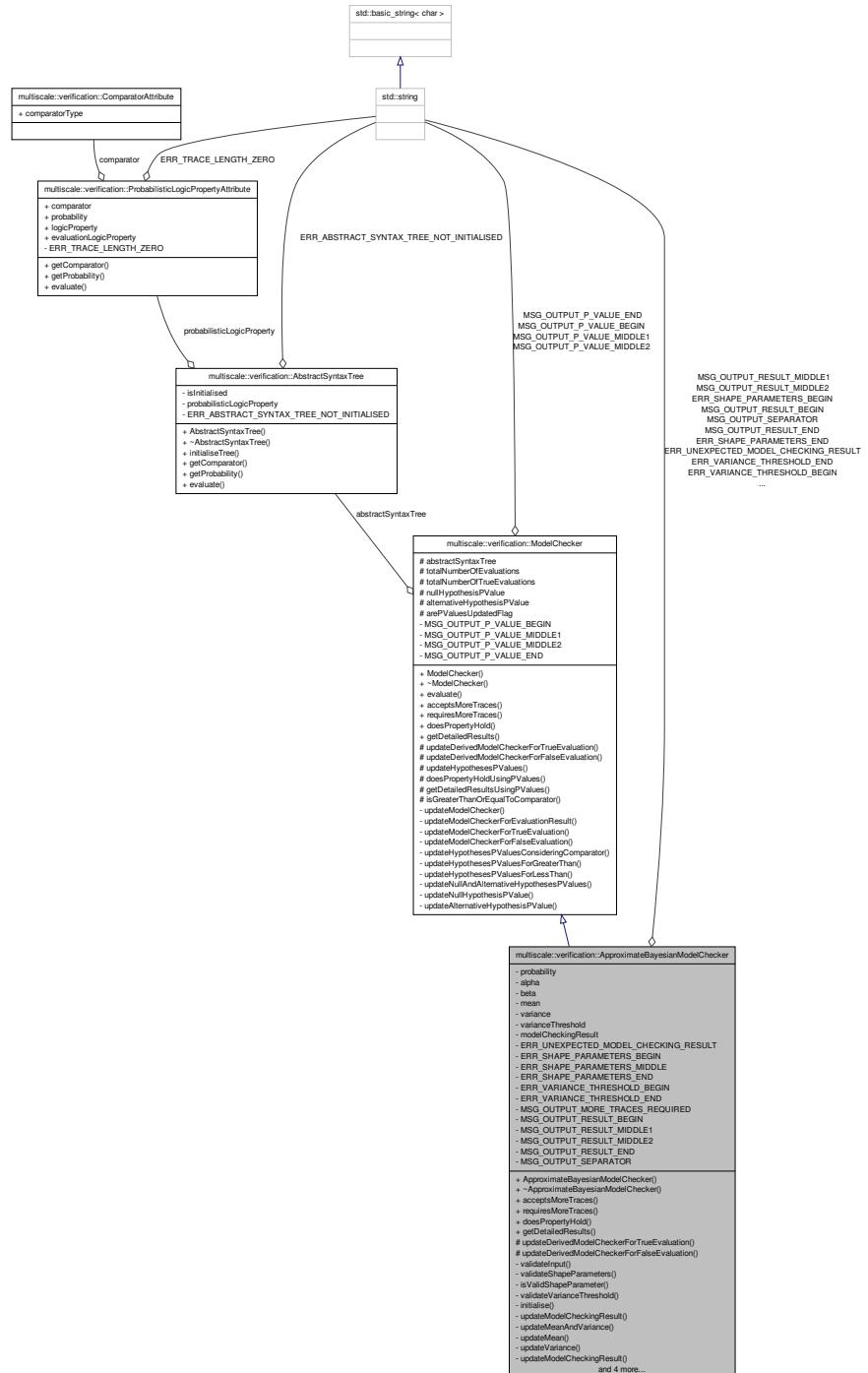
Class used to run approximate Bayesian model checking tasks.

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

Inheritance diagram for multiscale::verification::ApproximateBayesianModelChecker:



Collaboration diagram for multiscale::verification::ApproximateBayesianModelChecker:



Public Member Functions

- `ApproximateBayesianModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, 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 α , β and the variance threshold.
- `void validateShapeParameters (double alpha, double beta)`
Validate the shape parameters α and β .
- `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)`

- void `updateModelCheckingResultEnoughTraces (double variance)`

Update the result of the model checking task considering the given variance value.
- bool `isModelCheckingResultTrueConsideringComparator (double variance)`

Check if the result of the model checking task is true considering the probabilistic comparator (i.e. <=, >=)
- 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 ApproximateBayesian 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 `varianceThreshold` value = "
- static const std::string `MSG_OUTPUT_RESULT_END` = ""
- static const std::string `MSG_OUTPUT_SEPARATOR` = " "

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` = α

`beta` = β

`mean` = $\hat{\rho}$

`variance` = \hat{v}

`varianceThreshold` = T

`totalNumberOfEvaluations` = n

`totalNumberOfTrueEvaluations` = k

Definition at line 50 of file `ApproximateBayesianModelChecker.hpp`.

7.7.2 Constructor & Destructor Documentation

7.7.2.1 ApproximateBayesianModelChecker::ApproximateBayesianModelChecker (const AbstractSyntaxTree & abstractSyntaxTree, 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 23 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 25 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 35 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 143 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 41 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 162 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 82 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 131 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>shape- Parameter</i>	The given shape parameter
-----------------------------	---------------------------

Definition at line 67 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 31 of file ApproximateBayesianModelChecker.cpp.

References acceptsMoreTraces().

**7.7.3.10 void ApproximateBayesianModelChecker::updateDerivedModel-
CheckerForFalseEvaluation () [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 49 of file ApproximateBayesianModelChecker.cpp.

**7.7.3.11 void ApproximateBayesianModelChecker::updateDerivedModel-
CheckerForTrueEvaluation () [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 47 of file ApproximateBayesianModelChecker.cpp.

7.7.3.12 void ApproximateBayesianModelChecker::updateMean() [private]

Update the value of the mean estimate.

Definition at line 99 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 94 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 89 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 115 of file ApproximateBayesianModelChecker.cpp.

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

7.7.3.16 void ApproximateBayesianModelChecker::update-
ModelCheckingResultEnoughTraces (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 123 of file ApproximateBayesianModelChecker.cpp.

References multiscale::verification::FALSE, isModelCheckingResultTrueConsidering-
Comparator(), modelCheckingResult, and multiscale::verification::TRUE.

Referenced by updateModelCheckingResult().

7.7.3.17 void ApproximateBayesianModelChecker::updateModelCheckingResult-
NotEnoughTraces () [private]

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

Definition at line 139 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 106 of file ApproximateBayesianModelChecker.cpp.

References alpha, beta, multiscale::verification::ModelChecker::totalNumberOf-
Evaluations, 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 α , β and the variance threshold.

α , β and variance threshold should be greater than zero

Parameters

<i>alpha</i>	The shape parameter α for the Beta distribution
<i>beta</i>	The shape parameter β for the Beta distribution
<i>variance-Threshold</i>	The variance threshold

Definition at line 51 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 α and β .

α and β should be greater than zero

Parameters

<i>alpha</i>	The shape parameter α for the Beta distribution
<i>beta</i>	The shape parameter β for the Beta distribution

Definition at line 56 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>variance-Threshold</i>	The variance threshold
---------------------------	------------------------

Definition at line 71 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 α for the Beta distribution prior

Definition at line 57 of file ApproximateBayesianModelChecker.hpp.

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

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

The shape parameter β for the Beta distribution prior

Definition at line 58 of file ApproximateBayesianModelChecker.hpp.

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

7.7.4.3 const std::string ApproximateBayesianModelChecker::ERR_SHAPE_PARA-
METERS_BEGIN = "The provided Beta distribution shape parameters alpha and
beta (" [static, private]

Definition at line 182 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateShapeParameters().

7.7.4.4 const std::string ApproximateBayesianModelChecker::ERR_SHAPE_PARA-
METERS_END = ") should be greater than zero. Please change." [static,
private]

Definition at line 184 of file ApproximateBayesianModelChecker.hpp.

Referenced by validateShapeParameters().

7.7.4.5 const std::string ApproximateBayesianModelChecker::E-
RR_SHAPE_PARAMETERS_MIDDLE = ", " [static,
private]

Definition at line 183 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 180 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 186 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 187 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 60 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 65 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 189 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 191 of file ApproximateBayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`.

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

Definition at line 194 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 192 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 193 of file ApproximateBayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`.

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

Definition at line 196 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 54 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 61 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 63 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](#)

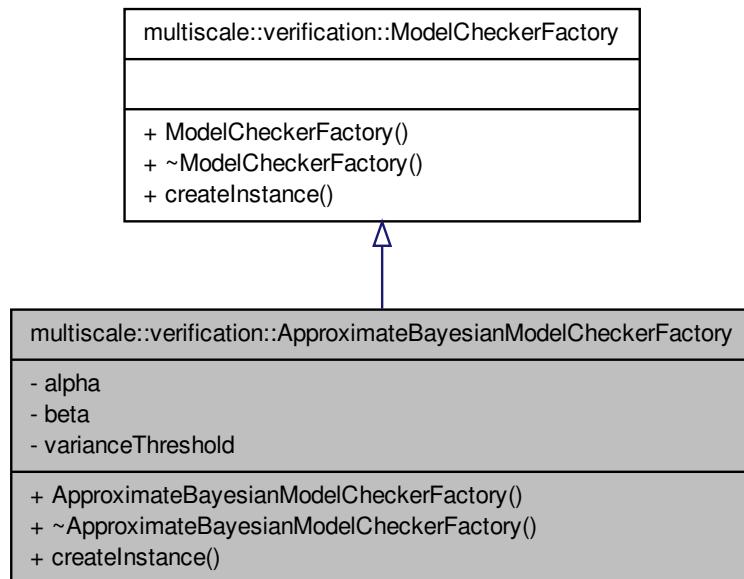
7.8 multiscale::verification::ApproximateBayesianModelChecker- Factory Class Reference

Class for creating [ApproximateBayesianModelChecker](#) instances.

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

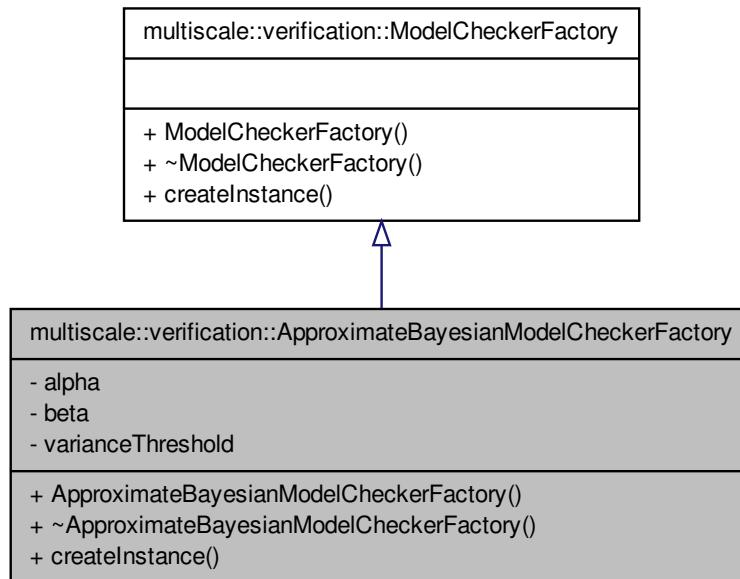
Inheritance diagram for multiscale::verification::ApproximateBayesianModelChecker-

Factory:



Collaboration diagram for `multiscale::verification::ApproximateBayesianModelChecker-`

Factory:



Public Member Functions

- `ApproximateBayesianModelCheckerFactory` (double `alpha`, double `beta`, double `varianceThreshold`)
- `~ApproximateBayesianModelCheckerFactory ()`
- `std::shared_ptr< ModelChecker > createInstance` (const `AbstractSyntaxTree` &`abstractSyntaxTree`) 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 `ApproximateBayesianModelCheckerFactory.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*) [override, virtual]**

Create an instance of [ApproximateBayesianModelChecker](#).

Parameters

<i>abstract-SyntaxTree</i>	The abstract syntax tree representing the logic property to be checked
----------------------------	--

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

Definition at line 15 of file ApproximateBayesianModelCheckerFactory.cpp.

7.8.4 Member Data Documentation

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

The shape parameter α for the Beta distribution prior

Definition at line 16 of file ApproximateBayesianModelCheckerFactory.hpp.

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

The shape parameter β for the Beta distribution prior

Definition at line 17 of file ApproximateBayesianModelCheckerFactory.hpp.

7.9 multiscaletest::ApproximateBayesianModelCheckerTest Class Reference 101

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

The variance threshold

Definition at line 19 of file ApproximateBayesianModelCheckerFactory.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/checking/[ApproximateBayesianModelCheckerFactory.hpp](#)

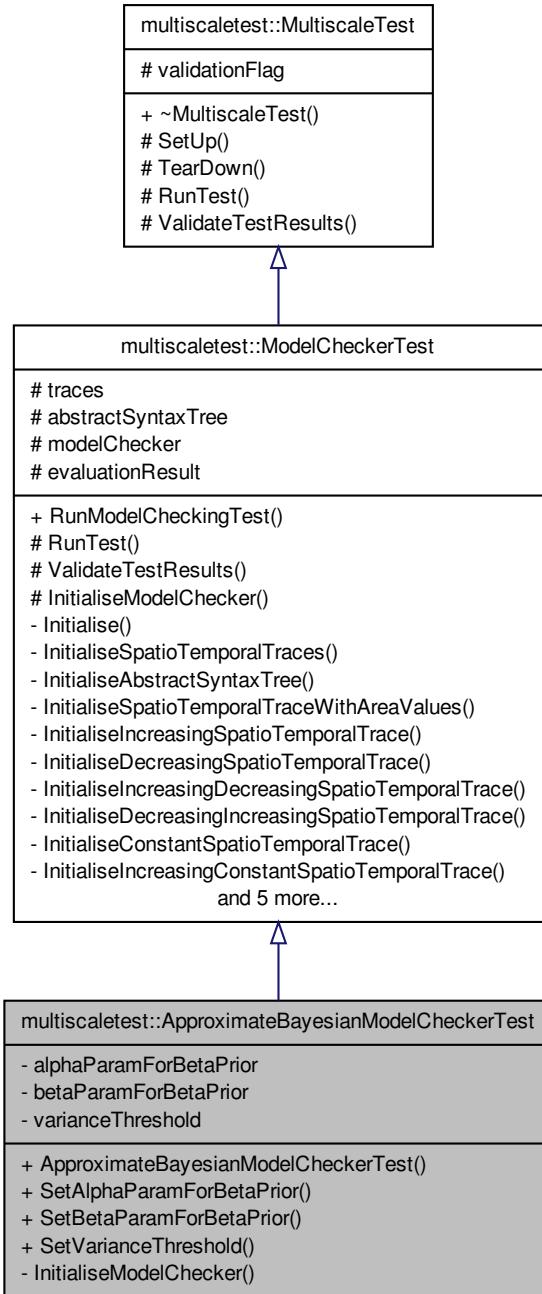
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ApproximateBayesianModelCheckerFactory.cpp](#)

7.9 multiscaletest::ApproximateBayesianModelCheckerTest Class Reference

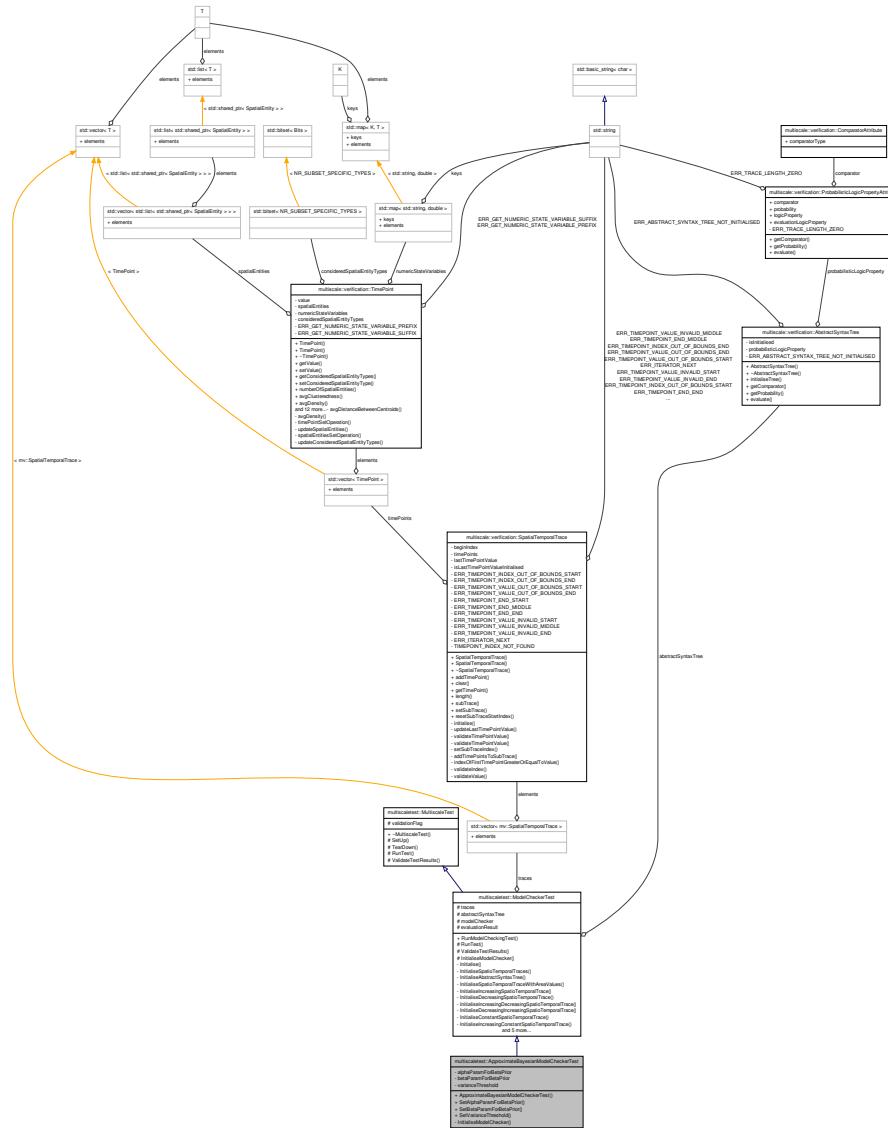
Class for testing the approximate Bayesian model checker.

```
#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 varianceThreshold)`

Set the value of the variance threshold.

Private Member Functions

- void [InitialiseModelChecker \(\) override](#)
Initialise the model checker.

Private Attributes

- double [alphaParamForBetaPrior](#)
- double [betaParamForBetaPrior](#)
- double [varianceThreshold](#)

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::ApproximateBayesianModelChecker-Test::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.

7.9 multiscaletest::ApproximateBayesianModelCheckerTest Class Reference 105

Parameters

<i>alphaParam-ForBetaPrior</i>	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>betaParam-ForBetaPrior</i>	The beta parameter for the beta prior
-------------------------------	---------------------------------------

Definition at line 60 of file ApproximateBayesianModelCheckerTest.hpp.

```
7.9.3.4 void multiscaletest::ApproximateBayesianModelChecker-  
        Test::SetVarianceThreshold ( double varianceThreshold  
        )
```

Set the value of the variance threshold.

Parameters

<i>variance-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::alpha-  
        ParamForBetaPrior [private]
```

The alpha parameter for the beta prior

Definition at line 19 of file ApproximateBayesianModelCheckerTest.hpp.

```
7.9.4.2 double multiscaletest::ApproximateBayesianModelCheckerTest::beta-  
        ParamForBetaPrior [private]
```

The beta parameter for the beta prior

Definition at line 20 of file ApproximateBayesianModelCheckerTest.hpp.

7.9.4.3 double multiscaletest::ApproximateBayesianModelCheckerTest::variance-Threshold [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](#)

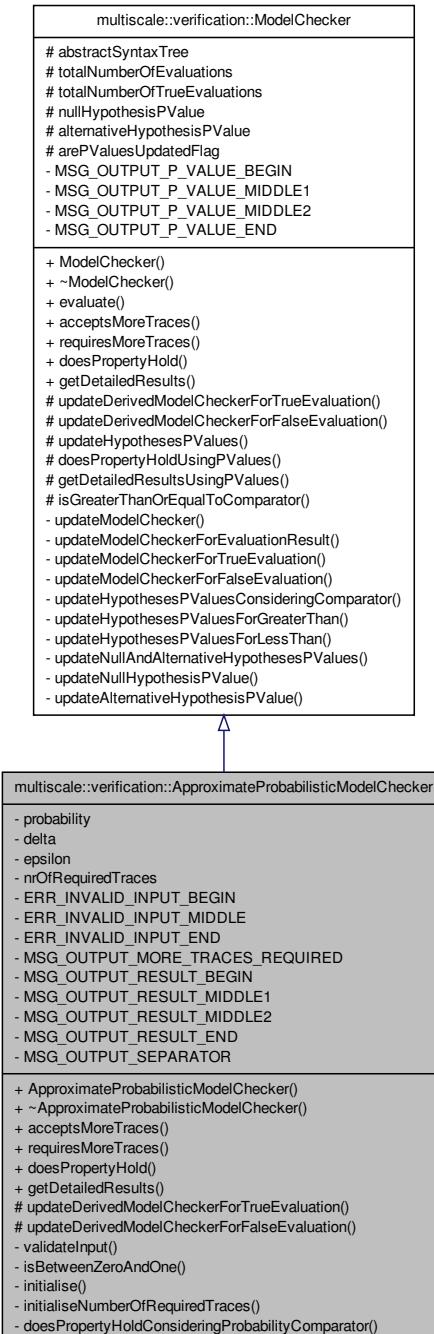
7.10 multiscale::verification::ApproximateProbabilisticModel- Checker Class Reference

Class used to run approximate probabilistic model checking tasks.

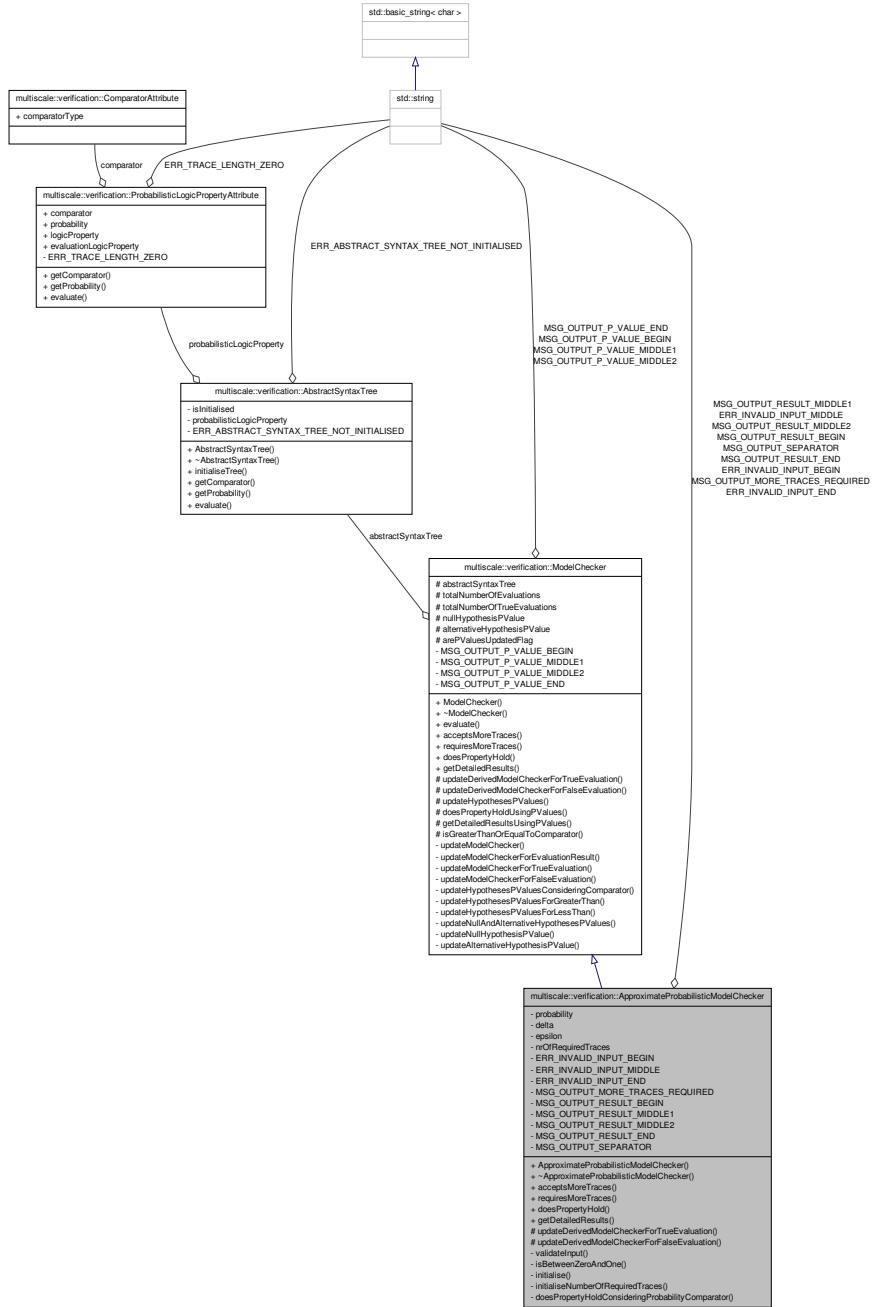
```
#include <ApproximateProbabilisticModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::ApproximateProbabilisticModel-

Checker:



Collaboration diagram for multiscale::verification::ApproximateProbabilisticModelChecker:



Public Member Functions

- `ApproximateProbabilisticModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, 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` = " "

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, 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 21 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 23 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 31 of file ApproximateProbabilisticModelChecker.cpp.

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

7.10.3.3 **bool ApproximateProbabilisticModelChecker::does-
PropertyHoldConsideringProbabilityComparator()**
[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 87 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 39 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 74 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 80 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

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

Definition at line 70 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 27 of file ApproximateProbabilisticModelChecker.cpp.

References acceptsMoreTraces().

Referenced by doesPropertyHold(), and getDetailedResults().

```
7.10.3.9 void ApproximateProbabilisticModelChecker::updateDerivedModel-
    CheckerForFalseEvaluation( ) [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 57 of file ApproximateProbabilisticModelChecker.cpp.

```
7.10.3.10 void ApproximateProbabilisticModelChecker::updateDerivedModel-
    CheckerForTrueEvaluation( ) [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 55 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 59 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_I-
INPUT_BEGIN = "The values of the provided input parameters delta and epsilon "
[static, private]**

Definition at line 103 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by validateInput().

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

Definition at line 105 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by validateInput().

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

Definition at line 104 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 107 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 109 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

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

Definition at line 112 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 110 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

7.10.4.10 `const std::string ApproximateProbabilisticModelChecker::MSG_OUT-
PUT_RESULT_MIDDLE2 = ". The number of required samples was N = "
[static, private]`

Definition at line 111 of file ApproximateProbabilisticModelChecker.hpp.

Referenced by getDetailedResults().

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

Definition at line 114 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](#)

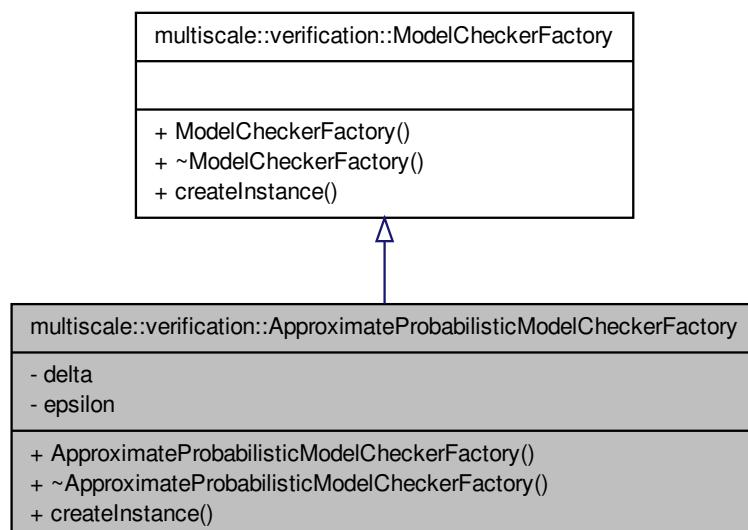
7.11 multiscale::verification::ApproximateProbabilisticModelCheckerFactory Class Reference

Class for creating [ApproximateProbabilisticModelChecker](#) instances.

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

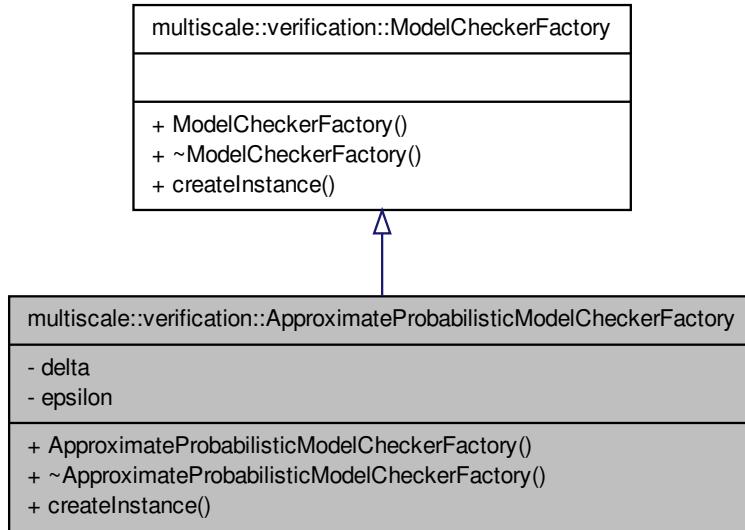
Inheritance diagram for multiscale::verification::ApproximateProbabilisticModel-

CheckerFactory:



Collaboration diagram for `multiscale::verification::ApproximateProbabilisticModel-`

CheckerFactory:



Public Member Functions

- `ApproximateProbabilisticModelCheckerFactory` (double `delta`, double `epsilon`)
- `~ApproximateProbabilisticModelCheckerFactory ()`
- `std::shared_ptr< ModelChecker > createInstance` (const `AbstractSyntaxTree` &`AbstractSyntaxTree`) 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 `ApproximateProbabilisticModelCheckerFactory.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*) [override, virtual]**

Create an instance of [ApproximateProbabilisticModelChecker](#).

Parameters

<i>abstract-SyntaxTree</i>	The abstract syntax tree representing the logic property to be checked
----------------------------	--

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

Definition at line 13 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](#)

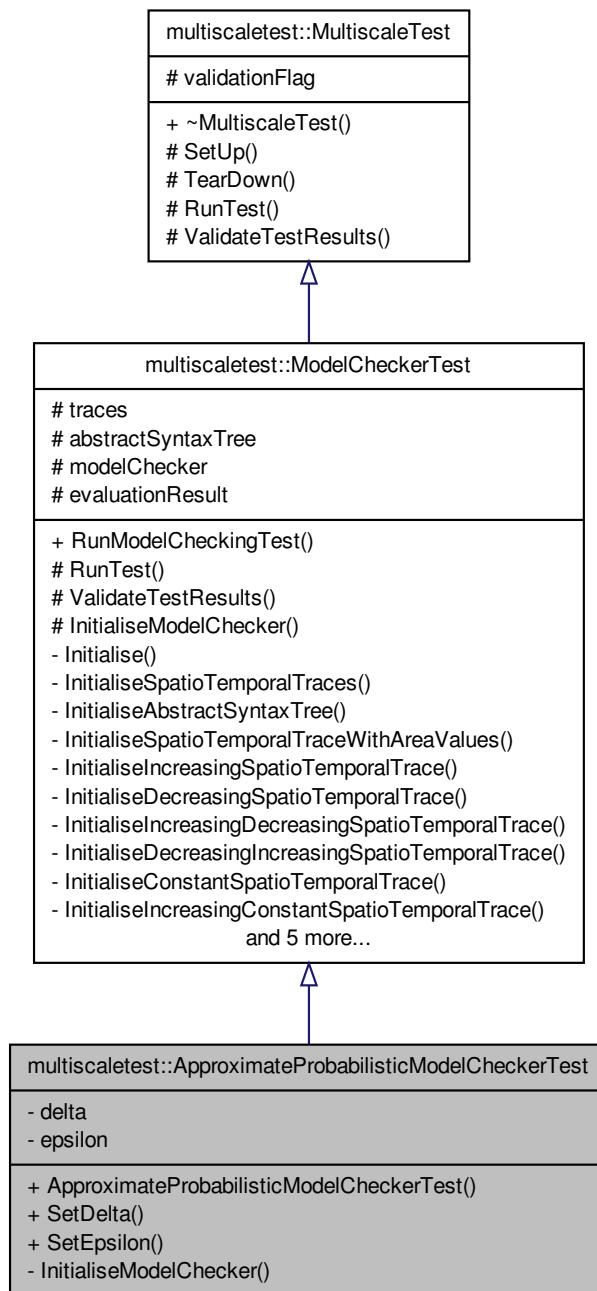
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ApproximateProbabilisticModelCheckerFactory.cpp](#)

7.12 multiscaletest::ApproximateProbabilisticModelCheckerTest Class Reference

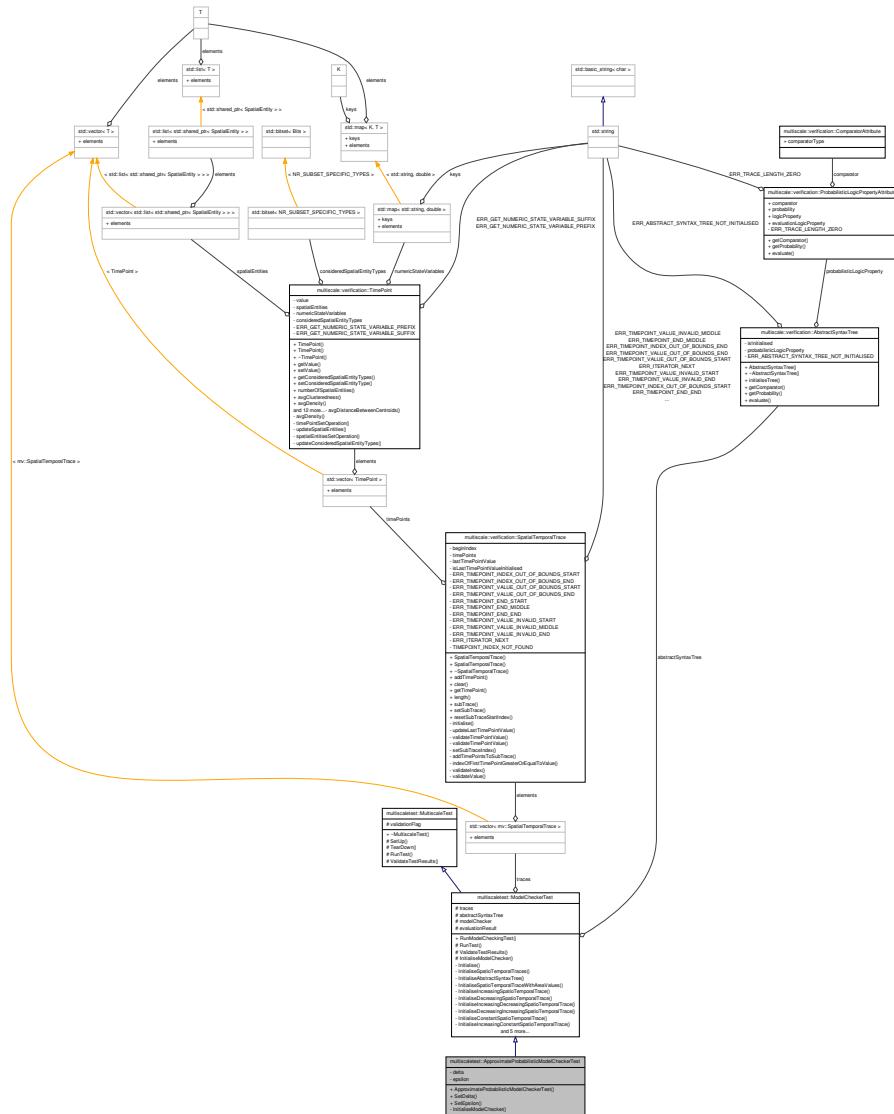
Class for testing the approximate probabilistic model checker.

```
#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](#)

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

<code>delta</code>	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

<i>epsilon</i>	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:

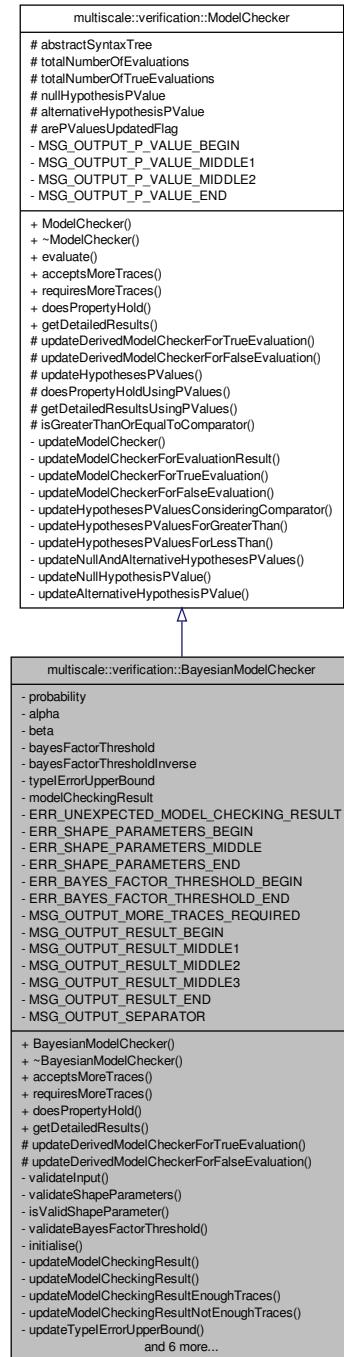
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/[ApproximateProbabilisticModelCheckerTest.hpp](#)

7.13 multiscale::verification::BayesianModelChecker Class - Reference

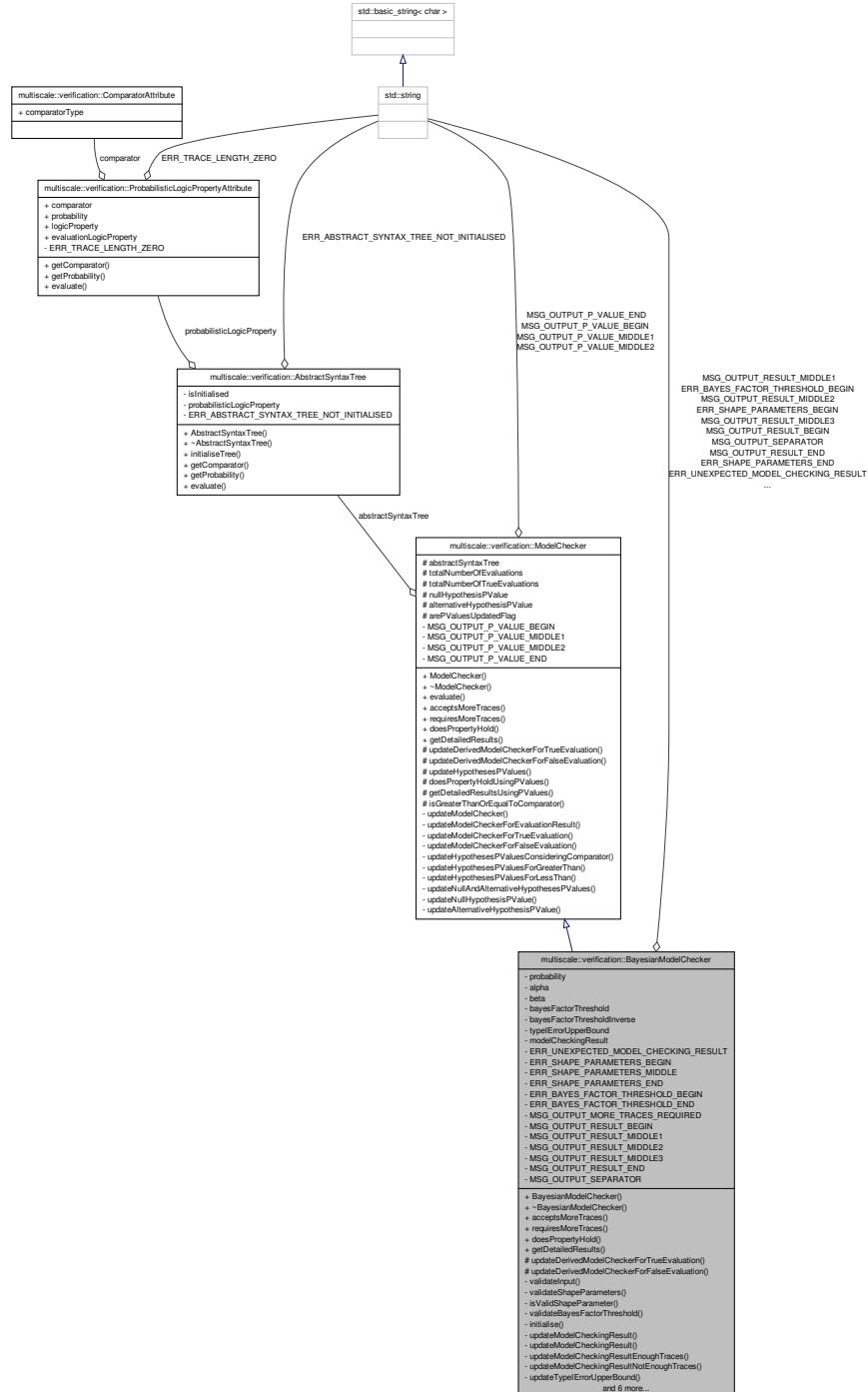
Class used to run Bayesian model checking tasks.

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

Inheritance diagram for multiscale::verification::BayesianModelChecker:



Collaboration diagram for multiscale::verification::BayesianModelChecker:



Public Member Functions

- BayesianModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, 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 α , β and the Bayes factor threshold.*
- void validateShapeParameters (double alpha, double beta)
 - Validate the shape parameters α and β .*
- 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` = " "

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` = θ

`alpha` = α

`beta` = β

`bayesFactor` = \mathcal{B}_n

`bayesFactorThreshold` = T

`totalNumberOfEvaluations` = n

`totalNumberOfTrueEvaluations` = x

Definition at line 49 of file BayesianModelChecker.hpp.

7.13.2 Constructor & Destructor Documentation

7.13.2.1 BayesianModelChecker::BayesianModelChecker (const AbstractSyntaxTree & *abstractSyntaxTree*, 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 27 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 29 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

<i>nrOf- Observations</i>	The total number of observations
<i>nrOf- Successes</i>	The total number of successes

Definition at line 157 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>nrOf- Successes</i>	The number of successful observations/trials
<i>probability</i>	The probability of success

Definition at line 151 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>nrOf- Successes</i>	The number of successful observations/trials
----------------------------	--

Definition at line 141 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 39 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 *!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 188 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 169 of file BayesianModelChecker.cpp.

References doesPropertyHoldConsideringProbabilityComparator(), 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().

7.13.3.8 std::string BayesianModelChecker::getDetailedResults () [override, virtual]

Get the detailed description of the results.

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

Definition at line 45 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 196 of file BayesianModelChecker.cpp.

References alpha, bayesFactorThreshold, 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_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 135 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 86 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>shape-Parameter</i>	The given shape parameter
------------------------	---------------------------

Definition at line 71 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 35 of file BayesianModelChecker.cpp.

References acceptsMoreTraces().

7.13.3.14 void BayesianModelChecker::updateDerivedModelChecker-ForFalseEvaluation() [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 BayesianModelChecker.cpp.

7.13.3.15 void BayesianModelChecker::updateDerivedModelChecker-ForTrueEvaluation() [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 BayesianModelChecker.cpp.

7.13.3.16 void BayesianModelChecker::updateModelCheckingResult() [private]

Update the result of the model checking task.

Definition at line 96 of file BayesianModelChecker.cpp.

References computeBayesFactorValue(), [multiscale::verification::ModelChecker::total-Number-Of-Evaluations](#), [multiscale::verification::ModelChecker::totalNumber-Of-True-Evaluations](#), 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 104 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 113 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 121 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 125 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>bayesFactor- Threshold</i>	The Bayes factor threshold
-----------------------------------	----------------------------

Definition at line 75 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 α , β and the Bayes factor threshold.

α and β should be greater than zero, and Bayes factor threshold should be greater than 1

Parameters

<i>alpha</i>	The shape parameter α for the Beta distribution
<i>beta</i>	The shape parameter β for the Beta distribution
<i>bayesFactor- Threshold</i>	The Bayes factor threshold

Definition at line 55 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 α and β .

α and β should be greater than zero

Parameters

<i>alpha</i>	The shape parameter α for the Beta distribution
<i>beta</i>	The shape parameter β for the Beta distribution

Definition at line 60 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 α for the Beta distribution prior

Definition at line 56 of file BayesianModelChecker.hpp.

Referenced by BayesianModelChecker(), computeBayesFactorValue(), and getDetailedUpdatedResults().

7.13.4.2 double multiscale::verification::BayesianModelChecker::bayesFactor- Threshold [private]

The Bayes factor threshold

Definition at line 59 of file BayesianModelChecker.hpp.

Referenced by BayesianModelChecker(), getDetailedUpdatedResults(), initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.13.4.3 double multiscale::verification::BayesianModelChecker::bayesFactor- ThresholdInverse [private]

The Bayes factor threshold to the power "-1"

Definition at line 60 of file BayesianModelChecker.hpp.

Referenced by indicatorFunction(), initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.13.4.4 double multiscale::verification::BayesianModelChecker::beta [private]

The shape parameter β for the Beta distribution prior

Definition at line 57 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 208 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 209 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 204 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 206 of file BayesianModelChecker.hpp.

Referenced by validateShapeParameters().

```
7.13.4.9 const std::string BayesianModelChecker::ERR_SHAPE_PARAMETERS_MIDDLE = ", " [static, private]
```

Definition at line 205 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 202 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 64 of file BayesianModelChecker.hpp.

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

**7.13.4.12 const std::string BayesianModelChecker::MSG_OUTPUT_MORE_TR-
ACES_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 211 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 213 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

**7.13.4.14 const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_END =
"" [static, private]**

Definition at line 217 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

**7.13.4.15 const std::string BayesianModelChecker::MSG_OUTP-
UT_RESULT_MIDDLE1 = " and beta = " [static,
private]**

Definition at line 214 of file BayesianModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.13.4.16 `const std::string BayesianModelChecker::MSG_OUTPUT_RESU-LT_MIDDLE2 = "", and Bayes factor threshold value = "` [static, private]

Definition at line 215 of file BayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`.

7.13.4.17 `const std::string BayesianModelChecker::MSG_OUTPUT_RESULT_MIDD-LE3 = ". The type I error upper bound for the provided answer is = "` [static, private]

Definition at line 216 of file BayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`.

7.13.4.18 `const std::string BayesianModelChecker::MSG_OUTPUT_SEPARATOR = "` [static, private]

Definition at line 219 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 53 of file BayesianModelChecker.hpp.

Referenced by `computeBayesFactorValue()`, `computeMaximumBinomialPDF()`, and `initialise()`.

7.13.4.20 `double multiscale::verification::BayesianModelChecker::typeIError-UppernBound` [private]

The type I error upper bound

Definition at line 62 of file BayesianModelChecker.hpp.

Referenced by `getDetailedUpdatedResults()`, `initialise()`, and `updateTypeIErrorUpperBound()`.

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

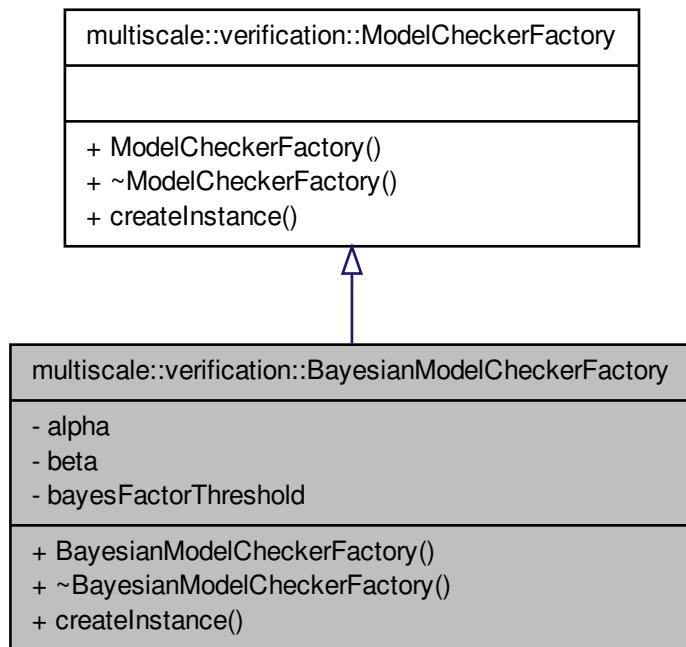
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[Bayesian-ModelChecker.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[BayesianModelChecker.cpp](#)

7.14 multiscale::verification::BayesianModelCheckerFactory Class Reference

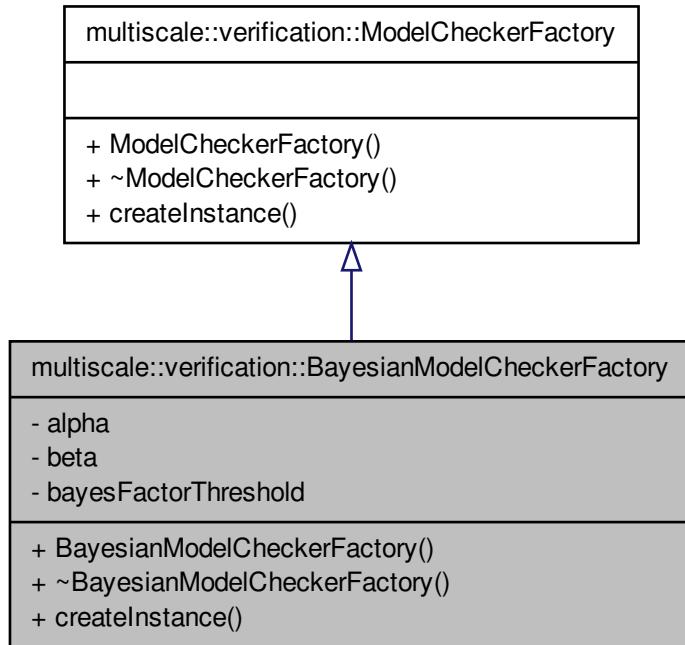
Class for creating [BayesianModelChecker](#) instances.

```
#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) 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::create-Instance (const AbstractSyntaxTree & *abstractSyntaxTree*) [override, virtual]

Create an instance of [BayesianModelChecker](#).

Parameters

<i>abstract-SyntaxTree</i>	The abstract syntax tree representing the logic property to be checked
----------------------------	--

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

Definition at line 15 of file BayesianModelCheckerFactory.cpp.

7.14.4 Member Data Documentation

7.14.4.1 double multiscale::verification::BayesianModelCheckerFactory::alpha [private]

The shape parameter α for the Beta distribution prior

Definition at line 16 of file BayesianModelCheckerFactory.hpp.

7.14.4.2 double multiscale::verification::BayesianModelCheckerFactory::bayes-FactorThreshold [private]

The Bayes factor threshold

Definition at line 19 of file BayesianModelCheckerFactory.hpp.

7.14.4.3 double multiscale::verification::BayesianModelCheckerFactory::beta [private]

The shape parameter β for the Beta distribution prior

Definition at line 17 of file BayesianModelCheckerFactory.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/checking/[Bayesian-ModelCheckerFactory.hpp](#)

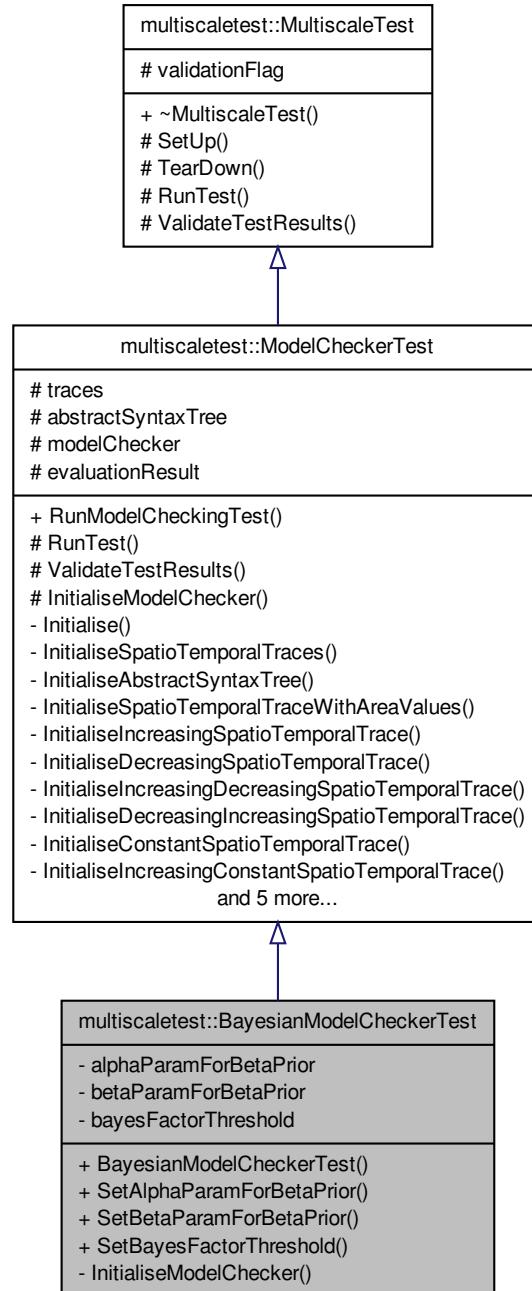
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[BayesianModelCheckerFactory.cpp](#)

7.15 multiscaletest::BayesianModelCheckerTest Class Reference

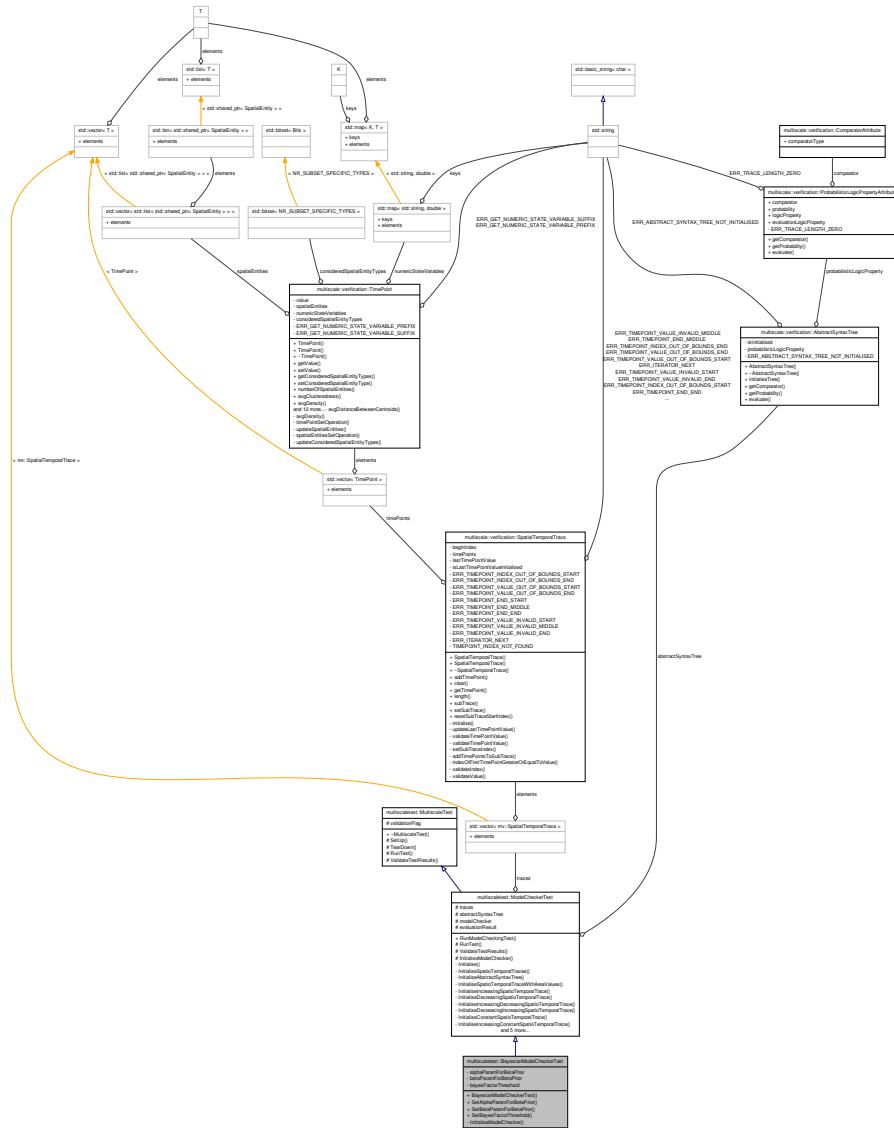
Class for testing the Bayesian model checker.

```
#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](#)

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::Set- AlphaParamForBetaPrior** (double *alphaParamForBetaPrior*)

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

Parameters

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

Definition at line 56 of file BayesianModelCheckerTest.hpp.

```
7.15.3.3 void multiscaletest::BayesianModelCheckerTest::Set-
    BayesFactorThreshold ( double bayesFactorThreshold
    )
```

Set the value of the Bayes factor threshold.

Parameters

<i>bayesFactor-</i>	The value of the Bayes factor threshold
<i>Threshold</i>	

Definition at line 64 of file BayesianModelCheckerTest.hpp.

```
7.15.3.4 void multiscaletest::BayesianModelCheckerTest::Set-
    BetaParamForBetaPrior ( double betaParamForBetaPrior
    )
```

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

Parameters

<i>betaParam-</i>	The beta parameter for the beta prior
<i>ForBetaPrior</i>	

Definition at line 60 of file BayesianModelCheckerTest.hpp.

7.15.4 Member Data Documentation

```
7.15.4.1 double multiscaletest::BayesianModelCheckerTest::alphaParamForBeta-
    Prior [private]
```

The alpha parameter for the beta prior

Definition at line 19 of file BayesianModelCheckerTest.hpp.

```
7.15.4.2 double multiscaletest::BayesianModelCheckerTest::bayesFactor-
    Threshold [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:

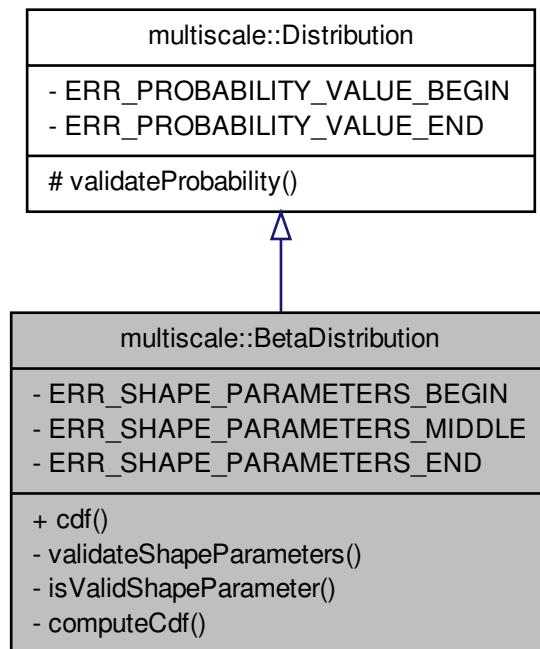
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/[BayesianModelCheckerTest.hpp](#)

7.16 multiscale::BetaDistribution Class Reference

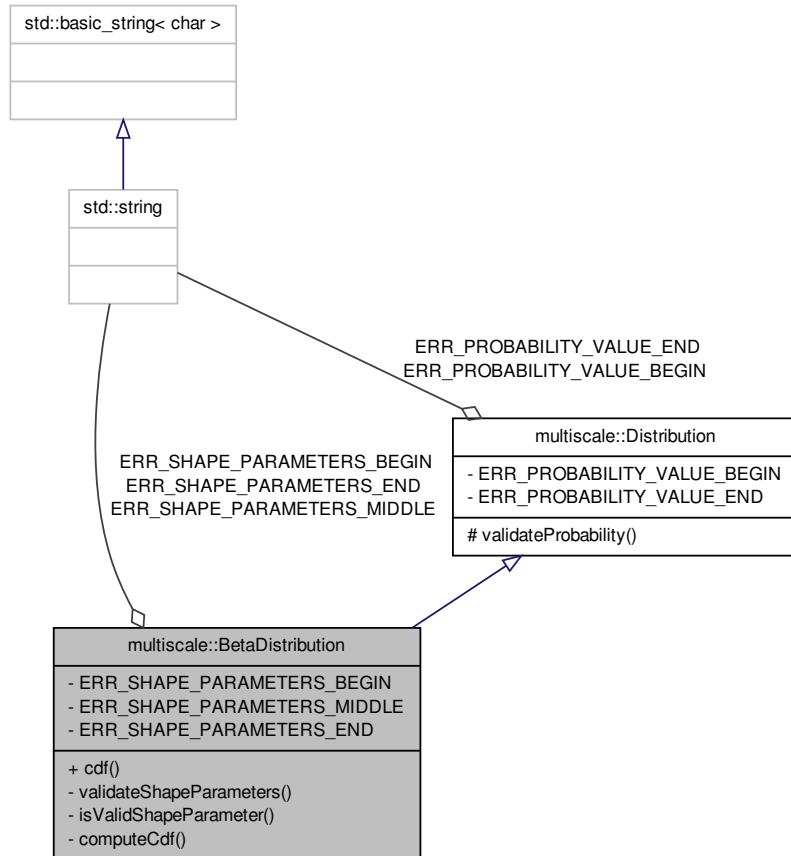
Class for analysing Beta distributed data.

```
#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 α and β .
- 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."

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

<code>alpha</code>	Shape parameter <i>alpha</i>
<code>beta</code>	Shape parameter <i>beta</i>
<code>probability</code>	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()`, `computeCdf()`, 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.

References `cdf()`.

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>shape-Parameter</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 α and β .

α and β should be greater than zero

Parameters

<i>alpha</i>	The shape parameter α for the Beta distribution
<i>beta</i>	The shape parameter β 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:

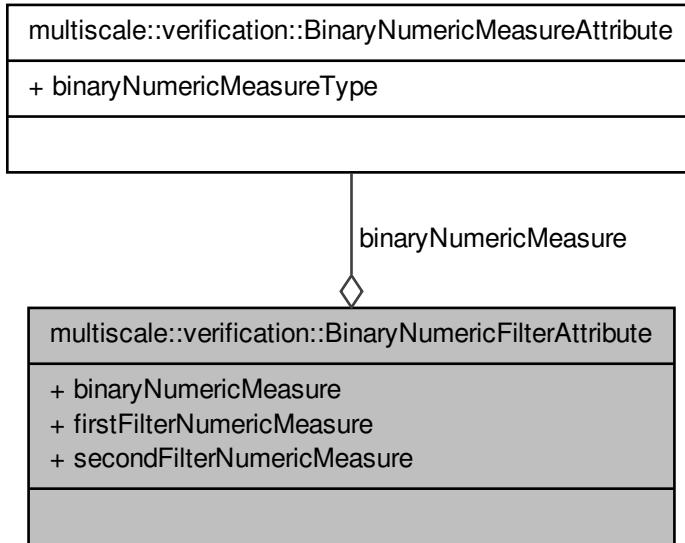
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/-BetaDistribution.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-BetaDistribution.cpp](#)

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 19 of file [BinaryNumericFilterAttribute.hpp](#).

7.18 multiscale::verification::BinaryNumericMeasureAttribute Class Reference

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

7.17.2.2 FilterNumericMeasureAttributeType multiscale::verification::Binary- NumericFilterAttribute::firstFilterNumericMeasure

The first filter numeric measure

Definition at line 20 of file BinaryNumericFilterAttribute.hpp.

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

7.17.2.3 FilterNumericMeasureAttributeType multiscale::verification::Binary- NumericFilterAttribute::secondFilterNumericMeasure

The second filter numeric measure

Definition at line 21 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>
```

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::BinaryNumericMeasureTypeParser Struct Reference

Symbol table and parser for the binary numeric measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [BinaryNumericMeasureTypeParser \(\)](#)

7.19.1 Detailed Description

Symbol table and parser for the binary numeric measure type.

Definition at line 98 of file SymbolTables.hpp.

7.19.2 Constructor & Destructor Documentation

7.19.2.1 **multiscale::verification::BinaryNumericMeasureTypeParser::BinaryNumericMeasureTypeParser () [inline]**

Definition at line 100 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:

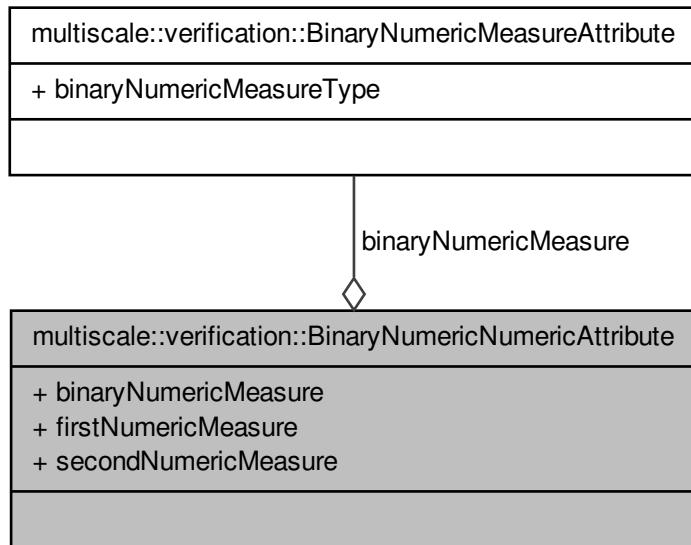
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.hpp](#)

7.20 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`
- `NumericMeasureAttributeType firstNumericMeasure`
- `NumericMeasureAttributeType secondNumericMeasure`

7.20.1 Detailed Description

Class for representing a binary numeric numeric measure attribute.

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

7.20.2 Member Data Documentation

7.20.2.1 **BinaryNumericMeasureAttribute multiscale::verification::BinaryNumericNumericAttribute::binaryNumericMeasure**

The binary numeric measure

Definition at line 19 of file `BinaryNumericNumericAttribute.hpp`.

7.20.2.2 **NumericMeasureAttributeType multiscale::verification::BinaryNumericNumericAttribute::firstNumericMeasure**

The first numeric measure

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

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

7.20.2.3 **NumericMeasureAttributeType 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:

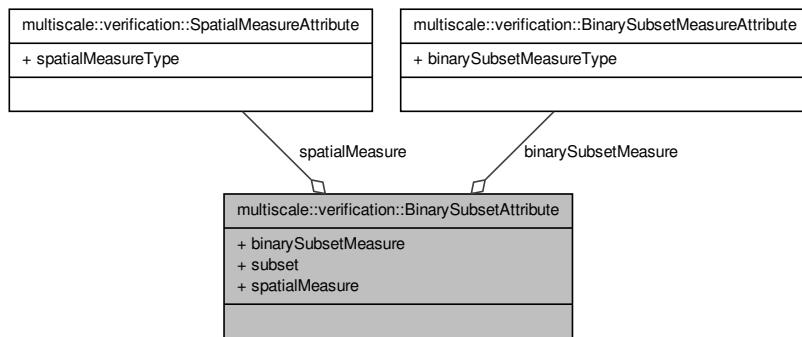
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinaryNumericNumericAttribute.hpp`

7.21 **multiscale::verification::BinarySubsetAttribute Class Reference**

Class for representing a binary subset attribute.

```
#include <BinarySubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::BinarySubsetAttribute:



Public Attributes

- `BinarySubsetMeasureAttribute binarySubsetMeasure`
- `SubsetAttributeType subset`
- `SpatialMeasureAttribute spatialMeasure`

7.21.1 Detailed Description

Class for representing a binary subset attribute.

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

7.21.2 Member Data Documentation

7.21.2.1 `BinarySubsetMeasureAttribute multiscale::verification::BinarySubsetAttribute::binarySubsetMeasure`

The binary subset measure

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

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

7.21.2.2 `SpatialMeasureAttribute multiscale::verification::BinarySubsetAttribute::spatialMeasure`

The considered spatial measure

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

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

7.21.2.3 **SubsetAttributeType multiscale::verification::BinarySubsetAttribute::subset**

The considered subset

Definition at line 21 of file BinarySubsetAttribute.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/[BinarySubsetAttribute.hpp](#)

7.22 multiscale::verification::BinarySubsetMeasureAttribute Class Reference

Class for representing a binary subset measure attribute.

```
#include <BinarySubsetMeasureAttribute.hpp>
```

Public Attributes

- [BinarySubsetMeasureType binarySubsetMeasureType](#)

7.22.1 Detailed Description

Class for representing a binary subset measure attribute.

Definition at line 39 of file BinarySubsetMeasureAttribute.hpp.

7.22.2 Member Data Documentation

7.22.2.1 **BinarySubsetMeasureType multiscale::verification::BinarySubsetMeasureAttribute::binarySubsetMeasureType**

The binary subset measure type

Definition at line 43 of file BinarySubsetMeasureAttribute.hpp.

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

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

7.23 multiscale::verification::BinarySubsetMeasureTypeParser Struct Reference

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[BinarySubsetMeasureAttribute.hpp](#)

7.23 multiscale::verification::BinarySubsetMeasureTypeParser - Struct Reference

Symbol table and parser for the binary subset measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [BinarySubsetMeasureTypeParser \(\)](#)

7.23.1 Detailed Description

Symbol table and parser for the binary subset measure type.

Definition at line 155 of file SymbolTables.hpp.

7.23.2 Constructor & Destructor Documentation

7.23.2.1 multiscale::verification::BinarySubsetMeasureTypeParser::BinarySubsetMeasureTypeParser () [inline]

Definition at line 157 of file SymbolTables.hpp.

References multiscale::verification::Avg, 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:

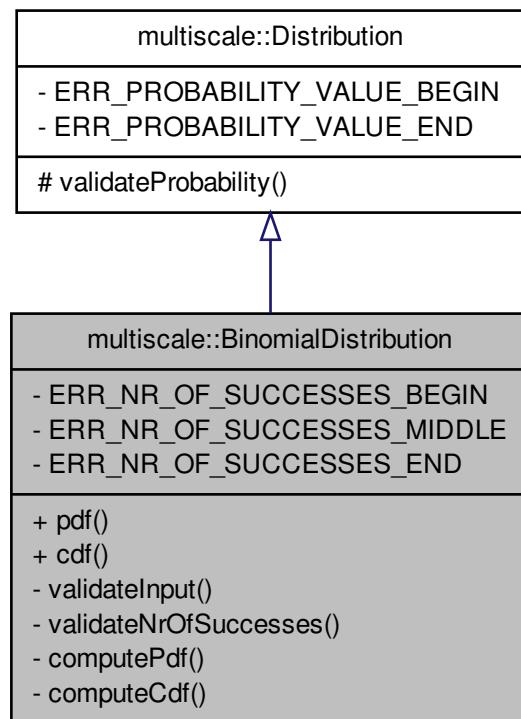
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.hpp](#)

7.24 multiscale::BinomialDistribution Class Reference

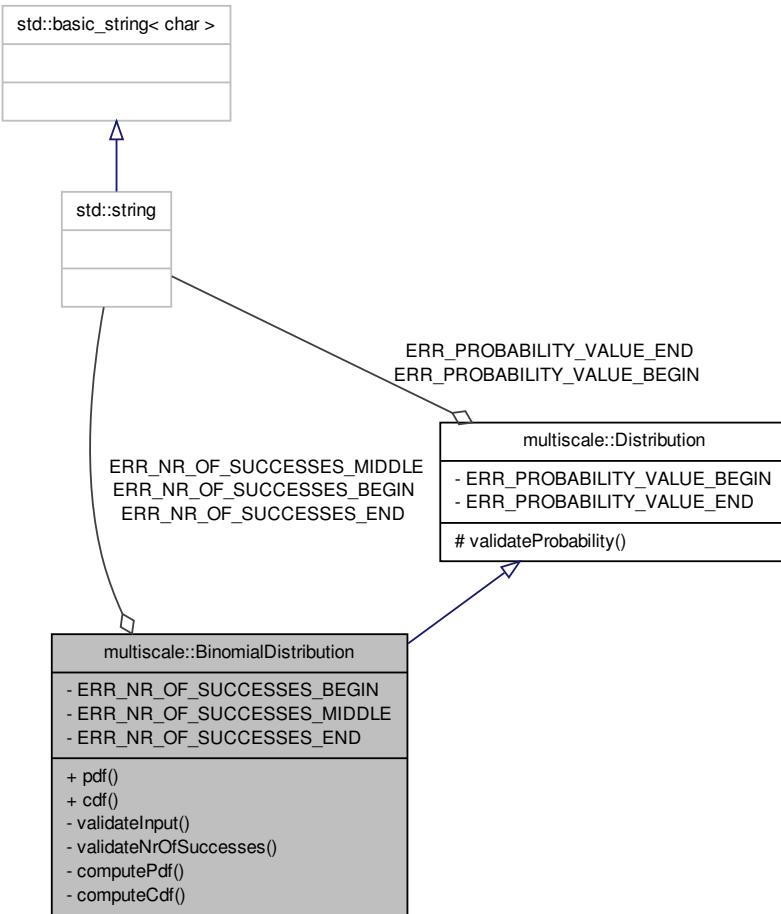
Class for analysing Binomial distributed data.

```
#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)

Validate the given input data.
- static void `validateNrOfSuccesses` (unsigned int nrOfObservations, unsigned int nrOfSuccesses)

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` = ")."

7.24.1 Detailed Description

Class for analysing Binomial distributed data.

Definition at line 12 of file BinomialDistribution.hpp.

7.24.2 Member Function Documentation

7.24.2.1 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

<code>nrOfObservations</code>	The total number of observations
<code>nrOfSuccesses</code>	The number of successes
<code>probability</code>	The probability p used by the cumulative distribution function

Definition at line 17 of file BinomialDistribution.cpp.

References computeCdf(), and validateInput().

Referenced by computeCdf(), TEST(), multiscale::verification::ModelChecker::updateAlternativeHypothesisPValue(), and multiscale::verification::ModelChecker::updateNullHypothesisPValue().

**7.24.2.2 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.

References cdf().

Referenced by cdf().

**7.24.2.3 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.

References pdf().

Referenced by pdf().

**7.24.2.4 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>nrOf- Observations</i>	The total number of observations
<i>nrOf- Successes</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(), computePdf(), and TEST().

**7.24.2.5 void BinomialDistribution::validateInput (unsigned int *nrOfObservations*,
unsigned int *nrOfSuccesses*, double *probability*) [static, private]**

Validate the given input data.

Parameters

<i>nrOf- Observations</i>	The total number of observations
<i>nrOf- Successes</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().

**7.24.2.6 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>nrOf- Observations</i>	The total number of observations
<i>nrOf- Successes</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().

7.24.3 Member Data Documentation

7.24.3.1 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().

7.24.3.2 const std::string BinomialDistribution::ERR_NR_OF_SUCCESSES_END = ")." [static, private]

Definition at line 75 of file BinomialDistribution.hpp.

Referenced by validateNrOfSuccesses().

7.24.3.3 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:

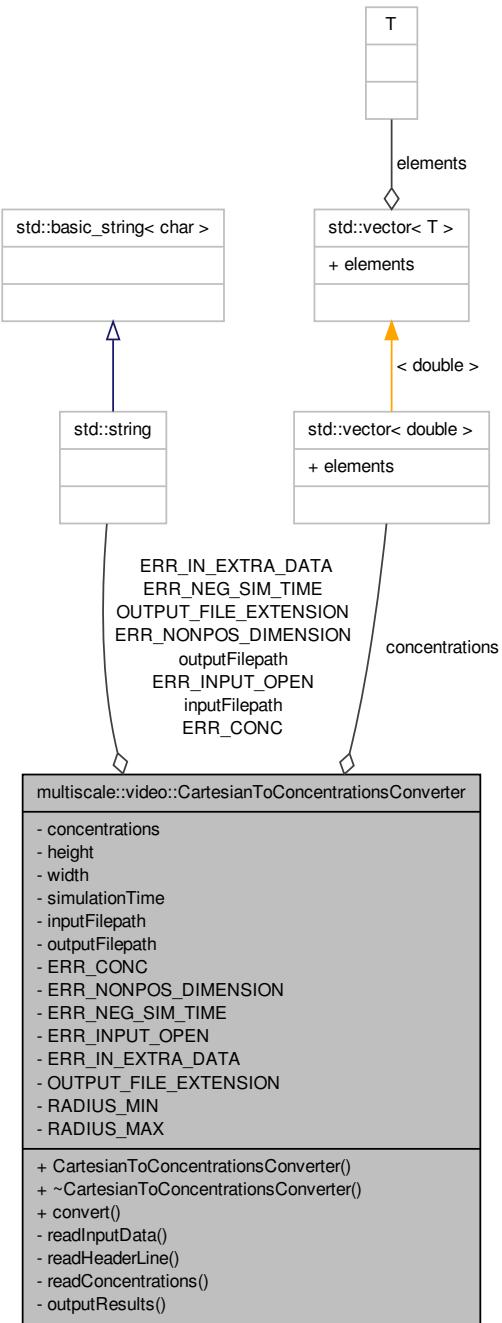
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/BinomialDistribution.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/statistics/BinomialDistribution.cpp](#)

7.25 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 string &`inputFilepath`, const string &`outputFilepath`)
- `~CartesianToConcentrationsConverter` ()
- void `convert` ()

Start the conversion.

Private Member Functions

- void `readInputData` ()
Read the input data.
- void `readHeaderLine` (ifstream &`fin`)
Read the header line.
- void `readConcentrations` (ifstream &`fin`)
Read the concentrations.
- void `outputResults` ()
Output the results.

Private Attributes

- vector< double > `concentrations`
- unsigned long `height`
- unsigned long `width`
- double `simulationTime`
- string `inputFilepath`
- string `outputFilepath`

Static Private Attributes

- static const string `ERR_CONC` = "All concentrations have to be between 0 and 1."
- static const string `ERR_NONPOS_DIMENSION` = "The dimensions N and M must be positive."
- static const string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const string `ERR_INPUT_OPEN` = "The input file could not be opened"
- static const string `ERR_IN_EXTRA_DATA` = "The input file contains more data than required."
- static const string `OUTPUT_FILE_EXTENSION` = ".out"
- static const double `RADIUS_MIN` = 0.001
- static const double `RADIUS_MAX` = 0.3

7.25.1 Detailed Description

Scale the values of the rectangular geometry grid cells.

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

7.25.2 Constructor & Destructor Documentation

7.25.2.1 `CartesianToConcentrationsConverter::CartesianToConcentrationsConverter (const string & inputFilepath, const string & outputFilepath)`

Definition at line 16 of file `CartesianToConcentrationsConverter.cpp`.

References `height`, `simulationTime`, and `width`.

7.25.2.2 `CartesianToConcentrationsConverter::~CartesianToConcentrationsConverter ()`

Definition at line 25 of file `CartesianToConcentrationsConverter.cpp`.

7.25.3 Member Function Documentation

7.25.3.1 `void CartesianToConcentrationsConverter::convert ()`

Start the conversion.

Definition at line 27 of file `CartesianToConcentrationsConverter.cpp`.

References `outputResults()`, and `readInputData()`.

Referenced by `main()`.

7.25.3.2 `void CartesianToConcentrationsConverter::outputResults () [private]`

Output the results.

Definition at line 85 of file `CartesianToConcentrationsConverter.cpp`.

References `concentrations`, `multiscale::video::RectangularGnuplotScriptGenerator::generateScript()`, `height`, `outputFilepath`, `simulationTime`, and `width`.

Referenced by `convert()`.

7.25.3.3 `void CartesianToConcentrationsConverter::readConcentrations (ifstream & fin) [private]`

Read the concentrations.

7.25 multiscale::video::CartesianToConcentrationsConverter Class Reference

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 65 of file CartesianToConcentrationsConverter.cpp.

References concentrations, ERR_CONC, height, MS_throw, and width.

Referenced by readInputData().

7.25.3.4 void *CartesianToConcentrationsConverter::readHeaderLine*(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 56 of file CartesianToConcentrationsConverter.cpp.

References ERR_NEG_SIM_TIME, ERR_NONPOS_DIMENSION, height, MS_throw, simulationTime, and width.

Referenced by readInputData().

7.25.3.5 void *CartesianToConcentrationsConverter::readInputData*() [private]

Read the input data.

Definition at line 32 of file CartesianToConcentrationsConverter.cpp.

References ERR_IN_EXTRA_DATA, ERR_INPUT_OPEN, inputFilepath, MS_throw, readConcentrations(), and readHeaderLine().

Referenced by convert().

7.25.4 Member Data Documentation

7.25.4.1 vector<double> *multiscale::video::CartesianToConcentrationsConverter::concentrations* [private]

Concentrations received as input

Definition at line 19 of file CartesianToConcentrationsConverter.hpp.

Referenced by outputResults(), and readConcentrations().

7.25.4.2 `const string CartesianToConcentrationsConverter::ERR_CONC = "All concentrations have to be between 0 and 1." [static, private]`

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

Referenced by `readConcentrations()`.

7.25.4.3 `const string CartesianToConcentrationsConverter::ERR_IN_EXTRA_DATA = "The input file contains more data than required." [static, private]`

Definition at line 66 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

7.25.4.4 `const string CartesianToConcentrationsConverter::ERR_INPUT_OPEN = "The input file could not be opened" [static, private]`

Definition at line 65 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

7.25.4.5 `const string CartesianToConcentrationsConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative." [static, private]`

Definition at line 64 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readHeaderLine()`.

7.25.4.6 `const string CartesianToConcentrationsConverter::ERR_NONPOS_DIMENSION = "The dimensions N and M must be positive." [static, private]`

Definition at line 63 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readHeaderLine()`.

7.25.4.7 `unsigned long multiscale::video::CartesianToConcentrationsConverter::height [private]`

Height of the grid

Definition at line 21 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `CartesianToConcentrationsConverter()`, `outputResults()`, `readConcentrations()`, and `readHeaderLine()`.

7.25.4.8 string multiscale::video::CartesianToConcentrationsConverter::input-Filepath [private]

Path to the input file

Definition at line 25 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `readInputData()`.

7.25.4.9 const string CartesianToConcentrationsConverter-::OUTPUT_FILE_EXTENSION = ".out" [static, private]

Definition at line 68 of file `CartesianToConcentrationsConverter.hpp`.

7.25.4.10 string multiscale::video::CartesianToConcentrationsConverter::output-Filepath [private]

Path to the output file

Definition at line 26 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `outputResults()`.

7.25.4.11 const double CartesianToConcentrationsConverter::RADIUS_MAX = 0.3 [static, private]

Definition at line 71 of file `CartesianToConcentrationsConverter.hpp`.

7.25.4.12 const double CartesianToConcentrationsConverter::RADIUS_MIN = 0.001 [static, private]

Definition at line 70 of file `CartesianToConcentrationsConverter.hpp`.

7.25.4.13 double multiscale::video::CartesianToConcentrationsConverter-::simulationTime [private]

Simulation time

Definition at line 23 of file `CartesianToConcentrationsConverter.hpp`.

Referenced by `CartesianToConcentrationsConverter()`, `outputResults()`, and `readHeaderLine()`.

7.25.4.14 unsigned long multiscale::video::CartesianToConcentrationsConverter-::width [private]

Width of the grid

Definition at line 22 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`

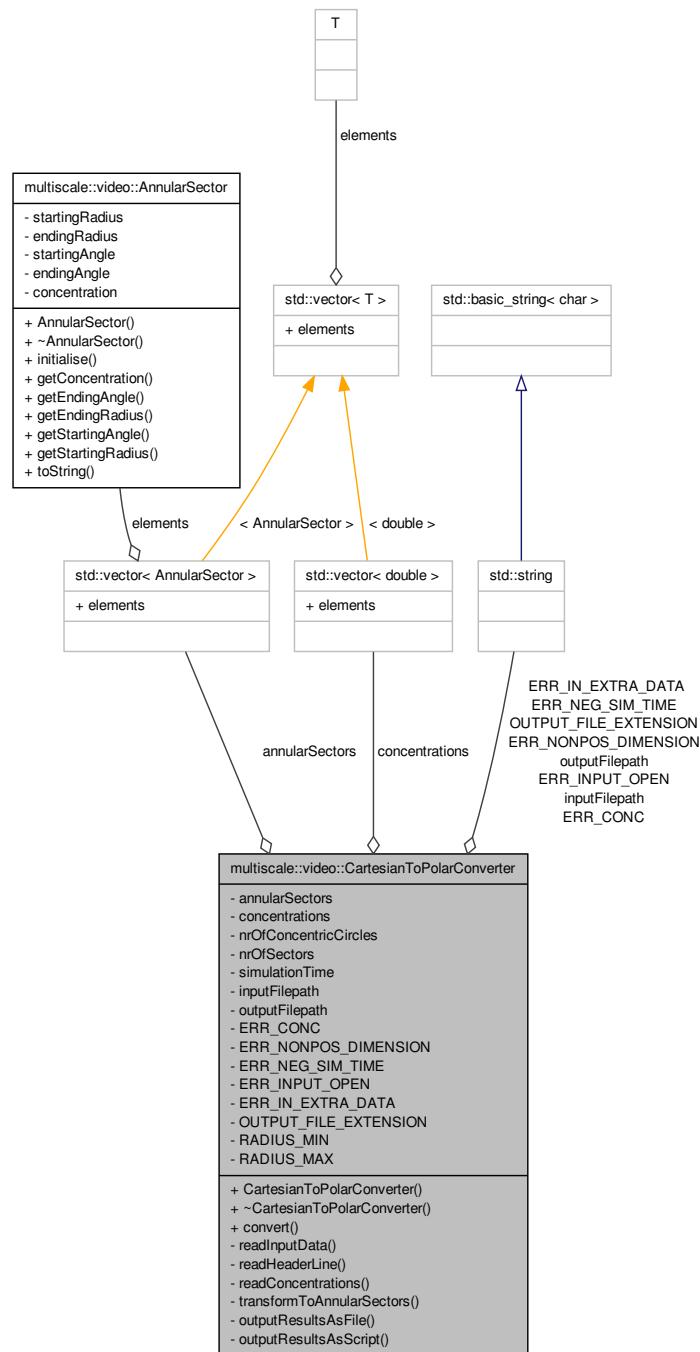
- `/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/CartesianToConcentrationsConverter.cpp`

7.26 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 string &`filepath`, const string &`outputFilepath`)
- `~CartesianToPolarConverter` ()
- void `convert` (bool `outputToScript`)

Start the conversion.

Private Member Functions

- void `readInputData` ()
Read the input data.
- void `readHeaderLine` (ifstream &`fin`)
Read the header line.
- void `readConcentrations` (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

- vector< `AnnularSector` > `annularSectors`
- vector< double > `concentrations`
- unsigned long `nrOfConcentricCircles`
- unsigned long `nrOfSectors`
- double `simulationTime`
- string `filepath`
- string `outputfilepath`

Static Private Attributes

- static const string `ERR_CONC` = "All `concentrations` have to be between 0 and 1."
- static const string `ERR_NONPOS_DIMENSION` = "The dimensions N and M must be positive."
- static const string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const string `ERR_INPUT_OPEN` = "The input file could not be opened"
- static const string `ERR_IN_EXTRA_DATA` = "The input file contains more data than required."

- static const string `OUTPUT_FILE_EXTENSION` = ".out"
- static const double `RADIUS_MIN` = 0.001
- static const double `RADIUS_MAX` = 0.3

7.26.1 Detailed Description

Converter from the rectangular geometry grid cells to annular sectors.

Definition at line 17 of file `CartesianToPolarConverter.hpp`.

7.26.2 Constructor & Destructor Documentation

7.26.2.1 `CartesianToPolarConverter::CartesianToPolarConverter (const string & inputfilepath, const string & outputfilepath)`

Definition at line 16 of file `CartesianToPolarConverter.cpp`.

References `nrOfConcentricCircles`, `nrOfSectors`, and `simulationTime`.

7.26.2.2 `CartesianToPolarConverter::~CartesianToPolarConverter ()`

Definition at line 25 of file `CartesianToPolarConverter.cpp`.

7.26.3 Member Function Documentation

7.26.3.1 `void CartesianToPolarConverter::convert (bool outputToScript)`

Start the conversion.

Parameters

<code>outputToScript</code>	Output to script or to plain file
-----------------------------	-----------------------------------

Definition at line 27 of file `CartesianToPolarConverter.cpp`.

References `outputResultsAsFile()`, `outputResultsAsScript()`, `readInputData()`, and `transformToAnnularSectors()`.

Referenced by `main()`.

7.26.3.2 `void CartesianToPolarConverter::outputResultsAsFile () [private]`

Output the results as a plain file.

Definition at line 116 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `OUTPUT_FILE_EXTENSION`, and `outputfilepath`.

Referenced by convert().

7.26.3.3 void `CartesianToPolarConverter::outputResultsAsScript()`
[private]

Output the results as a gnuplot script.

Definition at line 131 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `multiscale::video::PolarGnuplotScriptGenerator::generateScript()`, `outputFilepath`, and `simulationTime`.

Referenced by convert().

7.26.3.4 void `CartesianToPolarConverter::readConcentrations(ifstream & fin)`
[private]

Read the concentrations.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 71 of file `CartesianToPolarConverter.cpp`.

References `concentrations`, `ERR_CONC`, `MS_throw`, `nrOfConcentricCircles`, and `nrOfSectors`.

Referenced by `readInputData()`.

7.26.3.5 void `CartesianToPolarConverter::readHeaderLine(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 62 of file `CartesianToPolarConverter.cpp`.

References `ERR_NEG_SIM_TIME`, `ERR_NONPOS_DIMENSION`, `MS_throw`, `nrOfConcentricCircles`, `nrOfSectors`, and `simulationTime`.

Referenced by `readInputData()`.

7.26.3.6 void CartesianToPolarConverter::readInputData() [private]

Read the input data.

Definition at line 38 of file `CartesianToPolarConverter.cpp`.

References `ERR_IN_EXTRA_DATA`, `ERR_INPUT_OPEN`, `inputFilepath`, `MS_throw`, `readConcentrations()`, and `readHeaderLine()`.

Referenced by `convert()`.

7.26.3.7 void CartesianToPolarConverter::transformToAnnularSectors() [private]

Convert the concentrations to annular sectors.

Definition at line 91 of file `CartesianToPolarConverter.cpp`.

References `annularSectors`, `concentrations`, `nrOfConcentricCircles`, `nrOfSectors`, `RADIUS_MAX`, and `RADIUS_MIN`.

Referenced by `convert()`.

7.26.4 Member Data Documentation

7.26.4.1 vector<AnnularSector> multiscale::video::CartesianToPolarConverter::annularSectors [private]

Resulting annular sectors

Definition at line 21 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`, `outputResultsAsScript()`, and `transformToAnnularSectors()`.

7.26.4.2 vector<double> multiscale::video::CartesianToPolarConverter::concentrations [private]

Concentrations received as input

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

Referenced by `readConcentrations()`, and `transformToAnnularSectors()`.

7.26.4.3 const string CartesianToPolarConverter::ERR_CONC = "All concentrations have to be between 0 and 1." [static, private]

Definition at line 74 of file `CartesianToPolarConverter.hpp`.

Referenced by `readConcentrations()`.

7.26.4.4 `const string CartesianToPolarConverter::ERR_IN_EXTRA_DATA = "The input file contains more data than required." [static, private]`

Definition at line 78 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.26.4.5 `const string CartesianToPolarConverter::ERR_INPUT_OPEN = "The input file could not be opened" [static, private]`

Definition at line 77 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.26.4.6 `const string CartesianToPolarConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative." [static, private]`

Definition at line 76 of file `CartesianToPolarConverter.hpp`.

Referenced by `readHeaderLine()`.

7.26.4.7 `const string CartesianToPolarConverter::ERR_NONPOS_DIMENSION = "The dimensions N and M must be positive." [static, private]`

Definition at line 75 of file `CartesianToPolarConverter.hpp`.

Referenced by `readHeaderLine()`.

7.26.4.8 `string multiscale::video::CartesianToPolarConverter::inputFilepath [private]`

Path to the input file

Definition at line 28 of file `CartesianToPolarConverter.hpp`.

Referenced by `readInputData()`.

7.26.4.9 `unsigned long multiscale::video::CartesianToPolarConverter::nrOfConcentricCircles [private]`

Number of concentric circles

Definition at line 24 of file `CartesianToPolarConverter.hpp`.

Referenced by `CartesianToPolarConverter()`, `readConcentrations()`, `readHeaderLine()`, and `transformToAnnularSectors()`.

7.26.4.10 **unsigned long multiscale::video::CartesianToPolarConverter::nrOfSectors**
[private]

Number of sectors

Definition at line 25 of file `CartesianToPolarConverter.hpp`.

Referenced by `CartesianToPolarConverter()`, `readConcentrations()`, `readHeaderLine()`, and `transformToAnnularSectors()`.

7.26.4.11 **const string CartesianToPolarConverter::OUTPUT_FILE_EXTENSION = ".out"** [static, private]

Definition at line 80 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`.

7.26.4.12 **string multiscale::video::CartesianToPolarConverter::outputFilepath**
[private]

Path to the output file

Definition at line 29 of file `CartesianToPolarConverter.hpp`.

Referenced by `outputResultsAsFile()`, and `outputResultsAsScript()`.

7.26.4.13 **const double CartesianToPolarConverter::RADIUS_MAX = 0.3** [static, private]

Definition at line 83 of file `CartesianToPolarConverter.hpp`.

Referenced by `transformToAnnularSectors()`.

7.26.4.14 **const double CartesianToPolarConverter::RADIUS_MIN = 0.001** [static, private]

Definition at line 82 of file `CartesianToPolarConverter.hpp`.

Referenced by `transformToAnnularSectors()`.

7.26.4.15 **double multiscale::video::CartesianToPolarConverter::simulationTime**
[private]

Simulation time corresponding to the input data

Definition at line 26 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

7.27 multiscale::analysis::CircularityMeasure Class Reference

Class for computing the circularity measure for the given collection of points.

```
#include <CircularityMeasure.hpp>
```

Static Public Member Functions

- static double **compute** (const vector< Point2f > &points)
Compute circularity measure for the given collection of points.
- static double **compute** (const vector< Point > &points)
Compute circularity measure for the given collection of points.

7.27.1 Detailed Description

Class for computing the circularity measure for the given collection of points.

Definition at line 18 of file CircularityMeasure.hpp.

7.27.2 Member Function Documentation

7.27.2.1 double CircularityMeasure::compute (const vector< 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.

7.27.2.2 double CircularityMeasure::compute (const vector< 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](#)

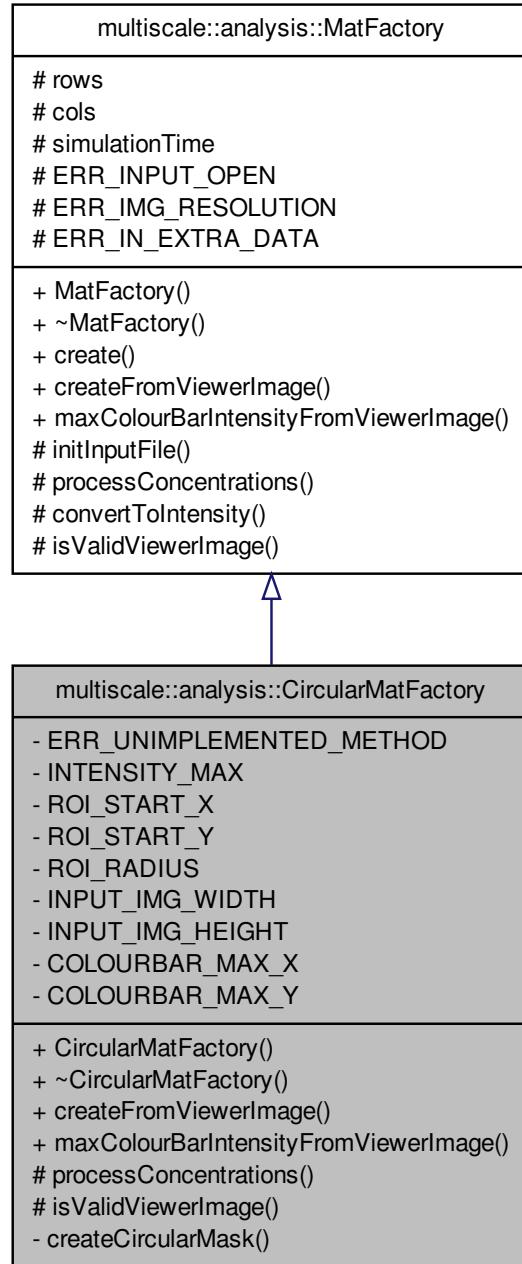
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-
[CircularityMeasure.cpp](#)

7.28 multiscale::analysis::CircularMatFactory Class Reference

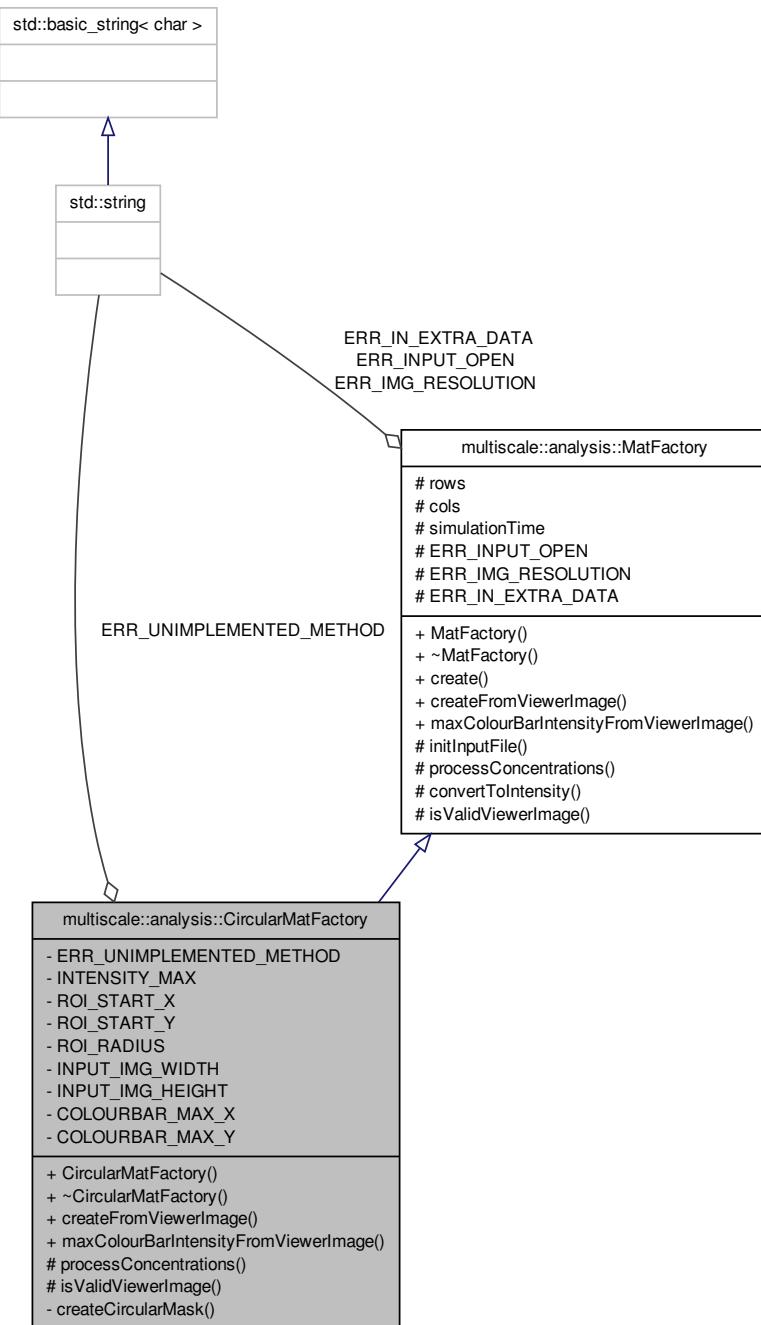
Class for creating a Mat object considering a circular grid.

```
#include <CircularMatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::CircularMatFactory:



Collaboration diagram for multiscale::analysis::CircularMatFactory:



Public Member Functions

- `CircularMatFactory ()`
- `~CircularMatFactory ()`
- Mat `createFromViewerImage (const string &inputFile) override`
Create a Mat object from the image file obtained from the CircularGeometryViewer.
- double `maxColourBarIntensityFromViewerImage (const string &inputFile) override`
Get the maximum grayscale intensity of the colour bar in the image.

Protected Member Functions

- unsigned char * `processConcentrations (ifstream &fin) override`
Process the concentrations from the input file.
- bool `isValidViewerImage (const Mat &image) override`
Check if the image generated by the viewer has the required resolution.

Private Member Functions

- Mat `createCircularMask (unsigned int originX, unsigned int originY, unsigned int radius, const Mat &image)`
Create a mask with 255 intensity pixels inside the circle with origin at (originX, originY) and the given radius.

Static Private Attributes

- static const 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

7.28.1 Detailed Description

Class for creating a Mat object considering a circular grid.

Definition at line 15 of file CircularMatFactory.hpp.

7.28.2 Constructor & Destructor Documentation

7.28.2.1 CircularMatFactory::CircularMatFactory()

Definition at line 10 of file CircularMatFactory.cpp.

7.28.2.2 CircularMatFactory::~CircularMatFactory()

Definition at line 12 of file CircularMatFactory.cpp.

7.28.3 Member Function Documentation

7.28.3.1 Mat CircularMatFactory::createCircularMask(unsigned int *originX*, unsigned int *originY*, unsigned int *radius*, const Mat & *image*) [private]

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 getting the 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().

7.28.3.2 Mat CircularMatFactory::createFromViewerImage(const 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().

7.28.3.3 bool CircularMatFactory::isValidViewerImage (const 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 57 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().

7.28.3.4 double CircularMatFactory::maxColourBarIntensityFromViewerImage (const 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().

7.28.3.5 unsigned char * CircularMatFactory::processConcentrations (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`.

7.28.4 Member Data Documentation

7.28.4.1 const int CircularMatFactory::COLOURBAR_MAX_X = 1775 [static, private]

Definition at line 82 of file CircularMatFactory.hpp.

Referenced by `maxColourBarIntensityFromViewerImage()`.

7.28.4.2 const int CircularMatFactory::COLOURBAR_MAX_Y = 56 [static, private]

Definition at line 83 of file CircularMatFactory.hpp.

Referenced by `maxColourBarIntensityFromViewerImage()`.

7.28.4.3 const string CircularMatFactory::ERR_UNIMPLEMENTED_METHOD = "The method you called is not implemented." [static, private]

Definition at line 71 of file CircularMatFactory.hpp.

Referenced by `processConcentrations()`.

7.28.4.4 const int CircularMatFactory::INPUT_IMG_HEIGHT = 1572 [static, private]

Definition at line 80 of file CircularMatFactory.hpp.

Referenced by `isValidViewerImage()`.

7.28.4.5 const int CircularMatFactory::INPUT_IMG_WIDTH = 2048 [static, private]

Definition at line 79 of file CircularMatFactory.hpp.

Referenced by `isValidViewerImage()`.

7.28.4.6 const int CircularMatFactory::INTENSITY_MAX = 255 [static, private]

Definition at line 73 of file CircularMatFactory.hpp.

Referenced by `createCircularMask()`.

7.28.4.7 **const int CircularMatFactory::ROI_RADIUS = 615** [static, private]

Definition at line 77 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.28.4.8 **const int CircularMatFactory::ROI_START_X = 1024** [static, private]

Definition at line 75 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

7.28.4.9 **const int CircularMatFactory::ROI_START_Y = 786** [static, private]

Definition at line 76 of file CircularMatFactory.hpp.

Referenced by `createFromViewerImage()`.

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

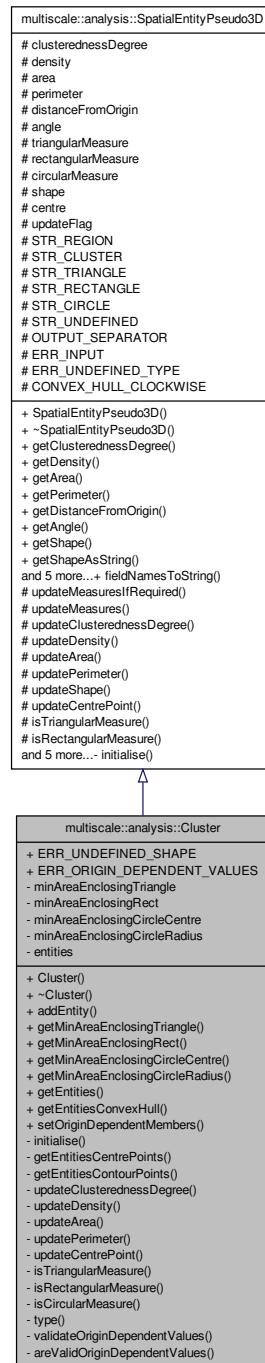
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spat/CircularMatFactory.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-CircularMatFactory.cpp](#)

7.29 multiscale::analysis::Cluster Class Reference

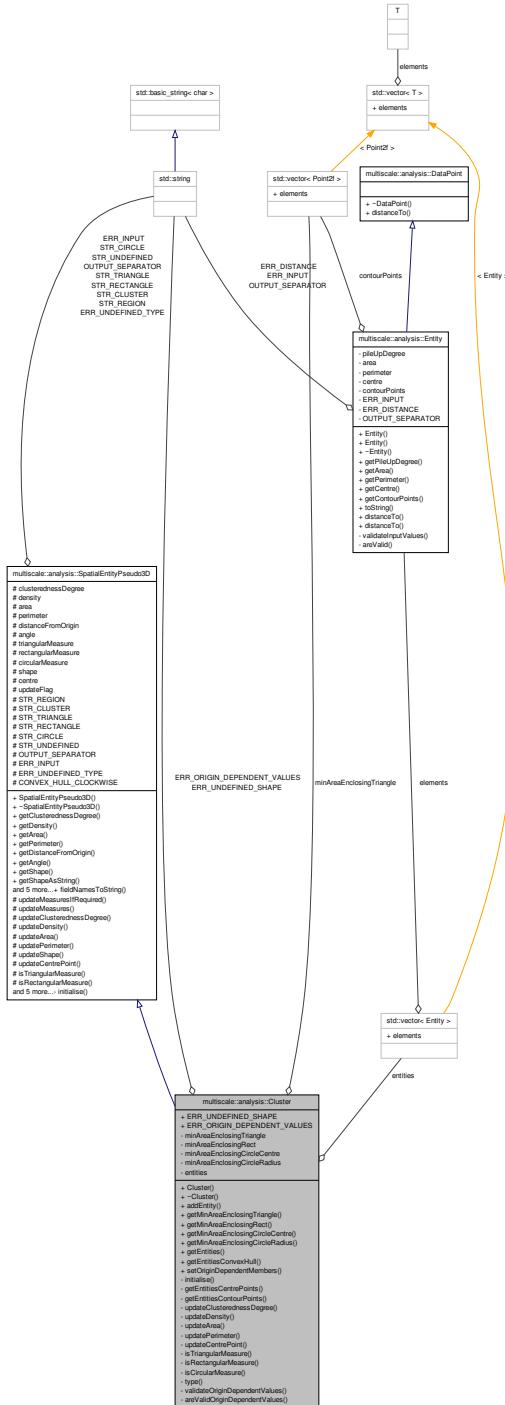
Class for representing a cluster of entities in an image.

```
#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.
- `vector< Point2f > getMinAreaEnclosingTriangle ()`
Get the minimum area enclosing triangle.
- `RotatedRect getMinAreaEnclosingRect ()`
Get the minimum area enclosing rectangle.
- `Point2f getMinAreaEnclosingCircleCentre ()`
Get the minimum area enclosing circle centre.
- `float getMinAreaEnclosingCircleRadius ()`
Get the minimum area enclosing circle radius.
- `vector< Entity > getEntities () const`
Get the collection of underlying entities.
- `vector< 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 string ERR_UNDEFINED_SHAPE = "The shape of the given cluster is undefined."`
- `static const string ERR_ORIGIN_DEPENDENT_VALUES = "The origin dependent values are invalid (i.e. negative)."`

Private Member Functions

- `void initialise ()`
Initialisation function for the class.
- `vector< Point2f > getEntitiesCentrePoints ()`
Get the collection of entities' centres.
- `vector< 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`

- void `updateCentrePoint ()` override
 - Update the value of 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.*
- `SpatialEntityPseudo3DType type ()` override
 - Return the type of the pseudo 3D spatial entity.*
- void `validateOriginDependentValues (double distanceFromOrigin, double angle-WrtOrigin)`
 - Validate the origin dependent values (i.e. non-negative)*
- bool `isValidOriginDependentValues (double distanceFromOrigin, double angle-WrtOrigin)`
 - Check if the origin dependent values are valid (i.e. non-negative)*

Private Attributes

- `vector< Point2f > minAreaEnclosingTriangle`
- `RotatedRect minAreaEnclosingRect`
- `Point2f minAreaEnclosingCircleCentre`
- `float minAreaEnclosingCircleRadius`
- `vector< Entity > entities`

7.29.1 Detailed Description

Class for representing a cluster of entities in an image.

Definition at line 21 of file Cluster.hpp.

7.29.2 Constructor & Destructor Documentation

7.29.2.1 Cluster::Cluster()

Definition at line 11 of file Cluster.cpp.

References `initialise()`.

Referenced by `type()`.

7.29.2.2 Cluster::~Cluster()

Definition at line 15 of file Cluster.cpp.

7.29.3 Member Function Documentation

7.29.3.1 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.

7.29.3.2 bool Cluster::isValidOriginDependentValues (double *distanceFromOrigin*, double *angleWrtOrigin*) [private]

Check if the origin dependent values are valid (i.e. non-negative)

Parameters

<i>distance-FromOrigin</i>	Distance from the origin
<i>angleWrt-Origin</i>	Angle with respect to the origin

Definition at line 199 of file Cluster.cpp.

References multiscale::Numeric::greaterOrEqual().

Referenced by validateOriginDependentValues().

7.29.3.3 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().

7.29.3.4 vector< Point2f > Cluster::getEntitiesCentrePoints () [private]

Get the collection of entities' centres.

Definition at line 84 of file Cluster.cpp.

References entities.

7.29.3.5 vector< 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()`.

7.29.3.6 `vector< 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 `getEntitiesContourPoints()`.

Referenced by `multiscale::analysis::ClusterDetector::getClusterConvexHull()`, `isTriangularMeasure()`, `updateCentrePoint()`, and `updatePerimeter()`.

7.29.3.7 `Point2f Cluster::getMinAreaEnclosingCircleCentre()`

Get the minimum area enclosing circle centre.

Definition at line 35 of file Cluster.cpp.

References `minAreaEnclosingCircleCentre`, and `multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIfRequired()`.

Referenced by `multiscale::analysis::SimulationClusterDetector::outputClusterCircularShape()`.

7.29.3.8 `float Cluster::getMinAreaEnclosingCircleRadius()`

Get the minimum area enclosing circle radius.

Definition at line 41 of file Cluster.cpp.

References `minAreaEnclosingCircleRadius`, and `multiscale::analysis::SpatialEntityPseudo3D::updateMeasuresIfRequired()`.

Referenced by `multiscale::analysis::SimulationClusterDetector::outputClusterCircularShape()`.

7.29.3.9 `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()`.

7.29.3.10 vector< 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().

7.29.3.11 void Cluster::initialise() [private]

Initialisation function for the class.

Reimplemented from [multiscale::analysis::SpatialEntityPseudo3D](#).

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().

7.29.3.12 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.

7.29.3.13 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().

7.29.3.14 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().

7.29.3.15 void Cluster::setOriginDependentMembers(double distanceFromOrigin, double angleWrtOrigin)

Set the values of the origin dependent members.

Parameters

<i>distance-FromOrigin</i>	Distance from the origin
<i>angleWrt-Origin</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().

7.29.3.16 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 Cluster().

7.29.3.17 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.

```
7.29.3.18 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\(\)](#).

```
7.29.3.19 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](#).

```
7.29.3.20 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](#).

```
7.29.3.21 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](#).

```
7.29.3.22 void Cluster::validateOriginDependentValues( double distanceFromOrigin,  
double angleWrtOrigin ) [private]
```

Validate the origin dependent values (i.e. non-negative)

Parameters

<i>distance-FromOrigin</i>	Distance from the origin
<i>angleWrt-Origin</i>	Angle with respect to the origin

Definition at line 193 of file Cluster.cpp.

References areValidOriginDependentValues(), ERR_ORIGIN_DEPENDENT_VALUES, and MS_throw.

Referenced by setOriginDependentMembers().

7.29.4 Member Data Documentation

7.29.4.1 `vector<Entity> multiscale::analysis::Cluster::entities` [private]

Entities which belong to this cluster

Definition at line 32 of file Cluster.hpp.

Referenced by addEntity(), getEntities(), getEntitiesCentrePoints(), getEntitiesContourPoints(), getEntitiesConvexHull(), initialise(), updateArea(), updateClusterednessDegree(), and updateDensity().

7.29.4.2 `const string Cluster::ERR_ORIGIN_DEPENDENT_VALUES = "The origin dependent values are invalid (i.e. negative)." [static]`

Definition at line 123 of file Cluster.hpp.

Referenced by validateOriginDependentValues().

7.29.4.3 `const string Cluster::ERR_UNDEFINED_SHAPE = "The shape of the given cluster is undefined." [static]`

Definition at line 122 of file Cluster.hpp.

Referenced by multiscale::analysis::SimulationClusterDetector::outputClusterShape().

7.29.4.4 `Point2f multiscale::analysis::Cluster::minAreaEnclosingCircleCentre [private]`

The minimum area enclosing circle centre point

Definition at line 29 of file Cluster.hpp.

Referenced by getMinAreaEnclosingCircleCentre(), and isCircularMeasure().

**7.29.4.5 float multiscale::analysis::Cluster::minAreaEnclosingCircleRadius
[private]**

The minimum area enclosing circle radius

Definition at line 30 of file Cluster.hpp.

Referenced by getMinAreaEnclosingCircleRadius(), initialise(), and isCircularMeasure().

**7.29.4.6 RotatedRect multiscale::analysis::Cluster::minAreaEnclosingRect
[private]**

The minimum area enclosing rectangle

Definition at line 27 of file Cluster.hpp.

Referenced by getMinAreaEnclosingRect(), and isRectangularMeasure().

**7.29.4.7 vector<Point2f> multiscale::analysis::Cluster::minAreaEnclosingTriangle
[private]**

The minimum area enclosing triangle

Definition at line 25 of file Cluster.hpp.

Referenced by getMinAreaEnclosingTriangle(), initialise(), and isTriangularMeasure().

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

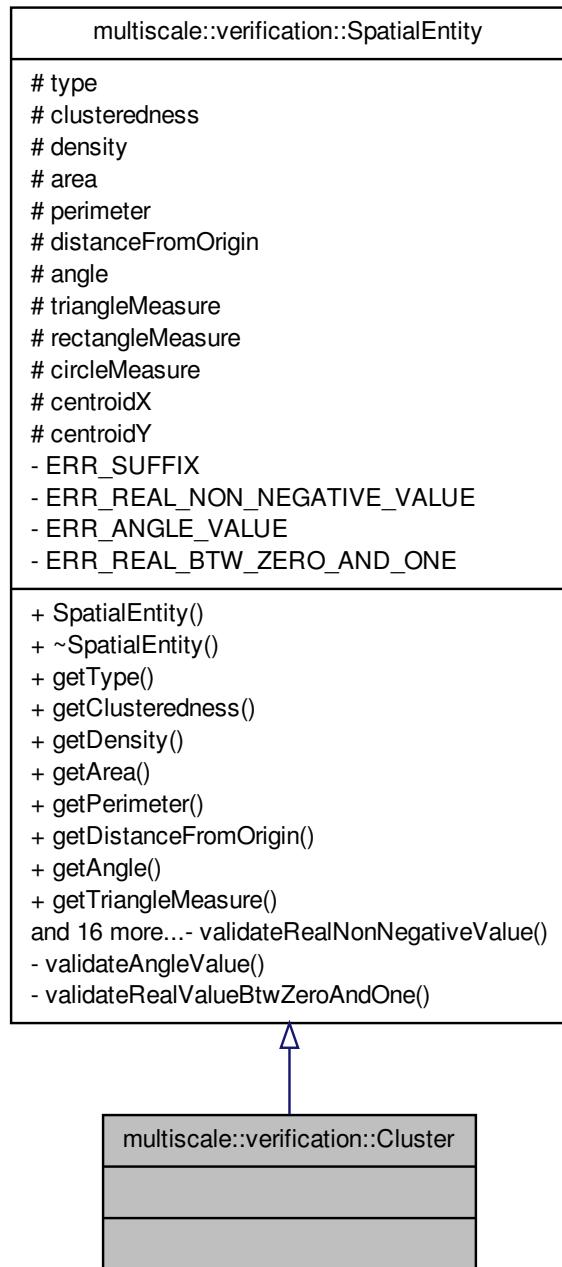
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/-Cluster.hpp](#)
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Cluster.cpp](#)

7.30 multiscale::verification::Cluster Class Reference

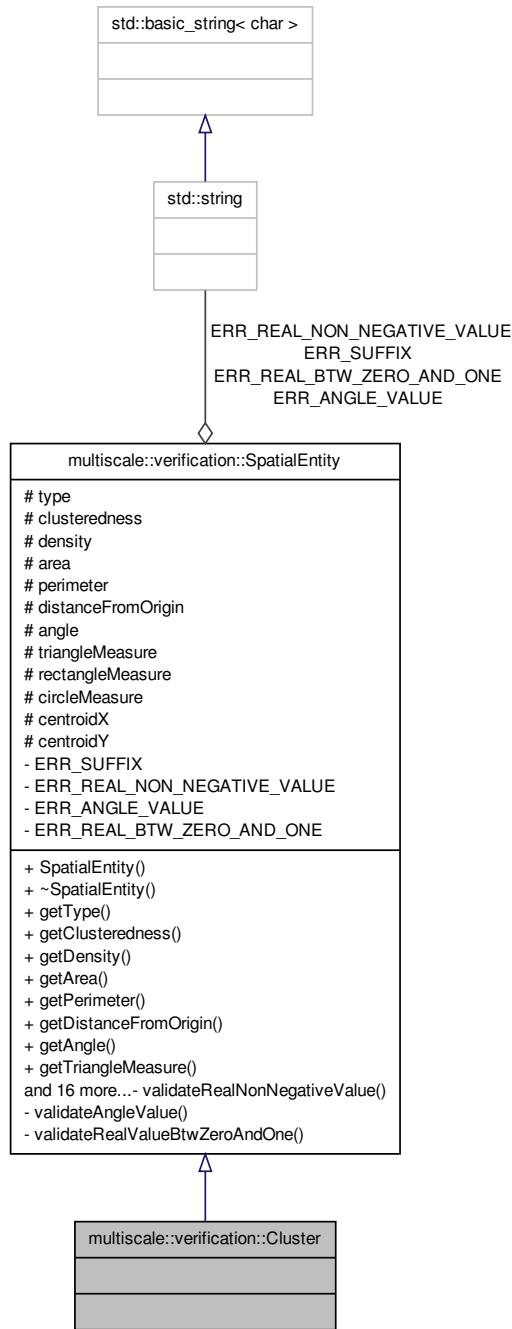
Class for representing a cluster.

```
#include <Cluster.hpp>
```

Inheritance diagram for multiscale::verification::Cluster:



Collaboration diagram for multiscale::verification::Cluster:



7.30.1 Detailed Description

Class for representing a cluster.

Definition at line 12 of file Cluster.hpp.

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

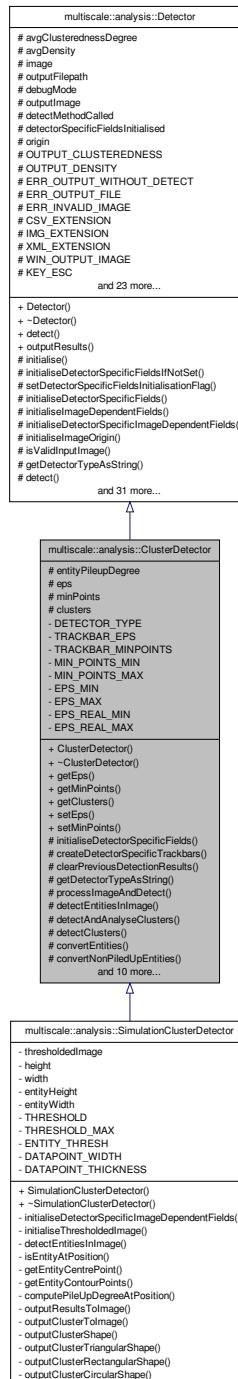
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[Cluster.hpp](#)

7.31 multiscale::analysis::ClusterDetector Class Reference

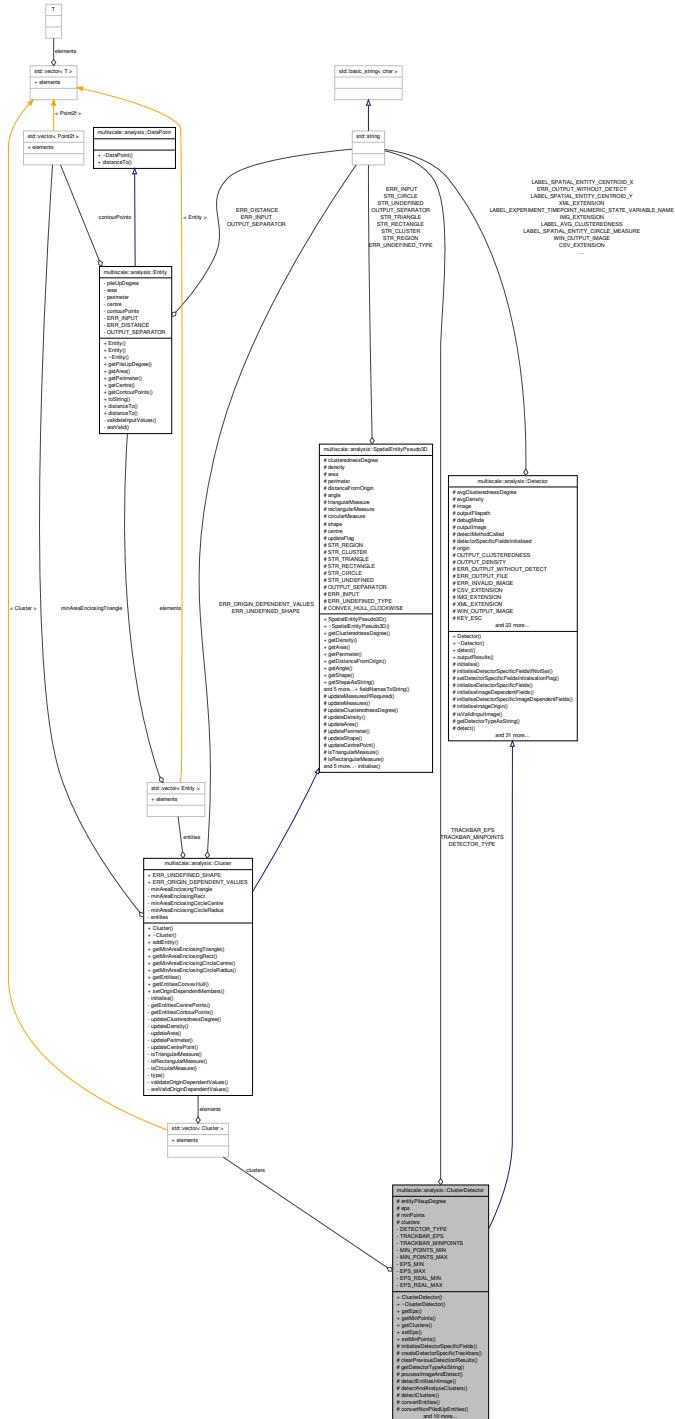
Class for detecting clusters in 2D images.

```
#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 eps.`
- int `getMinPoints ()`
`Get the value of the clustering algorithm parameter MinPoints.`
- vector< `Cluster` > const & `getClusters ()`
`Get a const reference to the vector of detected clusters.`
- 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.`
- string `getDetectorTypeAsString () override`
`Get the type of the detector as a string.`
- void `processImageAndDetect () override`
`Process the provided image and detect clusters in it.`
- virtual void `detectEntitiesInImage (vector< Entity > &entities)=0`
`Detect the entities in the image.`
- void `detectAndAnalyseClusters (const vector< Entity > &entities, vector< - Cluster > &clusters)`
`Detect and analyse the clusters of entities in the image.`
- void `detectClusters (const vector< Entity > &entities, vector< int > &clusterIndexes, int &nrOfClusters)`
`Detect the clusters of entities in the image.`
- vector< shared_ptr< `DataPoint` > > `convertEntities` (const vector< `Entity` > &entities)
`Convert the entities to the format required by the DBSCAN class.`
- void `convertNonPiledUpEntities (const vector< Entity > &entities, vector< shared_ptr< DataPoint > > &dataPoints)`
`Convert the non pile up entities to the format required by the DBSCAN class.`
- void `convertPiledUpEntities (const vector< Entity > &entities, vector< shared_ptr< DataPoint > > &dataPoints)`

- Convert the entities to the required format by the `DBSCAN` class.
- void `addEntitiesToClusters` (const vector< `Entity` > &entities, const vector< int > &clusterIndexes, int nrOfClusters, vector< `Cluster` > &clusters)
 - Add the entities to the clusters as indicated by the clusterIndexes parameter.*
- void `analyseClusters` (vector< `Cluster` > &clusters)
 - Analyse the clusters.*
- void `analyseClustersOriginDependentValues` (vector< `Cluster` > &clusters)
 - Analyse the clusters and compute the origin dependent values.*
- void `updateClusterOriginDependentValues` (`Cluster` &cluster, const vector< `Point` > &clusterConvexHull)
 - Update the cluster and compute the origin dependent values considering the convex hull.*
- vector< `Point` > `getClusterConvexHull` (`Cluster` &cluster)
 - Return the convex hull of the given cluster.*
- double `computeClusterednessIndex` (const vector< `Cluster` > &clusters)
 - Compute the clusteredness index for all the entities detected in the image.*
- double `computeAveragePileUpDegree` (vector< `Cluster` > &clusters)
 - Compute the average pile up degree for all entities in the image.*
- vector< 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`
- vector< `Cluster` > `clusters`

Static Private Attributes

- static const string `DETECTOR_TYPE` = "Clusters"
- static const string `TRACKBAR_EPS` = "Eps (Multiplied by 10)"
- static const 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

7.31.1 Detailed Description

Class for detecting clusters in 2D images.

Definition at line 20 of file ClusterDetector.hpp.

7.31.2 Constructor & Destructor Documentation

7.31.2.1 ClusterDetector::ClusterDetector (int *maxPileupNumber*, double *maxPileupIntensity*, bool *debugMode* = false)

Parameters

<i>debugMode</i>	Flag indicating if detector should run in debug mode or not
<i>maxPileup- Number</i>	The maximum number of entities which can occupy a grid position at the same time
<i>maxPileup- Intensity</i>	The grayscale intensity of a maximally piled up grid position

Definition at line 15 of file ClusterDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, entityPileupDegree, eps, and minPoints.

7.31.2.2 ClusterDetector::~ClusterDetector () [virtual]

Definition at line 25 of file ClusterDetector.cpp.

7.31.3 Member Function Documentation

7.31.3.1 void ClusterDetector::addEntitiesToClusters (const vector< Entity > & *entities*, const vector< int > & *clusterIndexes*, int *nrOfClusters*, vector< Cluster > & *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>cluster- Indexes</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 115 of file ClusterDetector.cpp.

Referenced by detectAndAnalyseClusters().

7.31.3.2 void ClusterDetector::analyseClusters (vector< Cluster > & *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 132 of file ClusterDetector.cpp.

References analyseClustersOriginDependentValues(), multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, computeAveragePileUpDegree(), and computeClusterednessIndex().

Referenced by detectAndAnalyseClusters().

7.31.3.3 void ClusterDetector::analyseClustersOriginDependentValues (vector< Cluster > & *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 139 of file ClusterDetector.cpp.

References getClusterConvexHull(), and updateClusterOriginDependentValues().

Referenced by analyseClusters().

7.31.3.4 void ClusterDetector::clearPreviousDetectionResults () [override, protected, virtual]

Clear the clusters from the previous detection.

Implements [multiscale::analysis::Detector](#).

Definition at line 61 of file ClusterDetector.cpp.

References clusters.

7.31.3.5 double ClusterDetector::computeAveragePileUpDegree (`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

<code>clusters</code>	Clusters of entities detected in the image
-----------------------	--

Definition at line 172 of file ClusterDetector.cpp.

Referenced by analyseClusters().

7.31.3.6 double ClusterDetector::computeClusterednessIndex (`const 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

<code>clusters</code>	Collection of clusters, each one with the updated measures
-----------------------	--

Definition at line 168 of file ClusterDetector.cpp.

References multiscale::analysis::Silhouette::computeOverallAverageMeasure().

Referenced by analyseClusters().

7.31.3.7 `vector< shared_ptr< DataPoint > > ClusterDetector::convertEntities (const vector< Entity > & entities)` [protected]

Convert the entities to the format required by the [DBSCAN](#) class.

Parameters

<code>entities</code>	Entities detected in the image
-----------------------	--------------------------------

Definition at line 89 of file ClusterDetector.cpp.

References convertNonPiledUpEntities(), and convertPiledUpEntities().

Referenced by detectClusters().

7.31.3.8 double ClusterDetector::convertEpsValue () [protected]

Convert the value of eps from integer to double.

Definition at line 195 of file ClusterDetector.cpp.

References eps, EPS_MAX, EPS_MIN, EPS_REAL_MAX, and EPS_REAL_MIN.

Referenced by detectClusters(), and getEps().

**7.31.3.9 void ClusterDetector::convertNonPiledUpEntities (const vector< Entity >
& entities, vector< 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 DataPoint instances required by the DBSCAN class

Definition at line 98 of file ClusterDetector.cpp.

Referenced by convertEntities().

**7.31.3.10 void ClusterDetector::convertPiledUpEntities (const vector< Entity >
& entities, vector< 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 DataPoint instances required by the DBSCAN class

Definition at line 104 of file ClusterDetector.cpp.

Referenced by convertEntities().

**7.31.3.11 void ClusterDetector::createDetectorSpecificTrackbars ()
[override, protected, virtual]**

Create the trackbars.

Implements [multiscale::analysis::Detector](#).

Definition at line 56 of file ClusterDetector.cpp.

References eps, EPS_MAX, MIN_POINTS_MAX, minPoints, TRACKBAR_EPS, TRACKBAR_MINPOINTS, and [multiscale::analysis::Detector::WIN_OUTPUT_IMAGE](#).

**7.31.3.12 void ClusterDetector::detectAndAnalyseClusters (const vector< Entity >
& entities, 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 76 of file ClusterDetector.cpp.

References addEntitiesToClusters(), analyseClusters(), multiscale::analysis::DBSCAN::CLUSTERING_UNCLASSIFIED, and detectClusters().

Referenced by processImageAndDetect().

7.31.3.13 void ClusterDetector::detectClusters (const vector< Entity > & *entities*, 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>cluster-Indexes</i>	Indexes to which cluster each entity belongs
<i>nrOfClusters</i>	Total number of clusters

Definition at line 85 of file ClusterDetector.cpp.

References convertEntities(), convertEpsValue(), getValidMinPointsValue(), and multiscale::analysis::DBSCAN::run().

Referenced by detectAndAnalyseClusters().

7.31.3.14 virtual void multiscale::analysis::ClusterDetector::detectEntitiesInImage (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().

7.31.3.15 `vector< Point > ClusterDetector::getClusterConvexHull (Cluster & cluster)` [protected]

Return the convex hull of the given cluster.

Parameters

<code>cluster</code>	The given cluster
----------------------	-------------------

Definition at line 158 of file ClusterDetector.cpp.

References multiscale::analysis::Cluster::getEntitiesConvexHull().

Referenced by analyseClustersOriginDependentValues().

7.31.3.16 `vector< Cluster > const & ClusterDetector::getClusters ()`

Get a const reference to the vector of detected clusters.

Definition at line 35 of file ClusterDetector.cpp.

References clusters.

7.31.3.17 `vector< shared_ptr< SpatialEntityPseudo3D > > ClusterDetector::getCollectionOfSpatialEntityPseudo3D ()` [override, protected, virtual]

Get the collection of clusters detected in the image.

Implements [multiscale::analysis::Detector](#).

Definition at line 185 of file ClusterDetector.cpp.

References multiscale::analysis::Cluster, and clusters.

7.31.3.18 `string ClusterDetector::getDetectorTypeAsString ()` [override, protected, virtual]

Get the type of the detector as a string.

Implements [multiscale::analysis::Detector](#).

Definition at line 65 of file ClusterDetector.cpp.

References DETECTOR_TYPE.

7.31.3.19 `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().

7.31.3.20 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().

7.31.3.21 int ClusterDetector::getValidMinPointsValue() [protected]

Return non-zero value for minPoints.

Definition at line 199 of file ClusterDetector.cpp.

References minPoints.

Referenced by detectClusters().

7.31.3.22 void ClusterDetector::initialiseDetectorSpecificFields() [override, protected, virtual]

Initialise clustering values.

Implements [multiscale::analysis::Detector](#).

Definition at line 51 of file ClusterDetector.cpp.

References eps, and minPoints.

7.31.3.23 void ClusterDetector::processImageAndDetect() [override, protected, virtual]

Process the provided image and detect clusters in it.

Implements [multiscale::analysis::Detector](#).

Definition at line 69 of file ClusterDetector.cpp.

References clusters, detectAndAnalyseClusters(), and detectEntitiesInImage().

7.31.3.24 void ClusterDetector::setEps(double eps)

Set the value of the clustering algorithm parameter eps.

Parameters

<code>eps</code>	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()`.

7.31.3.25 void ClusterDetector::setMinPoints (int *minPoints*)

Set the value of the clustering algorithm parameter MinPoints.

Parameters

<code>minPoints</code>	Value of the clustering algorithm parameter MinPoints
------------------------	---

Definition at line 45 of file ClusterDetector.cpp.

References `minPoints`, and `multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag()`.

Referenced by `loadDetectorParameterValues()`.

7.31.3.26 void ClusterDetector::updateClusterOriginDependentValues (Cluster & *cluster*, const vector< 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

<code>cluster</code>	Cluster
<code>cluster-ConvexHull</code>	Convex hull of the cluster

Definition at line 149 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()`.

7.31.4 Member Data Documentation

7.31.4.1 `vector<Cluster> multiscale::analysis::ClusterDetector::clusters` [protected]

Clusters found in the image

Definition at line 32 of file ClusterDetector.hpp.

Referenced by clearPreviousDetectionResults(), getClusters(), getCollectionOfSpatialEntityPseudo3D(), multiscale::analysis::SimulationClusterDetector::outputResultsToImage(), and processImageAndDetect().

7.31.4.2 `const string ClusterDetector::DETECTOR_TYPE = "Clusters"` [static, private]

Definition at line 201 of file ClusterDetector.hpp.

Referenced by getDetectorTypeAsString().

7.31.4.3 `double multiscale::analysis::ClusterDetector::entityPileupDegree` [protected]

The pile up degree (intensity) of a grid position occupied by only one entity

Definition at line 24 of file ClusterDetector.hpp.

Referenced by ClusterDetector(), and multiscale::analysis::SimulationClusterDetector::computePileUpDegreeAtPosition().

7.31.4.4 `int multiscale::analysis::ClusterDetector::eps` [protected]

DBSCAN algorithm parameter for specifying the maximum radius of the neighbourhood

Definition at line 27 of file ClusterDetector.hpp.

Referenced by ClusterDetector(), convertEpsValue(), createDetectorSpecificTrackbars(), initialiseDetectorSpecificFields(), and setEps().

7.31.4.5 `const int ClusterDetector::EPS_MAX = 10000` [static, private]

Definition at line 210 of file ClusterDetector.hpp.

Referenced by convertEpsValue(), createDetectorSpecificTrackbars(), and setEps().

7.31.4.6 `const int ClusterDetector::EPS_MIN = 0` [static, private]

Definition at line 209 of file ClusterDetector.hpp.

Referenced by convertEpsValue(), and setEps().

7.31.4.7 **const int ClusterDetector::EPS_REAL_MAX = 1000** [static, private]

Definition at line 212 of file ClusterDetector.hpp.

Referenced by convertEpsValue(), and setEps().

7.31.4.8 **const int ClusterDetector::EPS_REAL_MIN = 0** [static, private]

Definition at line 211 of file ClusterDetector.hpp.

Referenced by convertEpsValue(), and setEps().

7.31.4.9 **const int ClusterDetector::MIN_POINTS_MAX = 100** [static, private]

Definition at line 207 of file ClusterDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.31.4.10 **const int ClusterDetector::MIN_POINTS_MIN = 0** [static, private]

Definition at line 206 of file ClusterDetector.hpp.

7.31.4.11 **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 29 of file ClusterDetector.hpp.

Referenced by ClusterDetector(), createDetectorSpecificTrackbars(), getMinPoints(), getValidMinPointsValue(), initialiseDetectorSpecificFields(), and setMinPoints().

7.31.4.12 **const string ClusterDetector::TRACKBAR_EPS = "Eps (Multiplied by 10)"** [static, private]

Definition at line 203 of file ClusterDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.31.4.13 **const string ClusterDetector::TRACKBAR_MINPOINTS = "Minimum number of points"** [static, private]

Definition at line 204 of file ClusterDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

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

7.32 multiscale::verification::CommandLineModelChecking Class Reference 219

- [/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](#)

7.32 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, const 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, const 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`
- 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` = ",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` = ",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` = ",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 = ",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 = ",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 = ",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_VERBOSE_NAME_LONG = "verbose"
- static const std::string ARG_VERBOSE_NAME_BOTH = ",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 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`

7.32.1 Detailed Description

Class for running model checkers from the command line.

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

7.32.2 Constructor & Destructor Documentation

7.32.2.1 `CommandLineModelChecking::CommandLineModelChecking()`

Definition at line 25 of file `CommandLineModelChecking.cpp`.

7.32.2.2 `CommandLineModelChecking::~CommandLineModelChecking()`

Definition at line 34 of file `CommandLineModelChecking.cpp`.

7.32.3 Member Function Documentation

7.32.3.1 `bool CommandLineModelChecking::areApproximateBayesianModelCheckingArgumentsPresent(const po::variables_map& variablesMap)` [private]

Check if the arguments specific to approximate Bayesian model checking are present.

Parameters

<code>variables-Map</code>	The map containing all parsed command line arguments
----------------------------	--

Definition at line 309 of file `CommandLineModelChecking.cpp`.

References `ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG`, `ARG_APPROXIMATE_BAYESIAN_BETA_NAME_LONG`, and `ARG_VARIANCE_THRESHOLD_NAME_LONG`.

Referenced by `areModelCheckingTypeSpecificArgumentsPresent()`.

7.32 multiscale::verification::CommandLineModelChecking Class Reference 229

7.32.3.2 bool CommandLineModelChecking::areApproximateProbabilisticModelCheckingArgumentsPresent (const po::variables_map & variablesMap) [private]

Check if the arguments specific to approximate probabilistic model checking are present.

Parameters

variables-Map	The map containing all parsed command line arguments
----------------------	--

Definition at line 294 of file CommandLineModelChecking.cpp.

References ARG_DELTA_NAME_LONG, and ARG_EPSILON_NAME_LONG.

Referenced by areModelCheckingTypeSpecificArgumentsPresent().

7.32.3.3 bool CommandLineModelChecking::areBayesianModelCheckingArgumentsPresent (const po::variables_map & variablesMap) [private]

Check if the arguments specific to Bayesian model checking are present.

Parameters

variables-Map	The map containing all parsed command line arguments
----------------------	--

Definition at line 301 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().

7.32.3.4 bool CommandLineModelChecking::areInvalidExecutionArguments (const po::parsed_options & parsedArguments) [private]

Check if any invalid execution arguments were provided.

Parameters

parsed-Arguments	The parsed command line arguments
-------------------------	-----------------------------------

Definition at line 152 of file CommandLineModelChecking.cpp.

References areUnrecognizedArgumentsPresent(), and isHelpArgumentPresent().

Referenced by areValidArgumentsConsideringConfiguration().

7.32.3.5 bool CommandLineModelChecking::areInvalidModelCheckingArguments() [private]

Check if any invalid model checker type dependent arguments are present.

Definition at line 211 of file CommandLineModelChecking.cpp.

References areInvalidModelCheckingArgumentsPresent(), ERR_INVALID_MODEL_CHECKING_ARGUMENTS, and MS_throw.

Referenced by areValidArgumentsConsideringConfiguration().

7.32.3.6 bool CommandLineModelChecking::areInvalidModelCheckingArgumentsPresent() [private]

Check if any model checker type dependent arguments are invalid.

Definition at line 219 of file CommandLineModelChecking.cpp.

References areInvalidModelCheckingTypeSpecificArguments(), ARG_MODEL_CHECKER_TYPE_NAME_LONG, modelCheckerType, removeOptionalArguments(), removeRequiredArguments(), and variablesMap.

Referenced by areInvalidModelCheckingArguments().

7.32.3.7 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>model- Checker- Type</i>	The type of the model checker
<i>variables- Map</i>	The map containing all parsed command line arguments

Definition at line 250 of file CommandLineModelChecking.cpp.

References areModelCheckingTypeSpecificArgumentsPresent(), and removeModelCheckingTypeSpecificArguments().

Referenced by areInvalidModelCheckingArgumentsPresent().

7.32.3.8 bool CommandLineModelChecking::areModelCheckingTypeSpecificArgumentsPresent(unsigned int modelCheckerType, const po::variables_map & variablesMap) [private]

Check if all model checking type specific arguments are present.

Parameters

<i>model- Checker- Type</i>	The type of the model checker
<i>variables- Map</i>	The map containing all parsed command line arguments

Definition at line 261 of file CommandLineModelChecking.cpp.

References `areApproximateBayesianModelCheckingArgumentsPresent()`, `areApproximateProbabilisticModelCheckingArgumentsPresent()`, `areBayesianModelCheckingArgumentsPresent()`, `areStatisticalModelCheckingArgumentsPresent()`, `E-
RR_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()`.

**7.32.3.9 bool CommandLineModelChecking::areStatisticalModelChecking-
ArgumentsPresent (const po::variables_map & variablesMap)**
[private]

Check if the arguments specific to statistical model checking are present.

Parameters

<i>variables- Map</i>	The map containing all parsed command line arguments
---------------------------	--

Definition at line 287 of file CommandLineModelChecking.cpp.

References `ARG_TYPE_I_ERROR_NAME_LONG`, and `ARG_TYPE_II_ERROR_NAME_LONG`.

Referenced by `areModelCheckingTypeSpecificArgumentsPresent()`.

**7.32.3.10 bool CommandLineModelChecking::areUnrecognized-
ArgumentsPresent (const po::parsed_options & parsedArguments)**
[private]

Check if any unrecognized command line arguments are present.

Parameters

<i>parsed- Arguments</i>	The parsed command line arguments
------------------------------	-----------------------------------

Definition at line 204 of file CommandLineModelChecking.cpp.

Referenced by `areInvalidExecutionArguments()`.

7.32.3.11 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 `isValidArgumentsConsideringConfiguration()`, and `initialiseAllowedArgumentsConfiguration()`.

Referenced by `initialise()`.

7.32.3.12 bool CommandLineModelChecking::isValidArgumentsConsideringConfiguration (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 130 of file CommandLineModelChecking.cpp.

References `areInvalidExecutionArguments()`, `areInvalidModelCheckingArguments()`, `parseAndStoreArgumentsValues()`, and `variablesMap`.

Referenced by `isValidArguments()`.

7.32.3.13 void CommandLineModelChecking::execute ()

Execute the model checking task.

Definition at line 47 of file CommandLineModelChecking.cpp.

References `modelCheckerFactory`, and `modelCheckingManager`.

Referenced by `runModelCheckingTask()`.

7.32.3.14 void CommandLineModelChecking::handleHelpRequest () [private]

Handle the help request i.e. if the `--help` flag was provided.

Definition at line 163 of file CommandLineModelChecking.cpp.

References MS_throw, MSG_MODEL_CHECKING_HELP_REQUESTED, and printHelpMessage().

Referenced by initialise().

7.32.3.15 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().

7.32.3.16 void CommandLineModelChecking::initialiseAllowedArguments- Configuration () [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().

7.32.3.17 void CommandLineModelChecking::initialiseApproximateBayesian- ModelChecker () [private]

Initialise the approximate Bayesian model checker.

Definition at line 479 of file CommandLineModelChecking.cpp.

References ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG, ARG_APPR- OXIMATE_BAYESIAN_BETA_NAME_LONG, ARG_VARIANCE_THRESHOLD_NA- ME_LONG, MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME, MODEL_CH- ECKER_APPROXIMATE_BAYESIAN_PARAMETERS_BEGIN, MODEL_CHECKER- _APPROXIMATE_BAYESIAN_PARAMETERS_END, MODEL_CHECKER_APPRO- XIMATE_BAYESIAN_PARAMETERS_MIDDLE1, MODEL_CHECKER_APPROXIM- ATE_BAYESIAN_PARAMETERS_MIDDLE2, modelCheckerFactory, modelChecker- Parameters, modelCheckerTypeName, multiscale::StringManipulator::toString(), and variablesMap.

Referenced by initialiseModelChecker().

7.32.3.18 po::options_description CommandLineModelChecking::initialiseApproximateBayesianModelCheckerArgumentsConfiguration() [private]

Initialise the configuration of the approximate Bayesian model checker command line arguments.

Definition at line 120 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_CAPTION_APPROXIMATE_BAYESIAN_MODEL_CHECKER_ARGUMENTS.

Referenced by initialiseModelCheckerTypeSpecificArgumentsConfiguration().

7.32.3.19 void CommandLineModelChecking::initialiseApproximateProbabilisticModelChecker() [private]

Initialise the approximate probabilistic model checker.

Definition at line 444 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().

7.32.3.20 po::options_description CommandLineModelChecking::initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration() [private]

Initialise the configuration of the approximate probabilistic model checker command line arguments.

Definition at line 101 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().

7.32.3.21 void CommandLineModelChecking::initialiseBayesianModelChecker()
[private]

Initialise the Bayesian model checker.

Definition at line 460 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().

7.32.3.22 po::options_description CommandLineModelChecking::initialiseBayesianModelCheckerArgumentsConfiguration()
[private]

Initialise the configuration of the Bayesian model checker command line arguments.

Definition at line 110 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().

7.32.3.23 void CommandLineModelChecking::initialiseClassMembers()
[private]

Initialise the class members using the command line arguments.

Definition at line 366 of file CommandLineModelChecking.cpp.

References initialiseModelCheckerTypeDependentClassMembers(), initialiseOptionalArgumentsDependentClassMembers(), and initialiseRequiredArgumentsDependentClassMembers().

Referenced by initialise().

7.32.3.24 void CommandLineModelChecking::initialiseModelChecker()
[private]

Initialise the model checker.

Definition at line 394 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()`.

7.32.3.25 void CommandLineModelChecking::initialiseModelCheckerTypeDependentClassMembers () [private]

Initialise the class members dependent on the model checker type.

Definition at line 389 of file `CommandLineModelChecking.cpp`.

References `initialiseModelChecker()`, and `initialiseModelCheckingManager()`.

Referenced by `initialiseClassMembers()`.

7.32.3.26 void CommandLineModelChecking::initialiseModelCheckerTypeSpecificArgumentsConfiguration () [private]

Initialise the configuration of model checker type specific command line arguments.

Definition at line 80 of file `CommandLineModelChecking.cpp`.

References `initialiseApproximateBayesianModelCheckerArgumentsConfiguration()`, `initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration()`, `initialiseBayesianModelCheckerArgumentsConfiguration()`, `initialiseStatisticalModelCheckerArgumentsConfiguration()`, and `modelCheckerTypeSpecificArguments`.

Referenced by `initialiseAllowedArgumentsConfiguration()`.

7.32.3.27 void CommandLineModelChecking::initialiseModelCheckingManager () [private]

Initialise the model checking manager.

Definition at line 498 of file `CommandLineModelChecking.cpp`.

References `extraEvaluationProgramPath`, `extraEvaluationTime`, `logicQueriesFilepath`, `modelCheckingManager`, `shouldVerboseDetailedResults`, and `tracesFolderPath`.

Referenced by `initialiseModelCheckerTypeDependentClassMembers()`.

7.32.3.28 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_VERBOSE_DESCRIPTION, ARG_VERBOSE_NAME_BOTH, and optionalArguments.

Referenced by initialiseAllowedArgumentsConfiguration().

7.32.3.29 void CommandLineModelChecking::initialiseOptionalArgumentsDependentClassMembers() [private]

Initialise the class members dependent on optional command line arguments.

Definition at line 379 of file CommandLineModelChecking.cpp.

References ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG, ARG_VERBOSE_NAME_LONG, extraEvaluationProgramPath, shouldVerboseDetailedResults, and variablesMap.

Referenced by initialiseClassMembers().

7.32.3.30 void CommandLineModelChecking::initialiseProbabilisticBlackBoxModelChecker() [private]

Initialise the probabilistic black box model checker.

Definition at line 421 of file CommandLineModelChecking.cpp.

References MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME, MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_PARAMETERS, modelCheckerFactory, modelCheckerParameters, and modelCheckerTypeName.

Referenced by initialiseModelChecker().

7.32.3.31 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().

7.32.3.32 void CommandLineModelChecking::initialiseRequiredArgumentsDependentClassMembers ()
[private]

Initialise the class members dependent on required command line arguments.

Definition at line 372 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().

7.32.3.33 void CommandLineModelChecking::initialiseStatisticalModelChecker ()
[private]

Initialise the statistical model checker.

Definition at line 428 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().

7.32.3.34 po::options_description CommandLineModelChecking::initialiseStatisticalModelCheckerArgumentsConfiguration ()
[private]

Initialise the configuration of the statistical model checker command line arguments.

Definition at line 92 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().

7.32 multiscale::verification::CommandLineModelChecking Class Reference 239

7.32.3.35 bool CommandLineModelChecking::isHelpArgumentPresent()
[private]

Check if the help command line argument is present.

Definition at line 159 of file CommandLineModelChecking.cpp.

References ARG_HELP_NAME_LONG, and variablesMap.

Referenced by areInvalidExecutionArguments(), and initialise().

7.32.3.36 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 143 of file CommandLineModelChecking.cpp.

References allowedArguments, and variablesMap.

Referenced by areValidArgumentsConsideringConfiguration().

7.32.3.37 void CommandLineModelChecking::printHelpClosingMessage()
[private]

Print the help closing message to the console.

Definition at line 191 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().

7.32.3.38 void CommandLineModelChecking::printHelpContentsMessage()
[private]

Print the help contents message to the console.

Definition at line 187 of file CommandLineModelChecking.cpp.

References allowedArguments.

Referenced by printHelpMessage().

7.32.3.39 void CommandLineModelChecking::printHelpIntroMessage ()
[private]

Print the help intro message to the console.

Definition at line 175 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().

7.32.3.40 void CommandLineModelChecking::printHelpMessage ()
[private]

Print help message to the console.

Definition at line 169 of file CommandLineModelChecking.cpp.

References printHelpClosingMessage(), printHelpContentsMessage(), and printHelpIntroMessage().

Referenced by handleHelpRequest().

**7.32.3.41 void CommandLineModelChecking::printModelCheckingInitialisation-
Message () [private]**

Print the model checking initialisation message.

Definition at line 507 of file CommandLineModelChecking.cpp.

References extraEvaluationTime, logicQueriesFilepath, modelCheckerParameters, modelCheckerTypeName, multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage(), multiscale::verification::ModelCheckingOutputWriter::printIntroductionMessage(), and tracesFolderPath.

Referenced by initialise().

**7.32.3.42 void CommandLineModelChecking::removeApproximateBayesian-
ModelCheckingArguments (po::variables_map & variablesMap)**
[private]

Remove the approximate Bayesian model checking arguments from the given variables_map.

Parameters

variables- Map	The map containing all parsed command line arguments
---------------------------	--

Definition at line 360 of file CommandLineModelChecking.cpp.

7.32 multiscale::verification::CommandLineModelChecking Class Reference 241

References ARG_APPROXIMATE_BAYESIAN_ALPHA_NAME_LONG, ARG_APPROXIMATE_BAYESIAN_BETA_NAME_LONG, and ARG_VARIANCE_THRESHOLD_NAME_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

7.32.3.43 void CommandLineModelChecking::removeApproximateProbabilisticModelCheckingArguments (po::variables_map & variablesMap) [private]

Remove the approximate probabilistic model checking arguments from the given variables_map.

Parameters

variables-Map	The map containing all parsed command line arguments
----------------------	--

Definition at line 349 of file CommandLineModelChecking.cpp.

References ARG_DELTA_NAME_LONG, and ARG_EPSILON_NAME_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

7.32.3.44 void CommandLineModelChecking::removeBayesianModelCheckingArguments (po::variables_map & variablesMap) [private]

Remove the Bayesian model checking arguments from the given variables_map.

Parameters

variables-Map	The map containing all parsed command line arguments
----------------------	--

Definition at line 354 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().

7.32.3.45 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>model- Checker- Type</i>	The type of the model checker
<i>variables- Map</i>	The map containing all parsed command line arguments

Definition at line 317 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()`.

7.32.3.46 void `CommandLineModelChecking::removeOptionalArguments(po::variables_map & variablesMap)` [private]

Remove the optional arguments from the given `variables_map`.

Parameters

<i>variables- Map</i>	The map containing all parsed command line arguments
---------------------------	--

Definition at line 236 of file CommandLineModelChecking.cpp.

References `ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG`, `ARG_HELP_NAME_LONG`, and `ARG_VERBOSE_NAME_LONG`.

Referenced by `areInvalidModelCheckingArgumentsPresent()`.

7.32.3.47 void `CommandLineModelChecking::removeRequiredArguments(po::variables_map & variablesMap)` [private]

Remove the required arguments from the given `variables_map`.

Parameters

<i>variables- Map</i>	The map containing all parsed command line arguments
---------------------------	--

Definition at line 229 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`.

7.32 multiscale::verification::CommandLineModelChecking Class Reference 243

Referenced by areInvalidModelCheckingArgumentsPresent().

```
7.32.3.48 void CommandLineModelChecking::removeStatistical-
    ModelCheckingArguments ( po::variables_map & variablesMap )
    [private]
```

Remove the statistical model checking arguments from the given variables_map.

Parameters

variables- Map	The map containing all parsed command line arguments
---------------------------	--

Definition at line 344 of file CommandLineModelChecking.cpp.

References ARG_TYPE_I_ERROR_NAME_LONG, and ARG_TYPE_II_ERROR_NAME_LONG.

Referenced by removeModelCheckingTypeSpecificArguments().

7.32.4 Member Data Documentation

```
7.32.4.1 po::options_description multiscale::verification::CommandLineModel-
    Checking::allowedArguments [private]
```

The configuration indicating which command line arguments are allowed

Definition at line 41 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), parseAndStoreArgumentsValues(), and printHelpContentsMessage().

```
7.32.4.2 const std::string CommandLineModelChecking::ARG_APPROXIMATE_B-
    AYESIAN_ALPHA_DESCRIPTION = "the alpha shape parameter of the Beta
    distribution prior" [static, private]
```

Definition at line 331 of file CommandLineModelChecking.hpp.

```
7.32.4.3 const std::string CommandLineModelChecking::ARG_APPROXIMATE_BA-
    YESIAN_ALPHA_NAME_LONG = "approximate-bayesian-alpha" [static,
    private]
```

Definition at line 330 of file CommandLineModelChecking.hpp.

Referenced by areApproximateBayesianModelCheckingArgumentsPresent(), initialiseApproximateBayesianModelChecker(), initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and removeApproximateBayesianModelCheckingArguments().

```
7.32.4.4 const std::string CommandLineModelChecking::ARG_APPROXIMATE_-  
BAYESIAN_BETA_DESCRIPTION = "the beta shape parameter of the Beta  
distribution prior" [static, private]
```

Definition at line 334 of file CommandLineModelChecking.hpp.

```
7.32.4.5 const std::string CommandLineModelChecking::ARG_APPROXIMATE_BA-  
YESIAN_BETA_NAME_LONG = "approximate-bayesian-beta" [static,  
private]
```

Definition at line 333 of file CommandLineModelChecking.hpp.

Referenced by areApproximateBayesianModelCheckingArgumentsPresent(), initialiseApproximateBayesianModelChecker(), initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and removeApproximateBayesianModelCheckingArguments().

```
7.32.4.6 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 328 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelCheckerArgumentsConfiguration().

```
7.32.4.7 const std::string CommandLineModelChecking::ARG_BAYES_FACTOR-  
_THRESHOLD_NAME_LONG = "bayes-factor-threshold" [static,  
private]
```

Definition at line 327 of file CommandLineModelChecking.hpp.

Referenced by areBayesianModelCheckingArgumentsPresent(), initialiseBayesianModelChecker(), initialiseBayesianModelCheckerArgumentsConfiguration(), and removeBayesianModelCheckingArguments().

```
7.32.4.8 const std::string CommandLineModelChecking::ARG_BAYESIAN_ALPH-  
A_DESCRIPTION = "the alpha shape parameter of the Beta distribution prior"  
[static, private]
```

Definition at line 322 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and initialiseBayesianModelCheckerArgumentsConfiguration().

```
7.32.4.9 const std::string CommandLineModelChecking::ARG_BAYE-  
SIAN_ALPHA_NAME_LONG = "bayesian-alpha" [static,  
private]
```

Definition at line 321 of file CommandLineModelChecking.hpp.

Referenced by areBayesianModelCheckingArgumentsPresent(), initialiseBayesianModelChecker(), initialiseBayesianModelCheckerArgumentsConfiguration(), and removeBayesianModelCheckingArguments().

7.32.4.10 const std::string CommandLineModelChecking::ARG_BAYESIAN_BETA_DESCRIPTION = "the beta shape parameter of the Beta distribution prior" [static, private]

Definition at line 325 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and initialiseBayesianModelCheckerArgumentsConfiguration().

7.32.4.11 const std::string CommandLineModelChecking::ARG_BAYESIAN_BETA_NAME_LONG = "bayesian-beta" [static, private]

Definition at line 324 of file CommandLineModelChecking.hpp.

Referenced by areBayesianModelCheckingArgumentsPresent(), initialiseBayesianModelChecker(), initialiseBayesianModelCheckerArgumentsConfiguration(), and removeBayesianModelCheckingArguments().

7.32.4.12 const std::string CommandLineModelChecking::ARG_DELTA_DESCRIPTION = "the upper bound on the probability to deviate from the true probability" [static, private]

Definition at line 316 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

7.32.4.13 const std::string CommandLineModelChecking::ARG_DELTA_NAME_LONG = "delta" [static, private]

Definition at line 315 of file CommandLineModelChecking.hpp.

Referenced by areApproximateProbabilisticModelCheckingArgumentsPresent(), initialiseApproximateProbabilisticModelChecker(), initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration(), and removeApproximateProbabilisticModelCheckingArguments().

7.32.4.14 const std::string CommandLineModelChecking::ARG_EPSILON_DESCRIPTION = "the considered deviation from the true probability" [static, private]

Definition at line 319 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

```
7.32.4.15 const std::string CommandLineModelChecking::ARG_EPSILON_NAME_LONG = "epsilon" [static, private]
```

Definition at line 318 of file CommandLineModelChecking.hpp.

Referenced by areApproximateProbabilisticModelCheckingArgumentsPresent(), initialiseApproximateProbabilisticModelChecker(), initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration(), and removeApproximateProbabilisticModelCheckingArguments().

```
7.32.4.16 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 303 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.32.4.17 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_PROGRAM_NAME_BOTH = ",p" [static, private]
```

Definition at line 302 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.32.4.18 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_PROGRAM_NAME_LONG = "extra-evaluation-program" [static, private]
```

Definition at line 301 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsDependentClassMembers(), and removeOptionalArguments().

```
7.32.4.19 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 291 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.32 multiscale::verification::CommandLineModelChecking Class Reference 247

7.32.4.20 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_TIME_NAME_BOTH = ",e" [static, private]

Definition at line 290 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.32.4.21 const std::string CommandLineModelChecking::ARG_EXTRA_EVALUATION_TIME_NAME_LONG = "extra-evaluation-time" [static, private]

Definition at line 289 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.32.4.22 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 299 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.32.4.23 const std::string CommandLineModelChecking::ARG_HELP_NAME_BOTH = ",h" [static, private]

Definition at line 298 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

7.32.4.24 const std::string CommandLineModelChecking::ARG_HELP_NAME_LONG = "help" [static, private]

Definition at line 297 of file CommandLineModelChecking.hpp.

Referenced by isHelpArgumentPresent(), and removeOptionalArguments().

7.32.4.25 const std::string CommandLineModelChecking::ARG_LOGIC_QUERIES_DESCRIPTION = "the path to the spatio-temporal queries input file" [static, private]

Definition at line 283 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.32.4.26 const std::string CommandLineModelChecking::ARG_
    _LOGIC_QUERIES_NAME_BOTH = ",q" [static,
    private]
```

Definition at line 282 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.32.4.27 const std::string CommandLineModelChecking::ARG_LOG-
    IC_QUERIES_NAME_LONG = "logic-queries" [static,
    private]
```

Definition at line 281 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

```
7.32.4.28 const std::string CommandLineModelChecking::ARG_MODEL_CHECKE-
    R_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 295 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.32.4.29 const std::string CommandLineModelChecking::ARG_MO-
    DEL_CHECKER_TYPE_NAME_BOTH = ",m" [static,
    private]
```

Definition at line 294 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

```
7.32.4.30 const std::string CommandLineModelChecking::ARG_MODEL_CH-
    ECKER_TYPE_NAME_LONG = "model-checker-type" [static,
    private]
```

Definition at line 293 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArgumentsPresent(), initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.32 multiscale::verification::CommandLineModelChecking Class Reference 249

7.32.4.31 const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_DESCRIPTION = "the path to the folder containing spatio-temporal traces" [static, private]

Definition at line 287 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.32.4.32 const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_NAME_BOTH = ",t" [static, private]

Definition at line 286 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsConfiguration().

7.32.4.33 const std::string CommandLineModelChecking::ARG_SPATIAL_TEMPORAL_TRACES_NAME_LONG = "spatial-temporal-traces" [static, private]

Definition at line 285 of file CommandLineModelChecking.hpp.

Referenced by initialiseRequiredArgumentsDependentClassMembers(), and removeRequiredArguments().

7.32.4.34 const std::string CommandLineModelChecking::ARG_TYPE_I_ER_ROR_DESCRIPTION = "the probability of type I errors" [static, private]

Definition at line 310 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

7.32.4.35 const std::string CommandLineModelChecking::ARG_TYPE_I_ERROR_NAME_LONG = "type-I-error" [static, private]

Definition at line 309 of file CommandLineModelChecking.hpp.

Referenced by areStatisticalModelCheckingArgumentsPresent(), initialiseStatisticalModelChecker(), initialiseStatisticalModelCheckerArgumentsConfiguration(), and removeStatisticalModelCheckingArguments().

7.32.4.36 const std::string CommandLineModelChecking::ARG_TYPE_II_ER_ROR_DESCRIPTION = "the probability of type II errors" [static, private]

Definition at line 313 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

```
7.32.4.37 const std::string CommandLineModelChecking::ARG_TY-
PE_II_ERROR_NAME_LONG = "type-II-error" [static,
private]
```

Definition at line 312 of file CommandLineModelChecking.hpp.

Referenced by areStatisticalModelCheckingArgumentsPresent(), initialiseStatisticalModelChecker(), initialiseStatisticalModelCheckerArgumentsConfiguration(), and removeStatisticalModelCheckingArguments().

```
7.32.4.38 const std::string CommandLineModelChecking::ARG_VARIANCE_THRE-
SHOLD_DESCRIPTION = "the variance threshold used to fix the confidence level
of the answer" [static, private]
```

Definition at line 337 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration().

```
7.32.4.39 const std::string CommandLineModelChecking::ARG_VARIANC-
E_THRESHOLD_NAME_LONG = "variance-threshold" [static,
private]
```

Definition at line 336 of file CommandLineModelChecking.hpp.

Referenced by areApproximateBayesianModelCheckingArgumentsPresent(), initialiseApproximateBayesianModelChecker(), initialiseApproximateBayesianModelCheckerArgumentsConfiguration(), and removeApproximateBayesianModelCheckingArguments().

```
7.32.4.40 const std::string CommandLineModelChecking::ARG_VERBOSE_DES-
CRIPTION = "if this flag is set detailed evaluation results will be displayed"
[static, private]
```

Definition at line 307 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.32.4.41 const std::string CommandLineModelChecking::A-
RG_VERBOSE_NAME_BOTH = ",v" [static,
private]
```

Definition at line 306 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsConfiguration().

```
7.32.4.42 const std::string CommandLineModelChecking::AR-
    G_VERBOSE_NAME_LONG = "verbose" [static,
    private]
```

Definition at line 305 of file CommandLineModelChecking.hpp.

Referenced by initialiseOptionalArgumentsDependentClassMembers(), and removeOptionalArguments().

```
7.32.4.43 const std::string CommandLineModelChecking::CONFIG-
    _CAPTION_ALLOWED_ARGUMENTS = "" [static,
    private]
```

Definition at line 385 of file CommandLineModelChecking.hpp.

```
7.32.4.44 const std::string CommandLineModelChecking::CONFIG_CAPTION_-
    APPROXIMATE_BAYESIAN_MODEL_CHECKER_ARGUMENTS =
    MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME [static,
    private]
```

Definition at line 394 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelCheckerArgumentsConfiguration().

```
7.32.4.45 const std::string CommandLineModelChecking::CONFIG_CAPTION_A-
    PPROXIMATE_PROBABILISTIC_MODEL_CHECKER_ARGUMENTS
    = MODEL_CHECKER_APPROXIMATE_PROBABILISTIC_NAME
    [static, private]
```

Definition at line 392 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelCheckerArgumentsConfiguration().

```
7.32.4.46 const std::string CommandLineModelChecking::CONFIG_C-
    APTION_BAYESIAN_MODEL_CHECKER_ARGUMENTS =
    MODEL_CHECKER_BAYESIAN_NAME [static, private]
```

Definition at line 393 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelCheckerArgumentsConfiguration().

```
7.32.4.47 const std::string CommandLineModelChecking::CONFIG_CAPTION_MOD-
    EL_CHECKER_TYPE_SPECIFIC_ARGUMENTS = "MODEL CHECKING TYPE
    SPECIFIC ARGUMENTS" [static, private]
```

Definition at line 388 of file CommandLineModelChecking.hpp.

```
7.32.4.48 const std::string CommandLineModelChecking::CONFIG_CAPTION_-  
OPTIONAL_ARGUMENTS = "OPTIONAL ARGUMENTS" [static,  
private]
```

Definition at line 387 of file CommandLineModelChecking.hpp.

```
7.32.4.49 const std::string CommandLineModelChecking::CONFIG_CAPTION_-  
PROBABILISTIC_BLACK_BOX_MODEL_CHECKER_ARGUMENTS  
= MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME  
[static, private]
```

Definition at line 390 of file CommandLineModelChecking.hpp.

```
7.32.4.50 const std::string CommandLineModelChecking::CONFIG_CAPTION_-  
REQUIRED_ARGUMENTS = "REQUIRED ARGUMENTS" [static,  
private]
```

Definition at line 386 of file CommandLineModelChecking.hpp.

```
7.32.4.51 const std::string CommandLineModelChecking::CONFIG_CA-  
TION_STATISTICAL_MODEL_CHECKER_ARGUMENTS =  
MODEL_CHECKER_STATISTICAL_NAME [static, private]
```

Definition at line 391 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelCheckerArgumentsConfiguration().

```
7.32.4.52 const std::string CommandLineModelChecking::ERR_INVALID_COMMAN-  
D_LINE_ARGUMENTS = "Invalid command line arguments were provided and the  
model checker execution was stopped." [static, private]
```

Definition at line 276 of file CommandLineModelChecking.hpp.

Referenced by initialise().

```
7.32.4.53 const std::string CommandLineModelChecking::ERR_INVALID_MODE-  
L_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 277 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArguments().

**7.32.4.54 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 279 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModel-
Checker(), and removeModelCheckingTypeSpecificArguments().

**7.32.4.55 std::string multiscale::verification::CommandLine-
ModelChecking::extraEvaluationProgramPath
[private]**

The path to the program which will be executed whenever more traces are required

Definition at line 33 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), and initialiseOptionalArguments-
DependentClassMembers().

**7.32.4.56 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 31 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArguments-
DependentClassMembers(), and printModelCheckingInitialisationMessage().

**7.32.4.57 const std::string CommandLineModelChecking::HELP_AUTHOR_LABEL =
"AUTHOR:" [static, private]**

Definition at line 345 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

**7.32.4.58 const std::string CommandLineModelChecking::HELP_AUTHOR_MSG = "
The author of this software is Ovidiu Parvu." [static, private]**

Definition at line 346 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.32.4.59 const std::string CommandLineModelChecking::HEL-
P_COPYRIGHT_LABEL = "COPYRIGHT:" [static,
private]
```

Definition at line 347 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.32.4.60 const std::string CommandLineModelChecking::HELP_COPYRIGHT_MSG
= " Copyright Ovidiu Parvu 2014." [static, private]
```

Definition at line 348 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.32.4.61 const std::string CommandLineModelChecking::HELP_-
DESCRIPTION_LABEL = "DESCRIPTION:" [static,
private]
```

Definition at line 343 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.62 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 spatio-temporal logic and their validity is checked using Mule."
[static, private]
```

Definition at line 344 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.63 const std::string CommandLineModelChecking::HELP_NAME_LABEL =
"NAME:" [static, private]
```

Definition at line 339 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.64 const std::string CommandLineModelChecking::HELP_NAME_MSG =
" Mule - Multidimensional multiscale model checker" [static, private]
```

Definition at line 340 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.65 const std::string CommandLineModelChecking::HELP_REPORTING_BUGS_LABEL = "REPORTING BUGS:" [static, private]
```

Definition at line 349 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.32.4.66 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 350 of file CommandLineModelChecking.hpp.

Referenced by printHelpClosingMessage().

```
7.32.4.67 const std::string CommandLineModelChecking::HELP_USAGE_LABEL = "USAGE:" [static, private]
```

Definition at line 341 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.68 const std::string CommandLineModelChecking::HELP_USAGE_MSG = "Mule <required-arguments> [<optional-arguments>] <model-checking-type-specific-arguments>" [static, private]
```

Definition at line 342 of file CommandLineModelChecking.hpp.

Referenced by printHelpIntroMessage().

```
7.32.4.69 std::string multiscale::verification::CommandLineModelChecking::logicQueriesFilepath [private]
```

The path to the logic queries file

Definition at line 26 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArgumentsDependentClassMembers(), and printModelCheckingInitialisationMessage().

```
7.32.4.70 const std::string CommandLineModelChecking::MODEL_CHECKER_APPROXIMATE_BAYESIAN_NAME = "Approximate Bayesian (mean and variance estimate)" [static, private]
```

Definition at line 379 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

```
7.32.4.71 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_BAYESIAN_PARAMETERS_BEGIN = "Beta distribution prior
shape parameters alpha = " [static, private]
```

Definition at line 380 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

```
7.32.4.72 const std::string CommandLineModelChecking::MODEL_CHECKER_-
APPROXIMATE_BAYESIAN_PARAMETERS_END = ":" [static,
private]
```

Definition at line 383 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

```
7.32.4.73 const std::string CommandLineModelChecking::MODEL_CHECKER_A-
PPROXIMATE_BAYESIAN_PARAMETERS_MIDDLE1 = " and beta = "
[static, private]
```

Definition at line 381 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

```
7.32.4.74 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_BAYESIAN_PARAMETERS_MIDDLE2 = ". Variance threshold =
" [static, private]
```

Definition at line 382 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker().

```
7.32.4.75 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_PROBABILISTIC_NAME = "Frequentist approximate probabilistic
(Chernoff-Hoeffding)" [static, private]
```

Definition at line 368 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

```
7.32.4.76 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_PROBABILISTIC_PARAMETERS_BEGIN = "Upper bound on
probability to deviate more than epsilon = " [static, private]
```

Definition at line 369 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

```
7.32.4.77 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_PROBABILISTIC_PARAMETERS_END = "." [static,
private]
```

Definition at line 371 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

```
7.32.4.78 const std::string CommandLineModelChecking::MODEL_CHECKER_APP-
ROXIMATE_PROBABILISTIC_PARAMETERS_MIDDLE = " from the true
probability is delta = " [static, private]
```

Definition at line 370 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateProbabilisticModelChecker().

```
7.32.4.79 const std::string CommandLineModelChecking::MODEL_CHECKER_B-
AYESIAN_NAME = "Bayesian (statistical hypothesis testing)" [static,
private]
```

Definition at line 373 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.32.4.80 const std::string CommandLineModelChecking::MODEL_CHECKER_BAY-
ESIAN_PARAMETERS_BEGIN = "Beta distribution prior shape parameters alpha
= " [static, private]
```

Definition at line 374 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.32.4.81 const std::string CommandLineModelChecking::MODEL_CH-
ECKER_BAYESIAN_PARAMETERS_END = "." [static,
private]
```

Definition at line 377 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.32.4.82 const std::string CommandLineModelChecking::MODEL_CHECKER_B-
AYESIAN_PARAMETERS_MIDDLE1 = " and beta = " [static,
private]
```

Definition at line 375 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.32.4.83 const std::string CommandLineModelChecking::MODEL_CHECKER_BAYESIAN_PARAMETERS_MIDDLE2 = ". Bayes factor threshold = " [static, private]
```

Definition at line 376 of file CommandLineModelChecking.hpp.

Referenced by initialiseBayesianModelChecker().

```
7.32.4.84 const std::string CommandLineModelChecking::MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_NAME = "Probabilistic black-box" [static, private]
```

Definition at line 360 of file CommandLineModelChecking.hpp.

Referenced by initialiseProbabilisticBlackBoxModelChecker().

```
7.32.4.85 const std::string CommandLineModelChecking::MODEL_CHECKER_PROBABILISTIC_BLACK_BOX_PARAMETERS = "None" [static, private]
```

Definition at line 361 of file CommandLineModelChecking.hpp.

Referenced by initialiseProbabilisticBlackBoxModelChecker().

```
7.32.4.86 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_NAME = "Frequentist statistical" [static, private]
```

Definition at line 363 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.32.4.87 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_BEGIN = "Probability of type I errors (false negatives) = " [static, private]
```

Definition at line 364 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.32.4.88 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_END = ":" [static, private]
```

Definition at line 366 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.32.4.89 const std::string CommandLineModelChecking::MODEL_CHECKER_STATISTICAL_PARAMETERS_MIDDLE = " and of type II errors (false positives) = "
[static, private]
```

Definition at line 365 of file CommandLineModelChecking.hpp.

Referenced by initialiseStatisticalModelChecker().

```
7.32.4.90 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_APPROXIMATE_BAYESIAN = 4
[static, private]
```

Definition at line 358 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.32.4.91 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_APPROXIMATE_PROBABILISTIC = 2
[static, private]
```

Definition at line 356 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.32.4.92 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_BAYESIAN = 3 [static,
private]
```

Definition at line 357 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.32.4.93 const unsigned int multiscale::verification::CommandLineModelChecking::MODEL_CHECKER_TYPE_PROBABILISTIC_BLACK_BOX = 0
[static, private]
```

Definition at line 354 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.32.4.94 const unsigned int multiscale::verification::CommandLineModel-
    Checking::MODEL_CHECKER_TYPE_STATISTICAL = 1 [static,
    private]
```

Definition at line 355 of file CommandLineModelChecking.hpp.

Referenced by areModelCheckingTypeSpecificArgumentsPresent(), initialiseModelChecker(), and removeModelCheckingTypeSpecificArguments().

```
7.32.4.95 std::shared_ptr<ModelCheckerFactory> multiscale::verification-
    ::CommandLineModelChecking::modelCheckerFactory
    [private]
```

The model checker

Definition at line 53 of file CommandLineModelChecking.hpp.

Referenced by execute(), initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseProbabilisticBlackBoxModelChecker(), and initialiseStatisticalModelChecker().

```
7.32.4.96 std::string multiscale::verification::CommandLineModelChecking-
    ::modelCheckerParameters [private]
```

The parameters specific to the model checker

Definition at line 51 of file CommandLineModelChecking.hpp.

Referenced by initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseProbabilisticBlackBoxModelChecker(), initialiseStatisticalModelChecker(), and printModelCheckingInitialisationMessage().

```
7.32.4.97 unsigned int multiscale::verification::CommandLineModelChecking-
    ::modelCheckerType [private]
```

The type of the model checker

Definition at line 29 of file CommandLineModelChecking.hpp.

Referenced by areInvalidModelCheckingArgumentsPresent(), initialiseModelChecker(), and initialiseRequiredArgumentsDependentClassMembers().

```
7.32.4.98 std::string multiscale::verification::CommandLineModelChecking-
    ::modelCheckerTypeName [private]
```

The name of the model checker type

Definition at line 50 of file CommandLineModelChecking.hpp.

7.32 multiscale::verification::CommandLineModelChecking Class Reference 261

Referenced by initialiseApproximateBayesianModelChecker(), initialiseApproximateProbabilisticModelChecker(), initialiseBayesianModelChecker(), initialiseProbabilisticBlackBoxModelChecker(), initialiseStatisticalModelChecker(), and printModelCheckingInitialisationMessage().

7.32.4.99 po::options_description multiscale::verification::CommandLineModelChecking::modelCheckerTypeSpecificArguments
[private]

The configuration indicating which command line arguments are allowed

Definition at line 47 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), and initialiseModelCheckerTypeSpecificArgumentsConfiguration().

7.32.4.100 std::shared_ptr<ModelCheckingManager> multiscale::verification::CommandLineModelChecking::modelCheckingManager
[private]

The model checking task manager

Definition at line 54 of file CommandLineModelChecking.hpp.

Referenced by execute(), and initialiseModelCheckingManager().

7.32.4.101 const std::string CommandLineModelChecking::MSG_MODEL_CHECKING_HELP_REQUESTED = "A request for displaying help information was issued." [static, private]

Definition at line 352 of file CommandLineModelChecking.hpp.

Referenced by handleHelpRequest().

7.32.4.102 po::options_description multiscale::verification::CommandLineModelChecking::optionalArguments
[private]

The configuration indicating which command line arguments are allowed

Definition at line 45 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), and initialiseOptionalArgumentsConfiguration().

7.32.4.103 po::options_description multiscale::verification::CommandLineModelChecking::requiredArguments
[private]

The configuration indicating which command line arguments are allowed

Definition at line 43 of file CommandLineModelChecking.hpp.

Referenced by initialiseAllowedArgumentsConfiguration(), and initialiseRequiredArgumentsConfiguration().

7.32.4.104 bool multiscale::verification::CommandLineModelChecking::shouldVerboseDetailedResults [private]

The flag indicating if detailed results should be printed out

Definition at line 36 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), and initialiseOptionalArgumentsDependentClassMembers().

7.32.4.105 std::string multiscale::verification::CommandLineModelChecking::tracesFolderPath [private]

The path to the folder containing traces

Definition at line 27 of file CommandLineModelChecking.hpp.

Referenced by initialiseModelCheckingManager(), initialiseRequiredArgumentsDependentClassMembers(), and printModelCheckingInitialisationMessage().

7.32.4.106 po::variables_map multiscale::verification::CommandLineModelChecking::variablesMap [private]

The map containing <a, v> pairs where a = command line argument and v = value

Definition at line 39 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

7.33 multiscale::verification::ComparatorAttribute Class Reference

Class for representing a comparator attribute.

```
#include <ComparatorAttribute.hpp>
```

Public Attributes

- [ComparatorType comparatorType](#)

7.33.1 Detailed Description

Class for representing a comparator attribute.

Definition at line 31 of file ComparatorAttribute.hpp.

7.33.2 Member Data Documentation

7.33.2.1 ComparatorType multiscale::verification::ComparatorAttribute- ::comparatorType

The comparator type

Definition at line 35 of file ComparatorAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateDifference(), multiscale::verification::LogicPropertyVisitor::evaluateNumericNumericComparison(), multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatialNumericComparison(), 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](#)

7.34 multiscale::verification::ComparatorEvaluator Class Reference

Class for evaluating comparison expressions.

```
#include <ComparatorEvaluator.hpp>
```

Static Public Member Functions

- template<typename T >
static bool [evaluate](#) (T lhsElement, const [ComparatorType](#) &comparator, T rhsElement)

Compare two elements using a ComparatorType comparator.

7.34.1 Detailed Description

Class for evaluating comparison expressions.

Definition at line 13 of file ComparatorEvaluator.hpp.

7.34.2 Member Function Documentation

7.34.2.1 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_UNDEFINED_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::ConstraintEvaluator::filterSpatialEntitiesWrtSpatialMeasure(), and multiscale::verification::ConstraintEvaluator::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/visitor/[ComparatorEvaluator.hpp](#)

7.35 multiscale::verification::ComparatorNonEqualTypeParser - Struct Reference

Symbol table and parser for the comparator type which does not accept the "=" symbol.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [ComparatorNonEqualTypeParser \(\)](#)

7.35.1 Detailed Description

Symbol table and parser for the comparator type which does not accept the "=" symbol.

Definition at line 38 of file SymbolTables.hpp.

7.35.2 Constructor & Destructor Documentation

7.35.2.1 **multiscale::verification::ComparatorNonEqualTypeParser::ComparatorNonEqualTypeParser ()**
[inline]

Definition at line 40 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](#)

7.36 multiscale::verification::ComparatorTypeParser Struct Reference -

Reference

Symbol table and parser for the comparator type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [ComparatorTypeParser \(\)](#)

7.36.1 Detailed Description

Symbol table and parser for the comparator type.

Definition at line 23 of file SymbolTables.hpp.

7.36.2 Constructor & Destructor Documentation

7.36.2.1 **multiscale::verification::ComparatorTypeParser::ComparatorTypeParser () [inline]**

Definition at line 25 of file SymbolTables.hpp.

References multiscale::verification::Equal, multiscale::verification::GreaterThan, multiscale::verification::GreaterThanOrEqual, multiscale::verification::LessThan, and multiscale::verification::LessThanOrEqual.

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

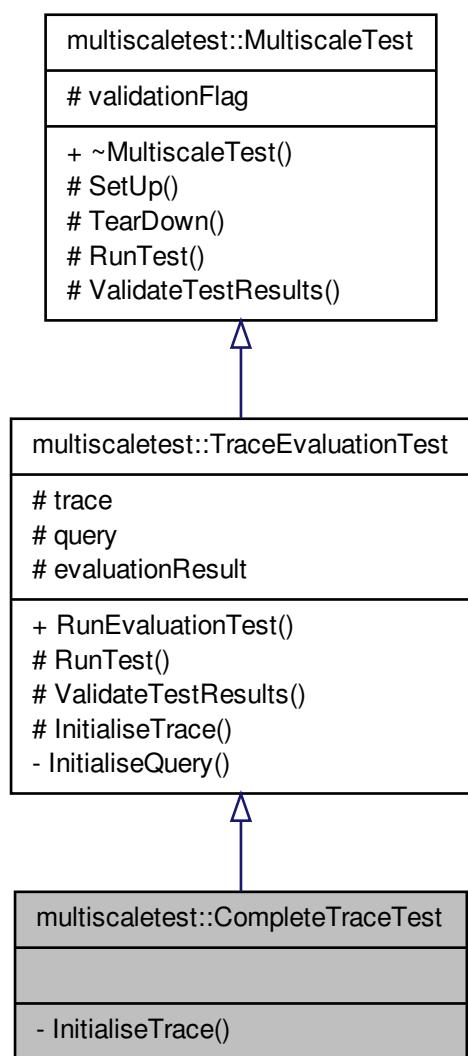
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.-hpp](#)

7.37 multiscaletest::CompleteTraceTest Class Reference

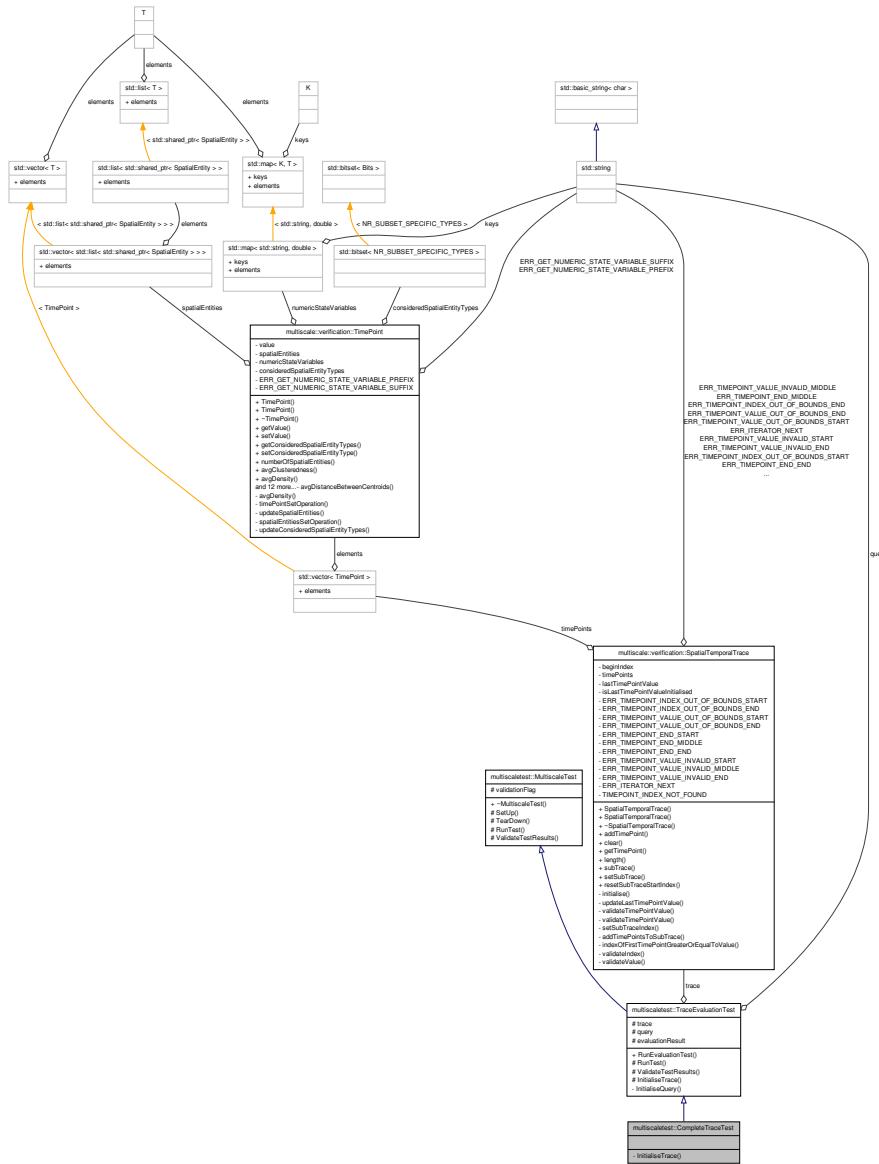
Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.

```
#include <CompleteTraceTest.hpp>
```

Inheritance diagram for multiscaletest::CompleteTraceTest:



Collaboration diagram for multiscaletest::CompleteTraceTest:



Private Member Functions

- virtual void `InitialiseTrace()` override

Initialise the trace

7.37.1 Detailed Description

Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.

Definition at line 13 of file CompleteTraceTest.hpp.

7.37.2 Member Function Documentation

7.37.2.1 void multiscaletest::CompleteTraceTest::InitialiseTrace () [override, private, virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 22 of file CompleteTraceTest.hpp.

References multiscale::verification::Clusters, and multiscale::verification::Regions.

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

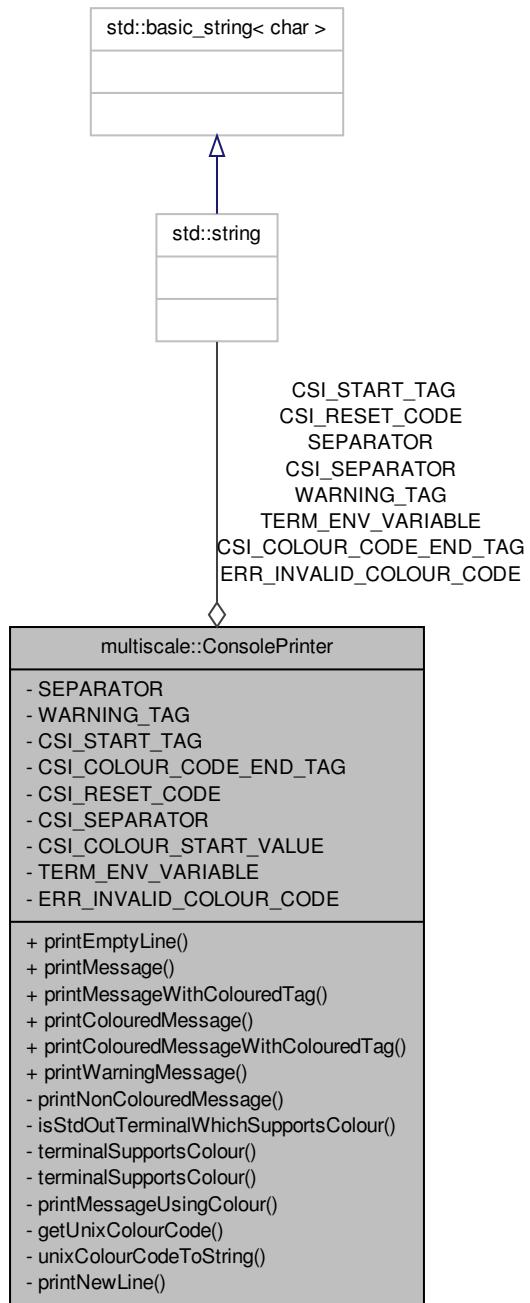
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[CompleteTraceTest.hpp](#)

7.38 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 [printNewLine \(bool shouldPrint=true\)](#)
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)."

7.38.1 Detailed Description

Class used to print (coloured) messages to the console.

Definition at line 57 of file ConsolePrinter.hpp.

7.38.2 Member Function Documentation

7.38.2.1 std::string ConsolePrinter::getUnixColourCode (const UnixColourCode & unixColourCode) [static, private]

Get the CSI string representation corresponding to the given UNIX colour code.

Parameters

<code>unixColour- Code</code>	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()`.

7.38.2.2 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()`.

7.38.2.3 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().

7.38.2.4 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>message-Colour</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().

7.38.2.5 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().

7.38.2.6 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().

7.38.2.7 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().

7.38.2.8 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::printDetailedEvaluationResultsIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printEvaluationResultsSummary(), multiscale::verification::ModelCheckingOutputWriter::printExecuteExtraEvaluationProgramMessage(), multiscale::verification::ModelCheckingOutputWriter::printInitialisationMessage(), multiscale::verification::ModelCheckingOutputWriter::printIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printLogicPropertyWithTag(), multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage(), multiscale::verification::ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage(), multiscale::verification::ModelCheckingOutputWriter::printStartTraceEvaluationMessage(), multiscale::verification::ModelCheckingOutputWriter::printTimeoutMessage(), multiscale::verification::ModelCheckingOutputWriter::printTruthValueDependentMessage(), and printWarningMessage().

7.38.2.9 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().

7.38.2.10 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 print.NewLine().

Referenced by printColouredMessage(), printColouredMessageWithColouredTag(), printMessage(), and printMessageWithColouredTag().

7.38.2.11 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::verification::LogicPropertyVisitor::printExceptionMessage(), and multiscale::Numeric::printNoValuesWarningMessage().

7.38.2.12 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().

7.38.2.13 `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().

7.38.2.14 `std::string ConsolePrinter::unixColourCodeToString(const UnixColourCode & unixColourCode) [static, private]`

Get the string representation corresponding to the given UNIX colour code.

Parameters

<code>unixColour- Code</code>	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().

7.38.3 Member Data Documentation

7.38.3.1 `const std::string ConsolePrinter::CSI_COLOUR_CODE_END_TAG = "m" [static, private]`

Definition at line 187 of file ConsolePrinter.hpp.

Referenced by getUnixColourCode().

7.38.3.2 `const int ConsolePrinter::CSI_COLOUR_START_VALUE = 30 [static, private]`

Definition at line 191 of file ConsolePrinter.hpp.

Referenced by unixColourCodeToString().

7.38.3.3 `const std::string ConsolePrinter::CSI_RESET_CODE = "0" [static, private]`

Definition at line 188 of file ConsolePrinter.hpp.

Referenced by `getUnixColourCode()`.

7.38.3.4 `const std::string ConsolePrinter::CSI_SEPARATOR = ";" [static, private]`

Definition at line 189 of file ConsolePrinter.hpp.

Referenced by `getUnixColourCode()`.

7.38.3.5 `const std::string ConsolePrinter::CSI_START_TAG = "\033[" [static, private]`

Definition at line 186 of file ConsolePrinter.hpp.

Referenced by `getUnixColourCode()`.

7.38.3.6 `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.

7.38.3.7 `const std::string ConsolePrinter::SEPARATOR = "" [static, private]`

Definition at line 182 of file ConsolePrinter.hpp.

Referenced by `printColouredMessageWithColouredTag()`, and `printMessageWithColouredTag()`.

7.38.3.8 `const std::string ConsolePrinter::TERM_ENV_VARIABLE = "TERM" [static, private]`

Definition at line 193 of file ConsolePrinter.hpp.

Referenced by `terminalSupportsColour()`.

7.38.3.9 `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](#)

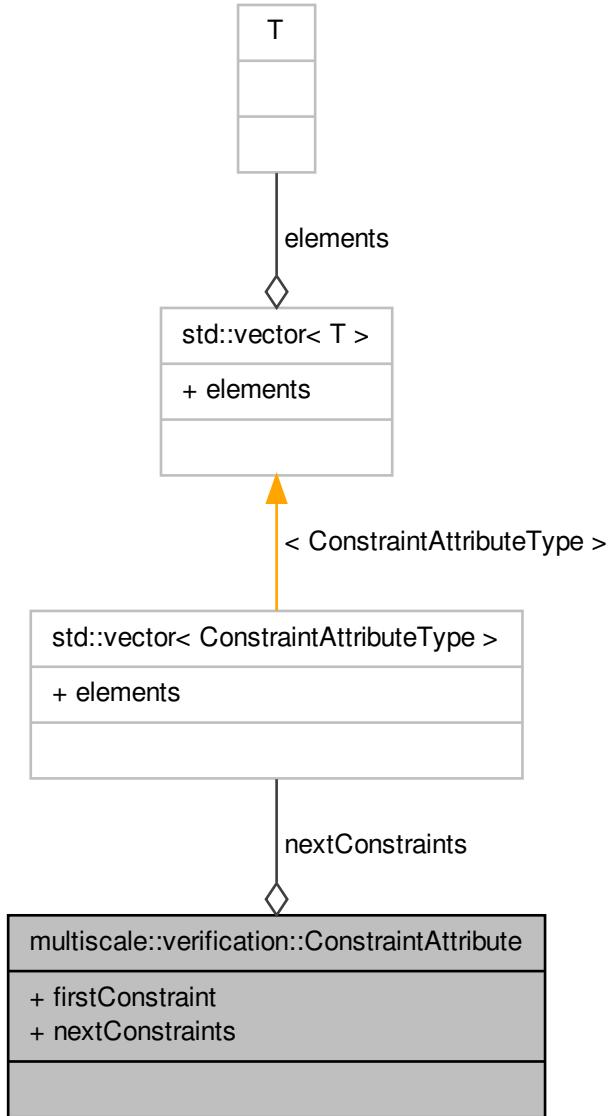
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[Console-
Printer.cpp](#)

7.39 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

7.39.1 Detailed Description

Class for representing a constraint attribute.

Definition at line 36 of file ConstraintAttribute.hpp.

7.39.2 Member Data Documentation

7.39.2.1 [ConstraintAttributeType](#) multiscale::verification::ConstraintAttribute::firstConstraint

The first constraint

Definition at line 40 of file ConstraintAttribute.hpp.

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

7.39.2.2 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](#)

7.40 multiscale::verification::ConstraintEvaluator Class Reference

Class for evaluating constraint expressions.

```
#include <ConstraintEvaluator.hpp>
```

Static Public Member Functions

- static void evalSpatialMeasureConstraint (TimePoint &timePoint, const [SpatialMeasureType](#) &spatialMeasure, const [ComparatorType](#) &comparator, const [FilterNumericMeasureAttributeType](#) &filterNumericMeasure)

Filter the timepoint's spatial entities considering the given spatial measure constraint.

- static void `evalTypeConstraint` (`TimePoint` &`timePoint`, const `ComparatorType` &`comparator`, const `FilterNumericMeasureAttributeType` &`filterNumericMeasure`)

Filter the timepoint's spatial entities considering the type of each spatial entity.

Static Private Member Functions

- static void `filterSpatialEntitiesWrtSpatialMeasure` (`TimePoint` &`timePoint`, const `SubsetSpecificType` &`spatialEntityType`, const `SpatialMeasureType` &`spatialMeasure`, const `ComparatorType` &`comparator`, const `FilterNumericMeasureAttributeType` &`filterNumericMeasure`)

Remove from the timepoint the spatial entities which fail to meet the spatial measure constraint.

- static void `filterSpatialEntitiesWrtType` (`TimePoint` &`timePoint`, const `SubsetSpecificType` &`spatialEntityType`, const `ComparatorType` &`comparator`, const `FilterNumericMeasureAttributeType` &`filterNumericMeasure`)

Remove from the timepoint the spatial entities which fail to meet the type constraint.

- static double `evalFilterNumericMeasure` (const `FilterNumericMeasureAttributeType` &`filterNumericMeasure`, const `TimePoint` &`timePoint`, const `SpatialEntity` &`spatialEntity`)

Evaluate the filter numeric measure considering the provided timepoint and spatial entity.

7.40.1 Detailed Description

Class for evaluating constraint expressions.

Definition at line 17 of file ConstraintEvaluator.hpp.

7.40.2 Member Function Documentation

- 7.40.2.1 static double `multiscale::verification::ConstraintEvaluator::evalFilterNumericMeasure` (const `FilterNumericMeasureAttributeType` & `filterNumericMeasure`, const `TimePoint` & `timePoint`, const `SpatialEntity` & `spatialEntity`) [inline, static, private]

Evaluate the filter numeric measure considering the provided timepoint and spatial entity.

Parameters

<code>filter-Numeric-Measure</code>	The filter numeric measure
<code>timePoint</code>	The considered timepoint
<code>spatialEntity</code>	The considered spatial entity

Definition at line 130 of file ConstraintEvaluator.hpp.

Referenced by filterSpatialEntitiesWrtSpatialMeasure(), and filterSpatialEntitiesWrtType().

```
7.40.2.2 static void multiscale::verification::ConstraintEvaluator::evalSpatial-
    MeasureConstraint ( TimePoint & timePoint, const SpatialMeasureType
    & spatialMeasure, const ComparatorType & comparator, const
    FilterNumericMeasureAttributeType & filterNumericMeasure ) [inline,
    static]
```

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>spatial- Measure</i>	The type of the spatial measure
<i>comparator</i>	The type of the comparator
<i>filter- Numeric- Measure</i>	The filter numeric measure

Definition at line 31 of file ConstraintEvaluator.hpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::TimePoint::getConsideredSpatialEntityTypes(), and multiscale::verification::NR_SUBSET_SPECIFIC_TYPES.

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

```
7.40.2.3 static void multiscale::verification::ConstraintEvaluator::evalType-
    Constraint ( TimePoint & timePoint, const ComparatorType & comparator,
    const FilterNumericMeasureAttributeType & filterNumericMeasure )
    [inline, static]
```

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>filter- Numeric- Measure</i>	The filter numeric measure

Definition at line 53 of file ConstraintEvaluator.hpp.

References multiscale::verification::subsetsspecific::computeSubsetSpecificType(), filterSpatialEntitiesWrtType(), multiscale::verification::TimePoint::getConsideredSpatialEntityTypes(), and multiscale::verification::NR_SUBSET_SPECIFIC_TYPES.

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

```
7.40.2.4 static void multiscale::verification::ConstraintEvaluator::filter-
    SpatialEntitiesWrtSpatialMeasure ( TimePoint & timePoint, const
    SubsetSpecificType & spatialEntityType, const SpatialMeasureType
    & spatialMeasure, const ComparatorType & comparator, const
    FilterNumericMeasureAttributeType & filterNumericMeasure ) [inline,
    static, 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 78 of file ConstraintEvaluator.hpp.

References evalFilterNumericMeasure(), multiscale::verification::SpatialMeasureEvaluator::evaluate(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), and multiscale::verification::TimePoint::removeSpatialEntity().

Referenced by evalSpatialMeasureConstraint().

```
7.40.2.5 static void multiscale::verification::ConstraintEvaluator::filterSpatial-
    EntitiesWrtType ( TimePoint & timePoint, const SubsetSpecificType
    & spatialEntityType, const ComparatorType & comparator, const
    FilterNumericMeasureAttributeType & filterNumericMeasure ) [inline,
    static, 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>spatialEntity-Type</i>	The considered spatial entity type
<i>comparator</i>	The type of the comparator
<i>filter-Numeric-Measure</i>	The filter numeric measure

Definition at line 105 of file ConstraintEvaluator.hpp.

References evalFilterNumericMeasure(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::TimePoint::getSpatialEntitiesEndIterator(), and multiscale::verification::TimePoint::removeSpatialEntity().

Referenced by evalTypeConstraint().

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

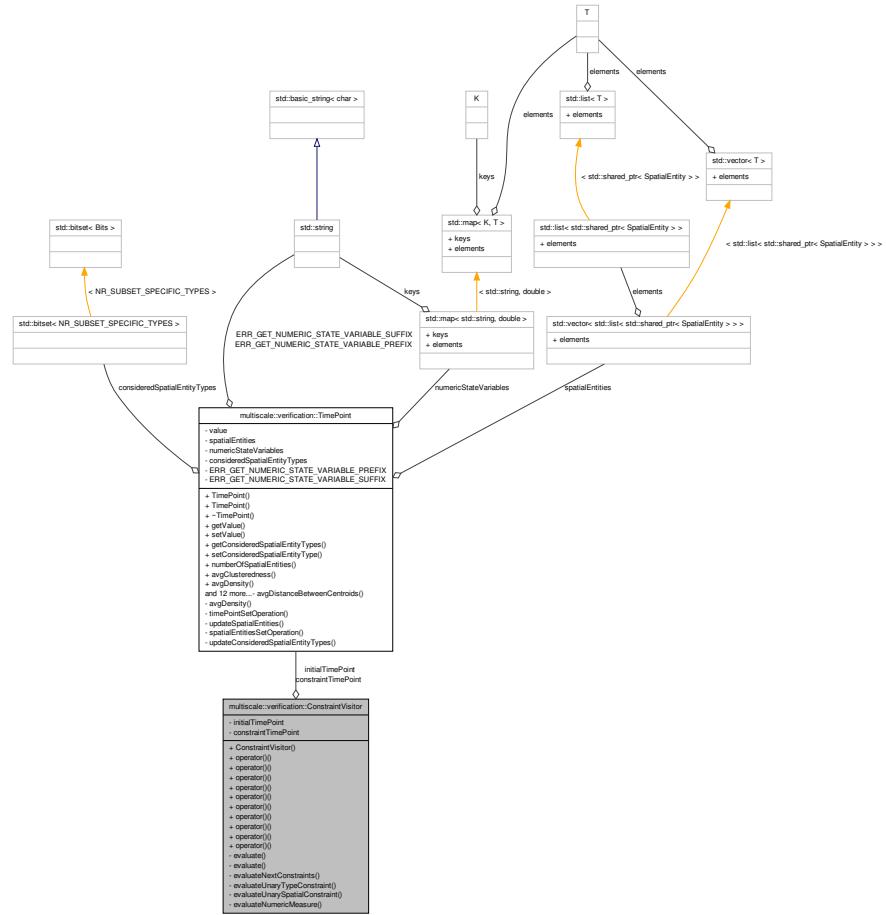
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[Constraint-Evaluator.hpp](#)

7.41 multiscale::verification::ConstraintVisitor Class Reference

Class used to evaluate constraints.

```
#include <ConstraintVisitor.hpp>
```

Collaboration diagram for multiscale::verification::ConstraintVisitor:



Public Member Functions

- **ConstraintVisitor** (const **TimePoint** &initialTimePoint, const **TimePoint** &constraintTimePoint)
 - **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 FilterNumericMeasureAttributeType &filterNumericMeasure, const TimePoint &timePoint) const`

Evaluate the unary type constraint considering the given spatial measure, comparator, numeric measure and timepoint.
- `TimePoint evaluateUnarySpatialConstraint (const SpatialMeasureType &spatialMeasure, const ComparatorType &comparator, const FilterNumericMeasureAttributeType &filterNumericMeasure, const TimePoint &timePoint) const`

Evaluate the unary spatial constraint considering the given spatial measure, comparator, numeric measure and timepoint.
- `double evaluateNumericMeasure (const NumericMeasureAttributeType &numericMeasure, const TimePoint &timePoint) const`

Evaluate the numeric measure considering the given timepoint.

Private Attributes

- `const TimePoint & initialTimePoint`
- `const TimePoint & constraintTimePoint`

7.41.1 Detailed Description

Class used to evaluate constraints.

Definition at line 12 of file ConstraintVisitor.hpp.

7.41.2 Constructor & Destructor Documentation

7.41.2.1 multiscale::verification::ConstraintVisitor::ConstraintVisitor (const TimePoint & initialTimePoint, const TimePoint & constraintTimePoint) [inline]

Definition at line 21 of file ConstraintVisitor.hpp.

Referenced by evaluate(), and evaluateNextConstraints().

7.41.3 Member Function Documentation

7.41.3.1 TimePoint multiscale::verification::ConstraintVisitor::evaluate (const ConstraintAttributeType & constraint, const TimePoint & timePoint) const [inline, private]

Evaluate the constraint considering the given timepoint.

Parameters

<i>constraint</i>	The given constraint
<i>timePoint</i>	The given timepoint

Definition at line 155 of file ConstraintVisitor.hpp.

References ConstraintVisitor(), and initialTimePoint.

Referenced by operator()().

7.41.3.2 TimePoint multiscale::verification::ConstraintVisitor::evaluate (const PrimaryConstraintAttributeType & primaryConstraint, const TimePoint & timePoint) const [inline, private]

Evaluate the primary constraint considering the given timepoints.

Parameters

<i>primary- Constraint</i>	The given primary constraint
<i>timePoint</i>	The given timepoint

Definition at line 164 of file ConstraintVisitor.hpp.

References ConstraintVisitor(), and initialTimePoint.

7.41.3.3 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 175 of file ConstraintVisitor.hpp.

References ConstraintVisitor(), initialTimePoint, and multiscale::verification::ConstraintAttribute::nextConstraints.

Referenced by operator()().

7.41.3.4 double multiscale::verification::ConstraintVisitor::evaluateNumericMeasure (const NumericMeasureAttributeType & *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 224 of file ConstraintVisitor.hpp.

7.41.3.5 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 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>filter- Numeric- Measure</i>	The filter numeric measure
<i>timePoint</i>	The considered timepoint

Definition at line 209 of file ConstraintVisitor.hpp.

References multiscale::verification::ConstraintEvaluator::evalSpatialMeasureConstraint().

Referenced by operator()().

**7.41.3.6 TimePoint multiscale::verification::ConstraintVisitor::evaluate-
UnaryTypeConstraint (const ComparatorType & comparator, const
FilterNumericMeasureAttributeType & *filterNumericMeasure*, const TimePoint
& *timePoint*) const [inline, private]**

Evaluate the unary type constraint considering the given spatial measure, comparator, numeric measure and timepoint.

Parameters

<i>comparator</i>	The comparator type
<i>filter- Numeric- Measure</i>	The filter numeric measure
<i>timePoint</i>	The considered timepoint

Definition at line 192 of file ConstraintVisitor.hpp.

References multiscale::verification::ConstraintEvaluator::evalTypeConstraint().

Referenced by operator()().

**7.41.3.7 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 28 of file ConstraintVisitor.hpp.

References initialTimePoint.

7.41.3.8 **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 36 of file ConstraintVisitor.hpp.

References constraintTimePoint, evaluate(), evaluateNextConstraints(), and multiscale-::verification::ConstraintAttribute::firstConstraint.

7.41.3.9 **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 46 of file ConstraintVisitor.hpp.

References multiscale::verification::OrConstraintAttribute::constraint, constraintTime-Point, evaluate(), and multiscale::verification::TimePoint::timePointUnion().

7.41.3.10 **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 58 of file ConstraintVisitor.hpp.

References multiscale::verification::AndConstraintAttribute::constraint, constraintTime-Point, evaluate(), and multiscale::verification::TimePoint::timePointIntersection().

7.41.3.11 **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 70 of file ConstraintVisitor.hpp.

References multiscale::verification::ImplicationConstraintAttribute::constraint, constraintTimePoint, evaluate(), initialTimePoint, multiscale::verification::TimePoint::timePointDifference(), and multiscale::verification::TimePoint::timePointUnion().

7.41.3.12 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 85 of file ConstraintVisitor.hpp.

References multiscale::verification::EquivalenceConstraintAttribute::constraint, constraintTimePoint, evaluate(), initialTimePoint, multiscale::verification::TimePoint::timePointDifference(), multiscale::verification::TimePoint::timePointIntersection(), and multiscale::verification::TimePoint::timePointUnion().

7.41.3.13 TimePoint multiscale::verification::ConstraintVisitor::operator() (const PrimaryConstraintAttribute & *primaryConstraint*) const [inline]

Overloading the "(") operator for the [PrimaryConstraintAttribute](#) alternative.

Parameters

<i>primary- Constraint</i>	The primary constraint
--------------------------------	------------------------

Definition at line 108 of file ConstraintVisitor.hpp.

References constraintTimePoint, evaluate(), and multiscale::verification::PrimaryConstraintAttribute::primaryConstraint.

7.41.3.14 TimePoint multiscale::verification::ConstraintVisitor::operator() (const NotConstraintAttribute & *primaryConstraint*) const [inline]

Overloading the "(") operator for the [NotConstraintAttribute](#) alternative.

Parameters

<i>primary- Constraint</i>	The primary constraint
--------------------------------	------------------------

Definition at line 116 of file ConstraintVisitor.hpp.

References multiscale::verification::NotConstraintAttribute::constraint, constraintTimePoint, evaluate(), initialTimePoint, and multiscale::verification::TimePoint::timePointDifference().

7.41.3.15 TimePoint multiscale::verification::ConstraintVisitor::operator() (const UnaryTypeConstraintAttribute & primaryConstraint) const [inline]

Overloading the "(" operator for the [UnaryTypeConstraintAttribute](#) alternative.

Parameters

<i>primary- Constraint</i>	The primary constraint
--------------------------------	------------------------

Definition at line 129 of file ConstraintVisitor.hpp.

References multiscale::verification::UnaryTypeConstraintAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, constraintTimePoint, evaluateUnaryTypeConstraint(), and multiscale::verification::UnaryTypeConstraintAttribute::filterNumericMeasure.

7.41.3.16 TimePoint multiscale::verification::ConstraintVisitor::operator() (const UnarySpatialConstraintAttribute & primaryConstraint) const [inline]

Overloading the "(" operator for the [UnarySpatialConstraintAttribute](#) alternative.

Parameters

<i>primary- Constraint</i>	The primary constraint
--------------------------------	------------------------

Definition at line 140 of file ConstraintVisitor.hpp.

References multiscale::verification::UnarySpatialConstraintAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, constraintTimePoint, evaluateUnarySpatialConstraint(), multiscale::verification::UnarySpatialConstraintAttribute::filterNumericMeasure, multiscale::verification::UnarySpatialConstraintAttribute::spatialMeasure, and multiscale::verification::SpatialMeasureAttribute::spatialMeasureType.

7.41.4 Member Data Documentation

7.41.4.1 const TimePoint& multiscale::verification::ConstraintVisitor::constraintTimePoint [private]

The currently obtained constraint timepoint

Definition at line 17 of file ConstraintVisitor.hpp.

Referenced by operator()().

7.41.4.2 const TimePoint& multiscale::verification::ConstraintVisitor::initialTime-Point [private]

A copy of the initial timepoint

Definition at line 16 of file ConstraintVisitor.hpp.

Referenced by evaluate(), evaluateNextConstraints(), and operator()().

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

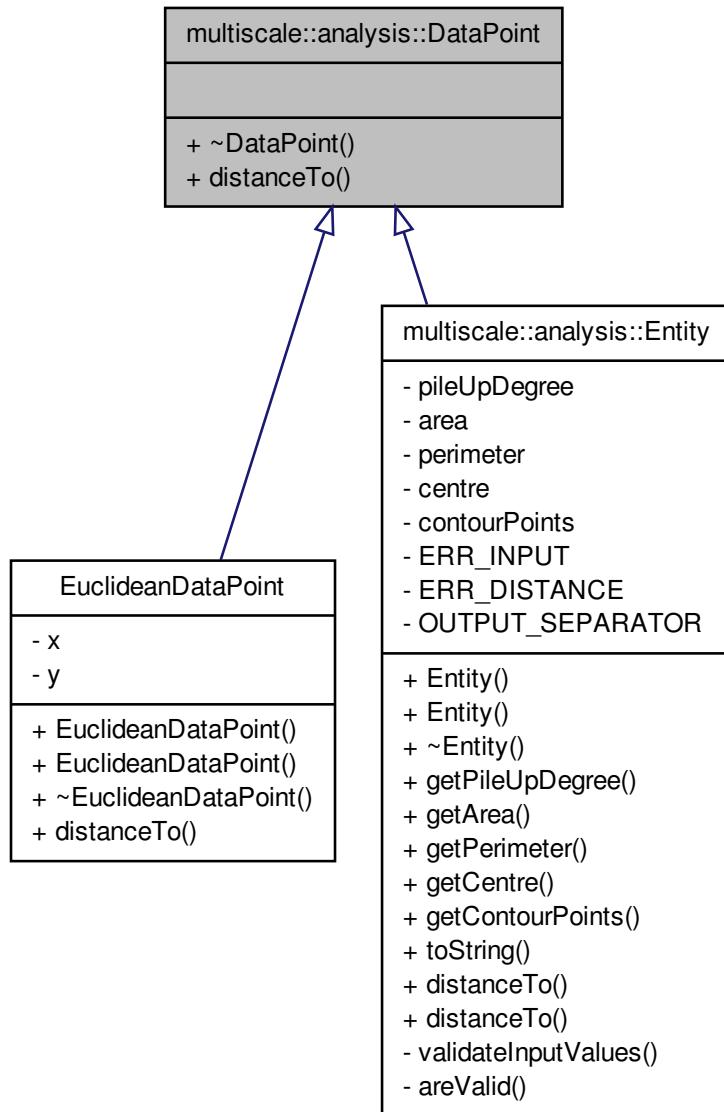
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[ConstraintVisitor.hpp](#)

7.42 multiscale::analysis::DataPoint Class Reference

Class for representing a data point.

```
#include <DataPoint.hpp>
```

Inheritance diagram for multiscale::analysis::DataPoint:



Public Member Functions

- `virtual ~DataPoint ()`

- virtual double `distanceTo` (`shared_ptr< DataPoint > point`)=0

Compute the distance between this data point and another one.

7.42.1 Detailed Description

Class for representing a data point.

Definition at line 13 of file `DataPoint.hpp`.

7.42.2 Constructor & Destructor Documentation

- 7.42.2.1 virtual `multiscale::analysis::DataPoint::~DataPoint()` [inline, virtual]

Definition at line 17 of file `DataPoint.hpp`.

7.42.3 Member Function Documentation

- 7.42.3.1 virtual double `multiscale::analysis::DataPoint::distanceTo(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:

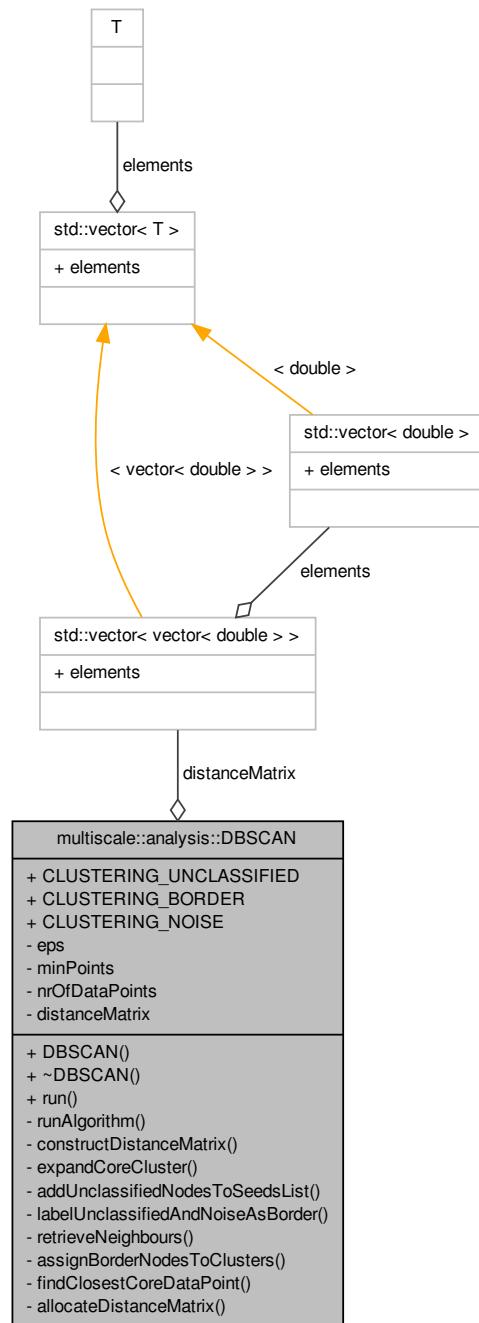
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/sp/
`DataPoint.hpp`

7.43 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 vector< shared_ptr< DataPoint >> &dataPoints, 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 vector< shared_ptr< DataPoint >> &dataPoints, vector< int > &clusterIndexes, int &nrOfClusters)`

Run the improved DBSCAN algorithm on the provided set of points.
- `void constructDistanceMatrix (const vector< shared_ptr< DataPoint >> &dataPoints)`

Construct the distance matrix between any two data points.
- `bool expandCoreCluster (vector< int > &clusterIndexes, int coreDataPointIndex, int clusterId)`

Expand the cluster around the given core data point.
- `void addUnclassifiedNodesToSeedsList (const vector< int > &neighbours, const vector< int > &clusterIndexes, vector< int > &seeds)`

Add all unclassified neighbour nodes to the seeds list.
- `void labelUnclassifiedAndNoiseAsBorder (const vector< int > &neighbours, const vector< int > &clusterIndexes)`

Label all unclassified and noise neighbour nodes as border nodes.
- `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 (vector< int > &clusterIndexes)`

Assign the border nodes to the clusters to which the closest core objects belong.
- `int findClosestCoreDataPoint (const vector< int > &neighbours, int borderDataPointIndex, const 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`
- vector<vector<double>> `distanceMatrix`

7.43.1 Detailed Description

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

Definition at line 17 of file DBSCAN.hpp.

7.43.2 Constructor & Destructor Documentation

7.43.2.1 DBSCAN::DBSCAN()

Definition at line 9 of file DBSCAN.cpp.

7.43.2.2 DBSCAN::~DBSCAN()

Definition at line 11 of file DBSCAN.cpp.

References `distanceMatrix`.

7.43.3 Member Function Documentation

7.43.3.1 void DBSCAN::addUnclassifiedNodesToSeedsList (const vector<int> & *neighbours*, const vector<int> & *clusterIndexes*, vector<int> & *seeds*) [private]

Add all unclassified neighbour nodes to the seeds list.

Parameters

<i>neighbours</i>	Neighbour nodes
<i>cluster-Indexes</i>	Indexes to which cluster each data point belongs
<i>seeds</i>	List of seeds (see DBSCAN algorithm)

Definition at line 85 of file DBSCAN.cpp.

References `CLUSTERING_UNCLASSIFIED`.

Referenced by `expandCoreCluster()`.

7.43.3.2 void DBSCAN::allocateDistanceMatrix() [private]

Allocate the distance matrix.

Definition at line 148 of file DBSCAN.cpp.

References distanceMatrix, and nrOfDataPoints.

Referenced by constructDistanceMatrix().

7.43.3.3 void DBSCAN::assignBorderNodesToClusters (vector< int > & clusterIndexes) [private]

Assign the border nodes to the clusters to which the closest core objects belong.

Parameters

<i>cluster-Indexes</i>	Indexes to which cluster each data point belongs
------------------------	--

Definition at line 117 of file DBSCAN.cpp.

References CLUSTERING_BORDER, findClosestCoreDataPoint(), nrOfDataPoints, and retrieveNeighbours().

Referenced by runAlgorithm().

7.43.3.4 void DBSCAN::constructDistanceMatrix (const vector< shared_ptr< DataPoint >> & dataPoints) [private]

Construct the distance matrix between any two data points.

Parameters

<i>dataPoints</i>	Data points
-------------------	-------------

Definition at line 44 of file DBSCAN.cpp.

References allocateDistanceMatrix(), distanceMatrix, and nrOfDataPoints.

Referenced by run().

7.43.3.5 bool DBSCAN::expandCoreCluster (vector< int > & clusterIndexes, int coreDataPointIndex, int clusterId) [private]

Expand the cluster around the given core data point.

Parameters

<i>cluster-Indexes</i>	Indexes to which cluster each data point belongs
------------------------	--

<i>coreData-PointIndex</i>	Core data point index
<i>clusterId</i>	Id of the cluster to which the core data point belongs

Definition at line 57 of file DBSCAN.cpp.

References addUnclassifiedNodesToSeedsList(), CLUSTERING_NOISE, labelUnclassifiedAndNoiseAsBorder(), minPoints, and retrieveNeighbours().

Referenced by runAlgorithm().

7.43.3.6 int DBSCAN::findClosestCoreDataPoint (const vector< int > & *neighbours*, int *borderDataPointIndex*, const 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>borderData-PointIndex</i>	Index of the border data point
<i>cluster-Indexes</i>	Indexes to which cluster each data point belongs

Definition at line 128 of file DBSCAN.cpp.

References distanceMatrix.

Referenced by assignBorderNodesToClusters().

7.43.3.7 void DBSCAN::labelUnclassifiedAndNoiseAsBorder (const vector< int > & *neighbours*, vector< int > & *clusterIndexes*) [private]

Label all unclassified and noise neighbour nodes as border nodes.

Parameters

<i>neighbours</i>	Neighbour nodes
<i>cluster-Indexes</i>	Indexes to which cluster each data point belongs

Definition at line 94 of file DBSCAN.cpp.

References CLUSTERING_BORDER, CLUSTERING_NOISE, and CLUSTERING_UNCLASSIFIED.

Referenced by expandCoreCluster().

7.43.3.8 `vector< int > DBSCAN::retrieveNeighbours (int dataPointIndex)`
[private]

Retrieve the list of neighbour indexes which are at a distance $< \text{eps}$ far from the given data point.

Parameters

<i>dataPoint-Index</i>	Index of the data point for which the neighbours will be retrieved
------------------------	--

Definition at line 103 of file DBSCAN.cpp.

References `distanceMatrix`, `eps`, and `nrOfDataPoints`.

Referenced by `assignBorderNodesToClusters()`, and `expandCoreCluster()`.

7.43.3.9 `void DBSCAN::run (const vector< shared_ptr< DataPoint >> & dataPoints, 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>cluster-Indexes</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()`.

7.43.3.10 `void DBSCAN::runAlgorithm (const vector< shared_ptr< DataPoint >> & dataPoints, 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 26 of file DBSCAN.cpp.

References assignBorderNodesToClusters(), CLUSTERING_UNCLASSIFIED, expandCoreCluster(), and nrOfDataPoints.

Referenced by run().

7.43.4 Member Data Documentation

7.43.4.1 const int DBSCAN::CLUSTERING_BORDER = -1 [static]

Definition at line 122 of file DBSCAN.hpp.

Referenced by assignBorderNodesToClusters(), and labelUnclassifiedAndNoiseAsBorder().

7.43.4.2 const int DBSCAN::CLUSTERING_NOISE = 0 [static]

Definition at line 123 of file DBSCAN.hpp.

Referenced by expandCoreCluster(), and labelUnclassifiedAndNoiseAsBorder().

7.43.4.3 const int DBSCAN::CLUSTERING_UNCLASSIFIED = -2 [static]

Definition at line 121 of file DBSCAN.hpp.

Referenced by addUnclassifiedNodesToSeedsList(), multiscale::analysis::ClusterDetector::detectAndAnalyseClusters(), labelUnclassifiedAndNoiseAsBorder(), and runAlgorithm().

7.43.4.4 vector<vector<double>> multiscale::analysis::DBSCAN::distanceMatrix [private]

The matrix containing the distances between any two data points

Definition at line 28 of file DBSCAN.hpp.

Referenced by allocateDistanceMatrix(), constructDistanceMatrix(), findClosestCoreDataPoint(), retrieveNeighbours(), and ~DBSCAN().

7.43.4.5 double multiscale::analysis::DBSCAN::eps [private]

[DBSCAN](#) algorithm parameter for specifying the maximum radius of the neighbourhood

Definition at line 21 of file DBSCAN.hpp.

Referenced by retrieveNeighbours(), and run().

7.43.4.6 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 23 of file DBSCAN.hpp.

Referenced by expandCoreCluster(), and run().

7.43.4.7 unsigned int multiscale::analysis::DBSCAN::nrOfDataPoints [private]

Number of data points in the data set

Definition at line 26 of file DBSCAN.hpp.

Referenced by allocateDistanceMatrix(), assignBorderNodesToClusters(), constructDistanceMatrix(), retrieveNeighbours(), run(), and runAlgorithm().

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

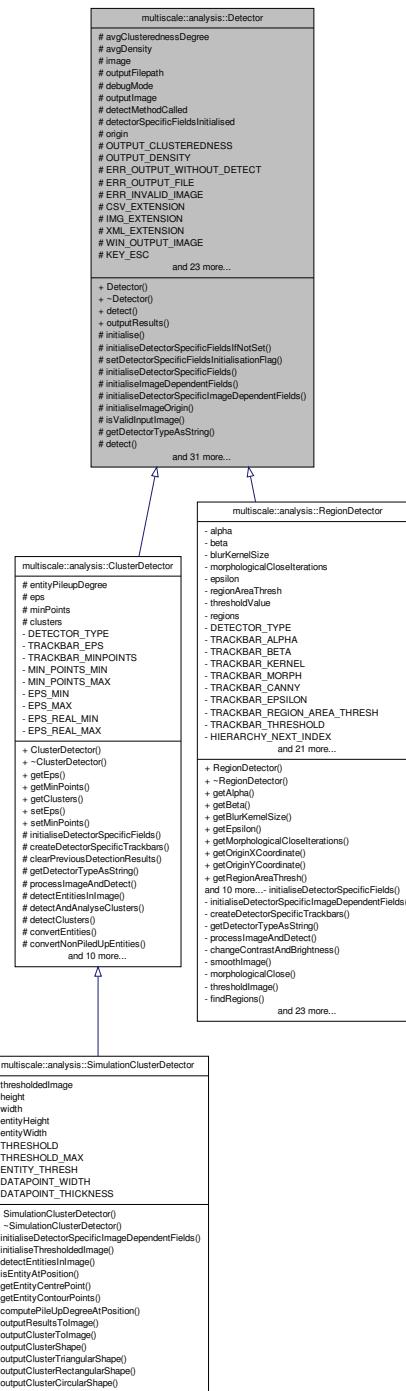
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/[DBSCAN.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[DBSCAN.cpp](#)

7.44 multiscale::analysis::Detector Class Reference

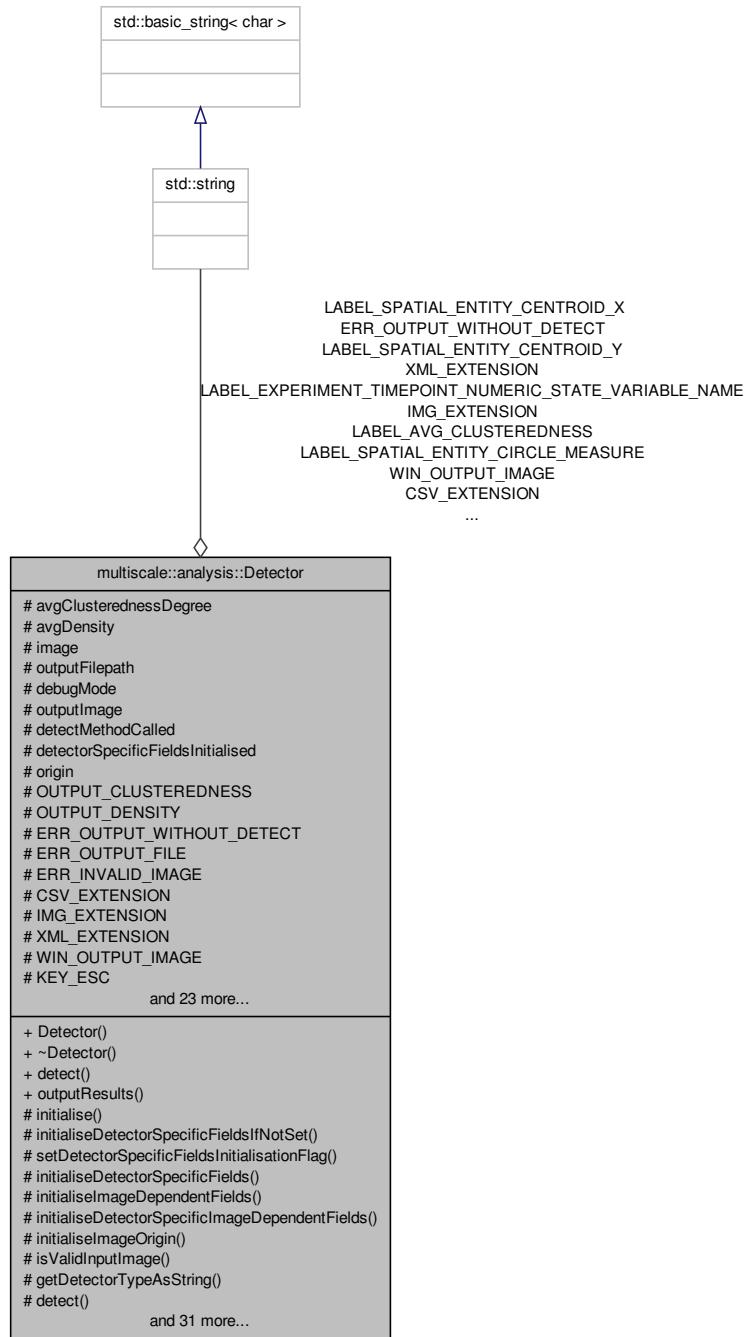
Abstract class for detecting entities of interest in images.

```
#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 Mat &inputImage)`
Run the detection procedure on the given image.
- `void outputResults (const 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 Mat &inputImage)`
Check if the image is valid.
- `virtual string getDetectorTypeAsString ()=0`
Get the type of the employed detector as a 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 vector< Point > &polygon, unsigned int closestPointIndex)`
Compute the angle of the polygon.
- `double polygonAngle (const vector< Point > &polygonConvexHull, const Point &closestPoint)`
Compute the angle of the polygon.
- `void minAreaRectCentre (const vector< Point > &polygon, Point ¢re)`
Get the centre of the minimum area bounding rectangle.

- void **findGoodPointsForAngle** (const vector< Point > &polygonConvexHull, const Point &boundingRectCentre, const Point &closestPoint, vector< Point > &goodPointsForAngle)

Find the points for determining the angle of the polygon.
- void **findGoodIntersectionPoints** (const vector< Point > &polygonConvexHull, const Point &edgePointA, const Point &edgePointB, vector< 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** (ofstream &fout)

Output the results to a file using the provided output file stream.
- void **outputSpatialEntitiesToCsvFile** (ofstream &fout)

Output the pseudo 3D spatial entities to a csv file.
- void **outputAveragedMeasuresToCsvFile** (ofstream &fout)

Output the averaged measures to a csv file.
- void **outputResultsToXMLFile** ()

Output the results to an xml file.
- void **outputResultsToXMLFile** (const 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 string &name, double value)

Add a numeric state variable to the property tree.
- pt::ptree **constructPropertyTree** (SpatialEntityPseudo3D &spatialEntity)

Construct the property tree corresponding to the given pseudo 3D 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 vector< shared_ptr < SpatialEntityPseudo3D > > getCollectionOf-SpatialEntityPseudo3D ()=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 Mat &image, const 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
- Mat image
- string outputPath
- bool debugMode
- Mat outputImage
- bool detectMethodCalled
- bool detectorSpecificFieldsInitialised
- Point origin

Static Protected Attributes

- static const string OUTPUT_CLUSTEREDNESS = "Average clusteredness degree: "
- static const string OUTPUT_DENSITY = "Average density: "
- static const string ERR_OUTPUT_WITHOUT_DETECT = "Unable to output results if the detect method was not called previously."
- static const string ERR_OUTPUT_FILE = "Unable to create output file."
- static const string ERR_INVALID_IMAGE = "The input image is invalid."
- static const string CSV_EXTENSION = ".out"
- static const string IMG_EXTENSION = ".png"

- static const string `XML_EXTENSION` = ".xml"
- static const string `WIN_OUTPUT_IMAGE` = "Output image"
- static const int `KEY_ESC` = 27
- static const int `KEY_SAVE` = 115
- static const string `LABEL_ATTRIBUTE` = "<xmllattr>"
- static const string `LABEL_COMMENT` = "<xmlcomment>"
- static const string `LABEL_COMMENT_CONTENTS` = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."
- static const string `LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE` = "experiment.timepoint.numericStateVariable"
- static const string `LABEL_EXPERIMENT_TIMEPOINT_SPATIAL_ENTITY` = "experiment.timepoint.spatialEntity"
- static const string `LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_NAME` = "name"
- static const string `LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_STATE_VARIABLE_VALUE` = "value"
- static const string `LABEL_SPATIAL_ENTITY_PSEUDO_3D` = "pseudo3D"
- static const string `LABEL_SPATIAL_ENTITY_TYPE` = "type"
- static const string `LABEL_SPATIAL_ENTITY_CLUSTEREDNESS` = "clusteredness"
- static const string `LABEL_SPATIAL_ENTITY_DENSITY` = "density"
- static const string `LABEL_SPATIAL_ENTITY_AREA` = "area"
- static const string `LABEL_SPATIAL_ENTITY_PERIMETER` = "perimeter"
- static const string `LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN` = "distanceFromOrigin"
- static const string `LABEL_SPATIAL_ENTITY_ANGLE` = "angle"
- static const string `LABEL_SPATIAL_ENTITY_SHAPE` = "shape"
- static const string `LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE` = "triangleMeasure"
- static const string `LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE` = "rectangleMeasure"
- static const string `LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE` = "circleMeasure"
- static const string `LABEL_SPATIAL_ENTITY_CENTROID_X` = "centroid.x"
- static const string `LABEL_SPATIAL_ENTITY_CENTROID_Y` = "centroid.y"
- static const string `LABEL_AVG_CLUSTEREDNESS` = "avgClusteredness"
- static const string `LABEL_AVG_DENSITY` = "avgDensity"

7.44.1 Detailed Description

Abstract class for detecting entities of interest in images.

Definition at line 25 of file Detector.hpp.

7.44.2 Constructor & Destructor Documentation

7.44.2.1 **Detector::Detector (bool *debugMode* = false)**

Definition at line 12 of file Detector.cpp.

7.44.2.2 **Detector::~Detector () [virtual]**

Definition at line 22 of file Detector.cpp.

7.44.3 Member Function Documentation

7.44.3.1 **void Detector::addAverageMeasuresToPropertyTree (pt::ptree & *propertyTree*) [protected]**

Add the average clusteredness and average density to the property tree.

Parameters

<i>propertyTree</i>	The property tree
---------------------	-------------------

Definition at line 245 of file Detector.cpp.

7.44.3.2 **void Detector::addNumericStateVariableToPropertyTree (pt::ptree & *propertyTree*, const string & *name*, double *value*) [protected]**

Add a numeric state variable to the property tree.

Parameters

<i>propertyTree</i>	The property tree
<i>name</i>	The name of the numeric state variable
<i>value</i>	The value of the numeric state variable

Definition at line 254 of file Detector.cpp.

7.44.3.3 **void Detector::addSpatialEntitiesToPropertyTree (pt::ptree & *propertyTree*) [protected]**

Add the pseudo 3D spatial entities to the property tree.

Parameters

<i>propertyTree</i>	The property tree
---------------------	-------------------

Definition at line 235 of file Detector.cpp.

7.44.3.4 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 276 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::getShapeAsString(), and multiscale::analysis::SpatialEntityPseudo3D::getTriangularMeasure().

7.44.3.5 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 291 of file Detector.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::typeAsString().

7.44.3.6 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](#).

7.44.3.7 pt::ptree Detector::constructPropertyTree (SpatialEntityPseudo3D & *spatialEntity*) [protected]

Construct the property tree corresponding to the given pseudo 3D spatial entity.

Parameters

<i>spatialEntity</i>	The spatial entity to be converted
----------------------	------------------------------------

Definition at line 264 of file Detector.cpp.

**7.44.3.8 virtual void multiscale::analysis::Detector::create-
DetectorSpecificTrackbars() [protected, pure
virtual]**

Create the trackbars specific to the used detector.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::Cluster-
Detector](#).

7.44.3.9 void Detector::createTrackbars() [protected]

Create the trackbars which allow the user to change the values of the parameters.

Definition at line 299 of file Detector.cpp.

7.44.3.10 void Detector::createTrackbarsWindow() [protected]

Create the window in which the trackbars are placed.

Definition at line 304 of file Detector.cpp.

7.44.3.11 void Detector::detect(const Mat & *inputImage*)

Run the detection procedure on the given image.

Parameters

<i>inputImage</i>	The input image
-------------------	-----------------

Definition at line 27 of file Detector.cpp.

References [MS_throw](#).

Referenced by [main\(\)](#).

7.44.3.12 void Detector::detect() [protected]

Run the detection procedure.

Definition at line 81 of file Detector.cpp.

7.44.3.13 void Detector::detectInDebugMode() [protected]

Run the detection procedure when in debug mode.

Definition at line 91 of file Detector.cpp.

References KEY_ESC.

7.44.3.14 void Detector::detectInReleaseMode() [protected]

Run the detection procedure when in release mode (i.e. non-debug mode)

Definition at line 106 of file Detector.cpp.

7.44.3.15 void Detector::displayImage(const Mat & image, const string & windowName) [protected]

Display an image in a particular window.

Parameters

<i>image</i>	The image
<i>window- Name</i>	The name of the window

Definition at line 315 of file Detector.cpp.

7.44.3.16 void Detector::displayResultsInWindow() [protected]

Display the results in a window.

Definition at line 161 of file Detector.cpp.

7.44.3.17 void Detector::findGoodIntersectionPoints(const vector< Point > & polygonConvexHull, const Point & edgePointA, const Point & edgePointB, vector< Point > & goodPointsForAngle) [protected]

Find good intersection points for computing the angle of the polygon.

Parameters

<i>polygon- ConvexHull</i>	The convex hull of the polygon
<i>edgePointA</i>	Point A on the edge
<i>edgePointB</i>	Point B on the edge
<i>goodPoints- ForAngle</i>	The "good" points for computing the angle

Definition at line 148 of file Detector.cpp.

References multiscale::Geometry2D::lineSegmentIntersection().

```
7.44.3.18 void Detector::findGoodPointsForAngle ( const vector< Point > &
    polygonConvexHull, const Point & boundingRectCentre, const Point & closestPoint,
    vector< Point > & goodPointsForAngle ) [protected]
```

Find the points for determining the angle of the polygon.

Parameters

<i>polygon-ConvexHull</i>	Convex hull of polygon
<i>bounding-RectCentre</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>goodPoints-ForAngle</i>	The points which are relevant for computing the angle

Definition at line 136 of file Detector.cpp.

References multiscale::Geometry2D::orthogonalLineToAnotherLineEdgePoints().

```
7.44.3.19 virtual vector<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](#).

```
7.44.3.20 virtual string multiscale::analysis::Detector::getDetectorTypeAsString ( )
[protected, pure virtual]
```

Get the type of the employed detector as a string.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

```
7.44.3.21 void Detector::initialise ( ) [protected]
```

Initialisation function for the class.

Definition at line 48 of file Detector.cpp.

7.44.3.22 **virtual void multiscale::analysis::Detector::initialiseDetectorSpecificFields()** [protected, pure virtual]

Initialisation of the detector specific values.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::ClusterDetector](#).

7.44.3.23 **void Detector::initialiseDetectorSpecificFieldsIfNotSet()** [protected]

Initialisation of the detector specific values in case they were not set.

Definition at line 53 of file [Detector.cpp](#).

7.44.3.24 **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](#).

7.44.3.25 **void Detector::initialiseImageDependentFields()** [protected]

Initialisation of the image dependent values.

Definition at line 65 of file [Detector.cpp](#).

7.44.3.26 **void Detector::initialiseImageOrigin()** [protected]

Definition at line 70 of file [Detector.cpp](#).

7.44.3.27 **bool Detector::isValidInputImage(const 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 77 of file [Detector.cpp](#).

7.44.3.28 void Detector::minAreaRectCentre (const vector< Point > & *polygon*, 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 130 of file Detector.cpp.

7.44.3.29 void Detector::outputAveragedMeasuresToCsvFile (ofstream & *fout*) [protected]

Output the averaged measures to a csv file.

Parameters

<i>fout</i>	Output file stream
-------------	--------------------

Definition at line 212 of file Detector.cpp.

7.44.3.30 void Detector::outputResults (const string & *outputfilepath*)

Output the results to the given file.

Parameters

<i>output-Filepath</i>	Path to the output file
------------------------	-------------------------

Definition at line 38 of file Detector.cpp.

Referenced by main().

7.44.3.31 void Detector::outputResultsToCsvFile () [protected]

Output the results to a csv file.

Definition at line 180 of file Detector.cpp.

References MS_throw.

7.44.3.32 void Detector::outputResultsToCsvFile (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 192 of file Detector.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::fieldNamesToString().

7.44.3.33 void Detector::outputResultsToFile() [protected]

Output the results to file(s)

Definition at line 166 of file Detector.cpp.

7.44.3.34 virtual void multiscale::analysis::Detector::outputResultsToImage() [protected, pure virtual]

Output the results to an image.

Implemented in [multiscale::analysis::RegionDetector](#), and [multiscale::analysis::SimulationClusterDetector](#).

7.44.3.35 void Detector::outputResultsToXMLFile() [protected]

Output the results to an xml file.

Definition at line 217 of file Detector.cpp.

7.44.3.36 void Detector::outputResultsToXMLFile(const string & *filepath*) [protected]

Output the clusters and averaged measures to an xml file.

Parameters

<i>filepath</i>	Output file path
-----------------	------------------

Definition at line 221 of file Detector.cpp.

7.44.3.37 void Detector::outputSpatialEntitiesToCsvFile(ostream & *fout*) [protected]

Output the pseudo 3D spatial entities to a csv file.

Parameters

<i>fout</i>	Output file stream
-------------	--------------------

Definition at line 204 of file Detector.cpp.

7.44.3.38 double Detector::polygonAngle (const vector< 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 111 of file Detector.cpp.

Referenced by multiscale::analysis::RegionDetector::createRegionFromPolygon(), and multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

7.44.3.39 double Detector::polygonAngle (const vector< Point > & *polygonConvexHull*, const 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

<i>polygon-ConvexHull</i>	Convex hull of polygon
<i>closestPoint</i>	Closest point to the origin from the set of points defining the polygon

Definition at line 119 of file Detector.cpp.

References multiscale::Geometry2D::angleBtwPoints().

7.44.3.40 void Detector::printOutputErrorMessage () [protected]

Print error message, because the detect method was not called before calling the output method.

Definition at line 320 of file Detector.cpp.

7.44.3.41 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](#).

7.44.3.42 void Detector::processPressedKeyRequest (char & pressedKey)
[protected]

Process the request of the user by pressing the key.

Parameters

<code>pressedKey</code>	Key pressed by the user, if a key was pressed, or "-1", otherwise
-------------------------	---

Definition at line 309 of file Detector.cpp.

7.44.3.43 void Detector::setDetectorSpecificFieldsInitialisationFlag (bool flag = true) [protected]

Set the detector specific fields initialisation flag to true.

Definition at line 61 of file Detector.cpp.

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\(\)](#).

7.44.3.44 void Detector::storeOutputImageOnDisk() [protected]

Store the image with the output results on disk.

Definition at line 174 of file Detector.cpp.

7.44.4 Member Data Documentation

7.44.4.1 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 29 of file Detector.hpp.

Referenced by multiscale::analysis::ClusterDetector::analyseClusters(), multiscale::analysis::ClusterDetector::ClusterDetector(), multiscale::analysis::RegionDetector::computeAverageClusterednessDegree(), multiscale::analysis::RegionDetector::RegionDetector(), and multiscale::analysis::RegionDetector::sumOfAverageCentroidDistances().

7.44.4.2 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 34 of file Detector.hpp.

Referenced by multiscale::analysis::ClusterDetector::analyseClusters(), multiscale::analysis::ClusterDetector::ClusterDetector(), multiscale::analysis::RegionDetector::computeAverageDensity(), and multiscale::analysis::RegionDetector::RegionDetector().

7.44.4.3 const string Detector::CSV_EXTENSION = ".out" [static, protected]

Definition at line 291 of file Detector.hpp.

7.44.4.4 bool multiscale::analysis::Detector::debugMode [protected]

Flag for indicating if debug mode is set

Definition at line 42 of file Detector.hpp.

7.44.4.5 bool multiscale::analysis::Detector::detectMethodCalled [protected]

Flag for indicating if the detect method was called

Definition at line 46 of file Detector.hpp.

7.44.4.6 bool multiscale::analysis::Detector::detectorSpecificFieldsInitialised [protected]

Flag for indicating if the parameters were

Definition at line 47 of file Detector.hpp.

7.44.4.7 **const string Detector::ERR_INVALID_IMAGE = "The input image is invalid."**
[static, protected]

Definition at line 289 of file Detector.hpp.

7.44.4.8 **const string Detector::ERR_OUTPUT_FILE = "Unable to create output file."**
[static, protected]

Definition at line 288 of file Detector.hpp.

7.44.4.9 **const string Detector::ERR_OUTPUT_WITHOUT_DETECT = "Unable to output results if the detect method was not called previously."** [static, protected]

Definition at line 287 of file Detector.hpp.

7.44.4.10 **Mat multiscale::analysis::Detector::image** [protected]

Input image

Definition at line 40 of file Detector.hpp.

Referenced by multiscale::analysis::RegionDetector::changeContrastAndBrightness(), multiscale::analysis::SimulationClusterDetector::computePileUpDegreeAtPosition(), multiscale::analysis::SimulationClusterDetector::initialiseDetectorSpecificImage-DependentFields(), multiscale::analysis::SimulationClusterDetector::initialiseThresholdedImage(), multiscale::analysis::SimulationClusterDetector::outputResultsToImage(), multiscale::analysis::RegionDetector::outputResultsToImage(), and multiscale-::analysis::RegionDetector::regionDensity().

7.44.4.11 **const string Detector::IMG_EXTENSION = ".png"** [static, protected]

Definition at line 292 of file Detector.hpp.

7.44.4.12 **const int Detector::KEY_ESC = 27** [static, protected]

Definition at line 297 of file Detector.hpp.

7.44.4.13 **const int Detector::KEY_SAVE = 115** [static, protected]

Definition at line 298 of file Detector.hpp.

```
7.44.4.14 const string Detector::LABEL_ATTRIBUTE = "<xmattr>" [static,  
protected]
```

Definition at line 300 of file Detector.hpp.

```
7.44.4.15 const string Detector::LABEL_AVG_CLUSTEREDNESS = "avgClusteredness"  
[static, protected]
```

Definition at line 327 of file Detector.hpp.

```
7.44.4.16 const string Detector::LABEL_AVG_DENSITY = "avgDensity" [static,  
protected]
```

Definition at line 328 of file Detector.hpp.

```
7.44.4.17 const string Detector::LABEL_COMMENT = "<xmlcomment>" [static,  
protected]
```

Definition at line 301 of file Detector.hpp.

```
7.44.4.18 const 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 303 of file Detector.hpp.

```
7.44.4.19 const string Detector::LABEL_EXPERIMENT_TIMEPOINT_NUMERIC_ST-  
ATE_VARIABLE = "experiment.timepoint.numericStateVariable" [static,  
protected]
```

Definition at line 305 of file Detector.hpp.

```
7.44.4.20 const string Detector::LABEL_EXPERIMENT_TIMEPOINT_N-  
UMERIC_STATE_VARIABLE_NAME = "name" [static,  
protected]
```

Definition at line 308 of file Detector.hpp.

```
7.44.4.21 const string Detector::LABEL_EXPERIMENT_TIMEPOINT_N-  
UMERIC_STATE_VARIABLE_VALUE = "value" [static,  
protected]
```

Definition at line 309 of file Detector.hpp.

```
7.44.4.22 const string Detector::LABEL_EXPERIMENT_TIMEPOINT_SP-
ATIAL_ENTITY = "experiment.timepoint.spatialEntity" [static,
protected]
```

Definition at line 306 of file Detector.hpp.

```
7.44.4.23 const string Detector::LABEL_SPATIAL_ENTITY_ANGLE = "angle"
[static, protected]
```

Definition at line 319 of file Detector.hpp.

```
7.44.4.24 const string Detector::LABEL_SPATIAL_ENTITY_AREA = "area"
[static, protected]
```

Definition at line 316 of file Detector.hpp.

```
7.44.4.25 const string Detector::LABEL_SPATIAL_ENTITY_CENTROID_X =
"centroid.x" [static, protected]
```

Definition at line 324 of file Detector.hpp.

```
7.44.4.26 const string Detector::LABEL_SPATIAL_ENTITY_CENTROID_Y =
"centroid.y" [static, protected]
```

Definition at line 325 of file Detector.hpp.

```
7.44.4.27 const string Detector::LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE =
"circleMeasure" [static, protected]
```

Definition at line 323 of file Detector.hpp.

```
7.44.4.28 const string Detector::LABEL_SPATIAL_ENTITY_CLUSTEREDNESS =
"clusteredness" [static, protected]
```

Definition at line 314 of file Detector.hpp.

```
7.44.4.29 const string Detector::LABEL_SPATIAL_ENTITY_DENSITY = "density"
[static, protected]
```

Definition at line 315 of file Detector.hpp.

```
7.44.4.30 const string Detector::LABEL_SPATIAL_ENTITY_DISTA-
NCE_FROM_ORIGIN = "distanceFromOrigin" [static,
protected]
```

Definition at line 318 of file Detector.hpp.

```
7.44.4.31 const string Detector::LABEL_SPATIAL_ENTITY_PERIMETER = "perimeter"
[static, protected]
```

Definition at line 317 of file Detector.hpp.

```
7.44.4.32 const string Detector::LABEL_SPATIAL_ENTITY_PSEUDO_3D =
"pseudo3D" [static, protected]
```

Definition at line 311 of file Detector.hpp.

```
7.44.4.33 const string Detector::LABEL_SPATIAL_ENTITY_REC-
TANGLE_MEASURE = "rectangleMeasure" [static,
protected]
```

Definition at line 322 of file Detector.hpp.

```
7.44.4.34 const string Detector::LABEL_SPATIAL_ENTITY_SHAPE = "shape"
[static, protected]
```

Definition at line 320 of file Detector.hpp.

```
7.44.4.35 const string Detector::LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE =
"triangleMeasure" [static, protected]
```

Definition at line 321 of file Detector.hpp.

```
7.44.4.36 const string Detector::LABEL_SPATIAL_ENTITY_TYPE = "type"
[static, protected]
```

Definition at line 313 of file Detector.hpp.

```
7.44.4.37 Point multiscale::analysis::Detector::origin [protected]
```

The point representing the origin

Definition at line 49 of file Detector.hpp.

Referenced by multiscale::analysis::RegionDetector::createRegionFromPolygon(), multiscale::analysis::RegionDetector::getOriginXCoordinate(), multiscale::analysis::RegionDetector::getOriginYCoordinate(), multiscale::analysis::RegionDetector::setOriginXCoordinate(), multiscale::analysis::RegionDetector::setOriginYCoordinate(), and multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

7.44.4.38 const string Detector::OUTPUT_CLUSTEREDNESS = "Average clusteredness degree:" [static, protected]

Definition at line 284 of file Detector.hpp.

7.44.4.39 const string Detector::OUTPUT_DENSITY = "Average density:" [static, protected]

Definition at line 285 of file Detector.hpp.

7.44.4.40 string multiscale::analysis::Detector::outputFilepath [protected]

Path of the output file

Definition at line 41 of file Detector.hpp.

7.44.4.41 Mat multiscale::analysis::Detector::outputImage [protected]

Image for displaying the results

Definition at line 44 of file Detector.hpp.

Referenced by multiscale::analysis::RegionDetector::outputRegionToImage(), multiscale::analysis::SimulationClusterDetector::outputResultsToImage(), and multiscale::analysis::RegionDetector::outputResultsToImage().

7.44.4.42 const string Detector::WIN_OUTPUT_IMAGE = "Output image" [static, protected]

Definition at line 295 of file Detector.hpp.

Referenced by multiscale::analysis::ClusterDetector::createDetectorSpecificTrackbars(), and multiscale::analysis::RegionDetector::createDetectorSpecificTrackbars().

7.44.4.43 const string Detector::XML_EXTENSION = ".xml" [static, protected]

Definition at line 293 of file Detector.hpp.

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

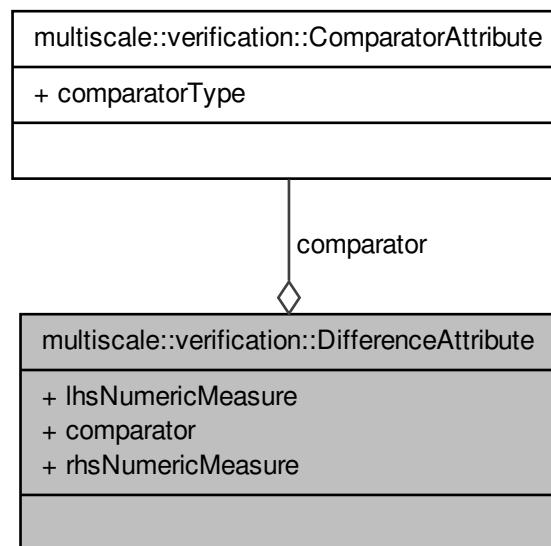
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/-
Detector.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-
Detector.cpp

7.45 multiscale::verification::DifferenceAttribute Class Reference

Class for representing a difference attribute.

```
#include <DifferenceAttribute.hpp>
```

Collaboration diagram for multiscale::verification::DifferenceAttribute:



Public Attributes

- NumericMeasureAttributeType lhsNumericMeasure
- ComparatorAttribute comparator
- NumericMeasureAttributeType rhsNumericMeasure

7.45.1 Detailed Description

Class for representing a difference attribute.

Definition at line 16 of file DifferenceAttribute.hpp.

7.45.2 Member Data Documentation

7.45.2.1 ComparatorAttribute multiscale::verification::DifferenceAttribute- ::comparator

The comparator

Definition at line 21 of file DifferenceAttribute.hpp.

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

7.45.2.2 NumericMeasureAttributeType multiscale::verification::Difference- Attribute::lhsNumericMeasure

The left hand side numeric measure

Definition at line 20 of file DifferenceAttribute.hpp.

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

7.45.2.3 NumericMeasureAttributeType multiscale::verification::Difference- Attribute::rhsNumericMeasure

The right hand side numeric measure

Definition at line 22 of file DifferenceAttribute.hpp.

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

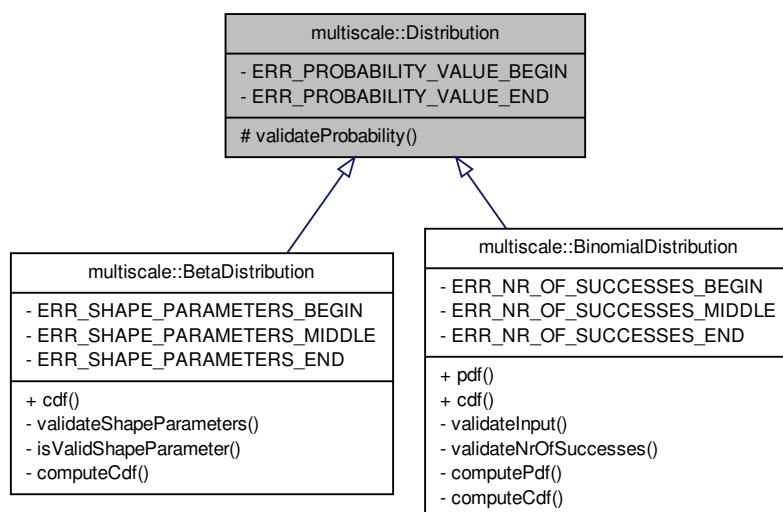
The documentation for this class was generated from the following file:

- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[Difference-Attribute.hpp](#)

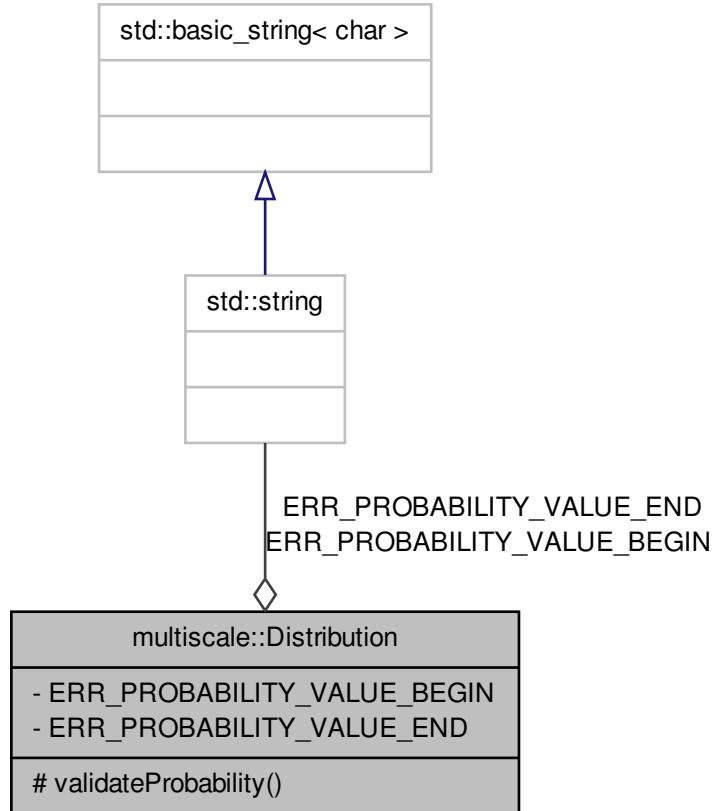
7.46 multiscale::Distribution Class Reference

```
#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."

7.46.1 Detailed Description

Definition at line 10 of file Distribution.hpp.

7.46.2 Member Function Documentation

7.46.2.1 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().

7.46.3 Member Data Documentation

7.46.3.1 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().

7.46.3.2 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](#)

7.47 multiscale::DivisionOperation Class Reference

Functor representing a division operation.

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

Public Member Functions

- template<typename Operand >
 Operand **operator()** (Operand operand1, Operand operand2) const
Divide the two operands.

7.47.1 Detailed Description

Functor representing a division operation.

Definition at line 34 of file Numeric.hpp.

7.47.2 Member Function Documentation

- 7.47.2.1 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 44 of file Numeric.hpp.

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

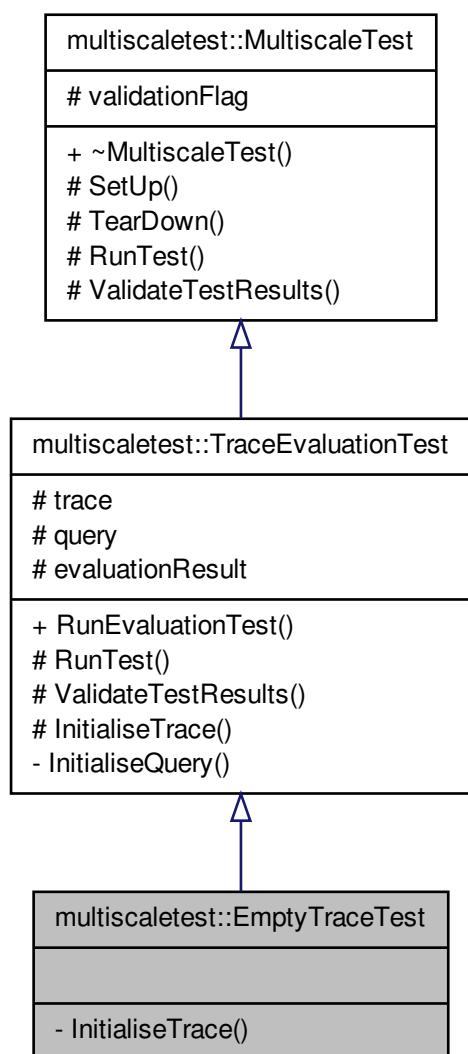
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
[Numeric.hpp](#)

7.48 multiscaletest::EmptyTraceTest Class Reference

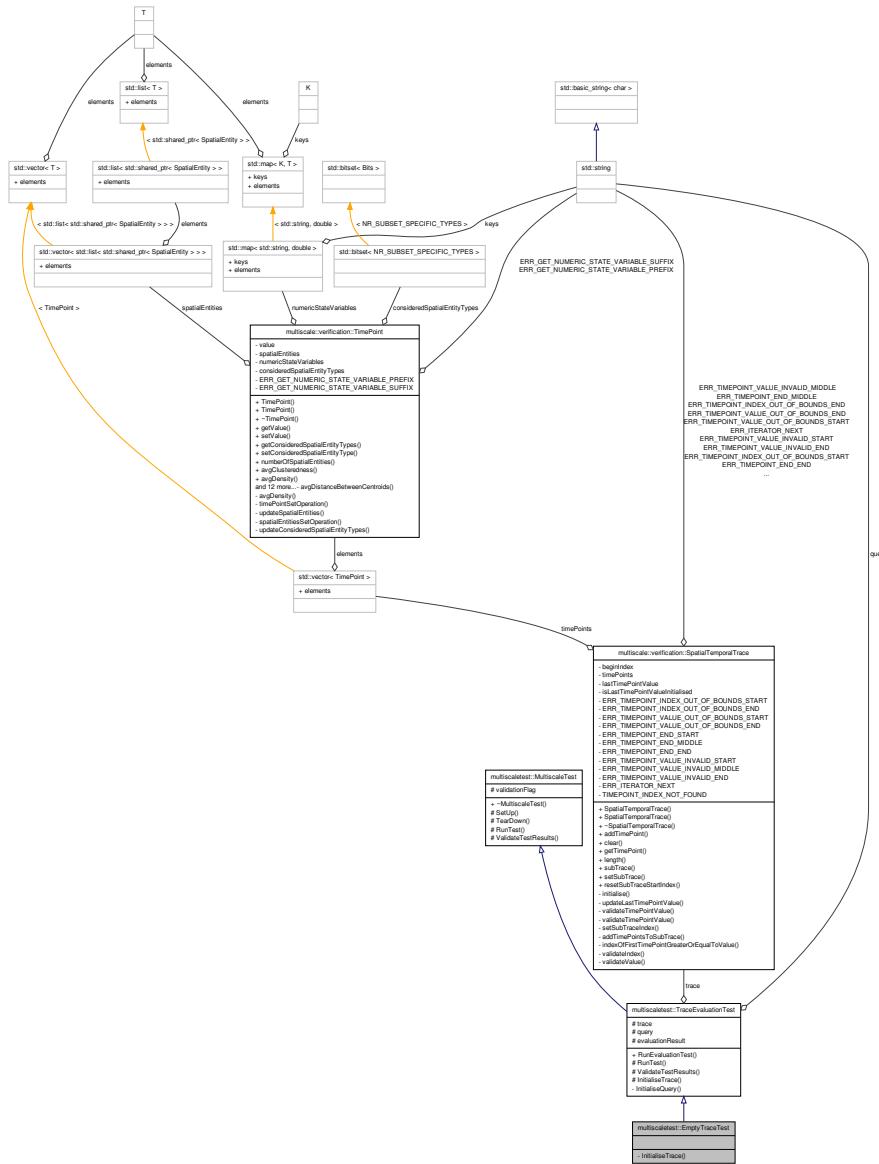
Class for testing evaluation of empty traces.

```
#include <EmptyTraceTest.hpp>
```

Inheritance diagram for multiscaletest::EmptyTraceTest:



Collaboration diagram for multiscaletest::EmptyTraceTest



Private Member Functions

- virtual void `InitialiseTrace()` override

Initialise the trace.

7.48.1 Detailed Description

Class for testing evaluation of empty traces.

Definition at line 13 of file `EmptyTraceTest.hpp`.

7.48.2 Member Function Documentation

7.48.2.1 void multiscaletest::EmptyTraceTest::InitialiseTrace() [override, private, virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

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

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

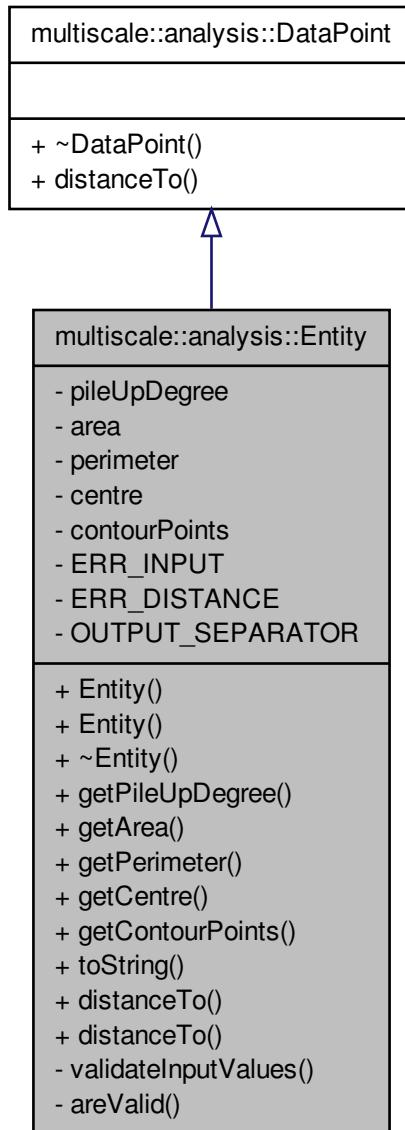
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[EmptyTraceTest.hpp](#)

7.49 multiscale::analysis::Entity Class Reference

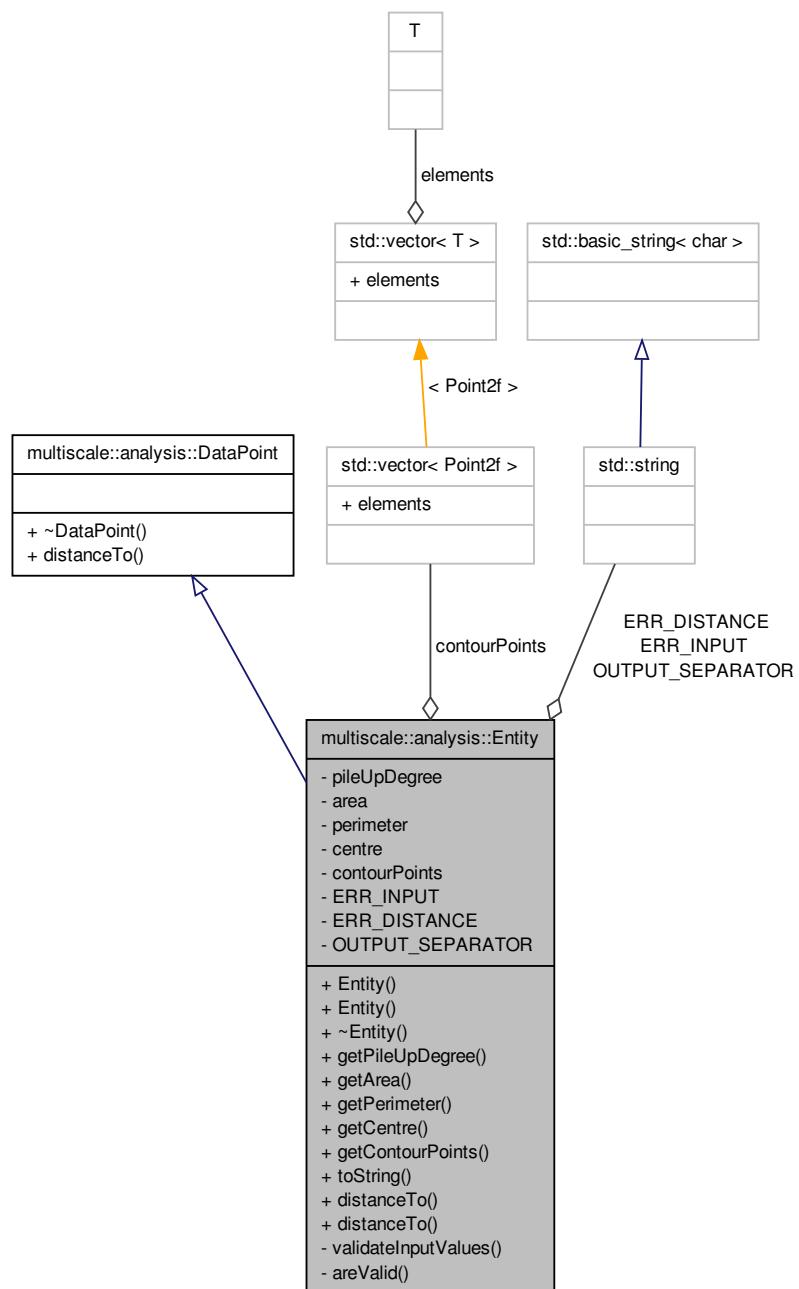
Class for representing an entity in an image (e.g. cell, organism etc.)

```
#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 `Point2f ¢re`, const `vector< Point2f > &contourPoints`)
- `Entity` (const `Entity` &`entity`)
- `~Entity` ()
- unsigned int `getPileUpDegree` () const
`Get the degree of pile up.`
- double `getArea` () const
`Get the area.`
- double `getPerimeter` () const
`Get the perimeter.`
- `Point2f getCentre` () const
`Get the point defining the centre of the entity.`
- `vector< Point2f > getContourPoints` () const
`Get the set of points defining the contour of the entity.`
- string `toString` ()
`Get a string representation of all the field values.`
- double `distanceTo` (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 `Point2f ¢re`, const `vector< Point2f > &contourPoints`)
- bool `isValid` (unsigned int `pileUpDegree`, double `area`, double `perimeter`, const `Point2f ¢re`, const `vector< 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`
- `Point2f centre`
- `vector< Point2f > contourPoints`

Static Private Attributes

- static const string `ERR_INPUT` = "Invalid input parameters were provided to the constructor."
- static const string `ERR_DISTANCE` = "The distance to an object of a different type cannot be computed."
- static const string `OUTPUT_SEPARATOR` = ","

7.49.1 Detailed Description

Class for representing an entity in an image (e.g. cell, organism etc.)

Definition at line 19 of file Entity.hpp.

7.49.2 Constructor & Destructor Documentation

7.49.2.1 Entity::Entity (*unsigned int pileUpDegree, double area, double perimeter, const Point2f & centre, const vector< Point2f > & contourPoints*)

Definition at line 9 of file Entity.cpp.

References area, centre, contourPoints, perimeter, pileUpDegree, and validateInputValues().

7.49.2.2 Entity::Entity (*const Entity & entity*)

Definition at line 20 of file Entity.cpp.

References area, centre, contourPoints, perimeter, pileUpDegree, and validateInputValues().

7.49.2.3 Entity::~Entity ()

Definition at line 30 of file Entity.cpp.

7.49.3 Member Function Documentation

7.49.3.1 bool Entity::isValid (*unsigned int pileUpDegree, double area, double perimeter, const Point2f & centre, const vector< Point2f > & contourPoints*) [private]

Check if the provided degree of pile up, area, centre and contour points are valid.

Parameters

<i>pileUp-Degree</i>	Degree of pile up
<i>area</i>	Area
<i>perimeter</i>	Perimeter
<i>centre</i>	Centre of the entity
<i>contour-Points</i>	Points defining the contour of the entity

Definition at line 75 of file Entity.cpp.

References multiscale::Numeric::greaterOrEqual().

Referenced by validateInputValues().

7.49.3.2 `double Entity::distanceTo (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\(\)](#).

7.49.3.3 `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\(\)](#).

7.49.3.4 `double Entity::getArea () const`

Get the area.

Definition at line 36 of file Entity.cpp.

References area.

7.49.3.5 `Point2f Entity::getCentre () const`

Get the point defining the centre of the entity.

Definition at line 44 of file Entity.cpp.

References centre.

7.49.3.6 `vector< 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.

7.49.3.7 `double Entity::getPerimeter () const`

Get the perimeter.

Definition at line 40 of file Entity.cpp.

References perimeter.

7.49.3.8 `unsigned int Entity::getPileUpDegree() const`

Get the degree of pile up.

Definition at line 32 of file Entity.cpp.

References pileUpDegree.

7.49.3.9 `string Entity::toString()`

Get a string representation of all the field values.

Definition at line 52 of file Entity.cpp.

References centre, OUTPUT_SEPARATOR, and pileUpDegree.

7.49.3.10 `void Entity::validateInputValues(unsigned int pileUpDegree, double area, double perimeter, const Point2f & centre, const vector< Point2f > & contourPoints) [private]`

Parameters

<i>pileUp-Degree</i>	Degree of pile up
<i>area</i>	Area
<i>perimeter</i>	Perimeter
<i>centre</i>	Centre of the entity
<i>contour-Points</i>	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().

7.49.4 Member Data Documentation

7.49.4.1 `double multiscale::analysis::Entity::area [private]`

Area of the entity

Definition at line 24 of file Entity.hpp.

Referenced by Entity(), and getArea().

7.49.4.2 `Point2f multiscale::analysis::Entity::centre [private]`

Point defining the centre of the entity

Definition at line 27 of file Entity.hpp.

Referenced by distanceTo(), Entity(), getCentre(), and toString().

7.49.4.3 `vector<Point2f> multiscale::analysis::Entity::contourPoints`
[private]

Set of points defining the contour of the entity

Definition at line 28 of file Entity.hpp.

Referenced by Entity(), and getContourPoints().

7.49.4.4 `const string Entity::ERR_DISTANCE = "The distance to an object of a different type cannot be computed."` [static, private]

Definition at line 89 of file Entity.hpp.

7.49.4.5 `const string Entity::ERR_INPUT = "Invalid input parameters were provided to the constructor."` [static, private]

Definition at line 88 of file Entity.hpp.

Referenced by validateInputValues().

7.49.4.6 `const string Entity::OUTPUT_SEPARATOR = ","` [static, private]

Definition at line 91 of file Entity.hpp.

Referenced by toString().

7.49.4.7 `double multiscale::analysis::Entity::perimeter` [private]

Perimeter of the entity

Definition at line 25 of file Entity.hpp.

Referenced by Entity(), and getPerimeter().

7.49.4.8 `unsigned int multiscale::analysis::Entity::pileUpDegree` [private]

Degree of pile up (relevant only if entities can pile up onto each other)

Definition at line 23 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/sp Entity.hpp

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/[Entity.cpp](#)

7.50 multiscale::verification::EquivalenceConstraintAttribute Class Reference

Class for representing an "equivalence" constraint attribute.

```
#include <EquivalenceConstraintAttribute.hpp>
```

Public Attributes

- [ConstraintAttributeType constraint](#)

7.50.1 Detailed Description

Class for representing an "equivalence" constraint attribute.

Definition at line 14 of file EquivalenceConstraintAttribute.hpp.

7.50.2 Member Data Documentation

7.50.2.1 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](#)

7.51 multiscale::verification::EquivalenceLogicPropertyAttribute Class Reference

Class for representing an "equivalence" logic property attribute.

```
#include <EquivalenceLogicPropertyAttribute.hpp>
```

Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

7.51.1 Detailed Description

Class for representing an "equivalence" logic property attribute.

Definition at line 14 of file EquivalenceLogicPropertyAttribute.hpp.

7.51.2 Member Data Documentation

7.51.2.1 LogicPropertyAttributeType multiscale::verification::EquivalenceLogicPropertyAttribute::logicProperty

The logical property following the "equivalence" operator

Definition at line 18 of file EquivalenceLogicPropertyAttribute.hpp.

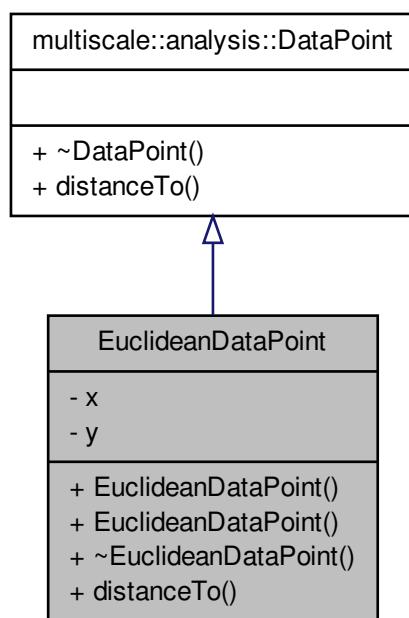
Referenced by multiscale::verification::LogicPropertyVisitor::operator()().

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

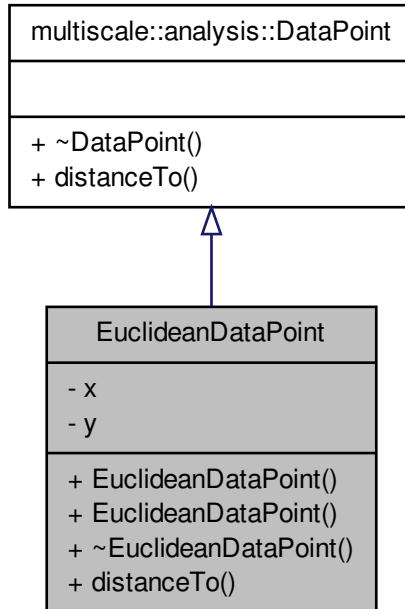
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[EquivalenceLogicPropertyAttribute.hpp](#)

7.52 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 \(shared_ptr< DataPoint > point\) override](#)

Compute the distance between this data point and another one.

Private Attributes

- [double x](#)
- [double y](#)

7.52.1 Detailed Description

Definition at line 16 of file DBSCANTest.cpp.

7.52.2 Constructor & Destructor Documentation

7.52.2.1 **EuclideanDataPoint::EuclideanDataPoint (double x, double y)**
[inline]

Definition at line 23 of file DBSCANTest.cpp.

7.52.2.2 **EuclideanDataPoint::EuclideanDataPoint (const EuclideanDataPoint & point)** [inline]

Definition at line 24 of file DBSCANTest.cpp.

7.52.2.3 **EuclideanDataPoint::~EuclideanDataPoint ()** [inline]

Definition at line 25 of file DBSCANTest.cpp.

7.52.3 Member Function Documentation

7.52.3.1 **double EuclideanDataPoint::distanceTo (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 27 of file DBSCANTest.cpp.

7.52.4 Member Data Documentation

7.52.4.1 **double EuclideanDataPoint::x** [private]

Definition at line 19 of file DBSCANTest.cpp.

7.52.4.2 **double EuclideanDataPoint::y** [private]

Definition at line 20 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](#)

7.53 multiscale::ExceptionHandler Class Reference

Exception handler class.

```
#include <ExceptionHandler.hpp>
```

Static Public Member Functions

- static void [printErrorMessage \(const exception &ex\)](#)
Print the error message.

7.53.1 Detailed Description

Exception handler class.

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

7.53.2 Member Function Documentation

7.53.2.1 static void multiscale::ExceptionHandler::printErrorMessage (const exception & ex) [inline, static]

Print the error message.

The error message is printed using the `ex.what()` method

Parameters

<code>ex</code>	Exception
-----------------	-----------

Definition at line 24 of file `ExceptionHandler.hpp`.

References `multiscale::ERR_MSG`.

Referenced by `multiscale::OperatingSystem::executeProgram()`, `main()`, `multiscale::verification::ModelCheckingManager::parseLogicProperty()`, `printParsingResult()`, `readQueriesFromFile()`, and `readValidXmlFilesFromFolder()`.

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

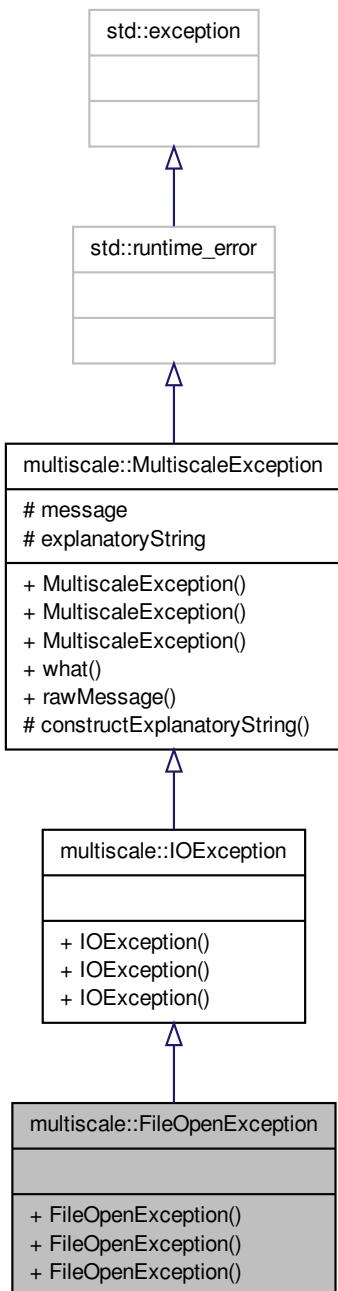
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/-ExceptionHandler.hpp](#)

7.54 multiscale::FileOpenException Class Reference

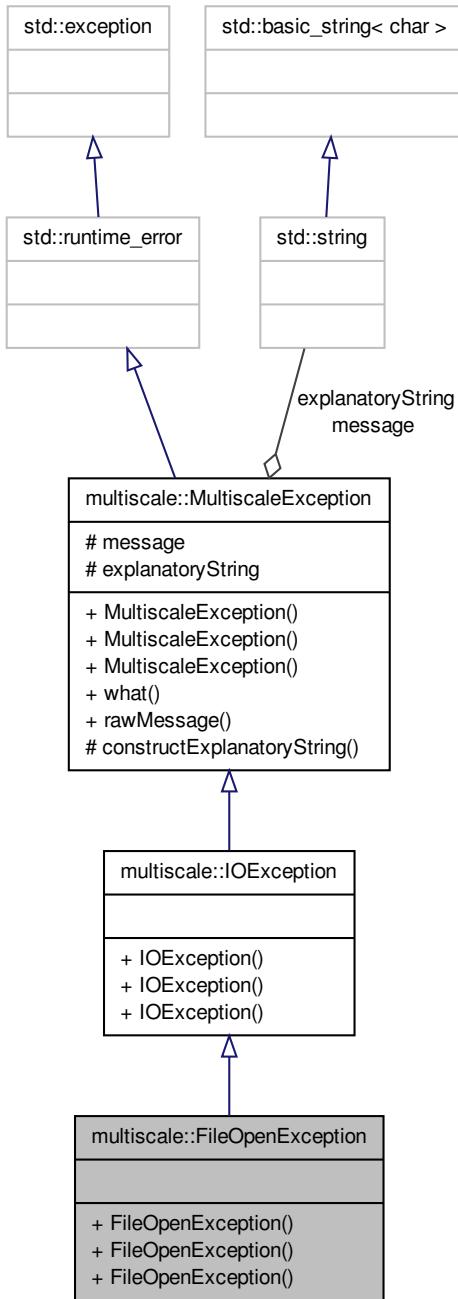
Class for representing exceptions when opening a file.

```
#include <FileOpenException.hpp>
```

Inheritance diagram for multiscale::FileOpenException:



Collaboration diagram for multiscale::FileOpenException:



Public Member Functions

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

7.54.1 Detailed Description

Class for representing exceptions when opening a file.

Definition at line 14 of file FileOpenException.hpp.

7.54.2 Constructor & Destructor Documentation

7.54.2.1 **multiscale::FileOpenException::FileOpenException() [inline]**

Definition at line 18 of file FileOpenException.hpp.

7.54.2.2 **multiscale::FileOpenException::FileOpenException(const string & file, int line, const string & msg) [inline, explicit]**

Definition at line 20 of file FileOpenException.hpp.

7.54.2.3 **multiscale::FileOpenException::FileOpenException(const string & file, int line, const char * msg) [inline, explicit]**

Definition at line 24 of file FileOpenException.hpp.

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

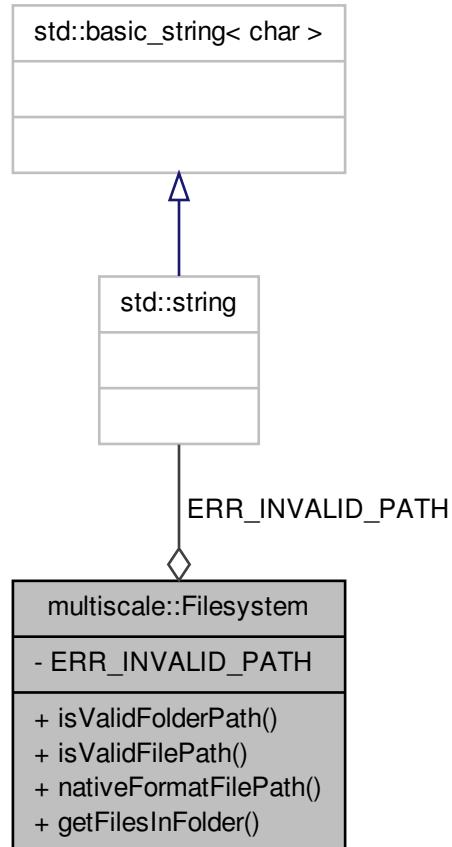
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-[FileOpenException.hpp](#)

7.55 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 `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 Attributes

- static const std::string **ERR_INVALID_PATH** = "The given input file path is invalid.
Please change."

7.55.1 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.

7.55.2 Member Function Documentation

7.55.2.1 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.

Parameters

<i>folderPath</i>	The path to the folder
<i>extension</i>	The given extension

Definition at line 37 of file Filesystem.cpp.

References isValidFolderPath().

7.55.2.2 bool **Filesystem::isValidFilePath** (const std::string & *path*) [static]

Check if the given path is a valid file path.

Parameters

<i>path</i>	The given path
-------------	----------------

Definition at line 17 of file Filesystem.cpp.

Referenced by multiscale::OperatingSystem::executeProgramAndVerifyPath(), multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile(), multiscale::XmlValidator::validateXmlFilepath(), and multiscale::XmlValidator::validateXmlSchemaPath().

7.55.2.3 bool **Filesystem::isValidFolderPath** (const std::string & *path*) [static]

Check if the given path is a valid folder path.

Parameters

<i>path</i>	The given path
-------------	----------------

Definition at line 7 of file Filesystem.cpp.

Referenced by `getFilesInFolder()`, and `multiscale::verification::SpatialTemporalDataReader::validateFolderPath()`.

**7.55.2.4 std::string Filesystem::nativeFormatFilePath (const std::string & *path*)
[static]**

Return the given path as an absolute path in native format.

Parameters

<i>path</i>	The given path
-------------	----------------

Definition at line 27 of file Filesystem.cpp.

References `ERR_INVALID_PATH`, and `MS_throw`.

7.55.3 Member Data Documentation

7.55.3.1 const std::string Filesystem::ERR_INVALID_PATH = "The given input file path is invalid. Please change." [static, private]

Definition at line 51 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](#)

7.56 multiscale::verification::FilterNumericMeasureAttribute Class Reference

Class for representing a filter numeric measure.

```
#include <FilterNumericMeasureAttribute.hpp>
```

Public Attributes

- [FilterNumericMeasureAttributeType filterNumericMeasure](#)

7.56.1 Detailed Description

Class for representing a filter numeric measure.

Definition at line 32 of file FilterNumericMeasureAttribute.hpp.

7.56.2 Member Data Documentation

7.56.2.1 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:

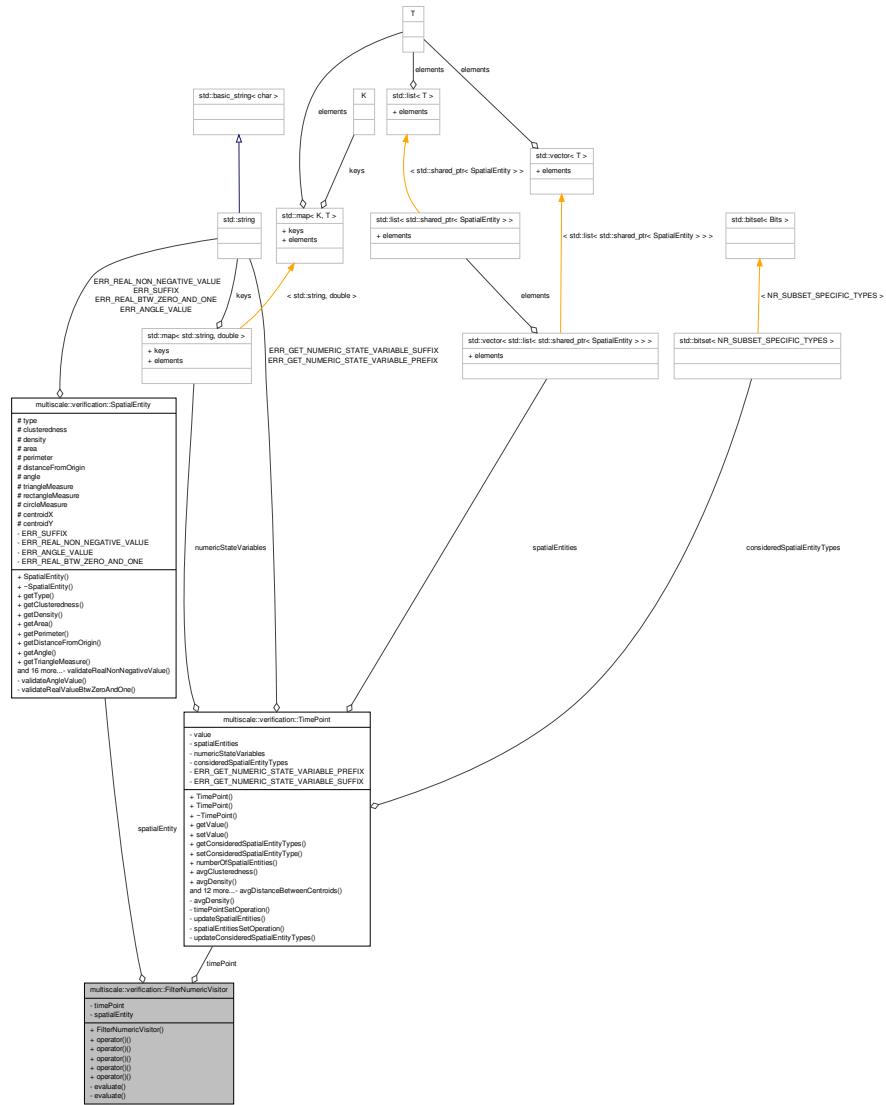
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[FilterNumericMeasureAttribute.hpp](#)

7.57 multiscale::verification::FilterNumericVisitor Class Reference

Class for evaluating filter numeric measures.

```
#include <FilterNumericVisitor.hpp>
```

Collaboration diagram for multiscale::verification::FilterNumericVisitor:



Public Member Functions

- `FilterNumericVisitor` (const `TimePoint` &`timePoint`, const `SpatialEntity` &`spatialEntity`)
 - 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`

7.57.1 Detailed Description

Class for evaluating filter numeric measures.

Definition at line 16 of file FilterNumericVisitor.hpp.

7.57.2 Constructor & Destructor Documentation

7.57.2.1 multiscale::verification::FilterNumericVisitor::FilterNumericVisitor (const `TimePoint` & `timePoint`, const `SpatialEntity` & `spatialEntity`) [inline]

Definition at line 25 of file FilterNumericVisitor.hpp.

Referenced by `evaluate()`.

7.57.3 Member Function Documentation

7.57.3.1 `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><i>filter-</i> <i>Numeric-</i> <i>Measure</i></code>	The given filter numeric measure
--	----------------------------------

Definition at line 81 of file FilterNumericVisitor.hpp.

References FilterNumericVisitor(), spatialEntity, and timePoint.

Referenced by operator()().

7.57.3.2 `double multiscale::verification::FilterNumericVisitor::evaluate (const PrimaryNumericMeasureAttributeType & primaryNumericMeasure) const [inline, private]`

Evaluate the given primary numeric measure considering the timePoint field.

Parameters

<code><i>primary-</i> <i>Numeric-</i> <i>Measure</i></code>	The given primary numeric measure
---	-----------------------------------

Definition at line 89 of file FilterNumericVisitor.hpp.

References timePoint.

7.57.3.3 `double multiscale::verification::FilterNumericVisitor::operator() (const FilterNumericMeasureAttribute & filterNumericMeasure) const [inline]`

Overloading the "(" operator for the [FilterNumericMeasureAttribute](#) alternative.

Parameters

<code><i>filter-</i> <i>Numeric-</i> <i>Measure</i></code>	The filter numeric measure
--	----------------------------

Definition at line 32 of file FilterNumericVisitor.hpp.

References `evaluate()`, and `multiscale::verification::FilterNumericMeasureAttribute::filterNumericMeasure`.

7.57.3.4 double multiscale::verification::FilterNumericVisitor::operator() (const PrimaryNumericMeasureAttribute & *primaryNumericMeasure*) const [inline]

Overloading the "(") operator for the `PrimaryNumericMeasureAttribute` alternative.

Parameters

<code>primary-Numeric-Measure</code>	The primary numeric measure
--------------------------------------	-----------------------------

Definition at line 40 of file `FilterNumericVisitor.hpp`.

References `evaluate()`, and `multiscale::verification::PrimaryNumericMeasureAttribute::primaryNumericMeasure`.

7.57.3.5 double multiscale::verification::FilterNumericVisitor::operator() (const SpatialMeasureAttribute & *spatialMeasure*) const [inline]

Overloading the "(") operator for the `SpatialMeasureAttribute` alternative.

Parameters

<code>spatial-Measure</code>	The spatial measure
------------------------------	---------------------

Definition at line 48 of file `FilterNumericVisitor.hpp`.

References `evaluate()`, `spatialEntity`, and `multiscale::verification::SpatialMeasureAttribute::spatialMeasureType`.

7.57.3.6 double multiscale::verification::FilterNumericVisitor::operator() (const UnaryNumericFilterAttribute & *unaryNumericFilter*) const [inline]

Overloading the "(") operator for the `UnaryNumericFilterAttribute` alternative.

Parameters

<code>unary-Numeric-Filter</code>	The unary numeric filter
-----------------------------------	--------------------------

Definition at line 56 of file `FilterNumericVisitor.hpp`.

References `evaluate()`, `multiscale::verification::UnaryNumericFilterAttribute::filter-NumericMeasure`, `multiscale::verification::UnaryNumericFilterAttribute::unaryNumeric-Measure`, and `multiscale::verification::UnaryNumericMeasureAttribute::unaryNumeric-MeasureType`.

**7.57.3.7 double multiscale::verification::FilterNumericVisitor::operator() (const
BinaryNumericFilterAttribute & *binaryNumericFilter*) const [inline]**

Overloading the "(" operator for the `BinaryNumericFilterAttribute` alternative.

Parameters

<code>binary-</code> <code>Numeric-</code> <code>Filter</code>	The binary numeric filter
--	---------------------------

Definition at line 67 of file `FilterNumericVisitor.hpp`.

References `multiscale::verification::BinaryNumericFilterAttribute::binaryNumeric-Measure`, `multiscale::verification::BinaryNumericMeasureAttribute::binaryNumeric-MeasureType`, `evaluate()`, `multiscale::verification::BinaryNumericFilterAttribute::first-FilterNumericMeasure`, and `multiscale::verification::BinaryNumericFilterAttribute::secondFilterNumericMeasure`.

7.57.4 Member Data Documentation

**7.57.4.1 const SpatialEntity& multiscale::verification::FilterNumericVisitor-
::spatialEntity [private]**

The considered spatial entity

Definition at line 21 of file `FilterNumericVisitor.hpp`.

Referenced by `evaluate()`, and `operator()()`.

**7.57.4.2 const TimePoint& multiscale::verification::FilterNumericVisitor::timePoint
[private]**

The considered timepoint

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

Referenced by `evaluate()`.

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

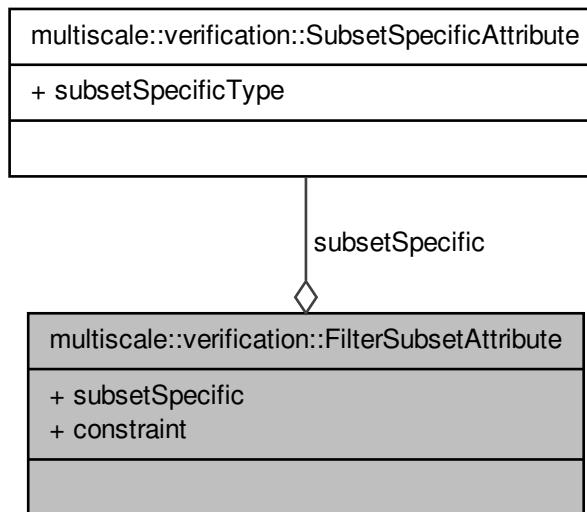
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[FilterNumeric-Visitor.hpp](#)

7.58 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`

7.58.1 Detailed Description

Class for representing a filter subset attribute.

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

7.58.2 Member Data Documentation

7.58.2.1 ConstraintAttributeType multiscale::verification::FilterSubsetAttribute- ::constraint

The constraint

Definition at line 20 of file FilterSubsetAttribute.hpp.

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

7.58.2.2 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](#)

7.59 multiscale::verification::FutureLogicPropertyAttribute Class - Reference

Class for representing a "future" logic property attribute.

```
#include <FutureLogicPropertyAttribute.hpp>
```

Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [LogicPropertyAttributeType logicProperty](#)

7.59.1 Detailed Description

Class for representing a "future" logic property attribute.

Definition at line 14 of file FutureLogicPropertyAttribute.hpp.

7.59.2 Member Data Documentation

7.59.2.1 unsigned long multiscale::verification::FutureLogicPropertyAttribute::end-Timepoint

The considered end timepoint

Definition at line 19 of file FutureLogicPropertyAttribute.hpp.

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

7.59.2.2 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().

7.59.2.3 unsigned long multiscale::verification::FutureLogicPropertyAttribute::startTimepoint

The considered start timepoint

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](#)

7.60 multiscale::Geometry2D Class Reference

Two-dimensional geometric operations.

```
#include <Geometry2D.hpp>
```

Static Public Member Functions

- static double [angleOfLineWrtOxAxis](#) (const Point2f &a, const 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)

Check if the opposite of angle1, ((angle1 + 180) % 360), lies between angles 2 and 3.
- static bool `isAngleBetweenNonReflex` (double angle1, double angle2, double angle3)

Check if angle1 lies between non reflex angle determined by angles 2 and 3.
- static bool `isOppositeAngleBetweenNonReflex` (double angle1, double angle2, double angle3)

Check if the opposite of angle1, ((angle1 + 180) % 360), 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 Point2f &a, const Point2f &b, double &slope)

Compute the slope of the line defined by points "a" and "b".
- static double `distanceBtwPoints` (const Point2f &a, const 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 Point2f &a, const Point2f &linePointB, const Point2f &linePointC)

Compute the distance from a point "a" to a line specified by two points "B" and "C".
- static Point2f `middlePoint` (const Point2f &a, const Point2f &b)

Get the point in the middle of the segment determined by points "a" and "b".
- static void `orthogonalLineToAnotherLineEdgePoints` (const Point &a1, const Point &b1, Point &a2, 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 Point2f &p1, const Point2f &p2, const Point2f &a, const 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 Point2f &p, const Point2f &q, double &a, double &b, double &c)

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.
- 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 Point2f &a1, const Point2f &b1, const Point2f &a2, const Point2f &b2)

Check if two lines are identical.
- static bool `lineIntersection` (const Point2f &a1, const Point2f &b1, const Point2f &a2, const Point2f &b2, Point2f &intersection)

Determine the intersection point of two lines, if this point exists.
- static bool `lineIntersection` (const Point &a1, const Point &b1, const Point &a2, const Point &b2, 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, Point2f &intersection)

Determine the intersection point of two lines, if this point exists.

- static bool `lineSegmentIntersection` (const Point &a1, const Point &b1, const Point &a2, const Point &b2, Point &intersection)

Determine the intersection point of two line segments, if this point exists.

- static bool `lineCircleIntersection` (Point2f a, Point2f b, const Point2f &circleOrigin, double radius, vector< Point2f > &intersectionPoints)

Determine if a line and a circle intersect and return the intersection points if they exist.

- static bool `lineSegmentCircleIntersection` (const Point2f &a, const Point2f &b, const Point2f &circleOrigin, double radius, vector< Point2f > &intersectionPoints)

Determine if a line segment and a circle intersect and return the intersection points if they exist.

- static double `angleBtwPoints` (const Point2f &a, const Point2f &b, const Point2f &c)

Compute the angle between three points.

- static vector< Point2f > `findPointsOnEdge` (const vector< 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 vector< Point > &points, const Point2f &origin)

Get the index of the point which is the closest to the origin.

- static double `areaOfTriangle` (const Point2f &a, const Point2f &b, const Point2f &c)

Compute the area of a triangle defined by three points.

- static bool `isPointOnLineSegment` (const Point2f &point, const Point2f &lineSegmentStart, const Point2f &lineSegmentEnd)

Check if one point lies between two other points.

- static bool `areEqualPoints` (const Point2f &point1, const Point2f &point2)

Check if points point1 and point2 are equal or not.

- static bool `areCollinear` (const Point2f &point1, const Point2f &point2, const 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 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` (Point2f &point, const Point2f &translation)
Translate a point by the given values.
- static void `inverseTranslate` (Point2f &point, const Point2f &translation)
Inverse translate a point by the given values.
- static void `lineCircleTwoIntersectionPoints` (const Point2f &circleOrigin, double A, double B, double C, double delta, vector< Point2f > &intersectionPoints)
Treat the case when the line and circle intersect in two points.
- static void `lineCircleOneIntersectionPoint` (const Point2f &circleOrigin, double A, double B, double C, double delta, vector< Point2f > &intersectionPoints)
Treat the case when the line and circle intersect in one point.

7.60.1 Detailed Description

Two-dimensional geometric operations.

Definition at line 16 of file Geometry2D.hpp.

7.60.2 Member Function Documentation

7.60.2.1 double Geometry2D::angleBtwPoints (const Point2f & a, const Point2f & b, const 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>	Point2f a
<i>b</i>	Point2f b
<i>c</i>	Point2f c

Definition at line 315 of file Geometry2D.cpp.

References PI.

Referenced by multiscale::analysis::Detector::polygonAngle().

7.60.2.2 double Geometry2D::angleOfLineWrtOxAxis (const Point2f & a, const 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

a	Point2f a
b	Point2f b

Definition at line 10 of file Geometry2D.cpp.

References PI.

Referenced by multiscale::MinEnclosingTriangleFinder::intersects(), multiscale::MinEnclosingTriangleFinder::intersectsAbove(), and multiscale::MinEnclosingTriangleFinder::intersectsBelow().

7.60.2.3 double Geometry2D::areaOfTriangle (const Point2f & a, const Point2f & b, const 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

a	Point2f a
b	Point2f b
c	Point2f c

Definition at line 360 of file Geometry2D.cpp.

Referenced by multiscale::MinEnclosingTriangleFinder::returnMinEnclosingTriangle(), and multiscale::MinEnclosingTriangleFinder::updateMinEnclosingTriangle().

7.60.2.4 bool Geometry2D::areCollinear (const Point2f & point1, const Point2f & point2, const Point2f & point3) [static]

Check if the three points are collinear.

Parameters

point1	Point 1
point2	Point 2
point3	Point 3

Definition at line 382 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

7.60.2.5 bool Geometry2D::areEqualPoints (const Point2f & point1, const 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 378 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle(), and lineEquationDeterminedByPoints().

7.60.2.6 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 169 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::areIdenticalLines().

7.60.2.7 bool Geometry2D::areIdenticalLines (const Point2f & a1, const Point2f & b1, const Point2f & a2, const 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 180 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

7.60.2.8 bool Geometry2D::areOnTheSameSideOfLine (const Point2f & *p1*, const Point2f & *p2*, const Point2f & *a*, const 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 <i>p1</i>
<i>p2</i>	Point <i>p2</i>
<i>a</i>	First point for determining line
<i>b</i>	Second point for determining line

Definition at line 149 of file Geometry2D.cpp.

References lineEquationDeterminedByPoints(), and multiscale::Numeric::sign().

Referenced by multiscale::MinEnclosingTriangleFinder::findVertexCOnSideB(), and multiscale::MinEnclosingTriangleFinder::gamma().

7.60.2.9 double Geometry2D::distanceBtwPoints (const Point2f & *a*, const Point2f & *b*) [static]

Compute the distance between two points.

Compute the Euclidean distance between two points

Parameters

<i>a</i>	Point2f <i>a</i>
<i>b</i>	Point2f <i>b</i>

Definition at line 70 of file Geometry2D.cpp.

Referenced by multiscale::verification::TimePoint::avgDistanceBetweenCentroids(), multiscale::analysis::Silhouette::computeAverageDissimilarityBtwEntityAndCluster(), multiscale::analysis::Silhouette::computeAverageDissimilarityWithinCluster(), multiscale::analysis::RegionDetector::createRegionFromPolygon(), multiscale::analysis::Entity-

::distanceTo(), isPointOnLineSegment(), minimumDistancePointIndex(), multiscale::analysis::RegionDetector::sumOfAverageCentroidDistances(), and multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues().

7.60.2.10 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 77 of file Geometry2D.cpp.

7.60.2.11 double Geometry2D::distanceFromPointToLine (const Point2f & *a*, const Point2f & *linePointB*, const 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/Point-LineDistance2--Dimensional.html>

Parameters

<i>a</i>	Point2f 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 84 of file Geometry2D.cpp.

Referenced by multiscale::MinEnclosingTriangleFinder::height().

7.60.2.12 vector< Point2f > Geometry2D::findPointsOnEdge (const vector< 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 327 of file Geometry2D.cpp.

References [isPointOnEdge\(\)](#).

7.60.2.13 void Geometry2D::inverseTranslate (Point2f & *point*, const 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 408 of file Geometry2D.cpp.

Referenced by [lineCircleOneIntersectionPoint\(\)](#), and [lineCircleTwoIntersectionPoints\(\)](#).

7.60.2.14 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\(\)](#).

7.60.2.15 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().

7.60.2.16 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

c	Coordinate c
c1	Coordinate c1
c2	Coordinate c2

Definition at line 399 of file Geometry2D.cpp.

7.60.2.17 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

angle1	The angle for which the opposite angle lies between angle2 and angle3 or not
angle2	One of the boundary angles
angle3	The other boundary angle

Definition at line 28 of file Geometry2D.cpp.

References isAngleBetween(), and oppositeAngle().

7.60.2.18 bool Geometry2D::isOppositeAngleBetweenNonReflex (double angle1, double angle2, double angle3) [static]

Check if the opposite of angle1, ((angle1 + 180) % 360), lies between non reflex angle determined by angles 2 and 3.

Parameters

angle1	The angle which lies between angle2 and angle3 or not
angle2	One of the boundary angles
angle3	The other boundary angle

Definition at line 46 of file Geometry2D.cpp.

References `isAngleBetweenNonReflex()`, and `oppositeAngle()`.

Referenced by `multiscale::MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor()`.

7.60.2.19 `bool Geometry2D::isPointOnEdge (const 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: $((p.x == 1) \&\& (p.y > 1) \&\& (p.y < \text{nrOfCols}))$ OR $((p.x == \text{nrOfRows}) \&\& (p.y > 1) \&\& (p.y < \text{nrOfCols}))$ OR $((p.y == 1) \&\& (p.x > 1) \&\& (p.x < \text{nrOfRows}))$ OR $((p.y == \text{nrOfCols}) \&\& (p.x > 1) \&\& (p.x < \text{nrOfRows}))$

Parameters

<i>p</i>	Point2f p
<i>nrOfRows</i>	The number of rows
<i>nrOfCols</i>	The number of columns

Definition at line 389 of file Geometry2D.cpp.

References `MATRIX_START_INDEX`.

Referenced by `findPointsOnEdge()`, and `orthogonalLineToAnotherLineEdgePoints()`.

7.60.2.20 `bool Geometry2D::isPointOnLineSegment (const Point2f & point, const Point2f & lineSegmentStart, const Point2f & lineSegmentEnd) [static]`

Check if one point lies between two other points.

Parameters

<i>point</i>	Point lying possibly outside the line segment
<i>line-Segment-Start</i>	First point determining the line segment
<i>line-Segment-End</i>	Second point determining the line segment

Definition at line 369 of file Geometry2D.cpp.

References `multiscale::Numeric::almostEqual()`, and `distanceBtwPoints()`.

Referenced by `multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle()`.

7.60.2.21 **bool Geometry2D::lineCircleIntersection (Point2f a, Point2f b, const Point2f & circleOrigin, double radius, vector<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>intersection- Points</i>	The intersection points between the circle and the line

< Two intersection points

< One intersection point

Definition at line 267 of file Geometry2D.cpp.

References `lineCircleOneIntersectionPoint()`, `lineCircleTwoIntersectionPoints()`, and `translate()`.

Referenced by `lineSegmentCircleIntersection()`.

7.60.2.22 **void Geometry2D::lineCircleOneIntersectionPoint (const Point2f & circleOrigin, double A, double B, double C, double delta, vector<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>	$A \cdot x_1 + B \cdot y_1$
<i>delta</i>	$(4 * B^2 * C^2) - (4 * (A^2 + B^2) * (C^2 - (R^2 * A^2)))$
<i>intersection- Points</i>	Intersection points

Definition at line 431 of file Geometry2D.cpp.

References `inverseTranslate()`.

Referenced by `lineCircleIntersection()`.

7.60.2.23 void Geometry2D::lineCircleTwoIntersectionPoints (const Point2f & *circleOrigin*, double *A*, double *B*, double *C*, double *delta*, vector< 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>	$A \cdot x_1 + B \cdot y_1$
<i>delta</i>	$(4 * B^2 * C^2) - (4 * (A^2 + B^2) * (C^2 - (R^2 * A^2)))$
<i>intersection-Points</i>	Intersection points

Definition at line 413 of file Geometry2D.cpp.

References inverseTranslate().

Referenced by lineCircleIntersection().

7.60.2.24 void Geometry2D::lineEquationDeterminedByPoints (const Point2f & *p*, const Point2f & *q*, double & *a*, double & *b*, double & *c*) [static]

Get the values of "a", "b" and "c" of the line equation $ax + by + c = 0$ 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>	Point2f <i>p</i>
<i>q</i>	Point2f <i>q</i>
<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 161 of file Geometry2D.cpp.

References areEqualPoints().

Referenced by areOnTheSameSideOfLine(), and multiscale::MinEnclosingTriangleFinder::lineEquationParameters().

7.60.2.25 bool Geometry2D::lineIntersection (const Point2f & *a1*, const Point2f & *b1*, const Point2f & *a2*, const Point2f & *b2*, 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 $\det (= A_1B_2 - A_2B_1) == 0$, then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with $P(x, y)$ where: $x = (C_1B_2 - C_2B_1) / (\det)$ $y = (C_2A_1 - C_1A_2) / (\det)$

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 220 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

Referenced by multiscale::MinEnclosingTriangleFinder::areIntersectingLines(), multiscale::MinEnclosingTriangleFinder::isLocalMinimalTriangle(), lineSegmentIntersection(), and multiscale::MinEnclosingTriangleFinder::middlePointOfSideB().

7.60.2.26 bool Geometry2D::lineIntersection (const Point & *a1*, const Point & *b1*, const Point & *a2*, const Point & *b2*, 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 $\det (= A_1B_2 - A_2B_1) == 0$, then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with $P(x, y)$ where: $x = (C_1B_2 - C_2B_1) / (\det)$ $y = (C_2A_1 - C_1A_2) / (\det)$

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 199 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

7.60.2.27 bool Geometry2D::lineIntersection (double *a1*, double *b1*, double *c1*, double *a2*, double *b2*, double *c2*, 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: $A_1x + B_1y = C_1$ $A_2x + B_2y = C_2$

If $\det (= A_1B_2 - A_2B_1) == 0$, then lines are parallel else they intersect

If they intersect, then let us denote the intersection point with $P(x, y)$ where: $x = (C_1B_2 - C_2B_1) / (\det)$ $y = (C_2A_1 - C_1A_2) / (\det)$

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 241 of file Geometry2D.cpp.

References multiscale::Numeric::almostEqual().

7.60.2.28 bool Geometry2D::lineSegmentCircleIntersection (const Point2f & *a*, const Point2f & *b*, const Point2f & *circleOrigin*, double *radius*, vector< 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>intersection- Points</i>	The intersection points between the circle and the line

Definition at line 296 of file Geometry2D.cpp.

References `lineCircleIntersection()`.

7.60.2.29 bool Geometry2D::lineSegmentIntersection (const Point & *a1*, const Point & *b1*, const Point & *a2*, const Point & *b2*, 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 254 of file `Geometry2D.cpp`.

References `lineIntersection()`.

Referenced by `multiscale::analysis::Detector::findGoodIntersectionPoints()`.

7.60.2.30 Point2f Geometry2D::middlePoint (const Point2f & *a*, const Point2f & *b*) [static]

Get the point in the middle of the segment determined by points "a" and "b".

Parameters

<i>a</i>	Point2f a
<i>b</i>	Point2f b

Definition at line 96 of file `Geometry2D.cpp`.

Referenced by `multiscale::MinEnclosingTriangleFinder::isValidMinimalTriangle()`, and `multiscale::MinEnclosingTriangleFinder::middlePointOfSideB()`.

7.60.2.31 unsigned int Geometry2D::minimumDistancePointIndex (const vector< Point > & *points*, const 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 341 of file Geometry2D.cpp.

References [distanceBtwPoints\(\)](#).

Referenced by [multiscale::analysis::RegionDetector::createRegionFromPolygon\(\)](#), and [multiscale::analysis::ClusterDetector::updateClusterOriginDependentValues\(\)](#).

7.60.2.32 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 52 of file Geometry2D.cpp.

Referenced by [multiscale::MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessorAndSuccessor\(\)](#), [isOppositeAngleBetween\(\)](#), and [isOppositeAngleBetweenNonReflex\(\)](#).

7.60.2.33 void Geometry2D::orthogonalLineToAnotherLineEdgePoints (const Point & *a1*, const Point & *b1*, Point & *a2*, 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 103 of file Geometry2D.cpp.

References [isPointOnEdge\(\)](#).

Referenced by [multiscale::analysis::Detector::findGoodPointsForAngle\(\)](#).

7.60.2.34 bool Geometry2D::slopeOfLine (const Point2f & a, const 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>	Point2f a
<i>b</i>	Point2f b
<i>slope</i>	Slope of the line if it is different from (+/-)infinity

Definition at line 57 of file Geometry2D.cpp.

7.60.2.35 void Geometry2D::translate (Point2f & point, const 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 403 of file Geometry2D.cpp.

Referenced by lineCircleIntersection().

7.60.3 Member Data Documentation

7.60.3.1 const int Geometry2D::MATRIX_START_INDEX = 1 [static]

Definition at line 453 of file Geometry2D.hpp.

Referenced by isPointOnEdge().

**7.60.3.2 const double Geometry2D::PI = 3.
14159265358979323846264338327950288419716939937510 [static]**

Definition at line 452 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](#)

7.61 multiscale::verification::GlobalLogicPropertyAttribute Class - Reference

Class for representing a "globally" logic property attribute.

```
#include <GlobalLogicPropertyAttribute.hpp>
```

Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [LogicPropertyAttributeType logicProperty](#)

7.61.1 Detailed Description

Class for representing a "globally" logic property attribute.

Definition at line 14 of file GlobalLogicPropertyAttribute.hpp.

7.61.2 Member Data Documentation

7.61.2.1 unsigned long multiscale::verification::GlobalLogicPropertyAttribute::end-Timepoint

The considered end timepoint

Definition at line 19 of file GlobalLogicPropertyAttribute.hpp.

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

7.61.2.2 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().

7.61.2.3 **unsigned long multiscale::verification::GlobalLogicPropertyAttribute-
::startTimestep**

The considered start timestep

Definition at line 18 of file [GlobalLogicPropertyAttribute.hpp](#).

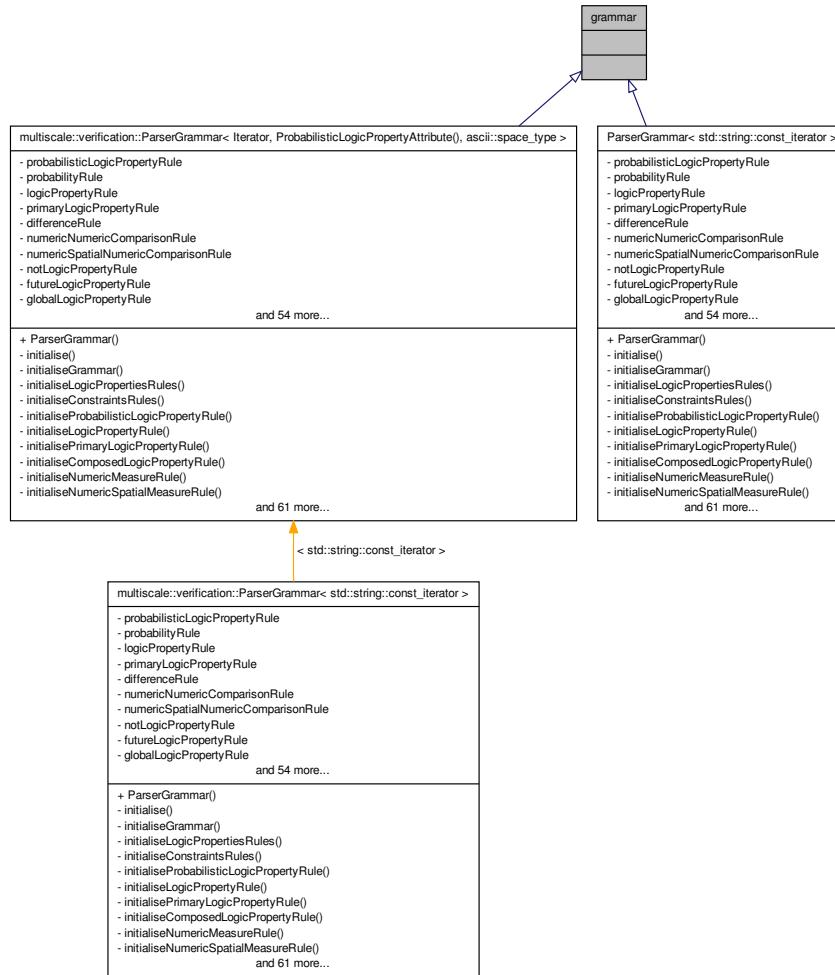
Referenced by [multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogic-
Property\(\)](#).

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/**GlobalLogic-
PropertyAttribute.hpp**](#)

7.62 grammar Class Reference

Inheritance diagram for grammar:



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/parsing/[ParserGrammar.hpp](#)

7.63 multiscale::verification::ImplicationConstraintAttribute Class Reference

Class for representing an "implication" constraint attribute.

```
#include <ImplicationConstraintAttribute.hpp>
```

Public Attributes

- [ConstraintAttributeType constraint](#)

7.63.1 Detailed Description

Class for representing an "implication" constraint attribute.

Definition at line 14 of file ImplicationConstraintAttribute.hpp.

7.63.2 Member Data Documentation

7.63.2.1 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**](#)

7.64 multiscale::verification::ImplicationLogicPropertyAttribute - Class Reference

Class for representing an "implication" logic property attribute.

```
#include <ImplicationLogicPropertyAttribute.hpp>
```

Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

7.64.1 Detailed Description

Class for representing an "implication" logic property attribute.

Definition at line 14 of file ImplicationLogicPropertyAttribute.hpp.

7.64.2 Member Data Documentation

7.64.2.1 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:

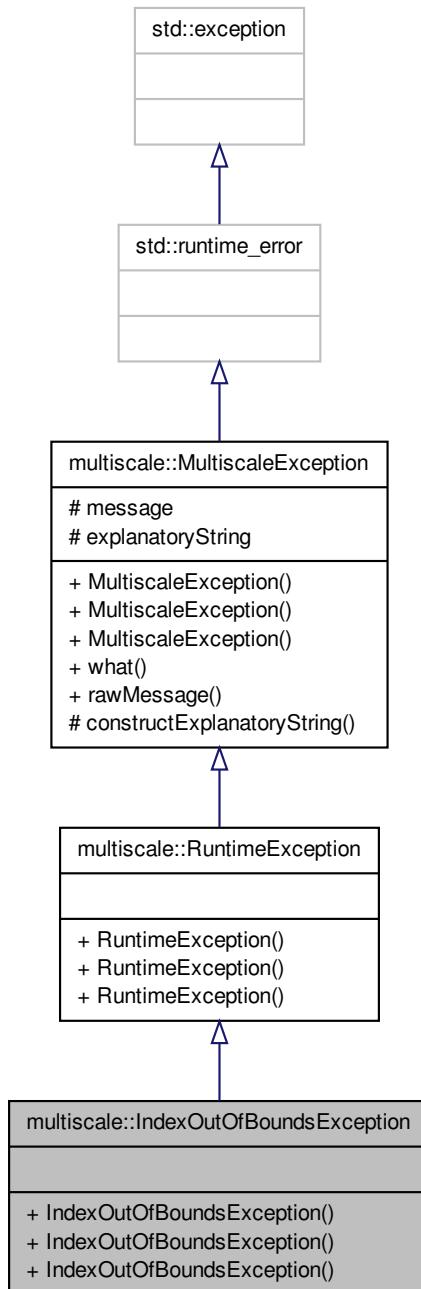
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[ImplicationLogicPropertyAttribute.hpp](#)

7.65 multiscale::IndexOutOfBoundsException Class Reference

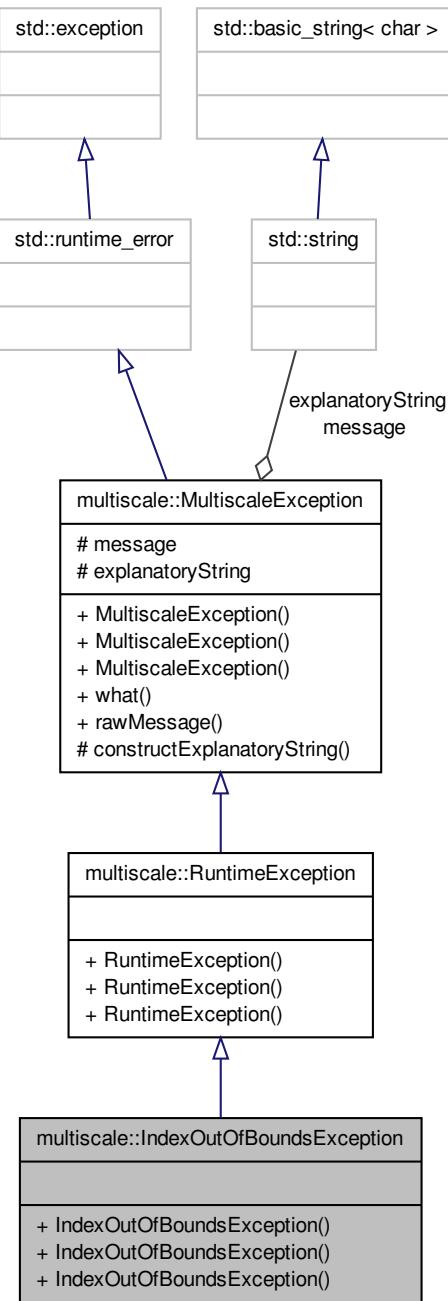
Class for representing an index out of bounds exception.

```
#include <IndexOutOfBoundsException.hpp>
```

Inheritance diagram for multiscale::IndexOutOfBoundsException:



Collaboration diagram for multiscale::IndexOutOfBoundsException:



Public Member Functions

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

7.65.1 Detailed Description

Class for representing an index out of bounds exception.

Definition at line 14 of file `IndexOutOfBoundsException.hpp`.

7.65.2 Constructor & Destructor Documentation

7.65.2.1 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException () [inline]

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

7.65.2.2 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException (const string & file, int line, const string & msg) [inline, explicit]

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

7.65.2.3 multiscale::IndexOutOfBoundsException::IndexOutOfBoundsException (const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 of file `IndexOutOfBoundsException.hpp`.

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

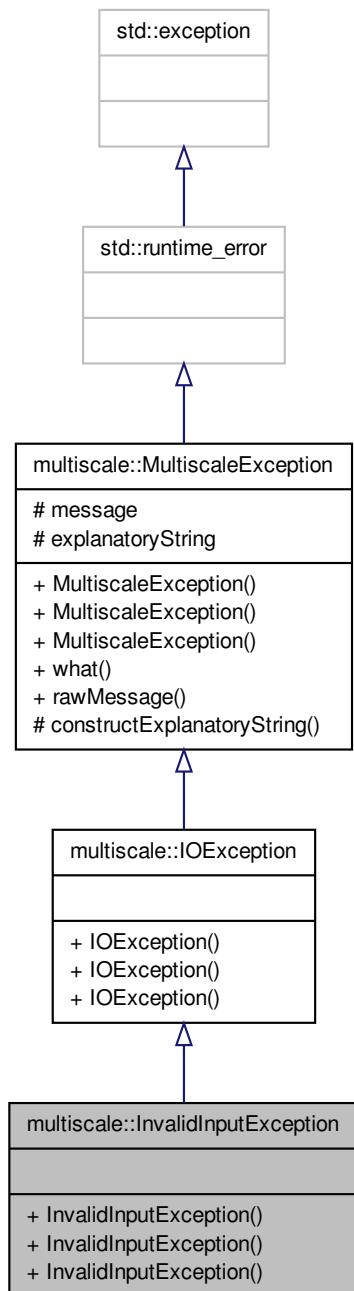
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/IndexOutOfBoundsException.hpp](#)

7.66 multiscale::InvalidInputException Class Reference

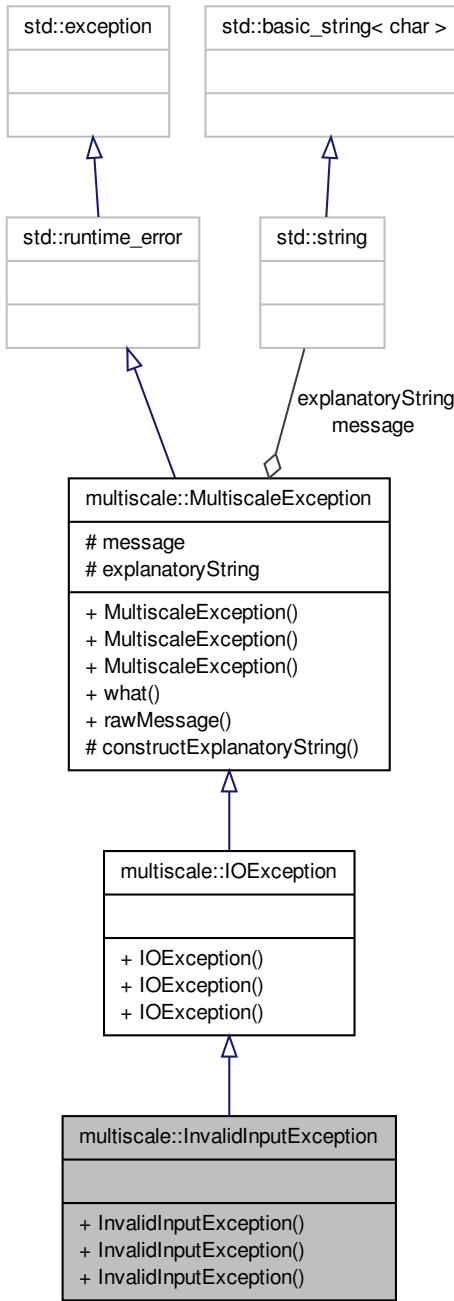
Class for representing invalid input exceptions.

```
#include <InvalidInputException.hpp>
```

Inheritance diagram for multiscale::InvalidInputException:



Collaboration diagram for multiscale::InvalidInputException:



Public Member Functions

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

7.66.1 Detailed Description

Class for representing invalid input exceptions.

Definition at line 14 of file InvalidInputException.hpp.

7.66.2 Constructor & Destructor Documentation

7.66.2.1 **multiscale::InvalidInputException::InvalidInputException ()** [inline]

Definition at line 18 of file InvalidInputException.hpp.

7.66.2.2 **multiscale::InvalidInputException::InvalidInputException (const string &file, int line, const string & msg)** [inline, explicit]

Definition at line 20 of file InvalidInputException.hpp.

7.66.2.3 **multiscale::InvalidInputException::InvalidInputException (const string &file, int line, const char * msg)** [inline, explicit]

Definition at line 24 of file InvalidInputException.hpp.

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

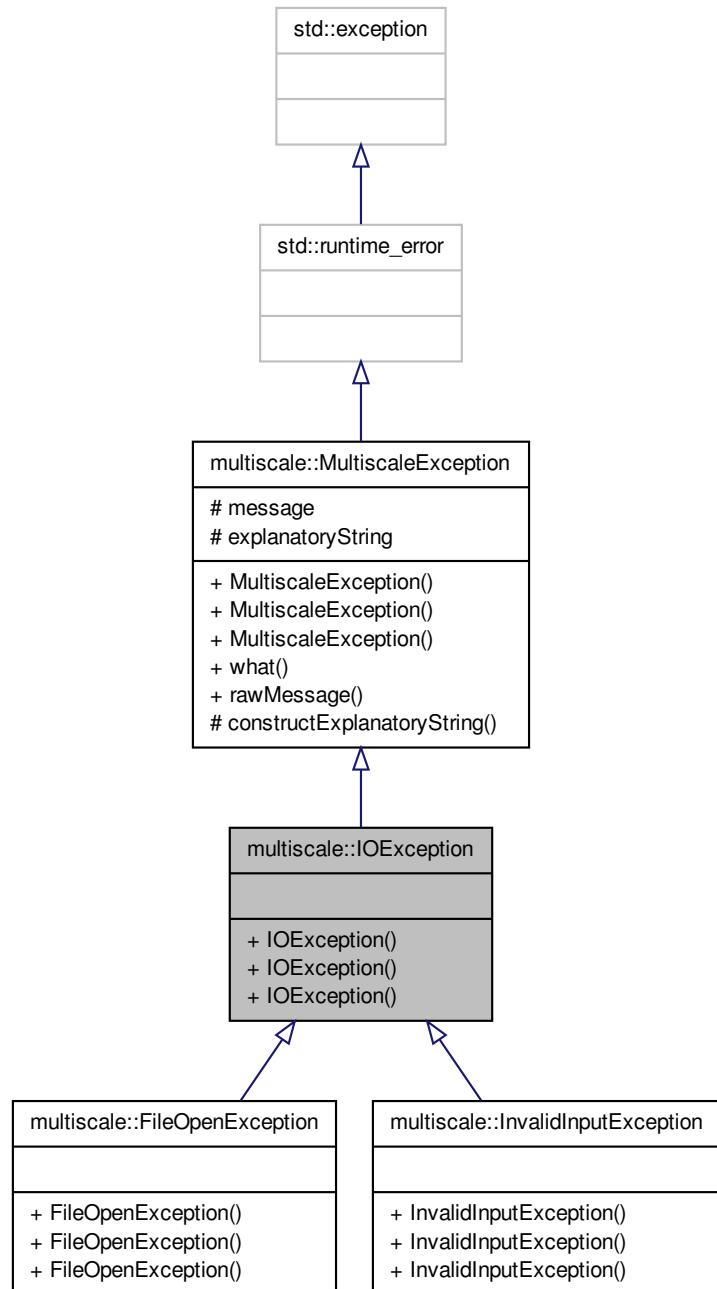
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/InvalidInputException.hpp](#)

7.67 multiscale::IOException Class Reference

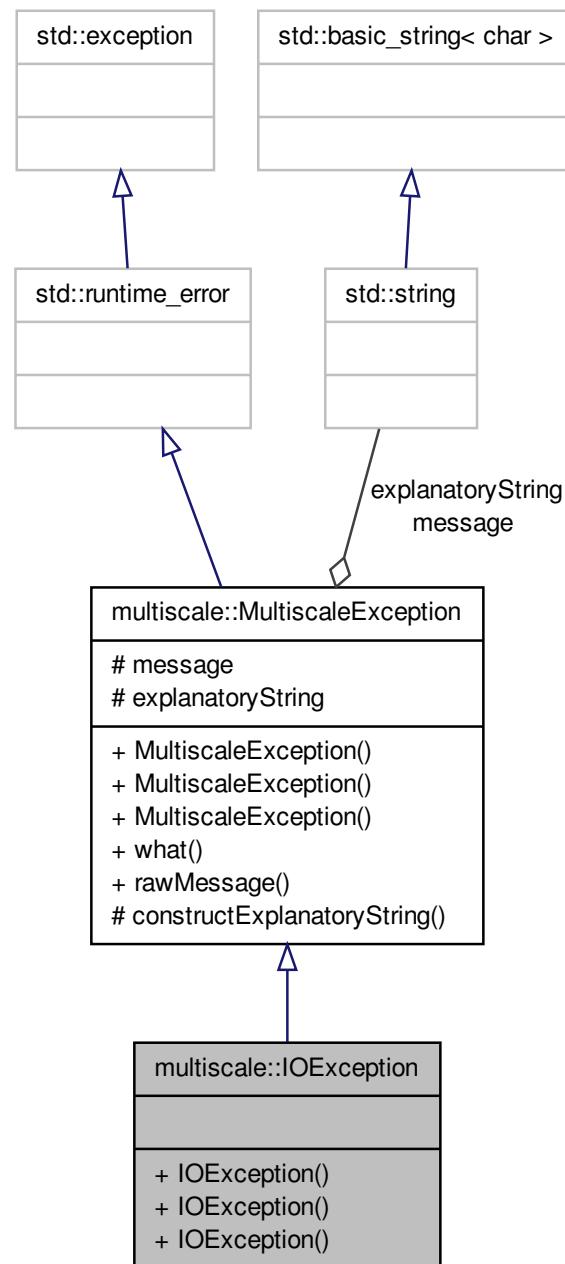
Class for representing input and output exceptions.

```
#include <IOException.hpp>
```

Inheritance diagram for multiscale::IOException:



Collaboration diagram for multiscale::IOException:



Public Member Functions

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

7.67.1 Detailed Description

Class for representing input and output exceptions.

Definition at line 14 of file IOException.hpp.

7.67.2 Constructor & Destructor Documentation

7.67.2.1 multiscale::IOException::IOException() [inline]

Definition at line 18 of file IOException.hpp.

7.67.2.2 multiscale::IOException::IOException(const string & file, int line, const string & msg) [inline, explicit]

Definition at line 20 of file IOException.hpp.

7.67.2.3 multiscale::IOException::IOException(const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 of file IOException.hpp.

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

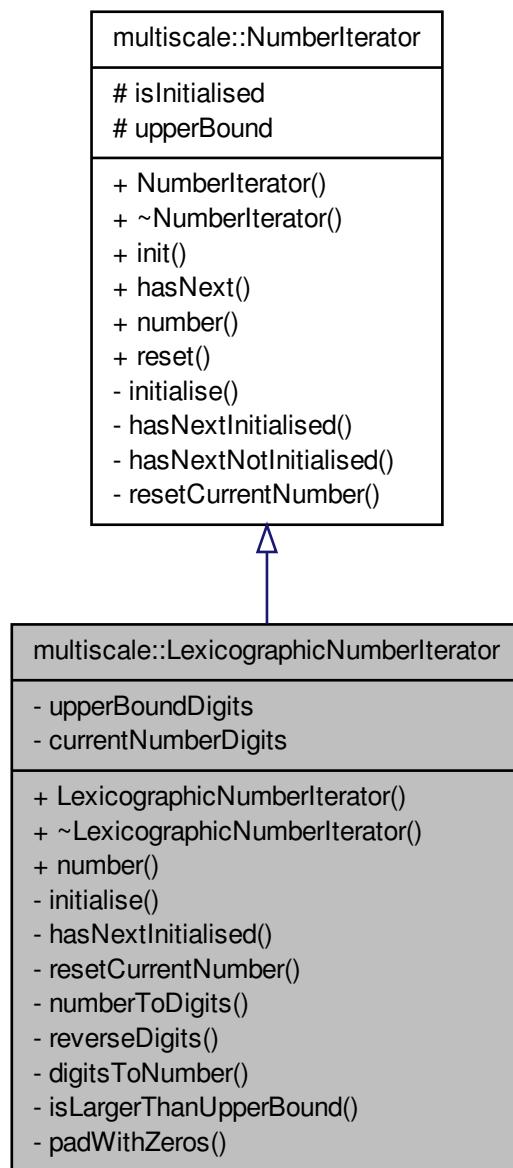
- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-
[IOException.hpp](#)

7.68 multiscale::LexicographicNumberIterator Class Reference

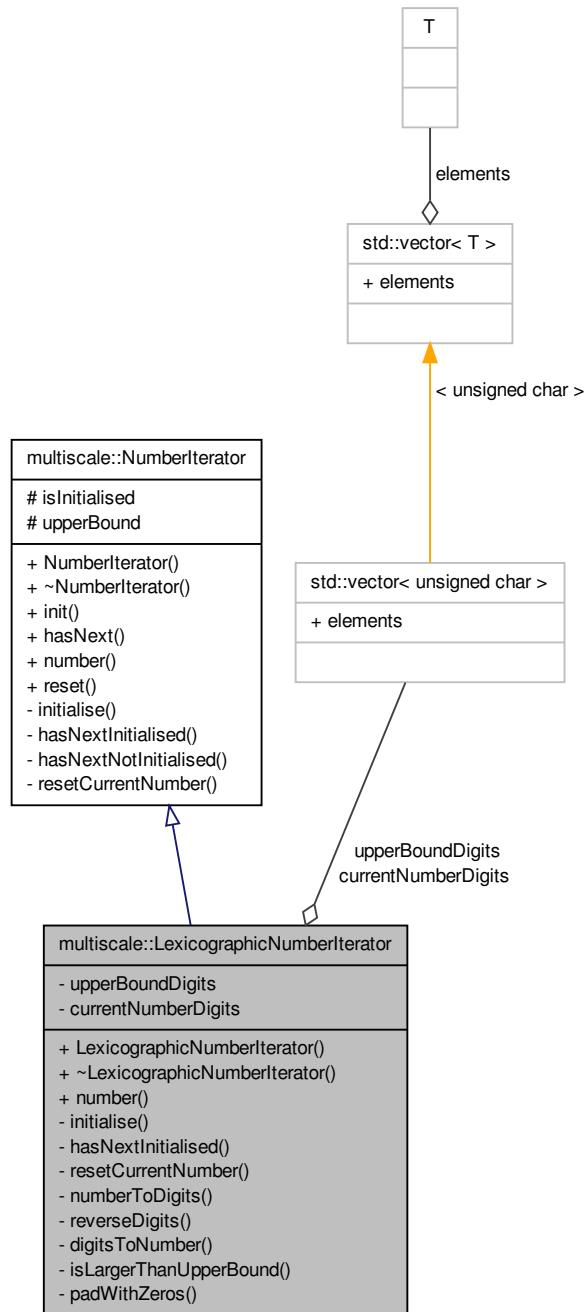
Iterator class starting at 1 and ending at the provided upper bound considering that each number is followed by an "_".

```
#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 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, vector< unsigned char > &digits)`
Convert the number to a vector of digits.
- `void reverseDigits (vector< unsigned char > &digits)`
Reverse the order of the digits.
- `unsigned int digitsToNumber (vector< unsigned char > &digits)`
Convert the 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

- `vector< unsigned char > upperBoundDigits`
- `vector< unsigned char > currentNumberDigits`

7.68.1 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 14 of file LexicographicNumberIterator.hpp.

7.68.2 Constructor & Destructor Documentation

7.68.2.1 LexicographicNumberIterator::LexicographicNumberIterator (`unsigned int upperBound`)

Definition at line 6 of file LexicographicNumberIterator.cpp.

References `initialise()`, and `multiscale::NumberIterator::reset()`.

7.68.2.2 LexicographicNumberIterator::~LexicographicNumberIterator()

Definition at line 11 of file LexicographicNumberIterator.cpp.

References currentNumberDigits, and upperBoundDigits.

7.68.3 Member Function Documentation**7.68.3.1 unsigned int LexicographicNumberIterator::digitsToNumber(vector< unsigned char > & digits) [private]**

Convert the 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().

7.68.3.2 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().

7.68.3.3 void LexicographicNumberIterator::initialise() [private, virtual]

Initialise the 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().

7.68.3.4 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()`.

7.68.3.5 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()`.

7.68.3.6 void LexicographicNumberIterator::numberToDigits(unsigned int *number*, vector< unsigned char > & *digits*) [private]

Convert the number to a 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()`.

7.68.3.7 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()`.

7.68.3.8 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`.

7.68.3.9 void LexicographicNumberIterator::reverseDigits(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()`.

7.68.4 Member Data Documentation

7.68.4.1 vector<unsigned char> multiscale::LexicographicNumberIterator::currentNumberDigits [private]

The digits of the number to which the iterator points

Definition at line 19 of file LexicographicNumberIterator.hpp.

Referenced by `hasNextInitialised()`, `initialise()`, `isLargerThanUpperBound()`, `number()`, `padWithZeros()`, `resetCurrentNumber()`, and `~LexicographicNumberIterator()`.

7.68.4.2 vector<unsigned char> multiscale::LexicographicNumberIterator::upperBoundDigits [private]

The digits of the upper bound

Definition at line 18 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](#)

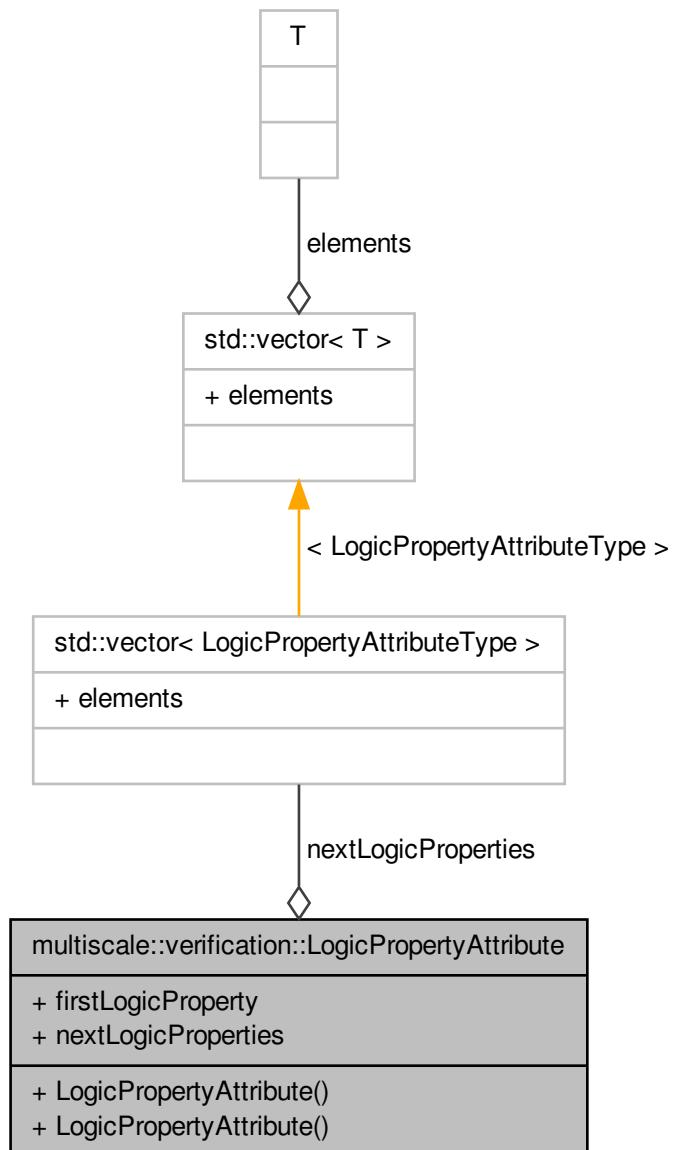
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/-LexicographicNumberIterator.cpp](#)

7.69 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`

7.69.1 Detailed Description

Class for representing a logic property attribute.

Definition at line 40 of file LogicPropertyAttribute.hpp.

7.69.2 Constructor & Destructor Documentation

7.69.2.1 multiscale::verification::LogicPropertyAttribute::LogicPropertyAttribute () [inline]

Definition at line 49 of file LogicPropertyAttribute.hpp.

7.69.2.2 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`.

7.69.3 Member Data Documentation

7.69.3.1 LogicPropertyAttributeType multiscale::verification::LogicProperty- Attribute::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()()`.

7.69.3.2 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:

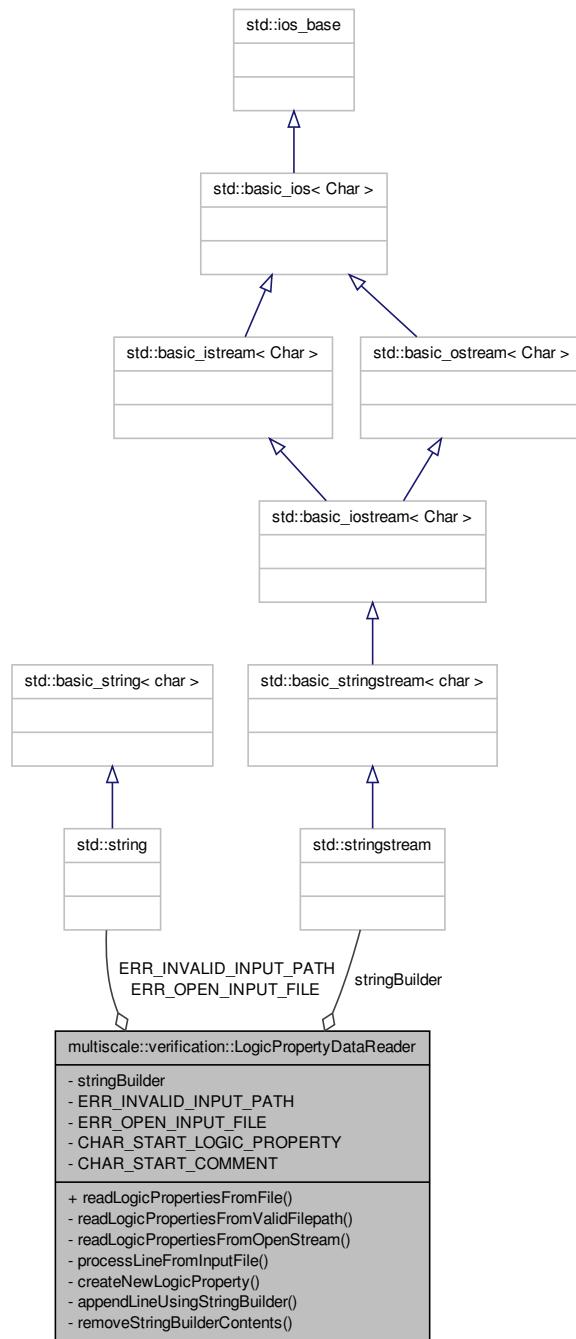
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[LogicPropertyAttribute.hpp](#)

7.70 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 &inputFilepath)

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** = '#'

7.70.1 Detailed Description

Class used to input logic properties.

Definition at line 15 of file LogicPropertyDataReader.hpp.

7.70.2 Member Function Documentation

7.70.2.1 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().

7.70.2.2 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().

7.70.2.3 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()`.

7.70.2.4 `std::vector< std::string > LogicPropertyDataReader::readLogicPropertiesFromFile (const std::string & filepath)`

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

<code><i>filepath</i></code>	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()`.

7.70.2.5 `std::vector< std::string > LogicPropertyDataReader::readLogicPropertiesFromOpenStream (std::ifstream & fin) [private]`

Read the logic properties from the given already opened input stream.

Parameters

<code><i>fin</i></code>	The input stream
-------------------------	------------------

Definition at line 31 of file `LogicPropertyDataReader.cpp`.

References `createNewLogicProperty()`, and `processLineFromInputFile()`.

Referenced by `readLogicPropertiesFromValidFilepath()`.

7.70.2.6 `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()`.

7.70.2.7 **void LogicPropertyDataReader::removeStringBuilderContents()**
[private]

Remove the contents of the string builder.

Definition at line 74 of file LogicPropertyDataReader.cpp.

References stringBuilder.

Referenced by createNewLogicProperty().

7.70.3 Member Data Documentation

7.70.3.1 **const char LogicPropertyDataReader::CHAR_START_COMMENT = '#'**
[static, private]

Definition at line 78 of file LogicPropertyDataReader.hpp.

Referenced by processLineFromInputFile().

7.70.3.2 **const char LogicPropertyDataReader::CHAR_START_LOGIC_PROPERTY
= 'P'** [static, private]

Definition at line 77 of file LogicPropertyDataReader.hpp.

Referenced by processLineFromInputFile().

7.70.3.3 **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().

7.70.3.4 **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().

7.70.3.5 **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](#)

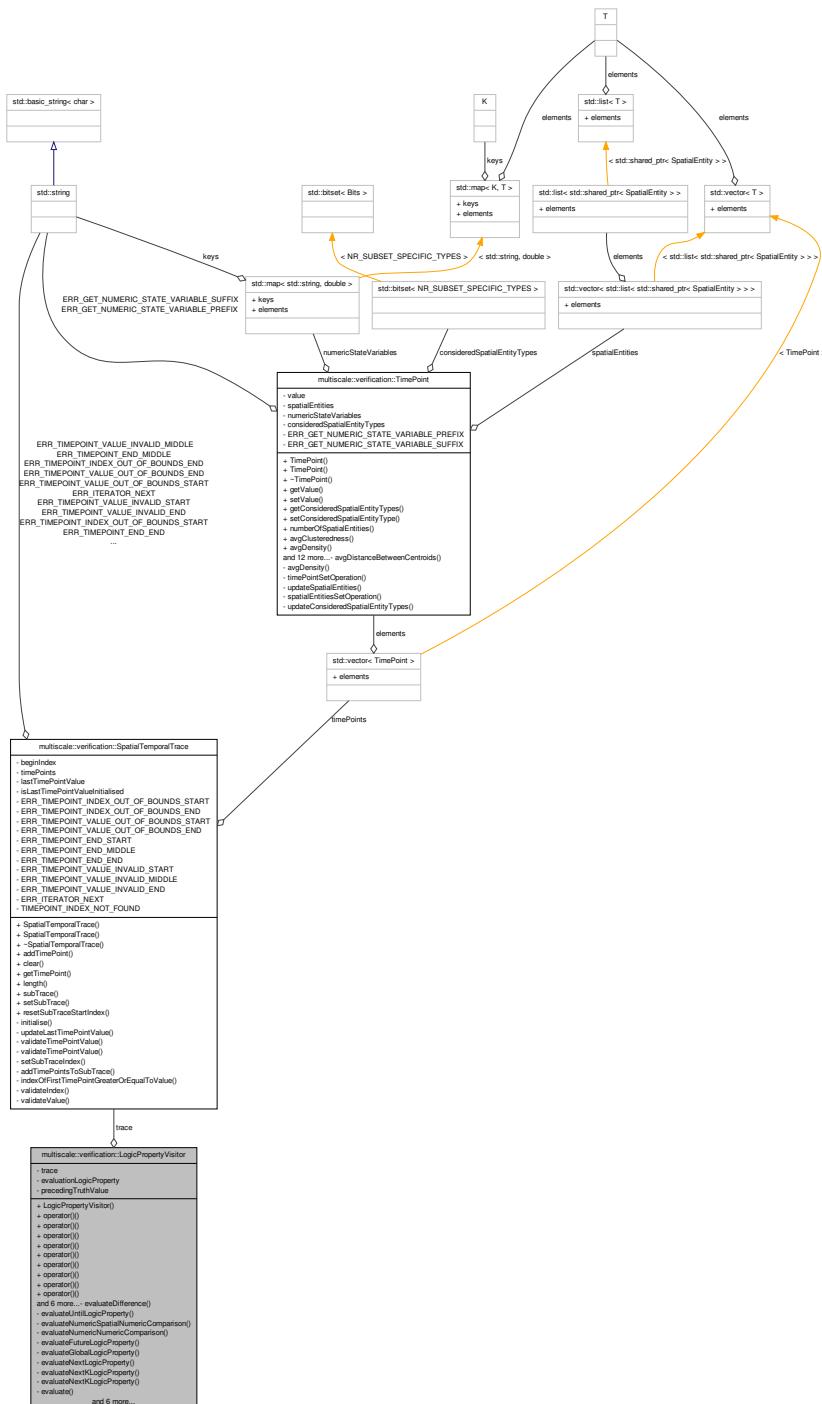
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[LogicPropertyDataReader.cpp](#)

7.71 multiscale::verification::LogicPropertyVisitor Class Reference

Class used to evaluate logic properties.

```
#include <LogicPropertyVisitor.hpp>
```

Collaboration diagram for multiscale::verification::LogicPropertyVisitor:



Public Member Functions

- `LogicPropertyVisitor` (const `SpatialTemporalTrace &trace`, 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 `DifferenceAttribute &primaryLogicProperty`, const `T &lhsLogicProperty`) const
Overloading the "(" operator for the `DifferenceAttribute` alternative.
- template<typename T>
`bool operator()` (const `NumericSpatialNumericComparisonAttribute &primaryLogicProperty`, const `T &lhsLogicProperty`) const
Overloading the "(" operator for the `NumericSpatialNumericComparisonAttribute` alternative.
- template<typename T>
`bool operator()` (const `NumericNumericComparisonAttribute &primaryLogicProperty`, const `T &lhsLogicProperty`) const

Overloading the "(" operator for the NumericNumericComparisonAttribute 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 **evaluateDifference** (const DifferenceAttribute &differenceAttribute, const T &lhsLogicProperty) const

Evaluate the given DifferenceAttribute.

- template<typename T >
bool **evaluateUntilLogicProperty** (const UntilLogicPropertyAttribute &untilLogicProperty, const T &lhsLogicProperty) const

Evaluate the given UntilLogicPropertyAttribute.

- template<typename T >
bool **evaluateNumericSpatialNumericComparison** (const NumericSpatialNumericComparisonAttribute &comparisonAttribute, const T &lhsLogicProperty) const

Evaluate the given NumericSpatialNumericComparisonAttribute.

- template<typename T >
bool **evaluateNumericNumericComparison** (const NumericNumericComparisonAttribute &comparisonAttribute, const T &lhsLogicProperty) const

Evaluate the given NumericNumericComparisonAttribute.

- 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.
- `double evaluateNumericMeasure (const NumericMeasureAttributeType &numericMeasure, const SpatialTemporalTrace &trace, unsigned int timePointIndex=0) const`
Evaluate the numeric measure considering the given spatial temporal trace.
- `double evaluateNumericSpatialMeasure (const NumericSpatialAttributeType &numericSpatialMeasure, const SpatialTemporalTrace &trace, unsigned int timePointIndex=0) const`
Evaluate the numeric spatial measure considering the given spatial temporal trace.
- `bool printExceptionMessage (const char *message) const`
Print a warning message regarding the exception and return false.

Private Attributes

- const `SpatialTemporalTrace` & `trace`
- `LogicPropertyAttributeType` `evaluationLogicProperty`
- bool `precedingTruthValue`

7.71.1 Detailed Description

Class used to evaluate logic properties.

Definition at line 20 of file LogicPropertyVisitor.hpp.

7.71.2 Constructor & Destructor Documentation

7.71.2.1 multiscale::verification::LogicPropertyVisitor::LogicPropertyVisitor
`(const SpatialTemporalTrace & trace, bool precedingTruthValue = true)`
`[inline]`

Definition at line 32 of file LogicPropertyVisitor.hpp.

Referenced by `evaluate()`, and `evaluateNextLogicProperties()`.

7.71.3 Member Function Documentation

7.71.3.1 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

<code>logic- Property</code>	The logic property containing the currently evaluated logic subproperty
<code>evaluation- Logic- Properties</code>	The logic properties preceding the currently evaluated logic subproperty

Definition at line 475 of file LogicPropertyVisitor.hpp.

References `multiscale::verification::LogicPropertyAttribute::firstLogicProperty`.

Referenced by `evaluateNextLogicProperties()`.

7.71.3.2 **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>logic- Property</i>	The logic property
<i>trace</i>	The given spatial temporal trace

Definition at line 427 of file LogicPropertyVisitor.hpp.

References evaluationLogicProperty, and LogicPropertyVisitor().

Referenced by evaluateDifference(), evaluateFutureLogicProperty(), evaluateGlobalLogicProperty(), evaluateNextKLogicProperty(), evaluateNumericNumericComparison(), evaluateNumericSpatialNumericComparison(), evaluatePrecedingLogicProperties(), evaluateUntilLogicProperty(), and operator()().

7.71.3.3 **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>primary- Logic- Property</i>	The primary logic property
<i>trace</i>	The given spatial temporal trace

Definition at line 437 of file LogicPropertyVisitor.hpp.

References evaluationLogicProperty, and LogicPropertyVisitor().

7.71.3.4 **template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateDifference (const DifferenceAttribute & *differenceAttribute*, const T & *lhsLogicProperty*) const [inline, private]**

Evaluate the given [DifferenceAttribute](#).

Parameters

<i>difference- Attribute</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 271 of file LogicPropertyVisitor.hpp.

References multiscale::verification::DifferenceAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, evaluate(), evaluateNumericMeasure(), multiscale::verification::DifferenceAttribute::lhsNumericMeasure, multiscale::verification::DifferenceAttribute::rhsNumericMeasure, and trace.

Referenced by operator()().

```
7.71.3.5 template<typename T> bool multiscale::verification::LogicPropertyVisitor-
    ::evaluateFutureLogicProperty( const FutureLogicPropertyAttribute
    & futureLogicProperty, const T & lhsLogicProperty ) const [inline,
    private]
```

Evaluate the given [FutureLogicPropertyAttribute](#).

Parameters

<i>futureLogic- Property</i>	The future logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 343 of file LogicPropertyVisitor.hpp.

References multiscale::verification::FutureLogicPropertyAttribute::endTimepoint, evaluate(), multiscale::verification::FutureLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::setSubTrace(), multiscale::verification::FutureLogicPropertyAttribute::startTimepoint, and trace.

Referenced by operator()().

```
7.71.3.6 template<typename T> bool multiscale::verification::LogicPropertyVisitor-
    ::evaluateGlobalLogicProperty( const GlobalLogicPropertyAttribute
    & globalLogicProperty, const T & lhsLogicProperty ) const [inline,
    private]
```

Evaluate the given [GlobalLogicPropertyAttribute](#).

Parameters

<i>globalLogic- Property</i>	The global logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 367 of file LogicPropertyVisitor.hpp.

References multiscale::verification::GlobalLogicPropertyAttribute::endTimepoint, evaluate(), multiscale::verification::GlobalLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::setSubTrace(), multiscale::verification::Global-

LogicPropertyAttribute::startTimepoint, and trace.

Referenced by operator()().

```
7.71.3.7 template<typename T> bool multiscale::verification::LogicPropertyVisitor-  
::evaluateNextKLogicProperty ( const NextKLogicPropertyAttribute  
& nextKLogicProperty, const T & lhsLogicProperty ) const [inline,  
private]
```

Evaluate the given [NextKLogicPropertyAttribute](#).

Parameters

<i>nextKLogic- Property</i>	The next "k" logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 402 of file LogicPropertyVisitor.hpp.

References [multiscale::verification::NextKLogicPropertyAttribute::logicProperty](#), [multiscale::verification::NextKLogicPropertyAttribute::nrOfTimepointsAhead](#), and [trace](#).

Referenced by [evaluateNextLogicProperty\(\)](#), and [operator\(\)\(\)](#).

```
7.71.3.8 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>logic- Property</i>	The logic property enclosed by the next "k" logic property
<i>lhsLogic- Property</i>	The left hand side logic property
<i>kValue</i>	The value of "k"

Definition at line 415 of file LogicPropertyVisitor.hpp.

References [evaluate\(\)](#), [multiscale::verification::SpatialTemporalTrace::subTrace\(\)](#), and [trace](#).

```
7.71.3.9 bool multiscale::verification::LogicPropertyVisitor::evaluateNextLogic-  
Properties ( 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>logic- Property</i>	The given logic property
<i>truthValue</i>	The given truth value

Definition at line 450 of file LogicPropertyVisitor.hpp.

References `constructEvaluationLogicProperty()`, `LogicPropertyVisitor()`, `multiscale-
::verification::LogicPropertyAttribute::nextLogicProperties`, and `trace`.

Referenced by `operator()()`.

**7.71.3.10 template<typename T > bool multiscale::verification::LogicPropertyVisitor-
::evaluateNextLogicProperty (const NextLogicPropertyAttribute &
nextLogicProperty, const T & lhsLogicProperty) const [inline, private]**

Evaluate the given [NextLogicPropertyAttribute](#).

Parameters

<i>nextLogic- Property</i>	The next logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 391 of file LogicPropertyVisitor.hpp.

References `evaluateNextKLogicProperty()`, `multiscale::verification::NextLogicProperty-
Attribute::logicProperty`, and `trace`.

Referenced by `operator()()`.

**7.71.3.11 double multiscale::verification::LogicPropertyVisitor::evaluateNumeric-
Measure (const NumericMeasureAttributeType & numericMeasure, const
SpatialTemporalTrace & trace, unsigned int timePointIndex = 0) const
[inline, private]**

Evaluate the numeric measure considering the given spatial temporal trace.

Parameters

<i>numeric- Measure</i>	The given numeric measure
<i>trace</i>	The given spatial temporal trace
<i>timePoint- Index</i>	The index of the considered timepoint from the trace

Definition at line 510 of file LogicPropertyVisitor.hpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint().

Referenced by evaluateDifference(), evaluateNumericNumericComparison(), and evaluateNumericSpatialNumericComparison().

```
7.71.3.12 template<typename T > bool multiscale::verification::Logic-
PropertyVisitor::evaluateNumericNumericComparison ( const
NumericNumericComparisonAttribute & comparisonAttribute, const T &
lhsLogicProperty ) const [inline, private]
```

Evaluate the given [NumericNumericComparisonAttribute](#).

Parameters

<i>comparison-Attribute</i>	The numeric numeric comparison attribute
<i>lhsLogic-Property</i>	The left hand side logic property

Definition at line 327 of file LogicPropertyVisitor.hpp.

References multiscale::verification::NumericNumericComparisonAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, evaluate(), evaluate-NumericMeasure(), multiscale::verification::NumericNumericComparisonAttribute::numericMeasure, multiscale::verification::NumericNumericComparisonAttribute::numericStateVariable, and trace.

Referenced by operator()().

```
7.71.3.13 double multiscale::verification::LogicPropertyVisitor::evaluate-
NumericSpatialMeasure ( const NumericSpatialAttributeType &
numericSpatialMeasure, const SpatialTemporalTrace & trace, unsigned int
timePointIndex = 0 ) const [inline, private]
```

Evaluate the numeric spatial measure considering the given spatial temporal trace.

Parameters

<i>numeric-Spatial-Measure</i>	The given numeric spatial measure
<i>trace</i>	The given spatial temporal trace
<i>timePoint-Index</i>	The index of the considered timepoint from the trace

Definition at line 524 of file LogicPropertyVisitor.hpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint().

Referenced by evaluateNumericSpatialNumericComparison().

7.71.3.14 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatialNumericComparison (const NumericSpatialNumericComparisonAttribute & *comparisonAttribute*, const T & *lhsLogicProperty*) const [inline, private]

Evaluate the given [NumericSpatialNumericComparisonAttribute](#).

Parameters

<i>comparisonAttribute</i>	The numeric spatial numeric comparison attribute
<i>lhsLogicProperty</i>	The left hand side logic property

Definition at line 311 of file LogicPropertyVisitor.hpp.

References multiscale::verification::NumericSpatialNumericComparisonAttribute::comparator, multiscale::verification::ComparatorAttribute::comparatorType, evaluate(), evaluateNumericMeasure(), evaluateNumericSpatialMeasure(), multiscale::verification::NumericSpatialNumericComparisonAttribute::numericMeasure, multiscale::verification::NumericSpatialNumericComparisonAttribute::numericSpatialMeasure, and trace.

Referenced by operator()().

7.71.3.15 bool multiscale::verification::LogicPropertyVisitor::evaluatePrecedingLogicProperties (unsigned long *startTime*, unsigned long *endTime*, const LogicPropertyAttributeType & *precedingLogicProperties*) const [inline, private]

Evaluate the preceding logic properties.

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 489 of file LogicPropertyVisitor.hpp.

References evaluate(), multiscale::verification::SpatialTemporalTrace::setSubTrace(), and trace.

Referenced by evaluateUntilLogicProperty().

7.71.3.16 template<typename T > bool multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty (const UntilLogicPropertyAttribute & *untilLogicProperty*, const T & *lhsLogicProperty*) const [inline, private]

Evaluate the given [UntilLogicPropertyAttribute](#).

Parameters

<i>untilLogic-Property</i>	The until logic property
<i>IhsLogic-Property</i>	The left hand side logic property

Definition at line 287 of file LogicPropertyVisitor.hpp.

References multiscale::verification::UntilLogicPropertyAttribute::endTimepoint, evaluate(), evaluatePrecedingLogicProperties(), multiscale::verification::UntilLogicPropertyAttribute::logicProperty, multiscale::verification::SpatialTemporalTrace::setSubTrace(), multiscale::verification::UntilLogicPropertyAttribute::startTimepoint, and trace.

Referenced by operator()().

**7.71.3.17 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator()
(const Nil & *logicProperty*, const T & *IhsLogicProperty*) const [inline]**

Overloading the "(") operator for the [Nil](#) alternative.

Parameters

<i>logic-Property</i>	The logic property
<i>IhsLogic-Property</i>	The left hand side logic property

Definition at line 41 of file LogicPropertyVisitor.hpp.

**7.71.3.18 template<typename T > bool multiscale::verification::LogicPropertyVisitor::operator()
(const LogicPropertyAttribute & *logicProperty*, const T & *IhsLogicProperty*)
const [inline]**

Overloading the "(") operator for the [LogicPropertyAttribute](#) alternative.

Parameters

<i>logic-Property</i>	The logic property
<i>IhsLogic-Property</i>	The left hand side logic property

Definition at line 51 of file LogicPropertyVisitor.hpp.

References evaluate(), evaluateNextLogicProperties(), multiscale::verification::LogicPropertyAttribute::firstLogicProperty, and trace.

```
7.71.3.19 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>logic- Property</i>	The logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 65 of file LogicPropertyVisitor.hpp.

References evaluate(), multiscale::verification::OrLogicPropertyAttribute::logicProperty, precedingTruthValue, and trace.

```
7.71.3.20 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>logic- Property</i>	The logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 82 of file LogicPropertyVisitor.hpp.

References evaluate(), multiscale::verification::AndLogicPropertyAttribute::logicProperty, precedingTruthValue, and trace.

```
7.71.3.21 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>logic- Property</i>	The logic property
----------------------------	--------------------

<i>IhsLogic-Property</i>	The left hand side logic property
--------------------------	-----------------------------------

Definition at line 99 of file LogicPropertyVisitor.hpp.

References evaluate(), multiscale::verification::ImplicationLogicPropertyAttribute::logic-Property, precedingTruthValue, and trace.

```
7.71.3.22 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const EquivalenceLogicPropertyAttribute & logicProperty, const T &
IhsLogicProperty ) const [inline]
```

Overloading the "()" operator for the [EquivalenceLogicPropertyAttribute](#) alternative.

Parameters

<i>logic-Property</i>	The logic property
<i>IhsLogic-Property</i>	The left hand side logic property

Definition at line 114 of file LogicPropertyVisitor.hpp.

References evaluate(), multiscale::verification::EquivalenceLogicPropertyAttribute::logicProperty, precedingTruthValue, and trace.

```
7.71.3.23 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>logic-Property</i>	The logic property
<i>IhsLogic-Property</i>	The left hand side logic property

Definition at line 128 of file LogicPropertyVisitor.hpp.

References evaluateUntilLogicProperty(), printExceptionMessage(), and multiscale::-MultiscaleException::what().

```
7.71.3.24 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const PrimaryLogicPropertyAttribute & logicProperty, const T &
lhsLogicProperty ) const [inline]
```

Overloading the "(" operator for the [PrimaryLogicPropertyAttribute](#) alternative.

Parameters

<i>logic- Property</i>	The logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 142 of file LogicPropertyVisitor.hpp.

References [evaluate\(\)](#), [multiscale::verification::PrimaryLogicPropertyAttribute::primary-LogicProperty](#), and [trace](#).

```
7.71.3.25 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const DifferenceAttribute & primaryLogicProperty, const T & lhsLogicProperty )
const [inline]
```

Overloading the "(" operator for the [DifferenceAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 152 of file LogicPropertyVisitor.hpp.

References [evaluateDifference\(\)](#), [printExceptionMessage\(\)](#), and [multiscale::Multiscale-Exception::what\(\)](#).

```
7.71.3.26 template<typename T> bool multiscale::verification::LogicPropertyVisitor-
::operator() ( const NumericSpatialNumericComparisonAttribute &
primaryLogicProperty, const T & lhsLogicProperty ) const [inline]
```

Overloading the "(" operator for the [NumericSpatialNumericComparisonAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 166 of file LogicPropertyVisitor.hpp.

References `evaluateNumericSpatialNumericComparison()`, `printExceptionMessage()`, and `multiscale::MultiscaleException::what()`.

```
7.71.3.27 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const NumericNumericComparisonAttribute & primaryLogicProperty, const
T & lhsLogicProperty ) const [inline]
```

Overloading the "()" operator for the [NumericNumericComparisonAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 181 of file LogicPropertyVisitor.hpp.

References `evaluateNumericNumericComparison()`, `printExceptionMessage()`, and `multiscale::MultiscaleException::what()`.

```
7.71.3.28 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const NotLogicPropertyAttribute & primaryLogicProperty, const T &
lhsLogicProperty ) const [inline]
```

Overloading the "()" operator for the [NotLogicPropertyAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 196 of file LogicPropertyVisitor.hpp.

References `evaluate()`, `multiscale::verification::NotLogicPropertyAttribute::logic-
Property`, and `trace`.

```
7.71.3.29 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const FutureLogicPropertyAttribute & primaryLogicProperty, const T &
lhsLogicProperty ) const [inline]
```

Overloading the "()" operator for the [FutureLogicPropertyAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 209 of file LogicPropertyVisitor.hpp.

References evaluateFutureLogicProperty(), printExceptionMessage(), and multiscale::-MultiscaleException::what().

```
7.71.3.30 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const GlobalLogicPropertyAttribute & primaryLogicProperty, const T &
lhsLogicProperty ) const [inline]
```

Overloading the "(" operator for the [GlobalLogicPropertyAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 224 of file LogicPropertyVisitor.hpp.

References evaluateGlobalLogicProperty(), printExceptionMessage(), and multiscale::-MultiscaleException::what().

```
7.71.3.31 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
( const NextLogicPropertyAttribute & primaryLogicProperty, const T &
lhsLogicProperty ) const [inline]
```

Overloading the "(" operator for the [NextLogicPropertyAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 239 of file LogicPropertyVisitor.hpp.

References evaluateNextLogicProperty(), printExceptionMessage(), and multiscale::-MultiscaleException::what().

7.71.3.32 template<typename T> bool multiscale::verification::LogicPropertyVisitor::operator()
 (const NextKLogicPropertyAttribute & *primaryLogicProperty*, const T &
lhsLogicProperty) const [inline]

Overloading the "(" operator for the [NextKLogicPropertyAttribute](#) alternative.

Parameters

<i>primary- Logic- Property</i>	The primary logic property
<i>lhsLogic- Property</i>	The left hand side logic property

Definition at line 254 of file LogicPropertyVisitor.hpp.

References [evaluateNextKLogicProperty\(\)](#), [printExceptionMessage\(\)](#), and [multiscale::MultiscaleException::what\(\)](#).

7.71.3.33 bool multiscale::verification::LogicPropertyVisitor::print-
 ExceptionMessage (const char * *message*) const [inline,
 private]

Print a warning message regarding the exception and return false.

Parameters

<i>message</i>	The exception message
----------------	-----------------------

Definition at line 536 of file LogicPropertyVisitor.hpp.

References [multiscale::ConsolePrinter::printWarningMessage\(\)](#), and [multiscale::verification::WRN_LOGIC_PROPERTY_EVAL_FALSE](#).

Referenced by [operator\(\)\(\)](#).

7.71.4 Member Data Documentation

7.71.4.1 **LogicPropertyAttributeType multiscale::verification-
 ::LogicPropertyVisitor::evaluationLogicProperty
 [private]**

The logic property used only for evaluation purposes

Definition at line 25 of file LogicPropertyVisitor.hpp.

Referenced by [evaluate\(\)](#).

7.71.4.2 `bool multiscale::verification::LogicPropertyVisitor::precedingTruthValue [private]`

The truth value of the preceding logic property

Definition at line 28 of file LogicPropertyVisitor.hpp.

Referenced by operator()().

7.71.4.3 `const SpatialTemporalTrace& multiscale::verification::LogicPropertyVisitor::trace [private]`

The spatial temporal trace

Definition at line 24 of file LogicPropertyVisitor.hpp.

Referenced by evaluateDifference(), evaluateFutureLogicProperty(), evaluateGlobalLogicProperty(), evaluateNextKLogicProperty(), evaluateNextLogicProperties(), evaluateNextLogicProperty(), evaluateNumericNumericComparison(), evaluateNumericSpatialNumericComparison(), evaluatePrecedingLogicProperties(), evaluateUntilLogicProperty(), and operator()().

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

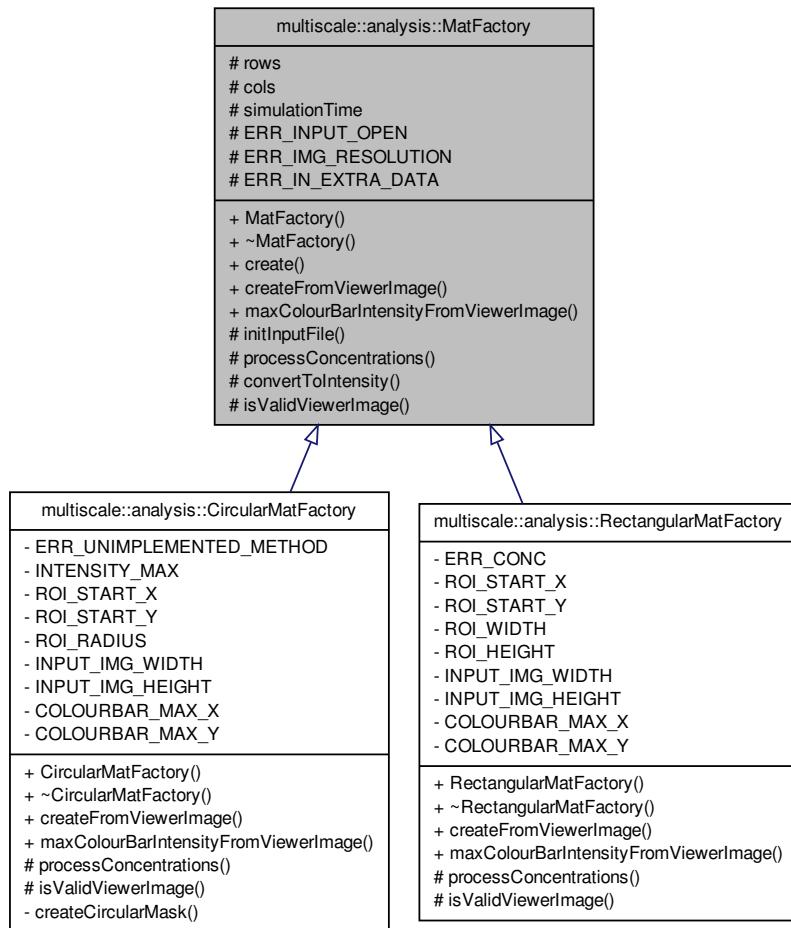
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[LogicPropertyVisitor.hpp](#)

7.72 multiscale::analysis::MatFactory Class Reference

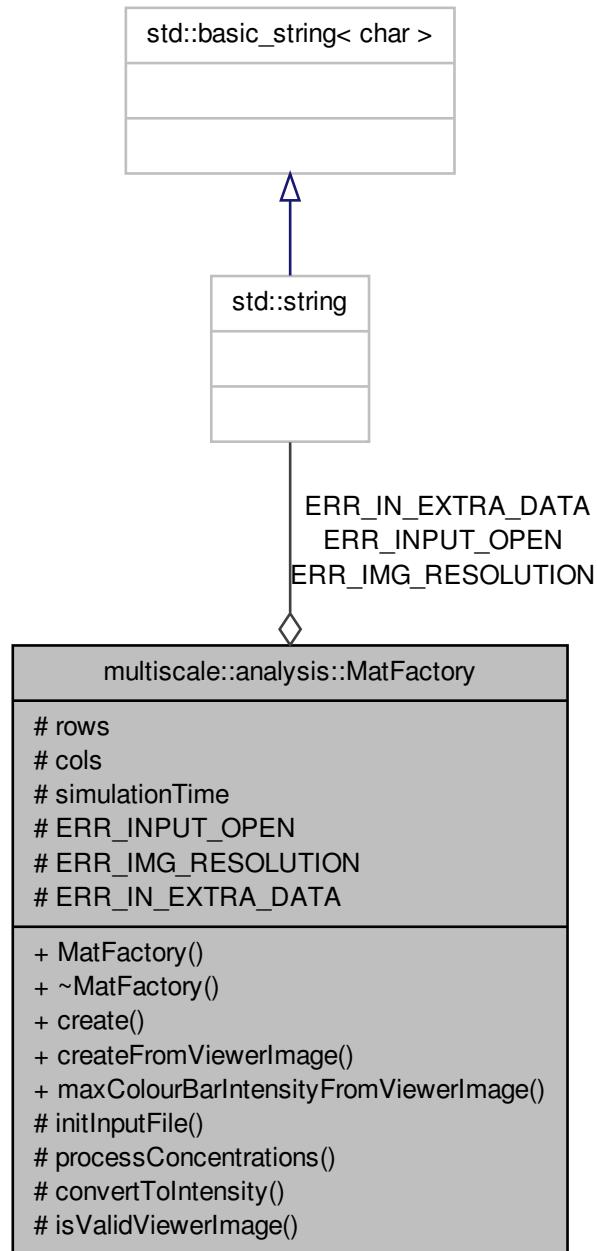
Class for creating a Mat object.

```
#include <MatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::MatFactory:



Collaboration diagram for multiscale::analysis::MatFactory:



Public Member Functions

- `MatFactory ()`
- `virtual ~MatFactory ()`
- `Mat create (const string &inputFile)`
`Create a Mat object from the input file.`
- `virtual Mat createFromViewerImage (const string &inputFile)=0`
`Create a Mat object from the image file obtained from Rectangular/CircularGeometry-Viewer.`
- `virtual double maxColourBarIntensityFromViewerImage (const string &inputFile)=0`
`Get the maximum grayscale intensity of the colour bar in the image.`

Protected Member Functions

- `void initInputFile (ifstream &fin, const string &inputFile)`
`Initialise the input file.`
- `virtual unsigned char * processConcentrations (ifstream &fin)=0`
`Process concentrations from file.`
- `unsigned char convertToIntensity (double concentration)`
`Convert concentration to intensity.`
- `virtual bool isValidViewerImage (const 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 string ERR_INPUT_OPEN = "The input file could not be opened."`
- `static const string ERR_IMG_RESOLUTION = "The resolution of the input image is not the expected one."`
- `static const string ERR_IN_EXTRA_DATA = "The input file contains more data than required."`

7.72.1 Detailed Description

Class for creating a Mat object.

Definition at line 16 of file MatFactory.hpp.

7.72.2 Constructor & Destructor Documentation

7.72.2.1 MatFactory::MatFactory()

Definition at line 10 of file MatFactory.cpp.

7.72.2.2 MatFactory::~MatFactory() [virtual]

Definition at line 12 of file MatFactory.cpp.

7.72.3 Member Function Documentation

7.72.3.1 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>concentra-</i> <i>tion</i>	A value between 0 and 1
----------------------------------	-------------------------

Definition at line 44 of file MatFactory.cpp.

Referenced by multiscale::analysis::RectangularMatFactory::processConcentrations().

7.72.3.2 Mat MatFactory::create (const string & *inputFile*)

Create a Mat object from the input file.

Create the 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.

7.72.3.3 virtual Mat multiscale::analysis::MatFactory::createFromViewerImage (const string & *inputFile*) [pure virtual]

Create a Mat object from the image file obtained from Rectangular/CircularGeometry-Viewer.

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](#).

7.72.3.4 void MatFactory::initInputFile (ifstream & *fin*, const 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().

7.72.3.5 virtual bool multiscale::analysis::MatFactory::isValidViewerImage (const 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](#).

7.72.3.6 virtual double multiscale::analysis::MatFactory::maxColourBar-IntensityFromViewerImage (const string & *inputFile*) [pure virtual]

Get the maximum grayscale intensity of the colour bar in the image.

Parameters

<code>inputFile</code>	The path to the image file
------------------------	----------------------------

Implemented in [multiscale::analysis::CircularMatFactory](#), and [multiscale::analysis::RectangularMatFactory](#).

7.72.3.7 virtual unsigned char* multiscale::analysis::MatFactory::processConcentrations (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()`.

7.72.4 Member Data Documentation

7.72.4.1 int multiscale::analysis::MatFactory::cols [protected]

Number of columns in the Mat object

Definition at line 21 of file `MatFactory.hpp`.

Referenced by `create()`, `initInputFile()`, and [multiscale::analysis::RectangularMatFactory::processConcentrations\(\)](#).

7.72.4.2 const string MatFactory::ERR_IMG_RESOLUTION = "The resolution of the input image is not the expected one." [static, protected]

Definition at line 91 of file `MatFactory.hpp`.

Referenced by [multiscale::analysis::CircularMatFactory::isValidViewerImage\(\)](#), and [multiscale::analysis::RectangularMatFactory::isValidViewerImage\(\)](#).

7.72.4.3 const string MatFactory::ERR_IN_EXTRA_DATA = "The input file contains more data than required." [static, protected]

Definition at line 92 of file `MatFactory.hpp`.

Referenced by `create()`.

7.72.4.4 **const string MatFactory::ERR_INPUT_OPEN = "The input file could not be opened."** [static, protected]

Definition at line 90 of file MatFactory.hpp.

Referenced by initInputFile(), multiscale::analysis::CircularMatFactory::isValidViewerImage(), and multiscale::analysis::RectangularMatFactory::isValidViewerImage().

7.72.4.5 **int multiscale::analysis::MatFactory::rows** [protected]

Number of rows in the Mat object

Definition at line 20 of file MatFactory.hpp.

Referenced by create(), initInputFile(), and multiscale::analysis::RectangularMatFactory::processConcentrations().

7.72.4.6 **double multiscale::analysis::MatFactory::simulationTime**
[protected]

Simulation time read from the input file

Definition at line 22 of file MatFactory.hpp.

Referenced by initInputFile().

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

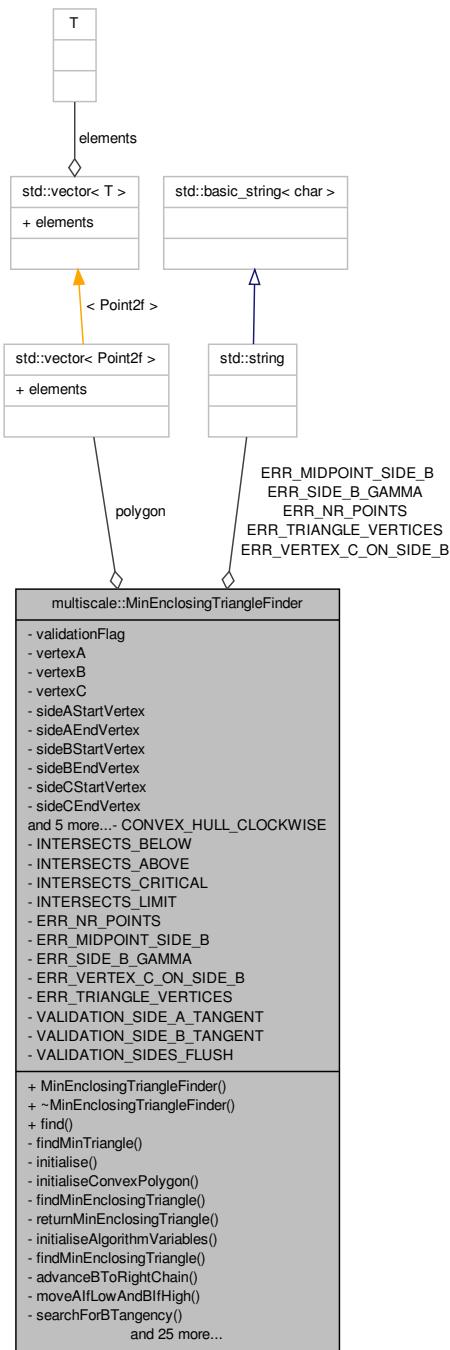
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/sp MatFactory.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-
MatFactory.cpp

7.73 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 vector< Point2f > &points, vector< Point2f > &minEnclosingTriangle)

Find the minimum area enclosing triangle for the given 2D point set.

Private Member Functions

- double [findMinTriangle](#) (const vector< Point2f > &points, vector< Point2f > &minEnclosingTriangle)

Find the minimum area enclosing triangle for the given 2D point set.

- void [initialise](#) (const vector< Point2f > &points, vector< Point2f > &minEnclosingTriangle)

Initialisation function for the class.

- void [initialiseConvexPolygon](#) (const vector< Point2f > &points)

Initialise polygon as the convex hull of the given set of points.

- double [findMinEnclosingTriangle](#) (const vector< Point2f > &polygon, vector< - Point2f > &minEnclosingTriangle)

Find the minimum area enclosing triangle for the given polygon.

- double [returnMinEnclosingTriangle](#) (const vector< Point2f > &polygon, vector< 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](#) (vector< 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 or not.
- bool `isValidMinimalTriangle ()`
Check if the found minimal triangle is valid.
- void `updateMinEnclosingTriangle (vector< Point2f > &minEnclosingTriangle, double &minEnclosingTriangleArea)`
Update the current minimum area enclosing triangle if the newly obtained one has a smaller area.
- bool `middlePointOfSideB (Point2f &middlePointOfSideB)`
Return the middle point of side B.
- bool `intersectsBelow (const Point2f &gammaPoint, unsigned int polygonPointIndex)`
Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon below the point polygon[polygonPointIndex].
- bool `intersectsAbove (const Point2f &gammaPoint, unsigned int polygonPointIndex)`
Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon above the point polygon[polygonPointIndex].
- 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 angle of the flush edge or its opposite angle lie between the angle of the predecessor and successor.
- 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 Point2f &polygonPoint)`
Compute the height of the point.
- bool `gamma (unsigned int polygonPointIndex, Point2f &gammaPoint)`
Find gamma for a given point "p" specified by its index.
- 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 Point2f &side1StartVertex, const Point2f &side1EndVertex, const Point2f &side2StartVertex, const Point2f &side2EndVertex, Point2f &intersectionPoint1, Point2f &intersectionPoint2)

Find the intersection points to compute gamma(point)
- bool `areIdenticalLines` (const vector< double > &side1Params, const vector< double > &side2Params, double sideCEExtraParam)

Check if the given lines are identical or not.
- bool `areIntersectingLines` (const vector< double > &side1Params, const vector< double > &side2Params, double sideCEExtraParam, Point2f &intersectionPoint1, Point2f &intersectionPoint2)

Check if the given lines intersect or not. If the lines intersect find their intersection points.
- vector< double > `lineEquationParameters` (const Point2f &p, const 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 successor of the provided point index.
- unsigned int `predecessor` (unsigned int index)

Return the predecessor of the provided point index.

Private Attributes

- unsigned int `validationFlag`
- Point2f `vertexA`
- Point2f `vertexB`
- Point2f `vertexC`
- Point2f `sideAStartVertex`
- Point2f `sideAEndVertex`
- Point2f `sideBStartVertex`
- Point2f `sideBEndVertex`
- Point2f `sideCStartVertex`
- Point2f `sideCEndVertex`
- double `area`
- unsigned int `a`
- unsigned int `b`
- unsigned int `c`
- unsigned int `nrOfPoints`
- vector< 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 string `ERR_NR_POINTS` = "The number of 2D points in the input vector should be greater than 0."
- static const string `ERR_MIDPOINT_SIDE_B` = "The position of the middle point of side B could not be determined."
- static const string `ERR_SIDE_B_GAMMA` = "The position of side B could not be determined, because `gamma(b)` could not be computed."
- static const 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 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

7.73.1 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 20 of file `MinEnclosingTriangleFinder.hpp`.

7.73.2 Constructor & Destructor Documentation

7.73.2.1 `MinEnclosingTriangleFinder::MinEnclosingTriangleFinder()`

Definition at line 13 of file `MinEnclosingTriangleFinder.cpp`.

References `a`, `area`, `b`, `c`, `nrOfPoints`, and `validationFlag`.

7.73.2.2 `MinEnclosingTriangleFinder::~MinEnclosingTriangleFinder()`

Definition at line 25 of file `MinEnclosingTriangleFinder.cpp`.

7.73.3 Member Function Documentation

7.73.3.1 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 415 of file MinEnclosingTriangleFinder.cpp.

References successor().

Referenced by advanceBToRightChain(), moveAIfLowAndBIfHigh(), and searchForBTangency().

7.73.3.2 void MinEnclosingTriangleFinder::advanceBToRightChain ()
 [private]

Advance b to the right chain.

See paper for more details

Definition at line 112 of file MinEnclosingTriangleFinder.cpp.

References advance(), b, multiscale::Numeric::greaterOrEqual(), height(), and successor().

Referenced by findMinEnclosingTriangle().

7.73.3.3 bool MinEnclosingTriangleFinder::areIdenticalLines (const vector< double > & side1Params, const 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>side1-Params</i>	Vector containing the values of a, b and c for side 1
<i>side2-Params</i>	Vector containing the values of a, b and c for side 2
<i>sideCExtra-Param</i>	Extra parameter for the flush edge C

Definition at line 380 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::areIdenticalLines().

Referenced by `findGammaIntersectionPoints()`.

```
7.73.3.4 bool MinEnclosingTriangleFinder::areIntersectingLines ( const vector<
    double > & side1Params, const vector< double > & side2Params, double
    sideCEExtraParam, Point2f & intersectionPoint1, 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 $ax + by + c (+/-) sideCEExtraParam = 0$

Parameters

<i>side1-Params</i>	Vector containing the values of a, b and c for side 1
<i>side2-Params</i>	Vector containing the values of a, b and c for side 2
<i>sideCEExtraParam</i>	Extra parameter for the flush edge C
<i>intersection-Point1</i>	The first intersection point, if it exists
<i>intersection-Point2</i>	The second intersection point, if it exists

Definition at line 390 of file `MinEnclosingTriangleFinder.cpp`.

References `multiscale::Geometry2D::lineIntersection()`.

Referenced by `findGammaIntersectionPoints()`.

```
7.73.3.5 double MinEnclosingTriangleFinder::find ( const vector< Point2f > & points,
    vector< 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>min-Enclosing-Triangle</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()`.

```
7.73.3.6 bool MinEnclosingTriangleFinder::findGammaIntersectionPoints (
    unsigned int polygonPointIndex, const Point2f & side1StartVertex, const Point2f &
    side1EndVertex, const Point2f & side2StartVertex, const Point2f & side2EndVertex,
    Point2f & intersectionPoint1, Point2f & intersectionPoint2 ) [private]
```

Find the intersection points to compute gamma(point)

Parameters

<i>polygon-PointIndex</i>	Index of the polygon point for which the distance is known
<i>side1Start-Vertex</i>	Start vertex for side 1
<i>side1End-Vertex</i>	End vertex for side 1
<i>side2Start-Vertex</i>	Start vertex for side 2
<i>side2End-Vertex</i>	End vertex for side 2
<i>intersection-Point1</i>	First intersection point between one pair of lines
<i>intersection-Point2</i>	Second intersection point between another pair of lines

Definition at line 357 of file MinEnclosingTriangleFinder.cpp.

References areIdenticalLines(), areIntersectingLines(), height(), and lineEquationParameters().

Referenced by findVertexCOnSideB(), and gamma().

```
7.73.3.7 double MinEnclosingTriangleFinder::findMinEnclosingTriangle ( const
    vector< Point2f > & polygon, vector< 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>min-Enclosing-Triangle</i>	Minimum area triangle enclosing the given polygon

Definition at line 61 of file MinEnclosingTriangleFinder.cpp.

References initialiseAlgorithmVariables().

Referenced by findMinTriangle().

```
7.73.3.8 void MinEnclosingTriangleFinder::findMinEnclosingTriangle ( vector<
    Point2f > & minEnclosingTriangle, double & minEnclosingTriangleArea )
    [private]
```

Find the minimum area enclosing triangle for the given polygon.

Parameters

<i>min- Enclosing- Triangle</i>	Minimum area triangle enclosing the given polygon
<i>min- Enclosing- TriangleArea</i>	Area of the minimum area enclosing triangle

Definition at line 91 of file MinEnclosingTriangleFinder.cpp.

References advanceBToRightChain(), c, isLocalMinimalTriangle(), isNotBTangency(), moveAlfLowAndBIfHigh(), nrOfPoints, searchForBTangency(), updateMinEnclosingTriangle(), updateSideB(), updateSidesBA(), and updateSidesCA().

```
7.73.3.9 double MinEnclosingTriangleFinder::findMinTriangle ( const vector< Point2f
    > & points, vector< Point2f > & minEnclosingTriangle ) [private]
```

Find the minimum area enclosing triangle for the given 2D point set.

Parameters

<i>points</i>	Set of points
<i>min- Enclosing- Triangle</i>	Minimum area triangle enclosing the given polygon

Definition at line 38 of file MinEnclosingTriangleFinder.cpp.

References findMinEnclosingTriangle(), initialise(), polygon, and returnMinEnclosingTriangle().

Referenced by find().

```
7.73.3.10 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 340 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().

7.73.3.11 bool MinEnclosingTriangleFinder::gamma (unsigned int *polygonPointIndex*, *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>polygon-PointIndex</i>	Index of the polygon point
<i>gammaPoint</i>	<i>Point2f gamma(polygon[polygonPointIndex])</i>

Definition at line 321 of file MinEnclosingTriangleFinder.cpp.

References a, multiscale::Geometry2D::areOnTheSameSideOfLine(), c, findGammaIntersectionPoints(), polygon, predecessor(), and successor().

Referenced by isNotBTangency(), moveAlfLowAndBlfHigh(), searchForBTangency(), and updateSideB().

7.73.3.12 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>polygon-PointIndex</i>	Index of the polygon point
---------------------------	----------------------------

Definition at line 305 of file MinEnclosingTriangleFinder.cpp.

References c, multiscale::Geometry2D::distanceFromPointToLine(), polygon, and pre-

decessor().

Referenced by advanceBToRightChain(), findGammaIntersectionPoints(), intersectsAboveOrBelow(), isNotBTangency(), moveAIfLowAndBIfHigh(), searchForBTangency(), and updateSidesBA().

7.73.3.13 double MinEnclosingTriangleFinder::height (const Point2f & *polygonPoint*) [private]

Compute the height of the point.

See paper for more details

Parameters

<i>polygonPoint</i>	Polygon point
---------------------	---------------

Definition at line 314 of file MinEnclosingTriangleFinder.cpp.

References c, multiscale::Geometry2D::distanceFromPointToLine(), polygon, and predecessor().

7.73.3.14 void MinEnclosingTriangleFinder::initialise (const vector< Point2f > & *points*, vector< Point2f > & *minEnclosingTriangle*) [private]

Initialisation function for the class.

Initialise the polygon and other class' fields.

Parameters

<i>points</i>	Set of points
<i>min- Enclosing- Triangle</i>	Minimum area triangle enclosing the given polygon

Definition at line 48 of file MinEnclosingTriangleFinder.cpp.

References initialiseConvexPolygon().

Referenced by findMinTriangle().

7.73.3.15 void MinEnclosingTriangleFinder::initialiseAlgorithmVariables () [private]

Initialisation of the algorithm variables.

Definition at line 83 of file MinEnclosingTriangleFinder.cpp.

References a, b, c, nrOfPoints, and polygon.

Referenced by findMinEnclosingTriangle().

7.73.3.16 void MinEnclosingTriangleFinder::initialiseConvexPolygon (const vector< Point2f > & *points*) [private]

Initialise polygon as the convex hull of the given set of points.

Parameters

<i>points</i>	Set of points
---------------	---------------

Definition at line 55 of file MinEnclosingTriangleFinder.cpp.

References CONVEX_HULL_CLOCKWISE, and polygon.

Referenced by initialise().

7.73.3.17 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 252 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().

7.73.3.18 bool MinEnclosingTriangleFinder::intersectsAbove (const Point2f & *gammaPoint*, unsigned int *polygonPointIndex*) [private]

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 246 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::angleOfLineWrtOxAxis(), intersects(), INTERSECTS_ABOVE, and polygon.

Referenced by isNotBTangency().

7.73.3.19 `unsigned int MinEnclosingTriangleFinder::intersectsAboveOrBelow(unsigned int successorOrPredecessorIndex, unsigned int pointIndex) [private]`

If ($\gamma(x)$ x) intersects P between successorOrPredecessorIndex and pointIndex is it above/below?

Parameters

<i>successor- Or- Predecessor- Index</i>	Index of the successor or predecessor
<i>pointIndex</i>	Index of the point x in the polygon

Definition at line 276 of file MinEnclosingTriangleFinder.cpp.

References height(), INTERSECTS_ABOVE, and INTERSECTS_BELOW.

Referenced by intersects().

7.73.3.20 `bool MinEnclosingTriangleFinder::intersectsBelow(const Point2f & gammaPoint, unsigned int polygonPointIndex) [private]`

Check if the line determined by gammaPoint and polygon[polygonPointIndex] intersects the polygon below the point polygon[polygonPointIndex].

Parameters

<i>gammaPoint</i>	Gamma(p)
<i>polygon- PointIndex</i>	Index of the polygon point which is considered when determining the line

Definition at line 240 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::angleOfLineWrtOxAxis(), intersects(), INTERSECTS_BELOW, and polygon.

Referenced by moveAIfLowAndBIfHigh(), and searchForBTangency().

7.73.3.21 bool MinEnclosingTriangleFinder::isFlushAngleBetweenPredecessor-AndSuccessor (double & *angleFlushEdge*, double *anglePredecessor*, double *angleSuccessor*) [private]

Check if the angle of the flush edge or its opposite angle lie between the angle of the predecessor and successor.

Parameters

<i>angleFlush-Edge</i>	Angle of the flush edge
<i>angle-Predecessor</i>	Angle of the predecessor
<i>angle-Successor</i>	Angle of the successor

Definition at line 284 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::isAngleBetweenNonReflex(), multiscale::Geometry2D::isOppositeAngleBetweenNonReflex(), and multiscale::Geometry2D::oppositeAngle().

Referenced by intersects().

7.73.3.22 bool MinEnclosingTriangleFinder::isGammaAngleBetween (double & *gammaAngle*, double *angle1*, double *angle2*) [private]

Check if the angle of the line (*gamma(p)* p) or its opposite angle lie between *angle1* and *angle2*.

Parameters

<i>gamma-Angle</i>	Angle of the line (<i>gamma(p)</i> p)
<i>angle1</i>	One of the boundary angles
<i>angle2</i>	Another boundary angle

Definition at line 297 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::isAngleBetweenNonReflex().

Referenced by intersects().

7.73.3.23 bool MinEnclosingTriangleFinder::isGammaAngleEqualTo (double & *gammaAngle*, double *angle*) [private]

Check if the angle of the line (*gamma(p)* p) or its opposite angle is equal to the given angle.

Parameters

<i>gamma-Angle</i>	Angle of the line ($\text{gamma}(p)$) p
<i>angle</i>	Angle to compare against

Definition at line 301 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Numeric::almostEqual().

Referenced by intersects().

7.73.3.24 bool MinEnclosingTriangleFinder::isLocalMinimalTriangle() [private]

Update the triangle vertices after all sides were set and check if a local minimal triangle was found or not.

See paper for more details

Definition at line 185 of file MinEnclosingTriangleFinder.cpp.

References isValidMinimalTriangle(), multiscale::Geometry2D::lineIntersection(), sideAEndVertex, sideAStartVertex, sideBEndVertex, sideBStartVertex, sideCEndVertex, sideCStartVertex, vertexA, vertexB, and vertexC.

Referenced by findMinEnclosingTriangle().

7.73.3.25 bool MinEnclosingTriangleFinder::isNotBTangency() [private]

Check if tangency for side B was not obtained.

See paper for more details

Definition at line 139 of file MinEnclosingTriangleFinder.cpp.

References a, b, gamma(), height(), intersectsAbove(), and predecessor().

Referenced by findMinEnclosingTriangle().

7.73.3.26 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 195 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()`.

7.73.3.27 `vector< double > MinEnclosingTriangleFinder::lineEquationParameters (const Point2f & p, const 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

<code>p</code>	One point for defining the equation of the line
<code>q</code>	Second point for defining the equation of the line

Definition at line 402 of file `MinEnclosingTriangleFinder.cpp`.

References `a`, `b`, `c`, and `multiscale::Geometry2D::lineEquationDeterminedByPoints()`.

Referenced by `findGammaIntersectionPoints()`.

7.73.3.28 `bool MinEnclosingTriangleFinder::middlePointOfSideB (Point2f & middlePointOfSideB) [private]`

Return the middle point of side B.

Definition at line 227 of file `MinEnclosingTriangleFinder.cpp`.

References `multiscale::Geometry2D::lineIntersection()`, `multiscale::Geometry2D::middlePoint()`, `sideAEndVertex`, `sideAStartVertex`, `sideBEndVertex`, `sideBStartVertex`, `sideCEndVertex`, `sideCStartVertex`, `vertexA`, and `vertexC`.

Referenced by `updateSidesBA()`.

7.73.3.29 `void MinEnclosingTriangleFinder::moveAIfLowAndBIfHigh () [private]`

Move "a" if it is low and "b" if it is high.

See paper for more details

Definition at line 118 of file `MinEnclosingTriangleFinder.cpp`.

References `a`, `advance()`, `b`, `gamma()`, `height()`, and `intersectsBelow()`.

Referenced by `findMinEnclosingTriangle()`.

7.73.3.30 `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 423 of file MinEnclosingTriangleFinder.cpp.

References nrOfPoints.

Referenced by findVertexCOnSideB(), gamma(), height(), intersects(), isNotBTangency(), isValidMinimalTriangle(), searchForBTangency(), updateSidesBA(), and updateSidesCA().

7.73.3.31 double MinEnclosingTriangleFinder::returnMinEnclosingTriangle (const vector< Point2f > & polygon, vector< 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>min- Enclosing- Triangle</i>	Minimum area triangle enclosing the given polygon

Definition at line 72 of file MinEnclosingTriangleFinder.cpp.

References multiscale::Geometry2D::areaOfTriangle().

Referenced by findMinTriangle().

7.73.3.32 void MinEnclosingTriangleFinder::searchForBTangency () [private]

Search for the tangency of side B.

See paper for more details

Definition at line 130 of file MinEnclosingTriangleFinder.cpp.

References a, advance(), b, gamma(), multiscale::Numeric::greaterOrEqual(), height(), intersectsBelow(), and predecessor().

Referenced by findMinEnclosingTriangle().

7.73.3.33 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 419 of file MinEnclosingTriangleFinder.cpp.

References nrOfPoints.

Referenced by advance(), advanceBToRightChain(), findVertexCOnSideB(), gamma(), and intersects().

7.73.3.34 void MinEnclosingTriangleFinder::updateMinEnclosingTriangle (
vector< Point2f > & *minEnclosingTriangle*, double & *minEnclosingTriangleArea*)
[private]

Update the current minimum area enclosing triangle if the newly obtained one has a smaller area.

Parameters

<i>min- Enclosing- Triangle</i>	Minimum area triangle enclosing the given polygon
<i>min- Enclosing- TriangleArea</i>	Area of the minimum area triangle enclosing the given polygon

Definition at line 213 of file MinEnclosingTriangleFinder.cpp.

References area, multiscale::Geometry2D::areaOfTriangle(), vertexA, vertexB, and vertexC.

Referenced by findMinEnclosingTriangle().

7.73.3.35 void MinEnclosingTriangleFinder::updateSideB () [private]

Set side B if tangency for side B was obtained.

See paper for more details

Definition at line 175 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().

7.73.3.36 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 157 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().

7.73.3.37 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 149 of file MinEnclosingTriangleFinder.cpp.

References a, c, polygon, predecessor(), sideAEndVertex, sideAStartVertex, sideCEndVertex, and sideCStartVertex.

Referenced by findMinEnclosingTriangle().

7.73.4 Member Data Documentation

7.73.4.1 unsigned int multiscale::MinEnclosingTriangleFinder::a [private]

Index of point "a"; see paper for more details

Definition at line 45 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), gamma(), initialiseAlgorithmVariables(), isNotBTangency(), isValidMinimalTriangle(), lineEquationParameters(), MinEnclosingTriangleFinder(), moveAlfLowAndBlfHigh(), searchForBTangency(), updateSidesBA(), and updateSidesCA().

7.73.4.2 double multiscale::MinEnclosingTriangleFinder::area [private]

Area of the current considered enclosing triangle

Definition at line 43 of file MinEnclosingTriangleFinder.hpp.

Referenced by MinEnclosingTriangleFinder(), and updateMinEnclosingTriangle().

7.73.4.3 unsigned int multiscale::MinEnclosingTriangleFinder::b [private]

Index of point "b"; see paper for more details

Definition at line 46 of file MinEnclosingTriangleFinder.hpp.

Referenced by advanceBToRightChain(), initialiseAlgorithmVariables(), isNotBTangency(), isValidMinimalTriangle(), lineEquationParameters(), MinEnclosingTriangleFinder(), moveAlfLowAndBlfHigh(), searchForBTangency(), updateSideB(), and updateSidesBA().

7.73.4.4 **unsigned int multiscale::MinEnclosingTriangleFinder::c** [private]

Index of point "c"; see paper for more details

Definition at line 47 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinEnclosingTriangle(), findVertexCOnSideB(), gamma(), height(), initialiseAlgorithmVariables(), intersects(), lineEquationParameters(), MinEnclosingTriangleFinder(), and updateSidesCA().

7.73.4.5 **const bool MinEnclosingTriangleFinder::CONVEX_HULL_CLOCKWISE = true** [static, private]

Definition at line 360 of file MinEnclosingTriangleFinder.hpp.

Referenced by initialiseConvexPolygon().

7.73.4.6 **const string MinEnclosingTriangleFinder::ERR_MIDPOINT_SIDE_B = "The position of the middle point of side B could not be determined."** [static, private]

Definition at line 368 of file MinEnclosingTriangleFinder.hpp.

7.73.4.7 **const string MinEnclosingTriangleFinder::ERR_NR_POINTS = "The number of 2D points in the input vector should be greater than 0."** [static, private]

Definition at line 367 of file MinEnclosingTriangleFinder.hpp.

Referenced by find().

7.73.4.8 **const 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 369 of file MinEnclosingTriangleFinder.hpp.

Referenced by updateSideB().

7.73.4.9 **const 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 371 of file MinEnclosingTriangleFinder.hpp.

7.73.4.10 **const 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 370 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB().

7.73.4.11 **const unsigned int MinEnclosingTriangleFinder::INTERSECTS_ABOVE = 2**
[static, private]

Definition at line 363 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersectsAbove(), and intersectsAboveOrBelow().

7.73.4.12 **const unsigned int MinEnclosingTriangleFinder::INTERSECTS_BELOW = 1**
[static, private]

Definition at line 362 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersects(), intersectsAboveOrBelow(), and intersectsBelow().

7.73.4.13 **const unsigned int MinEnclosingTriangleFinder::INTERSECTS_CRITICAL =**
3 [static, private]

Definition at line 364 of file MinEnclosingTriangleFinder.hpp.

Referenced by intersects().

7.73.4.14 **const unsigned int MinEnclosingTriangleFinder::INTERSECTS_LIMIT = 4**
[static, private]

Definition at line 365 of file MinEnclosingTriangleFinder.hpp.

7.73.4.15 **unsigned int multiscale::MinEnclosingTriangleFinder::nrOfPoints**
[private]

Number of points defining the polygon

Definition at line 49 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinEnclosingTriangle(), initialiseAlgorithmVariables(), MinEnclosingTriangleFinder(), predecessor(), and successor().

7.73.4.16 **vector<Point2f> multiscale::MinEnclosingTriangleFinder::polygon**
[private]

Polygon for which the minimum area enclosing triangle is computed

Definition at line 51 of file MinEnclosingTriangleFinder.hpp.

Referenced by findMinTriangle(), findVertexCOnSideB(), gamma(), height(), initialiseAlgorithmVariables(), initialiseConvexPolygon(), intersects(), intersectsAbove(), intersectsBelow(), isValidMinimalTriangle(), updateSideB(), updateSidesBA(), and updateSidesCA().

7.73.4.17 Point2f multiscale::MinEnclosingTriangleFinder::sideAEndVertex
[private]

Ending vertex for side A of triangle

Definition at line 35 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSidesBA(), and updateSidesCA().

7.73.4.18 Point2f multiscale::MinEnclosingTriangleFinder::sideAStartVertex
[private]

Starting vertex for side A of triangle

Definition at line 34 of file MinEnclosingTriangleFinder.hpp.

Referenced by isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSidesBA(), and updateSidesCA().

7.73.4.19 Point2f multiscale::MinEnclosingTriangleFinder::sideBEndVertex
[private]

Ending vertex for side B of triangle

Definition at line 38 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSideB(), and updateSidesBA().

7.73.4.20 Point2f multiscale::MinEnclosingTriangleFinder::sideBStartVertex
[private]

Starting vertex for side B of triangle

Definition at line 37 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), updateSideB(), and updateSidesBA().

7.73.4.21 Point2f multiscale::MinEnclosingTriangleFinder::sideCEndVertex
[private]

Ending vertex for side C of triangle

Definition at line 41 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), and updateSidesCA().

7.73.4.22 Point2f multiscale::MinEnclosingTriangleFinder::sideCStartVertex [private]

Starting vertex for side C of triangle

Definition at line 40 of file MinEnclosingTriangleFinder.hpp.

Referenced by findVertexCOnSideB(), isLocalMinimalTriangle(), isValidMinimalTriangle(), middlePointOfSideB(), and updateSidesCA().

7.73.4.23 const unsigned int MinEnclosingTriangleFinder::VALIDATION_SIDE_A_TANGENT = 0 [static, private]

Definition at line 373 of file MinEnclosingTriangleFinder.hpp.

Referenced by isValidMinimalTriangle(), and updateSidesBA().

7.73.4.24 const unsigned int MinEnclosingTriangleFinder::VALIDATION_SIDE_B_TANGENT = 1 [static, private]

Definition at line 374 of file MinEnclosingTriangleFinder.hpp.

Referenced by isValidMinimalTriangle(), and updateSideB().

7.73.4.25 const unsigned int MinEnclosingTriangleFinder::VALIDATION_SIDES_FLUSH = 2 [static, private]

Definition at line 375 of file MinEnclosingTriangleFinder.hpp.

Referenced by updateSidesBA().

7.73.4.26 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 24 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `isValidMinimalTriangle()`, `MinEnclosingTriangleFinder()`, `updateSideB()`, and `updateSidesBA()`.

7.73.4.27 `Point2f multiscale::MinEnclosingTriangleFinder::vertexA` [private]

Vertex A of the current considered enclosing triangle

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

Referenced by `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, and `updateMinEnclosingTriangle()`.

7.73.4.28 `Point2f multiscale::MinEnclosingTriangleFinder::vertexB` [private]

Vertex B of the current considered enclosing triangle

Definition at line 31 of file `MinEnclosingTriangleFinder.hpp`.

Referenced by `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, and `updateMinEnclosingTriangle()`.

7.73.4.29 `Point2f multiscale::MinEnclosingTriangleFinder::vertexC` [private]

Vertex C of the current considered enclosing triangle

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

Referenced by `isLocalMinimalTriangle()`, `isValidMinimalTriangle()`, `middlePointOfSideB()`, and `updateMinEnclosingTriangle()`.

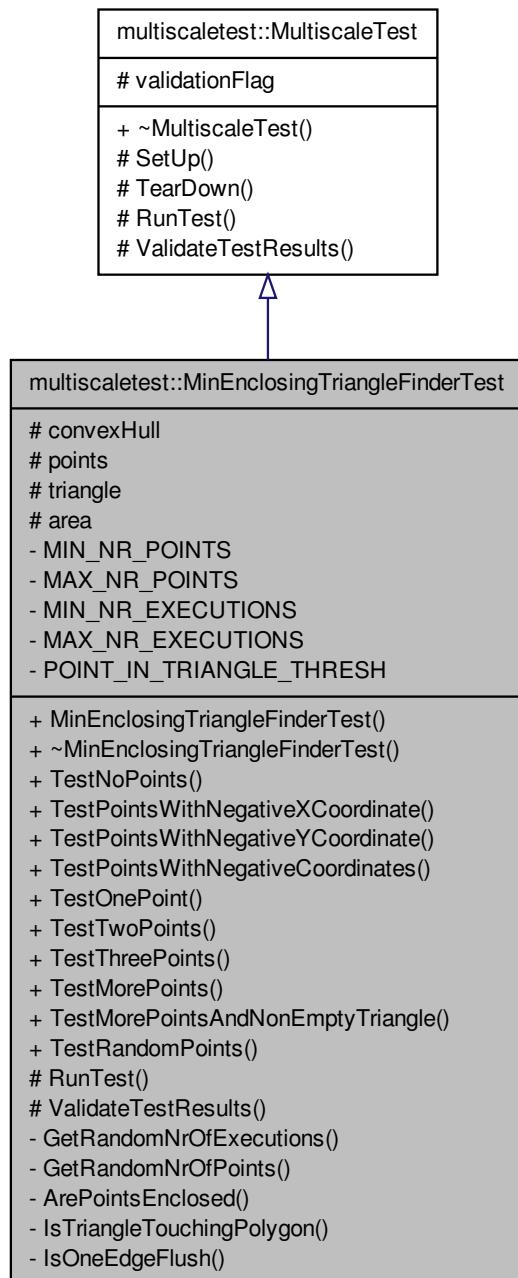
The documentation for this class was generated from the following files:

- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util-/MinEnclosingTriangleFinder.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/MinEnclosingTriangleFinder.cpp](#)

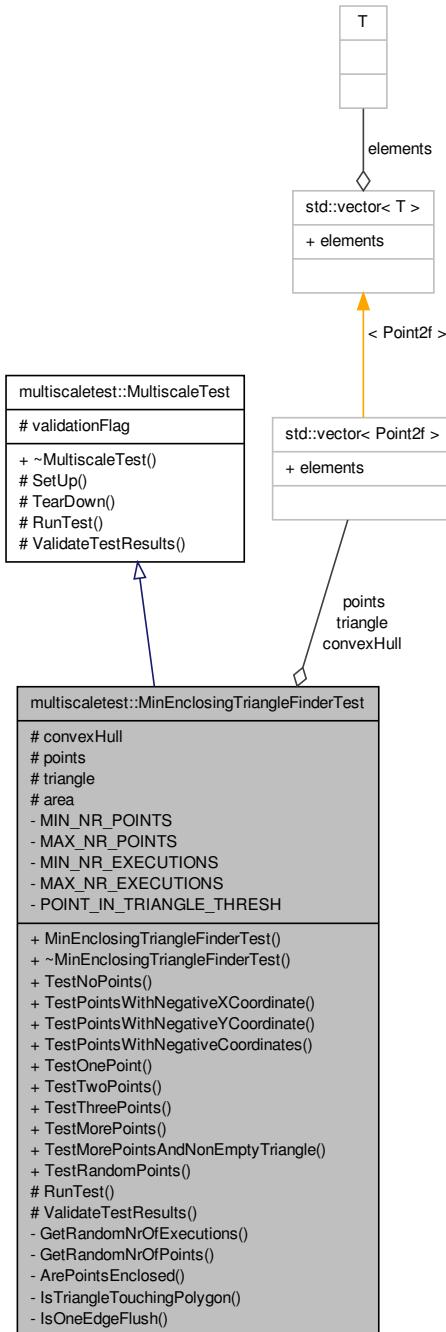
7.74 `multiscaletest::MinEnclosingTriangleFinderTest` Class - Reference

Class for testing the minimum enclosing triangle algorithm.

Inheritance diagram for multiscaletest::MinEnclosingTriangleFinderTest:



Collaboration diagram for multiscaletest::MinEnclosingTriangleFinderTest:



Public Member Functions

- [MinEnclosingTriangleFinderTest \(\)](#)
Test the scenario when an empty vector of points is provided.
- [~MinEnclosingTriangleFinderTest \(\)](#)
- [bool TestNoPoints \(\)](#)
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 vector is not empty.
- [bool TestRandomPoints \(\)](#)
Test the scenario when randomly initialised 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

- [vector< Point2f > convexHull](#)
- [vector< Point2f > points](#)
- [vector< 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

7.74.1 Detailed Description

Class for testing the minimum enclosing triangle algorithm.

Definition at line 16 of file MinEnclosingTriangleFinderTest.cpp.

7.74.2 Constructor & Destructor Documentation

7.74.2.1 **multiscaletest::MinEnclosingTriangleFinderTest::MinEnclosingTriangleFinderTest()**

Definition at line 98 of file MinEnclosingTriangleFinderTest.cpp.

7.74.2.2 **multiscaletest::MinEnclosingTriangleFinderTest::~MinEnclosingTriangleFinderTest()**

Definition at line 106 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3 Member Function Documentation

7.74.3.1 **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.

7.74.3.2 int multiscaletest::MinEnclosingTriangleFinderTest::GetRandomNrOfExecutions() [private]

Get a random number of executions.

Definition at line 234 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.3 int multiscaletest::MinEnclosingTriangleFinderTest::GetRandomNrOfPoints() [private]

Get a random number of points.

Definition at line 239 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.4 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.

7.74.3.5 bool multiscaletest::MinEnclosingTriangleFinderTest::IsTriangleTouchingPolygon() [private]

Check if the triangle's middle points are touching the polygon.

Definition at line 258 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.6 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().

7.74.3.7 bool multiscaletest::MinEnclosingTriangleFinderTest::TestMorePoints()

Test the scenario when more than three input points are provided.

Definition at line 175 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.8 bool multiscaletest::MinEnclosingTriangleFinderTest::TestMorePoints-AndNonEmptyTriangle()

Test the scenario when the output vector is not empty.

Definition at line 184 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.9 bool multiscaletest::MinEnclosingTriangleFinderTest::TestNoPoints()

Test the scenario when an empty vector of points is provided.

Definition at line 114 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.10 bool multiscaletest::MinEnclosingTriangleFinderTest::TestOnePoint()

Test the scenario when only one input point is provided.

Definition at line 148 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.11 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWith-NegativeCoordinates()

Test the scenario when there exists at least one point with negative coordinates.

Definition at line 139 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.12 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWith-NegativeXCoordinate()

Test the scenario when there exists at least one point with negative x coordinate.

Definition at line 121 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.13 bool multiscaletest::MinEnclosingTriangleFinderTest::TestPointsWith-NegativeYCoordinate()

Test the scenario when there exists at least one point with negative y coordinate.

Definition at line 130 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.14 bool multiscaletest::MinEnclosingTriangleFinderTest::TestRandom-Points()

Test the scenario when randomly initialised vectors of input points are provided.

Definition at line 195 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.15 `bool multiscaletest::MinEnclosingTriangleFinderTest::TestThreePoints()`

Test the scenario when only three input points are provided.

Definition at line 166 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.16 `bool multiscaletest::MinEnclosingTriangleFinderTest::TestTwoPoints()`

Test the scenario when only two input points are provided.

Definition at line 157 of file MinEnclosingTriangleFinderTest.cpp.

7.74.3.17 `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.

7.74.4 Member Data Documentation

7.74.4.1 `double multiscaletest::MinEnclosingTriangleFinderTest::area [protected]`

Area of the minimum enclosing triangle

Definition at line 24 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.2 `vector<Point2f> multiscaletest::MinEnclosingTriangleFinderTest::convexHull [protected]`

Convex hull of the 2D point set

Definition at line 20 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.3 `const int multiscaletest::MinEnclosingTriangleFinderTest::MAX_NR_EXECUTIONS = 10000 [static, private]`

Definition at line 92 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.4 const int multiscaletest::MinEnclosingTriangleFinderTest::MAX_NR_POINTS = 10000 [static, private]

Definition at line 90 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.5 const int multiscaletest::MinEnclosingTriangleFinderTest::MIN_NR_EXECUTIONS = 5000 [static, private]

Definition at line 91 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.6 const int multiscaletest::MinEnclosingTriangleFinderTest::MIN_NR_POINTS = 1 [static, private]

Definition at line 89 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.7 const double multiscaletest::MinEnclosingTriangleFinderTest::POINT_IN_TRIANGLE_THRESH = 1E-4 [static, private]

Definition at line 94 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.8 vector<Point2f> multiscaletest::MinEnclosingTriangleFinderTest::points [protected]

Collection of 2D points

Definition at line 22 of file MinEnclosingTriangleFinderTest.cpp.

7.74.4.9 vector<Point2f> multiscaletest::MinEnclosingTriangleFinderTest::triangle [protected]

Minimum enclosing triangle

Definition at line 23 of file MinEnclosingTriangleFinderTest.cpp.

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

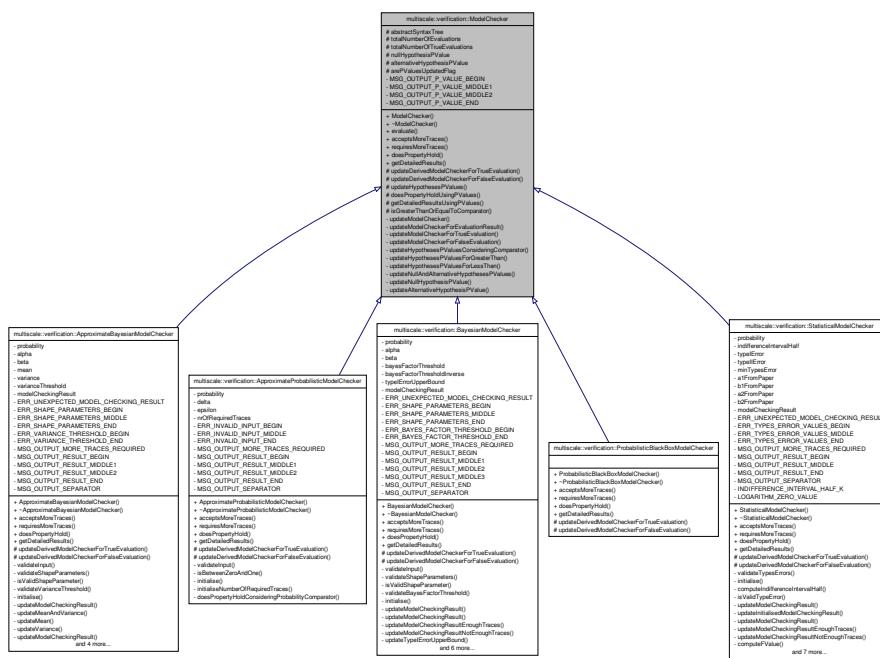
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/[MinEnclosingTriangleFinderTest.cpp](#)

7.75 multiscale::verification::ModelChecker Class Reference

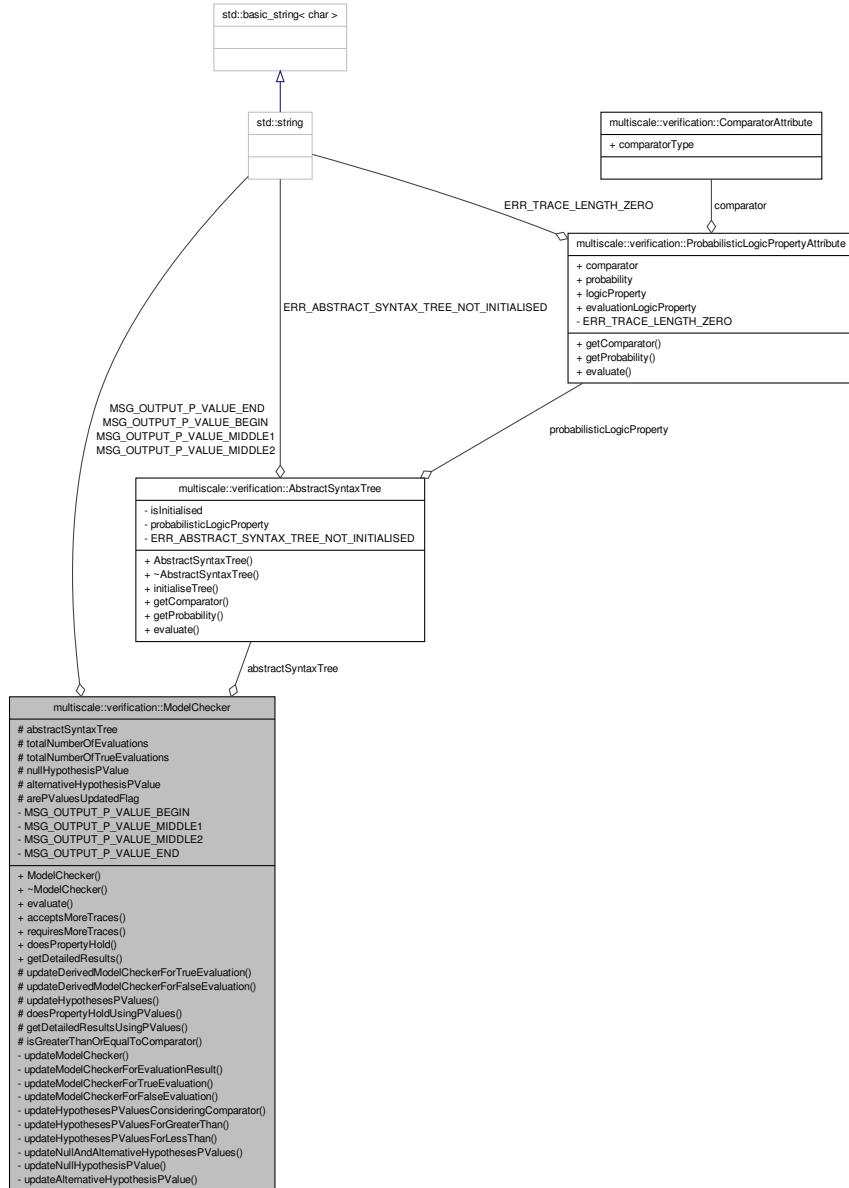
Abstract class representing a generic model checker.

```
#include <ModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::ModelChecker:



Collaboration diagram for multiscale::verification::ModelChecker:



Public Member Functions

- `ModelChecker (const AbstractSyntaxTree &abstractSyntaxTree)`
- `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](#)

- 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 [updateHypothesesPValuesForGreaterThan \(\)](#)
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](#) = ")"

7.75.1 Detailed Description

Abstract class representing a generic model checker.

Definition at line 13 of file ModelChecker.hpp.

7.75.2 Constructor & Destructor Documentation

7.75.2.1 [multiscale::verification::ModelChecker::ModelChecker \(const AbstractSyntaxTree & abstractSyntaxTree \) \[inline\]](#)

Definition at line 32 of file ModelChecker.hpp.

7.75.2.2 **virtual multiscale::verification::ModelChecker::~ModelChecker()**
[inline, virtual]

Definition at line 39 of file ModelChecker.hpp.

7.75.3 Member Function Documentation

7.75.3.1 **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](#).

7.75.3.2 **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](#).

7.75.3.3 **bool ModelChecker::doesPropertyHoldUsingPValues()**
[protected]

Check if the property holds considering the given p-values.

Definition at line 23 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\(\)](#).

7.75.3.4 **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 7 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::evaluate(), and updateModelChecker().

7.75.3.5 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](#).

7.75.3.6 std::string ModelChecker::getDetailedResultsUsingPValues() [protected]

Get the detailed results when deciding if the property holds based on p-values.

Definition at line 29 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\(\)](#).

7.75.3.7 bool ModelChecker::isGreaterThanOrEqualToComparator() [protected]

Check if the comparator used by the probabilistic logic property is greater than or equal to.

Definition at line 41 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getComparator(), multiscale::verification::GreaterThan, and multiscale::verification::GreaterThanOrEqual.

Referenced by [multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator\(\)](#), [multiscale::verification::Bayesian-](#)

`ModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::StatisticalModelChecker::doesPropertyHoldConsideringProbabilityComparator()`, `multiscale::verification::ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator()`, and `updateHypothesesPValuesConsideringComparator()`.

7.75.3.8 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](#).

7.75.3.9 void ModelChecker::updateAlternativeHypothesisPValue (unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability) [private]

Update the alternative 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 114 of file `ModelChecker.cpp`.

References `alternativeHypothesisPValue`, and `multiscale::BinomialDistribution::cdf()`.

Referenced by `updateNullAndAlternativeHypothesesPValues()`.

7.75.3.10 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](#).

7.75.3.11 **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](#).

7.75.3.12 **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 15 of file ModelChecker.cpp.

References `arePValuesUpdatedFlag`, and `updateHypothesesPValuesConsideringComparator()`.

Referenced by `doesPropertyHoldUsingPValues()`, and `getDetailedResultsUsingPValues()`.

7.75.3.13 **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 73 of file ModelChecker.cpp.

References `isGreaterThanOrEqualToComparator()`, `updateHypothesesPValuesForGreaterThanOrEqual()`, and `updateHypothesesPValuesForLessThan()`.

Referenced by `updateHypothesesPValues()`.

7.75.3.14 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 θ = probability specified in the logic property

Definition at line 81 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), totalNumberOfEvaluations, totalNumberOfTrueEvaluations, and updateNullAndAlternativeHypothesesPValues().

Referenced by updateHypothesesPValuesConsideringComparator().

7.75.3.15 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 θ = probability specified in the logic property

Definition at line 89 of file ModelChecker.cpp.

References abstractSyntaxTree, multiscale::verification::AbstractSyntaxTree::getProbability(), totalNumberOfEvaluations, totalNumberOfTrueEvaluations, and updateNullAndAlternativeHypothesesPValues().

Referenced by updateHypothesesPValuesConsideringComparator().

7.75.3.16 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>evaluation-Result</i>	The result of evaluating the logic property considering the last trace
--------------------------	--

Definition at line 50 of file ModelChecker.cpp.

References arePValuesUpdatedFlag, and updateModelCheckerForEvaluationResult().

Referenced by evaluate().

7.75.3.17 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>evaluation-Result</i>	The result of evaluating the logic property considering the last trace
--------------------------	--

Definition at line 56 of file ModelChecker.cpp.

References updateModelCheckerForFalseEvaluation(), and updateModelCheckerForTrueEvaluation().

Referenced by updateModelChecker().

7.75.3.18 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 69 of file ModelChecker.cpp.

References totalNumberOfEvaluations.

Referenced by updateModelCheckerForEvaluationResult().

7.75.3.19 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 64 of file ModelChecker.cpp.

References totalNumberOfEvaluations, and totalNumberOfTrueEvaluations.

Referenced by updateModelCheckerForEvaluationResult().

7.75.3.20 void ModelChecker::updateNullAndAlternativeHypothesesPValues (unsigned int nrOfEvaluations, unsigned int nrOfSuccesses, double probability) [private]

Update the null and alternative hypotheses p-values.

Parameters

<i>nrOf-Evaluations</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 97 of file ModelChecker.cpp.

References updateAlternativeHypothesisPValue(), and updateNullHypothesisPValue().

Referenced by updateHypothesesPValuesForGreaterThan(), and updateHypothesesPValuesForLessThan().

7.75.3.21 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 104 of file ModelChecker.cpp.

References multiscale::BinomialDistribution::cdf(), and nullHypothesisPValue().

Referenced by updateNullAndAlternativeHypothesesPValues().

7.75.4 Member Data Documentation

7.75.4.1 AbstractSyntaxTree multiscale::verification::ModelChecker::abstractSyntaxTree [protected]

The abstract syntax tree representing the logic property which this model checker instance evaluates

Definition at line 17 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().

7.75.4.2 **double multiscale::verification::ModelChecker::alternativeHypothesisP-Value** [protected]

The p-value for the alternative hypothesis to hold

Definition at line 26 of file ModelChecker.hpp.

Referenced by doesPropertyHoldUsingPValues(), getDetailedResultsUsingPValues(), and updateAlternativeHypothesisPValue().

7.75.4.3 **bool multiscale::verification::ModelChecker::arePValuesUpdatedFlag** [protected]

Flag indicating if the p-values were updated

Definition at line 28 of file ModelChecker.hpp.

Referenced by updateHypothesesPValues(), and updateModelChecker().

7.75.4.4 **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 163 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

7.75.4.5 **const std::string ModelChecker::MSG_OUTPUT_P_VALUE_END = ")"** [static, private]

Definition at line 166 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

7.75.4.6 **const std::string ModelChecker::MSG_OUTPUT_P_VALUE_MIDDLE1 = "(p-value H0: "** [static, private]

Definition at line 164 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

7.75.4.7 **const std::string ModelChecker::MSG_OUTPUT_P_VALUE_MIDDLE2 = ", p-value H1: "** [static, private]

Definition at line 165 of file ModelChecker.hpp.

Referenced by getDetailedResultsUsingPValues().

**7.75.4.8 double multiscale::verification::ModelChecker::nullHypothesisPValue
[protected]**

The p-value for the null hypothesis to hold

Definition at line 25 of file ModelChecker.hpp.

Referenced by doesPropertyHoldUsingPValues(), getDetailedResultsUsingPValues(), and updateNullHypothesisPValue().

**7.75.4.9 unsigned int multiscale::verification::ModelChecker::totalNumberOf-
Evaluations [protected]**

The total number of evaluations

Definition at line 21 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().

**7.75.4.10 unsigned int multiscale::verification::ModelChecker::totalNumberOfTrue-
Evaluations [protected]**

The total number of times the abstract syntax tree was evaluated to true

Definition at line 22 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().

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.h](#)

[.hpp](#)

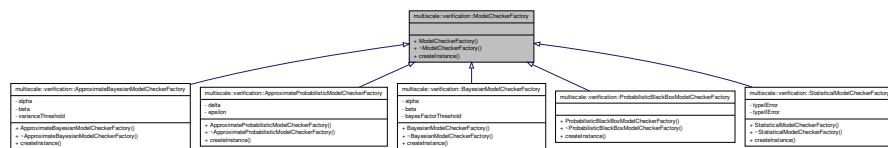
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ModelChecker.cpp](#)

7.76 multiscale::verification::ModelCheckerFactory Class Reference

Interface for different model checker factories.

```
#include <ModelCheckerFactory.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckerFactory:



Public Member Functions

- [ModelCheckerFactory \(\)](#)
- virtual [~ModelCheckerFactory \(\)](#)
- virtual std::shared_ptr<[ModelChecker](#)> [createInstance](#) (const [AbstractSyntaxTree](#) &[abstractSyntaxTree](#))=0

Create an instance of the model checker.

7.76.1 Detailed Description

Interface for different model checker factories.

Definition at line 15 of file [ModelCheckerFactory.hpp](#).

7.76.2 Constructor & Destructor Documentation

7.76.2.1 multiscale::verification::ModelCheckerFactory::ModelCheckerFactory () [inline]

Definition at line 19 of file [ModelCheckerFactory.hpp](#).

7.76.2.2 virtual multiscale::verification::ModelCheckerFactory::~ModelCheckerFactory() [inline, virtual]

Definition at line 20 of file [ModelCheckerFactory.hpp](#).

7.76.3 Member Function Documentation

7.76.3.1 `virtual std::shared_ptr<ModelChecker> multiscale::verification::- ModelCheckerFactory::createInstance (const AbstractSyntaxTree & abstractSyntaxTree) [pure virtual]`

Create an instance of the model checker.

Parameters

<code>abstract- SyntaxTree</code>	The abstract syntax tree representing the logic property to be checked
---------------------------------------	--

Implemented in `multiscale::verification::ApproximateBayesianModelCheckerFactory`, `multiscale::verification::BayesianModelCheckerFactory`, `multiscale::verification::- ApproximateProbabilisticModelCheckerFactory`, `multiscale::verification::Statistical- ModelCheckerFactory`, and `multiscale::verification::ProbabilisticBlackBoxModel- CheckerFactory`.

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

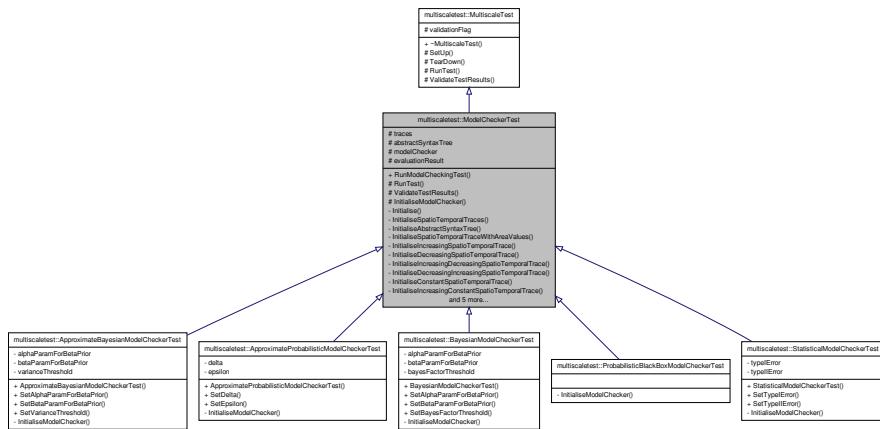
- `/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial- temporal/include/multiscale/verification/spatial-temporal/checking/ModelChecker- Factory.hpp`

7.77 multiscaletest::ModelCheckerTest Class Reference

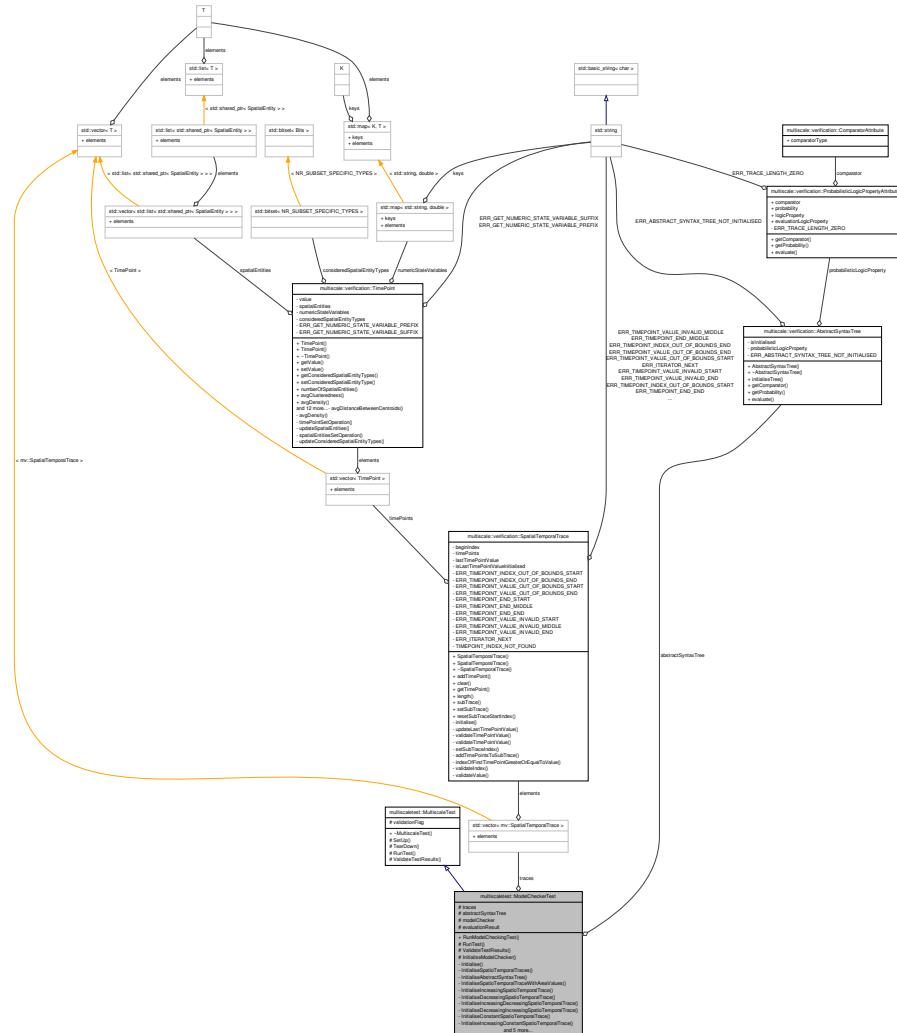
Class for testing model checkers.

```
#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 < [mv::SpatialTemporalTrace](#) > traces
- [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 [InitialiseAbstractSyntaxTree \(\)](#)
Initialise the abstract syntax tree.
- void [InitialiseSpatioTemporalTraceWithAreaValues \(const std::vector< double > areaValues\)](#)
Initialise the collection of spatio-temporal traces with the given spatial entity area values.
- void [InitialiseIncreasingSpatioTemporalTrace \(\)](#)
Initialise the globally increasing area spatio-temporal trace.
- void [InitialiseDecreasingSpatioTemporalTrace \(\)](#)
Initialise the globally decreasing area spatio-temporal trace.
- void [InitialiseIncreasingDecreasingSpatioTemporalTrace \(\)](#)
Initialise the increasing and then decreasing area spatio-temporal trace.
- void [InitialiseDecreasingIncreasingSpatioTemporalTrace \(\)](#)
Initialise the decreasing and then increasing area spatio-temporal trace.
- void [InitialiseConstantSpatioTemporalTrace \(\)](#)
Initialise the constant area spatio-temporal trace.
- void [InitialiseIncreasingConstantSpatioTemporalTrace \(\)](#)
Initialise the increasing and then constant area spatio-temporal trace.
- void [InitialiseDecreasingConstantSpatioTemporalTrace \(\)](#)
Initialise the decreasing and then constant area spatio-temporal trace.
- void [InitialiseConstantIncreasingSpatioTemporalTrace \(\)](#)
Initialise the constant and then increasing area spatio-temporal trace.
- void [InitialiseConstantDecreasingSpatioTemporalTrace \(\)](#)
Initialise the constant and then decreasing area spatio-temporal trace.
- void [InitialiseIncreasingConstantDecreasingSpatioTemporalTrace \(\)](#)
Initialise the increasing, constant and then decreasing area spatio-temporal trace.

- void [InitialiseDecreasingConstantIncreasingSpatioTemporalTrace \(\)](#)
Initialise the decreasing, constant and increasing area spatio-temporal trace.
- void [InitialiseIncreasingConstantIncreasingSpatioTemporalTrace \(\)](#)
Initialise the increasing, constant and then increasing area spatio-temporal trace.

7.77.1 Detailed Description

Class for testing model checkers.

Definition at line 21 of file ModelCheckerTest.hpp.

7.77.2 Member Function Documentation

7.77.2.1 void multiscaletest::ModelCheckerTest::Initialise() [private]

Initialisation function.

Definition at line 127 of file ModelCheckerTest.hpp.

References InitialiseAbstractSyntaxTree(), InitialiseModelChecker(), and InitialiseSpatioTemporalTraces().

Referenced by RunModelCheckingTest().

7.77.2.2 void multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree() [private]

Initialise the abstract syntax tree.

Definition at line 150 of file ModelCheckerTest.hpp.

References abstractSyntaxTree, INPUT_LOGIC_PROPERTY, and multiscale::verification::Parser::parse().

Referenced by Initialise().

7.77.2.3 void multiscaletest::ModelCheckerTest::InitialiseConstantDecreasingSpatioTemporalTrace() [private]

Initialise the constant and then decreasing area spatio-temporal trace.

7.77.2.4 void multiscaletest::ModelCheckerTest::InitialiseConstantIncreasingSpatioTemporalTrace() [private]

Initialise the constant and then increasing area spatio-temporal trace.

7.77.2.5 void multiscaletest::ModelCheckerTest::InitialiseConstantSpatio-TemporalTrace() [private]

Initialise the constant area spatio-temporal trace.

7.77.2.6 void multiscaletest::ModelCheckerTest::Initialise-DecreasingConstantIncreasingSpatioTemporalTrace() [private]

Initialise the decreasing, constant and increasing area spatio-temporal trace.

7.77.2.7 void multiscaletest::ModelCheckerTest::InitialiseDecreasingConstant-SpatioTemporalTrace() [private]

Initialise the decreasing and then constant area spatio-temporal trace.

7.77.2.8 void multiscaletest::ModelCheckerTest::Initialise-DecreasingIncreasingSpatioTemporalTrace() [private]

Initialise the decreasing and then increasing area spatio-temporal trace.

7.77.2.9 void multiscaletest::ModelCheckerTest::InitialiseDecreasingSpatio-TemporalTrace() [private]

Initialise the globally decreasing area spatio-temporal trace.

7.77.2.10 void multiscaletest::ModelCheckerTest::Initialise-IncreasingConstantDecreasingSpatioTemporalTrace() [private]

Initialise the increasing, constant and then decreasing area spatio-temporal trace.

7.77.2.11 void multiscaletest::ModelCheckerTest::Initialise-IncreasingConstantIncreasingSpatioTemporalTrace() [private]

Initialise the increasing, constant and then increasing area spatio-temporal trace.

7.77.2.12 void multiscaletest::ModelCheckerTest::Initialise-IncreasingConstantSpatioTemporalTrace() [private]

Initialise the increasing and then constant area spatio-temporal trace.

7.77.2.13 void multiscaletest::ModelCheckerTest::InitialiseIncreasingDecreasingSpatioTemporalTrace () [private]

Initialise the increasing and then decreasing area spatio-temporal trace.

7.77.2.14 void multiscaletest::ModelCheckerTest::InitialiseIncreasingSpatioTemporalTrace () [private]

Initialise the globally increasing area spatio-temporal trace.

7.77.2.15 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\(\)](#).

7.77.2.16 void multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraces () [private]

Initialise the collection of spatio-temporal traces.

Definition at line 133 of file [ModelCheckerTest.hpp](#).

References [InitialiseSpatioTemporalTraceWithAreaValues\(\)](#), and [traces](#).

Referenced by [Initialise\(\)](#).

7.77.2.17 void multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues (const std::vector< double > areaValues) [private]

Initialise the collection of spatio-temporal traces with the given spatial entity area values.

The assumption is that each timepoint contains only one spatial entity of the same time. Therefore each area value corresponds to a different timepoint and spatial entity.

Parameters

<i>areaValues</i>	The values of the areas
-------------------	-------------------------

Definition at line 156 of file [ModelCheckerTest.hpp](#).

References [multiscale::verification::TimePoint::addSpatialEntity\(\)](#), [multiscale::verification](#)

::SpatialTemporalTrace::addTimePoint(), multiscale::verification::TimePoint::getSpatialEntitiesBeginIterator(), multiscale::verification::Regions, multiscale::verification::TimePoint::removeSpatialEntity(), multiscale::verification::TimePoint::setConsideredSpatialEntityType(), and traces.

Referenced by InitialiseSpatioTemporalTraces().

7.77.2.18 bool multiscaletest::ModelCheckerTest::RunModelCheckingTest()

Run the test for the given logic property.

Definition at line 106 of file ModelCheckerTest.hpp.

References evaluationResult, Initialise(), RunTest(), and ValidateTestResults().

7.77.2.19 void multiscaletest::ModelCheckerTest::RunTest() [override, protected, virtual]

Run the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 115 of file ModelCheckerTest.hpp.

References evaluationResult, modelChecker, and traces.

Referenced by RunModelCheckingTest().

7.77.2.20 void multiscaletest::ModelCheckerTest::ValidateTestResults() [override, protected, virtual]

Validate the results of the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 125 of file ModelCheckerTest.hpp.

Referenced by RunModelCheckingTest().

7.77.3 Member Data Documentation

7.77.3.1 mv::AbstractSyntaxTree multiscaletest::ModelCheckerTest::abstractSyntaxTree [protected]

The abstract syntax tree corresponding to the logic property

Definition at line 27 of file ModelCheckerTest.hpp.

Referenced by InitialiseAbstractSyntaxTree().

7.77.3.2 bool multiscaletest::ModelCheckerTest::evaluationResult [protected]

The result of the model checking evaluation

Definition at line 30 of file ModelCheckerTest.hpp.

Referenced by RunModelCheckingTest(), and RunTest().

7.77.3.3 std::shared_ptr<mv::ModelChecker> multiscaletest::ModelCheckerTest::modelChecker [protected]

The specific type of model checker employed

Definition at line 28 of file ModelCheckerTest.hpp.

Referenced by RunTest().

7.77.3.4 std::vector<mv::SpatialTemporalTrace> multiscaletest::ModelCheckerTest::traces [protected]

The collection of spatio-temporal traces

Definition at line 25 of file ModelCheckerTest.hpp.

Referenced by InitialiseSpatioTemporalTraces(), InitialiseSpatioTemporalTraceWithAreaValues(), and RunTest().

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

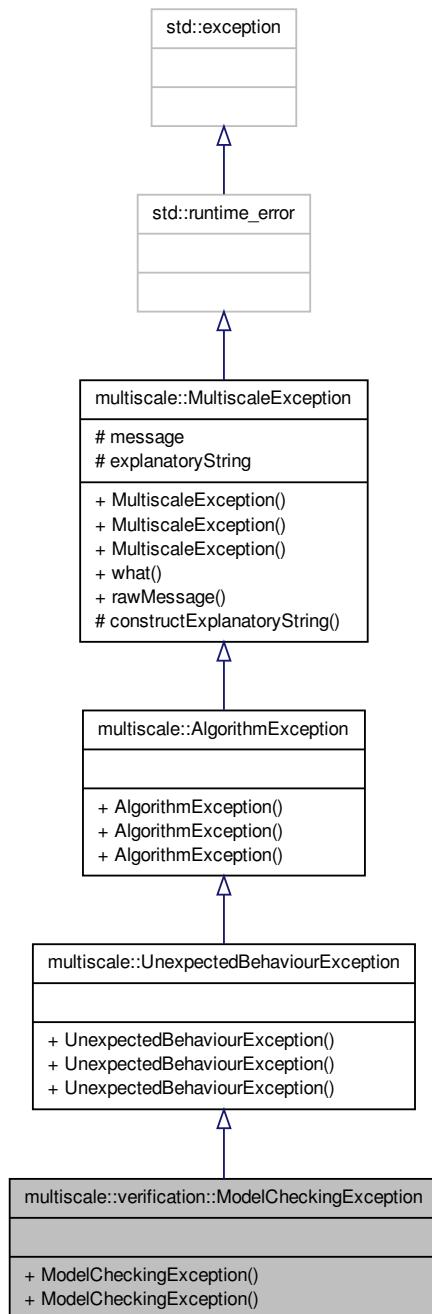
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/[ModelCheckerTest.hpp](#)

7.78 multiscale::verification::ModelCheckingException Class Reference

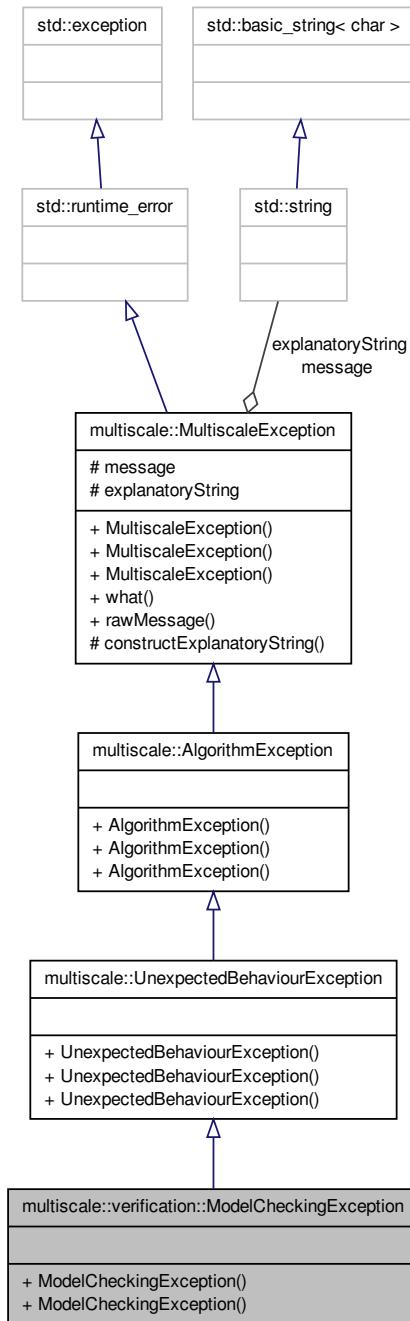
Class for representing a model checking exception.

```
#include <ModelCheckingException.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckingException:



Collaboration diagram for multiscale::verification::ModelCheckingException:



Public Member Functions

- [ModelCheckingException](#) (const string &file, int line, const string &msg)
- [ModelCheckingException](#) (const string &file, int line, const char *msg)

7.78.1 Detailed Description

Class for representing a model checking exception.

Definition at line 12 of file ModelCheckingException.hpp.

7.78.2 Constructor & Destructor Documentation

7.78.2.1 [multiscale::verification::ModelCheckingException::ModelCheckingException](#) (const string & *file*, int *line*, const string & *msg*) [inline]

Definition at line 16 of file ModelCheckingException.hpp.

References multiscale::MultiscaleException::explanatoryString.

7.78.2.2 [multiscale::verification::ModelCheckingException::ModelCheckingException](#) (const string & *file*, int *line*, const char * *msg*) [inline]

Definition at line 20 of file ModelCheckingException.hpp.

References multiscale::MultiscaleException::explanatoryString.

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

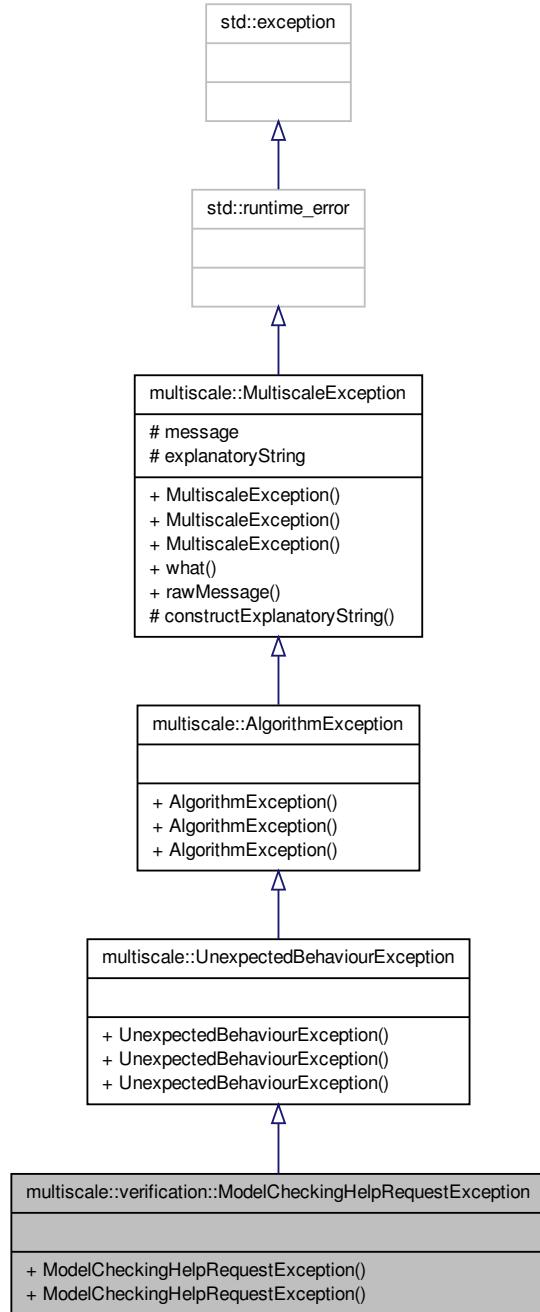
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ModelCheckingException.hpp](#)

7.79 multiscale::verification::ModelCheckingHelpRequestException Class Reference

Class for representing a model checking help request exception.

```
#include <ModelCheckingHelpRequestException.hpp>
```

Inheritance diagram for multiscale::verification::ModelCheckingHelpRequestException:

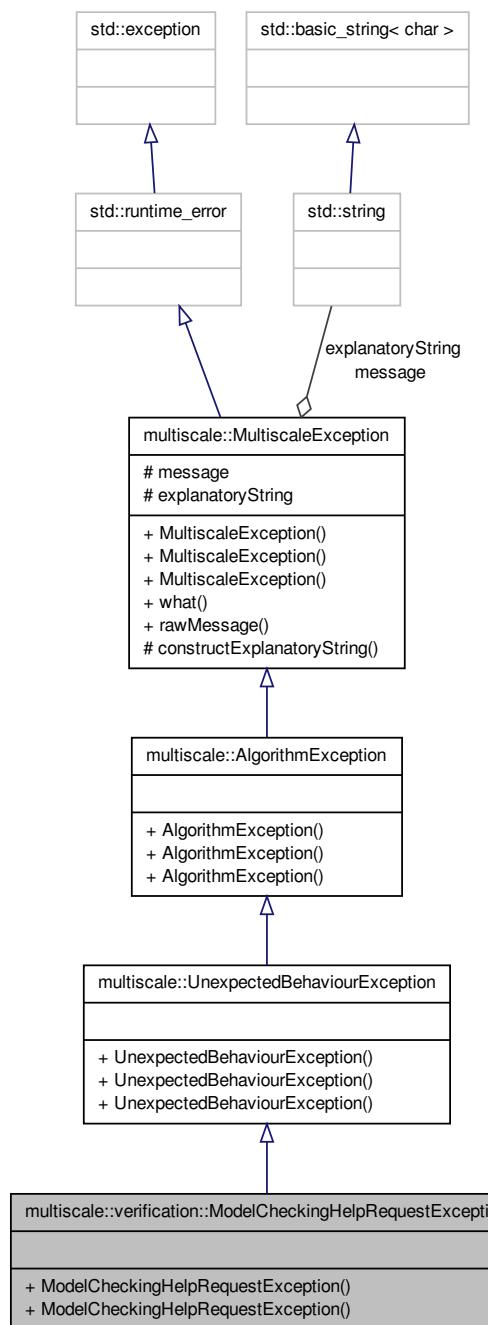


7.79 multiscale::verification::ModelCheckingHelpRequestException Class

Reference

495

Collaboration diagram for multiscale::verification::ModelCheckingHelpRequestException:



Public Member Functions

- [ModelCheckingHelpRequestException](#) (const string &file, int line, const string &msg)
- [ModelCheckingHelpRequestException](#) (const string &file, int line, const char *msg)

7.79.1 Detailed Description

Class for representing a model checking help request exception.

Definition at line 12 of file ModelCheckingHelpRequestException.hpp.

7.79.2 Constructor & Destructor Documentation

7.79.2.1 multiscale::verification::ModelCheckingHelpRequestException::ModelCheckingHelpRequestException (const string & file, int line, const string & msg) [inline]

Definition at line 16 of file ModelCheckingHelpRequestException.hpp.

References multiscale::MultiscaleException::explanatoryString.

7.79.2.2 multiscale::verification::ModelCheckingHelpRequestException::ModelCheckingHelpRequestException (const string & file, int line, const char * msg) [inline]

Definition at line 20 of file ModelCheckingHelpRequestException.hpp.

References multiscale::MultiscaleException::explanatoryString.

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

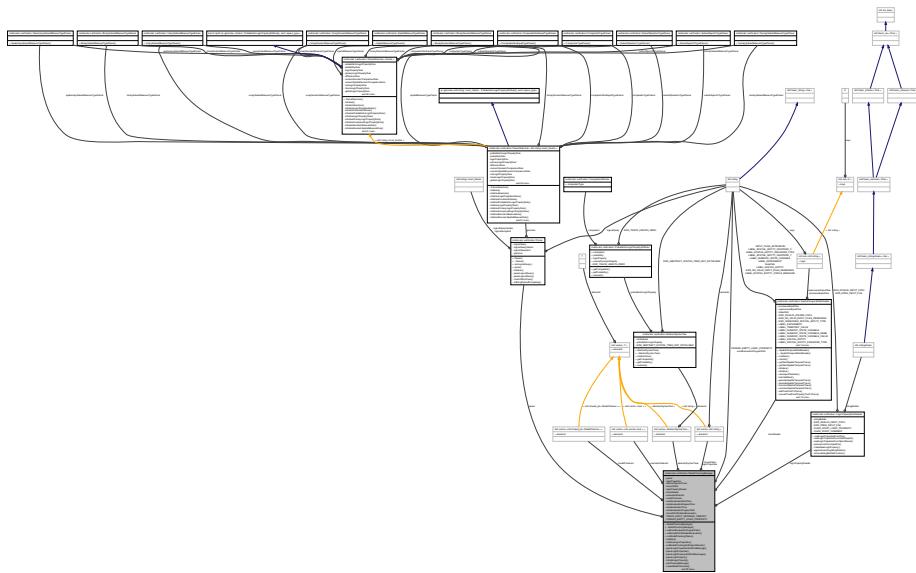
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ModelCheckingHelpRequestException.hpp](#)

7.80 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`)
- `~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`)
Initialise the model checking manager considering the given logic properties input file and extra evaluation time, and print the introduction message.
- 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 whenever a logic property was successfully parsed.
- 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 isParsingsSuccessful)

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.

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 and print a message informing the user about this.
- void [waitBeforeRetry \(\)](#)
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`
- 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` = ""

7.80.1 Detailed Description

Class for managing the model checking processes.

Definition at line 23 of file ModelCheckingManager.hpp.

7.80.2 Constructor & Destructor Documentation

7.80.2.1 `ModelCheckingManager::ModelCheckingManager (const std::string & logicPropertiesFilepath, const std::string & tracesFolderPath, unsigned long extraEvaluationTime)`

Definition at line 12 of file ModelCheckingManager.cpp.

References initialise().

7.80.2.2 `ModelCheckingManager::~ModelCheckingManager ()`

Definition at line 20 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, logicProperties, modelCheckers, and tracesPaths.

7.80.3 Member Function Documentation

7.80.3.1 `bool ModelCheckingManager::areUnfinishedModelCheckingTasks () [private]`

Check if there exist model checkers which require extra traces.

Definition at line 241 of file ModelCheckingManager.cpp.

References modelCheckers.

Referenced by runModelCheckers(), and runModelCheckersAndRequestAdditionalTraces().

7.80.3.2 `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>model- Checker- Factory</i>	The factory used to create model checkers
--	---

Definition at line 124 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, and modelCheckers.

Referenced by runModelCheckingAndOutputResults().

7.80.3.3 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 178 of file ModelCheckingManager.cpp.

References evaluationResults, modelCheckers, and shouldPrintDetailedEvaluation.

Referenced by runModelCheckersForCurrentlyExistingTraces().

7.80.3.4 void ModelCheckingManager::executeExtraEvaluationProgram()
[private]

Execute the extra evaluation program for generating potential new traces.

Definition at line 251 of file ModelCheckingManager.cpp.

References executeExtraEvaluationProgramAndPrintMessage(), and extraEvaluationProgramPath.

Referenced by runModelCheckersAndRequestAdditionalTraces().

**7.80.3.5 void ModelCheckingManager::executeExtraEvaluationProgramAndPrint-
Message()** [private]

Execute the extra evaluation program for generating potential new traces and print a message informing the user about this.

Definition at line 257 of file ModelCheckingManager.cpp.

References multiscale::OperatingSystem::executeProgram(), extraEvaluationProgramPath, and multiscale::verification::ModelCheckingOutputWriter::printExecuteExtraEvaluationProgramMessage().

Referenced by executeExtraEvaluationProgram().

7.80.3.6 SpatialTemporalTrace ModelCheckingManager::getNextSpatialTemporalTrace() [private]

Get the next spatial temporal trace and store its path.

Definition at line 157 of file ModelCheckingManager.cpp.

References multiscale::verification::SpatialTemporalDataReader::getNextSpatialTemporalTrace(), multiscale::verification::ModelCheckingOutputWriter::printStartTraceEvaluationMessage(), storeNewSpatialTemporalTracePath(), and traceReader.

Referenced by runModelCheckersForCurrentlyExistingTraces().

7.80.3.7 void ModelCheckingManager::initialise (const std::string & logicPropertiesFilepath, unsigned long extraEvaluationTime) [private]

Initialise the model checking manager considering the given logic properties input file and extra evaluation time, and print the introduction message.

Parameters

<i>logic-Properties-Filepath</i>	The path to the logic properties input file
<i>extra-Evaluation-Time</i>	The number of extra minutes allocated for evaluating logic properties

Definition at line 39 of file ModelCheckingManager.cpp.

References extraEvaluationElapsedTime, extraEvaluationStartTime, extraEvaluationTime, initialiseLogicProperties(), and shouldPrintDetailedEvaluation.

Referenced by ModelCheckingManager().

7.80.3.8 void ModelCheckingManager::initialiseLogicProperties (const std::string & logicPropertiesFilepath) [private]

Initialise the logic properties using the provided input file.

Parameters

<i>logic-Properties-Filepath</i>	The path to the logic properties input file
----------------------------------	---

Definition at line 50 of file ModelCheckingManager.cpp.

References logicProperties, logicPropertyReader, and multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile().

Referenced by initialise().

7.80.3.9 bool ModelCheckingManager::isEvaluationTimeRemaining ()
 [private]

Check if there is evaluation time remaining.

Definition at line 232 of file ModelCheckingManager.cpp.

References extraEvaluationElapsedTime, extraEvaluationStartTime, and extraEvaluationTime.

Referenced by runModelCheckersAndRequestAdditionalTraces().

7.80.3.10 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 caught in this method

Parameters

<i>logic- Property</i>	The given logic property
----------------------------	--------------------------

Definition at line 102 of file ModelCheckingManager.cpp.

References abstractSyntaxTrees, multiscale::verification::Parser::parse(), parser, and multiscale::verification::Parser::setLogicalQuery().

Referenced by parseLogicProperty().

7.80.3.11 void ModelCheckingManager::outputDetailedEvaluationResults ()
 [private]

Output the logic properties detailed evaluation results.

Definition at line 296 of file ModelCheckingManager.cpp.

References evaluationResults, logicProperties, multiscale::verification::ModelCheckingOutputWriter::printDetailedEvaluationResults(), shouldPrintDetailedEvaluation, and tracesPaths.

Referenced by runModelCheckingAndOutputResults().

7.80.3.12 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>model-Checker</i>	The given model checker
<i>logic-Property</i>	The logic property verified by the given model checker

Definition at line 287 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultMessage().

Referenced by outputModelCheckersResults().

7.80.3.13 void ModelCheckingManager::outputModelCheckersResults() [private]

Output the model checking results.

Definition at line 279 of file ModelCheckingManager.cpp.

References logicProperties, modelCheckers, and outputModelCheckerResults().

Referenced by outputModelCheckersResultsAndPrintMessage().

7.80.3.14 void ModelCheckingManager::outputModelCheckersResultsAndPrintMessage() [private]

Output the model checking results and print the message informing the user about this.

Definition at line 273 of file ModelCheckingManager.cpp.

References outputModelCheckersResults(), and multiscale::verification::ModelCheckingOutputWriter::printModelCheckingResultsIntroductionMessage().

Referenced by runModelCheckingAndOutputResults().

7.80.3.15 void ModelCheckingManager::parseLogicProperties() [private]

Parse the logic properties and create abstract syntax trees whenever a logic property was successfully parsed.

Definition at line 70 of file ModelCheckingManager.cpp.

References logicProperties, and parseLogicPropertyAndPrintMessages().

Referenced by parseLogicPropertiesAndPrintMessage().

7.80.3.16 void ModelCheckingManager::parseLogicPropertiesAndPrintMessage() [private]

Parse the logic properties and print message informing the user about this.

Definition at line 62 of file ModelCheckingManager.cpp.

References `parseLogicProperties()`, `multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesBeginMessage()`, and `multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertiesEndMessage()`.

Referenced by `runModelCheckingAndOutputResults()`.

7.80.3.17 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 caught in this method

Parameters

<i>logic- Property</i>	The given logic property
----------------------------	--------------------------

Definition at line 92 of file ModelCheckingManager.cpp.

References `isValidLogicProperty()`, and `multiscale::ExceptionHandler::printErrorMessage()`.

Referenced by `parseLogicPropertyAndPrintMessages()`.

7.80.3.18 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>logic- Property</i>	The given logic property
----------------------------	--------------------------

Definition at line 82 of file ModelCheckingManager.cpp.

References `parseLogicProperty()`, `multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage()`, and `printParsingMessage()`.

Referenced by `parseLogicProperties()`.

7.80.3.19 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 116 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printFailedMessage(), and multiscale::verification::ModelCheckingOutputWriter::printSuccessMessage().

Referenced by parseLogicPropertyAndPrintMessages().

7.80.3.20 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 198 of file ModelCheckingManager.cpp.

References modelCheckers, shouldPrintDetailedEvaluation, and updateEvaluationResults().

Referenced by runModelCheckersForTrace().

7.80.3.21 void ModelCheckingManager::runModelCheckers () [private]

Run the model checkers and verify the logic properties.

Definition at line 138 of file ModelCheckingManager.cpp.

References areUnfinishedModelCheckingTasks(), runModelCheckersAndRequestAdditionalTraces(), and runModelCheckersForCurrentlyExistingTraces().

Referenced by runModelCheckersAndPrintMessage().

7.80.3.22 void ModelCheckingManager::runModelCheckersAndPrintMessage () [private]

Run the model checkers and print a message informing the user about it.

Definition at line 132 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printStartModelCheckingExecutionMessage(), and runModelCheckers().

Referenced by runModelCheckingAndOutputResults().

7.80.3.23 void ModelCheckingManager::runModelCheckersAndRequestAdditionalTraces() [private]

Run the model checkers and request additional traces.

Definition at line 216 of file ModelCheckingManager.cpp.

References `areUnfinishedModelCheckingTasks()`, `executeExtraEvaluationProgram()`, `isEvaluationTimeRemaining()`, `runModelCheckersForCurrentlyExistingTraces()`, `updateExtraEvaluationStartTime()`, `updateTraceReader()`, and `waitBeforeRetry()`.

Referenced by `runModelCheckers()`.

7.80.3.24 void ModelCheckingManager::runModelCheckersForCurrentlyExistingTraces() [private]

Run the model checkers and verify the logic properties for the currently existing traces.

Definition at line 146 of file ModelCheckingManager.cpp.

References `createNewEvaluationResults()`, `getNextSpatialTemporalTrace()`, `multiscale::verification::SpatialTemporalDataReader::hasNext()`, `runModelCheckersForTrace()`, and `traceReader`.

Referenced by `runModelCheckers()`, and `runModelCheckersAndRequestAdditionalTraces()`.

7.80.3.25 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 184 of file ModelCheckingManager.cpp.

References `modelCheckers`, and `runModelCheckerForTrace()`.

Referenced by `runModelCheckersForCurrentlyExistingTraces()`.

7.80.3.26 void ModelCheckingManager::runModelCheckingAndOutputResults (const std::shared_ptr< ModelCheckerFactory > & modelCheckerFactory) [private]

Run the model checking tasks and output the results.

Parameters

<i>model- Checker- Factory</i>	The factory used to create model checkers
--	---

Definition at line 54 of file ModelCheckingManager.cpp.

References `createModelCheckers()`, `outputDetailedEvaluationResults()`, `outputModelCheckersResultsAndPrintMessage()`, `parseLogicPropertiesAndPrintMessage()`, and `runModelCheckersAndPrintMessage()`.

Referenced by `runModelCheckingTasks()`.

7.80.3.27 void ModelCheckingManager::runModelCheckingTasks (const std::shared_ptr< ModelCheckerFactory > & modelCheckerFactory)

Run the model checking tasks.

Parameters

<i>model- Checker- Factory</i>	The factory used to create model checkers
--	---

Definition at line 35 of file ModelCheckingManager.cpp.

References `runModelCheckingAndOutputResults()`.

7.80.3.28 void ModelCheckingManager::setExtraEvaluationProgramPath (const std::string & extraEvaluationProgramPath)

Set the path of the program which should be executed whenever extra evaluation is required.

Parameters

<i>extra- Evaluation- Program- Path</i>	The path to the program which will be executed when extra evaluation is required
---	--

Definition at line 27 of file ModelCheckingManager.cpp.

References `extraEvaluationProgramPath`.

7.80.3.29 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 31 of file ModelCheckingManager.cpp.

References shouldPrintDetailedEvaluation.

7.80.3.30 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 172 of file ModelCheckingManager.cpp.

References shouldPrintDetailedEvaluation, and tracesPaths.

Referenced by getNextSpatialTemporalTrace().

7.80.3.31 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 208 of file ModelCheckingManager.cpp.

References evaluationResults.

Referenced by runModelCheckerForTrace().

7.80.3.32 **void ModelCheckingManager::updateExtraEvaluationStartTime()**
[private]

Set the extra evaluation start time equal to current time.

Definition at line 228 of file ModelCheckingManager.cpp.

References extraEvaluationStartTime.

Referenced by runModelCheckersAndRequestAdditionalTraces().

7.80.3.33 **void ModelCheckingManager::updateTraceReader()** [private]

Update trace reader.

Definition at line 269 of file ModelCheckingManager.cpp.

References multiscale::verification::SpatialTemporalDataReader::refresh(), and trace-Reader.

Referenced by runModelCheckersAndRequestAdditionalTraces().

7.80.3.34 **void ModelCheckingManager::waitBeforeRetry()** [private]

Wait TRACE_INPUT_REFRESH_TIMEOUT minutes before updating the trace reader.

Definition at line 263 of file ModelCheckingManager.cpp.

References multiscale::verification::ModelCheckingOutputWriter::printTimeoutMessage(), and TRACE_INPUT_REFRESH_TIMEOUT.

Referenced by runModelCheckersAndRequestAdditionalTraces().

7.80.4 Member Data Documentation

7.80.4.1 **std::vector<AbstractSyntaxTree> multiscale::verification-
::ModelCheckingManager::abstractSyntaxTrees**
[private]

The collection of abstract syntax tree obtained after parsing the logic properties

Definition at line 31 of file ModelCheckingManager.hpp.

Referenced by createModelCheckers(), isValidLogicProperty(), and ~ModelCheckingManager().

7.80.4.2 **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 38 of file ModelCheckingManager.hpp.

Referenced by createNewEvaluationResults(), outputDetailedEvaluationResults(), and updateEvaluationResults().

7.80.4.3 double multiscale::verification::ModelCheckingManager::extraEvaluationElapsed Time [private]

The elapsed time for the extra evaluation process expressed in seconds

Definition at line 46 of file ModelCheckingManager.hpp.

Referenced by initialise(), and isEvaluationTimeRemaining().

7.80.4.4 std::string multiscale::verification::ModelCheckingManager::extraEvaluationProgramPath [private]

The path to the program which should be executed when extra evaluation is required

Definition at line 50 of file ModelCheckingManager.hpp.

Referenced by executeExtraEvaluationProgram(), executeExtraEvaluationProgramAndPrintMessage(), and setExtraEvaluationProgramPath().

7.80.4.5 std::chrono::time_point<std::chrono::system_clock> multiscale::verification::ModelCheckingManager::extraEvaluationStartTime [private]

The start time for the current evaluation process

Definition at line 45 of file ModelCheckingManager.hpp.

Referenced by initialise(), isEvaluationTimeRemaining(), and updateExtraEvaluationStartTime().

7.80.4.6 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 48 of file ModelCheckingManager.hpp.

Referenced by initialise(), and isEvaluationTimeRemaining().

7.80.4.7 std::vector<std::string> multiscale::verification::ModelCheckingManager::logicProperties [private]

The collection of logic properties

Definition at line 30 of file ModelCheckingManager.hpp.

Referenced by initialiseLogicProperties(), outputDetailedEvaluationResults(), outputModelCheckersResults(), parseLogicProperties(), and ~ModelCheckingManager().

**7.80.4.8 LogicPropertyDataReader multiscale::verification-
::ModelCheckingManager::logicPropertyReader
[private]**

The logic property reader

Definition at line 35 of file ModelCheckingManager.hpp.

Referenced by initialiseLogicProperties().

**7.80.4.9 std::vector<std::shared_ptr<ModelChecker> > multiscale-
::verification::ModelCheckingManager::modelCheckers
[private]**

The collection of model checkers

Definition at line 43 of file ModelCheckingManager.hpp.

Referenced by areUnfinishedModelCheckingTasks(), createModelCheckers(), createNewEvaluationResults(), outputModelCheckersResults(), runModelCheckerForTrace(), runModelCheckersForTrace(), and ~ModelCheckingManager().

**7.80.4.10 Parser multiscale::verification::ModelCheckingManager::parser
[private]**

The parser used to verify if logical properties are syntactically correct

Definition at line 27 of file ModelCheckingManager.hpp.

Referenced by isValidLogicProperty().

**7.80.4.11 const std::string ModelCheckingManager::PARSE-
R_EMPTY_LOGIC_PROPERTY = "" [static,
private]**

An empty logic property

Definition at line 235 of file ModelCheckingManager.hpp.

**7.80.4.12 bool multiscale::verification::ModelCheckingManager::shouldPrint-
DetailedEvaluation [private]**

Flag indicating if detailed evaluation results should be printed

Definition at line 53 of file ModelCheckingManager.hpp.

Referenced by `createNewEvaluationResults()`, `initialise()`, `outputDetailedEvaluationResults()`, `runModelCheckerForTrace()`, `setShouldPrintDetailedEvaluation()`, and `storeNewSpatialTemporalTracePath()`.

**7.80.4.13 const unsigned long ModelCheckingManager::TRA-
CE_INPUT_REFRESH_TIMEOUT = 30 [static,
private]**

The number of seconds for which the manager waits before updating the trace reader
Definition at line 233 of file `ModelCheckingManager.hpp`.
Referenced by `waitBeforeRetry()`.

**7.80.4.14 SpatialTemporalDataReader multiscale::verification::ModelChecking-
Manager::traceReader [private]**

The behaviour/trace reader

Definition at line 36 of file `ModelCheckingManager.hpp`.
Referenced by `getNextSpatialTemporalTrace()`, `runModelCheckersForCurrently-
ExistingTraces()`, and `updateTraceReader()`.

**7.80.4.15 std::vector<std::string> multiscale::verification::ModelCheckingManager-
::tracesPaths [private]**

The collection of traces paths

Definition at line 33 of file `ModelCheckingManager.hpp`.
Referenced by `outputDetailedEvaluationResults()`, `storeNewSpatialTemporalTrace-
Path()`, and `~ModelCheckingManager()`.

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

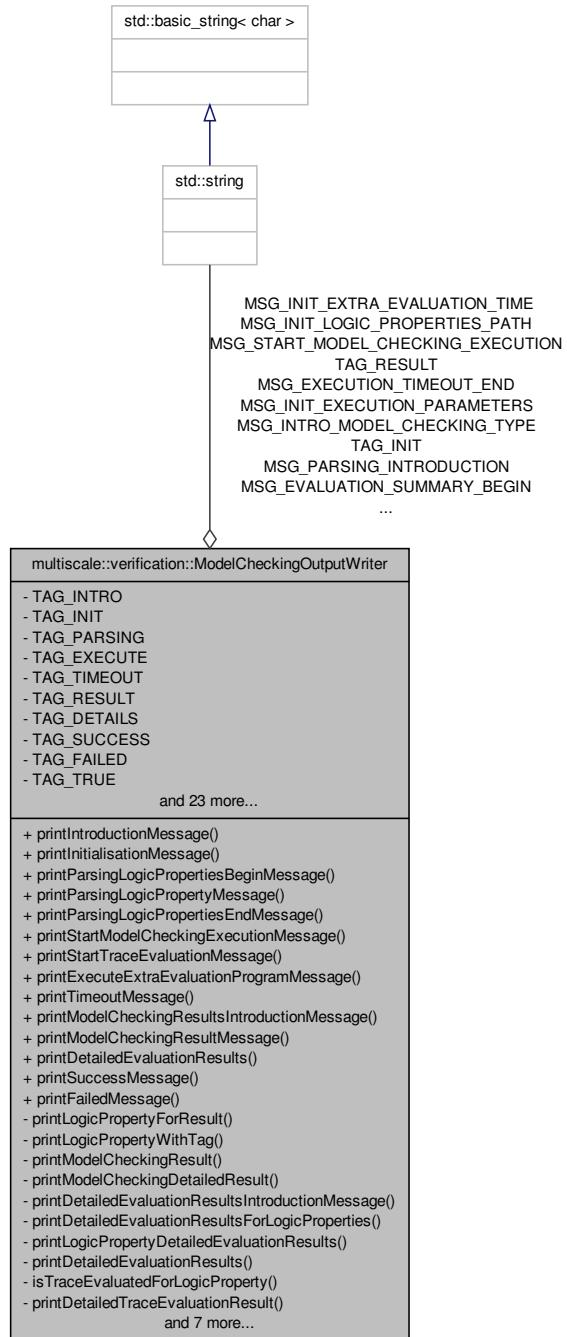
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[ModelCheckingManager.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[ModelCheckingManager.cpp](#)

7.81 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)
Print the model checker initialisation message.
- static void [printParsingLogicPropertiesBeginMessage](#) ()
Print an introduction message informing the user that the logic properties will be parsed.
- static void [printParsingLogicPropertyMessage](#) (const std::string &logicProperty)
Print a message informing the user which logic property will be parsed.
- static void [printParsingLogicPropertiesEndMessage](#) ()
Print a closing message after the logic properties were parsed.
- static void [printStartModelCheckingExecutionMessage](#) ()
Print a message informing the user that the model checking execution has started.
- static void [printStartTraceEvaluationMessage](#) (const std::string &tracePath)
Print a message informing the user which trace will be evaluated next by the model checkers.
- static void [printExecuteExtraEvaluationProgramMessage](#) (const std::string &programPath)
Print a message informing the user that the extra evaluation program located at the given path will be executed.
- static void [printTimeoutMessage](#) (unsigned long timeOut)
Print a message informing the user that the model checking execution is suspended for timeOut seconds.
- 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 of the evaluation results for the given logic property.

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 1.0.101 (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`"

7.81.1 Detailed Description

Class used to output the model checkers progress.

Definition at line 12 of file ModelCheckingOutputWriter.hpp.

7.81.2 Member Function Documentation

```
7.81.2.1 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>logic-Property-Index</i>	The index of the logic property in the collection of logic properties
<i>tracePath-Index</i>	The index of the trace path in the collection of trace paths
<i>evaluation-Results</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()`.

```
7.81.2.2 bool ModelCheckingOutputWriter::isTraceEvaluatedTrueForLogic-
    Property ( 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>logic-Property-Index</i>	The index of the logic property in the collection of logic properties
<i>tracePath-Index</i>	The index of the trace path in the collection of trace paths
<i>evaluation-Results</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 250 of file ModelCheckingOutputWriter.cpp.

References `isTraceEvaluatedForLogicProperty()`.

Referenced by `updateSummaryEvaluationResults()`.

```
7.81.2.3 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>logic-Properties</i>	The collection of logic properties
<i>tracesPaths</i>	The collection of trace paths
<i>evaluation-Results</i>	The evaluation results (i.e. a two-dimensional array of size $ \text{logic-Properties} \times 2 * \text{traces} $ where the first boolean value associated to a (<i>logicProperty</i> , <i>trace</i>) 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()`.

```
7.81.2.4 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>logic-Property-Index</i>	The index of the logic property in the collection of logic properties
<i>tracesPaths</i>	The collection of trace paths
<i>evaluation-Results</i>	The evaluation results (i.e. a two-dimensional array of size $ \text{logic-Properties} \times 2 * \text{traces} $ where the first boolean value associated to a (<i>logicProperty</i> , <i>trace</i>) 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()`.

```
7.81.2.5 void ModelCheckingOutputWriter::printDetailedEvaluationResultsFor-
    LogicProperties ( 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>logic-Properties</i>	The collection of logic properties
<i>tracesPaths</i>	The collection of trace paths
<i>evaluation-Results</i>	The evaluation results (i.e. a two-dimensional array of size $ \text{logic-Properties} \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().

7.81.2.6 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().

7.81.2.7 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>logic-Property-Index</i>	The index of the logic property in the collection of logic properties
<i>tracePath</i>	The path to the spatial temporal trace
<i>tracePath-Index</i>	The index of the trace path in the collection of trace paths
<i>evaluation-Results</i>	The evaluation results (i.e. a two-dimensional array of size $ \text{logic-Properties} \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().

```
7.81.2.8 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>logic- Property- Index</i>	The index of the logic property in the collection of logic properties
<i>tracesPaths</i>	The collection of trace paths
<i>evaluation- Results</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 182 of file ModelCheckingOutputWriter.cpp.

References updateSummaryEvaluationResults().

Referenced by printLogicPropertyDetailedEvaluationResults().

```
7.81.2.9 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>nrOfCorrect- Traces</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().

7.81.2.10 void ModelCheckingOutputWriter::printExecuteExtra-EvaluationProgramMessage (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().

7.81.2.11 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().

7.81.2.12 void ModelCheckingOutputWriter::printInitialisationMessage (const std::string & *logicProperty*, const std::string & *tracesFolderPath*, unsigned long *extraEvaluationTime*) [static]

Print the model checker initialisation message.

Parameters

<i>logic-Property</i>	The path to the input file containing logic properties
<i>traces-FolderPath</i>	The path to the folder containing the traces
<i>extra-Evaluation-Time</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().

7.81.2.13 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>model- Checker- Type</i>	The type of the model checker
<i>model- Checker- Parameters</i>	The model checking parameters

Definition at line 8 of file ModelCheckingOutputWriter.cpp.

References multiscale::CYAN, MSG_INTRO_CONTACT, MSG_INTRO_COPYRIGHT, MSG_INTRO_MODEL_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().

7.81.2.14 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>logic- Property- Index</i>	The index of the logic property in the collection of logic properties
<i>tracesPaths</i>	The collection of trace paths

<i>evaluation-Results</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 163 of file ModelCheckingOutputWriter.cpp.

References printDetailedEvaluationResults(), and printEvaluationResultsSummary().

Referenced by printDetailedEvaluationResultsForLogicProperties().

7.81.2.15 void ModelCheckingOutputWriter::printLogicPropertyForResult (const std::string & *logicProperty*) [static, private]

Print the given logic property in the context of a result message.

Parameters

<i>logic-Property</i>	The logic property
-----------------------	--------------------

Definition at line 116 of file ModelCheckingOutputWriter.cpp.

References printLogicPropertyWithTag(), and TAG_RESULT.

Referenced by printModelCheckingResultMessage().

7.81.2.16 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>logic-Property</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().

7.81.2.17 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().

7.81.2.18 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>does- Property- Hold</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().

7.81.2.19 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>does- Property- Hold</i>	The flag indicating if the logic property holds (with a given probability and/or confidence)
<i>detailed- Result</i>	The detailed result report indicating if the logic property holds (with a given probability and/or confidence)
<i>logic- Property</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().

7.81.2.20 void ModelCheckingOutputWriter::printModelCheckingResults-IntroductionMessage() [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().

7.81.2.21 void ModelCheckingOutputWriter::printParsingLogicPropertiesBegin-Message() [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::printMessage-WithColouredTag(), printSeparatorTag(), and TAG_PARSING.

Referenced by multiscale::verification::ModelCheckingManager::parseLogicProperties-AndPrintMessage().

7.81.2.22 void ModelCheckingOutputWriter::printParsingLogicPropertiesEnd-Message() [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::parseLogicProperties-AndPrintMessage().

7.81.2.23 void ModelCheckingOutputWriter::printParsingLogicPropertyMessage (const std::string & *logicProperty*) [static]

Print a message informing the user which logic property will be parsed.

Parameters

<i>logic- Property</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().

7.81.2.24 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().

7.81.2.25 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().

**7.81.2.26 void ModelCheckingOutputWriter::printStartModelCheckingExecution-
Message() [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().

7.81.2.27 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().

7.81.2.28 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().

7.81.2.29 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().

7.81.2.30 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().

7.81.2.31 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().

7.81.2.32 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>logic-Property-Index</i>	The index of the logic property in the collection of logic properties
<i>tracePath-Index</i>	The index of the trace path in the collection of trace paths
<i>evaluation-Results</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>nrOf-Evaluated-Traces</i>	The number of evaluated traces
<i>nrOfTraces-Evaluated-True</i>	The number of traces evaluated true

Definition at line 197 of file ModelCheckingOutputWriter.cpp.

References `isTraceEvaluatedForLogicProperty()`, and `isTraceEvaluatedTrueForLogicProperty()`.

Referenced by `printEvaluationResultsSummary()`.

7.81.3 Member Data Documentation

7.81.3.1 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()`.

7.81.3.2 const std::string ModelCheckingOutputWriter::MSG_EVALUATION_SUMMARY_BEGIN = "/" [static, private]

Definition at line 291 of file ModelCheckingOutputWriter.hpp.

Referenced by `printEvaluationResultsSummary()`.

7.81.3.3 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()`.

```
7.81.3.4 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().

```
7.81.3.5 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().

```
7.81.3.6 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().

```
7.81.3.7 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().

```
7.81.3.8 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().

```
7.81.3.9 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().

```
7.81.3.10 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().

```
7.81.3.11 const std::string ModelCheckingOutputWriter::MSG_INTRO_COPYRIGHT  
= "Copyright Ovidiu Pârvu 2014" [static, private]
```

Definition at line 270 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.81.3.12 const std::string ModelCheckingOutputWriter::MSG_INTRO_MODEL_CHECKING_PARAMETERS  
= "Parameters: " [static, private]
```

Definition at line 272 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.81.3.13 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().

```
7.81.3.14 const std::string ModelCheckingOutputWriter::MSG_INTRO_NAME = "Mule  
1.0.101 (Multidimensional multiscale model checker)" [static, private]
```

Definition at line 269 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.81.3.15 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().

```
7.81.3.16 const std::string ModelCheckingOutputWriter::MSG_LOG-
IC_PROPERTY HOLDS FALSE = "FALSE" [static,
private]
```

Definition at line 296 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResult().

```
7.81.3.17 const std::string ModelCheckingOutputWriter::MSG_OL-
GIC_PROPERTY HOLDS TRUE = "TRUE" [static,
private]
```

Definition at line 295 of file ModelCheckingOutputWriter.hpp.

Referenced by printModelCheckingResult().

```
7.81.3.18 const std::string ModelCheckingOutputWriter::MSG_PARSING_IN-
TRODUCTION = "I am starting to parse logic properties..." [static,
private]
```

Definition at line 280 of file ModelCheckingOutputWriter.hpp.

Referenced by printParsingLogicPropertiesBeginMessage().

```
7.81.3.19 const std::string ModelCheckingOutputWriter::MSG_RESULTS_INTRODU-
CTION = "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().

```
7.81.3.20 const std::string ModelCheckingOutputWriter::MSG_START_EXTRA_EV-
ALUATION_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().

```
7.81.3.21 const std::string ModelCheckingOutputWriter::MSG_START_MODEL_CH-
ECKING_EXECUTION = "I am starting the execution of the model checkers..." [static,
private]
```

Definition at line 282 of file ModelCheckingOutputWriter.hpp.

Referenced by printStartModelCheckingExecutionMessage().

```
7.81.3.22 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().

```
7.81.3.23 const std::string ModelCheckingOutputWriter::TAG_DETAILS = "[ DETAILS
    ]" [static, private]
```

Definition at line 262 of file ModelCheckingOutputWriter.hpp.

Referenced by printDetailedEvaluationResultsForLogicProperties(), printDetailedEvaluationResultsIntroductionMessage(), and printEvaluationResultsSummary().

```
7.81.3.24 const std::string ModelCheckingOutputWriter::TAG_EXECUTE = "[
    EXECUTE]" [static, private]
```

Definition at line 259 of file ModelCheckingOutputWriter.hpp.

Referenced by printExecuteExtraEvaluationProgramMessage(), printStartModelCheckingExecutionMessage(), and printStartTraceEvaluationMessage().

```
7.81.3.25 const std::string ModelCheckingOutputWriter::TAG_FAILED = "[ FAILED ]"
    [static, private]
```

Definition at line 264 of file ModelCheckingOutputWriter.hpp.

Referenced by printFailedMessage().

```
7.81.3.26 const std::string ModelCheckingOutputWriter::TAG_FALSE = "[ FALSE ]"
    [static, private]
```

Definition at line 266 of file ModelCheckingOutputWriter.hpp.

Referenced by printTraceEvaluationResult().

```
7.81.3.27 const std::string ModelCheckingOutputWriter::TAG_INIT = "[ INIT ]"
    [static, private]
```

Definition at line 257 of file ModelCheckingOutputWriter.hpp.

Referenced by printInitialisationMessage().

```
7.81.3.28 const std::string ModelCheckingOutputWriter::TAG_INTRO = "[ INTRO ]"  
[static, private]
```

Definition at line 256 of file ModelCheckingOutputWriter.hpp.

Referenced by printIntroductionMessage().

```
7.81.3.29 const std::string ModelCheckingOutputWriter::TAG_PARSING = "[ PARSING ]"  
[static, private]
```

Definition at line 258 of file ModelCheckingOutputWriter.hpp.

Referenced by printParsingLogicPropertiesBeginMessage(), and printParsingLogicPropertyMessage().

```
7.81.3.30 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().

```
7.81.3.31 const std::string ModelCheckingOutputWriter::TAG_SEPARATOR =  
"[=====]" [static, private]
```

Definition at line 267 of file ModelCheckingOutputWriter.hpp.

Referenced by printSeparatorTag().

```
7.81.3.32 const std::string ModelCheckingOutputWriter::TAG_SUCCESS = "[  
SUCCESS ]" [static, private]
```

Definition at line 263 of file ModelCheckingOutputWriter.hpp.

Referenced by printSuccessMessage().

```
7.81.3.33 const std::string ModelCheckingOutputWriter::TAG_TIMEOUT = "[ TIMEOUT ]"  
[static, private]
```

Definition at line 260 of file ModelCheckingOutputWriter.hpp.

Referenced by printTimeoutMessage().

7.81.3.34 `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
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.in.cpp

7.82 multiscale::MultiplicationOperation Class Reference

Functor representing a multiplication operation.

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

Public Member Functions

- template<typename Operand >
`Operand operator()` (`Operand operand1, Operand operand2`) const
Multiply the two operands.

7.82.1 Detailed Description

Functor representing a multiplication operation.

Definition at line 52 of file Numeric.hpp.

7.82.2 Member Function Documentation

7.82.2.1 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 62 of file Numeric.hpp.

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

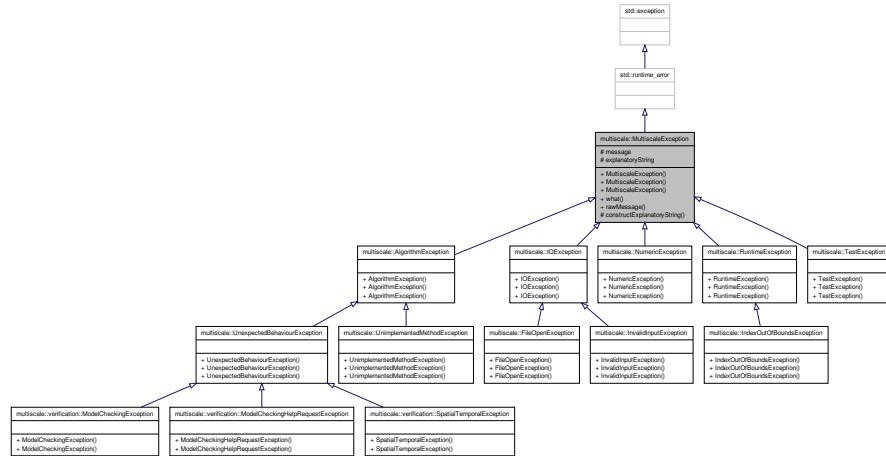
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
[Numeric.hpp](#)

7.83 multiscale::MultiscaleException Class Reference

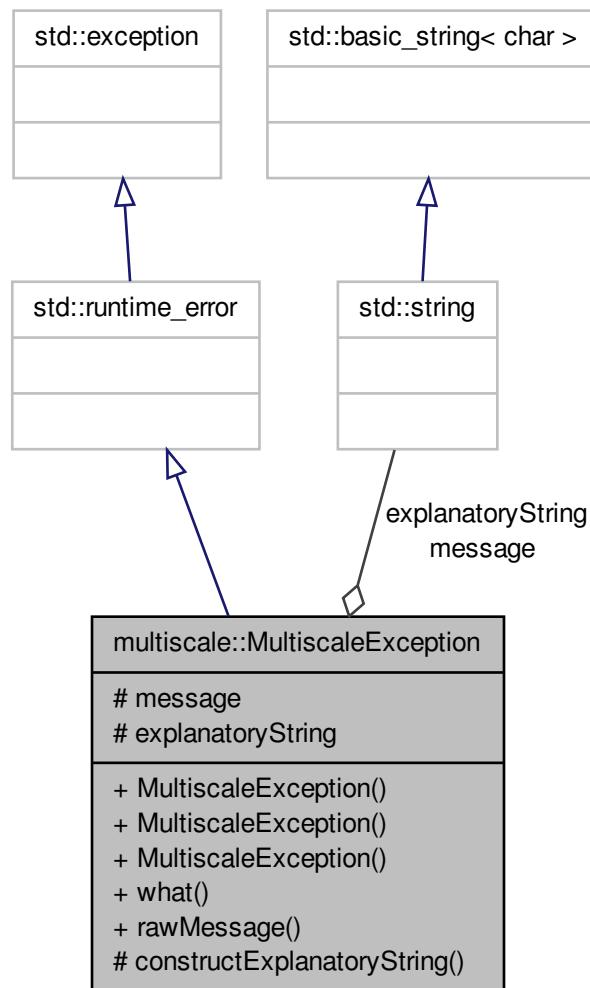
Parent exception class for the project.

```
#include <MultiscaleException.hpp>
```

Inheritance diagram for multiscale::MultiscaleException:



Collaboration diagram for multiscale::MultiscaleException:



Public Member Functions

- `MultiscaleException ()`
- `MultiscaleException (const string &file, int line, const string &msg)`
- `MultiscaleException (const string &file, int line, const char *msg)`
- `const char * what () const noexcept override`

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 string &file, int line, T msg)`

Construct the explanatory string.

Protected Attributes

- string `message`
- string `explanatoryString`

7.83.1 Detailed Description

Parent exception class for the project.

Definition at line 19 of file MultiscaleException.hpp.

7.83.2 Constructor & Destructor Documentation

7.83.2.1 `multiscale::MultiscaleException::MultiscaleException () [inline]`

Definition at line 28 of file MultiscaleException.hpp.

7.83.2.2 `multiscale::MultiscaleException::MultiscaleException (const string & file, int line, const string & msg) [inline, explicit]`

Definition at line 30 of file MultiscaleException.hpp.

7.83.2.3 `multiscale::MultiscaleException::MultiscaleException (const string & file, int line, const char * msg) [inline, explicit]`

Definition at line 32 of file MultiscaleException.hpp.

7.83.3 Member Function Documentation

7.83.3.1 `template<typename T > void multiscale::MultiscaleException::constructExplanatoryString (const 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 54 of file MultiscaleException.hpp.

7.83.3.2 std::string multiscale::MultiscaleException::rawMessage() const [inline]

Return the raw message of the exception.

Definition at line 41 of file MultiscaleException.hpp.

Referenced by multiscale::OperatingSystem::executeProgram().

7.83.3.3 const char* multiscale::MultiscaleException::what() const [inline, override]

Returns an explanatory string.

Definition at line 36 of file MultiscaleException.hpp.

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

7.83.4 Member Data Documentation

7.83.4.1 string multiscale::MultiscaleException::explanatoryString [protected]

User friendly exception message

Definition at line 24 of file MultiscaleException.hpp.

Referenced by multiscale::verification::ModelCheckingException::ModelCheckingException(), multiscale::verification::ModelCheckingHelpRequestException::ModelCheckingHelpRequestException(), and multiscale::verification::SpatialTemporalException::SpatialTemporalException().

7.83.4.2 string multiscale::MultiscaleException::message [protected]

The raw message of the exception

Definition at line 23 of file MultiscaleException.hpp.

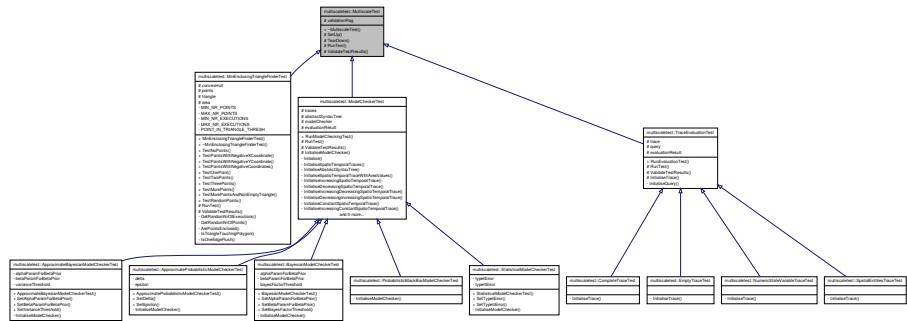
The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/[MultiscaleException.hpp](#)

7.84 multiscaletest::MultiscaleTest Class Reference

```
#include <MultiscaleTest.hpp>
```

Inheritance 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

7.84.1 Detailed Description

Definition at line 8 of file MultiscaleTest.hpp.

7.84.2 Constructor & Destructor Documentation

7.84.2.1 virtual multiscaletest::MultiscaleTest::~MultiscaleTest() [inline, virtual]

Definition at line 16 of file MultiscaleTest.hpp.

7.84.3 Member Function Documentation

7.84.3.1 virtual void multiscaletest::MultiscaleTest::RunTest() [protected, pure virtual]

Run the test.

Implemented in [multiscaletest::MinEnclosingTriangleFinderTest](#), [multiscaletest::ModelCheckerTest](#), and [multiscaletest::TraceEvaluationTest](#).

7.84.3.2 virtual void multiscaletest::MultiscaleTest::SetUp() [inline, protected, virtual]

Definition at line 20 of file MultiscaleTest.hpp.

7.84.3.3 virtual void multiscaletest::MultiscaleTest::TearDown() [inline, protected, virtual]

Definition at line 21 of file MultiscaleTest.hpp.

7.84.3.4 virtual void multiscaletest::MultiscaleTest::ValidateTestResults() [protected, pure virtual]

Validate the results of the test.

Implemented in [multiscaletest::MinEnclosingTriangleFinderTest](#), [multiscaletest::ModelCheckerTest](#), and [multiscaletest::TraceEvaluationTest](#).

7.84.4 Member Data Documentation

7.84.4.1 bool multiscaletest::MultiscaleTest::validationFlag [protected]

Flag indicating if the test results are valid

Definition at line 12 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](#)

7.85 multiscale::verification::NextKLogicPropertyAttribute Class - Reference

Class for representing a "next K" logic property attribute.

```
#include <NextKLogicPropertyAttribute.hpp>
```

Public Attributes

- unsigned long [nrOfTimepointsAhead](#)
- [LogicPropertyAttributeType logicProperty](#)

7.85.1 Detailed Description

Class for representing a "next K" logic property attribute.

Definition at line 14 of file [NextKLogicPropertyAttribute.hpp](#).

7.85.2 Member Data Documentation

7.85.2.1 [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\(\)](#).

7.85.2.2 [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*](#)

7.86 [multiscale::verification::NextLogicPropertyAttribute Class Reference](#)

Class for representing a "next" logic property attribute.

```
#include <NextLogicPropertyAttribute.hpp>
```

Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

7.86.1 Detailed Description

Class for representing a "next" logic property attribute.

Definition at line 14 of file [NextLogicPropertyAttribute.hpp](#).

7.86.2 Member Data Documentation

7.86.2.1 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](#)

7.87 multiscale::verification::Nil Class Reference

A class used to avoid run-time errors when defining a variant type.

```
#include <Nil.hpp>
```

7.87.1 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](#)

7.88 multiscale::verification::NotConstraintAttribute Class - Reference

Class for representing a "not" constraint attribute.

```
#include <NotConstraintAttribute.hpp>
```

Public Attributes

- [ConstraintAttributeType constraint](#)

7.88.1 Detailed Description

Class for representing a "not" constraint attribute.

Definition at line 14 of file NotConstraintAttribute.hpp.

7.88.2 Member Data Documentation

7.88.2.1 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](#)

7.89 multiscale::verification::NotLogicPropertyAttribute Class - Reference

Class for representing a "not" logic property attribute.

```
#include <NotLogicPropertyAttribute.hpp>
```

Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

7.89.1 Detailed Description

Class for representing a "not" logic property attribute.

Definition at line 14 of file NotLogicPropertyAttribute.hpp.

7.89.2 Member Data Documentation

7.89.2.1 LogicPropertyAttributeType multiscale::verification::NotLogicProperty-Attribute::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:

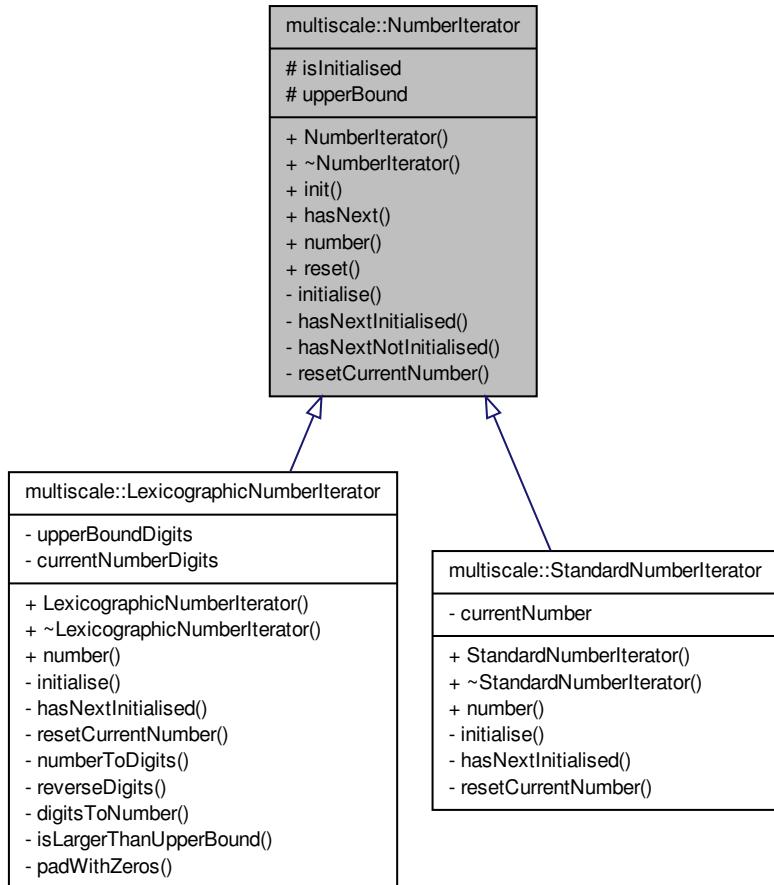
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[NotLogicPropertyAttribute.hpp](#)

7.90 multiscale::NumberIterator Class Reference

Abstract class representing a number iterator.

```
#include <NumberIterator.hpp>
```

Inheritance 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.

7.90.1 Detailed Description

Abstract class representing a number iterator.

Definition at line 7 of file NumberIterator.hpp.

7.90.2 Constructor & Destructor Documentation

7.90.2.1 NumberIterator::NumberIterator (`unsigned int upperBound`)

Definition at line 6 of file NumberIterator.cpp.

References `init()`.

7.90.2.2 virtual multiscale::NumberIterator::~NumberIterator () [inline, virtual]

Definition at line 17 of file NumberIterator.hpp.

7.90.3 Member Function Documentation

7.90.3.1 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()`.

7.90.3.2 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()`.

7.90.3.3 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()`.

7.90.3.4 void NumberIterator::init (unsigned int *upperBound*)

Initialise the iterator considering the given upper bound.

Parameters

<code>upperBound</code>	The upper bound
-------------------------	-----------------

Definition at line 10 of file `NumberIterator.cpp`.

References `upperBound`.

Referenced by `NumberIterator()`.

7.90.3.5 virtual void multiscale::NumberIterator::initialise () [private, pure virtual]

Initialisation of the members of the class.

Implemented in `multiscale::LexicographicNumberIterator`, and `multiscale::StandardNumberIterator`.

7.90.3.6 virtual unsigned int multiscale::NumberIterator::number () [pure virtual]

Get the number pointed by the iterator.

Implemented in [multiscale::LexicographicNumberIterator](#), and [multiscale::StandardNumberIterator](#).

7.90.3.7 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\(\)](#).

7.90.3.8 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\(\)](#).

7.90.4 Member Data Documentation

7.90.4.1 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\(\)](#).

7.90.4.2 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

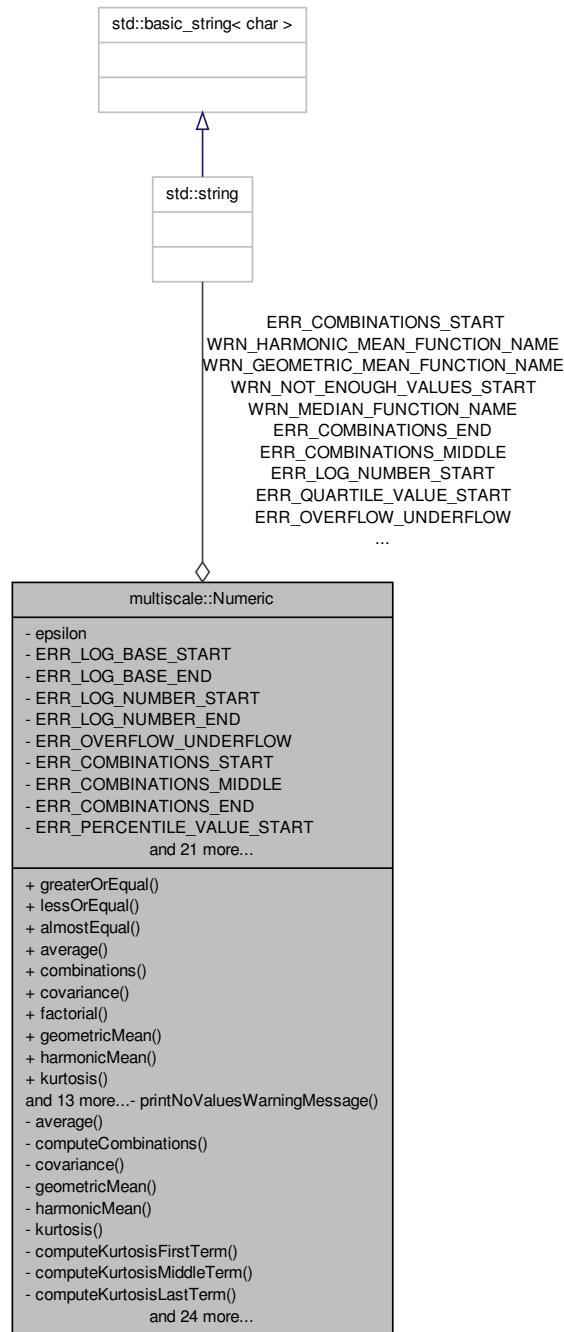
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[Number-Iterator.cpp](#)

7.91 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 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)

Return the sum of the provided numbers.
- static double `variance` (const std::vector< double > &numbers, unsigned int nrOfValues)

Return the variance of the provided set of values.
- 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"

7.91.1 Detailed Description

Class for processing numeric (shorts, ints, floats, doubles etc.) expressions.

Definition at line 87 of file Numeric.hpp.

7.91.2 Member Function Documentation

7.91.2.1 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 ($\text{Abs}(x - y) \leq \text{EPSILON} * \max(1.0f, \text{Abs}(x), \text{Abs}(y))$).

Parameters

<code>number1</code>	First number
<code>number2</code>	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()`, `computeQuartileValue()`, `multiscale::verification::ComparatorEvaluator::evaluate()`, `greaterOrEqual()`, `multiscale::MinEnclosingTriangleFinder::intersects()`, `multiscale::MinEnclosingTriangleFinder::isGammaAngleEqualTo()`, `multiscale::Geometry2D::isPointOnLineSegment()`, `lessOrEqual()`, `multiscale::Geometry2D::lineIntersection()`, `TEST()`, `validateLogBase()`, and `validateQuartile()`.

**7.91.2.2 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

<code>operation</code>	The operation
<code>operand1</code>	The first operand
<code>operand2</code>	The second operand

Definition at line 462 of file `Numeric.hpp`.

References `MS_throw`.

Referenced by `average()`, `computeKurtosisMiddleTerm()`, `computeSkewLastTerm()`, `covariance()`, `factorial()`, `geometricMean()`, `harmonicMean()`, `product()`, `standardDeviation()`, `sum()`, and `variance()`.

7.91.2.3 bool Numeric::areOverflowUnderflowFlagsSet () [static, private]

Reset the overflow and underflow flags.

Definition at line 549 of file `Numeric.cpp`.

7.91.2.4 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(), and variance().

7.91.2.5 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().

7.91.2.6 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_COMBINATIONS_START, MS_throw, and multiscale::StringManipulator::toString().

7.91.2.7 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().

7.91.2.8 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().

7.91.2.9 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().

**7.91.2.10 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().

7.91.2.11 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().

7.91.2.12 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().

7.91.2.13 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().

7.91.2.14 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().

7.91.2.15 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(), and multiscale::verification::NumericEvaluator::evaluate().

7.91.2.16 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().

7.91.2.17 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().

7.91.2.18 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().

7.91.2.19 double Numeric::geometricMean (const std::vector< double > & numbers, unsigned int nrOfValues) [static, private]

Return the geometric mean of the provided numbers.

$$\text{geometricMean} = e^{\frac{1}{nrOfValues} \sum_{i=1}^{nrOfValues} \log(x_i)}$$

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().

7.91.2.20 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()`, and `multiscale::MinEnclosingTriangleFinder::searchForBTangency()`.

7.91.2.21 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()`.

7.91.2.22 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()`.

7.91.2.23 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 516 of file Numeric.hpp.

Referenced by validateLogBase(), and validateLogNumber().

7.91.2.24 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().

7.91.2.25 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().

7.91.2.26 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::analysis::Region::isValidInputValues\(\)](#), [multiscale::verification::ApproximateProbabilisticModelChecker::doesPropertyHoldConsideringProbabilityComparator\(\)](#), [multiscale::verification::ComparatorEvaluator::evaluate\(\)](#), [multiscale::Geometry2D::isAngleBetweenNonReflex\(\)](#), [multiscale::verification::ApproximateBayesianModelChecker::isModelCheckingResultTrueConsideringComparator\(\)](#), [multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure\(\)](#), [multiscale::verification::BayesianModelChecker::validateBayesFactorThreshold\(\)](#), and [multiscale::verification::ApproximateBayesianModelChecker::validateVarianceThreshold\(\)](#).

7.91.2.27 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\(\)](#).

7.91.2.28 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(), and maximum().

7.91.2.29 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.

7.91.2.30 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().

7.91.2.31 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().

7.91.2.32 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.

7.91.2.33 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(), and minimum().

7.91.2.34 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().

7.91.2.35 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().

7.91.2.36 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().

7.91.2.37 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 527 of file Numeric.hpp.

Referenced by harmonicMean().

7.91.2.38 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().

7.91.2.39 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().

7.91.2.40 void Numeric::printNoValuesWarningMessage (const string & *functionName*) [static, private]

Print the no values warning message for the given function name.

Parameters

<i>function-Name</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().

7.91.2.41 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(), and product().

7.91.2.42 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().

7.91.2.43 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().

7.91.2.44 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().

7.91.2.45 void Numeric::resetOverflowUnderflowFlags() [static, private]

Reset the overflow and underflow flags.

Definition at line 544 of file Numeric.cpp.

7.91.2.46 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().

7.91.2.47 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().

7.91.2.48 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().

7.91.2.49 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(), and multiscale::verification::NumericEvaluator::evaluate().

7.91.2.50 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().

7.91.2.51 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(), and sum().

7.91.2.52 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().

7.91.2.53 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().

7.91.2.54 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().

7.91.2.55 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().

7.91.2.56 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().

7.91.2.57 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().

7.91.2.58 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().

7.91.2.59 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().

7.91.3 Member Data Documentation

7.91.3.1 double Numeric::epsilon = 1E-5 [static, private]

Value of epsilon used to compare two real numbers

Definition at line 91 of file Numeric.hpp.

Referenced by almostEqual().

7.91.3.2 const std::string Numeric::ERR_COMBINATIONS_END = ") when computing combinations." [static, private]

Definition at line 548 of file Numeric.hpp.

Referenced by combinations().

7.91.3.3 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 547 of file Numeric.hpp.

Referenced by combinations().

7.91.3.4 const std::string Numeric::ERR_COMBINATIONS_START = "The provided number of elements n (" [static, private]

Definition at line 546 of file Numeric.hpp.

Referenced by combinations().

7.91.3.5 `const std::string Numeric::ERR_LOG_BASE_END = ") should be a positive real number different from 1. Please change." [static, private]`

Definition at line 540 of file Numeric.hpp.

Referenced by validateLogBase().

7.91.3.6 `const std::string Numeric::ERR_LOG_BASE_START = "The base provided to the log function (" [static, private]`

Definition at line 539 of file Numeric.hpp.

Referenced by validateLogBase().

7.91.3.7 `const std::string Numeric::ERR_LOG_NUMBER_END = ") should be a positive real number. Please change." [static, private]`

Definition at line 542 of file Numeric.hpp.

Referenced by validateLogNumber().

7.91.3.8 `const std::string Numeric::ERR_LOG_NUMBER_START = "The number provided to the log function (" [static, private]`

Definition at line 541 of file Numeric.hpp.

Referenced by validateLogNumber().

7.91.3.9 `const std::string Numeric::ERR_OVERFLOW_UNDERFLOW = "An underflow/overflow exception occurred." [static, private]`

Definition at line 544 of file Numeric.hpp.

7.91.3.10 `const std::string Numeric::ERR_PERCENTILE_VALUE_END = ") should be between 0 and 100. Please change." [static, private]`

Definition at line 551 of file Numeric.hpp.

Referenced by validatePercentile().

7.91.3.11 `const std::string Numeric::ERR_PERCENTILE_VALUE_START = "The provided percentile value (" [static, private]`

Definition at line 550 of file Numeric.hpp.

Referenced by validatePercentile().

7.91.3.12 `const std::string Numeric::ERR_QUARTILE_VALUE_END = ") should be 25, 50 or 75. Please change." [static, private]`

Definition at line 554 of file Numeric.hpp.

Referenced by validateQuartile().

7.91.3.13 `const std::string Numeric::ERR_QUARTILE_VALUE_START = "The provided quartile value (" [static, private]`

Definition at line 553 of file Numeric.hpp.

Referenced by validateQuartile().

7.91.3.14 `const std::string Numeric::WRN_AVERAGE_FUNCTION_NAME = "average" [static, private]`

Definition at line 561 of file Numeric.hpp.

Referenced by average().

7.91.3.15 `const std::string Numeric::WRN_COVARIANCE_FUNCTION_NAME = "covariance" [static, private]`

Definition at line 562 of file Numeric.hpp.

Referenced by covariance().

7.91.3.16 `const std::string Numeric::WRN_GEOMETRIC_MEAN_FUNCTION_NAME = "geometricMean" [static, private]`

Definition at line 563 of file Numeric.hpp.

Referenced by geometricMean().

7.91.3.17 `const std::string Numeric::WRN_HARMONIC_MEAN_FUNCTION_NAME = "harmonicMean" [static, private]`

Definition at line 564 of file Numeric.hpp.

Referenced by harmonicMean().

7.91.3.18 `const std::string Numeric::WRN_KURTOSIS_FUNCTION_NAME = "kurtosis" [static, private]`

Definition at line 565 of file Numeric.hpp.

Referenced by kurtosis().

```
7.91.3.19 const std::string Numeric::WRN_MAXIMUM_FUNCTION_NAME =
    "maximum" [static, private]
```

Definition at line 566 of file Numeric.hpp.

Referenced by maximum().

```
7.91.3.20 const std::string Numeric::WRN_MEDIAN_FUNCTION_NAME = "median"
    [static, private]
```

Definition at line 567 of file Numeric.hpp.

Referenced by median().

```
7.91.3.21 const std::string Numeric::WRN_MINIMUM_FUNCTION_NAME = "minimum"
    [static, private]
```

Definition at line 569 of file Numeric.hpp.

Referenced by minimum().

```
7.91.3.22 const std::string Numeric::WRN_MODE_FUNCTION_NAME = "mode"
    [static, private]
```

Definition at line 568 of file Numeric.hpp.

Referenced by mode().

```
7.91.3.23 const std::string Numeric::WRN_NOT_ENOUGH_VALUES_END = "(...) function. The default value \"0\" was returned." [static, private]
```

Definition at line 559 of file Numeric.hpp.

Referenced by printNoValuesWarningMessage().

```
7.91.3.24 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 558 of file Numeric.hpp.

Referenced by printNoValuesWarningMessage().

```
7.91.3.25 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 556 of file Numeric.hpp.

```
7.91.3.26 const std::string Numeric::WRN_PERCENTILE_FUNCTION_NAME =
    "percentile" [static, private]
```

Definition at line 570 of file Numeric.hpp.

Referenced by percentile().

```
7.91.3.27 const std::string Numeric::WRN_PRODUCT_FUNCTION_NAME = "product"
    [static, private]
```

Definition at line 571 of file Numeric.hpp.

Referenced by product().

```
7.91.3.28 const std::string Numeric::WRN_QUARTILE_FUNCTION_NAME = "quartile"
    [static, private]
```

Definition at line 572 of file Numeric.hpp.

Referenced by quartile().

```
7.91.3.29 const std::string Numeric::WRN_SKEW_FUNCTION_NAME = "skew"
    [static, private]
```

Definition at line 573 of file Numeric.hpp.

Referenced by skew().

```
7.91.3.30 const std::string Numeric::WRN_STANDARD_DEVIATION_
    _FUNCTION_NAME = "standardDeviation" [static,
    private]
```

Definition at line 574 of file Numeric.hpp.

Referenced by standardDeviation().

```
7.91.3.31 const std::string Numeric::WRN_SUM_FUNCTION_NAME = "sum"
    [static, private]
```

Definition at line 575 of file Numeric.hpp.

Referenced by sum().

```
7.91.3.32 const std::string Numeric::WRN_VARIANCE_FUNCTION_NAME = "variance"
    [static, private]
```

Definition at line 576 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](#)

7.92 multiscale::verification::NumericEvaluator Class Reference

Class for evaluating numeric expressions.

```
#include <NumericEvaluator.hpp>
```

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 [UnarySubsetMeasureType](#) &unarySubset, const -
[TimePoint](#) &timePoint)
Evaluate the given unary subset measure expression.
- static double [evaluate](#) (const [BinarySubsetMeasureType](#) &binarySubset, const
std::vector< double > &values)
Evaluate the given binary subset measure expression.
- static double [evaluate](#) (const [TernarySubsetMeasureType](#) &ternarySubset, const
std::vector< double > &values, double parameter)
Evaluate the given ternary subset measure expression.
- static double [evaluate](#) (const [QuaternarySubsetMeasureType](#) &quaternarySubset, const
std::vector< double > &values1, const std::vector< double > &values2)
Evaluate the given quaternary subset measure expression.

7.92.1 Detailed Description

Class for evaluating numeric expressions.

Definition at line 14 of file NumericEvaluator.hpp.

7.92.2 Member Function Documentation

**7.92.2.1 template<typename T > static double multiscale::verification::-
NumericEvaluator::evaluate (const UnaryNumericMeasureType &
unaryNumericMeasure, T value) [inline, static]**

Evaluate the given unary numeric expression.

Parameters

<i>unary-Numeric-Measure</i>	The unary numeric measure type
<i>value</i>	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::NumericVisitor::operator()().

**7.92.2.2 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

<i>binary-Numeric-Measure</i>	The binary numeric measure type
<i>firstValue</i>	The first value for which the binary numeric measure is applied
<i>secondValue</i>	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.

**7.92.2.3 static double multiscale::verification::NumericEvaluator::evaluate (const
UnarySubsetMeasureType & unarySubset, const TimePoint & timePoint)
[inline, static]**

Evaluate the given unary subset measure expression.

Parameters

<i>unarySubset</i>	The unary subset measure type
<i>timePoint</i>	The considered timePoint

Definition at line 99 of file NumericEvaluator.hpp.

References multiscale::verification::TimePoint::avgClusteredness(), multiscale::verification::TimePoint::avgDensity(), multiscale::verification::Clusteredness, multiscale::verification::Count, multiscale::verification::Density, multiscale::ERR_UNDEFINED_ENUM_VALUE, MS_throw, and multiscale::verification::TimePoint::numberOfSpatialEntities().

```
7.92.2.4 static double multiscale::verification::NumericEvaluator::evaluate ( const
    BinarySubsetMeasureType & binarySubset, const std::vector< double > &
    values ) [inline, static]
```

Evaluate the given binary subset measure expression.

Parameters

<i>binary-Subset</i>	The binary subset measure type
<i>values</i>	The considered values

Definition at line 123 of file NumericEvaluator.hpp.

References multiscale::Numeric::average(), multiscale::verification::Avg, 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().

```
7.92.2.5 static double multiscale::verification::NumericEvaluator::evaluate ( const
    TernarySubsetMeasureType & ternarySubset, const std::vector< double > &
    values, double parameter ) [inline, static]
```

Evaluate the given ternary subset measure expression.

Parameters

<i>ternary-Subset</i>	The ternary subset measure type
<i>values</i>	The considered values
<i>parameter</i>	The parameter used by the ternary subset measure

Definition at line 178 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().

7.92.2.6 static double multiscale::verification::NumericEvaluator::evaluate (const QuaternarySubsetMeasureType & *quaternarySubset*, const std::vector< double > & *values1*, const std::vector< double > & *values2*) [inline, static]

Evaluate the given quaternary subset measure expression.

Parameters

<i>quaternary-Subet</i>	The quaternary subset measure type
<i>values1</i>	The first collection of considered values
<i>values2</i>	The second collection of considered values

Definition at line 201 of file NumericEvaluator.hpp.

References multiscale::verification::Covar, multiscale::Numeric::covariance(), multiscale::ERR_UNDEFINED_ENUM_VALUE, and MS_throw.

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

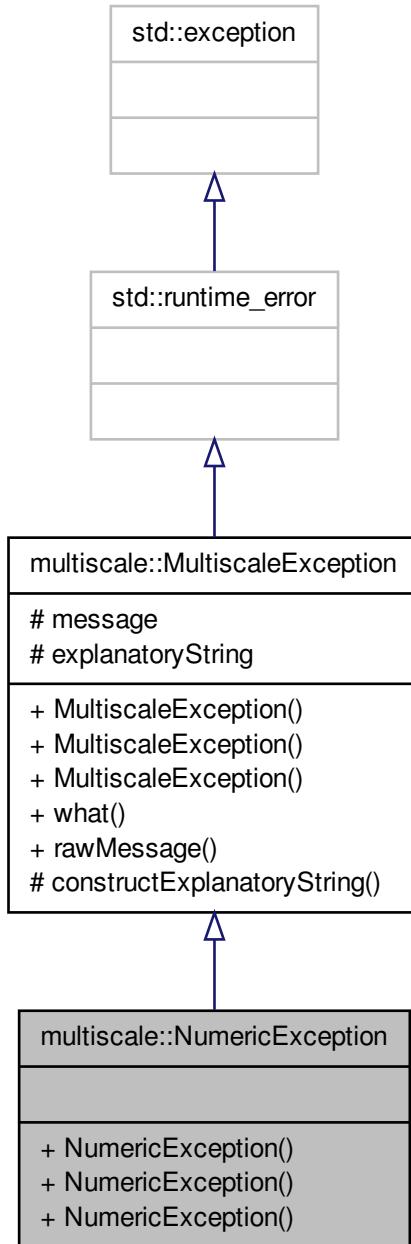
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericEvaluator.hpp](#)

7.93 multiscale::NumericException Class Reference

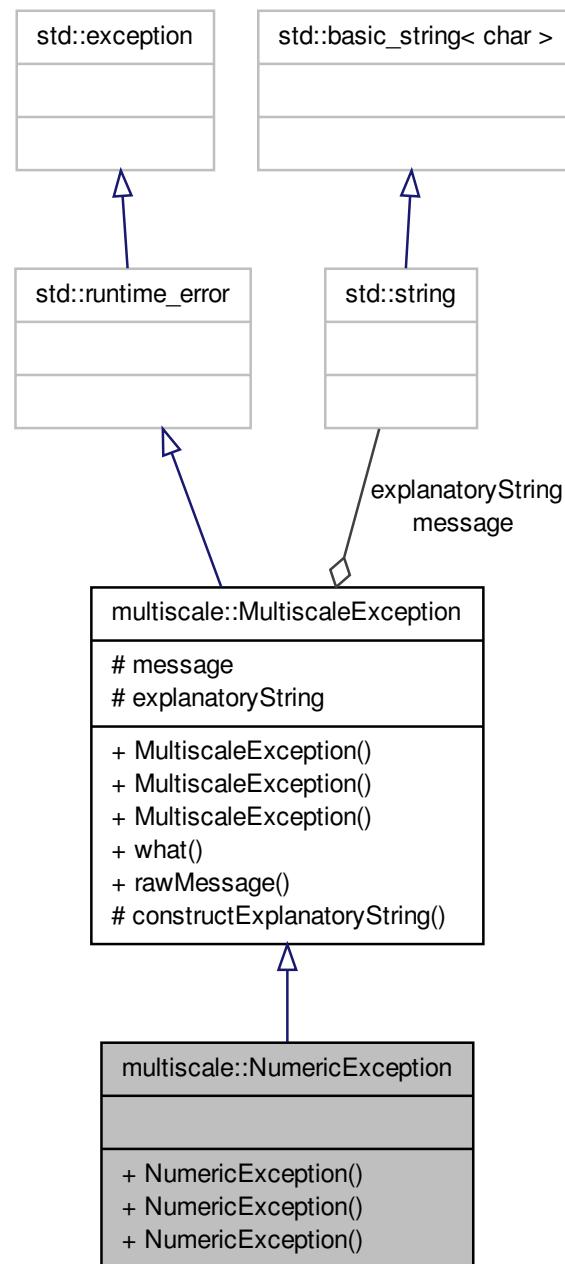
Class for representing algorithm exceptions.

```
#include <NumericException.hpp>
```

Inheritance diagram for multiscale::NumericException:



Collaboration diagram for multiscale::NumericException:



Public Member Functions

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

7.93.1 Detailed Description

Class for representing algorithm exceptions.

Definition at line 14 of file NumericException.hpp.

7.93.2 Constructor & Destructor Documentation

7.93.2.1 multiscale::NumericException::NumericException() [inline]

Definition at line 18 of file NumericException.hpp.

7.93.2.2 multiscale::NumericException::NumericException(const string & file, int line, const string & msg) [inline, explicit]

Definition at line 20 of file NumericException.hpp.

7.93.2.3 multiscale::NumericException::NumericException(const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 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](#)

7.94 multiscale::verification::NumericMeasureAttribute Class - Reference

Class for representing a numeric measure attribute.

```
#include <NumericMeasureAttribute.hpp>
```

Public Attributes

- [NumericMeasureAttributeType numericMeasure](#)

7.94.1 Detailed Description

Class for representing a numeric measure attribute.

Definition at line 34 of file NumericMeasureAttribute.hpp.

7.94.2 Member Data Documentation

7.94.2.1 **NumericMeasureAttributeType multiscale::verification::NumericMeasureAttribute::numericMeasure**

The numeric measure

Definition at line 38 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](#)

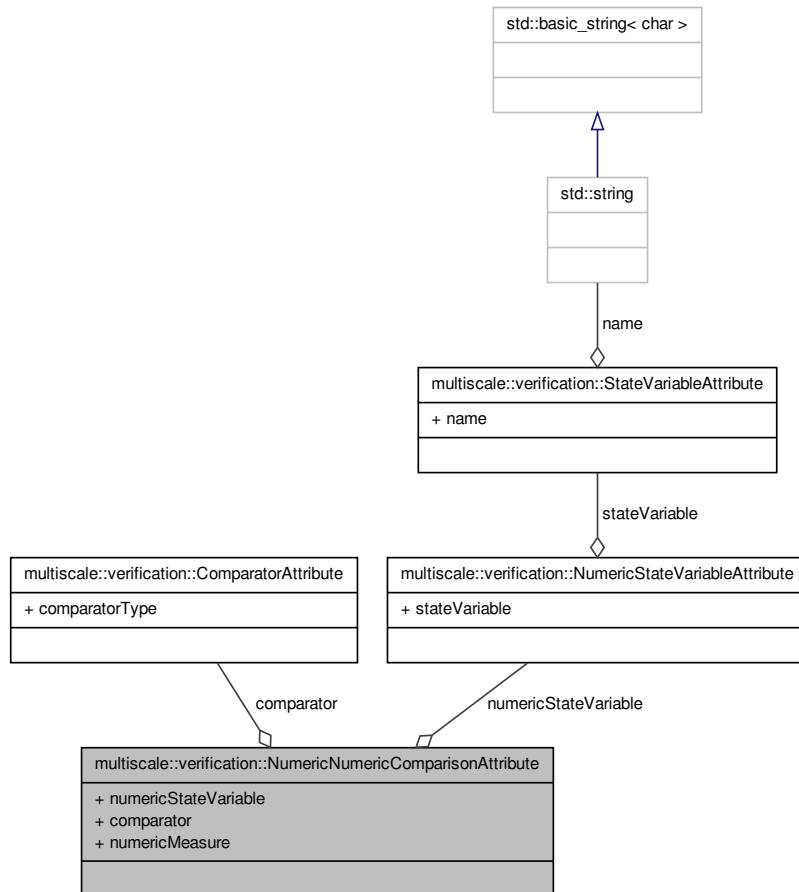
7.95 **multiscale::verification::NumericNumericComparisonAttribute Class Reference**

Class for representing a numeric numeric comparison attribute.

```
#include <NumericNumericComparisonAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericNumericComparison-

Attribute:



Public Attributes

- [NumericStateVariableAttribute numericStateVariable](#)
- [ComparatorAttribute comparator](#)
- [NumericMeasureAttributeType numericMeasure](#)

7.95.1 Detailed Description

Class for representing a numeric numeric comparison attribute.

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

7.95.2 Member Data Documentation

7.95.2.1 ComparatorAttribute multiscale::verification::NumericNumericComparisonAttribute::comparator

The comparator

Definition at line 21 of file NumericNumericComparisonAttribute.hpp.

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

7.95.2.2 NumericMeasureAttributeType multiscale::verification::NumericNumericComparisonAttribute::numericMeasure

The numeric measure following the comparator

Definition at line 22 of file NumericNumericComparisonAttribute.hpp.

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

7.95.2.3 NumericStateVariableAttribute multiscale::verification::NumericNumericComparisonAttribute::numericStateVariable

The numeric state variable preceding the comparator

Definition at line 20 of file NumericNumericComparisonAttribute.hpp.

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

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/[NumericNumericComparisonAttribute.hpp](#)

7.96 multiscale::NumericRangeManipulator Class Reference

Operations for ranges of numeric values.

```
#include <NumericRangeManipulator.hpp>
```

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.

7.96.1 Detailed Description

Operations for ranges of numeric values.

Definition at line 7 of file NumericRangeManipulator.hpp.

7.96.2 Member Function Documentation

```
7.96.2.1 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>oldRange- Min</i>	The minimum of the old range
<i>oldRange- Max</i>	The maximum of the old range
<i>newRange- Min</i>	The minimum of the new range
<i>newRange- Max</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](#)

7.97 multiscale::verification::NumericSpatialAttribute Class - Reference

Class for representing a numeric spatial attribute.

```
#include <NumericSpatialAttribute.hpp>
```

Public Attributes

- [NumericSpatialAttributeType numericSpatialMeasure](#)

7.97.1 Detailed Description

Class for representing a numeric spatial attribute.

Definition at line 33 of file NumericSpatialAttribute.hpp.

7.97.2 Member Data Documentation

7.97.2.1 NumericSpatialAttributeType multiscale::verification::NumericSpatialAttribute::numericSpatialMeasure

The numeric spatial measure

Definition at line 37 of file NumericSpatialAttribute.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/[NumericSpatialAttribute.hpp](#)

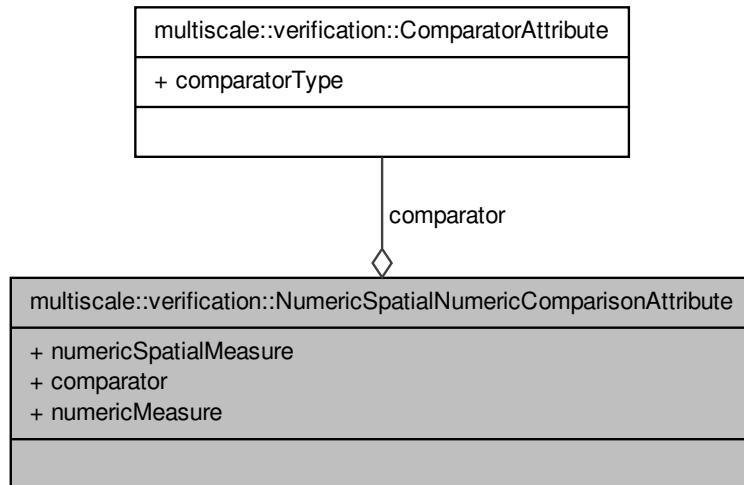
7.98 multiscale::verification::NumericSpatialNumericComparison-Attribute Class Reference

Class for representing a numeric spatial numeric comparison attribute.

```
#include <NumericSpatialNumericComparisonAttribute.hpp>
```

Collaboration diagram for multiscale::verification::NumericSpatialNumericComparison-

Attribute:



Public Attributes

- [NumericSpatialAttributeType numericSpatialMeasure](#)
- [ComparatorAttribute comparator](#)
- [NumericMeasureAttributeType numericMeasure](#)

7.98.1 Detailed Description

Class for representing a numeric spatial numeric comparison attribute.

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

7.98.2 Member Data Documentation

7.98.2.1 ComparatorAttribute multiscale::verification::NumericSpatialNumericComparisonAttribute::comparator

The comparator

Definition at line 21 of file `NumericSpatialNumericComparisonAttribute.hpp`.

Referenced by `multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatialNumericComparison()`.

7.99 multiscale::verification::NumericStateVariableAttribute Class Reference 595

7.98.2.2 NumericMeasureAttributeType multiscale::verification::NumericSpatial-NumericComparisonAttribute::numericMeasure

The numeric measure following the comparator

Definition at line 22 of file NumericSpatialNumericComparisonAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatial-NumericComparison().

7.98.2.3 NumericSpatialAttributeType multiscale::verification::NumericSpatial-NumericComparisonAttribute::numericSpatialMeasure

The numeric spatial measure preceding the comparator

Definition at line 20 of file NumericSpatialNumericComparisonAttribute.hpp.

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatial-NumericComparison().

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

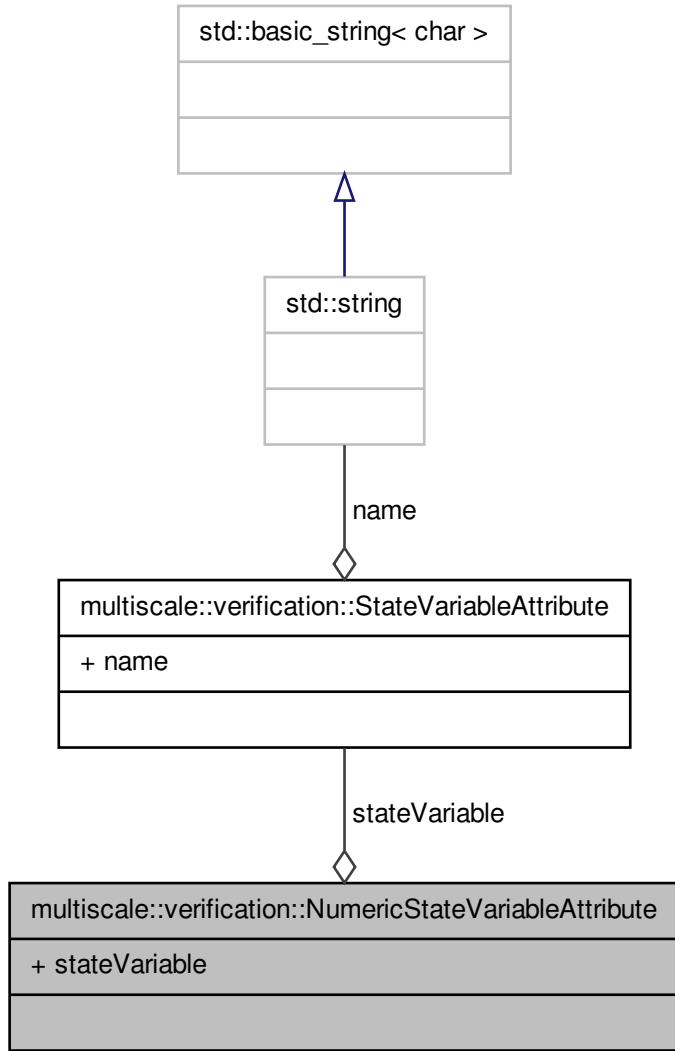
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[Numeric-SpatialNumericComparisonAttribute.hpp](#)

7.99 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`

7.99.1 Detailed Description

Class for representing a numeric state variable attribute.

Definition at line 14 of file NumericStateVariableAttribute.hpp.

7.99.2 Member Data Documentation

7.99.2.1 StateVariableAttribute multiscale::verification::NumericStateVariable-Attribute::stateVariable

The state variable

Definition at line 18 of file NumericStateVariableAttribute.hpp.

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

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

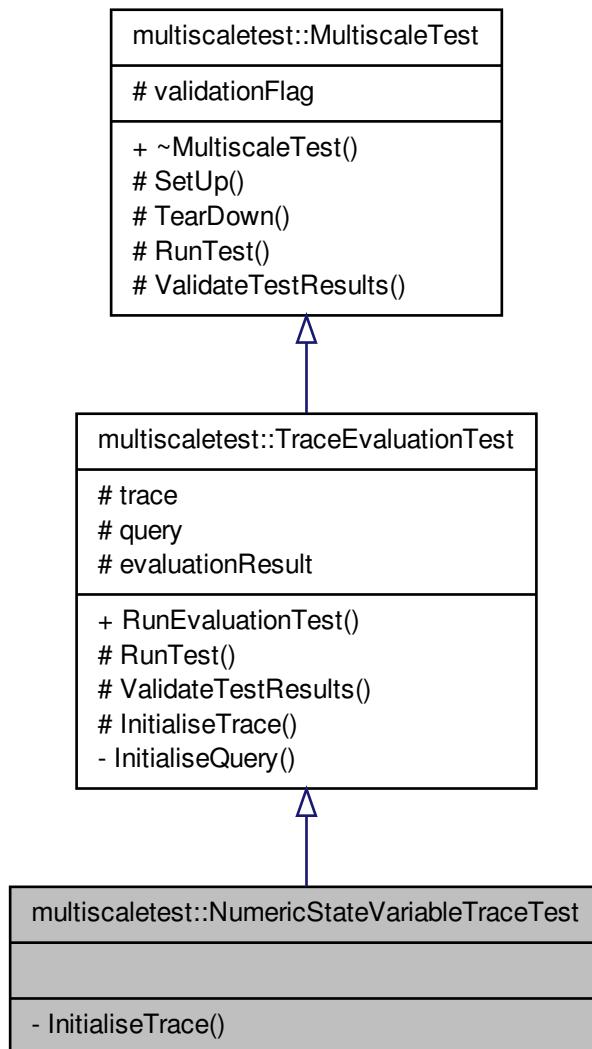
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[NumericState-VariableAttribute.hpp](#)

7.100 multiscaletest::NumericStateVariableTraceTest Class Reference

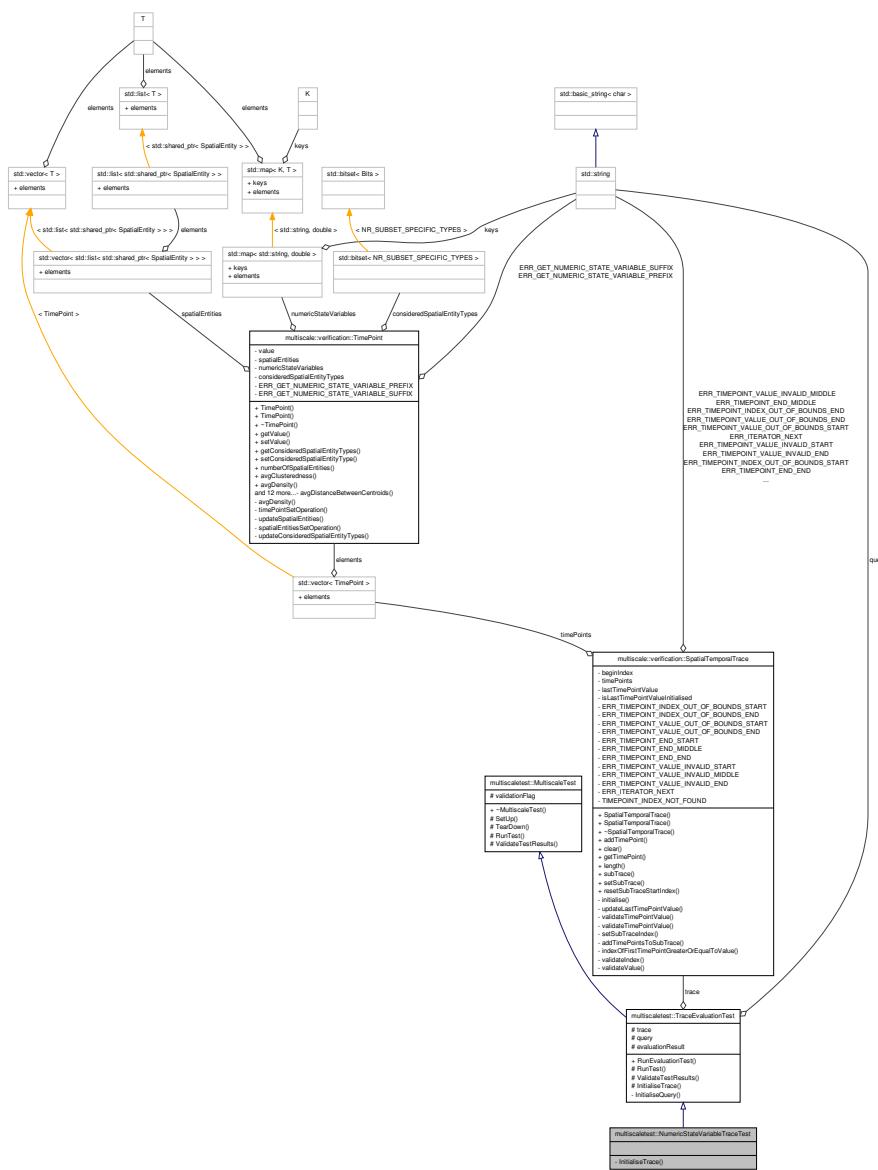
Class for testing evaluation of numeric state variable-only traces.

```
#include <NumericStateVariableTraceTest.hpp>
```

Inheritance diagram for multiscaletest::NumericStateVariableTraceTest:



Collaboration diagram for multiscaletest::NumericStateVariableTraceTest:



Private Member Functions

- virtual void `InitialiseTrace ()` override

Initialise the trace.

7.100.1 Detailed Description

Class for testing evaluation of numeric state variable-only traces.

Definition at line 13 of file NumericStateVariableTraceTest.hpp.

7.100.2 Member Function Documentation

7.100.2.1 void multiscaletest::NumericStateVariableTraceTest::InitialiseTrace() [override, private, virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 22 of file NumericStateVariableTraceTest.hpp.

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

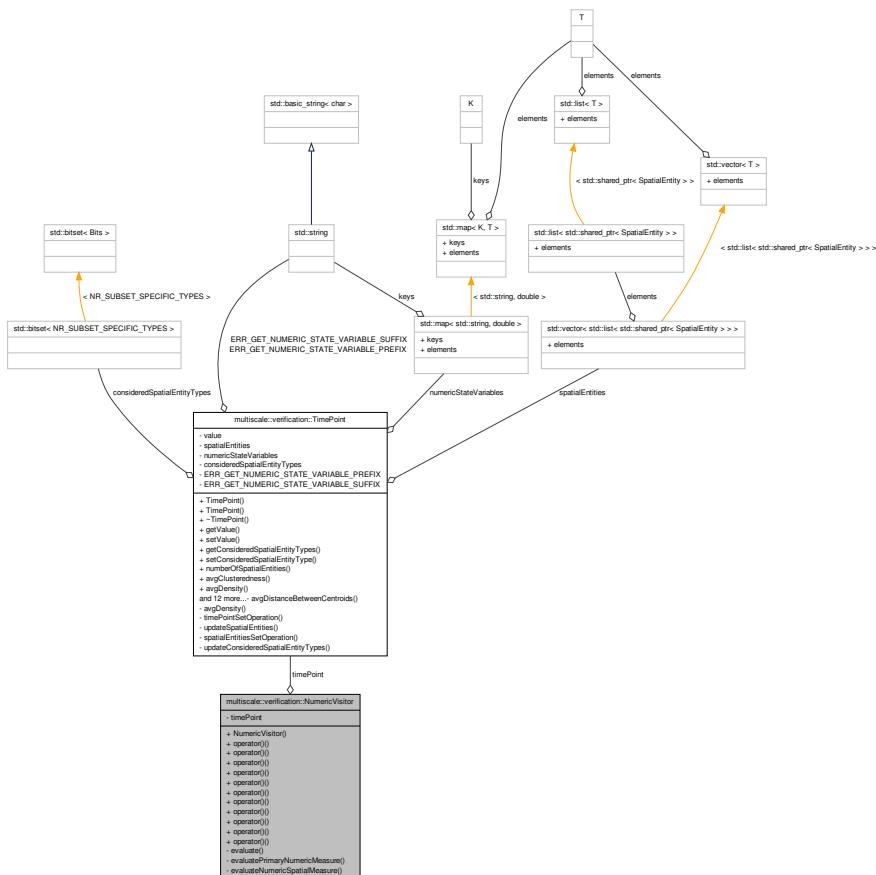
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[NumericStateVariableTraceTest.hpp](#)

7.101 multiscale::verification::NumericVisitor Class Reference

Class for evaluating numeric measures.

```
#include <NumericVisitor.hpp>
```

Collaboration diagram for multiscale::verification::NumericVisitor:



Public Member Functions

- `NumericVisitor` (const `TimePoint` &`timePoint`)
 - 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 `NumericSpatialAttribute` &`numericSpatialMeasure`) const
 - Overloading the "()" operator for the `NumericSpatialAttribute` alternative.*

- Overloading the "(" operator for the `NumericSpatialAttribute` alternative.
- double `operator()` (const `UnaryNumericNumericAttribute` &unaryNumeric-NumericMeasure) const
 - Overloading the "(" operator for the `UnaryNumericNumericAttribute` alternative.
- double `operator()` (const `BinaryNumericNumericAttribute` &binaryNumeric-NumericMeasure) const
 - Overloading the "(" operator for the `BinaryNumericNumericAttribute` alternative.
- double `operator()` (const `UnarySubsetAttribute` &unarySubset) const
 - Overloading the "(" operator for the `UnarySubsetAttribute` alternative.
- double `operator()` (const `BinarySubsetAttribute` &binarySubset) const
 - Overloading the "(" operator for the `BinarySubsetAttribute` alternative.
- double `operator()` (const `TernarySubsetAttribute` &ternarySubset) const
 - Overloading the "(" operator for the `TernarySubsetAttribute` alternative.
- double `operator()` (const `QuaternarySubsetAttribute` &quaternarySubset) const
 - Overloading the "(" operator for the `QuaternarySubsetAttribute` alternative.

Private Member Functions

- double `evaluate` (const `NumericMeasureAttributeType` &numericMeasure) const
 - Evaluate the given numeric measure considering the timePoint field.*
- double `evaluatePrimaryNumericMeasure` (const `PrimaryNumericMeasure-AttributeType` &primaryNumericMeasure) const
 - Evaluate the given primary numeric measure considering the timePoint field.*
- double `evaluateNumericSpatialMeasure` (const `NumericSpatialAttributeType` &numericSpatialMeasure) const
 - Evaluate the given numeric spatial measure considering the timePoint field.*

Private Attributes

- const `TimePoint` & `timePoint`

7.101.1 Detailed Description

Class for evaluating numeric measures.

Definition at line 18 of file NumericVisitor.hpp.

7.101.2 Constructor & Destructor Documentation

7.101.2.1 multiscale::verification::NumericVisitor::NumericVisitor (const `TimePoint` & `timePoint`) [inline]

Definition at line 26 of file NumericVisitor.hpp.

Referenced by evaluate(), evaluateNumericSpatialMeasure(), and evaluatePrimaryNumericMeasure().

7.101.3 Member Function Documentation

7.101.3.1 double multiscale::verification::NumericVisitor::evaluate (const NumericMeasureAttributeType & *numericMeasure*) const [inline, private]

Evaluate the given numeric measure considering the timePoint field.

Parameters

<i>numeric- Measure</i>	The given numeric measure
-----------------------------	---------------------------

Definition at line 129 of file NumericVisitor.hpp.

References NumericVisitor(), and timePoint.

Referenced by operator()().

7.101.3.2 double multiscale::verification::NumericVisitor::evaluateNumericSpatialMeasure (const NumericSpatialAttributeType & *numericSpatialMeasure*) const [inline, private]

Evaluate the given numeric spatial measure considering the timePoint field.

Parameters

<i>numeric- Spatial- Measure</i>	The given numeric spatial measure
--	-----------------------------------

Definition at line 145 of file NumericVisitor.hpp.

References NumericVisitor(), and timePoint.

Referenced by operator()().

7.101.3.3 double multiscale::verification::NumericVisitor::evaluatePrimaryNumericMeasure (const PrimaryNumericMeasureAttributeType & *primaryNumericMeasure*) const [inline, private]

Evaluate the given primary numeric measure considering the timePoint field.

Parameters

<i>primary- Numeric- Measure</i>	The given primary numeric measure
--	-----------------------------------

Definition at line 137 of file NumericVisitor.hpp.

References NumericVisitor(), and timePoint.

Referenced by operator()().

```
7.101.3.4 double multiscale::verification::NumericVisitor::operator() ( const
    NumericMeasureAttribute & numericMeasure ) const [inline]
```

Overloading the "(") operator for the [NumericMeasureAttribute](#) alternative.

Parameters

<i>numeric- Measure</i>	The numeric measure
-----------------------------	---------------------

Definition at line 32 of file NumericVisitor.hpp.

References evaluate(), and multiscale::verification::NumericMeasureAttribute::numericMeasure.

```
7.101.3.5 double multiscale::verification::NumericVisitor::operator() ( const
    PrimaryNumericMeasureAttribute & primaryNumericMeasure ) const
    [inline]
```

Overloading the "(") operator for the [PrimaryNumericMeasureAttribute](#) alternative.

Parameters

<i>primary- Numeric- Measure</i>	The primary numeric measure
--	-----------------------------

Definition at line 40 of file NumericVisitor.hpp.

References evaluatePrimaryNumericMeasure(), and multiscale::verification::PrimaryNumericMeasureAttribute::primaryNumericMeasure.

```
7.101.3.6 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 48 of file NumericVisitor.hpp.

7.101.3.7 double multiscale::verification::NumericVisitor::operator() (const NumericStateVariableAttribute & *numericStateVariable*) const [inline]

Overloading the "(") operator for the [NumericStateVariableAttribute](#) alternative.

Parameters

<i>numeric- State- Variable</i>	The numeric state variable
---	----------------------------

Definition at line 56 of file NumericVisitor.hpp.

References multiscale::verification::TimePoint::getNumericStateVariable(), multiscale::verification::StateVariableAttribute::name, multiscale::verification::NumericStateVariableAttribute::stateVariable, and timePoint.

7.101.3.8 double multiscale::verification::NumericVisitor::operator() (const NumericSpatialAttribute & *numericSpatialMeasure*) const [inline]

Overloading the "(") operator for the [NumericSpatialAttribute](#) alternative.

Parameters

<i>numeric- Spatial- Measure</i>	The numeric spatial measure attribute
--	---------------------------------------

Definition at line 66 of file NumericVisitor.hpp.

References evaluateNumericSpatialMeasure(), and multiscale::verification::NumericSpatialAttribute::numericSpatialMeasure.

7.101.3.9 double multiscale::verification::NumericVisitor::operator() (const UnaryNumericNumericAttribute & *unaryNumericNumericMeasure*) const [inline]

Overloading the "(") operator for the [UnaryNumericNumericAttribute](#) alternative.

Parameters

<i>unary- Numeric- Numeric- Measure</i>	The unary numeric numeric measure
---	-----------------------------------

Definition at line 74 of file NumericVisitor.hpp.

References evaluate(), and multiscale::verification::UnaryNumericNumericAttribute::numericMeasure.

7.101.3.10 double multiscale::verification::NumericVisitor::operator() (const
BinaryNumericNumericAttribute & binaryNumericNumericMeasure) const
[inline]

Overloading the "(" operator for the [BinaryNumericNumericAttribute](#) alternative.

Parameters

<i>binary-Numeric-Numeric-Measure</i>	The binary numeric numeric measure
---------------------------------------	------------------------------------

Definition at line 88 of file NumericVisitor.hpp.

References [evaluate\(\)](#), [multiscale::verification::BinaryNumericNumericAttribute::firstNumericMeasure](#), and [multiscale::verification::BinaryNumericNumericAttribute::secondNumericMeasure](#).

7.101.3.11 double multiscale::verification::NumericVisitor::operator() (const
UnarySubsetAttribute & unarySubset) const [inline]

Overloading the "(" operator for the [UnarySubsetAttribute](#) alternative.

Parameters

<i>unarySubset</i>	The unary subset
--------------------	------------------

Definition at line 163 of file NumericVisitor.hpp.

References [evaluate\(\)](#), [multiscale::verification::UnarySubsetAttribute::subset](#), [timePoint](#), [multiscale::verification::UnarySubsetAttribute::unarySubsetMeasure](#), and [multiscale::verification::UnarySubsetMeasureAttribute::unarySubsetMeasureType](#).

7.101.3.12 double multiscale::verification::NumericVisitor::operator() (const
BinarySubsetAttribute & binarySubset) const [inline]

Overloading the "(" operator for the [BinarySubsetAttribute](#) alternative.

Parameters

<i>binary-Subset</i>	The binary subset
----------------------	-------------------

Definition at line 169 of file NumericVisitor.hpp.

References [multiscale::verification::BinarySubsetAttribute::binarySubsetMeasure](#), [multiscale::verification::BinarySubsetMeasureAttribute::binarySubsetMeasureType](#), [multiscale::verification::NumericEvaluator::evaluate\(\)](#), [multiscale::verification::TimePointEvaluator::getSpatialMeasureValues\(\)](#), and [multiscale::verification::BinarySubset-](#)

Attribute::spatialMeasure, multiscale::verification::SpatialMeasureAttribute::spatialMeasureType, and multiscale::verification::BinarySubsetAttribute::subset.

7.101.3.13 double multiscale::verification::NumericVisitor::operator() (const TernarySubsetAttribute & *ternarySubset*) const [inline]

Overloading the "(") operator for the [TernarySubsetAttribute](#) alternative.

Parameters

<i>ternary- Subset</i>	The ternary subset
----------------------------	--------------------

Definition at line 178 of file NumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::TimePointEvaluator::getSpatialMeasureValues(), multiscale::verification::TernarySubsetAttribute::parameter, multiscale::verification::TernarySubsetAttribute::spatialMeasure, multiscale::verification::SpatialMeasureAttribute::spatialMeasureType, multiscale::verification::TernarySubsetAttribute::subset, multiscale::verification::TernarySubsetAttribute::ternarySubsetMeasure, and multiscale::verification::TernarySubsetMeasureAttribute::ternarySubsetMeasureType.

7.101.3.14 double multiscale::verification::NumericVisitor::operator() (const QuaternarySubsetAttribute & *quaternarySubset*) const [inline]

Overloading the "(") operator for the [QuaternarySubsetAttribute](#) alternative.

Parameters

<i>quaternary- Subset</i>	The quaternary subset
-------------------------------	-----------------------

Definition at line 188 of file NumericVisitor.hpp.

References multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::QuaternarySubsetAttribute::firstSpatialMeasure, multiscale::verification::QuaternarySubsetAttribute::firstSubset, multiscale::verification::TimePointEvaluator::getSpatialMeasureValues(), multiscale::verification::QuaternarySubsetAttribute::quaternarySubsetMeasure, multiscale::verification::QuaternarySubsetMeasureAttribute::quaternarySubsetMeasureType, multiscale::verification::QuaternarySubsetAttribute::secondSubset, and multiscale::verification::SpatialMeasureAttribute::spatialMeasureType.

7.101.4 Member Data Documentation

7.101.4.1 const TimePoint& multiscale::verification::NumericVisitor::timePoint [private]

The considered timepoint

Definition at line 22 of file NumericVisitor.hpp.

Referenced by evaluate(), evaluateNumericSpatialMeasure(), evaluatePrimaryNumericMeasure(), and operator()().

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

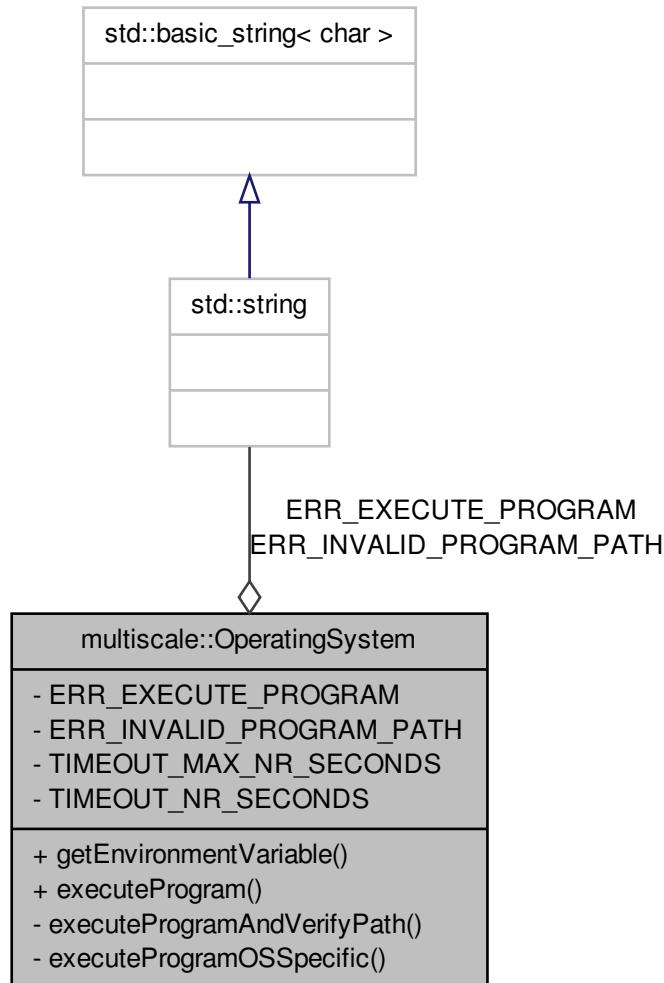
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[NumericVisitor.hpp](#)

7.102 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

7.102.1 Detailed Description

Class for executing operating system related functions.

Definition at line 23 of file OperatingSystem.hpp.

7.102.2 Member Function Documentation

7.102.2.1 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::printErrorMessage\(\)](#), [multiscale::ConsolePrinter::printWarningMessage\(\)](#), and [multiscale::MultiscaleException::rawMessage\(\)](#).

Referenced by [multiscale::verification::ModelCheckingManager::executeExtraEvaluationProgramAndPrintMessage\(\)](#), and [main\(\)](#).

7.102.2.2 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().

7.102.2.3 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().

7.102.2.4 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().

7.102.3 Member Data Documentation

7.102.3.1 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.

7.102.3.2 `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()`.

7.102.3.3 `const unsigned int OperatingSystem::TIMEOUT_MAX_NR_SECONDS = 100`
[static, private]

Definition at line 117 of file OperatingSystem.hpp.

7.102.3.4 `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/[Operating-
System.cpp](#)

7.103 multiscale::verification::OrConstraintAttribute Class - Reference

Class for representing an "or" constraint attribute.

```
#include <OrConstraintAttribute.hpp>
```

Public Attributes

- [ConstraintAttributeType constraint](#)

7.103.1 Detailed Description

Class for representing an "or" constraint attribute.

Definition at line 14 of file OrConstraintAttribute.hpp.

7.103.2 Member Data Documentation

7.103.2.1 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](#)

7.104 multiscale::verification::OrLogicPropertyAttribute Class Reference

Class for representing an "or" logic property attribute.

```
#include <OrLogicPropertyAttribute.hpp>
```

Public Attributes

- [LogicPropertyAttributeType logicProperty](#)

7.104.1 Detailed Description

Class for representing an "or" logic property attribute.

Definition at line 14 of file OrLogicPropertyAttribute.hpp.

7.104.2 Member Data Documentation

7.104.2.1 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:

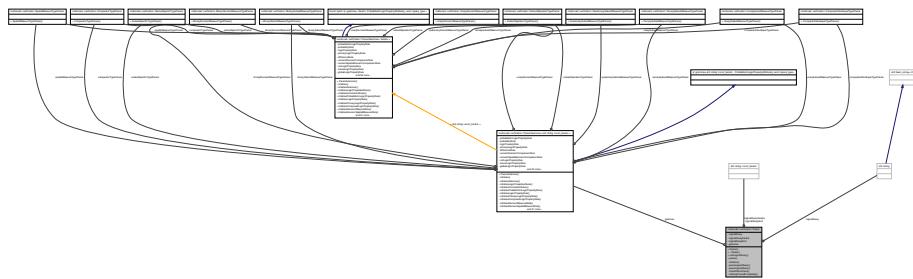
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[OrLogicPropertyAttribute.hpp](#)

7.105 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](#)
- [ParserGrammar](#) < std::string::const_iterator > [grammar](#)

7.105.1 Detailed Description

Class used for parsing (P)BLSTL logical queries.

Definition at line 17 of file Parser.hpp.

7.105.2 Constructor & Destructor Documentation

7.105.2.1 Parser::Parser (const std::string & *logicalQuery*)

Definition at line 12 of file Parser.cpp.

7.105.2.2 Parser::~Parser ()

Definition at line 18 of file Parser.cpp.

7.105.3 Member Function Documentation

7.105.3.1 void Parser::checkIfErrorCase (bool *isSuccessfulParse*) [private]

Check if an error case was encountered.

Parameters

<i>is-Successful-Parse</i>	The parse was successful or not
----------------------------	---------------------------------

Definition at line 69 of file Parser.cpp.

7.105.3.2 void Parser::initialise () [private]

Initialisation function.

Definition at line 46 of file Parser.cpp.

7.105.3.3 bool Parser::isStringParsedCompletely () [private]

Check if the string was parsed completely.

Definition at line 79 of file Parser.cpp.

7.105.3.4 bool Parser::parse (AbstractSyntaxTree & parseResult)

Parse the logical query.

Parameters

<i>parseResult</i>	The result of the parsing procedure
--------------------	-------------------------------------

Definition at line 26 of file Parser.cpp.

References multiscale::verification::ParserGrammarExtraInputException::getErrorMessage(), multiscale::verification::ParserGrammarUnparseableInputException::getErrorMessage(), multiscale::verification::ParserGrammarUnexpectedTokenException::getErrorMessage(), multiscale::verification::ParserGrammarProbabilityException::getErrorMessage(), multiscale::verification::ParserGrammarProbabilityException::getExpectedToken(), and multiscale::verification::ParserGrammarUnexpectedTokenException::getExpectedToken().

Referenced by multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree(), multiscale::verification::ModelCheckingManager::isValidLogicProperty(), main(), multiscaletest::verification::parseInputString(), printParsingResult(), and multiscaletest::TraceEvaluationTest::RunTest().

7.105.3.5 bool Parser::parseLogicalQuery (AbstractSyntaxTree & parseResult)

[private]

Parse the logical query and wrap the ProbabilisticLogicProperty into an [AbstractSyntaxTree](#) instance.

Parameters

<i>parseResult</i>	The result of the parsing procedure
--------------------	-------------------------------------

Definition at line 51 of file Parser.cpp.

References multiscale::verification::AbstractSyntaxTree::initialiseTree().

7.105.3.6 bool Parser::parseLogicalQuery (ProbabilisticLogicPropertyAttribute & parseResult) [private]

Parse the logical query and construct the abstract syntax tree.

Parameters

<i>parseResult</i>	The result of the parsing procedure
--------------------	-------------------------------------

Definition at line 61 of file Parser.cpp.

7.105.3.7 void Parser::setLogicalQuery (const std::string & *logicalQuery*)

Set the value of the logical query.

Definition at line 20 of file Parser.cpp.

Referenced by multiscale::verification::ModelCheckingManager::isValidLogicProperty(), and printQueries().

7.105.4 Member Data Documentation**7.105.4.1 ParserGrammar<std::string::const_iterator> multiscale::verification::Parser::grammar [private]**

The grammar used for parsing logical queries

Definition at line 27 of file Parser.hpp.

7.105.4.2 std::string multiscale::verification::Parser::logicalQuery [private]

The logical query to be parsed

Definition at line 21 of file Parser.hpp.

7.105.4.3 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.

7.105.4.4 std::string::const_iterator multiscale::verification::Parser::logicalQueryIterator [private]

Iterator of the logical query

Definition at line 23 of file Parser.hpp.

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

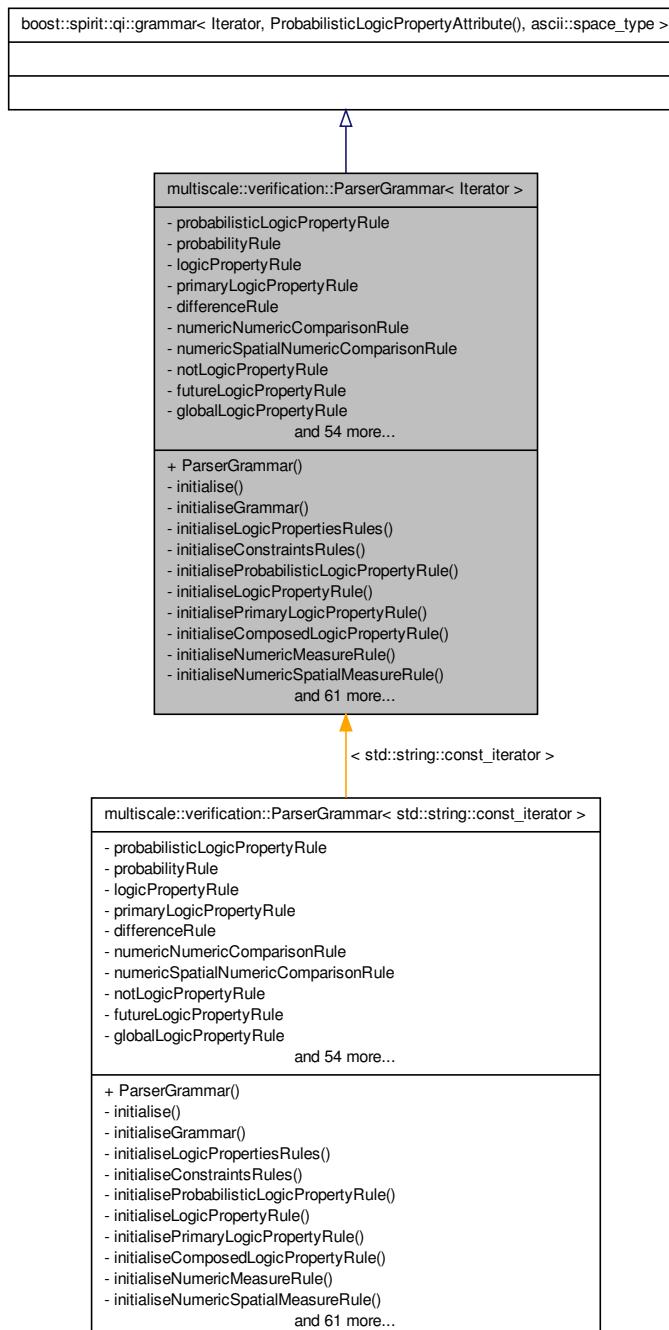
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[Parser.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/parsing/[Parser.cpp](#)

7.106 multiscale::verification::ParserGrammar< Iterator > Class - Template Reference

The grammar for parsing (P)BLSTL spatial-temporal logical queries.

```
#include <ParserGrammar.hpp>
```

Inheritance diagram for multiscale::verification::ParserGrammar< Iterator >:



Collaboration diagram for multiscale::verification::ParserGrammar< Iterator >:



Public Member Functions

- `ParserGrammar ()`

Private Member Functions

- `void initialise ()`
Initialisation function.
- `void initialiseGrammar ()`
Initialise the grammar.
- `void initialiseLogicPropertiesRules ()`
Initialise the logic properties rules.
- `void initialiseConstraintsRules ()`
Initialise the constraints 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 initialiseNumericMeasureRule ()`
Initialise the numeric measure rule.
- `void initialiseNumericSpatialMeasureRule ()`
Initialise the numeric spatial measure rule.
- `void initialiseNumericSpatialSubsetMeasureRule ()`
Initialise the numeric spatial subset measure rule.
- `void initialiseNaryNumericMeasureRule ()`
Initialise the n-ary numeric measure rule.
- `void initialiseSubsetRule ()`
Initialise the subset rule.
- `void initialiseConstraintRule ()`
Initialise the constraint rule.
- `void initialisePrimaryConstraintRule ()`

- void **initialiseFilterNumericMeasureRule ()**
Initialise the filter numeric measure rule.
 - void **initialiseComposedConstraintRule ()**
Initialise the composed constraint rule.
 - void **initialiseSpatialMeasureRule ()**
Initialise the spatial measure rule.
 - void **initialiseComparatorRules ()**
Initialise the comparator rules.
 - void **initialiseNumericStateVariableRule ()**
Initialise the numeric state variable rule.
 - void **initialiseDebugSupport ()**
Initialise debug support.
 - void **assignNamesToRules ()**
Assign names to the rules.
 - void **assignNamesToLogicPropertiesRules ()**
Assign names to logic properties rules.
 - void **assignNamesToConstraintsRules ()**
Assign names to constraints 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 **assignNamesToNumericMeasureRules ()**
Assign names to the numeric measure rules.
 - void **assignNamesToNumericSpatialMeasureRules ()**
Assign names to the numeric spatial measure rules.
 - void **assignNamesToNumericSpatialSubsetMeasureRules ()**
Assign names to the numeric spatial subset measure rules.
 - void **assignNamesToNaryNumericMeasureRules ()**
Assign names to the n-ary numeric measure rules.
 - void **assignNamesToSubsetRules ()**
Assign names to the subset 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 **assignNamesToComparatorRules ()**
Assign names to the comparator rules.
- void **assignNamesToNumericStateVariableRules ()**
Assign names to the numeric state variable rules.
- void **initialiseRulesDebugging ()**
Initialise the debugging of rules.
- void **initialiseLogicPropertiesRulesDebugging ()**
Initialise the debugging of the logic properties rules.
- void **initialiseConstraintsRulesDebugging ()**
Initialise the debugging of the constraints 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 **initialiseNumericMeasureRuleDebugging ()**
Initialise debugging for the numeric measure rule.
- void **initialiseNumericSpatialMeasureRuleDebugging ()**
Initialise debugging for the numeric spatial measure rule.
- void **initialiseSpatialSubsetMeasureRuleDebugging ()**
Initialise debugging for the spatial subset measure rule.
- void **initialiseNaryNumericMeasureRuleDebugging ()**
Initialise debugging for the n-ary numeric measure rule.
- void **initialiseSubsetRuleDebugging ()**
Initialise debugging for the subset 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 **initialiseSpatialMeasureRuleDebugging ()**
Initialise debugging for the spatial measure rule.
- void **initialiseComparatorRuleDebugging ()**

- *Initialise debugging for the comparator rule.*
 - void [initialiseNumericStateVariableRuleDebugging \(\)](#)
Initialise debugging for the state variable rule.
 - void [initialiseErrorHandlingSupport \(\)](#)
Initialise the error handling routines.
 - void [initialiseLogicPropertiesErrorHandlingSupport \(\)](#)
Initialise the logic properties error handling support.
 - void [initialiseConstraintsErrorHandlingSupport \(\)](#)
Initialise the constraints 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.
 - void [initialiseNumericMeasureErrorHandlingSupport \(\)](#)
Initialise the numeric measure error handling support.
 - void [initialiseNumericSpatialMeasureErrorHandlingSupport \(\)](#)
Initialise the numeric spatial measure error handling support.
 - void [initialiseSubsetErrorHandlingSupport \(\)](#)
Initialise the subset 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.
 - void [initialiseStateVariableErrorHandlingSupport \(\)](#)
Initialise the state variable error handling support.

Private Attributes

- qi::rule< Iterator, [ProbabilisticLogicPropertyAttribute\(\)](#), ascii::space_type > [probabilisticLogicPropertyRule](#)
- qi::rule< Iterator, double(), ascii::space_type > [probabilityRule](#)
- qi::rule< Iterator, [LogicPropertyAttribute\(\)](#), ascii::space_type > [logicPropertyRule](#)
- qi::rule< Iterator, [PrimaryLogicPropertyAttribute\(\)](#), ascii::space_type > [primaryLogicPropertyRule](#)
- qi::rule< Iterator, [DifferenceAttribute\(\)](#), ascii::space_type > [differenceRule](#)
- qi::rule< Iterator, [NumericNumericComparisonAttribute\(\)](#), ascii::space_type > [numericNumericComparisonRule](#)
- qi::rule< Iterator, [NumericSpatialNumericComparisonAttribute\(\)](#), ascii::space_type > [numericSpatialNumericComparisonRule](#)

- `qi::rule< Iterator, NotLogicPropertyAttribute(), ascii::space_type > notLogicPropertyRule`
- `qi::rule< Iterator, FutureLogicPropertyAttribute(), ascii::space_type > futureLogicPropertyRule`
- `qi::rule< Iterator, GlobalLogicPropertyAttribute(), ascii::space_type > globalLogicPropertyRule`
- `qi::rule< Iterator, NextLogicPropertyAttribute(), ascii::space_type > nextLogicPropertyRule`
- `qi::rule< Iterator, NextKLogicPropertyAttribute(), ascii::space_type > nextKLogicPropertyRule`
- `qi::rule< Iterator, AndLogicPropertyAttribute(), ascii::space_type > andLogicPropertyRule`
- `qi::rule< Iterator, OrLogicPropertyAttribute(), ascii::space_type > orLogicPropertyRule`
- `qi::rule< Iterator, ImplicationLogicPropertyAttribute(), ascii::space_type > implicationLogicPropertyRule`
- `qi::rule< Iterator, EquivalenceLogicPropertyAttribute(), ascii::space_type > equivalenceLogicPropertyRule`
- `qi::rule< Iterator, UntilLogicPropertyAttribute(), ascii::space_type > untilLogicPropertyRule`
- `qi::rule< Iterator, NumericMeasureAttribute(), ascii::space_type > numericMeasureRule`
- `qi::rule< Iterator, PrimaryNumericMeasureAttribute(), ascii::space_type > primaryNumericMeasureRule`
- `qi::rule< Iterator, UnaryNumericNumericAttribute(), ascii::space_type > unaryNumericNumericRule`
- `qi::rule< Iterator, BinaryNumericNumericAttribute(), ascii::space_type > binaryNumericNumericRule`
- `qi::rule< Iterator, NumericSpatialAttribute(), ascii::space_type > numericSpatialRule`
- `qi::rule< Iterator, UnarySubsetAttribute(), ascii::space_type > unarySubsetRule`
- `qi::rule< Iterator, BinarySubsetAttribute(), ascii::space_type > binarySubsetRule`
- `qi::rule< Iterator, TernarySubsetAttribute(), ascii::space_type > ternarySubsetRule`
- `qi::rule< Iterator, QuaternarySubsetAttribute(), ascii::space_type > quaternarySubsetRule`
- `qi::rule< Iterator, UnarySubsetMeasureAttribute(), ascii::space_type > unarySubsetMeasureRule`
- `qi::rule< Iterator, BinarySubsetMeasureAttribute(), ascii::space_type > binarySubsetMeasureRule`
- `qi::rule< Iterator, TernarySubsetMeasureAttribute(), ascii::space_type > ternarySubsetMeasureRule`
- `qi::rule< Iterator, QuaternarySubsetMeasureAttribute(), ascii::space_type > quaternarySubsetMeasureRule`
- `qi::rule< Iterator, UnaryNumericMeasureAttribute(), ascii::space_type > unaryNumericMeasureRule`

- `qi::rule< Iterator, BinaryNumericMeasureAttribute(), ascii::space_type > binaryNumericMeasureRule`
- `qi::rule< Iterator, SubsetAttribute(), ascii::space_type > subsetRule`
- `qi::rule< Iterator, SubsetSpecificAttribute(), ascii::space_type > subsetSpecificRule`
- `qi::rule< Iterator, FilterSubsetAttribute(), ascii::space_type > filterSubsetRule`
- `qi::rule< Iterator, SubsetSubsetOperationAttribute(), ascii::space_type > subsetSubsetOperationRule`
- `qi::rule< Iterator, ConstraintAttribute(), ascii::space_type > constraintRule`
- `qi::rule< Iterator, PrimaryConstraintAttribute(), ascii::space_type > primaryConstraintRule`
- `qi::rule< Iterator, NotConstraintAttribute(), ascii::space_type > notConstraintRule`
- `qi::rule< Iterator, UnarySpatialConstraintAttribute(), ascii::space_type > unarySpatialConstraintRule`
- `qi::rule< Iterator, UnaryTypeConstraintAttribute(), ascii::space_type > unaryTypeConstraintRule`
- `qi::rule< Iterator, FilterNumericMeasureAttribute(), ascii::space_type > filterNumericMeasureRule`
- `qi::rule< Iterator, UnaryNumericFilterAttribute(), ascii::space_type > unaryNumericFilterRule`
- `qi::rule< Iterator, BinaryNumericFilterAttribute(), ascii::space_type > binaryNumericFilterRule`
- `qi::rule< Iterator, AndConstraintAttribute(), ascii::space_type > andConstraintRule`
- `qi::rule< Iterator, OrConstraintAttribute(), ascii::space_type > orConstraintRule`
- `qi::rule< Iterator, ImplicationConstraintAttribute(), ascii::space_type > implicationConstraintRule`
- `qi::rule< Iterator, EquivalenceConstraintAttribute(), ascii::space_type > equivalenceConstraintRule`
- `qi::rule< Iterator, SpatialMeasureAttribute(), ascii::space_type > spatialMeasureRule`
- `qi::rule< Iterator, ComparatorAttribute(), ascii::space_type > comparatorRule`
- `qi::rule< Iterator, ComparatorAttribute(), ascii::space_type > probabilisticLogicPropertyComparatorRule`
- `qi::rule< Iterator, NumericStateVariableAttribute(), ascii::space_type > numericStateVariableRule`
- `qi::rule< Iterator, StateVariableAttribute(), ascii::space_type > stateVariableRule`
- `qi::rule< Iterator, std::string(), ascii::space_type > stateVariableNameRule`
- `UnarySubsetMeasureTypeParser unarySubsetMeasureTypeParser`
- `BinarySubsetMeasureTypeParser binarySubsetMeasureTypeParser`
- `TernarySubsetMeasureTypeParser ternarySubsetMeasureTypeParser`
- `QuaternarySubsetMeasureTypeParser quaternarySubsetMeasureTypeParser`
- `UnaryNumericMeasureTypeParser unaryNumericMeasureTypeParser`
- `BinaryNumericMeasureTypeParser binaryNumericMeasureTypeParser`
- `SubsetSpecificTypeParser subsetSpecificTypeParser`

- [SubsetOperationTypeParser subsetOperationTypeParser](#)
- [SpatialMeasureTypeParser spatialMeasureTypeParser](#)
- [ComparatorTypeParser comparatorTypeParser](#)
- [ComparatorNonEqualTypeParser comparatorNonEqualTypeParser](#)

7.106.1 Detailed Description

```
template<typename Iterator>class multiscale::verification::ParserGrammar< Iterator >
```

The grammar for parsing (P)BLSTL spatial-temporal logical queries.

Definition at line 37 of file ParserGrammar.hpp.

7.106.2 Constructor & Destructor Documentation

```
7.106.2.1 template<typename Iterator> multiscale::verification::ParserGrammar<  
Iterator >::ParserGrammar( ) [inline]
```

Definition at line 132 of file ParserGrammar.hpp.

7.106.3 Member Function Documentation

```
7.106.3.1 template<typename Iterator> void multiscale::verification::ParserGrammar<  
Iterator >::assignNamesToComparatorRules( ) [inline,  
private]
```

Assign names to the comparator rules.

Definition at line 694 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
::assignNamesToRules().

```
7.106.3.2 template<typename Iterator> void multiscale::verification::ParserGrammar<  
Iterator >::assignNamesToComposedConstraintRules( ) [inline,  
private]
```

Assign names to the composed constraint rules.

Definition at line 681 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
::assignNamesToConstraintsRules().

7.106.3.3 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToComposedLogicPropertyRules() [inline, private]

Assign names to the composed logic property rules.

Definition at line 613 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToLogicPropertiesRules().

7.106.3.4 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToConstraintRules() [inline, private]

Assign names to the constraint rules.

Definition at line 661 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToConstraintsRules().

7.106.3.5 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToConstraintsRules() [inline, private]

Assign names to constraints rules.

Definition at line 581 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

7.106.3.6 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToFilterNumericMeasureRules() [inline, private]

Assign names to the filter numeric measure rules.

Definition at line 674 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToConstraintsRules().

7.106.3.7 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToLogicPropertiesRules() [inline, private]

Assign names to logic properties rules.

Definition at line 573 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

7.106.3.8 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToLogicPropertyRules() [inline, private]**

Assign names to the logic property rules.

Definition at line 595 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToLogicPropertiesRules().

7.106.3.9 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToNaryNumericMeasureRules() [inline, private]**

Assign names to the n-ary numeric measure rules.

Definition at line 647 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

7.106.3.10 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToNumericMeasureRules() [inline, private]**

Assign names to the numeric measure rules.

Definition at line 622 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

7.106.3.11 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::assignNamesToNumericSpatialMeasureRules() [inline, private]**

Assign names to the numeric spatial measure rules.

Definition at line 630 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

```
7.106.3.12 template<typename Iterator> void multiscale-
    ::verification::ParserGrammar< Iterator
    >::assignNamesToNumericSpatialSubsetMeasureRules( )
    [inline, private]
```

Assign names to the numeric spatial subset measure rules.

Definition at line 639 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
 ::assignNamesToRules().

```
7.106.3.13 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::assignNamesToNumericStateVariableRules( )
    [inline, private]
```

Assign names to the numeric state variable rules.

Definition at line 700 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
 ::assignNamesToRules().

```
7.106.3.14 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::assignNamesToPrimaryConstraintRules( )
    [inline, private]
```

Assign names to the primary constraint rules.

Definition at line 666 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
 ::assignNamesToConstraintsRules().

```
7.106.3.15 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::assignNamesToPrimaryLogicPropertyRules( )
    [inline, private]
```

Assign names to the primary logic property rules.

Definition at line 600 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
 ::assignNamesToLogicPropertiesRules().

```
7.106.3.16 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::assignNamesToProbabilisticLogicPropertyRules
    ( ) [inline, private]
```

Assign names to the probabilistic logic property rules.

Definition at line 589 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToLogicPropertiesRules().

```
7.106.3.17 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::assignNamesToRules( ) [inline,
private]
```

Assign names to the rules.

Definition at line 559 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseDebugSupport().

```
7.106.3.18 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::assignNamesToSpatialMeasureRules( )
[inline, private]
```

Assign names to the spatial measure rules.

Definition at line 689 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

```
7.106.3.19 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::assignNamesToSubsetRules( ) [inline,
private]
```

Assign names to the subset rules.

Definition at line 653 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToRules().

```
7.106.3.20 template<typename Iterator> void multiscale::verification-
::ParserGrammar< Iterator >::initialise( ) [inline,
private]
```

Initialisation function.

Definition at line 139 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::ParserGrammar().

7.106.3.21 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseComparatorRuleDebugging() [inline, private]

Initialise debugging for the comparator rule.

Definition at line 842 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.22 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseComparatorRules() [inline, private]

Initialise the comparator rules.

Definition at line 530 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.23 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseComposedConstraintErrorHandlingSupport() [inline, private]

Initialise the composed constraint error handling support.

Definition at line 942 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsErrorHandlingSupport().

7.106.3.24 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseComposedConstraintRule() [inline, private]

Initialise the composed constraint rule.

Definition at line 509 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRules().

7.106.3.25 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseComposedConstraintRuleDebugging() [inline, private]

Initialise debugging for the composed constraint rule.

Definition at line 829 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRulesDebugging().

```
7.106.3.26 template<typename Iterator> void multiscale::verification-
    ::ParserGrammar< Iterator >::initialiseComposed-
    LogicPropertyErrorHandlingSupport( ) [inline,
    private]
```

Initialise the compose logic property error handling support.

Definition at line 898 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesErrorHandlingSupport().

```
7.106.3.27 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::initialiseComposedLogicPropertyRule( )
    [inline, private]
```

Initialise the composed logic property rule.

Definition at line 281 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRules().

```
7.106.3.28 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::initialiseComposedLogicPropertyRuleDebugging
    ( ) [inline, private]
```

Initialise debugging for the composed logic property rule.

Definition at line 761 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRulesDebugging().

```
7.106.3.29 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::initialiseConstraintRule( ) [inline,
    private]
```

Initialise the constraint rule.

Definition at line 448 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRules().

7.106.3.30 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseConstraintRuleDebugging() [inline, private]

Initialise debugging for the constraint rule.

Definition at line 809 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRulesDebugging().

7.106.3.31 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseConstraintsErrorHandlingSupport() [inline, private]

Initialise the constraints error handling support.

Definition at line 872 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseErrorHandlingSupport().

7.106.3.32 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseConstraintsRules() [inline, private]

Initialise the constraints rules.

Definition at line 168 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.33 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseConstraintsRulesDebugging() [inline, private]

Initialise the debugging of the constraints rules.

Definition at line 729 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.34 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseDebugSupport() [inline, private]

Initialise debug support.

Definition at line 551 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise().

7.106.3.35 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseErrorHandlingSupport() [inline, private]**

Initialise the error handling routines.

Definition at line 855 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise().

7.106.3.36 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseFilterNumericMeasureErrorHandlingSupport() [inline, private]**

Initialise the filter numeric measure error handling support.

Definition at line 935 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsErrorHandlingSupport().

7.106.3.37 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseFilterNumericMeasureRule() [inline, private]**

Initialise the filter numeric measure rule.

Definition at line 482 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRules().

7.106.3.38 **template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseFilterNumericMeasureRuleDebugging() [inline, private]**

Initialise debugging for the filter numeric measure rules.

Definition at line 822 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRulesDebugging().

7.106.3.39 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseGrammar() [inline, private]

Initialise the grammar.

Definition at line 146 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise().

7.106.3.40 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseLogicPropertiesErrorHandlingSupport() [inline, private]

Initialise the logic properties error handling support.

Definition at line 865 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseErrorHandlingSupport().

7.106.3.41 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseLogicPropertiesRules() [inline, private]

Initialise the logic properties rules.

Definition at line 160 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.42 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseLogicPropertiesRulesDebugging() [inline, private]

Initialise the debugging of the logic properties rules.

Definition at line 721 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.43 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseLogicPropertyRule() [inline, private]

Initialise the logic property rule.

Definition at line 191 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseLogicPropertiesRules().

7.106.3.44 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseLogicPropertyRuleDebugging()
[inline, private]

Initialise debugging for the logic property rule.

Definition at line 743 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseLogicPropertiesRulesDebugging().

7.106.3.45 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNaryNumericMeasureRule()
[inline, private]

Initialise the n-ary numeric measure rule.

Definition at line 408 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseGrammar().

7.106.3.46 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNaryNumericMeasureRuleDebugging()
[inline, private]

Initialise debugging for the n-ary numeric measure rule.

Definition at line 795 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseRulesDebugging().

7.106.3.47 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericMeasureErrorHandlingSupport()
[inline, private]

Initialise the numeric measure error handling support.

Definition at line 907 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseErrorHandlingSupport().

7.106.3.48 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericMeasureRule() [inline, private]

Initialise the numeric measure rule.

Definition at line 309 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.49 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericMeasureRuleDebugging() [inline, private]

Initialise debugging for the numeric measure rule.

Definition at line 770 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.50 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericSpatialMeasureErrorHandlingSupport() [inline, private]

Initialise the numeric spatial measure error handling support.

Definition at line 913 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseErrorHandlingSupport().

7.106.3.51 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericSpatialMeasureRule() [inline, private]

Initialise the numeric spatial measure rule.

Definition at line 340 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.52 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseNumericSpatialMeasureRuleDebugging() [inline, private]

Initialise debugging for the numeric spatial measure rule.

Definition at line 778 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

```
7.106.3.53 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialiseNumericSpatialSubsetMeasureRule( )
[inline, private]
```

Initialise the numeric spatial subset measure rule.

Definition at line 393 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

```
7.106.3.54 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialiseNumericStateVariableRule( )
[inline, private]
```

Initialise the numeric state variable rule.

Definition at line 539 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

```
7.106.3.55 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialiseNumericStateVariableRuleDebugging( )
[inline, private]
```

Initialise debugging for the state variable rule.

Definition at line 848 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

```
7.106.3.56 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialisePrimaryConstraintErrorHandlingSupport
( ) [inline, private]
```

Initialise the primary constraint error handling support.

Definition at line 927 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsErrorHandlingSupport().

7.106.3.57 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialisePrimaryConstraintRule() [inline, private]

Initialise the primary constraint rule.

Definition at line 460 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRules().

7.106.3.58 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialisePrimaryConstraintRuleDebugging() [inline, private]

Initialise debugging for the primary constraint rules.

Definition at line 814 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintsRulesDebugging().

7.106.3.59 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport() [inline, private]

Initialise the primary logic property error handling support.

Definition at line 885 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesErrorHandlingSupport().

7.106.3.60 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialisePrimaryLogicPropertyRule() [inline, private]

Initialise the primary logic property rule.

Definition at line 204 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRules().

7.106.3.61 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialisePrimaryLogicPropertyRuleDebugging() [inline, private]

Initialise debugging for the primary logic property rule.

Definition at line 748 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRulesDebugging().

```
7.106.3.62 template<typename Iterator> void multiscale::verification-
    ::ParserGrammar< Iterator >::initialiseProbabilistic-
    LogicPropertyErrorHandlingSupport( ) [inline,
    private]
```

Initialise the probabilistic logic property error handling support.

Definition at line 879 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesErrorHandlingSupport().

```
7.106.3.63 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::initialiseProbabilisticLogicPropertyRule( )
    [inline, private]
```

Initialise the probabilistic logic property rule.

Definition at line 176 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRules().

```
7.106.3.64 template<typename Iterator> void multiscale-
    ::verification::ParserGrammar< Iterator
    >::initialiseProbabilisticLogicPropertyRuleDebugging( ) [inline,
    private]
```

Initialise debugging for the probabilistic logic property rule.

Definition at line 737 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertiesRulesDebugging().

```
7.106.3.65 template<typename Iterator> void multiscale::verification::Parser-
    Grammar< Iterator >::initialiseRulesDebugging( ) [inline,
    private]
```

Initialise the debugging of rules.

Definition at line 707 of file ParserGrammar.hpp.

7.106.3.66 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseSpatialMeasureRule() [inline, private]

Initialise the spatial measure rule.

Definition at line 524 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

7.106.3.67 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseSpatialMeasureRuleDebugging() [inline, private]

Initialise debugging for the spatial measure rule.

Definition at line 837 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.68 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseSpatialSubsetMeasureRuleDebugging() [inline, private]

Initialise debugging for the spatial subset measure rule.

Definition at line 787 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.3.69 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseStateVariableErrorHandlingSupport() [inline, private]

Initialise the state variable error handling support.

Definition at line 950 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseErrorHandlingSupport().

7.106.3.70 template<typename Iterator> void multiscale::verification::ParserGrammar< Iterator >::initialiseSubsetErrorHandlingSupport() [inline, private]

Initialise the subset error handling support.

Definition at line 921 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseErrorHandlingSupport().

```
7.106.3.71 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialiseSubsetRule( ) [inline,
private]
```

Initialise the subset rule.

Definition at line 417 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseGrammar().

```
7.106.3.72 template<typename Iterator> void multiscale::verification::Parser-
Grammar< Iterator >::initialiseSubsetRuleDebugging( ) [inline,
private]
```

Initialise debugging for the subset rules.

Definition at line 801 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseRulesDebugging().

7.106.4 Member Data Documentation

```
7.106.4.1 template<typename Iterator> qi::rule<Iterator, AndConstraintAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::andConstraintRule [private]
```

The rule for parsing an "and" constraint

Definition at line 98 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRule().

```
7.106.4.2 template<typename Iterator> qi::rule<Iterator, AndLogicPropertyAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::andLogicPropertyRule [private]
```

The rule for parsing an "and" logic property

Definition at line 57 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule().

7.106.4.3 template<typename Iterator> qi::rule<Iterator, BinaryNumericFilterAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::binaryNumericFilterRule [private]

The rule for parsing a binary numeric filter measure

Definition at line 96 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToFilterNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRuleDebugging().

7.106.4.4 template<typename Iterator> qi::rule<Iterator, BinaryNumericMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::binaryNumericMeasureRule [private]

The rule for parsing a binary numeric measure

Definition at line 80 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNaryNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNaryNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNaryNumericMeasureRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule().

7.106.4.5 template<typename Iterator> BinaryNumericMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::binaryNumericMeasureTypeParser [private]

The binary numeric measure type parser

Definition at line 120 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseNaryNumericMeasureRule().

7.106.4.6 template<typename Iterator> qi::rule<Iterator, BinaryNumericNumericAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::binaryNumericNumericRule [private]

The rule for parsing a binary numeric numeric attribute

Definition at line 66 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRuleDebugging().

7.106.4.7 template<typename Iterator> qi::rule<Iterator, BinarySubsetMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::binarySubsetMeasureRule [private]

The rule for parsing a binary subset measure

Definition at line 75 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialSubsetMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSpatialSubsetMeasureRuleDebugging().

7.106.4.8 template<typename Iterator> BinarySubsetMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::binarySubsetMeasureTypeParser [private]

The binary subset measure type parser

Definition at line 115 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >->::initialiseNumericSpatialSubsetMeasureRule().

7.106.4.9 template<typename Iterator> qi::rule<Iterator, BinarySubsetAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::binarySubsetRule [private]

The rule for parsing a binary subset

Definition at line 70 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureErrorHandlerHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRuleDebugging().

**7.106.4.10 template<typename Iterator> ComparatorNonEqualTypeParser
multiscale::verification::ParserGrammar< Iterator
>::comparatorNonEqualTypeParser [private]**

The comparator type parser which does not accept the "=" symbol

Definition at line 128 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRules().

**7.106.4.11 template<typename Iterator> qi::rule<Iterator, ComparatorAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::comparatorRule [private]**

The rule for parsing a comparator

Definition at line 105 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComparatorRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRuleDebugging(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule().

**7.106.4.12 template<typename Iterator> ComparatorTypeParser
multiscale::verification::ParserGrammar< Iterator
>::comparatorTypeParser [private]**

The comparator type parser

Definition at line 127 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRules().

```
7.106.4.13 template<typename Iterator> qi::rule<Iterator, ConstraintAttribute(),  
ascii::space_type> multiscale::verification::ParserGrammar< Iterator  
>::constraintRule [private]
```

The rule for parsing a constraint

Definition at line 87 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRuleDebugging(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule().

```
7.106.4.14 template<typename Iterator> qi::rule<Iterator, DifferenceAttribute(),  
ascii::space_type> multiscale::verification::ParserGrammar< Iterator  
>::differenceRule [private]
```

The rule for parsing a difference attribute

Definition at line 48 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

```
7.106.4.15 template<typename Iterator> qi::rule<Iterator, EquivalenceConstraint-  
Attribute(), ascii::space_type> multiscale::verification::ParserGrammar<  
Iterator >::equivalenceConstraintRule [private]
```

The rule for parsing an "equivalence" constraint

Definition at line 101 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRule().

7.106.4.16 template<typename Iterator> qi::rule<Iterator, EquivalenceLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::equivalenceLogicPropertyRule [private]

The rule for parsing an "equivalence" logic property

Definition at line 60 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule().

7.106.4.17 template<typename Iterator> qi::rule<Iterator, FilterNumericMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::filterNumericMeasureRule [private]

The rule for parsing a filter numeric measure

Definition at line 94 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToFilterNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule().

7.106.4.18 template<typename Iterator> qi::rule<Iterator, FilterSubsetAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::filterSubsetRule [private]

The rule for parsing a subset filter

Definition at line 84 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToSubsetRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRuleDebugging().

7.106.4.19 template<typename Iterator> qi::rule<Iterator, FutureLogicProperty-Attribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::futureLogicPropertyRule [private]

The rule for parsing a "future" logic property

Definition at line 52 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.20 template<typename Iterator> qi::rule<Iterator, GlobalLogicProperty-Attribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::globalLogicPropertyRule [private]

The rule for parsing a "global" logic property

Definition at line 53 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.21 template<typename Iterator> qi::rule<Iterator, ImplicationConstraint-Attribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::implicationConstraintRule [private]

The rule for parsing an "implication" constraint

Definition at line 100 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule().

7.106.4.22 template<typename Iterator> qi::rule<Iterator, ImplicationLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::implicationLogicPropertyRule [private]

The rule for parsing an "implication" logic property

Definition at line 59 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule().

7.106.4.23 template<typename Iterator> qi::rule<Iterator, LogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::logicPropertyRule [private]

The rule for parsing a logic property

Definition at line 45 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRuleDebugging(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRule().

7.106.4.24 template<typename Iterator> qi::rule<Iterator, NextKLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::nextKLogicPropertyRule [private]

The rule for parsing a "next K" logic property

Definition at line 55 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.25 template<typename Iterator> qi::rule<Iterator, NextLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::nextLogicPropertyRule [private]

The rule for parsing a "next" logic property

Definition at line 54 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.26 template<typename Iterator> qi::rule<Iterator, NotConstraintAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::notConstraintRule [private]

The rule for parsing a "not" constraint

Definition at line 90 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRuleDebugging().

7.106.4.27 template<typename Iterator> qi::rule<Iterator, NotLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::notLogicPropertyRule [private]

The rule for parsing a "not" logic property

Definition at line 51 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.28 template<typename Iterator> qi::rule<Iterator, NumericMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::numericMeasureRule [private]

The rule for parsing a numeric measure

Definition at line 63 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule().

7.106.4.29 template<typename Iterator> qi::rule<Iterator, NumericNumericComparisonAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::numericNumericComparisonRule [private]

The rule for parsing a numeric numeric comparison

Definition at line 49 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.30 template<typename Iterator> qi::rule<Iterator, NumericSpatialNumericComparisonAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::numericSpatialNumericComparisonRule [private]

The rule for parsing a numeric spatial numeric comparison

Definition at line 50 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.31 template<typename Iterator> qi::rule<Iterator, NumericSpatialAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::numericSpatialRule [private]

The rule for parsing a numeric spatial measure

Definition at line 68 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(), multiscale-

::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule().

7.106.4.32 template<typename Iterator> qi::rule<Iterator, NumericStateVariableAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::numericStateVariableRule [private]

The rule for parsing a numeric state variable

Definition at line 108 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericStateVariableRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericStateVariableRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericStateVariableRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule().

7.106.4.33 template<typename Iterator> qi::rule<Iterator, OrConstraintAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::orConstraintRule [private]

The rule for parsing an "or" constraint

Definition at line 99 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedConstraintRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRule().

7.106.4.34 template<typename Iterator> qi::rule<Iterator, OrLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::orLogicPropertyRule [private]

The rule for parsing an "or" logic property

Definition at line 58 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >

>::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule().

7.106.4.35 template<typename Iterator> qi::rule<Iterator, PrimaryConstraintAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::primaryConstraintRule [private]

The rule for parsing a primary constraint

Definition at line 89 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseConstraintRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRuleDebugging().

7.106.4.36 template<typename Iterator> qi::rule<Iterator, PrimaryLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::primaryLogicPropertyRule [private]

The rule for parsing a primary logic property

Definition at line 47 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryLogicPropertyRuleDebugging().

7.106.4.37 template<typename Iterator> qi::rule<Iterator, PrimaryNumericMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::primaryNumericMeasureRule [private]

The rule for parsing a primary numeric numeric attribute

Definition at line 64 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(),

and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise-NumericMeasureRuleDebugging().

7.106.4.38 template<typename Iterator> qi::rule<Iterator, ComparatorAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::probabilisticLogicPropertyComparatorRule [private]

The rule for parsing a comparator for a probabilistic logic property

Definition at line 106 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComparatorRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRuleDebugging(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComparatorRules(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRule().

7.106.4.39 template<typename Iterator> qi::rule<Iterator, ProbabilisticLogicProperty-Attribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::probabilisticLogicPropertyRule [private]

The rule for parsing a probabilistic logic property

Definition at line 43 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToProbabilisticLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRuleDebugging().

7.106.4.40 template<typename Iterator> qi::rule<Iterator, double(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::probabilityRule [private]

The rule for parsing a probability

Definition at line 44 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToProbabilisticLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyErrorHandlerSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseProbabilisticLogicPropertyRuleDebugging().

7.106.4.41 template<typename Iterator> qi::rule<Iterator, QuaternarySubsetMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::quaternarySubsetMeasureRule [private]

The rule for parsing a quaternary subset measure

Definition at line 77 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialSubsetMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSpatialSubsetMeasureRuleDebugging().

7.106.4.42 template<typename Iterator> QuaternarySubsetMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::quaternarySubsetMeasureTypeParser [private]

The quaternary subset measure type parser

Definition at line 117 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule().

7.106.4.43 template<typename Iterator> qi::rule<Iterator, QuaternarySubsetAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::quaternarySubsetRule [private]

The rule for parsing a quaternary subset

Definition at line 72 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureErrorHandlerHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRuleDebugging().

7.106.4.44 template<typename Iterator> qi::rule<Iterator, SpatialMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::spatialMeasureRule [private]

The rule for parsing a spatial measure

Definition at line 103 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator

```
>::assignNamesToSpatialMeasureRules(), multiscale::verification::ParserGrammar<
std::string::const_iterator >::initialiseFilterNumericMeasureRule(), multiscale::verification-
::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasure-
Rule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise-
PrimaryConstraintRule(), multiscale::verification::ParserGrammar< std::string::const-
_iterator >::initialiseSpatialMeasureRule(), and multiscale::verification::Parser-
Grammar< std::string::const_iterator >::initialiseSpatialMeasureRuleDebugging().
```

7.106.4.45 template<typename Iterator> SpatialMeasureTypeParser
multiscale::verification::ParserGrammar< Iterator
>::spatialMeasureTypeParser [private]

The spatial measure type parser

Definition at line 125 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >-
::initialiseSpatialMeasureRule().

7.106.4.46 template<typename Iterator> qi::rule<Iterator, std::string(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::stateVariableNameRule [private]

The rule for parsing the name of a state variable without escaping white space

Definition at line 110 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator
>::assignNamesToNumericStateVariableRules(), multiscale::verification::Parser-
Grammar< std::string::const_iterator >::initialiseNumericStateVariableRule(), multiscale-
::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericState-
VariableRuleDebugging(), and multiscale::verification::ParserGrammar< std::string-
::const_iterator >::initialiseStateVariableErrorHandlingSupport().

7.106.4.47 template<typename Iterator> qi::rule<Iterator, StateVariableAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::stateVariableRule [private]

The rule for parsing a state variable

Definition at line 109 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator
>::assignNamesToNumericStateVariableRules(), multiscale::verification::Parser-
Grammar< std::string::const_iterator >::initialiseNumericStateVariableRule(), and
multiscale::verification::ParserGrammar< std::string::const_iterator >::initialise-
NumericStateVariableRuleDebugging().

**7.106.4.48 template<typename Iterator> SubsetOperationTypeParser
multiscale::verification::ParserGrammar< Iterator
>::subsetOperationTypeParser [private]**

The subset operation type parser

Definition at line 123 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule().

**7.106.4.49 template<typename Iterator> qi::rule<Iterator, SubsetAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::subsetRule [private]**

The rule for parsing a subset

Definition at line 82 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToSubsetRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRuleDebugging().

**7.106.4.50 template<typename Iterator> qi::rule<Iterator, SubsetSpecificAttribute(),
ascii::space_type> multiscale::verification::ParserGrammar< Iterator
>::subsetSpecificRule [private]**

The rule for parsing a specific subset

Definition at line 83 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToSubsetRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRuleDebugging().

**7.106.4.51 template<typename Iterator> SubsetSpecificTypeParser
multiscale::verification::ParserGrammar< Iterator
>::subsetSpecificTypeParser [private]**

The subset specific type parser

Definition at line 122 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule().

7.106.4.52 template<typename Iterator> qi::rule<Iterator, SubsetSubsetOperationAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::subsetSubsetOperationRule [private]

The rule for parsing a subset subset operation

Definition at line 85 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToSubsetRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSubsetRuleDebugging().

7.106.4.53 template<typename Iterator> qi::rule<Iterator, TernarySubsetMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::ternarySubsetMeasureRule [private]

The rule for parsing a ternary subset measure

Definition at line 76 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialSubsetMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSpatialSubsetMeasureRuleDebugging().

7.106.4.54 template<typename Iterator> TernarySubsetMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::ternarySubsetMeasureTypeParser [private]

The ternary subset measure type parser

Definition at line 116 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule().

7.106.4.55 template<typename Iterator> qi::rule<Iterator, TernarySubsetAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::ternarySubsetRule [private]

The rule for parsing a ternary subset

Definition at line 71 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialMeasureRules(), multiscale::verification::Parser-

Grammar< std::string::const_iterator >::initialiseNumericSpatialMeasureErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRuleDebugging().

7.106.4.56 template<typename Iterator> qi::rule<Iterator, UnaryNumericFilterAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::unaryNumericFilterRule [private]

The rule for parsing a unary numeric filter measure

Definition at line 95 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToFilterNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRuleDebugging().

7.106.4.57 template<typename Iterator> qi::rule<Iterator, UnaryNumericMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::unaryNumericMeasureRule [private]

The rule for parsing a unary numeric measure

Definition at line 79 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNaryNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseFilterNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNaryNumericMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNaryNumericMeasureRuleDebugging(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule().

7.106.4.58 template<typename Iterator> UnaryNumericMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::unaryNumericMeasureTypeParser [private]

The unary numeric measure type parser

Definition at line 119 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNaryNumericMeasureRule().

7.106.4.59 template<typename Iterator> qi::rule<Iterator, UnaryNumericNumericAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::unaryNumericNumericRule [private]

The rule for parsing a unary numeric numeric attribute

Definition at line 65 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericMeasureRuleDebugging().

7.106.4.60 template<typename Iterator> qi::rule<Iterator, UnarySpatialConstraintAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::unarySpatialConstraintRule [private]

The rule for parsing a unary spatial constraint

Definition at line 91 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintErrorHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRuleDebugging().

7.106.4.61 template<typename Iterator> qi::rule<Iterator, UnarySubsetMeasureAttribute(), ascii::space_type> multiscale::verification::ParserGrammar<Iterator>::unarySubsetMeasureRule [private]

The rule for parsing a unary subset measure

Definition at line 74 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialSubsetMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseSpatialSubsetMeasureRuleDebugging().

7.106.4.62 template<typename Iterator> UnarySubsetMeasureTypeParser multiscale::verification::ParserGrammar< Iterator >::unarySubsetMeasureTypeParser [private]

The unary subset measure type parser

Definition at line 114 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialSubsetMeasureRule().

7.106.4.63 template<typename Iterator> qi::rule<Iterator, UnarySubsetAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::unarySubsetRule [private]

The rule for parsing a unary subset

Definition at line 69 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToNumericSpatialMeasureRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureErrorHandlerHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseNumericSpatialMeasureRuleDebugging().

7.106.4.64 template<typename Iterator> qi::rule<Iterator, UnaryTypeConstraintAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::unaryTypeConstraintRule [private]

The rule for parsing a unary type constraint

Definition at line 92 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToPrimaryConstraintRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintErrorHandlerHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRule(), and multiscale::verification::ParserGrammar< std::string::const_iterator >::initialisePrimaryConstraintRuleDebugging().

7.106.4.65 template<typename Iterator> qi::rule<Iterator, UntilLogicPropertyAttribute(), ascii::space_type> multiscale::verification::ParserGrammar< Iterator >::untilLogicPropertyRule [private]

The rule for parsing an "until" logic property

Definition at line 61 of file ParserGrammar.hpp.

Referenced by multiscale::verification::ParserGrammar< std::string::const_iterator >::assignNamesToComposedLogicPropertyRules(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyErrorHandlerHandlingSupport(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRule(), multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseComposedLogicPropertyRuleDebugging(), and

`multiscale::verification::ParserGrammar< std::string::const_iterator >::initialiseLogicPropertyRule()`.

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/parsing/ParserGrammar.hpp`

7.107 multiscale::verification::ParserGrammarExceptionHandler Class Reference

Class for handling parser grammar exceptions.

```
#include <ParserGrammarExceptionHandler.hpp>
```

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.*

7.107 multiscale::verification::ParserGrammarExceptionHandler Class Reference

7.107.1 Detailed Description

Class for handling parser grammar exceptions.

Definition at line 16 of file ParserGrammarExceptionHandler.hpp.

7.107.2 Member Function Documentation

7.107.2.1 string ParserGrammarExceptionHandler::getIntroductoryErrorMessage () [static, private]

Return the generic introductory error message.

Definition at line 125 of file ParserGrammarExceptionHandler.cpp.

7.107.2.2 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

<i>initialString</i>	The initial std::string
<i>expectedToken</i>	The token which should replace the error token

Definition at line 107 of file ParserGrammarExceptionHandler.cpp.

7.107.2.3 void ParserGrammarExceptionHandler::handleExtraInputException (const std::string & initialString, const std::string & extraInput) [static]

Handle the exception when extra input is provided.

Parameters

<i>initialString</i>	The initial std::string
<i>extraInput</i>	Extra input

Definition at line 65 of file ParserGrammarExceptionHandler.cpp.

References MS_throw.

7.107.2.4 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>expected- Token</i>	The token which should replace the error token

Definition at line 27 of file ParserGrammarExceptionHandler.cpp.

References MS_throw.

**7.107.2.5 void ParserGrammarExceptionHandler::handleUnexpectedToken-
Exception (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>expected- Token</i>	The token which should replace the error token

Definition at line 13 of file ParserGrammarExceptionHandler.cpp.

References MS_throw.

**7.107.2.6 string ParserGrammarExceptionHandler::handleUnexpectedTokenIn-
String (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>expected- Token</i>	The token which should replace the error token

Definition at line 84 of file ParserGrammarExceptionHandler.cpp.

7.107.2.7 void ParserGrammarExceptionHandler::handleUnparseableInput-Exception (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 MS_throw.

7.107.2.8 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().

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

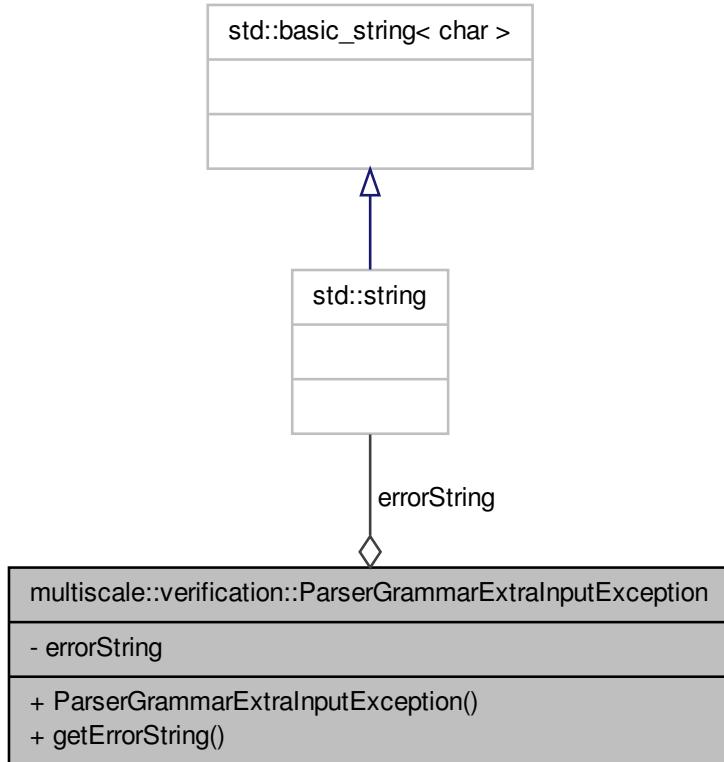
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ParserGrammarExceptionHandler.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/[ParserGrammarExceptionHandler.cpp](#)

7.108 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`

7.108.1 Detailed Description

Class for representing "extra input" exceptions in the parsing process.

Definition at line 14 of file ParserGrammarExtraInputException.hpp.

7.108.2 Constructor & Destructor Documentation

7.108.2.1 **multiscale::verification::ParserGrammarExtraInputException::-ParserGrammarExtraInputException (const std::string & errorString)**
[inline]

Definition at line 23 of file ParserGrammarExtraInputException.hpp.

7.108.3 Member Function Documentation

7.108.3.1 **std::string multiscale::verification::ParserGrammarExtraInputException::getErrorString () const**
[inline]

Get the error std::string.

Definition at line 28 of file ParserGrammarExtraInputException.hpp.

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

7.108.4 Member Data Documentation

7.108.4.1 **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.

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

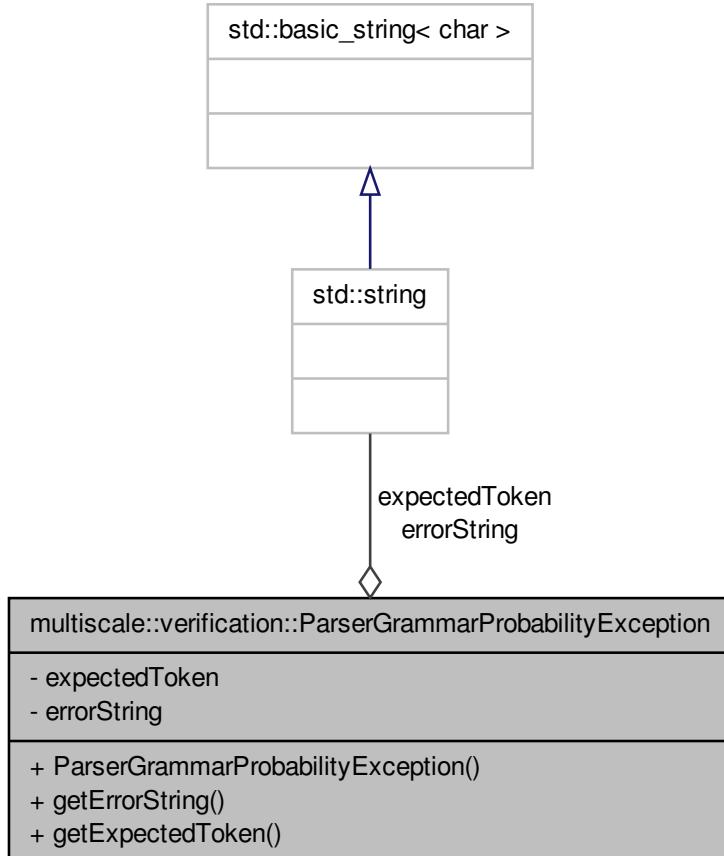
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ParserGrammarExtraInputException.hpp](#)

7.109 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](#)

7.109.1 Detailed Description

Class for representing "probability" exceptions in the parsing process.

Definition at line 12 of file ParserGrammarProbabilityException.hpp.

7.109.2 Constructor & Destructor Documentation**7.109.2.1 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](#).

7.109.3 Member Function Documentation**7.109.3.1 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\(\)](#).

7.109.3.2 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\(\)](#).

7.109.4 Member Data Documentation

7.109.4.1 **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().

7.109.4.2 **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](#)

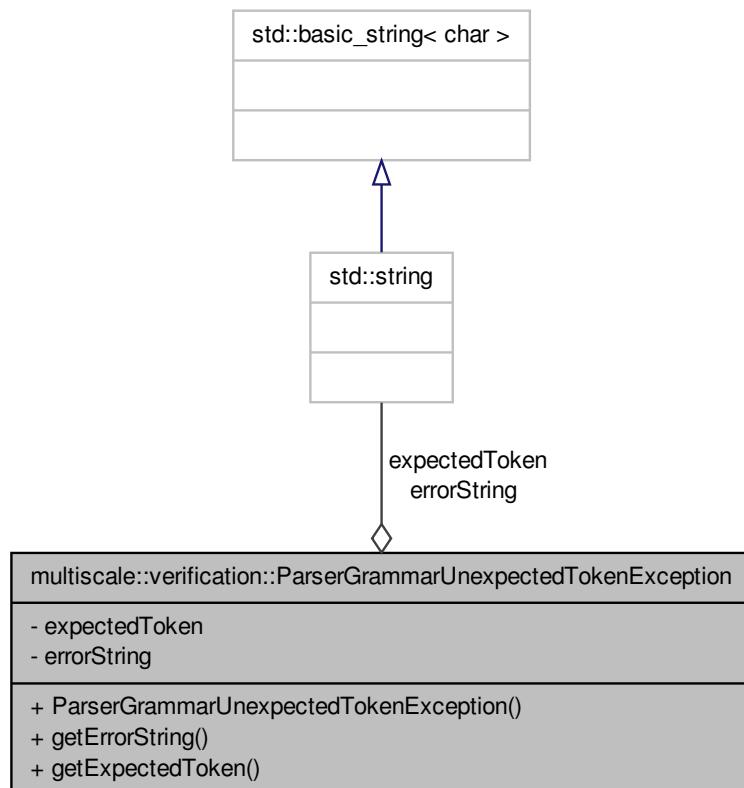
7.110 multiscale::verification::ParserGrammarUnexpectedToken- Exception Class Reference

Class for representing "unexpected token" exceptions in the parsing process.

```
#include <ParserGrammarUnexpectedTokenException.hpp>
```

Collaboration diagram for multiscale::verification::ParserGrammarUnexpectedToken-

Exception:



Public Member Functions

- `ParserGrammarUnexpectedTokenException` (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`

7.110.1 Detailed Description

Class for representing "unexpected token" exceptions in the parsing process.

Definition at line 12 of file ParserGrammarUnexpectedTokenException.hpp.

7.110.2 Constructor & Destructor Documentation

7.110.2.1 `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.

7.110.3 Member Function Documentation

7.110.3.1 `std::string multiscale::verification::ParserGrammar-
UnexpectedTokenException::getErrorString () const
[inline]`

Get the error string.

Definition at line 29 of file ParserGrammarUnexpectedTokenException.hpp.

References errorString.

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

7.110.3.2 `std::string multiscale::verification::ParserGrammar-
UnexpectedTokenException::getExpectedToken () const
[inline]`

Get the expected token.

Definition at line 34 of file ParserGrammarUnexpectedTokenException.hpp.

References expectedToken.

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

7.110.4 Member Data Documentation

7.110.4.1 `std::string multiscale::verification::ParserGrammarUnexpectedToken-
Exception::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 `getErrorMessage()`, and `ParserGrammarUnexpectedTokenException()`.

7.110.4.2 std::string multiscale::verification::ParserGrammarUnexpectedToken-Exception::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](#)

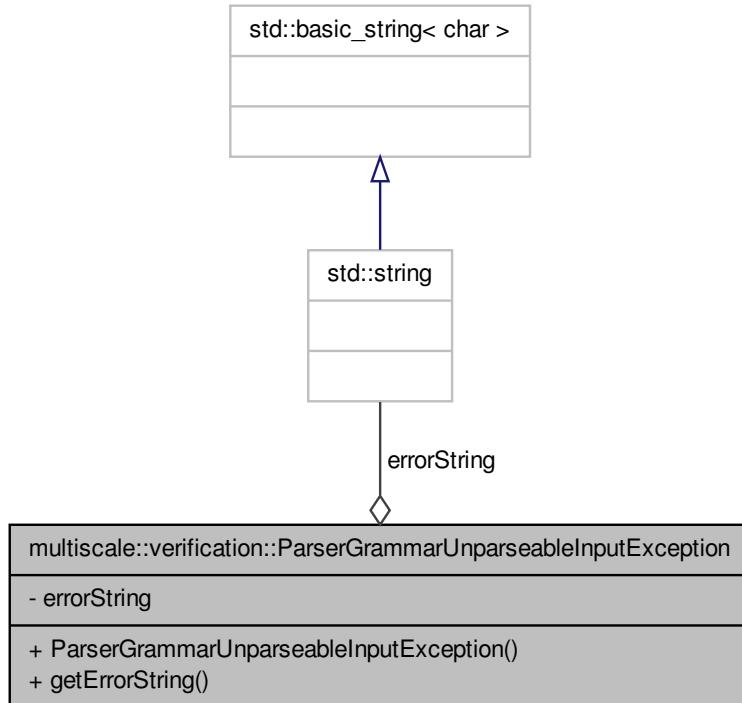
7.111 multiscale::verification::ParserGrammarUnparseableInputException Class Reference

Class for representing "unparseable input" exceptions in the parsing process.

```
#include <ParserGrammarUnparseableInputException.hpp>
```

Collaboration diagram for `multiscale::verification::ParserGrammarUnparseableInput-`

Exception:



Public Member Functions

- [ParserGrammarUnparseableInputException](#) (const std::string &[errorString](#))
- std::string [getErrorString](#) () const
Get the error string.

Private Attributes

- std::string [errorString](#)

7.111.1 Detailed Description

Class for representing "unparseable input" exceptions in the parsing process.

Definition at line 14 of file ParserGrammarUnparseableInputException.hpp.

7.111.2 Constructor & Destructor Documentation

7.111.2.1 **multiscale::verification::ParserGrammarUnparseableInputException::ParserGrammarUnparseableInputException (const std::string & *errorString*) [inline]**

Definition at line 23 of file ParserGrammarUnparseableInputException.hpp.

7.111.3 Member Function Documentation

7.111.3.1 **std::string multiscale::verification::ParserGrammarUnparseableInputException::getErrorResponse () const [inline]**

Get the error string.

Definition at line 28 of file ParserGrammarUnparseableInputException.hpp.

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

7.111.4 Member Data Documentation

7.111.4.1 **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.

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

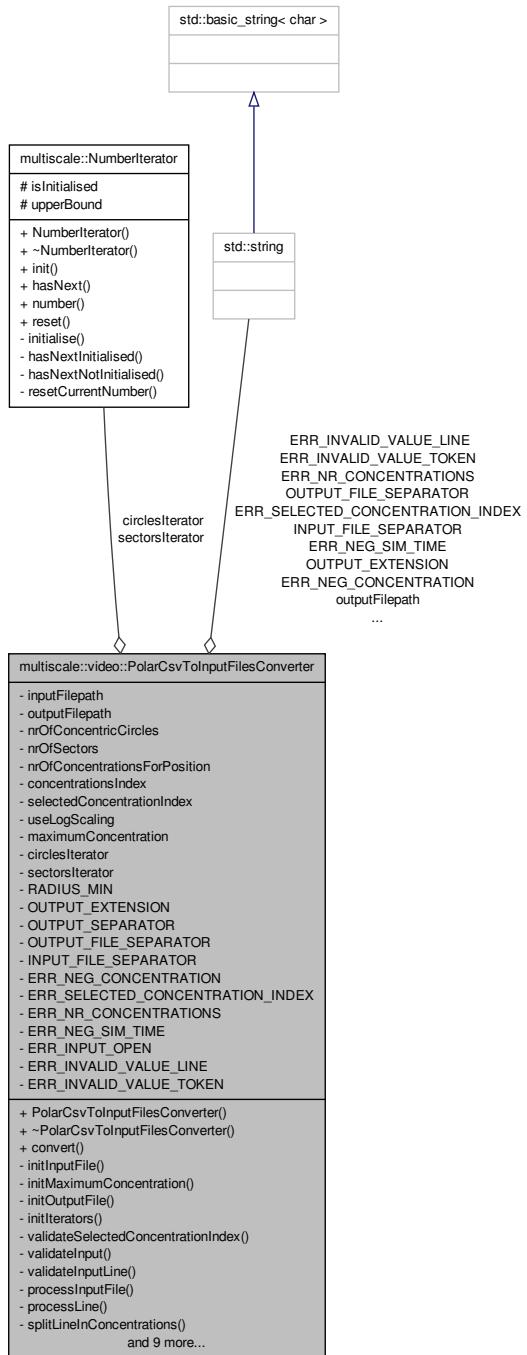
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[ParserGrammarUnparseableInputException.hpp](#)

7.112 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 string &inputFilepath, const 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 (ifstream &fin)`
Initialise the input file stream over the given input file.
- `void initMaximumConcentration (ifstream &fin)`
Compute the value of member maximum concentration.
- `void initOutputFile (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 (ifstream &fin)`
Validate the input.
- `void validateInputLine (const string &line, unsigned int lineNumber)`
Validate the provided line identified by a line number.
- `void processInputFile (ifstream &fin)`
Process the input file.
- `void processLine (const string &line, unsigned int outputIndex)`
Process the provided line.
- `vector< double > splitLineInConcentrations (const string &line, double &simulationTime)`
Split the line in concentrations.
- `void splitFirstPartInConcentrations (vector< double > &concentrations, const vector< string > &tokens, unsigned int circleIndex)`
Split first part of the line (i.e. part representing the origin) into concentrations.
- `void splitOtherPartsInConcentrations (vector< double > &concentrations, const vector< string > &tokens, unsigned int circleIndex)`
Split other parts of the line (i.e. non-first part) into concentrations.
- `double computeSimulationTime (const string &token)`
Compute the simulation time from the given token and check if it is valid.
- `double computeNextPositionConcentration (unsigned int circleIndex, int concentrationIndex, const vector< string > &tokens)`

- `double computeConcentration (const string &concentration, int circleIndex)`

Compute the concentration from the given string considering the index of the current concentric circle.
- `double computeNonScaledConcentration (const string &concentration, int circleIndex)`

Compute the non-scaled concentration from the given string considering the index of the current concentric circle.
- `double computeScaledConcentration (const string &concentration, int circleIndex)`

Compute the scaled concentration from the given string considering the index of the current concentric circle.
- `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 by dividing it to the maximum concentration.
- `void updateMaximumConcentration (const string &line, double &maximumConcentration)`

Update the maximum concentration if the values from the given line are greater than it.

Private Attributes

- `string inputfilepath`
- `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 string OUTPUT_EXTENSION = ".in"`
- `static const string OUTPUT_SEPARATOR = " "`
- `static const string OUTPUT_FILE_SEPARATOR = "_"`
- `static const string INPUT_FILE_SEPARATOR = ","`
- `static const string ERR_NEG_CONCENTRATION = "All concentrations must be non-negative."`

- static const string `ERR_SELECTED_CONCENTRATION_INDEX` = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations."
- static const 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 string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const string `ERR_INVALID_VALUE_LINE` = "Invalid value on line: "
- static const string `ERR_INVALID_VALUE_TOKEN` = ", value: "

7.112.1 Detailed Description

Csv file to input file converter considering polar coordinates.

Definition at line 18 of file PolarCsvToInputFilesConverter.hpp.

7.112.2 Constructor & Destructor Documentation

7.112.2.1 PolarCsvToInputFilesConverter::PolarCsvToInputFilesConverter
`(const string & inputFilepath, const string & outputFilepath, unsigned int nrOfConcentricCircles, unsigned int nrOfSectors, unsigned int nrOfConcentrationsForPosition, unsigned int selectedConcentrationIndex, bool useLogScaling, NumberIteratorType numberIteratorType)`

Definition at line 21 of file PolarCsvToInputFilesConverter.cpp.

7.112.2.2 PolarCsvToInputFilesConverter::~PolarCsvToInputFilesConverter()

Definition at line 45 of file PolarCsvToInputFilesConverter.cpp.

7.112.3 Member Function Documentation

7.112.3.1 double PolarCsvToInputFilesConverter::computeConcentration (const string & *concentration*, int *circleIndex*) [inline, private]

Compute the concentration from the given string considering the index of the current concentric circle.

Parameters

<code>concentra-</code> <code>tion</code>	String representing the concentration
<code>circleIndex</code>	Index of the concentric circle

Definition at line 307 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.2 double PolarCsvToInputFilesConverter::computeConcentrationWrtArea (double *amount*, int *circleIndex*) [inline, 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 333 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.3 double PolarCsvToInputFilesConverter::computeNextPosition- Concentration (unsigned int *circleIndex*, int *concentrationIndex*, const vector< string > & *tokens*) [inline, 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 278 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.4 double PolarCsvToInputFilesConverter::computeNonScaled- Concentration (const string & *concentration*, int *circleIndex*) [inline, private]

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 313 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.5 double PolarCsvToInputFilesConverter::computeNormalisedConcentration (double *concentration*, int *circleIndex*) [inline, private]

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 337 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.6 double PolarCsvToInputFilesConverter::computeScaledConcentration (const string & *concentration*, int *circleIndex*) [inline, private]

Compute the scaled concentration from the given string considering the index of the current concentric circle.

Compute the scaled concentration from the given string 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 319 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.7 double PolarCsvToInputFilesConverter::computeSimulationTime (const string & *token*) [inline, private]

Compute the simulation time from the given token and check if it is valid.

Parameters

<i>token</i>	Token (string)
--------------	----------------

Definition at line 268 of file PolarCsvToInputFilesConverter.cpp.

References MS_throw.

7.112.3.8 void PolarCsvToInputFilesConverter::convert ()

Start the conversion.

Definition at line 50 of file PolarCsvToInputFilesConverter.cpp.

Referenced by main().

7.112.3.9 void PolarCsvToInputFilesConverter::initInputModule (ifstream & *fin*) [private]

Initialise the input file stream over the given input file.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 63 of file PolarCsvToInputFilesConverter.cpp.

References MS_throw.

7.112.3.10 void PolarCsvToInputFilesConverter::initIterators (const NumberIteratorType & *numberIteratorType*) [private]

Initialise the iterators considering the given number iterator type.

Parameters

<i>number- IteratorType</i>	The type of the number iterator
---------------------------------	---------------------------------

Definition at line 112 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

7.112.3.11 void PolarCsvToInputFilesConverter::initMaximumConcentration (ifstream & *fin*) [private]

Compute the value of member maximum concentration.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 71 of file PolarCsvToInputFilesConverter.cpp.

References MS_throw.

7.112.3.12 void PolarCsvToInputFilesConverter::initOutputFile (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>simulation-Time</i>	Simulation time

Definition at line 95 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

7.112.3.13 void PolarCsvToInputFilesConverter::processInputFile (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 PolarCsvToInputFilesConverter.cpp.

7.112.3.14 void PolarCsvToInputFilesConverter::processLine (const 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 PolarCsvToInputFilesConverter.cpp.

7.112.3.15 void PolarCsvToInputFilesConverter::splitFirstPartInConcentrations (vector< double > & *concentrations*, const vector< 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 238 of file PolarCsvToInputFilesConverter.cpp.

```
7.112.3.16 vector< double > PolarCsvToInputFilesConverter::splitLine-
InConcentrations ( const string & line, double & simulationTime )
[private]
```

Split the line in concentrations.

Parameters

<i>line</i>	Line
<i>simulation- Time</i>	Simulation time associated with the line

Definition at line 213 of file PolarCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

```
7.112.3.17 void PolarCsvToInputFilesConverter::splitOtherPartsInConcentrations (
    vector< double > & concentrations, const vector< string > & tokens, unsigned int
    circleIndex ) [private]
```

Split other parts of the line (i.e. non-first part) into concentrations.

Parameters

<i>concentra- tions</i>	Concentrations extracted from tokens
<i>tokens</i>	Tokens representing the line
<i>circleIndex</i>	Index of the current concentric circle

Definition at line 252 of file PolarCsvToInputFilesConverter.cpp.

```
7.112.3.18 void PolarCsvToInputFilesConverter::updateMaximumConcentration (
    const 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>maximum- Concentration</i>	The maximum concentration

Definition at line 341 of file PolarCsvToInputFilesConverter.cpp.

7.112.3.19 void PolarCsvToInputFilesConverter::validateInput (ifstream & *fin*) [private]

Validate the input.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 135 of file PolarCsvToInputFilesConverter.cpp.

References MS_throw.

7.112.3.20 void PolarCsvToInputFilesConverter::validateInputLine (const 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().

7.112.3.21 void PolarCsvToInputFilesConverter::validateSelectedConcentration-Index() [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.

7.112.4 Member Data Documentation

7.112.4.1 NumberIterator* multiscale::video::PolarCsvToInputFilesConverter-::circlesIterator [private]

Iterator over the number of concentric circles

Definition at line 42 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.2 unsigned int multiscale::video::PolarCsvToInputFilesConverter-  
::concentrationsIndex [private]
```

Index of the current concentration

Definition at line 29 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.3 const string PolarCsvToInputFilesConverter::ERR_INPUT_OPEN = "The  
input file could not be opened." [static, private]
```

Definition at line 218 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.4 const string PolarCsvToInputFilesConverter::ERR_INVALID_VALUE_LINE  
= "Invalid value on line: " [static, private]
```

Definition at line 219 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.5 const string PolarCsvToInputFilesConverter::ERR_-  
INVALID_VALUE_TOKEN = ", value: " [static,  
private]
```

Definition at line 220 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.6 const string PolarCsvToInputFilesConverter::ERR_NEG_CONCE-  
NTRATION = "All concentrations must be non-negative." [static,  
private]
```

Definition at line 214 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.7 const string PolarCsvToInputFilesConverter::ERR_NEG_SIM_TIME = "The  
simulation time must be non-negative." [static, private]
```

Definition at line 217 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.8 const string PolarCsvToInputFilesConverter::ERR_NR_CONCENTRATIO-  
NS = "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 216 of file PolarCsvToInputFilesConverter.hpp.

```
7.112.4.9 const string PolarCsvToInputFilesConverter::ERR_SELECTED_CONCEN-  
TRATION_INDEX = "The selected concentration index (0-based indexing) should be  
smaller than the number of concentrations." [static, private]
```

Definition at line 215 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.10 const string PolarCsvToInputFilesConverter::INPUT_FILE_SEPARATOR = "," [static, private]

Definition at line 212 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.11 string multiscale::video::PolarCsvToInputFilesConverter::inputFilepath [private]

Path to the input file

Definition at line 22 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.12 double multiscale::video::PolarCsvToInputFilesConverter::maximum-Concentration [private]

The maximum concentration in the input file

Definition at line 40 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.13 unsigned int multiscale::video::PolarCsvToInput-FilesConverter::nrOfConcentrationsForPosition [private]

Number of concentrations for each position

Definition at line 27 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.14 unsigned int multiscale::video::PolarCsvToInputFilesConverter::nrOf-ConcentricCircles [private]

Number of concentric circles

Definition at line 25 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.15 unsigned int multiscale::video::PolarCsvToInputFilesConverter::nrOf-Sectors [private]

Number of sectors

Definition at line 26 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.16 const string PolarCsvToInputFilesConverter::OUTPUT_EXTENSION = ".in" [static, private]

Definition at line 209 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.17 **const string PolarCsvToInputFilesConverter::OUTPUT_FILE_SEPARATOR = "_" [static, private]**

Definition at line 211 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.18 **const string PolarCsvToInputFilesConverter::OUTPUT_SEPARATOR = " " [static, private]**

Definition at line 210 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.19 **string multiscale::video::PolarCsvToInputFilesConverter::outputFilepath [private]**

Path to the output file

Definition at line 23 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.20 **const int PolarCsvToInputFilesConverter::RADIUS_MIN = 1 [static, private]**

Definition at line 207 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.21 **NumberIterator* multiscale::video::PolarCsvToInputFilesConverter::sectorsIterator [private]**

Iterator over the number of sectors

Definition at line 43 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.22 **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 31 of file PolarCsvToInputFilesConverter.hpp.

7.112.4.23 **bool multiscale::video::PolarCsvToInputFilesConverter::useLogScaling [private]**

Flag for using logarithmic scaling for concentrations or not

Definition at line 38 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](#)

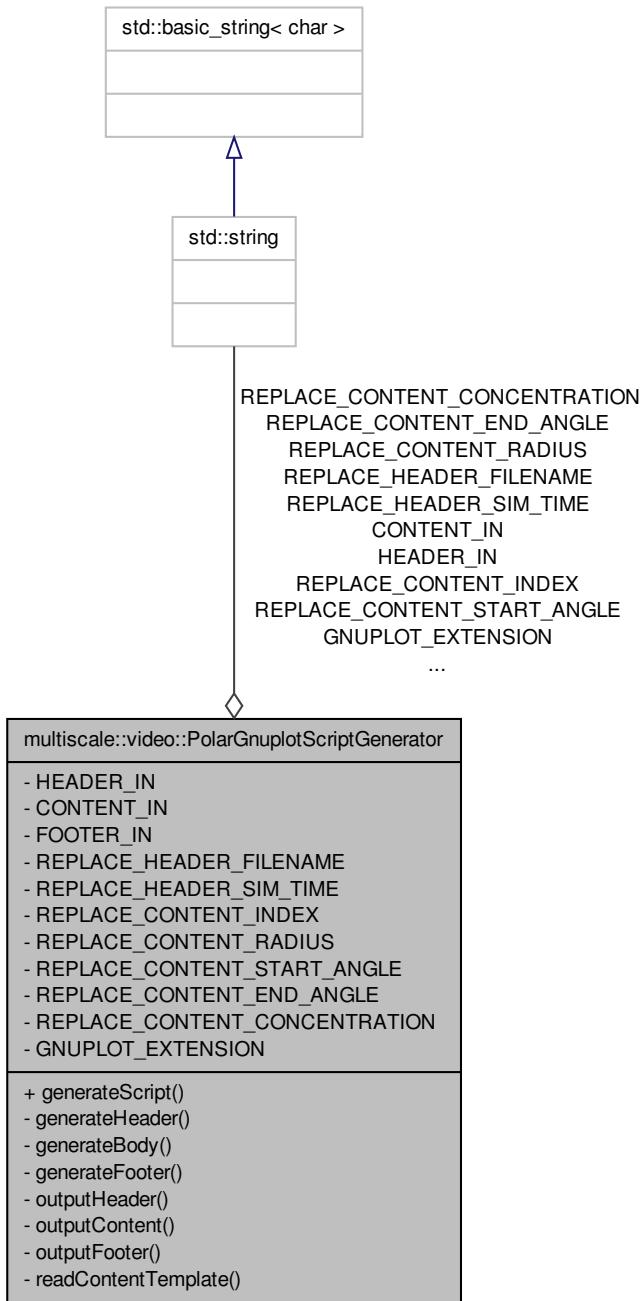
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/-PolarCsvToInputFilesConverter.cpp](#)

7.113 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 vector< `AnnularSector` > &annularSectors, double simulationTime, const string &outputFilepath)

Generate the script.

Static Private Member Functions

- static void `generateHeader` (ofstream &fout, const string &outputFilepath, double simulationTime)

Generate the header of the script.

- static void `generateBody` (const vector< `AnnularSector` > &annularSectors, ofstream &fout)

Generate the body/content of the script.

- static void `generateFooter` (ofstream &fout)

Generate the footer of the script.

- static void `outputHeader` (ifstream &fin, const string &outputFilename, double simulationTime, ofstream &fout)

Output the header of the script.

- static void `outputContent` (const vector< `AnnularSector` > &annularSectors, const string &contentTemplate, ofstream &fout)

Output the content of the script.

- static void `outputFooter` (ifstream &fin, ofstream &fout)

Output the footer of the script.

- static string `readContentTemplate` (ifstream &fin)

Read content template.

Static Private Attributes

- static const string `HEADER_IN` = "config/video/circular/header.in"
- static const string `CONTENT_IN` = "config/video/circular/content.in"
- static const string `FOOTER_IN` = "config/video/circular/footer.in"
- static const string `REPLACE_HEADER_FILENAME` = "OUTPUT_FILENAME"
- static const string `REPLACE_HEADER_SIM_TIME` = "OUTPUT_SIM_TIME"
- static const string `REPLACE_CONTENT_INDEX` = "OBJ_INDEX"
- static const string `REPLACE_CONTENT_RADIUS` = "OBJ_END_RADIUS"
- static const string `REPLACE_CONTENT_START_ANGLE` = "OBJ_START_ANGLE"
- static const string `REPLACE_CONTENT_END_ANGLE` = "OBJ_END_ANGLE"
- static const string `REPLACE_CONTENT_CONCENTRATION` = "OBJ_CONCENTRATION"
- static const string `GNUPLOT_EXTENSION` = ".plt"

7.113.1 Detailed Description

Gnuplot script generator from the provided annular sectors.

Definition at line 16 of file PolarGnuplotScriptGenerator.hpp.

7.113.2 Member Function Documentation

**7.113.2.1 void PolarGnuplotScriptGenerator::generateBody (const vector<
AnnularSector > & *annularSectors*, ofstream & *fout*) [static,
private]**

Generate the body/content of the script.

Parameters

<i>annular- Sectors</i>	Annular sectors
<i>fout</i>	Output file stream

Definition at line 40 of file PolarGnuplotScriptGenerator.cpp.

**7.113.2.2 void PolarGnuplotScriptGenerator::generateFooter (ofstream & *fout*)
[static, private]**

Generate the footer of the script.

Parameters

<i>fout</i>	Output file stream
-------------	--------------------

Definition at line 52 of file PolarGnuplotScriptGenerator.cpp.

**7.113.2.3 void PolarGnuplotScriptGenerator::generateHeader (ofstream & *fout*, const
string & *outputFilepath*, double *simulationTime*) [static, private]**

Generate the header of the script.

Parameters

<i>fout</i>	Output file stream
<i>output- Filepath</i>	Path to the output file
<i>simulation- Time</i>	Simulation time

Definition at line 28 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::filenameFromPath().

```
7.113.2.4 void PolarGnuplotScriptGenerator::generateScript ( const vector<
    AnnularSector > & annularSectors, double simulationTime, const string &
    outputfilepath ) [static]
```

Generate the script.

Parameters

<i>annular- Sectors</i>	Annular sectors
<i>simulation- Time</i>	Simulation time
<i>output- Filepath</i>	Path of the output file

Definition at line 14 of file PolarGnuplotScriptGenerator.cpp.

Referenced by multiscale::video::CartesianToPolarConverter::outputResultsAsScript().

```
7.113.2.5 void PolarGnuplotScriptGenerator::outputContent ( const vector<
    AnnularSector > & annularSectors, const string & contentTemplate, ofstream &
    fout ) [static, private]
```

Output the content of the script.

Parameters

<i>annular- Sectors</i>	Annular sectors
<i>content- Template</i>	Template used for generating output for each annular sector
<i>fout</i>	Output file stream

Definition at line 75 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::replace().

```
7.113.2.6 void PolarGnuplotScriptGenerator::outputFooter ( ifstream & fin, 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 93 of file PolarGnuplotScriptGenerator.cpp.

7.113.2.7 void PolarGnuplotScriptGenerator::outputHeader (ifstream & *fin*, const string & *outputFilename*, double *simulationTime*, ofstream &) [static, private]

Output the header of the script.

Parameters

<i>fin</i>	Input file stream
<i>output-Filename</i>	Name of the output file
<i>simulation-Time</i>	Simulation time
<ifout< i=""></ifout<>	Output file stream

Definition at line 62 of file PolarGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::replace().

7.113.2.8 string PolarGnuplotScriptGenerator::readContentTemplate (ifstream & *fin*) [static, private]

Read content template.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 103 of file PolarGnuplotScriptGenerator.cpp.

7.113.3 Member Data Documentation

7.113.3.1 const string PolarGnuplotScriptGenerator::CONTENT_IN = "config/video/circular/content.in" [static, private]

Definition at line 92 of file PolarGnuplotScriptGenerator.hpp.

7.113.3.2 const string PolarGnuplotScriptGenerator::FOOTER_IN = "config/video/circular/footer.in" [static, private]

Definition at line 93 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.3 const string PolarGnuplotScriptGenerator::GNUPLOT_EXTENSION = ".plt"
[static, private]
```

Definition at line 104 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.4 const string PolarGnuplotScriptGenerator::HEADER_IN =
"config/video/circular/header.in" [static, private]
```

Definition at line 91 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.5 const string PolarGnuplotScriptGenerator::REPLACE_CONTENT_CONCENTRATION = "OBJ_CONCENTRATION" [static,
private]
```

Definition at line 102 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.6 const string PolarGnuplotScriptGenerator::REPLACE_CONTENT_END_ANGLE = "OBJ_END_ANGLE" [static,
private]
```

Definition at line 101 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.7 const string PolarGnuplotScriptGenerator::REPLACE_CONTENT_INDEX =
"OBJ_INDEX" [static, private]
```

Definition at line 98 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.8 const string PolarGnuplotScriptGenerator::REPLACE_CONTENT_RADIUS = "OBJ_END_RADIUS" [static,
private]
```

Definition at line 99 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.9 const string PolarGnuplotScriptGenerator::REPLACE_CONTENT_START_ANGLE = "OBJ_START_ANGLE" [static,
private]
```

Definition at line 100 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.10 const string PolarGnuplotScriptGenerator::REPLACE_HEADER_FILENAME = "OUTPUT_FILENAME" [static,
private]
```

Definition at line 95 of file PolarGnuplotScriptGenerator.hpp.

```
7.113.3.11 const string PolarGnuplotScriptGenerator::REPLACE_-
    HEADER_SIM_TIME = "OUTPUT_SIM_TIME" [static,
    private]
```

Definition at line 96 of file PolarGnuplotScriptGenerator.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/[PolarGnuplotScriptGenerator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/[PolarGnuplotScriptGenerator.cpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/[PolarGnuplotScriptGenerator.in.cpp](#)

7.114 multiscale::verification::PrimaryConstraintAttribute Class Reference

Class for representing a primary constraint attribute.

```
#include <PrimaryConstraintAttribute.hpp>
```

Public Attributes

- [PrimaryConstraintAttributeType primaryConstraint](#)

7.114.1 Detailed Description

Class for representing a primary constraint attribute.

Definition at line 32 of file PrimaryConstraintAttribute.hpp.

7.114.2 Member Data Documentation

7.114.2.1 PrimaryConstraintAttributeType multiscale::verification::Primary- ConstraintAttribute::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/[Primary-
 ConstraintAttribute.hpp](#)

7.115 multiscale::verification::PrimaryLogicPropertyAttribute - Class Reference

Class for representing a primary logic property attribute.

```
#include <PrimaryLogicPropertyAttribute.hpp>
```

Public Attributes

- [PrimaryLogicPropertyAttributeType primaryLogicProperty](#)

7.115.1 Detailed Description

Class for representing a primary logic property attribute.

Definition at line 40 of file PrimaryLogicPropertyAttribute.hpp.

7.115.2 Member Data Documentation

7.115.2.1 PrimaryLogicPropertyAttributeType multiscale::verification::PrimaryLogicPropertyAttribute::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](#)

7.116 multiscale::verification::PrimaryNumericMeasureAttribute Class Reference

Class for representing a primary numeric measure attribute.

```
#include <PrimaryNumericMeasureAttribute.hpp>
```

Public Attributes

- [PrimaryNumericMeasureAttributeType primaryNumericMeasure](#)

7.116.1 Detailed Description

Class for representing a primary numeric measure attribute.

Definition at line 28 of file PrimaryNumericMeasureAttribute.hpp.

7.116.2 Member Data Documentation

7.116.2.1 PrimaryNumericMeasureAttributeType multiscale::verification::PrimaryNumericMeasureAttribute::primaryNumericMeasure

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/[Primary-NumericMeasureAttribute.hpp](#)

7.117 multiscale::verification::ProbabilisticBlackBoxModelChecker Class Reference

Class used to run probabilistic black-box model checking tasks.

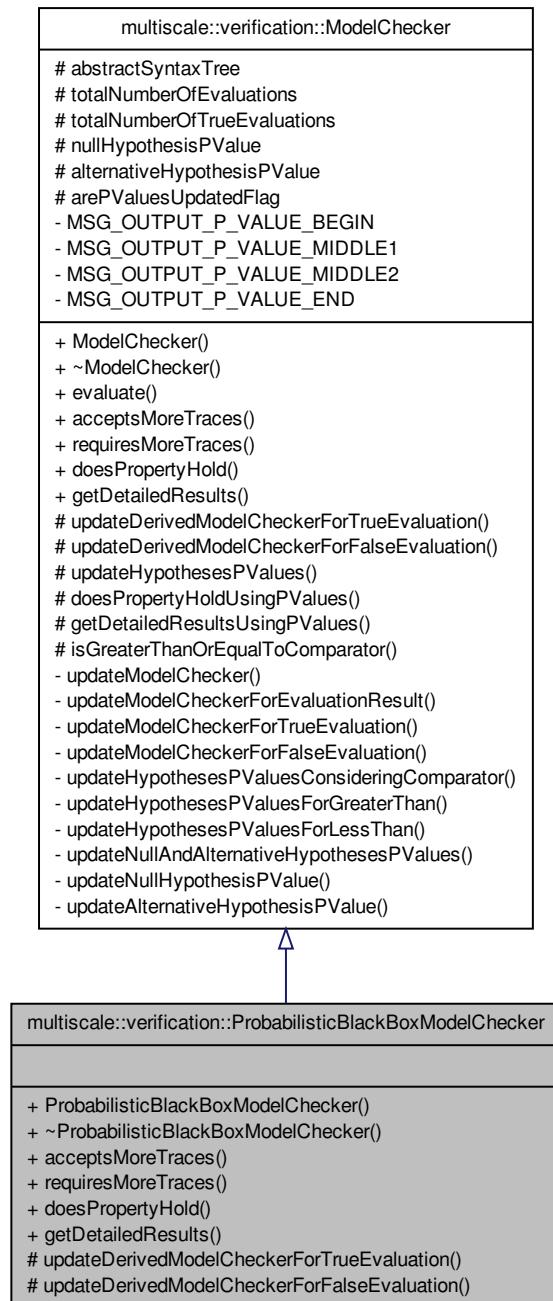
```
#include <ProbabilisticBlackBoxModelChecker.hpp>
```

7.117 multiscale::verification::ProbabilisticBlackBoxModelChecker Class

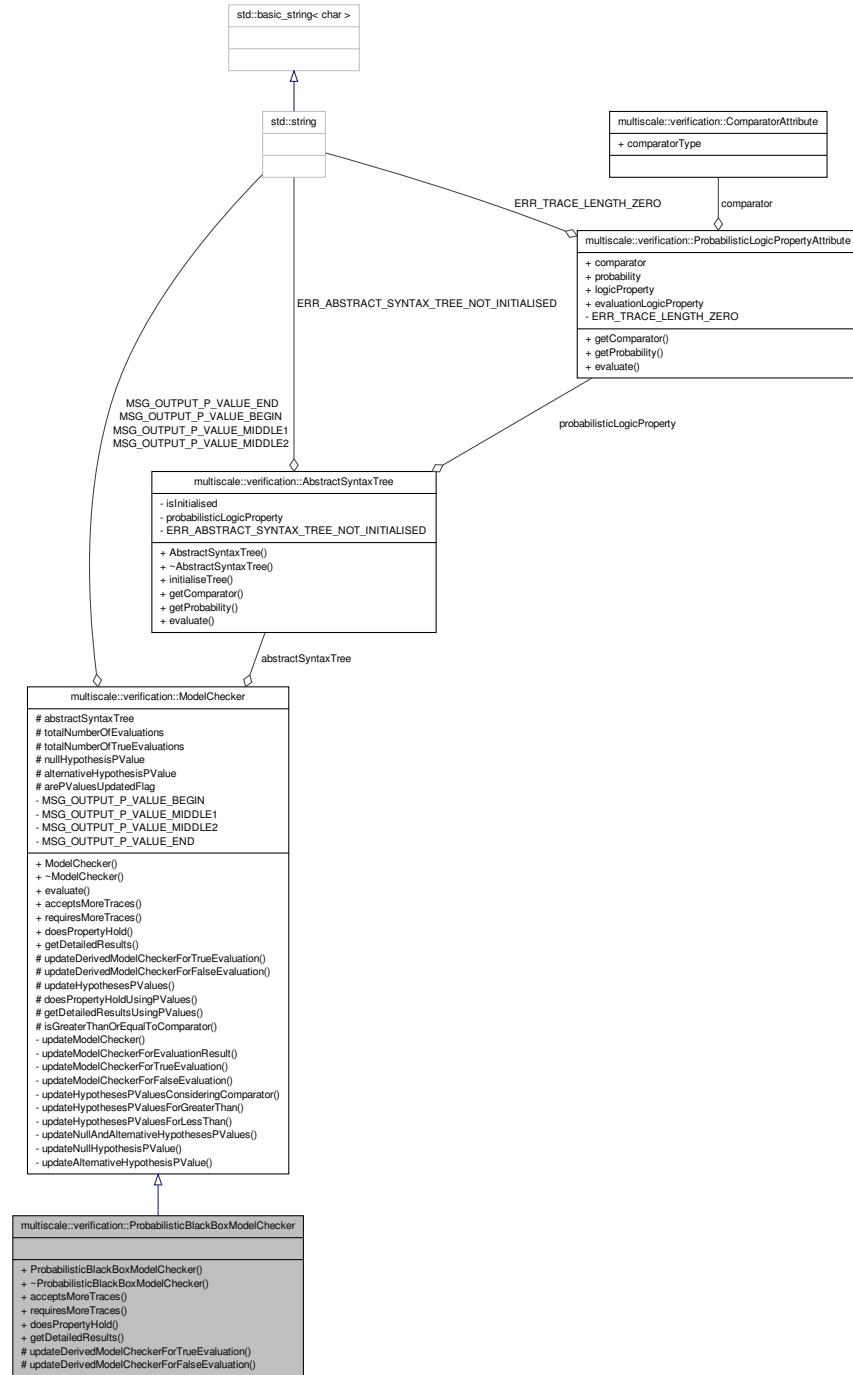
Reference

699

Inheritance diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker:



Collaboration diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker:



Public Member Functions

- `ProbabilisticBlackBoxModelChecker (const AbstractSyntaxTree &abstractSyntaxTree)`
- `~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.

7.117.1 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. 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.

7.117.2 Constructor & Destructor Documentation

7.117.2.1 ProbabilisticBlackBoxModelChecker::ProbabilisticBlackBoxModelChecker (const AbstractSyntaxTree & abstractSyntaxTree)

Definition at line 12 of file ProbabilisticBlackBoxModelChecker.cpp.

7.117.2.2 **ProbabilisticBlackBoxModelChecker::~ProbabilisticBlackBoxModelChecker()**

Definition at line 15 of file ProbabilisticBlackBoxModelChecker.cpp.

7.117.3 Member Function Documentation

7.117.3.1 **bool ProbabilisticBlackBoxModelChecker::acceptsMoreTraces()**
[override, virtual]

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

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

Definition at line 17 of file ProbabilisticBlackBoxModelChecker.cpp.

7.117.3.2 **bool ProbabilisticBlackBoxModelChecker::doesPropertyHold()**
[override, virtual]

Check if the given property holds.

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

Definition at line 25 of file ProbabilisticBlackBoxModelChecker.cpp.

References [multiscale::verification::ModelChecker::doesPropertyHoldUsingPValues\(\)](#).

7.117.3.3 **std::string ProbabilisticBlackBoxModelChecker::getDetailedResults()**
[override, virtual]

Get the detailed description of the results.

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

Definition at line 29 of file ProbabilisticBlackBoxModelChecker.cpp.

References [multiscale::verification::ModelChecker::getDetailedResultsUsingPValues\(\)](#).

7.117.3.4 **bool ProbabilisticBlackBoxModelChecker::requiresMoreTraces()**
[override, virtual]

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

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

Definition at line 21 of file ProbabilisticBlackBoxModelChecker.cpp.

**7.117.3.5 void ProbabilisticBlackBoxModelChecker::updateDerivedModel-
CheckerForFalseEvaluation() [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 35 of file ProbabilisticBlackBoxModelChecker.cpp.

**7.117.3.6 void ProbabilisticBlackBoxModelChecker::updateDerivedModel-
CheckerForTrueEvaluation() [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 33 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](#)

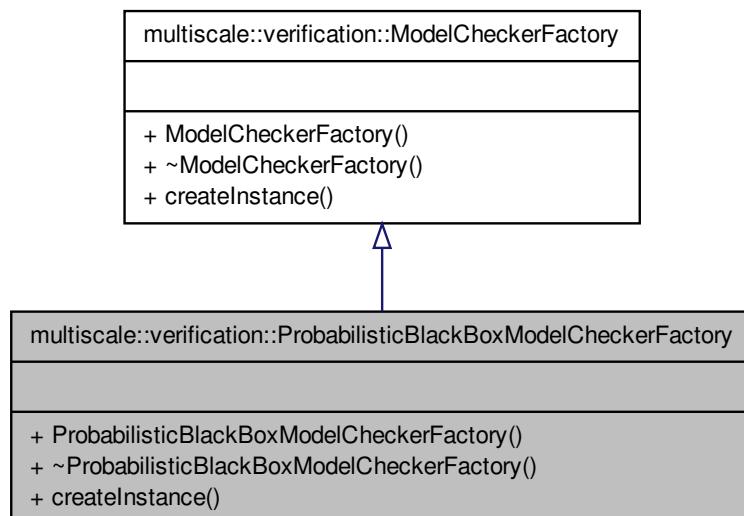
7.118 multiscale::verification::ProbabilisticBlackBoxModelChecker- Factory Class Reference

Class for creating [ProbabilisticBlackBoxModelChecker](#) instances.

```
#include <ProbabilisticBlackBoxModelCheckerFactory.hpp>
```

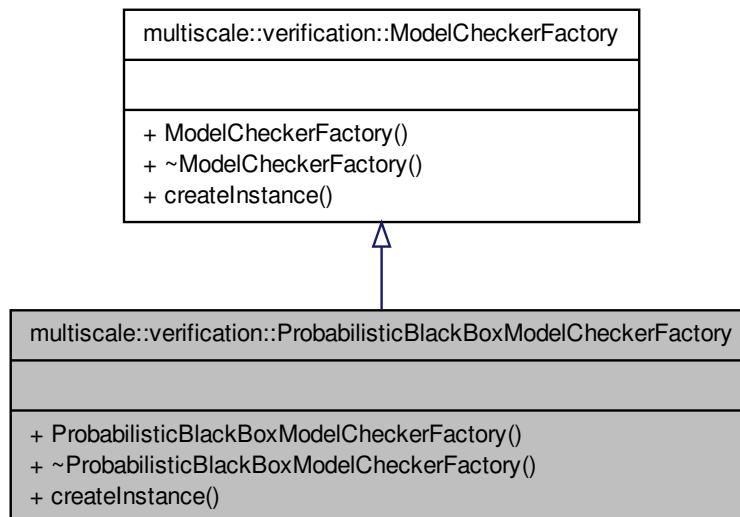
Inheritance diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker-

Factory:



Collaboration diagram for multiscale::verification::ProbabilisticBlackBoxModelChecker-

Factory:



Public Member Functions

- `ProbabilisticBlackBoxModelCheckerFactory ()`
- `~ProbabilisticBlackBoxModelCheckerFactory ()`
- `std::shared_ptr< ModelChecker > createInstance (const AbstractSyntaxTree &abstractSyntaxTree) override`

Create an instance of `ProbabilisticBlackBoxModelChecker`.

7.118.1 Detailed Description

Class for creating `ProbabilisticBlackBoxModelChecker` instances.

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

7.118.2 Constructor & Destructor Documentation

7.118.2.1 `ProbabilisticBlackBoxModelCheckerFactory::ProbabilisticBlackBoxModelCheckerFactory()`

Definition at line 7 of file `ProbabilisticBlackBoxModelCheckerFactory.cpp`.

7.118.2.2 ProbabilisticBlackBoxModelCheckerFactory::~ProbabilisticBlackBoxModelCheckerFactory()

Definition at line 9 of file ProbabilisticBlackBoxModelCheckerFactory.cpp.

7.118.3 Member Function Documentation

7.118.3.1 std::shared_ptr< ModelChecker > ProbabilisticBlackBoxModelCheckerFactory::createInstance(const AbstractSyntaxTree & abstractSyntaxTree) [override, virtual]

Create an instance of [ProbabilisticBlackBoxModelChecker](#).

Parameters

<i>abstract-SyntaxTree</i>	The abstract syntax tree representing the logic property to be checked
----------------------------	--

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](#)

7.119 multiscaletest::ProbabilisticBlackBoxModelCheckerTest - Class Reference

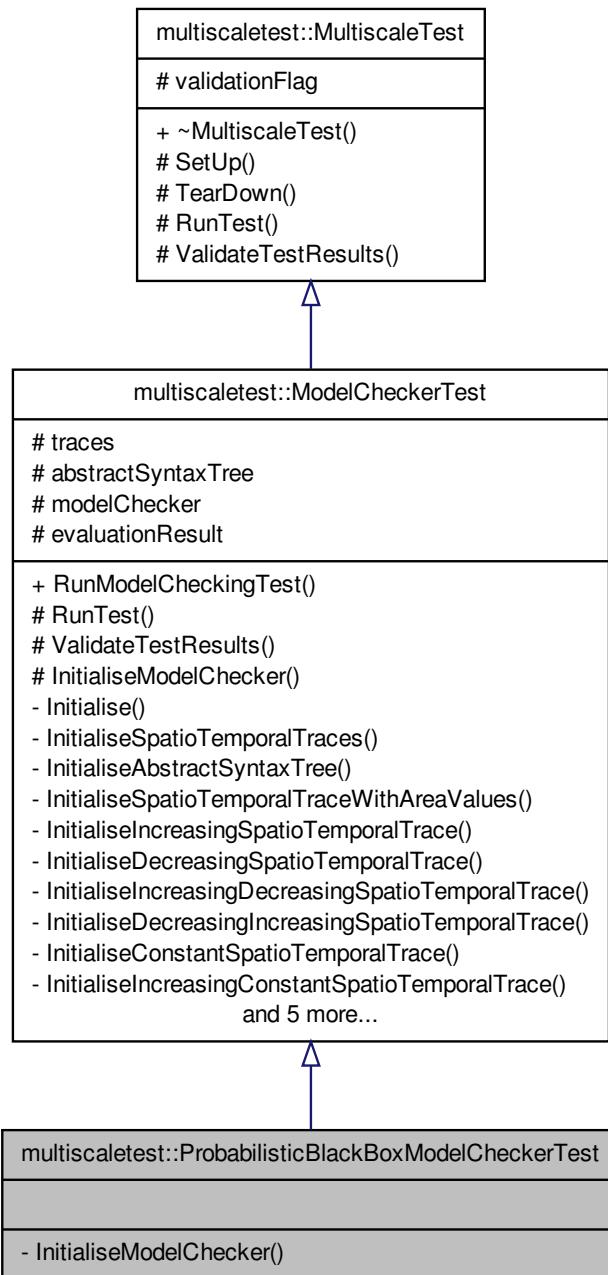
Class for testing the probabilistic black-box model checker.

```
#include <ProbabilisticBlackBoxModelCheckerTest.hpp>
```

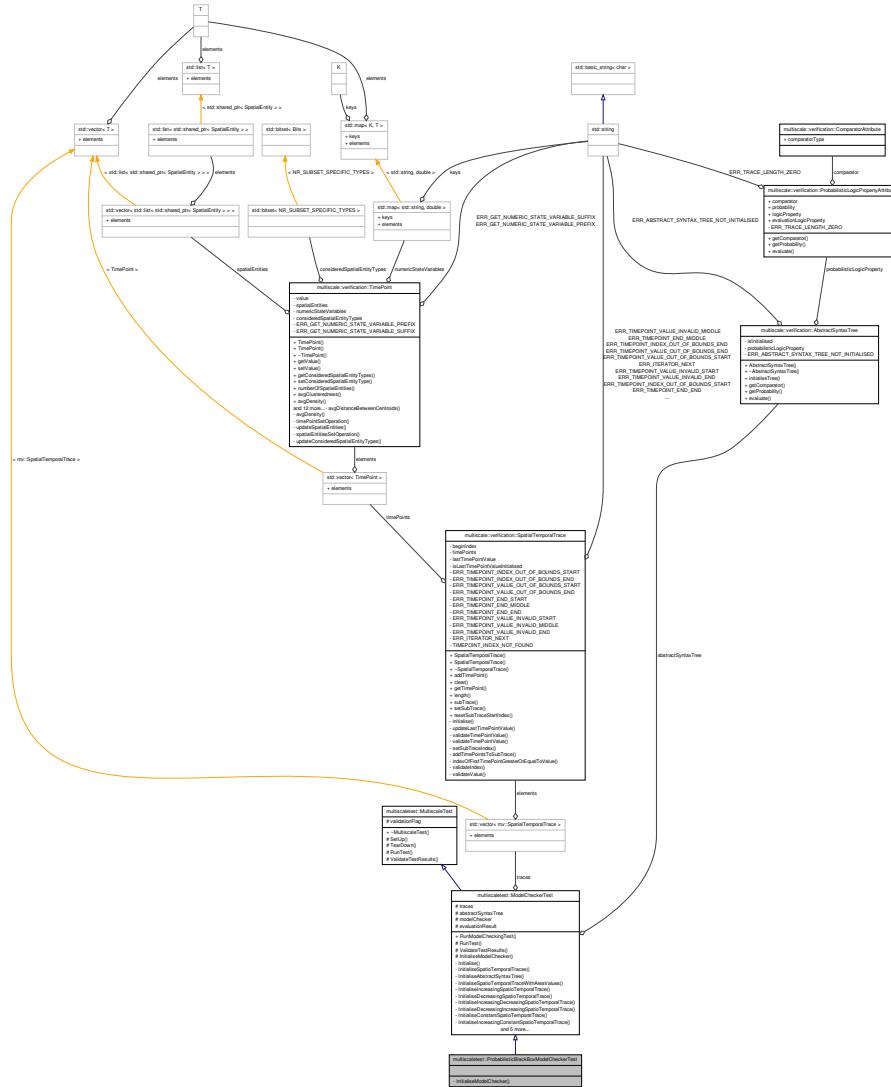
7.119 multiscaletest::ProbabilisticBlackBoxModelCheckerTest Class Reference

207

Inheritance diagram for multiscaletest::ProbabilisticBlackBoxModelCheckerTest:



Collaboration diagram for multiscaletest::ProbabilisticBlackBoxModelCheckerTest:



Private Member Functions

- void InitialiseModelChecker () override

Initialise the model checker.

7.119.1 Detailed Description

Class for testing the probabilistic black-box model checker.

Definition at line 15 of file ProbabilisticBlackBoxModelCheckerTest.hpp.

7.119.2 Member Function Documentation

**7.119.2.1 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:

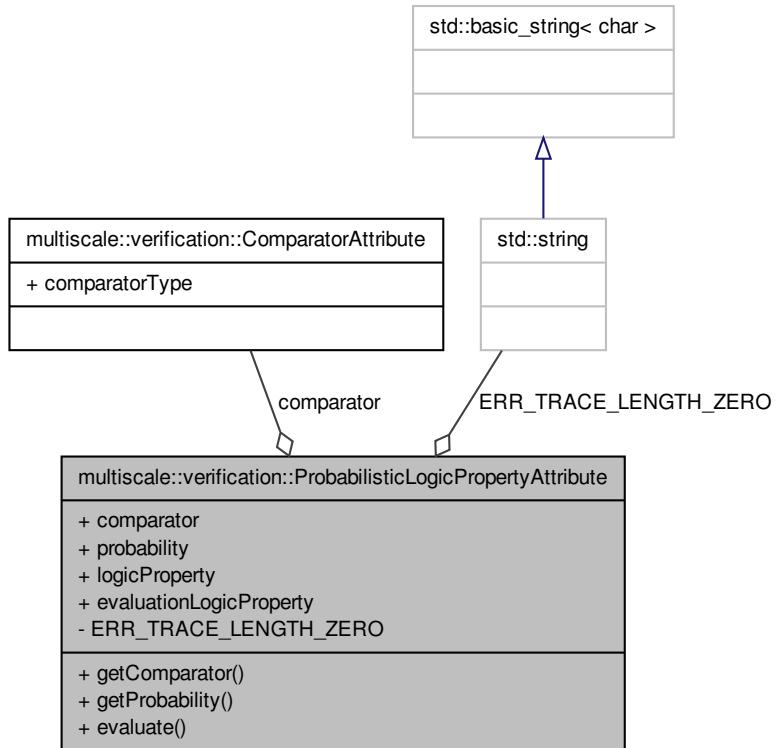
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/[ProbabilisticBlackBoxModelCheckerTest.hpp](#)

7.120 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)`
Evaluate the truth value of the logic property considering the given spatial temporal trace.

Public Attributes

- `ComparatorAttribute 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."

7.120.1 Detailed Description

Class for representing a probabilistic logic property attribute.

Definition at line 18 of file ProbabilisticLogicPropertyAttribute.hpp.

7.120.2 Member Function Documentation

7.120.2.1 bool ProbabilisticLogicPropertyAttribute::evaluate (const SpatialTemporalTrace & trace)

Evaluate the truth value of the logic property considering the given spatial temporal trace.

Parameters

<code>trace</code>	The spatial temporal trace
--------------------	----------------------------

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()`.

7.120.2.2 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()`.

7.120.2.3 double ProbabilisticLogicPropertyAttribute::getProbability ()

Get the probability.

Definition at line 11 of file ProbabilisticLogicPropertyAttribute.cpp.

References probability.

Referenced by multiscale::verification::AbstractSyntaxTree::getProbability().

7.120.3 Member Data Documentation

7.120.3.1 ComparatorAttribute multiscale::verification::ProbabilisticLogicPropertyAttribute::comparator

The comparator

Definition at line 22 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by getComparator().

7.120.3.2 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 46 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.120.3.3 LogicPropertyAttributeType multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluationLogicProperty

The logic property used only for evaluation purposes

Definition at line 26 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.120.3.4 LogicPropertyAttributeType multiscale::verification::ProbabilisticLogicPropertyAttribute::logicProperty

The logic property

Definition at line 24 of file ProbabilisticLogicPropertyAttribute.hpp.

Referenced by evaluate().

7.120.3.5 double multiscale::verification::ProbabilisticLogicPropertyAttribute::probability

The probability

Definition at line 23 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`

7.121 multiscale::verification::ProbabilityErrorHandler Struct Reference

Structure for defining the error handler for invalid probability errors.

```
#include <ProbabilityErrorHandler.hpp>
```

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.

7.121.1 Detailed Description

Structure for defining the error handler for invalid probability errors.

Definition at line 17 of file `ProbabilityErrorHandler.hpp`.

7.121.2 Member Function Documentation

7.121.2.1 `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

<code><i>expected- Token</i></code>	The expected token (not a string)
---	-----------------------------------

Definition at line 46 of file ProbabilityErrorHandler.hpp.

Referenced by operator()().

7.121.2.2 `template<typename Iterator > void multiscale::verification::ProbabilityErrorHandler::operator() (qi::info const & expectedToken, Iterator errorPosition, Iterator last) const [inline]`

Overloaded operator.

Parameters

<code><i>expected- Token</i></code>	The expected token
<code><i>errorPosition</i></code>	Iterator pointing to the error position
<code><i>last</i></code>	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:

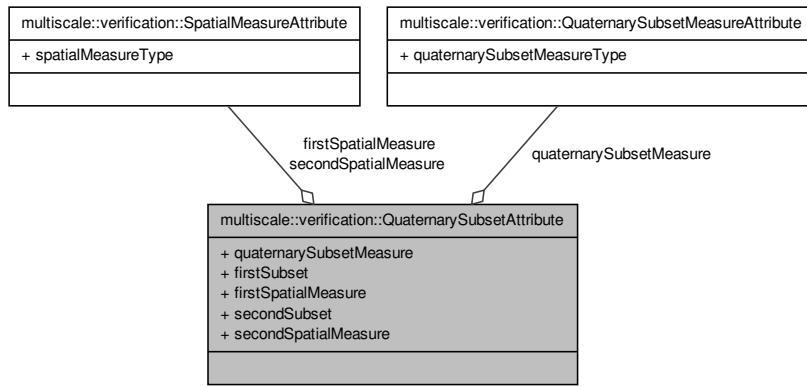
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/[ProbabilityErrorHandler.hpp](#)

7.122 `multiscale::verification::QuaternarySubsetAttribute` Class Reference

Class for representing a quaternary subset attribute.

```
#include <QuaternarySubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::QuaternarySubsetAttribute:



Public Attributes

- `QuaternarySubsetMeasureAttribute quaternarySubsetMeasure`
- `SubsetAttributeType firstSubset`
- `SpatialMeasureAttribute firstSpatialMeasure`
- `SubsetAttributeType secondSubset`
- `SpatialMeasureAttribute secondSpatialMeasure`

7.122.1 Detailed Description

Class for representing a quaternary subset attribute.

Definition at line 16 of file QuaternarySubsetAttribute.hpp.

7.122.2 Member Data Documentation

7.122.2.1 SpatialMeasureAttribute multiscale::verification::QuaternarySubsetAttribute::firstSpatialMeasure

The first considered spatial measure

Definition at line 22 of file QuaternarySubsetAttribute.hpp.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

7.122.2.2 SubsetAttributeType multiscale::verification::QuaternarySubsetAttribute::firstSubset

The first considered subset

Definition at line 21 of file QuaternarySubsetAttribute.hpp.

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

7.122.2.3 QuaternarySubsetMeasureAttribute multiscale::verification::QuaternarySubsetAttribute::quaternarySubsetMeasure

The quaternary subset measure

Definition at line 20 of file QuaternarySubsetAttribute.hpp.

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

7.122.2.4 SpatialMeasureAttribute multiscale::verification::QuaternarySubsetAttribute::secondSpatialMeasure

The second considered spatial measure

Definition at line 24 of file QuaternarySubsetAttribute.hpp.

7.122.2.5 SubsetAttributeType multiscale::verification::QuaternarySubsetAttribute::secondSubset

The second considered subset

Definition at line 23 of file QuaternarySubsetAttribute.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/[QuaternarySubsetAttribute.hpp](#)

7.123 multiscale::verification::QuaternarySubsetMeasureAttribute Class Reference

Class for representing a quaternary subset measure attribute.

```
#include <QuaternarySubsetMeasureAttribute.hpp>
```

Public Attributes

- [QuaternarySubsetMeasureType](#) `quaternarySubsetMeasureType`

7.123.1 Detailed Description

Class for representing a quaternary subset measure attribute.

Definition at line 27 of file QuaternarySubsetMeasureAttribute.hpp.

7.123.2 Member Data Documentation

7.123.2.1 [QuaternarySubsetMeasureType](#) `multiscale::verification::QuaternarySubsetMeasureAttribute::quaternarySubsetMeasureType`

The quaternary subset measure type

Definition at line 31 of file QuaternarySubsetMeasureAttribute.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/QuaternarySubsetMeasureAttribute.hpp](#)

7.124 [multiscale::verification::QuaternarySubsetMeasureTypeParser](#) Struct Reference

Symbol table and parser for the quaternary subset measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [QuaternarySubsetMeasureTypeParser \(\)](#)

7.124.1 Detailed Description

Symbol table and parser for the quaternary subset measure type.

Definition at line 132 of file SymbolTables.hpp.

7.124.2 Constructor & Destructor Documentation

7.124.2.1 **multiscale::verification::QuaternarySubsetMeasureTypeParser::QuaternarySubsetMeasureTypeParser()**
[inline]

Definition at line 134 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](#)

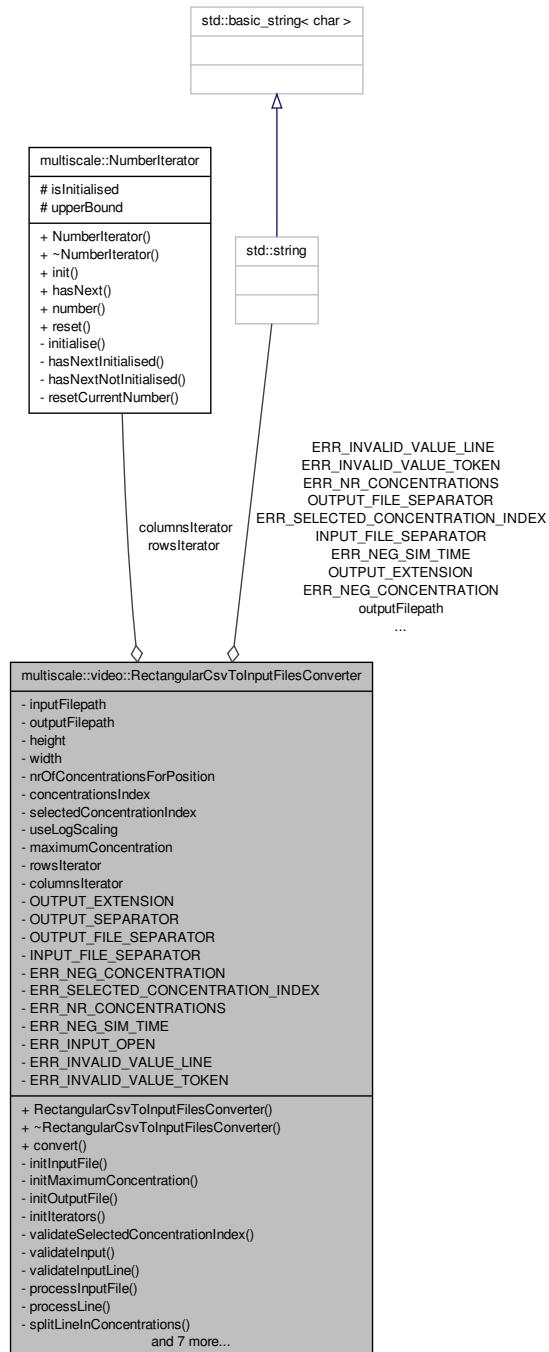
7.125 multiscale::video::RectangularCsvToInputFilesConverter - Class Reference

Csv file to input file converter considering cartesian coordinates.

```
#include <RectangularCsvToInputFilesConverter.hpp>
```

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

Collaboration diagram for multiscale::video::RectangularCsvToInputFilesConverter:



Public Member Functions

- `RectangularCsvToInputFilesConverter` (const string &`inputfilepath`, const 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` (ifstream &`fin`)

Initialise the input file stream over the given input file.
- void `initMaximumConcentration` (ifstream &`fin`)

Compute the value of member maximum concentration.
- void `initOutputFile` (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` (ifstream &`fin`)

Validate the input.
- void `validateInputLine` (const string &`line`, unsigned int `lineNumber`)

Validate the provided line identified by a line number.
- void `processInputFile` (ifstream &`fin`)

Process the input file.
- void `processLine` (const string &`line`, unsigned int `outputIndex`)

Process the provided line.
- vector< double > `splitLineInConcentrations` (const string &`line`, double &`simulationTime`)

Split the line in concentrations.
- void `splitLineInConcentrations` (vector< double > &`concentrations`, vector< string > &`tokens`, unsigned int `rowIndex`)

Split line into concentrations.
- double `computeSimulationTime` (const string &`token`)

Compute the simulation time from the given token and check if it is valid.
- double `computeNextPositionConcentration` (int `concentrationIndex`, vector< string > &`tokens`)

Compute the concentration for the next position.
- double `computeConcentration` (const string &`concentration`)

Compute the concentration from the given string.

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

- double `computeNonScaledConcentration` (const string &concentration)
Compute the non-scaled concentration from the given string.
- double `computeScaledConcentration` (const string &concentration)
Compute the scaled concentration from the given string.
- double `computeNormalisedConcentration` (double concentration)
Normalise the given concentration by dividing it to the maximum concentration.
- void `updateMaximumConcentration` (const string &line, double &`maximumConcentration`)
Update the maximum concentration if the values from the given line are greater than it.

Private Attributes

- string `inputFilepath`
- 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 string `OUTPUT_EXTENSION` = ".in"
- static const string `OUTPUT_SEPARATOR` = " "
- static const string `OUTPUT_FILE_SEPARATOR` = "_"
- static const string `INPUT_FILE_SEPARATOR` = ","
- static const string `ERR_NEG_CONCENTRATION` = "All concentrations must be non-negative."
- static const string `ERR_SELECTED_CONCENTRATION_INDEX` = "The selected concentration index (0-based indexing) should be smaller than the number of concentrations."
- static const 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 string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const string `ERR_INVALID_VALUE_LINE` = "Invalid value on line: "
- static const string `ERR_INVALID_VALUE_TOKEN` = ", value: "

7.125.1 Detailed Description

Csv file to input file converter considering cartesian coordinates.

Definition at line 18 of file RectangularCsvToInputFilesConverter.hpp.

7.125.2 Constructor & Destructor Documentation

**7.125.2.1 RectangularCsvToInputFilesConverter::RectangularCsvToInputFiles-
Converter (const string & *inputFilepath*, const string & *outputFilepath*, unsigned
int *height*, unsigned int *width*, unsigned int *nrOfConcentrationsForPosition*, unsigned
int *selectedConcentrationIndex*, bool *useLogScaling*, NumberIteratorType
numberIteratorType)**

Definition at line 20 of file RectangularCsvToInputFilesConverter.cpp.

**7.125.2.2 RectangularCsvToInputFilesConverter::~RectangularCsvToInputFiles-
Converter ()**

Definition at line 44 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3 Member Function Documentation

7.125.3.1 double RectangularCsvToInputFilesConverter::computeConcentration (const string & *concentration*) [inline, private]

Compute the concentration from the given string.

Parameters

<i>concentra- tion</i>	String representing the concentration
----------------------------	---------------------------------------

Definition at line 283 of file RectangularCsvToInputFilesConverter.cpp.

**7.125.3.2 double RectangularCsvToInputFilesConverter::computeNextPosition-
Concentration (int *concentrationIndex*, vector< string > & *tokens*) [inline, private]**

Compute the concentration for the next position.

Parameters

<i>concentration- Index</i>	Index of the current concentration from the vector of tokens
<i>tokens</i>	Vector of tokens

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

Definition at line 256 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.3 double RectangularCsvToInputFilesConverter::computeNon-ScaledConcentration (const string & *concentration*) [inline, private]

Compute the non-scaled concentration from the given string.

Parameters

<i>concentra-tion</i>	String representing the concentration
-----------------------	---------------------------------------

Definition at line 289 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.4 double RectangularCsvToInputFilesConverter::compute-NormalisedConcentration (double *concentration*) [inline, private]

Normalise the given concentration by dividing it to the maximum concentration.

Parameters

<i>concentra-tion</i>	The concentration
-----------------------	-------------------

Definition at line 305 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.5 double RectangularCsvToInputFilesConverter::computeScaled-Concentration (const string & *concentration*) [inline, private]

Compute the scaled concentration from the given string.

Compute the scaled concentration from the given string by applying a logit transformation to it

Parameters

<i>concentra-tion</i>	String representing the concentration
-----------------------	---------------------------------------

Definition at line 293 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.6 double RectangularCsvToInputFilesConverter::computeSimulationTime (const string & *token*) [inline, private]

Compute the simulation time from the given token and check if it is valid.

Parameters

<i>token</i>	Token (string)
--------------	----------------

Definition at line 246 of file RectangularCsvToInputFilesConverter.cpp.

References MS_throw.

7.125.3.7 void RectangularCsvToInputFilesConverter::convert()

Start the conversion.

Definition at line 49 of file RectangularCsvToInputFilesConverter.cpp.

Referenced by main().

7.125.3.8 void RectangularCsvToInputFilesConverter::initInputFile(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 RectangularCsvToInputFilesConverter.cpp.

References MS_throw.

7.125.3.9 void RectangularCsvToInputFilesConverter::initIterators(const NumberIteratorType & numberIteratorType) [private]

Initialise the iterators considering the given number iterator type.

Parameters

<i>number- IteratorType</i>	The type of the number iterator
---------------------------------	---------------------------------

Definition at line 111 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

**7.125.3.10 void RectangularCsvToInputFilesConverter::initMaximum-
Concentration(ifstream & fin) [private]**

Compute the value of member maximum concentration.

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 70 of file RectangularCsvToInputFilesConverter.cpp.

References MS_throw.

7.125.3.11 void RectangularCsvToInputFilesConverter::initOutputFile (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>simulation-Time</i>	Simulation time

Definition at line 94 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

7.125.3.12 void RectangularCsvToInputFilesConverter::processInputFile (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 178 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.13 void RectangularCsvToInputFilesConverter::processLine (const 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 193 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.14 `vector< double > RectangularCsvToInputFilesConverter::splitLineInConcentrations (const string & line, double & simulationTime) [private]`

Split the line in concentrations.

Parameters

<i>line</i>	Line
<i>simulation-Time</i>	Simulation time associated with the line

Definition at line 210 of file RectangularCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

7.125.3.15 `void RectangularCsvToInputFilesConverter::splitLineInConcentrations (vector< double > & concentrations, vector< 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 231 of file RectangularCsvToInputFilesConverter.cpp.

7.125.3.16 `void RectangularCsvToInputFilesConverter::updateMaximumConcentration (const 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>maximum-Concentration</i>	The maximum concentration

Definition at line 309 of file RectangularCsvToInputFilesConverter.cpp.

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

7.125.3.17 void RectangularCsvToInputFilesConverter::validateInput (ifstream & *fin*) [private]

Validate the input.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 134 of file RectangularCsvToInputFilesConverter.cpp.

References MS_throw.

7.125.3.18 void RectangularCsvToInputFilesConverter::validateInputLine (const 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().

7.125.3.19 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.

7.125.4 Member Data Documentation

7.125.4.1 NumberIterator* multiscale::video::RectangularCsvToInputFilesConverter::columnsIterator [private]

Iterator over the number of columns

Definition at line 43 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.2 **unsigned int multiscale::video::RectangularCsvToInputFilesConverter::concentrationsIndex** [private]

Index of the current concentration

Definition at line 29 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.3 **const string RectangularCsvToInputFilesConverter::ERR_INPUT_OPEN = "The input file could not be opened."** [static, private]

Definition at line 197 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.4 **const string RectangularCsvToInputFilesConverter::ERR_I-NVALID_VALUE_LINE = "Invalid value on line: "** [static, private]

Definition at line 198 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.5 **const string RectangularCsvToInputFilesConverter::E-RR_INVALID_VALUE_TOKEN = ", value: "** [static, private]

Definition at line 199 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.6 **const string RectangularCsvToInputFilesConverter::ERR_NEG_CONC-ENTRATION = "All concentrations must be non-negative."** [static, private]

Definition at line 193 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.7 **const string RectangularCsvToInputFilesConverter::ERR_NEG_SIM_TIME = "The simulation time must be non-negative."** [static, private]

Definition at line 196 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.8 **const 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 195 of file RectangularCsvToInputFilesConverter.hpp.

7.125 multiscale::video::RectangularCsvToInputFilesConverter Class Reference

7.125.4.9 `const string RectangularCsvToInputFilesConverter::ERR_SELECTED_C-
ONCENTRATION_INDEX = "The selected concentration index (0-based indexing)
should be smaller than the number of concentrations." [static, private]`

Definition at line 194 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.10 `unsigned int multiscale::video::RectangularCsvToInputFilesConverter-
::height [private]`

Height of the grid

Definition at line 25 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.11 `const string RectangularCsvToInputFilesConverter-
::INPUT_FILE_SEPARATOR = "," [static,
private]`

Definition at line 191 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.12 `string multiscale::video::RectangularCsvToInputFilesConverter::input-
Filepath [private]`

Path to the input file

Definition at line 22 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.13 `double multiscale::video::RectangularCsvToInputFilesConverter-
::maximumConcentration [private]`

The maximum concentration in the input file

Definition at line 40 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.14 `unsigned int multiscale::video::RectangularCsvTo-
InputFilesConverter::nrOfConcentrationsForPosition
[private]`

Number of concentrations for each position

Definition at line 27 of file RectangularCsvToInputFilesConverter.hpp.

7.125.4.15 `const string RectangularCsvToInputFilesConverter::OUTPUT_EXTENSI-
ON = ".in" [static, private]`

Definition at line 188 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.16 const string RectangularCsvToInputFilesConverter-  
          ::OUTPUT_FILE_SEPARATOR = "_" [static,  
          private]
```

Definition at line 190 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.17 const string RectangularCsvToInputFilesConverter::OUTPUT_SEPARA-  
          TOR = "" [static, private]
```

Definition at line 189 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.18 string multiscale::video::RectangularCsvToInputFilesConverter-  
          ::outputFilepath [private]
```

Path to the output file

Definition at line 23 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.19 NumberIterator* multiscale::video::RectangularCsvToInputFiles-  
          Converter::rowsIterator [private]
```

Iterator over the number of rows

Definition at line 42 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.20 unsigned int multiscale::video::RectangularCsvTo-  
          InputFilesConverter::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 31 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.21 bool multiscale::video::RectangularCsvToInputFilesConverter::use-  
          LogScaling [private]
```

Flag for using logarithmic scaling for concentrations or not

Definition at line 38 of file RectangularCsvToInputFilesConverter.hpp.

```
7.125.4.22 unsigned int multiscale::video::RectangularCsvToInputFilesConverter-  
          ::width [private]
```

Width of the grid

Definition at line 26 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](#)

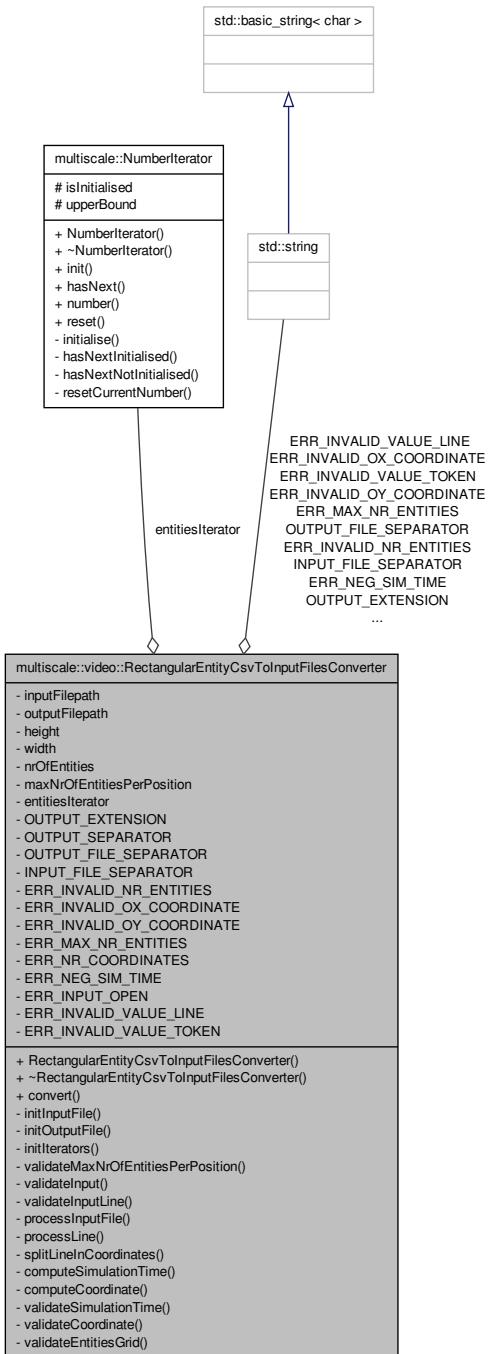
7.126 multiscale::video::RectangularEntityCsvToInputFilesConverter Class Reference

Csv entity file to input file converter considering cartesian coordinates.

```
#include <RectangularEntityCsvToInputFilesConverter.hpp>
```

Collaboration diagram for multiscale::video::RectangularEntityCsvToInputFiles-

Converter:



Public Member Functions

- `RectangularEntityCsvToInputFilesConverter` (const string &`filepath`, const 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` (ifstream &`fin`)

Initialise the input file stream over the given input file.
- `void initOutputFile` (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` (ifstream &`fin`)

Validate the input.
- `void validateInputLine` (const string &`line`, unsigned int `lineNumber`)

Validate the provided line identified by a line number.
- `void processInputFile` (ifstream &`fin`)

Process the input file.
- `void processLine` (const string &`line`, unsigned int `outputIndex`)

Process the provided line.
- `vector< double > splitLineInCoordinates` (const string &`line`, double &`simulationTime`)

*Split the line in coordinates and return the grid of size height * width showing the position of the entities.*
- `double computeSimulationTime` (const string &`token`)

Compute the simulation time from the given token and check if it is valid.
- `unsigned int computeCoordinate` (const string &`token`, bool `isOxCoordinate`)

Compute the coordinate from the given string and check if it is valid.
- `void validateSimulationTime` (const string &`token`, unsigned int `lineNumber`)

Check if the simulation time is valid.
- `void validateCoordinate` (const string &`token`, unsigned int `lineNumber`, bool `isOxCoordinate`)

Check if the coordinate is valid.
- `void validateEntitiesGrid` (const vector< double > &`entitiesGrid`)

Check if the entities grid contains only values between zero and one.

Private Attributes

- string `inputfilepath`
- string `outputfilepath`
- unsigned int `height`
- unsigned int `width`
- unsigned int `nrOfEntities`
- unsigned int `maxNrOfEntitiesPerPosition`
- `NumberIterator * entitiesIterator`

Static Private Attributes

- static const string `OUTPUT_EXTENSION` = ".in"
- static const string `OUTPUT_SEPARATOR` = " "
- static const string `OUTPUT_FILE_SEPARATOR` = "_"
- static const string `INPUT_FILE_SEPARATOR` = ","
- static const string `ERR_INVALID_NR_ENTITIES` = "The number of entities at the given position is invalid."
- static const string `ERR_INVALID_OX_COORDINATE` = "The value of the Ox coordinate is invalid."
- static const string `ERR_INVALID_OY_COORDINATE` = "The value of the Oy coordinate is invalid."
- static const string `ERR_MAX_NR_ENTITIES` = "The maximum number of entities per grid position is equal to zero."
- static const 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 string `ERR_NEG_SIM_TIME` = "The simulation time must be non-negative."
- static const string `ERR_INPUT_OPEN` = "The input file could not be opened."
- static const string `ERR_INVALID_VALUE_LINE` = "Invalid value on line: "
- static const string `ERR_INVALID_VALUE_TOKEN` = ", value: "

7.126.1 Detailed Description

Csv entity file to input file converter considering cartesian coordinates.

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

7.126.2 Constructor & Destructor Documentation

7.126.2.1 `RectangularEntityCsvToInputFilesConverter::RectangularEntityCsvToInputFilesConverter (const string & inputfilepath, const string & outputfilepath, unsigned int height, unsigned int width, unsigned int nrOfEntities, unsigned int maxNrOfEntitiesPerPosition, NumberIteratorType numberIteratorType)`

Definition at line 20 of file `RectangularEntityCsvToInputFilesConverter.cpp`.

7.126.2.2 RectangularEntityCsvToInputFilesConverter::~RectangularEntityCsvToInputFilesConverter()

Definition at line 39 of file RectangularEntityCsvToInputFilesConverter.cpp.

7.126.3 Member Function Documentation

7.126.3.1 unsigned int RectangularEntityCsvToInputFilesConverter::computeCoordinate(const string & token, bool isOxCoordinate) [inline, private]

Compute the coordinate from the given string and check if it is valid.

Parameters

<i>token</i>	Token (string)
<i>isOxCoordinate</i>	Flag which indicates if the coordinate corresponds to Ox axis or not

Definition at line 209 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.2 double RectangularEntityCsvToInputFilesConverter::computeSimulationTime(const string & token) [inline, private]

Compute the simulation time from the given token and check if it is valid.

Parameters

<i>token</i>	Token (string)
--------------	----------------

Definition at line 199 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.3 void RectangularEntityCsvToInputFilesConverter::convert()

Start the conversion.

Definition at line 43 of file RectangularEntityCsvToInputFilesConverter.cpp.

Referenced by main().

7.126.3.4 void RectangularEntityCsvToInputFilesConverter::initInputFile(ifstream & fin) [private]

Initialise the input file stream over the given input file.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 55 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.5 void RectangularEntityCsvToInputFilesConverter::initIterators (const NumberIteratorType & *numberIteratorType*) [private]

Initialise the iterators considering the given number iterator type.

Parameters

<i>number- IteratorType</i>	The type of the number iterator
---------------------------------	---------------------------------

Definition at line 80 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::LEXICOGRAPHIC, and multiscale::STANDARD.

7.126.3.6 void RectangularEntityCsvToInputFilesConverter::initOutputFile (ostream & *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>simulation- Time</i>	Simulation time

Definition at line 63 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::toString().

7.126.3.7 void RectangularEntityCsvToInputFilesConverter::processInputFile (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 143 of file RectangularEntityCsvToInputFilesConverter.cpp.

7.126.3.8 void RectangularEntityCsvToInputFilesConverter::processLine (const 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 158 of file RectangularEntityCsvToInputFilesConverter.cpp.

7.126.3.9 vector< double > RectangularEntityCsvToInputFilesConverter::splitLineInCoordinates (const string & *line*, double & *simulationTime*) [private]

Split the line in coordinates and return the grid of size height * width showing 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>simulation-Time</i>	Simulation time associated with the line

Definition at line 177 of file RectangularEntityCsvToInputFilesConverter.cpp.

References multiscale::StringManipulator::split().

7.126.3.10 void RectangularEntityCsvToInputFilesConverter::validateCoordinate (const string & *token*, unsigned int *lineNumber*, bool *isOxCoordinate*) [inline, private]

Check if the coordinate is valid.

Parameters

<i>token</i>	Token (string)
<i>lineNumber</i>	Number of the line
<i>isOx-Coordinate</i>	Flag which indicates if the coordinate corresponds to Ox axis or not

Definition at line 238 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.11 void RectangularEntityCsvToInputFilesConverter::validateEntitiesGrid (const vector< double > & entitiesGrid) [inline, private]

Check if the entities grid contains only values between zero and one.

Parameters

<i>entitiesGrid</i>	The grid of entities
---------------------	----------------------

Definition at line 252 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.12 void RectangularEntityCsvToInputFilesConverter::validateInput (ifstream & fin) [private]

Validate the input.

Parameters

<i>fin</i>	Input file stream
------------	-------------------

Definition at line 102 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.3.13 void RectangularEntityCsvToInputFilesConverter::validateInputLine (const 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 126 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw, and multiscale::StringManipulator::split().

7.126.3.14 void RectangularEntityCsvToInputFilesConverter::validateMaxNrOfEntitiesPerPosition () [private]

Check if the maximum number of entities per position is a non-zero natural number.

Definition at line 96 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

```
7.126.3.15 void RectangularEntityCsvToInputFilesConverter::validateSimulation-
    Time ( const string & token, unsigned int lineNumber ) [inline,
    private]
```

Check if the simulation time is valid.

Parameters

<i>token</i>	Token (string)
<i>lineNumber</i>	Number of the line

Definition at line 225 of file RectangularEntityCsvToInputFilesConverter.cpp.

References MS_throw.

7.126.4 Member Data Documentation

```
7.126.4.1 NumberIterator* multiscale::video::Rectangular-
    EntityCsvToInputFilesConverter::entitiesIterator
    [private]
```

Iterator over the number of rows

Definition at line 31 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.2 const string RectangularEntityCsvToInputFilesConverter::ERR-
    _INPUT_OPEN = "The input file could not be opened." [static,
    private]
```

Definition at line 158 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.3 const string RectangularEntityCsvToInputFilesConverter::ERR_INVA-
    LID_NR_ENTITIES = "The number of entities at the given position is invalid."
    [static, private]
```

Definition at line 152 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.4 const string RectangularEntityCsvToInputFilesConverter::ERR_INVALID-
    _OX_COORDINATE = "The value of the Ox coordinate is invalid." [static,
    private]
```

Definition at line 153 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.5 const string RectangularEntityCsvToInputFilesConverter::ERR_INVALID_
_OY_COORDINATE = "The value of the Oy coordinate is invalid." [static,
private]
```

Definition at line 154 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.6 const string RectangularEntityCsvToInputFilesConverter::ER-
R_INVALID_VALUE_LINE = "Invalid value on line: " [static,
private]
```

Definition at line 159 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.7 const string RectangularEntityCsvToInputFilesConverter-
::ERR_INVALID_VALUE_TOKEN = ", value: " [static,
private]
```

Definition at line 160 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.8 const string RectangularEntityCsvToInputFilesConverter::ERR_MAX_NR-
_ENTITIES = "The maximum number of entities per grid position is equal to zero."
[static, private]
```

Definition at line 155 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.9 const string RectangularEntityCsvToInputFilesConverter::ERR_NEG-
_SIM_TIME = "The simulation time must be non-negative." [static,
private]
```

Definition at line 157 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.10 const string RectangularEntityCsvToInputFilesConverter::ERR_NR_CO-
ORDINATES = "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 156 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.11 unsigned int multiscale::video::RectangularEntityCsvToInputFiles-
Converter::height [private]
```

Height of the grid

Definition at line 25 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.12 const string RectangularEntityCsvToInputFiles-
    Converter::INPUT_FILE_SEPARATOR = "," [static,
    private]
```

Definition at line 150 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.13 string multiscale::video::RectangularEntityCsvToInputFilesConverter-
    ::filepath [private]
```

Path to the input file

Definition at line 22 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.14 unsigned int multiscale::video::RectangularEntityCsv-
    ToInputFilesConverter::maxNrOfEntitiesPerPosition
    [private]
```

The maximum number of entities per position

Definition at line 29 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.15 unsigned int multiscale::video::RectangularEntityCsvToInputFiles-
    Converter::nrOfEntities [private]
```

Number of entities

Definition at line 27 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.16 const string RectangularEntityCsvToInputFiles-
    Converter::OUTPUT_EXTENSION = ".in" [static,
    private]
```

Definition at line 147 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.17 const string RectangularEntityCsvToInputFilesConverter-
    ::OUTPUT_FILE_SEPARATOR = "_" [static,
    private]
```

Definition at line 149 of file RectangularEntityCsvToInputFilesConverter.hpp.

```
7.126.4.18 const string RectangularEntityCsvToInputFiles-
    Converter::OUTPUT_SEPARATOR = " " [static,
    private]
```

Definition at line 148 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.126.4.19 `string multiscale::video::RectangularEntityCsvToInputFilesConverter-
::outputfilepath [private]`

Path to the output file

Definition at line 23 of file RectangularEntityCsvToInputFilesConverter.hpp.

7.126.4.20 `unsigned int multiscale::video::RectangularEntityCsvToInputFiles-
Converter::width [private]`

Width of the grid

Definition at line 26 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/rec
[RectangularEntityCsvToInputFilesConverter.hpp](#)

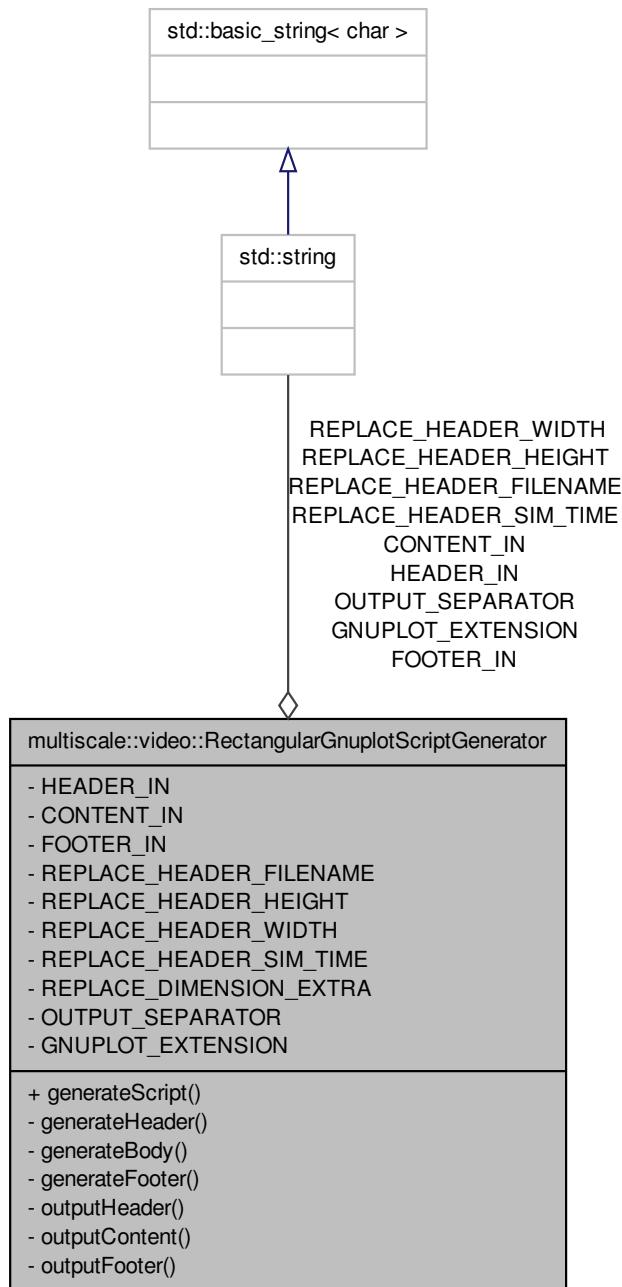
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-
[RectangularEntityCsvToInputFilesConverter.cpp](#)

7.127 `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 vector< double > &concentrations, double simulationTime, unsigned long height, unsigned long width, const string &outputFilepath)

Generate the script.

Static Private Member Functions

- static void `generateHeader` (ofstream &fout, const string &outputFilepath, double simulationTime, unsigned long height, unsigned long width)

Generate the header of the script.

- static void `generateBody` (const vector< double > &concentrations, unsigned long height, unsigned long width, ofstream &fout)

Generate the body/content of the script.

- static void `generateFooter` (ofstream &fout)

Generate the footer of the script.

- static void `outputHeader` (ifstream &fin, const string &outputFilename, double simulationTime, unsigned long height, unsigned long width, ofstream &fout)

Output the header of the script.

- static void `outputContent` (const vector< double > &concentrations, unsigned long height, unsigned long width, ofstream &fout)

Output the content of the script.

- static void `outputFooter` (ifstream &fin, ofstream &fout)

Output the footer of the script.

Static Private Attributes

- static const string `HEADER_IN` = "config/video/rectangular/header.in"
- static const string `CONTENT_IN` = "config/video/rectangular/content.in"
- static const string `FOOTER_IN` = "config/video/rectangular/footer.in"
- static const string `REPLACE_HEADER_FILENAME` = "OUTPUT_FILENAME"
- static const string `REPLACE_HEADER_HEIGHT` = "OUTPUT_DIMENSION1"
- static const string `REPLACE_HEADER_WIDTH` = "OUTPUT_DIMENSION2"
- static const string `REPLACE_HEADER_SIM_TIME` = "OUTPUT_SIM_TIME"
- static const double `REPLACE_DIMENSION_EXTRA` = 0.5
- static const string `OUTPUT_SEPARATOR` = " "
- static const string `GNUPLOT_EXTENSION` = ".plt"

7.127.1 Detailed Description

Gnuplot script generator from the provided concentrations considering a rectangular geometry.

Definition at line 15 of file RectangularGnuplotScriptGenerator.hpp.

7.127.2 Member Function Documentation

7.127.2.1 void **RectangularGnuplotScriptGenerator::generateBody** (const vector< double > & *concentrations*, unsigned long *height*, unsigned long *width*, 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 44 of file RectangularGnuplotScriptGenerator.cpp.

7.127.2.2 void **RectangularGnuplotScriptGenerator::generateFooter** (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.

7.127.2.3 void **RectangularGnuplotScriptGenerator::generateHeader** (ofstream & *fout*, const 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 30 of file RectangularGnuplotScriptGenerator.cpp.

References multiscale::StringManipulator::filenameFromPath().

7.127.2.4 void RectangularGnuplotScriptGenerator::generateScript (const vector< double > & concentrations, double simulationTime, unsigned long height, unsigned long width, const string & outputfilepath) [static]

Generate the script.

Parameters

<i>concentra-</i> <i>tions</i>	Concentrations
<i>simulation-</i> <i>Time</i>	Simulation time
<i>height</i>	Height of the grid
<i>width</i>	Width of the grid
<i>output-</i> <i>Filepath</i>	Path of the output file

Definition at line 14 of file RectangularGnuplotScriptGenerator.cpp.

Referenced by multiscale::video::CartesianToConcentrationsConverter::outputResults().

7.127.2.5 void RectangularGnuplotScriptGenerator::outputContent (const vector< double > & concentrations, unsigned long height, unsigned long width, ofstream & fout) [static, private]

Output the content of the script.

Parameters

<i>concentra-</i> <i>tions</i>	The concentrations
<i>height</i>	The height of the grid
<i>width</i>	The width of the grid
<ifout< i=""></ifout<>	Output file stream

Definition at line 81 of file RectangularGnuplotScriptGenerator.cpp.

7.127.2.6 void RectangularGnuplotScriptGenerator::outputFooter (ifstream & fin, ofstream & fout) [static, private]

Output the footer of the script.

Parameters

<i>fin</i>	Input file stream
<ifout< i=""></ifout<>	Output file stream

Definition at line 94 of file RectangularGnuplotScriptGenerator.cpp.

7.127.2.7 void **RectangularGnuplotScriptGenerator::outputHeader** (ifstream & *fin*, const string & *outputFilename*, double *simulationTime*, unsigned long *height*, unsigned long *width*, 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().

7.127.3 Member Data Documentation

7.127.3.1 const string **RectangularGnuplotScriptGenerator::CONTENT_IN** = "config/video/rectangular/content.in" [static, private]

Definition at line 104 of file RectangularGnuplotScriptGenerator.hpp.

7.127.3.2 const string **RectangularGnuplotScriptGenerator::FOOTER_IN** = "config/video/rectangular//footer.in" [static, private]

Definition at line 105 of file RectangularGnuplotScriptGenerator.hpp.

7.127.3.3 const string **RectangularGnuplotScriptGenerator::GNUPLOT_EXTENSION** = ".plt" [static, private]

Definition at line 116 of file RectangularGnuplotScriptGenerator.hpp.

7.127.3.4 const string **RectangularGnuplotScriptGenerator::HEADER_IN** = "config/video/rectangular/header.in" [static, private]

Definition at line 103 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.5 const string RectangularGnuplotScriptGenerator::OUTPUT_SEPARATOR  
= " " [static, private]
```

Definition at line 114 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.6 const double RectangularGnuplotScriptGenerator::-  
REPLACE_DIMENSION_EXTRA = 0.5 [static,  
private]
```

Definition at line 112 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.7 const string RectangularGnuplotScriptGenerator::REPLAC-  
E_HEADER_FILENAME = "OUTPUT_FILENAME" [static,  
private]
```

Definition at line 107 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.8 const string RectangularGnuplotScriptGenerator::REPLAC-  
E_HEADER_HEIGHT = "OUTPUT_DIMENSION1" [static,  
private]
```

Definition at line 108 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.9 const string RectangularGnuplotScriptGenerator::REPLA-  
CE_HEADER_SIM_TIME = "OUTPUT_SIM_TIME" [static,  
private]
```

Definition at line 110 of file RectangularGnuplotScriptGenerator.hpp.

```
7.127.3.10 const string RectangularGnuplotScriptGenerator::REPLA-  
CE_HEADER_WIDTH = "OUTPUT_DIMENSION2" [static,  
private]
```

Definition at line 109 of file RectangularGnuplotScriptGenerator.hpp.

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

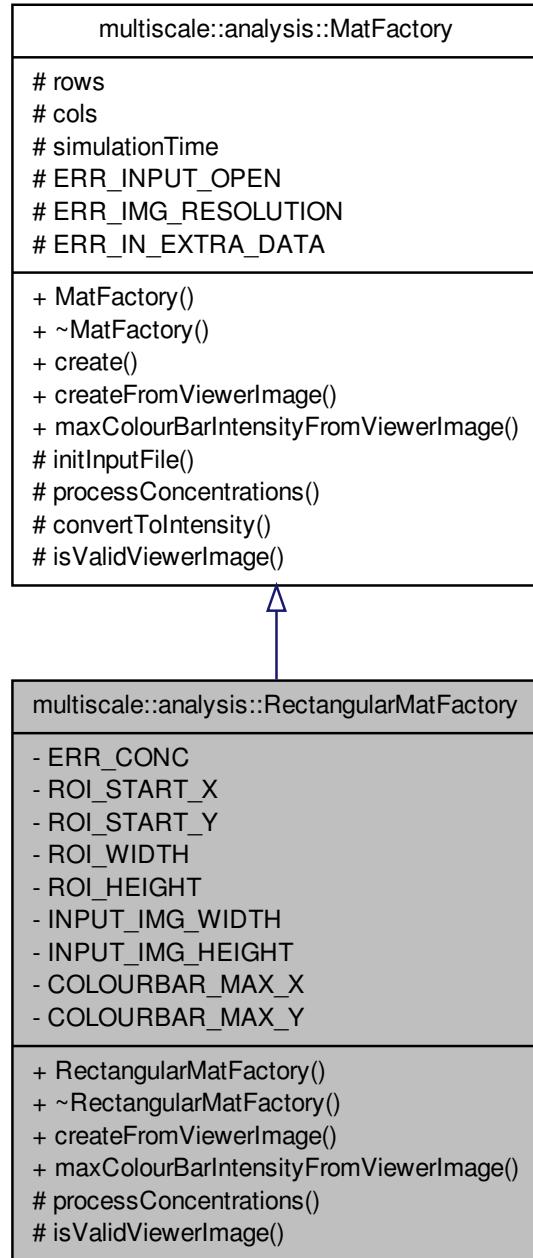
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rec
[RectangularGnuplotScriptGenerator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/
[RectangularGnuplotScriptGenerator.cpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/
[RectangularGnuplotScriptGenerator.in.cpp](#)

7.128 multiscale::analysis::RectangularMatFactory Class Reference

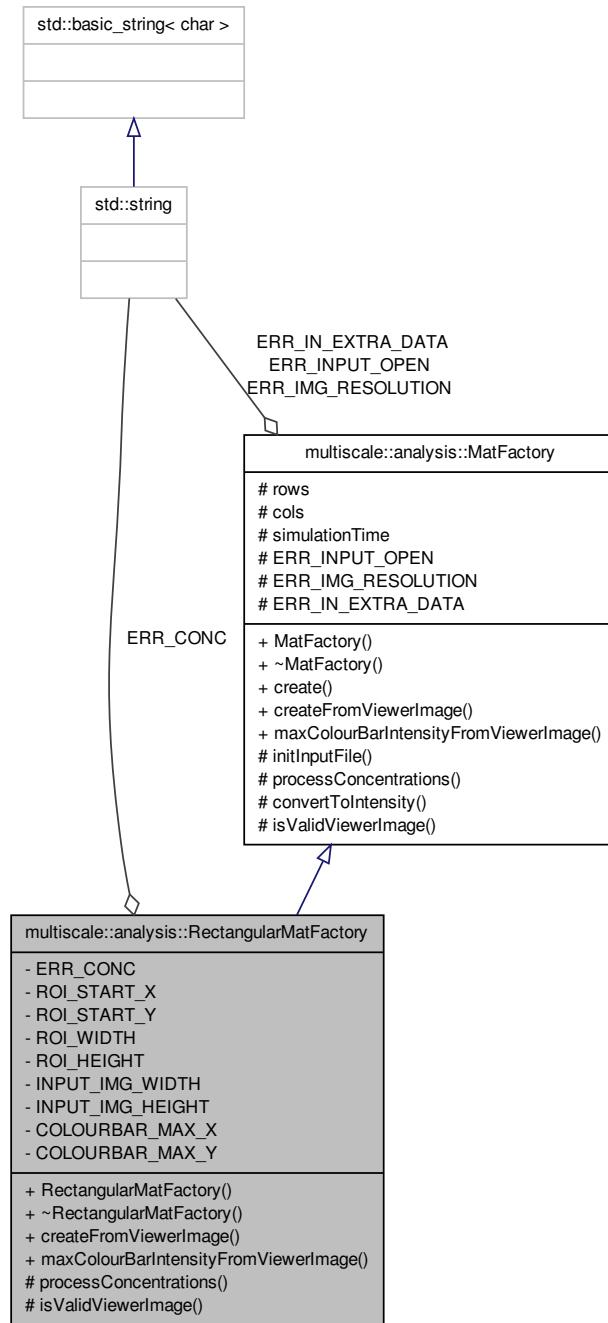
Class for creating a Mat object considering a rectangular grid.

```
#include <RectangularMatFactory.hpp>
```

Inheritance diagram for multiscale::analysis::RectangularMatFactory:



Collaboration diagram for multiscale::analysis::RectangularMatFactory:



Public Member Functions

- `RectangularMatFactory ()`
- `~RectangularMatFactory ()`
- `Mat createFromViewerImage (const string &inputFile) override`
Create a Mat object from the image file obtained from the RectangularGeometry-Viewer.
- `double maxColourBarIntensityFromViewerImage (const string &inputFile) override`
Get the maximum grayscale intensity of the colour bar in the image.

Protected Member Functions

- `unsigned char * processConcentrations (ifstream &fin) override`
Process the concentrations from the input file.
- `bool isValidViewerImage (const Mat &image) override`
Check if the image generated by the viewer has the required resolution.

Static Private Attributes

- `static const 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`

7.128.1 Detailed Description

Class for creating a Mat object considering a rectangular grid.

Definition at line 14 of file RectangularMatFactory.hpp.

7.128.2 Constructor & Destructor Documentation

7.128.2.1 `RectangularMatFactory::RectangularMatFactory()`

Definition at line 9 of file RectangularMatFactory.cpp.

7.128.2.2 RectangularMatFactory::~RectangularMatFactory()

Definition at line 11 of file RectangularMatFactory.cpp.

7.128.3 Member Function Documentation

7.128.3.1 Mat RectangularMatFactory::createFromViewerImage (const string & *inputFile*) [override, virtual]

Create a Mat object from the image file obtained from the RectangularGeometryViewer.

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 13 of file RectangularMatFactory.cpp.

References isValidViewerImage(), ROI_HEIGHT, ROI_START_X, ROI_START_Y, and ROI_WIDTH.

Referenced by main().

7.128.3.2 bool RectangularMatFactory::isValidViewerImage (const 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().

7.128.3.3 double RectangularMatFactory::maxColourBarIntensity-FromViewerImage (const 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().

7.128.3.4 unsigned char * RectangularMatFactory::processConcentrations (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 Mat object from them

REMARK: The constructor of 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_CONC, MS_throw, and multiscale::analysis::MatFactory::rows.

7.128.4 Member Data Documentation

7.128.4.1 const int RectangularMatFactory::COLOURBAR_MAX_X = 1799 [static, private]

Definition at line 68 of file RectangularMatFactory.hpp.

Referenced by maxColourBarIntensityFromViewerImage().

7.128.4.2 const int RectangularMatFactory::COLOURBAR_MAX_Y = 320 [static, private]

Definition at line 69 of file RectangularMatFactory.hpp.

Referenced by maxColourBarIntensityFromViewerImage().

7.128.4.3 **const string RectangularMatFactory::ERR_CONC = "All concentrations have to be between 0 and 1."** [static, private]

Definition at line 58 of file RectangularMatFactory.hpp.

Referenced by processConcentrations().

7.128.4.4 **const int RectangularMatFactory::INPUT_IMG_HEIGHT = 2048**
[static, private]

Definition at line 66 of file RectangularMatFactory.hpp.

Referenced by isValidViewerImage().

7.128.4.5 **const int RectangularMatFactory::INPUT_IMG_WIDTH = 2048** [static,
private]

Definition at line 65 of file RectangularMatFactory.hpp.

Referenced by isValidViewerImage().

7.128.4.6 **const int RectangularMatFactory::ROI_HEIGHT = 1358** [static,
private]

Definition at line 63 of file RectangularMatFactory.hpp.

Referenced by createFromViewerImage().

7.128.4.7 **const int RectangularMatFactory::ROI_START_X = 321** [static,
private]

Definition at line 60 of file RectangularMatFactory.hpp.

Referenced by createFromViewerImage().

7.128.4.8 **const int RectangularMatFactory::ROI_START_Y = 318** [static,
private]

Definition at line 61 of file RectangularMatFactory.hpp.

Referenced by createFromViewerImage().

7.128.4.9 **const int RectangularMatFactory::ROI_WIDTH = 1407** [static,
private]

Definition at line 62 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/spatia
[RectangularMatFactory.hpp](#)

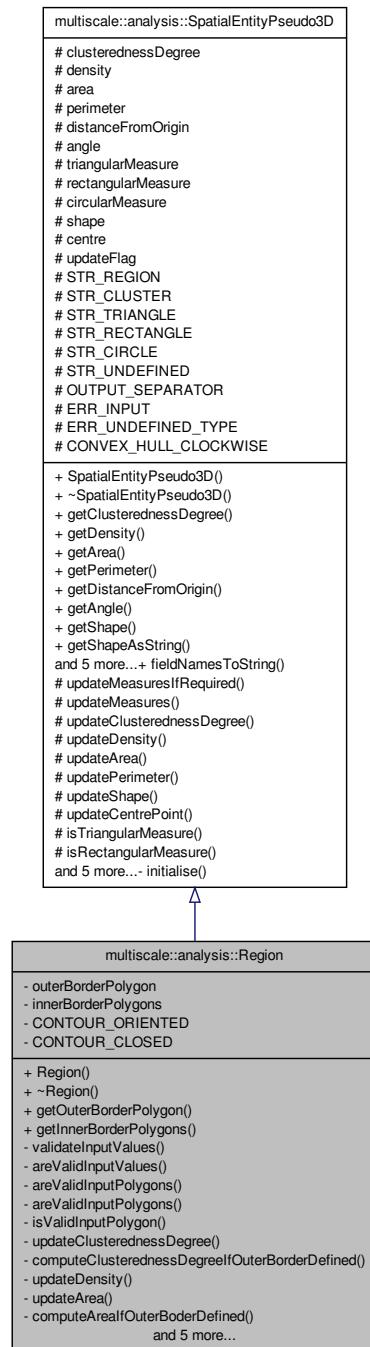
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/-
[RectangularMatFactory.cpp](#)

7.129 multiscale::analysis::Region Class Reference

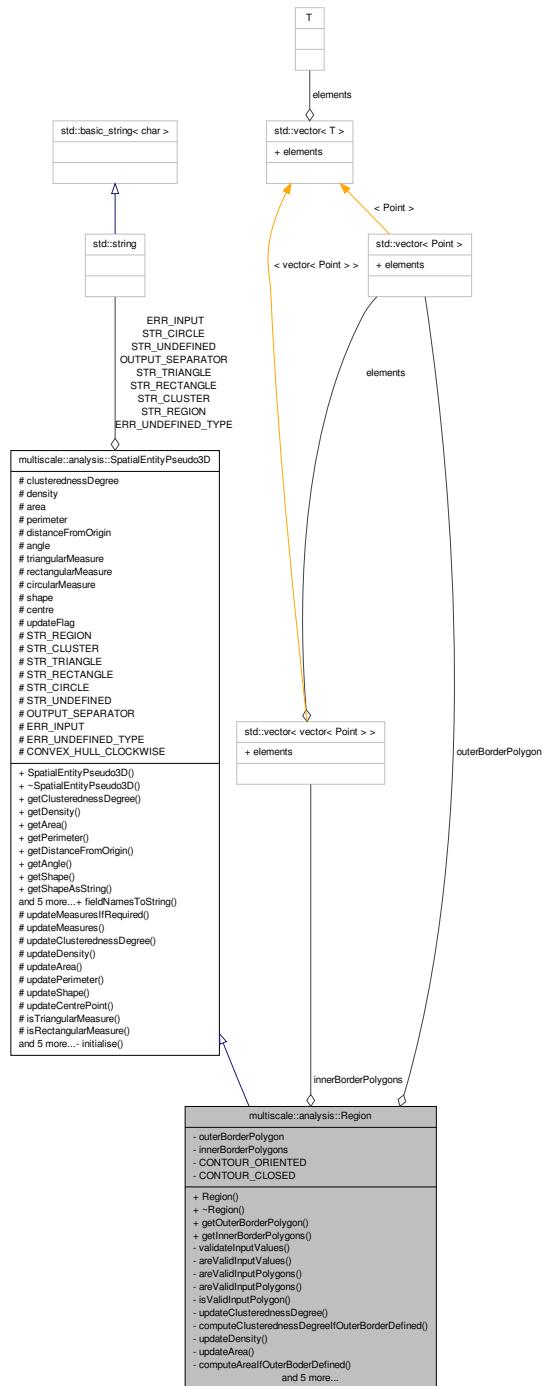
Class for representing a region.

```
#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 vector< Point > &`outerBorderPolygon`, const vector< vector< Point > > &`innerBorderPolygons`)
- `~Region ()`
- const vector< Point > & `getOuterBorderPolygon () const`
Get the polygon defining the outer border of the region.
- const vector< vector< 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 vector< Point > &`outerBorderPolygon`, const vector< vector< Point > > &`innerBorderPolygons`)
Validate the input values.
- bool `isValidInputValues` (double `density`, double `distanceFromOrigin`, double `angleWrtOrigin`, const vector< Point > &`outerBorderPolygon`, const vector< vector< Point > > &`innerBorderPolygons`)
Check if the input values are valid or not.
- bool `isValidInputPolygons` (const vector< Point > &`outerBorderPolygon`, const vector< vector< Point > > &`innerBorderPolygons`)
Check if the given input outer/inner border polygons are valid.
- bool `isValidInputPolygons` (const vector< vector< Point > > &`polygons`)
Check if the given input polygons are valid.
- bool `isValidInputPolygon` (const vector< 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

- vector< Point > [outerBorderPolygon](#)
- vector< vector< Point > > [innerBorderPolygons](#)

Static Private Attributes

- static const bool [CONTOUR_ORIENTED](#) = false
- static const bool [CONTOUR_CLOSED](#) = true

7.129.1 Detailed Description

Class for representing a region.

Definition at line 19 of file Region.hpp.

7.129.2 Constructor & Destructor Documentation

7.129.2.1 Region::Region (double *density*, double *distanceFromOrigin*, double *angleWrtOrigin*, const vector< Point > & *outerBorderPolygon*, const vector< vector< Point > > & *innerBorderPolygons*)

Definition at line 11 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::angle, multiscale::analysis::SpatialEntityPseudo3D::area, multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree, multiscale::analysis::SpatialEntityPseudo3D::density, multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin, innerBorderPolygons, outerBorderPolygon, and validateInputValues().

Referenced by [type\(\)](#).

7.129.2.2 Region::~Region ()

Definition at line 27 of file Region.cpp.

7.129.3 Member Function Documentation

7.129.3.1 bool Region::isValidInputPolygons (const vector< Point > & outerBorderPolygon, const vector< vector< 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>outerBorder-Polygon</i>	The polygon defining the outer border of the region
<i>innerBorder-Polygons</i>	The polygon defining the inner borders of the region

Definition at line 63 of file Region.cpp.

References innerBorderPolygons, and isValidInputPolygon().

Referenced by areValidInputValues().

7.129.3.2 bool Region::isValidInputPolygons (const vector< vector< 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 71 of file Region.cpp.

References isValidInputPolygon().

7.129.3.3 bool Region::areValidInputValues (double density, double distanceFromOrigin, double angleWrtOrigin, const vector< Point > & outerBorderPolygon, const vector< vector< Point > > & innerBorderPolygons) [private]

Check if the input values are valid or not.

Validation rules: $0 < \text{density} \leq 1$ $0 \leq \text{distanceFromOrigin} \leq \text{angleWrtOrigin} \leq 360$

For each polygon point p: $0 \leq p.x \leq p.y$

Parameters

<i>density</i>	The density of the region
<i>distance-FromOrigin</i>	The distance from the origin

<i>angleWrt- Origin</i>	The angle computed wrt to the origin
<i>outerBorder- Polygon</i>	The polygon defining the outer border of the region
<i>innerBorder- Polygons</i>	The polygon defining the inner borders of the region

Definition at line 46 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::area, areValidInputPolygons(), multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree, multiscale::Numeric::greaterOrEqual(), innerBorderPolygons, and multiscale::Numeric::lessOrEqual().

Referenced by validateInputValues().

7.129.3.4 double Region::computeAreaIfOuterBoderDefined() [private]

Compute the value of the area if the outer border of the region is defined.

Definition at line 118 of file Region.cpp.

References CONTOUR_ORIENTED, innerBorderPolygons, and outerBorderPolygon.

Referenced by updateArea().

7.129.3.5 double Region::computeClusterednessDegreeIfOuterBorderDefined() [private]

Compute the value of the clusteredness degree if the outer border of the region is defined.

Definition at line 97 of file Region.cpp.

References CONTOUR_ORIENTED, innerBorderPolygons, and outerBorderPolygon.

Referenced by updateClusterednessDegree().

7.129.3.6 const vector< vector< Point > > & Region::getInnerBorderPolygons() const

Get the polygons defining the inner borders of the region.

Definition at line 33 of file Region.cpp.

References innerBorderPolygons.

Referenced by multiscale::analysis::RegionDetector::outputRegionToImage().

7.129.3.7 const vector< Point > & Region::getOuterBorderPolygon() const

Get the polygon defining the outer border of the region.

Definition at line 29 of file Region.cpp.

References outerBorderPolygon.

Referenced by multiscale::analysis::RegionDetector::outputRegionToImage().

7.129.3.8 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.

7.129.3.9 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.

7.129.3.10 double Region::isTriangularMeasure() [override, private, virtual]

Get the measure that the cluster has a triangular shape.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 133 of file Region.cpp.

References multiscale::analysis::SpatialEntityPseudo3D::convertPoints(), multiscale::analysis::SpatialEntityPseudo3D::CONVEX_CLOCKWISE, multiscale::MinEnclosingTriangleFinder::find(), multiscale::analysis::SpatialEntityPseudo3D::normalisedShapeMeasure(), and outerBorderPolygon.

7.129.3.11 bool Region::isValidInputPolygon(const vector< 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 81 of file Region.cpp.

Referenced by `areValidInputPolygons()`.

7.129.3.12 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 `Region()`.

7.129.3.13 void Region::updateArea() [override, private, virtual]

Update the area.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 112 of file Region.cpp.

References `multiscale::analysis::SpatialEntityPseudo3D::area`, `computeAreaIfOuterBorderDefined()`, and `outerBorderPolygon`.

7.129.3.14 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`.

7.129.3.15 void Region::updateClusterednessDegree() [override, private, virtual]

Update the value of the clusteredness degree.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 91 of file Region.cpp.

References [multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree](#), [computeClusterednessDegreeIfOuterBorderDefined\(\)](#), and [outerBorderPolygon](#).

7.129.3.16 void Region::updateDensity() [override, private, virtual]

Update the value of the density.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 110 of file Region.cpp.

7.129.3.17 void Region::updatePerimeter() [override, private, virtual]

Update the perimeter.

Implements [multiscale::analysis::SpatialEntityPseudo3D](#).

Definition at line 129 of file Region.cpp.

References [CONTOUR_CLOSED](#), [outerBorderPolygon](#), and [multiscale::analysis::SpatialEntityPseudo3D::perimeter](#).

7.129.3.18 void Region::validateInputValues(double density, double distanceFromOrigin, double angleWrtOrigin, const vector< Point > & outerBorderPolygon, const vector< vector< 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>distance-FromOrigin</i>	The distance from the origin
<i>angleWrt-Origin</i>	The angle computed wrt to the origin
<i>outerBorder-Polygon</i>	The polygon defining the outer border of the region
<i>innerBorder-Polygons</i>	The polygon defining the inner borders of the region

Definition at line 37 of file Region.cpp.

References [areValidInputValues\(\)](#), [multiscale::analysis::SpatialEntityPseudo3D::ERR_-INPUT](#), [innerBorderPolygons](#), and [MS_throw](#).

Referenced by [Region\(\)](#).

7.129.4 Member Data Documentation

7.129.4.1 `const bool Region::CONTOUR_CLOSED = true [static, private]`

Definition at line 152 of file Region.hpp.

Referenced by updatePerimeter().

7.129.4.2 `const bool Region::CONTOUR_ORIENTED = false [static, private]`

Definition at line 151 of file Region.hpp.

Referenced by computeAreafOuterBoderDefined(), and computeClusterednessDegreeIfOuterBorderDefined().

7.129.4.3 `vector<vector<Point>> multiscale::analysis::Region::innerBorderPolygons [private]`

Polygon defining the inner borders of the region

Definition at line 24 of file Region.hpp.

Referenced by areValidInputPolygons(), areValidInputValues(), computeAreafOuterBoderDefined(), computeClusterednessDegreeIfOuterBorderDefined(), getInnerBorderPolygons(), Region(), and validateInputValues().

7.129.4.4 `vector<Point> multiscale::analysis::Region::outerBorderPolygon [private]`

Polygon defining the outer border of the region

Definition at line 23 of file Region.hpp.

Referenced by computeAreafOuterBoderDefined(), computeClusterednessDegreeIfOuterBorderDefined(), getOuterBorderPolygon(), isCircularMeasure(), isRectangularMeasure(), isTriangularMeasure(), Region(), updateArea(), updateCentrePoint(), updateClusterednessDegree(), and updatePerimeter().

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

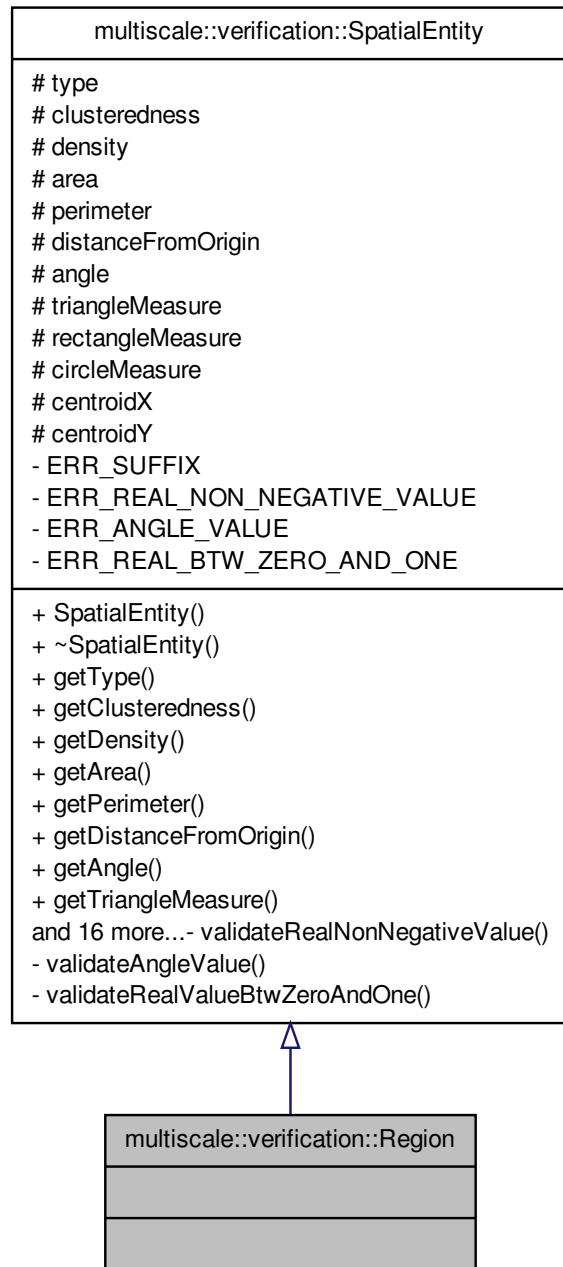
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Region.hpp](#)
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Region.cpp](#)

7.130 multiscale::verification::Region Class Reference

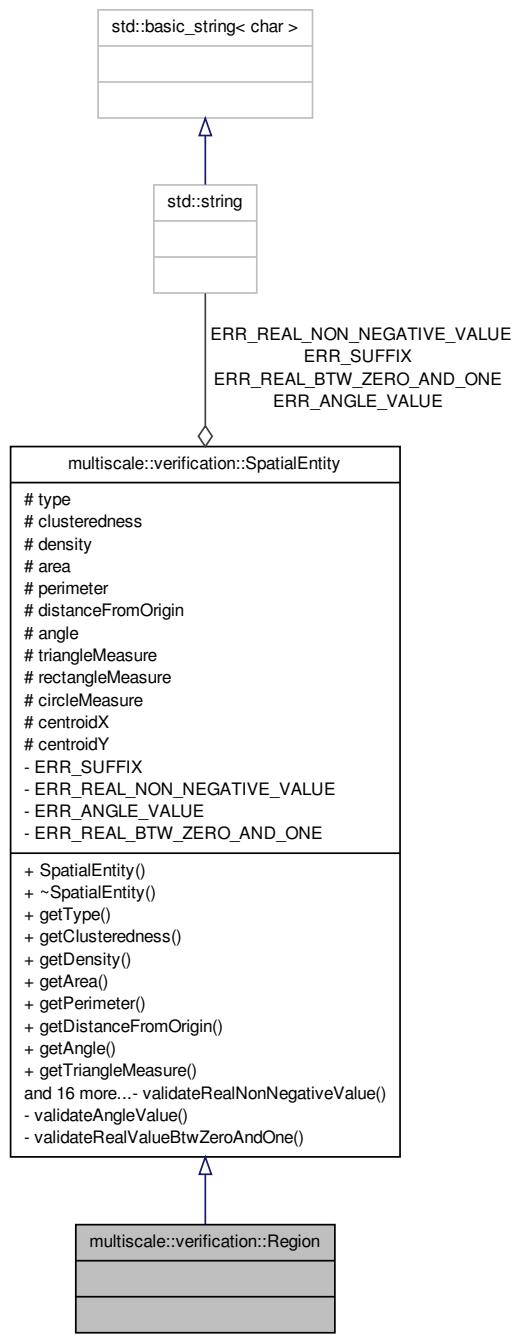
Class for representing a region.

```
#include <Region.hpp>
```

Inheritance diagram for multiscale::verification::Region:



Collaboration diagram for multiscale::verification::Region:



7.130.1 Detailed Description

Class for representing a region.

Definition at line 12 of file Region.hpp.

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

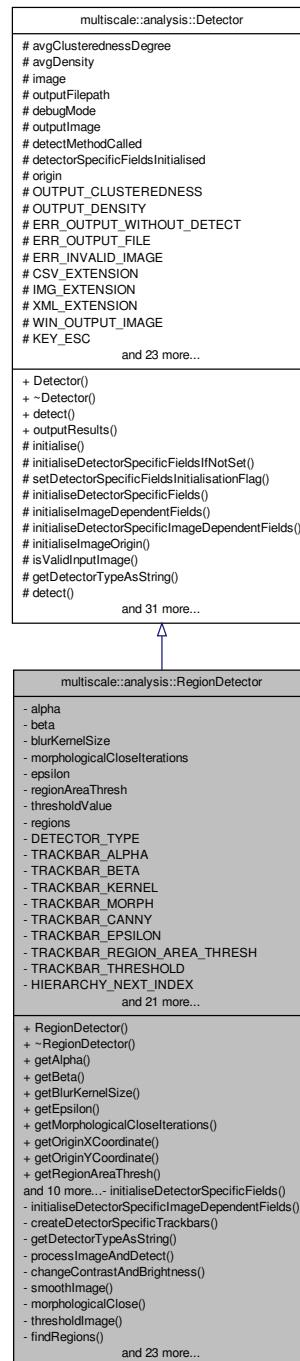
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/[Region.hpp](#)

7.131 multiscale::analysis::RegionDetector Class Reference

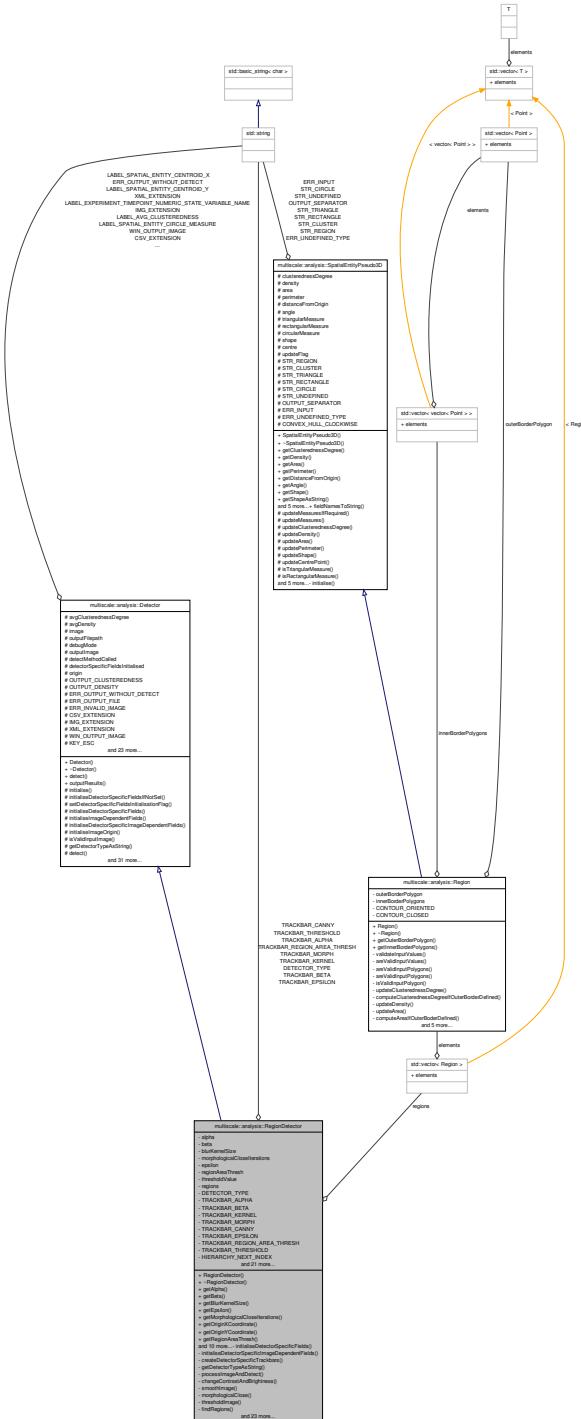
Class for detecting regions of high intensity in grayscale images.

```
#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 ()`
Get the value of field beta.
- `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.
- `vector< Region > const & getRegions ()`
Get a const reference to the 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.
- string `getDetectorTypeAsString ()` override
Get the type of the detector as a string.
- void `processImageAndDetect ()` override
Process the given image.
- void `changeContrastAndBrightness (Mat &processedImage)`
Change the contrast and brightness of the image.
- void `smoothImage (Mat &image)`
Smooth out differences in the image.
- void `morphologicalClose (Mat &image)`
Apply the morphological close operator on the image.
- void `thresholdImage (const Mat &image, Mat &thresholdedImage)`
Apply the threshold filter on the image.
- void `findRegions (const Mat &image, vector< Region > ®ions)`
Find the regions in the image.
- void `computeAverageMeasures (vector< Region > ®ions)`
Compute the average clusteredness degree and average density.
- void `computeAverageClusterednessDegree (vector< Region > ®ions)`
Compute the average clusteredness degree.
- double `sumOfAverageCentroidDistances (vector< Region > ®ions)`
Compute the sum of the average distances from each region centroid to all the other regions' centroids.
- void `computeAverageDensity (vector< Region > ®ions)`
Compute the average density.
- vector< `Polygon` > `findPolygonsInImage (const Mat &image)`
Find polygons in image.
- vector< `Polygon` > `createPolygons (const vector< vector< Point > > &contours, const vector< Vec4i > &hierarchy)`
Create polygons from the given contours and hierarchy information.
- bool `existContours (const vector< vector< Point > > &contours)`
Check if the number of contours is greater than 0.
- void `createPolygonsFromContours (const vector< vector< Point > > &contours, const vector< Vec4i > &hierarchy, vector< Polygon > &polygons)`
Create polygons from the given contours and hierarchy information.
- `Polygon createPolygon (int contourIndex, const vector< vector< Point > > &contours, const vector< Vec4i > &hierarchy)`

Create a new polygon considering the given contour index, contours and hierarchy information.

- void `setPolygonOuterContour` (int contourIndex, const vector< vector< Point > > &contours, const vector< Vec4i > &hierarchy, `Polygon` &polygon)

Set the outer contour of the polygon.

- void `setPolygonInnerContours` (int contourIndex, const vector< vector< Point > > &contours, const vector< 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 vector< Point > &contour)

Check if the contour is valid.

- bool `isValidHole` (const vector< 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 vector.

- vector< 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` ®ion, Mat &`outputImage`)

Output the region to the outputImage instance.

- void `outputRegionOuterBorderToImage` (const vector< Point > &outerBorder, - Mat &`outputImage`)

Output the outer border polygon of a region to the outputImage instance.

- void `outputRegionInnerBordersToImage` (const vector< vector< Point > > &innerBorders, 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`
- vector< `Region` > `regions`

Static Private Attributes

- static const string `DETECTOR_TYPE` = "Regions"
- static const string `TRACKBAR_ALPHA` = "Alpha"
- static const string `TRACKBAR_BETA` = "Beta"
- static const string `TRACKBAR_KERNEL` = "Gaussian blur kernel size"
- static const string `TRACKBAR_MORPH` = "Morphological open, number of iterations"
- static const string `TRACKBAR_CANNY` = "Canny lower threshold"
- static const string `TRACKBAR_EPSILON` = "Epsilon"
- static const string `TRACKBAR_REGION_AREA_THRESH` = "Region area threshold"
- static const 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

7.131.1 Detailed Description

Class for detecting regions of high intensity in grayscale images.

Definition at line 27 of file RegionDetector.hpp.

7.131.2 Constructor & Destructor Documentation

7.131.2.1 RegionDetector::RegionDetector (bool *debugMode* = false)

Definition at line 15 of file RegionDetector.cpp.

References alpha, multiscale::analysis::Detector::avgClusterednessDegree, multiscale::analysis::Detector::avgDensity, beta, blurKernelSize, epsilon, morphologicalCloseIterations, regionAreaThresh, and thresholdValue.

7.131.2.2 RegionDetector::~RegionDetector ()

Definition at line 28 of file RegionDetector.cpp.

7.131.3 Member Function Documentation

7.131.3.1 void RegionDetector::approximatePolygonOuterBorder (Polygon & *polygon*) [private]

Approximate the outer contour of the given polygon.

Parameters

<i>polygon</i>	The given polygon
----------------	-------------------

Definition at line 317 of file RegionDetector.cpp.

References epsilon.

Referenced by findRegions().

7.131.3.2 void RegionDetector::changeContrastAndBrightness (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>processed-Image</i>	The processed image
------------------------	---------------------

Definition at line 162 of file RegionDetector.cpp.

References alpha, beta, convertAlpha(), convertBeta(), and multiscale::analysis::Detector::image.

Referenced by processImageAndDetect().

7.131.3.3 void RegionDetector::clearPreviousDetectionResults() [override, private, virtual]

Clear the element present in the regions vector.

Implements [multiscale::analysis::Detector](#).

Definition at line 356 of file RegionDetector.cpp.

References regions.

7.131.3.4 void RegionDetector::computeAverageClusterednessDegree (vector< Region > & regions) [private]

Compute the average clusteredness degree.

Parameters

<i>regions</i>	The regions in the image
----------------	--------------------------

Definition at line 201 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, and sumOfAverageCentroidDistances().

Referenced by computeAverageMeasures().

7.131.3.5 void RegionDetector::computeAverageDensity (vector< Region > & regions) [private]

Compute the average density.

Parameters

<i>regions</i>	The regions in the image
----------------	--------------------------

Definition at line 236 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgDensity.

Referenced by computeAverageMeasures().

7.131.3.6 void RegionDetector::computeAverageMeasures (vector< Region > &
regions) [private]

Compute the average clusteredness degree and average density.

Parameters

regions	The regions in the image
---------	--------------------------

Definition at line 196 of file RegionDetector.cpp.

References computeAverageClusterednessDegree(), and computeAverageDensity().

Referenced by processImageAndDetect().

7.131.3.7 double RegionDetector::convertAlpha (int alpha) [private]

Convert alpha from the range [0, ALPHA_MAX] to [ALPHA_REAL_MIN, ALPHA_REAL_MAX].

Parameters

alpha	Alpha
-------	-------

Definition at line 405 of file RegionDetector.cpp.

References alpha, ALPHA_MAX, ALPHA_REAL_MAX, and ALPHA_REAL_MIN.

Referenced by changeContrastAndBrightness().

7.131.3.8 int RegionDetector::convertBeta (int beta) [private]

Convert beta from the range [0, BETA_MAX] to [BETA_REAL_MIN, BETA_REAL_MAX].

Parameters

beta	Beta
------	------

Definition at line 409 of file RegionDetector.cpp.

References beta, BETA_MAX, BETA_REAL_MAX, and BETA_REAL_MIN.

Referenced by changeContrastAndBrightness().

7.131.3.9 void RegionDetector::createDetectorSpecificTrackbars ()
[override, private, virtual]

Create the trackbars.

Implements [multiscale::analysis::Detector](#).

Definition at line 136 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.

7.131.3.10 `Polygon RegionDetector::createPolygon (int contourIndex, const vector< vector< Point > > & contours, const vector< 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 290 of file RegionDetector.cpp.

References setPolygonInnerContours(), and setPolygonOuterContour().

Referenced by createPolygonsFromContours().

7.131.3.11 `vector< Polygon > RegionDetector::createPolygons (const vector< vector< Point > > & contours, const vector< 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 263 of file RegionDetector.cpp.

References createPolygonsFromContours(), and existContours().

Referenced by findPolygonsInImage().

```
7.131.3.12 void RegionDetector::createPolygonsFromContours ( const vector<
    vector< Point > > & contours, const vector< Vec4i > & hierarchy, 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 278 of file RegionDetector.cpp.

References createPolygon(), HIERARCHY_NEXT_INDEX, and isValidContour().

Referenced by createPolygons().

```
7.131.3.13 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 323 of file RegionDetector.cpp.

References multiscale::Geometry2D::distanceBtwPoints(), multiscale::Geometry2D::minimumDistancePointIndex(), multiscale::analysis::Detector::origin, multiscale::analysis::Detector::polygonAngle(), multiscale::analysis::Region, and regionDensity().

Referenced by findRegions().

```
7.131.3.14 bool RegionDetector::existContours ( const vector< vector< Point > > &
    contours ) [private]
```

Check if the number of contours is greater than 0.

Parameters

<i>contours</i>	The given contours
-----------------	--------------------

Definition at line 274 of file RegionDetector.cpp.

Referenced by createPolygons().

7.131.3.15 `vector< Polygon > RegionDetector::findPolygonsInImage (const Mat & image) [private]`

Find polygons in image.

Parameters

<code>image</code>	The image
--------------------	-----------

Definition at line 247 of file RegionDetector.cpp.

References `createPolygons()`.

Referenced by `findRegions()`.

7.131.3.16 `void RegionDetector::findRegions (const Mat & image, vector< Region > & regions) [private]`

Find the regions in the image.

Find the contours, approximate the polygons and extract the required information from them.

Parameters

<code>image</code>	The image
<code>regions</code>	The regions in the image

Definition at line 184 of file RegionDetector.cpp.

References `approximatePolygonOuterBorder()`, `createRegionFromPolygon()`, and `findPolygonsInImage()`.

Referenced by `processImageAndDetect()`.

7.131.3.17 `int RegionDetector::getAlpha ()`

Get the value of field alpha.

Definition at line 30 of file RegionDetector.cpp.

References alpha.

Referenced by `saveDetectorParameterValues()`.

7.131.3.18 `int RegionDetector::getBeta ()`

Get the value of field beta.

Definition at line 34 of file RegionDetector.cpp.

References beta.

Referenced by `saveDetectorParameterValues()`.

7.131.3.19 int RegionDetector::getBlurKernelSize()

Get the value of field blurKernelSize.

Definition at line 38 of file RegionDetector.cpp.

References blurKernelSize.

Referenced by saveDetectorParameterValues().

**7.131.3.20 vector< shared_ptr< SpatialEntityPseudo3D > >
RegionDetector::getCollectionOfSpatialEntityPseudo3D()
[override, private, virtual]**

Get the collection of clusters detected in the image.

Implements [multiscale::analysis::Detector](#).

Definition at line 360 of file RegionDetector.cpp.

References multiscale::analysis::Region, and regions.

**7.131.3.21 string RegionDetector::getDetectorTypeAsString() [override,
private, virtual]**

Get the type of the detector as a string.

Implements [multiscale::analysis::Detector](#).

Definition at line 146 of file RegionDetector.cpp.

References DETECTOR_TYPE.

7.131.3.22 int RegionDetector::getEpsilon()

Get the value of field epsilon.

Definition at line 46 of file RegionDetector.cpp.

References epsilon.

Referenced by saveDetectorParameterValues().

7.131.3.23 int RegionDetector::getMorphologicalCloselterations()

Get the value of field morphologicalCloselterations.

Definition at line 42 of file RegionDetector.cpp.

References morphologicalCloselterations.

Referenced by saveDetectorParameterValues().

7.131.3.24 int RegionDetector::getOriginXCoordinate()

Get the value of field originXCoordinate.

Definition at line 54 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin.

7.131.3.25 int RegionDetector::getOriginYCoordinate()

Get the value of field originYCoordinate.

Definition at line 58 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin.

7.131.3.26 int RegionDetector::getRegionAreaThresh()

Get the value of field regionAreaThresh.

Definition at line 50 of file RegionDetector.cpp.

References regionAreaThresh.

Referenced by saveDetectorParameterValues().

7.131.3.27 vector< Region > const & RegionDetector::getRegions()

Get a const reference to the vector of detected regions.

Definition at line 66 of file RegionDetector.cpp.

References regions.

7.131.3.28 int RegionDetector::getThresholdValue()

Get the value of field thresholdValue.

Definition at line 62 of file RegionDetector.cpp.

References thresholdValue.

Referenced by saveDetectorParameterValues().

7.131.3.29 void RegionDetector::initialiseDetectorSpecificFields()
[override, private, virtual]

Initialise the vision members.

Implements [multiscale::analysis::Detector](#).

Definition at line 124 of file RegionDetector.cpp.

References alpha, beta, blurKernelSize, epsilon, morphologicalCloseIterations, regionAreaThresh, and thresholdValue.

7.131.3.30 void RegionDetector::initialiseDetectorSpecificImageDependentFields () [override, private, virtual]

Initialisation of the detector specific image dependent values.

Implements [multiscale::analysis::Detector](#).

Definition at line 134 of file RegionDetector.cpp.

7.131.3.31 bool RegionDetector::isValidContour (const vector< 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 333 of file RegionDetector.cpp.

References CONTOUR_AREA_ORIENTED, and regionAreaThresh.

Referenced by [createPolygonsFromContours\(\)](#).

7.131.3.32 bool RegionDetector::isValidHole (const vector< 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 339 of file RegionDetector.cpp.

References CONTOUR_AREA_ORIENTED, and THRESHOLD_HOLE_AREA.

Referenced by [setPolygonInnerContours\(\)](#).

7.131.3.33 void RegionDetector::morphologicalClose (Mat & *image*) [private]

Apply the morphological close operator on the image.

Parameters

<i>image</i>	The image
--------------	-----------

Definition at line 174 of file RegionDetector.cpp.

References morphologicalCloselterations.

Referenced by processImageAndDetect().

7.131.3.34 void RegionDetector::outputRegionInnerBordersToImage (const vector< vector< Point > > & innerBorders, 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 398 of file RegionDetector.cpp.

References DISPLAY_LINE_THICKNESS, INTENSITY_MAX, and POLYGON_CLOSED.

Referenced by outputRegionToImage().

7.131.3.35 void RegionDetector::outputRegionOuterBorderToImage (const vector< Point > & outerBorder, 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 393 of file RegionDetector.cpp.

References DISPLAY_LINE_THICKNESS, INTENSITY_MAX, and POLYGON_CLOSED.

Referenced by outputRegionToImage().

7.131.3.36 void RegionDetector::outputRegionToImage (const Region & region, 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 388 of file RegionDetector.cpp.

References multiscale::analysis::Region::getInnerBorderPolygons(), multiscale::analysis::Region::getOuterBorderPolygon(), multiscale::analysis::Detector::outputImage, outputRegionInnerBordersToImage(), and outputRegionOuterBorderToImage().

Referenced by outputResultsToImage().

7.131.3.37 void RegionDetector::outputResultsToImage() [override, private, virtual]

Output the results to the outputImage instance.

Implements [multiscale::analysis::Detector](#).

Definition at line 370 of file RegionDetector.cpp.

References multiscale::analysis::Detector::image, multiscale::analysis::Detector::outputImage, outputRegionToImage(), and regions.

7.131.3.38 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 vector of regions.

Implements [multiscale::analysis::Detector](#).

Definition at line 150 of file RegionDetector.cpp.

References changeContrastAndBrightness(), computeAverageMeasures(), findRegions(), morphologicalClose(), regions, smoothImage(), and thresholdImage().

7.131.3.39 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 345 of file RegionDetector.cpp.

References multiscale::analysis::Detector::image, and INTENSITY_MAX.

Referenced by createRegionFromPolygon().

7.131.3.40 void RegionDetector::setAlpha (int *alpha*)

Set the value of field alpha.

Parameters

<i>alpha</i>	Value of alpha
--------------	----------------

Definition at line 70 of file RegionDetector.cpp.

References alpha, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.41 void RegionDetector::setBeta (int *beta*)

Set the value of field beta.

Parameters

<i>beta</i>	Value of beta
-------------	---------------

Definition at line 76 of file RegionDetector.cpp.

References beta, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.42 void RegionDetector::setBlurKernelSize (int *blurKernelSize*)

Set the value of field blurKernelSize.

Parameters

<i>blurKernel- Size</i>	Value of blurKernelSize
-----------------------------	-------------------------

Definition at line 82 of file RegionDetector.cpp.

References blurKernelSize, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.43 void RegionDetector::setEpsilon (int *epsilon*)

Set the value of field epsilon.

Parameters

<i>epsilon</i>	Value of epsilon
----------------	------------------

Definition at line 88 of file RegionDetector.cpp.

References epsilon, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.44 void RegionDetector::setMorphologicalCloselterations (int *morphologicalCloselterations*)

Set the value of field morphologicalCloselterations.

Parameters

<i>morphological- Close- Iterations</i>	Value of morphologicalCloselterations
---	---------------------------------------

Definition at line 94 of file RegionDetector.cpp.

References morphologicalCloselterations, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.45 void RegionDetector::setOriginXCoordinate (int *originXCoordinate*)

Set the value of field originXCoordinate.

Parameters

<i>originX- Coordinate</i>	Value of originXCoordinate
--------------------------------	----------------------------

Definition at line 100 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

7.131.3.46 void RegionDetector::setOriginYCoordinate (int *originYCoordinate*)

Set the value of field originYCoordinate.

Parameters

<i>originY-Coordinate</i>	Value of originYCoordinate
---------------------------	----------------------------

Definition at line 106 of file RegionDetector.cpp.

References multiscale::analysis::Detector::origin, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

7.131.3.47 void RegionDetector::setPolygonInnerContours (int *contourIndex*, const vector< vector< Point > > & *contours*, const vector< Vec4i > & *hierarchy*, Polygon & *polygon*) [private]

Set the inner contours of the polygon.

Parameters

<i>contour-Index</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 305 of file RegionDetector.cpp.

References HIERARCHY_PARENT_INDEX, and isValidHole().

Referenced by createPolygon().

7.131.3.48 void RegionDetector::setPolygonOuterContour (int *contourIndex*, const vector< vector< Point > > & *contours*, const vector< Vec4i > & *hierarchy*, Polygon & *polygon*) [private]

Set the outer contour of the polygon.

Parameters

<i>contour-Index</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 300 of file RegionDetector.cpp.

Referenced by createPolygon().

7.131.3.49 void RegionDetector::setRegionAreaThresh (int *regionAreaThresh*)

Set the value of field regionAreaThresh.

Parameters

<i>regionArea- Thresh</i>	Value of regionAreaThresh
-------------------------------	---------------------------

Definition at line 112 of file RegionDetector.cpp.

References regionAreaThresh, and multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag().

Referenced by loadDetectorParameterValues().

7.131.3.50 void RegionDetector::setThresholdValue (int *thresholdValue*)

Set the value of field thresholdValue.

Parameters

<i>threshold- Value</i>	Value of thresholdValue
-----------------------------	-------------------------

Definition at line 118 of file RegionDetector.cpp.

References multiscale::analysis::Detector::setDetectorSpecificFieldsInitialisationFlag(), and thresholdValue.

Referenced by loadDetectorParameterValues().

7.131.3.51 void RegionDetector::smoothImage (Mat & *image*) [private]

Smooth out differences in the image.

Apply a Gaussian blur filter

Parameters

<i>image</i>	The image
--------------	-----------

Definition at line 166 of file RegionDetector.cpp.

References blurKernelSize.

Referenced by processImageAndDetect().

7.131.3.52 **double RegionDetector::sumOfAverageCentroidDistances (vector<Region > & regions)** [private]

Compute the sum of the average distances from each region centroid to all the other regions' centroids.

Parameters

<i>regions</i>	The regions in the image
----------------	--------------------------

Definition at line 216 of file RegionDetector.cpp.

References multiscale::analysis::Detector::avgClusterednessDegree, and multiscale::Geometry2D::distanceBtwPoints().

Referenced by computeAverageClusterednessDegree().

7.131.3.53 **void RegionDetector::thresholdImage (const Mat & image, Mat & thresholdedImage)** [private]

Apply the threshold filter on the image.

Parameters

<i>image</i>	The image
<i>thresholded-Image</i>	The thresholded image

Definition at line 180 of file RegionDetector.cpp.

References THRESHOLD_MAX, and thresholdValue.

Referenced by processImageAndDetect().

7.131.4 Member Data Documentation

7.131.4.1 **int multiscale::analysis::RegionDetector::alpha** [private]

Alpha for brightness and contrast adjustments

Definition at line 31 of file RegionDetector.hpp.

Referenced by changeContrastAndBrightness(), convertAlpha(), createDetectorSpecificTrackbars(), getAlpha(), initialiseDetectorSpecificFields(), RegionDetector(), and setAlpha().

7.131.4.2 **const int RegionDetector::ALPHA_MAX = 1000** [static, private]

Definition at line 382 of file RegionDetector.hpp.

Referenced by convertAlpha(), and createDetectorSpecificTrackbars().

7.131.4.3 **const double RegionDetector::ALPHA_REAL_MAX = 3.0** [static, private]

Definition at line 377 of file RegionDetector.hpp.

Referenced by convertAlpha().

7.131.4.4 **const double RegionDetector::ALPHA_REAL_MIN = 1.0** [static, private]

Definition at line 376 of file RegionDetector.hpp.

Referenced by convertAlpha().

7.131.4.5 **int multiscale::analysis::RegionDetector::beta** [private]

Beta for brightness and contrast adjustments

Definition at line 32 of file RegionDetector.hpp.

Referenced by changeContrastAndBrightness(), convertBeta(), createDetectorSpecificTrackbars(), getBeta(), initialiseDetectorSpecificFields(), RegionDetector(), and setBeta().

7.131.4.6 **const int RegionDetector::BETA_MAX = 200** [static, private]

Definition at line 383 of file RegionDetector.hpp.

Referenced by convertBeta(), and createDetectorSpecificTrackbars().

7.131.4.7 **const int RegionDetector::BETA_REAL_MAX = 100** [static, private]

Definition at line 380 of file RegionDetector.hpp.

Referenced by convertBeta().

7.131.4.8 **const int RegionDetector::BETA_REAL_MIN = -100** [static, private]

Definition at line 379 of file RegionDetector.hpp.

Referenced by convertBeta().

7.131.4.9 **int multiscale::analysis::RegionDetector::blurKernelSize** [private]

Kernel size for Gaussian blur

Definition at line 33 of file RegionDetector.hpp.

Referenced by `createDetectorSpecificTrackbars()`, `getBlurKernelSize()`, `initialiseDetectorSpecificFields()`, `RegionDetector()`, `setBlurKernelSize()`, and `smoothImage()`.

7.131.4.10 const int RegionDetector::CANNY_THRESH_MAX = 100 [static, private]

Definition at line 386 of file `RegionDetector.hpp`.

7.131.4.11 const bool RegionDetector::CONTOUR_AREA_ORIENTED = false [static, private]

Definition at line 374 of file `RegionDetector.hpp`.

Referenced by `isValidContour()`, and `isValidHole()`.

7.131.4.12 const string RegionDetector::DETECTOR_TYPE = "Regions" [static, private]

Definition at line 357 of file `RegionDetector.hpp`.

Referenced by `getDetectorTypeAsString()`.

7.131.4.13 const int RegionDetector::DISPLAY_LINE_THICKNESS = 10 [static, private]

Definition at line 397 of file `RegionDetector.hpp`.

Referenced by `outputRegionInnerBordersToImage()`, and `outputRegionOuterBorderToImage()`.

7.131.4.14 int multiscale::analysis::RegionDetector::epsilon [private]

Epsilon for polygon approximation

Definition at line 35 of file `RegionDetector.hpp`.

Referenced by `approximatePolygonOuterBorder()`, `createDetectorSpecificTrackbars()`, `getEpsilon()`, `initialiseDetectorSpecificFields()`, `RegionDetector()`, and `setEpsilon()`.

7.131.4.15 const int RegionDetector::EPSILON_MAX = 100 [static, private]

Definition at line 387 of file `RegionDetector.hpp`.

Referenced by `createDetectorSpecificTrackbars()`.

7.131.4.16 **const int RegionDetector::HIERARCHY_FIRST_CHILD_INDEX = 2** [static, private]

Definition at line 370 of file RegionDetector.hpp.

7.131.4.17 **const int RegionDetector::HIERARCHY_NEXT_INDEX = 0** [static, private]

Definition at line 368 of file RegionDetector.hpp.

Referenced by createPolygonsFromContours().

7.131.4.18 **const int RegionDetector::HIERARCHY_PARENT_INDEX = 3** [static, private]

Definition at line 371 of file RegionDetector.hpp.

Referenced by setPolygonInnerContours().

7.131.4.19 **const int RegionDetector::HIERARCHY_PREV_INDEX = 1** [static, private]

Definition at line 369 of file RegionDetector.hpp.

7.131.4.20 **const int RegionDetector::INTENSITY_MAX = 255** [static, private]

Definition at line 391 of file RegionDetector.hpp.

Referenced by outputRegionInnerBordersToImage(), outputRegionOuterBorderToImage(), and regionDensity().

7.131.4.21 **const int RegionDetector::KERNEL_MAX = 2000** [static, private]

Definition at line 384 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.131.4.22 **const int RegionDetector::MORPH_ITER_MAX = 100** [static, private]

Definition at line 385 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.131.4.23 **int multiscale::analysis::RegionDetector::morphologicalCloseIterations** [private]

Number of iterations for morphological close operator

Definition at line 34 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getMorphologicalCloselterations(), initialiseDetectorSpecificFields(), morphologicalClose(), RegionDetector(), and setMorphologicalCloselterations().

7.131.4.24 **const bool RegionDetector::POLYGON_CLOSED = true** [static, private]

Definition at line 395 of file RegionDetector.hpp.

Referenced by outputRegionInnerBordersToImage(), and outputRegionOuterBorderToImage().

7.131.4.25 **const int RegionDetector::REGION_AREA_THRESH_MAX = 200000** [static, private]

Definition at line 388 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.131.4.26 **int multiscale::analysis::RegionDetector::regionAreaThresh** [private]

Threshold for considering a region

Definition at line 36 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getRegionAreaThresh(), initialiseDetectorSpecificFields(), isValidContour(), RegionDetector(), and setRegionAreaThresh().

7.131.4.27 **vector<Region> multiscale::analysis::RegionDetector::regions** [private]

Regions detected in the image

Definition at line 39 of file RegionDetector.hpp.

Referenced by clearPreviousDetectionResults(), getCollectionOfSpatialEntityPseudo3D(), getRegions(), outputResultsToImage(), and processImageAndDetect().

7.131.4.28 **const int RegionDetector::THRESHOLD_CLUSTEREDNESS = 0**
[static, private]

Definition at line 390 of file RegionDetector.hpp.

7.131.4.29 **const int RegionDetector::THRESHOLD_HOLE_AREA = 1000**
[static, private]

Definition at line 393 of file RegionDetector.hpp.

Referenced by isValidHole().

7.131.4.30 **const int RegionDetector::THRESHOLD_MAX = 255** [static,
private]

Definition at line 389 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), and thresholdImage().

7.131.4.31 **int multiscale::analysis::RegionDetector::thresholdValue**
[private]

Value of the threshold for the threshold filter

Definition at line 37 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars(), getThresholdValue(), initialiseDetectorSpecificFields(), RegionDetector(), setThresholdValue(), and thresholdImage().

7.131.4.32 **const string RegionDetector::TRACKBAR_ALPHA = "Alpha"** [static,
private]

Definition at line 359 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.131.4.33 **const string RegionDetector::TRACKBAR_BETA = "Beta"** [static,
private]

Definition at line 360 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

7.131.4.34 **const string RegionDetector::TRACKBAR_CANNY = "Canny lower threshold"**
[static, private]

Definition at line 363 of file RegionDetector.hpp.

```
7.131.4.35 const string RegionDetector::TRACKBAR_EPSILON = "Epsilon"  
[static, private]
```

Definition at line 364 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

```
7.131.4.36 const string RegionDetector::TRACKBAR_KERNEL = "Gaussian blur kernel  
size" [static, private]
```

Definition at line 361 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

```
7.131.4.37 const string RegionDetector::TRACKBAR_MORPH = "Morphological open,  
number of iterations" [static, private]
```

Definition at line 362 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

```
7.131.4.38 const string RegionDetector::TRACKBAR_REGION_AREA_THRESH =  
"Region area threshold" [static, private]
```

Definition at line 365 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

```
7.131.4.39 const string RegionDetector::TRACKBAR_THRESHOLD = "Threshold value"  
[static, private]
```

Definition at line 366 of file RegionDetector.hpp.

Referenced by createDetectorSpecificTrackbars().

```
7.131.4.40 const bool RegionDetector::USE_CANNY_L2 = true [static,  
private]
```

Definition at line 373 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/sp/RegionDetector.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-RegionDetector.cpp

**7.132 multiscale::verification::ProbabilityErrorHandler::result<
typename, typename, typename > Struct Template -
Reference**

Structure for specifying the type of the result.

```
#include <ProbabilityErrorHandler.hpp>
```

Public Types

- `typedef void type`

7.132.1 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.

7.132.2 Member Typedef Documentation

**7.132.2.1 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](#)

**7.133 multiscale::verification::UnexpectedErrorHandler:-
:result< typename, typename, typename > Struct Template
Reference**

Structure for specifying the type of the result.

```
#include <UnexpectedErrorHandler.hpp>
```

Public Types

- `typedef void type`

7.133.1 Detailed Description

```
template<typename, typename, typename>struct multiscale::verification::UnexpectedToken-
ErrorHandler::result< typename, typename, typename >
```

Structure for specifying the type of the result.

Definition at line 23 of file `UnexpectedTokenErrorHandler.hpp`.

7.133.2 Member Typedef Documentation

```
7.133.2.1 template<typename , typename , typename > typedef void
multiscale::verification::UnexpectedTokenErrorHandler::result<
typename, typename, typename >::type
```

Definition at line 23 of file `UnexpectedTokenErrorHandler.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/Unexpected-
TokenErrorHandler.hpp`

7.134 multiscale::RGBColourGenerator Class Reference

Generate a RGB colour.

```
#include <RGBColourGenerator.hpp>
```

Public Member Functions

- string `generate` (double concentrationMin, double concentrationMax, double concentration)
 - Generate a RGB colour for the given concentration.*
- Scalar `generate` (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

- 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.
- string **convertRGBToString** ()
Convert the RGB colour to a string.

Private Attributes

- double **red**
- double **green**
- double **blue**

7.134.1 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 20 of file RGBColourGenerator.hpp.

7.134.2 Member Function Documentation

7.134.2.1 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 42 of file RGBColourGenerator.cpp.

7.134.2.2 **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 28 of file RGBColourGenerator.cpp.

7.134.2.3 **string RGBColourGenerator::convertRGBToString () [private]**

Convert the RGB colour to a string.

Definition at line 87 of file RGBColourGenerator.cpp.

7.134.2.4 **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>concentrationMin</i>	The minimum of the range of values a concentration can take
<i>concentrationMax</i>	The maximum of the range of values a concentration can take
<i>concentration</i>	The concentration

Definition at line 12 of file RGBColourGenerator.cpp.

Referenced by main(), and multiscale::analysis::SimulationClusterDetector::outputResultsToImage().

7.134.2.5 **Scalar RGBColourGenerator::generate (RNG & *randomNumberGenerator*)**

Generate a random RGB colour.

Generate a random RGB colour using the given random number generator

Parameters

<i>random- Number- Generator</i>	Random number generator
--	-------------------------

Definition at line 22 of file RGBColourGenerator.cpp.

7.134.3 Member Data Documentation

7.134.3.1 double multiscale::RGBColourGenerator::blue [private]

The amount of blue

Definition at line 26 of file RGBColourGenerator.hpp.

7.134.3.2 double multiscale::RGBColourGenerator::green [private]

The amount of green

Definition at line 25 of file RGBColourGenerator.hpp.

7.134.3.3 const int RGBColourGenerator::HUE_MAX = 120 [static]

Definition at line 75 of file RGBColourGenerator.hpp.

7.134.3.4 const int RGBColourGenerator::HUE_MIN = 0 [static]

Definition at line 74 of file RGBColourGenerator.hpp.

7.134.3.5 double multiscale::RGBColourGenerator::red [private]

The amount of red

Definition at line 24 of file RGBColourGenerator.hpp.

7.134.3.6 const int RGBColourGenerator::SATURATION = 1 [static]

Definition at line 76 of file RGBColourGenerator.hpp.

7.134.3.7 const int RGBColourGenerator::VALUE = 1 [static]

Definition at line 77 of file RGBColourGenerator.hpp.

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

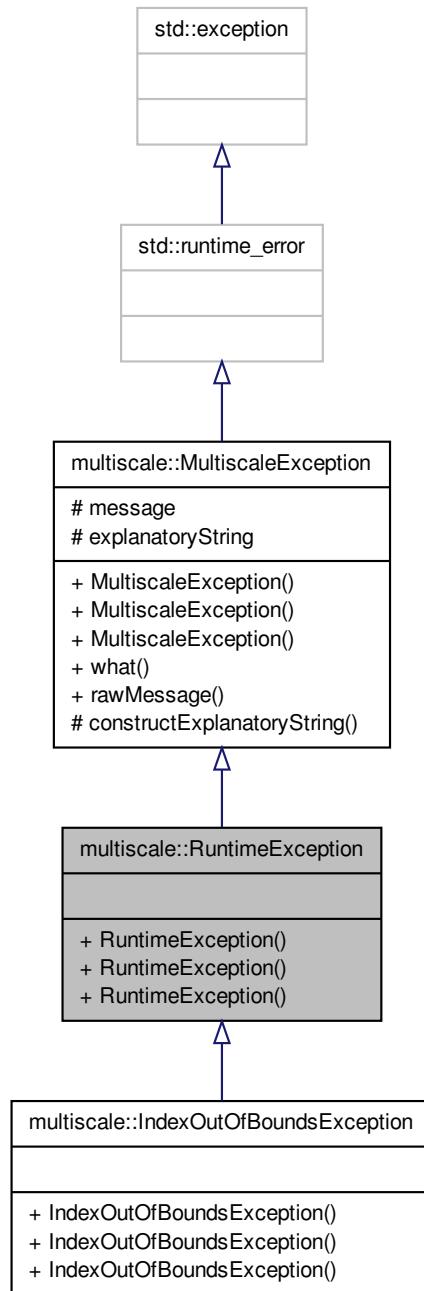
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
[RGBColourGenerator.hpp](#)
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/[RGBColour-Generator.cpp](#)

7.135 multiscale::RuntimeException Class Reference

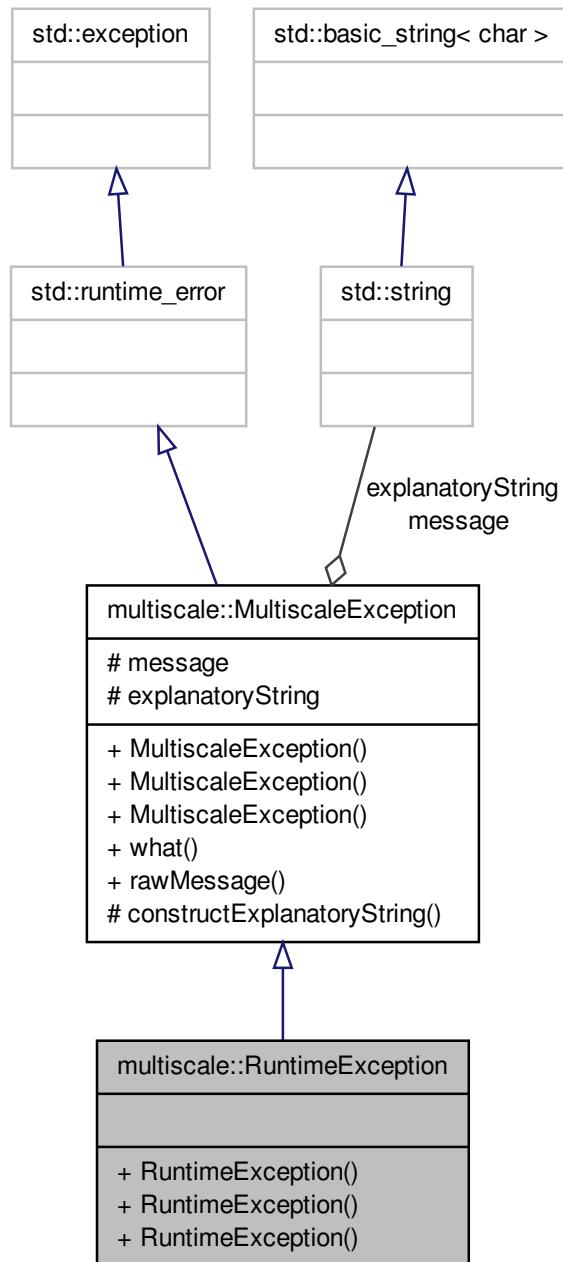
Class for representing runtime exceptions.

```
#include <RuntimeException.hpp>
```

Inheritance diagram for multiscale::RuntimeException:



Collaboration diagram for multiscale::RuntimeException:



Public Member Functions

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

7.135.1 Detailed Description

Class for representing runtime exceptions.

Definition at line 14 of file `RuntimeException.hpp`.

7.135.2 Constructor & Destructor Documentation

7.135.2.1 `multiscale::RuntimeException::RuntimeException() [inline]`

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

7.135.2.2 `multiscale::RuntimeException::RuntimeException(const string & file, int line, const string & msg) [inline, explicit]`

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

7.135.2.3 `multiscale::RuntimeException::RuntimeException(const string & file, int line, const char * msg) [inline, explicit]`

Definition at line 24 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](#)

7.136 multiscale::analysis::Silhouette Class Reference

Class for computing the "Silhouette" clustering index.

```
#include <Silhouette.hpp>
```

Static Public Member Functions

- static double [computeOverallAverageMeasure \(const vector< Cluster > &clusters\)](#)

Compute the overall average silhouette measure for the given collection of clusters.

- static double `computeAverageMeasure` (unsigned int clusterIndex, const vector<Cluster> &clusters)

Compute the average silhouette measure for the given cluster.
- static double `computeMeasure` (unsigned int entityIndex, unsigned int clusterIndex, const vector<Cluster> &clusters)

Compute the silhouette measure for the given entity.

Static Private Member Functions

- static double `computeAverageDissimilarityWithinCluster` (unsigned int entityIndex, unsigned int clusterIndex, const vector<Cluster> &clusters)

Compute the average dissimilarity within cluster to which the entity belongs.
- static double `computeAverageDissimilarityToOtherClusters` (unsigned int entityIndex, unsigned int clusterIndex, const vector<Cluster> &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)
- static double `computeAverageDissimilarityBtwEntityAndCluster` (unsigned int entityIndex, unsigned int entityClusterIndex, unsigned int clusterIndex, const vector<Cluster> &clusters)

Compute the average dissimilarity between entity and cluster.

7.136.1 Detailed Description

Class for computing the "Silhouette" clustering index.

Definition at line 14 of file Silhouette.hpp.

7.136.2 Member Function Documentation

7.136.2.1 double `Silhouette::computeAverageDissimilarityBtwEntityAndCluster` (unsigned int *entityIndex*, unsigned int *entityClusterIndex*, unsigned int *clusterIndex*, const 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>entity-ClusterIndex</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 82 of file Silhouette.cpp.

References multiscale::Geometry2D::distanceBtwPoints().

```
7.136.2.2 double Silhouette::computeAverageDissimilarityToOtherClusters (
    unsigned int entityIndex, unsigned int clusterIndex, const vector< Cluster > &
    clusters ) [static, private]
```

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 65 of file Silhouette.cpp.

```
7.136.2.3 double Silhouette::computeAverageDissimilarityWithinCluster ( unsigned
    int entityIndex, unsigned int clusterIndex, const 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 51 of file Silhouette.cpp.

References multiscale::Geometry2D::distanceBtwPoints().

```
7.136.2.4 double Silhouette::computeAverageMeasure ( unsigned int clusterIndex,
    const 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.

7.136.2.5 double **Silhouette::computeMeasure** (unsigned int *entityIndex*, unsigned int *clusterIndex*, const 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.

7.136.2.6 double **Silhouette::computeOverallAverageMeasure** (const 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 11 of file Silhouette.cpp.

Referenced by multiscale::analysis::ClusterDetector::computeClusterednessIndex().

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

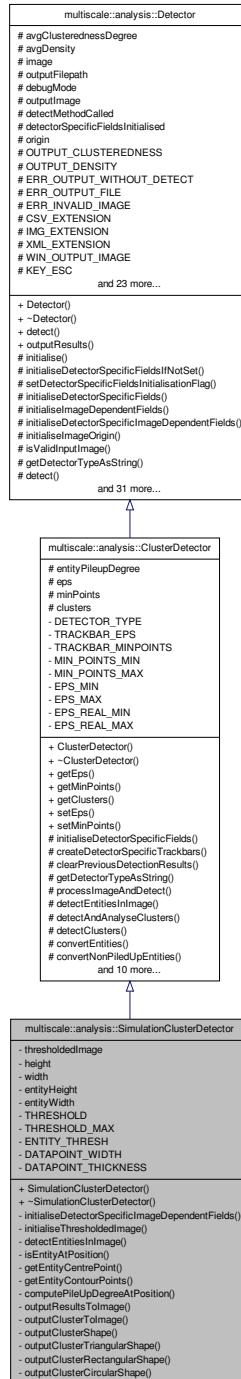
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/sp/Silhouette.hpp
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Silhouette.cpp

7.137 multiscale::analysis::SimulationClusterDetector Class - Reference

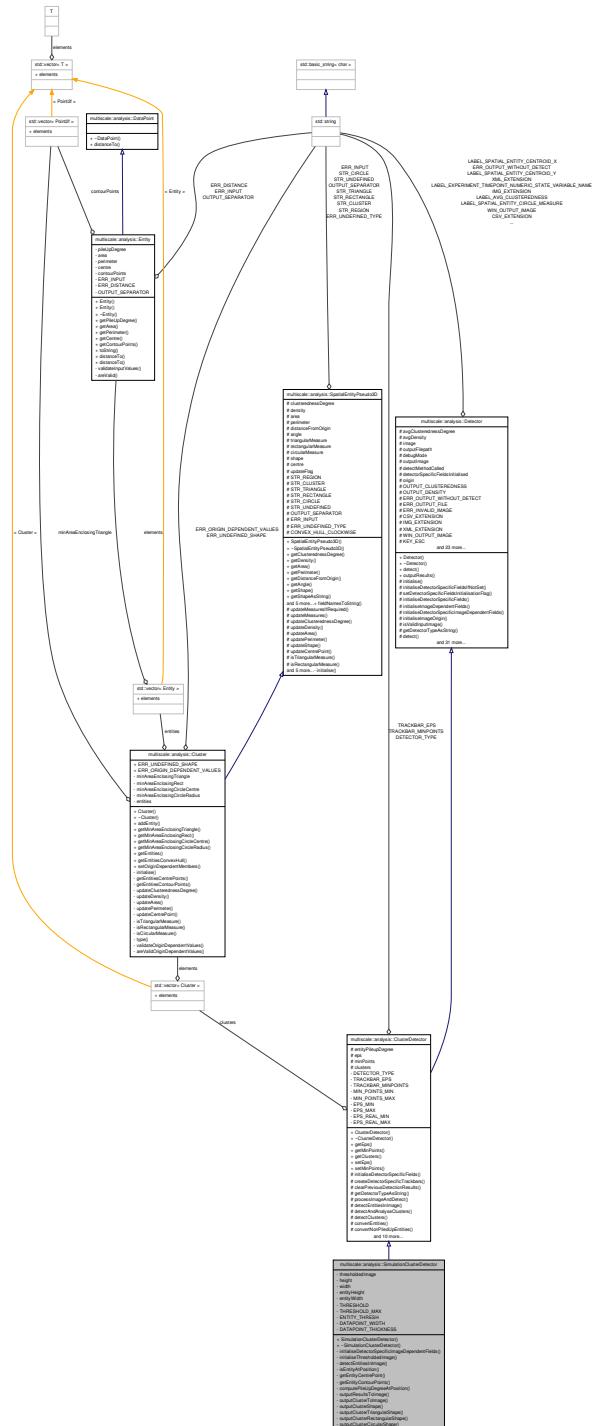
Class for detecting clusters in 2D images obtained from simulations.

```
#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 (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.
- `Point2f getEntityCentrePoint (int x, int y)`
Get the point representing the centre of the entity.
- `vector< 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, Scalar colour, Mat &image)`
Display cluster on the image.
- void `outputClusterShape (Cluster &cluster, Scalar colour, Mat &image)`
Draw the best matching shape (triangular, rectangular, circular) of the cluster on the image.
- void `outputClusterTriangularShape (Cluster &cluster, Scalar colour, Mat &image)`
Draw the best matching triangular shape of the cluster on the image.
- void `outputClusterRectangularShape (Cluster &cluster, Scalar colour, Mat &image)`
Draw the best matching rectangular shape of the cluster on the image.
- void `outputClusterCircularShape (Cluster &cluster, Scalar colour, Mat &image)`
Draw the best matching circular shape of the cluster on the image.

Private Attributes

- `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

7.137.1 Detailed Description

Class for detecting clusters in 2D images obtained from simulations.

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

7.137.2 Constructor & Destructor Documentation

7.137.2.1 `SimulationClusterDetector::SimulationClusterDetector(unsigned int height, unsigned int width, int maxPileupNumber, double maxPileupIntensity, bool debugMode = false)`

Parameters

<code>height</code>	Height of the grid used in the simulation
<code>width</code>	Width of the grid used in the simulation
<code>debugMode</code>	Flag indicating if detector should run in debug mode or not
<code>maxPileup- Number</code>	The maximum number of entities which can occupy a grid position at the same time
<code>maxPileup- Intensity</code>	The grayscale intensity of a maximally piled up grid position

Definition at line 10 of file `SimulationClusterDetector.cpp`.

References `entityHeight`, `entityWidth`, `height`, and `width`.

7.137.2.2 `SimulationClusterDetector::~SimulationClusterDetector()`

Definition at line 20 of file `SimulationClusterDetector.cpp`.

7.137.3 Member Function Documentation

7.137.3.1 `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()`.

7.137.3.2 void SimulationClusterDetector::detectEntitiesInImage (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`.

7.137.3.3 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()`.

7.137.3.4 vector< 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().

7.137.3.5 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.

7.137.3.6 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().

7.137.3.7 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().

7.137.3.8 void SimulationClusterDetector::outputClusterCircularShape (Cluster & *cluster*, Scalar *colour*, 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().

7.137.3.9 void SimulationClusterDetector::outputClusterRectangularShape (Cluster & *cluster*, Scalar *colour*, 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().

7.137.3.10 void SimulationClusterDetector::outputClusterShape (Cluster & *cluster*, Scalar *colour*, 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()`.

7.137.3.11 void **SimulationClusterDetector::outputClusterToImage (Cluster & cluster, Scalar colour, Mat & image)** [private]

Display cluster on the image.

Parameters

<code>cluster</code>	<code>Cluster</code>
<code>colour</code>	Colour associated to all entities in the cluster
<code>image</code>	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()`.

7.137.3.12 void **SimulationClusterDetector::outputClusterTriangularShape (Cluster & cluster, Scalar colour, Mat & image)** [private]

Draw the best matching triangular shape of the cluster on the image.

Parameters

<code>cluster</code>	<code>Cluster</code>
<code>colour</code>	Colour associated to all entities in the cluster
<code>image</code>	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()`.

7.137.3.13 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.

7.137.4 Member Data Documentation

7.137.4.1 **const int SimulationClusterDetector::DATAPOINT_THICKNESS = -1**
[static, private]

Definition at line 142 of file SimulationClusterDetector.hpp.

Referenced by outputClusterToImage().

7.137.4.2 **const int SimulationClusterDetector::DATAPOINT_WIDTH = 10**
[static, private]

Definition at line 141 of file SimulationClusterDetector.hpp.

Referenced by outputClusterCircularShape(), outputClusterRectangularShape(), outputClusterToImage(), and outputClusterTriangularShape().

7.137.4.3 **const int SimulationClusterDetector::ENTITY_THRESH = 200** [static, private]

Definition at line 139 of file SimulationClusterDetector.hpp.

Referenced by isEntityAtPosition().

7.137.4.4 **double multiscale::analysis::SimulationClusterDetector::entityHeight**
[private]

Height of an entity

Definition at line 27 of file SimulationClusterDetector.hpp.

Referenced by computePileUpDegreeAtPosition(), detectEntitiesInImage(), getEntityCentrePoint(), getEntityContourPoints(), initialiseDetectorSpecificImageDependentFields(), isEntityAtPosition(), and SimulationClusterDetector().

7.137.4.5 **double multiscale::analysis::SimulationClusterDetector::entityWidth**
[private]

Width of an entity

Definition at line 28 of file SimulationClusterDetector.hpp.

Referenced by computePileUpDegreeAtPosition(), detectEntitiesInImage(), getEntityCentrePoint(), getEntityContourPoints(), initialiseDetectorSpecificImageDependentFields(), isEntityAtPosition(), and SimulationClusterDetector().

7.137.4.6 **unsigned int multiscale::analysis::SimulationClusterDetector::height**
[private]

Height of the grid used in the simulation

Definition at line 24 of file SimulationClusterDetector.hpp.

Referenced by detectEntitiesInImage(), initialiseDetectorSpecificImageDependentFields(), and SimulationClusterDetector().

7.137.4.7 **const int SimulationClusterDetector::THRESHOLD = 1** [static,
private]

Definition at line 136 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage().

7.137.4.8 **const int SimulationClusterDetector::THRESHOLD_MAX = 255**
[static, private]

Definition at line 137 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage().

7.137.4.9 **Mat multiscale::analysis::SimulationClusterDetector::thresholdedImage**
[private]

Thresholded version of the image

Definition at line 22 of file SimulationClusterDetector.hpp.

Referenced by initialiseThresholdedImage(), and isEntityAtPosition().

7.137.4.10 **unsigned int multiscale::analysis::SimulationClusterDetector::width**
[private]

Width of the grid used in the simulation

Definition at line 25 of file SimulationClusterDetector.hpp.

Referenced by detectEntitiesInImage(), initialiseDetectorSpecificImageDependentFields(), and SimulationClusterDetector().

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

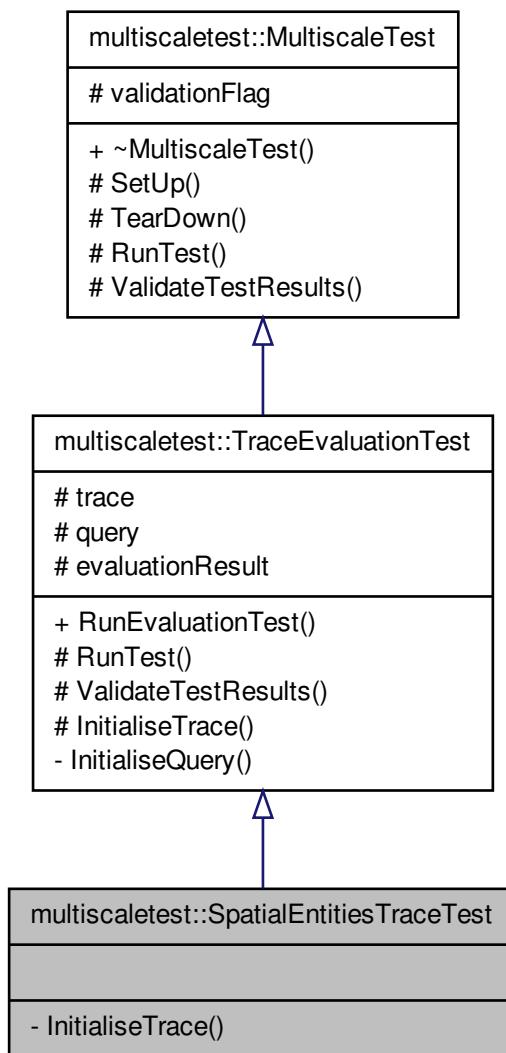
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/[SimulationClusterDetector.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/cluster/[SimulationClusterDetector.cpp](#)

7.138 multiscaletest::SpatialEntitiesTraceTest Class Reference

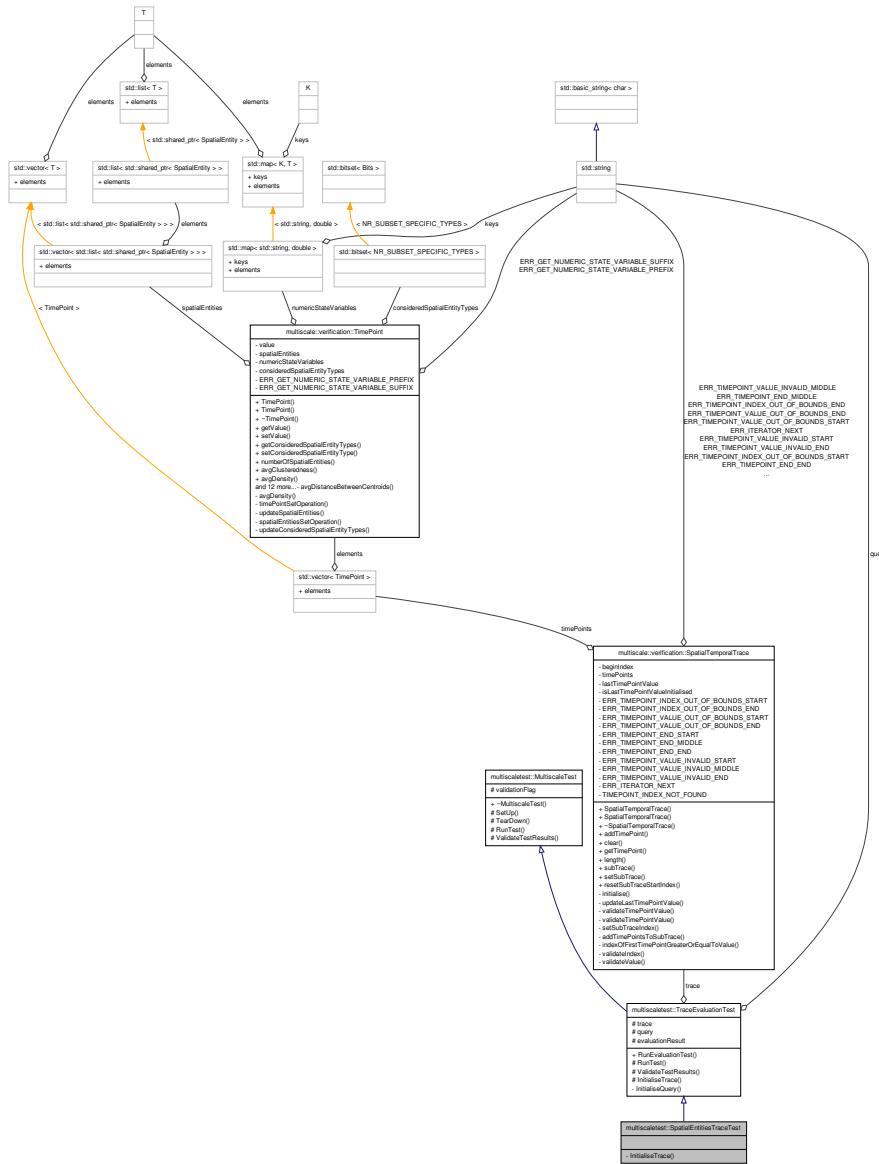
Class for testing evaluation of spatial entities-only traces.

```
#include <SpatialEntitiesTraceTest.hpp>
```

Inheritance diagram for multiscaletest::SpatialEntitiesTraceTest:



Collaboration diagram for multiscaletest::SpatialEntitiesTraceTest::



Private Member Functions

- virtual void `InitialiseTrace()` override

Initialise the trace.

7.138.1 Detailed Description

Class for testing evaluation of spatial entities-only traces.

Definition at line 13 of file SpatialEntitiesTraceTest.hpp.

7.138.2 Member Function Documentation

7.138.2.1 void multiscaletest::SpatialEntitiesTraceTest::InitialiseTrace () [override, private, virtual]

Initialise the trace.

Implements [multiscaletest::TraceEvaluationTest](#).

Definition at line 22 of file SpatialEntitiesTraceTest.hpp.

References multiscale::verification::Clusters, and multiscale::verification::Regions.

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

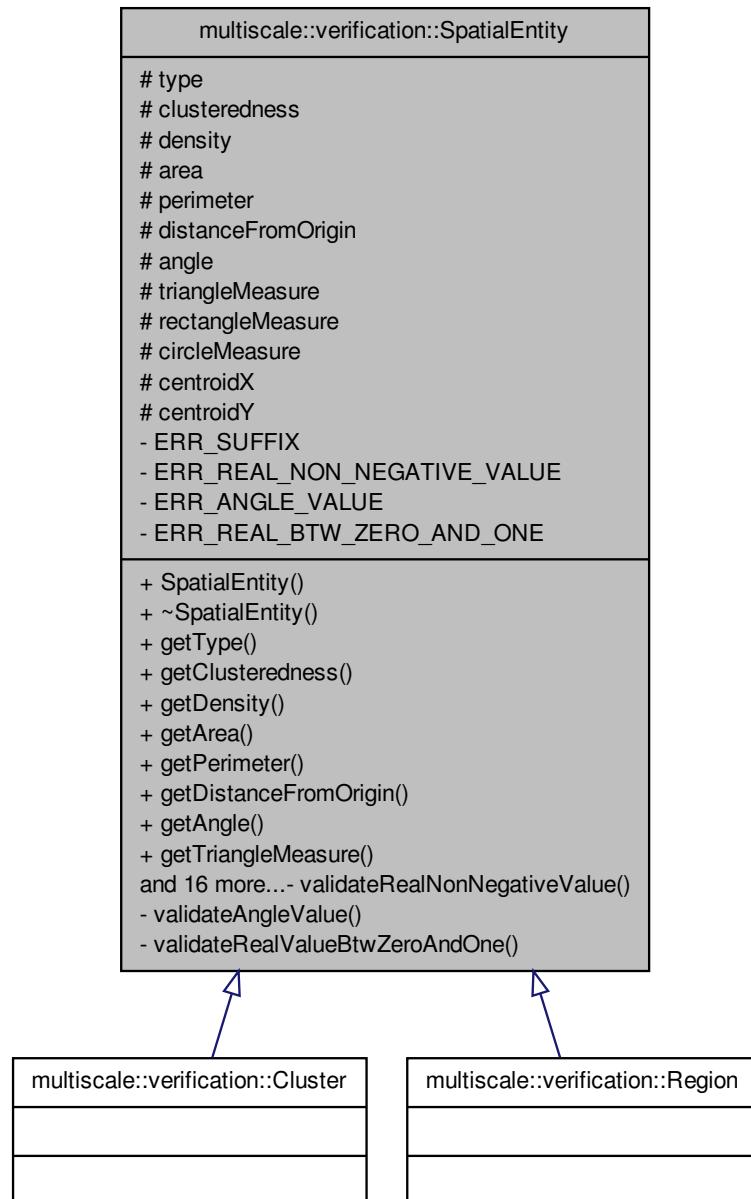
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[SpatialEntitiesTraceTest.hpp](#)

7.139 multiscale::verification::SpatialEntity Class Reference

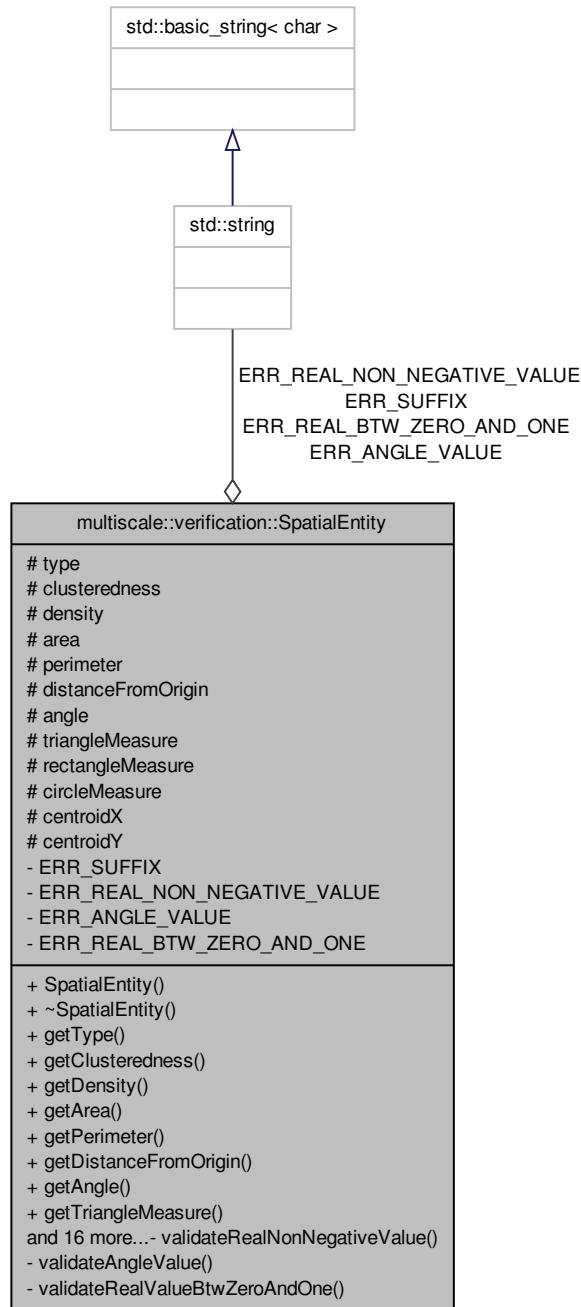
Class for representing a pseudo-3D spatial entity.

```
#include <SpatialEntity.hpp>
```

Inheritance diagram for multiscale::verification::SpatialEntity:



Collaboration diagram for multiscale::verification::SpatialEntity:



Public Member Functions

- `SpatialEntity ()`
- `~SpatialEntity ()`
- `unsigned long getType () const`

Get the type.
- `double getClusteredness () const`

Get the degree of clusteredness.
- `double getDensity () const`

Get the density.
- `double getArea () const`

Get the value of the area.
- `double getPerimeter () const`

Get the value of the perimeter.
- `double getDistanceFromOrigin () const`

Get the distance from the centre of the discretised space.
- `double getAngle () const`

Get the value of the angle in degrees.
- `double getTriangleMeasure () const`

Get the triangle measure.
- `double getRectangleMeasure () const`

Get the rectangle measure.
- `double getCircleMeasure () const`

Get the circle measure.
- `double getCentroidX () const`

Get the x-coordinate of the centroid.
- `double getCentroidY () const`

Get the y-coordinate of the centroid.
- `void setType (unsigned long type)`

Set the value of the type.
- `void setClusteredness (double clusteredness)`

Set the value of the clusteredness.
- `void setDensity (double density)`

Set the value of the density.
- `void setArea (double area)`

Set the value of the area.
- `void setPerimeter (double perimeter)`

Set the value of the perimeter.
- `void setDistanceFromOrigin (double distanceFromOrigin)`

Set the value of the distance from the origin (i.e. centre of the discretised space)
- `void setAngle (double angle)`

Set the value of the angle in degrees.
- `void setTriangleMeasure (double triangleMeasure)`

- Set the value of the triangle measure.
- void `setTriangleMeasure` (double `triangleMeasure`)
- Set the value of the rectangle measure.
- void `setRectangleMeasure` (double `rectangleMeasure`)
- Set the value of the circle measure.
- void `setCircleMeasure` (double `circleMeasure`)
- Set the value of the x-coordinate of the centroid.
- void `setCentroidX` (double `centroidX`)
- Set the value of the y-coordinate of the centroid.
- void `setCentroidY` (double `centroidY`)
- Overload the "<" operator for spatial entities.
- bool `operator<` (const `SpatialEntity` &rhsSpatialEntity) const

Protected Attributes

- unsigned long `type`
- double `clusteredness`
- double `density`
- double `area`
- double `perimeter`
- double `distanceFromOrigin`
- double `angle`
- double `triangleMeasure`
- double `rectangleMeasure`
- double `circleMeasure`
- double `centroidX`
- double `centroidY`

Private Member Functions

- void `validateRealNonNegativeValue` (double value)

Check if the provided value is a real non-negative value and throw an exception if it is not.
- void `validateAngleValue` (double value)

Check if the provided value is between 0 and 360 and throw an exception if it is not.
- void `validateRealValueBtwZeroAndOne` (double value)

Check if the provided value is a real number between 0 and 1 and throw an exception if it is not.

Static Private Attributes

- static const std::string `ERR_SUFFIX` = "."

"
- static const std::string `ERR_REAL_NON_NEGATIVE_VALUE` = "Please update the values of the spatial measures such that they are all positive (Reported error value: "

"

- static const std::string `ERR_ANGLE_VALUE` = "Please update the value of the angle such that it is positive (Reported error value: "
- static const std::string `ERR_REAL_BTW_ZERO_AND_ONE` = "Please update the values of the shape (triangular, rectangular, circular) measures such that they are between 0 and 1 (Reported error value: "

7.139.1 Detailed Description

Class for representing a pseudo-3D spatial entity.

Definition at line 15 of file SpatialEntity.hpp.

7.139.2 Constructor & Destructor Documentation

7.139.2.1 `SpatialEntity::SpatialEntity()`

Definition at line 9 of file SpatialEntity.cpp.

7.139.2.2 `multiscale::verification::SpatialEntity::~SpatialEntity() [inline]`

Definition at line 36 of file SpatialEntity.hpp.

7.139.3 Member Function Documentation

7.139.3.1 `double SpatialEntity::getAngle() const`

Get the value of the angle in degrees.

Definition at line 49 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.2 `double SpatialEntity::getArea() const`

Get the value of the area.

Definition at line 37 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.3 `double SpatialEntity::getCentroidX() const`

Get the x-coordinate of the centroid.

Definition at line 65 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.4 double SpatialEntity::getCentroidY() const

Get the y-coordinate of the centroid.

Definition at line 69 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.5 double SpatialEntity::getCircleMeasure() const

Get the circle measure.

Definition at line 61 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.6 double SpatialEntity::getClusteredness() const

Get the degree of clusteredness.

Definition at line 29 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.7 double SpatialEntity::getDensity() const

Get the density.

Definition at line 33 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.8 double SpatialEntity::getDistanceFromOrigin() const

Get the distance from the centre of the discretised space.

Definition at line 45 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.9 double SpatialEntity::getPerimeter () const

Get the value of the perimeter.

Definition at line 41 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.10 double SpatialEntity::getRectangleMeasure () const

Get the rectangle measure.

Definition at line 57 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.11 double SpatialEntity::getTriangleMeasure () const

Get the triangle measure.

Definition at line 53 of file SpatialEntity.cpp.

Referenced by multiscale::verification::SpatialEntityEvaluator::evaluate(), and multiscale-::verification::SpatialMeasureEvaluator::evaluate().

7.139.3.12 unsigned long SpatialEntity::getType () const

Get the type.

Definition at line 25 of file SpatialEntity.cpp.

7.139.3.13 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>rhsSpatial- Entity</i>	The spatial entity lying on the right hand side of the comparison operator
---------------------------	--

Definition at line 143 of file SpatialEntity.cpp.

References angle, area, centroidX, centroidY, circleMeasure, clusteredness, density, distanceFromOrigin, perimeter, rectangleMeasure, triangleMeasure, and type.

7.139.3.14 void SpatialEntity::setAngle (double *angle*)

Set the value of the angle in degrees.

Parameters

<i>angle</i>	The value of the angle in degrees
--------------	-----------------------------------

Definition at line 107 of file SpatialEntity.cpp.

7.139.3.15 void SpatialEntity::setArea (double *area*)

Set the value of the area.

Parameters

<i>area</i>	The value of the area
-------------	-----------------------

Definition at line 89 of file SpatialEntity.cpp.

7.139.3.16 void SpatialEntity::setCentroidX (double *centroidX*)

Set the value of the x-coordinate of the centroid.

Parameters

<i>centroidX</i>	The value of the x-coordinate of the centroid
------------------	---

Definition at line 131 of file SpatialEntity.cpp.

7.139.3.17 void SpatialEntity::setCentroidY (double *centroidY*)

Set the value of the y-coordinate of the centroid.

Parameters

<i>centroidY</i>	The value of the y-coordinate of the centroid
------------------	---

Definition at line 137 of file SpatialEntity.cpp.

7.139.3.18 void SpatialEntity::setCircleMeasure (double *circleMeasure*)

Set the value of the circle measure.

Parameters

<i>circle- Measure</i>	The value of the circle measure
----------------------------	---------------------------------

Definition at line 125 of file SpatialEntity.cpp.

7.139.3.19 void SpatialEntity::setClusteredness (double *clusteredness*)

Set the value of the clusteredness.

Parameters

<i>clustered-</i> <i>ness</i>	The value of the clusteredness
----------------------------------	--------------------------------

Definition at line 77 of file SpatialEntity.cpp.

7.139.3.20 void SpatialEntity::setDensity (double *density*)

Set the value of the density.

Parameters

<i>density</i>	The value of the density
----------------	--------------------------

Definition at line 83 of file SpatialEntity.cpp.

7.139.3.21 void SpatialEntity::setDistanceFromOrigin (double *distanceFromOrigin*)

Set the value of the distance from the origin (i.e. centre of the discretised space)

Parameters

<i>distance-</i> <i>FromOrigin</i>	The value of the distance from the origin (i.e. centre of the discretised space)
---------------------------------------	---

Definition at line 101 of file SpatialEntity.cpp.

7.139.3.22 void SpatialEntity::setPerimeter (double *perimeter*)

Set the value of the perimeter.

Parameters

<i>perimeter</i>	The value of the perimeter
------------------	----------------------------

Definition at line 95 of file SpatialEntity.cpp.

7.139.3.23 void SpatialEntity::setRectangleMeasure (double *rectangleMeasure*)

Set the value of the rectangle measure.

Parameters

<i>rectangle-Measure</i>	The value of the rectangle measure
--------------------------	------------------------------------

Definition at line 119 of file SpatialEntity.cpp.

7.139.3.24 void SpatialEntity::setTriangleMeasure (double triangleMeasure)

Set the value of the triangle measure.

Parameters

<i>triangle-Measure</i>	The value of the triangle measure
-------------------------	-----------------------------------

Definition at line 113 of file SpatialEntity.cpp.

7.139.3.25 void SpatialEntity::setType (unsigned long type)

Set the value of the type.

Parameters

<i>type</i>	The value of the type
-------------	-----------------------

Definition at line 73 of file SpatialEntity.cpp.

7.139.3.26 void SpatialEntity::validateAngleValue (double value) [private]

Check if the provided value is between 0 and 360 and throw an exception if it is not.

Parameters

<i>value</i>	The value to be checked
--------------	-------------------------

Definition at line 167 of file SpatialEntity.cpp.

References MS_throw_detailed.

7.139.3.27 void SpatialEntity::validateRealNonNegativeValue (double value) [private]

Check if the provided value is a real non-negative value and throw an exception if it is not.

Parameters

<i>value</i>	The value to be checked
--------------	-------------------------

Definition at line 160 of file SpatialEntity.cpp.

References MS_throw_detailed.

7.139.3.28 void SpatialEntity::validateRealValueBtwZeroAndOne (double *value*) [private]

Check if the provided value is a real number between 0 and 1 and throw an exception if it is not.

Parameters

<i>value</i>	The value to be checked
--------------	-------------------------

Definition at line 174 of file SpatialEntity.cpp.

References MS_throw_detailed.

7.139.4 Member Data Documentation

7.139.4.1 double multiscale::verification::SpatialEntity::angle [protected]

The angle described by the spatial entity with respect to the centre of the discretised space expressed in degrees

Definition at line 26 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.2 double multiscale::verification::SpatialEntity::area [protected]

The area of the spatial entity

Definition at line 23 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.3 double multiscale::verification::SpatialEntity::centroidX [protected]

The x-coordinate of the centroid of the spatial entity

Definition at line 30 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.4 **double multiscale::verification::SpatialEntity::centroidY**
[protected]

The y-coordinate of the centroid of the spatial entity

Definition at line 31 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.5 **double multiscale::verification::SpatialEntity::circleMeasure**
[protected]

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

Definition at line 29 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.6 **double multiscale::verification::SpatialEntity::clusteredness**
[protected]

The clusteredness of the spatial entity

Definition at line 21 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.7 **double multiscale::verification::SpatialEntity::density** [protected]

The density of the spatial entity

Definition at line 22 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.8 **double multiscale::verification::SpatialEntity::distanceFromOrigin**
[protected]

The distance of the spatial entity from the centre of the discretised space

Definition at line 25 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.9 **const std::string SpatialEntity::ERR_ANGLE_VALUE = "Please update the
value of the angle such that it is positive (Reported error value: "** [static,
private]

Definition at line 179 of file SpatialEntity.hpp.

```
7.139.4.10 const std::string SpatialEntity::ERR_REAL_BTW_ZERO_AND_ONE =
    "Please update the values of the shape (triangular, rectangular, circular) measures
     such that they are between 0 and 1 (Reported error value: " [static,
     private]
```

Definition at line 180 of file SpatialEntity.hpp.

```
7.139.4.11 const std::string SpatialEntity::ERR_REAL_NON_NEGATIVE_VALUE =
    "Please update the values of the spatial measures such that they are all positive
     (Reported error value: " [static, private]
```

Definition at line 178 of file SpatialEntity.hpp.

```
7.139.4.12 const std::string SpatialEntity::ERR_SUFFIX = ")." [static,
     private]
```

Definition at line 176 of file SpatialEntity.hpp.

```
7.139.4.13 double multiscale::verification::SpatialEntity::perimeter
     [protected]
```

The perimeter of the spatial entity

Definition at line 24 of file SpatialEntity.hpp.

Referenced by operator<().

```
7.139.4.14 double multiscale::verification::SpatialEntity::rectangleMeasure
     [protected]
```

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

Definition at line 28 of file SpatialEntity.hpp.

Referenced by operator<().

```
7.139.4.15 double multiscale::verification::SpatialEntity::triangleMeasure
     [protected]
```

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

Definition at line 27 of file SpatialEntity.hpp.

Referenced by operator<().

7.139.4.16 unsigned long multiscale::verification::SpatialEntity::type
 [protected]

The type of the spatial entity

Definition at line 19 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](#)

7.140 multiscale::verification::SpatialEntityEvaluator Class Reference

Class used to evaluate spatial entities.

```
#include <SpatialEntityEvaluator.hpp>
```

Static Public Member Functions

- static double [evaluate](#) (const [SpatialEntity](#) &*spatialEntity*, const [SpatialMeasureType](#) &*spatialMeasure*)
Evaluate the numeric spatial measure for the given spatial entity.

7.140.1 Detailed Description

Class used to evaluate spatial entities.

Definition at line 9 of file SpatialEntityEvaluator.hpp.

7.140.2 Member Function Documentation

7.140.2.1 static double multiscale::verification::SpatialEntityEvaluator::evaluate (
const [SpatialEntity](#) & *spatialEntity*, const [SpatialMeasureType](#) & *spatialMeasure*
) [inline, static]

Evaluate the numeric spatial measure for the given spatial entity.

Parameters

<i>spatialEntity</i>	The spatial entity
<i>spatialMeasure</i>	The considered spatial measure

Definition at line 18 of file SpatialEntityEvaluator.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::ERR_UNDEFIN-ED_ENUM_VALUE, multiscale::verification::SpatialEntity::getAngle(), multiscale::verifica-tion::SpatialEntity::getArea(), multiscale::verification::SpatialEntity::get-CentroidX(), multiscale::verification::SpatialEntity::getCentroidY(), multiscale::verifica-tion::SpatialEntity::getCircleMeasure(), multiscale::verification::SpatialEntity::getClusteredness(), multiscale::verification::SpatialEntity::getDensity(), multiscale::verification::Spatial-Entity::getDistanceFromOrigin(), multiscale::verification::SpatialEntity::getPerimeter(), multiscale::verification::SpatialEntity::getRectangleMeasure(), multiscale::verifica-tion::SpatialEntity::getTriangleMeasure(), MS_throw, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, and multiscale::verification::Triangle-Measure.

Referenced by multiscale::verification::TimePointEvaluator::getSpatialMeasure-Values().

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

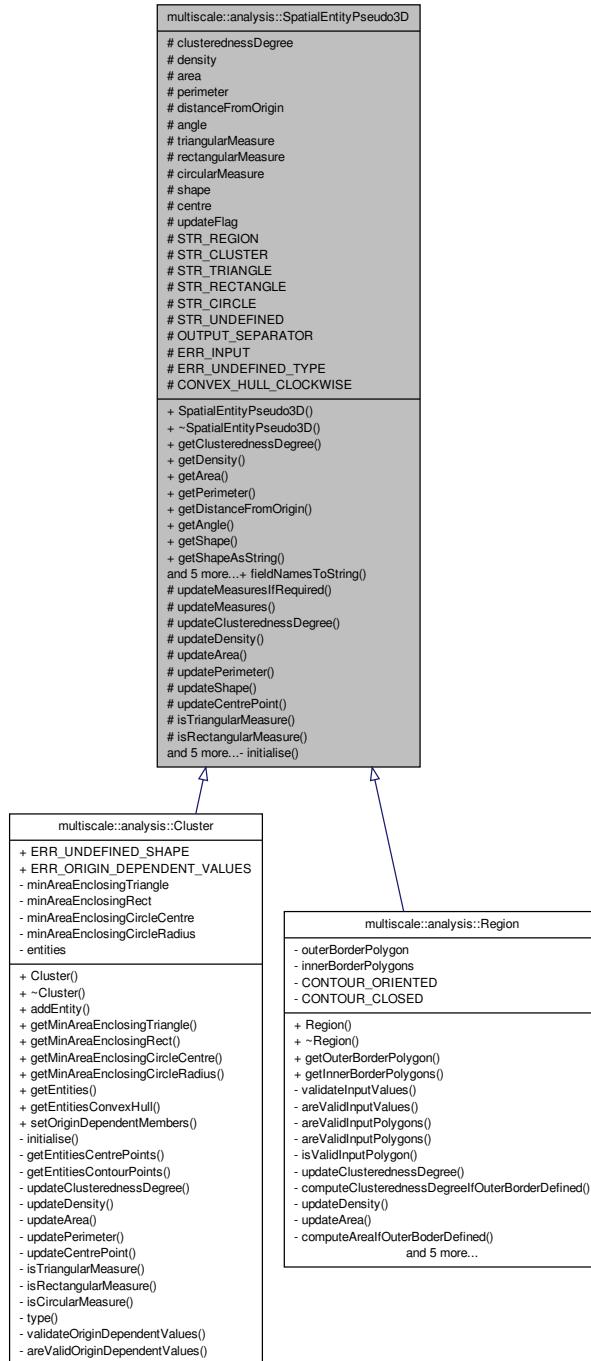
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[SpatialEntity-Evaluator.hpp](#)

7.141 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 ()`
Get the clusteredness degree.
- `virtual ~SpatialEntityPseudo3D ()`
- `double getClusterednessDegree ()`
Get the clusteredness degree.
- `double getDensity ()`
Get the density.
- `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.
- `string getShapeAsString ()`
Get the shape best fitting the spatial collection as a 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 circle.
- `Point2f getCentre ()`
Get the point defining the centre of the entity.
- `string toString ()`
Get the string representation of all field values.
- `string typeAsString ()`
Return the type of the pseudo 3D spatial entity as a string.

Static Public Member Functions

- `static string fieldNamesToString ()`
Get a 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`
Get the measure that the cluster has a triangular shape.
- virtual double `isRectangularMeasure ()=0`
Get the measure that the cluster has a rectangular shape.
- virtual double `isCircularMeasure ()=0`
Get the measure that the cluster has a circular shape.
- double `normalisedShapeMeasure (double shapeArea)`
Get the normalised shape measure ([0, 1]) that the cluster has a particular shape.
- string `shapeAsString ()`
Return the shape of the cluster as a string.
- string `fieldValuesToString ()`
Return the values of the fields as a string.
- virtual `SpatialEntityPseudo3DType type ()=0`
Return the type of the pseudo 3D spatial entity.
- vector< Point2f > `convertPoints (const vector< Point > &points)`
Convert the collection of points from type Point to type Point2f.

Protected Attributes

- double `clusterednessDegree`
- double `density`
- double `area`
- double `perimeter`
- double `distanceFromOrigin`
- double `angle`
- double `triangularMeasure`

- double `rectangularMeasure`
- double `circularMeasure`
- `Shape2D shape`
- `Point2f centre`
- bool `updateFlag`

Static Protected Attributes

- static const string `STR_REGION` = "region"
- static const string `STR_CLUSTER` = "cluster"
- static const string `STR_TRIANGLE` = "triangular"
- static const string `STR_RECTANGLE` = "rectangular"
- static const string `STR_CIRCLE` = "circular"
- static const string `STR_UNDEFINED` = "undefined"
- static const string `OUTPUT_SEPARATOR` = ","
- static const string `ERR_INPUT` = "Invalid input parameters were provided to the constructor."
- static const 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.

7.141.1 Detailed Description

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

Definition at line 18 of file SpatialEntityPseudo3D.hpp.

7.141.2 Constructor & Destructor Documentation

7.141.2.1 SpatialEntityPseudo3D::SpatialEntityPseudo3D()

Definition at line 8 of file SpatialEntityPseudo3D.cpp.

References `initialise()`.

7.141.2.2 SpatialEntityPseudo3D::~SpatialEntityPseudo3D() [virtual]

Definition at line 12 of file SpatialEntityPseudo3D.cpp.

7.141.3 Member Function Documentation

7.141.3.1 `vector< Point2f > SpatialEntityPseudo3D::convertPoints (const vector< Point > & points) [protected]`

Convert the collection of points from type Point to type Point2f.

Parameters

<code><i>points</i></code>	Collection of points
----------------------------	----------------------

Definition at line 199 of file SpatialEntityPseudo3D.cpp.

Referenced by multiscale::analysis::Region::isTriangularMeasure().

7.141.3.2 `string SpatialEntityPseudo3D::fieldNamesToString () [static]`

Get a 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().

7.141.3.3 `string SpatialEntityPseudo3D::fieldValuesToString () [protected]`

Return the values of the fields as a string.

Definition at line 180 of file SpatialEntityPseudo3D.cpp.

References angle, area, centre, circularMeasure, clusterednessDegree, density, distanceFromOrigin, OUTPUT_SEPARATOR, perimeter, rectangularMeasure, shapeAsString(), and triangularMeasure.

Referenced by toString().

7.141.3.4 `double SpatialEntityPseudo3D::getAngle ()`

Get the angle.

Definition at line 44 of file SpatialEntityPseudo3D.cpp.

References angle, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.5 `double SpatialEntityPseudo3D::getArea ()`

Get the area.

Definition at line 26 of file SpatialEntityPseudo3D.cpp.

References area, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.6 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().

7.141.3.7 double SpatialEntityPseudo3D::getCircularMeasure ()

Get the measure indicating how much the shape of the contour resembles a circle.

Definition at line 68 of file SpatialEntityPseudo3D.cpp.

References circularMeasure.

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.8 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().

7.141.3.9 double SpatialEntityPseudo3D::getDensity ()

Get the density.

Definition at line 20 of file SpatialEntityPseudo3D.cpp.

References density, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.10 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().

7.141.3.11 double SpatialEntityPseudo3D::getPerimeter()

Get the perimeter.

Definition at line 32 of file SpatialEntityPseudo3D.cpp.

References perimeter, and updateMeasuresIfRequired().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.12 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().

7.141.3.13 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().

7.141.3.14 string SpatialEntityPseudo3D::getShapeAsString()

Get the shape best fitting the spatial collection as a string.

Definition at line 56 of file SpatialEntityPseudo3D.cpp.

References shapeAsString().

Referenced by multiscale::analysis::Detector::addSpatialEntityPropertiesToTree().

7.141.3.15 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().

7.141.3.16 void SpatialEntityPseudo3D::initialise() [private]

Initialisation function for the class.

Reimplemented in [multiscale::analysis::Cluster](#).

Definition at line 209 of file SpatialEntityPseudo3D.cpp.

References area, circularMeasure, perimeter, rectangularMeasure, shape, triangularMeasure, multiscale::analysis::Undefined, and updateFlag.

Referenced by SpatialEntityPseudo3D().

7.141.3.17 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().

7.141.3.18 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().

7.141.3.19 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().

7.141.3.20 double SpatialEntityPseudo3D::normalisedShapeMeasure(double shapeArea) [protected]

Get the normalised shape measure ([0, 1]) that the cluster has a particular shape.

Parameters

<code>shapeArea</code>	The area of the considered shape
------------------------	----------------------------------

Definition at line 146 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().

7.141.3.21 string SpatialEntityPseudo3D::shapeAsString() [protected]

Return the shape of the cluster as a string.

Definition at line 158 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().

7.141.3.22 string SpatialEntityPseudo3D::toString()

Get the string representation of all field values.

Definition at line 82 of file SpatialEntityPseudo3D.cpp.

References fieldValuesToString(), and updateMeasuresIfRequired().

7.141.3.23 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().

7.141.3.24 string SpatialEntityPseudo3D::typeAsString()

Return the type of the pseudo 3D spatial entity as a string.

Definition at line 88 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().

7.141.3.25 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\(\)](#).

7.141.3.26 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\(\)](#).

7.141.3.27 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\(\)](#).

7.141.3.28 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\(\)](#).

7.141.3.29 void SpatialEntityPseudo3D::updateMeasures() [protected]

Update the values of all measures.

Definition at line 117 of file [SpatialEntityPseudo3D.cpp](#).

References [updateArea\(\)](#), [updateCentrePoint\(\)](#), [updateClusterednessDegree\(\)](#), [updateDensity\(\)](#), [updatePerimeter\(\)](#), and [updateShape\(\)](#).

Referenced by [updateMeasuresIfRequired\(\)](#).

7.141.3.30 void SpatialEntityPseudo3D::updateMeasuresIfRequired() [protected]

Update the values of all measures if required.

Definition at line 109 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().

7.141.3.31 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().

7.141.3.32 void SpatialEntityPseudo3D::updateShape() [protected]

Update the shape of the cluster.

Definition at line 126 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().

7.141.4 Member Data Documentation

7.141.4.1 double multiscale::analysis::SpatialEntityPseudo3D::angle [protected]

Angle of the region wrt the origin

Definition at line 34 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getAngle(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::setOriginDependentMembers().

7.141.4.2 double multiscale::analysis::SpatialEntityPseudo3D::area [protected]

Area of the spatial collection

Definition at line 30 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Region::areValidInputValues(), fieldValuesToString(), getArea(), initialise(), normalisedShapeMeasure(), multiscale::analysis::-

Region::Region(), multiscale::analysis::Cluster::updateArea(), and multiscale::analysis::Region::updateArea().

**7.141.4.3 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().

**7.141.4.4 double multiscale::analysis::SpatialEntityPseudo3D::circularMeasure
[protected]**

Measure ([0, 1]) indicating that the shape of the spatial collection is circular

Definition at line 38 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getCircularMeasure(), initialise(), and updateShape().

**7.141.4.5 double multiscale::analysis::SpatialEntityPseudo3D::clusterednessDegree
[protected]**

Degree of clusteredness

Definition at line 22 of file SpatialEntityPseudo3D.hpp.

Referenced by multiscale::analysis::Region::isValidInputValues(), fieldValuesToString(), getClusterednessDegree(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), multiscale::analysis::Cluster::updateClusterednessDegree(), and multiscale::analysis::Region::updateClusterednessDegree().

**7.141.4.6 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().

**7.141.4.7 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 23 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getDensity(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::updateDensity().

7.141.4.8 double multiscale::analysis::SpatialEntityPseudo3D::distanceFromOrigin [protected]

Distance from the origin

Definition at line 33 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getDistanceFromOrigin(), multiscale::analysis::Cluster::initialise(), multiscale::analysis::Region::Region(), and multiscale::analysis::Cluster::setOriginDependentMembers().

7.141.4.9 const 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().

7.141.4.10 const 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().

7.141.4.11 const string SpatialEntityPseudo3D::OUTPUT_SEPARATOR = "," [static, protected]

Definition at line 170 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString().

7.141.4.12 double multiscale::analysis::SpatialEntityPseudo3D::perimeter [protected]

Perimeter of the spatial collection

Definition at line 31 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getPerimeter(), initialise(), multiscale::analysis::Cluster::updatePerimeter(), and multiscale::analysis::Region::updatePerimeter().

7.141.4.13 **double multiscale::analysis::SpatialEntityPseudo3D::rectangularMeasure** [protected]

Measure ([0, 1]) indicating that the shape of the spatial collection is rectangular

Definition at line 37 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getRectangularMeasure(), initialise(), and updateShape().

7.141.4.14 **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().

7.141.4.15 **const string SpatialEntityPseudo3D::STR_CIRCLE = "circular"** [static, protected]

Definition at line 167 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString().

7.141.4.16 **const string SpatialEntityPseudo3D::STR_CLUSTER = "cluster"** [static, protected]

Definition at line 163 of file SpatialEntityPseudo3D.hpp.

Referenced by typeAsString().

7.141.4.17 **const string SpatialEntityPseudo3D::STR_RECTANGLE = "rectangular"** [static, protected]

Definition at line 166 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString().

7.141.4.18 **const string SpatialEntityPseudo3D::STR_REGION = "region"** [static, protected]

Definition at line 162 of file SpatialEntityPseudo3D.hpp.

Referenced by typeAsString().

7.141.4.19 **const string SpatialEntityPseudo3D::STR_TRIANGLE = "triangular"**
 [static, protected]

Definition at line 165 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString().

7.141.4.20 **const string SpatialEntityPseudo3D::STR_UNDEFINED = "undefined"**
 [static, protected]

Definition at line 168 of file SpatialEntityPseudo3D.hpp.

Referenced by shapeAsString(), and typeAsString().

7.141.4.21 **double multiscale::analysis::SpatialEntityPseudo3D::triangularMeasure**
 [protected]

Measure ([0, 1]) indicating that the shape of the spatial collection is triangular

Definition at line 36 of file SpatialEntityPseudo3D.hpp.

Referenced by fieldValuesToString(), getTriangularMeasure(), initialise(), and updateShape().

7.141.4.22 **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/[spatial/SpatialEntityPseudo3D.cpp](#)

7.142 multiscale::verification::SpatialMeasureAttribute Class - Reference

Class for representing a spatial measure attribute.

```
#include <SpatialMeasureAttribute.hpp>
```

Public Attributes

- [SpatialMeasureType spatialMeasureType](#)

7.142.1 Detailed Description

Class for representing a spatial measure attribute.

Definition at line 37 of file SpatialMeasureAttribute.hpp.

7.142.2 Member Data Documentation

7.142.2.1 [SpatialMeasureType multiscale::verification::SpatialMeasureAttribute::spatialMeasureType](#)

The spatial measure type

Definition at line 41 of file SpatialMeasureAttribute.hpp.

Referenced by [multiscale::verification::FilterNumericVisitor::operator\(\)](#)(), [multiscale::verification::NumericVisitor::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](#)

7.143 multiscale::verification::SpatialMeasureEvaluator Class Reference

Class for evaluating spatial measures.

```
#include <SpatialMeasureEvaluator.hpp>
```

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.

7.143.1 Detailed Description

Class for evaluating spatial measures.

Definition at line 13 of file SpatialMeasureEvaluator.hpp.

7.143.2 Member Function Documentation

7.143.2.1 **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

<i>spatialEntity</i>	The given spatial entity
<i>type</i>	The type of the spatial measure

Definition at line 22 of file SpatialMeasureEvaluator.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::ERR_UNDEFIN-ED_ENUM_VALUE, multiscale::verification::SpatialEntity::getAngle(), multiscale::verification::SpatialEntity::getArea(), multiscale::verification::SpatialEntity::get-CentroidX(), multiscale::verification::SpatialEntity::getCentroidY(), multiscale::verification::SpatialEntity::getCircleMeasure(), multiscale::verification::SpatialEntity::getClusteredness(), multiscale::verification::SpatialEntity::getDensity(), multiscale::verification::Spatial-Entity::getDistanceFromOrigin(), multiscale::verification::SpatialEntity::getPerimeter(), multiscale::verification::SpatialEntity::getRectangleMeasure(), multiscale::verification::SpatialEntity::getTriangleMeasure(), MS_throw, multiscale::verification::Perimeter, multiscale::verification::RectangleMeasure, and multiscale::verification::Triangle-Measure.

Referenced by multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrt-SpatialMeasure().

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/[SpatialMeasure-Evaluator.hpp](#)

7.144 multiscale::verification::SpatialMeasureTypeParser Struct Reference

Symbol table and parser for the spatial measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [SpatialMeasureTypeParser \(\)](#)

7.144.1 Detailed Description

Symbol table and parser for the spatial measure type.

Definition at line 52 of file `SymbolTables.hpp`.

7.144.2 Constructor & Destructor Documentation

7.144.2.1 multiscale::verification::SpatialMeasureTypeParser::SpatialMeasureTypeParser() [inline]

Definition at line 54 of file `SymbolTables.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:

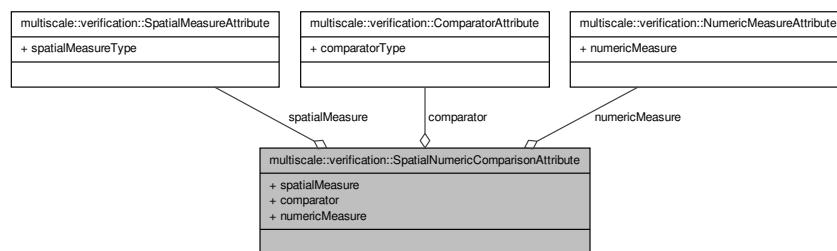
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/[SymbolTables.hpp](#)

7.145 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](#)

7.145.1 Detailed Description

Class for representing a spatial numeric comparison attribute.

Definition at line 19 of file SpatialNumericComparisonAttribute.hpp.

7.145.2 Member Data Documentation

7.145.2.1 ComparatorAttribute multiscale::verification::SpatialNumericComparisonAttribute::comparator

The comparator

Definition at line 24 of file SpatialNumericComparisonAttribute.hpp.

7.145.2.2 NumericMeasureAttribute multiscale::verification::SpatialNumericComparisonAttribute::numericMeasure

The numeric measure

Definition at line 25 of file SpatialNumericComparisonAttribute.hpp.

7.145.2.3 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:

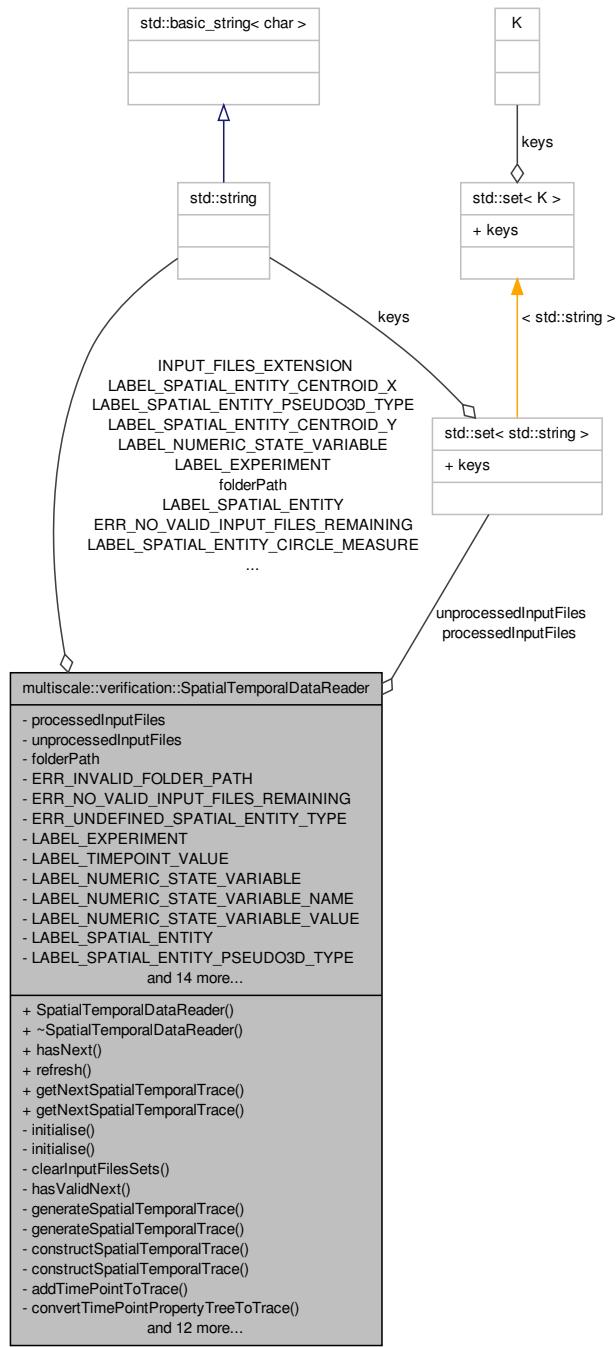
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[SpatialNumericComparisonAttribute.hpp](#)

7.146 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.
- `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 `setSpatialEntityValues` (const `pt::ptree` &`spatialEntityTree`, const std::shared_ptr< `SpatialEntity` > &`spatialEntity`)
Initialise the spatial entity values using the given spatialEntityTree.
- std::string `getFirstValidUnprocessedInputFilepath` ()
Get the first valid unprocessed input file.
- std::string `getRandomValidUnprocessedInputFilepath` ()
Get a random valid unprocessed input file.
- 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` = "<xmattr>.value"
- static const std::string `LABEL_NUMERIC_STATE_VARIABLE` = "numericStateVariable"
- 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_PSEUDO3D_TYPE` = "pseudo3D.<xmattr>.type"
- static const std::string `LABEL_SPATIAL_ENTITY_CLUSTEREDNESS` = "pseudo3D.clusteredness"
- static const std::string `LABEL_SPATIAL_ENTITY_DENSITY` = "pseudo3D.-density"
- static const std::string `LABEL_SPATIAL_ENTITY_AREA` = "pseudo3D.area"
- static const std::string `LABEL_SPATIAL_ENTITY_PERIMETER` = "pseudo3D.-perimeter"
- static const std::string `LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN` = "pseudo3D.distanceFromOrigin"
- static const std::string `LABEL_SPATIAL_ENTITY_ANGLE` = "pseudo3D.angle"
- static const std::string `LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE` = "pseudo3D.triangleMeasure"
- static const std::string `LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE` = "pseudo3D.rectangleMeasure"
- static const std::string `LABEL_SPATIAL_ENTITY_CIRCLE_MEASURE` = "pseudo3D.circleMeasure"
- static const std::string `LABEL_SPATIAL_ENTITY_CENTROID_X` = "pseudo3D.-centroid.x"
- static const std::string `LABEL_SPATIAL_ENTITY_CENTROID_Y` = "pseudo3D.-centroid.y"
- static const std::string `PSEUDO3D_SPATIAL_ENTITY_TYPE_REGION` = "region"
- static const std::string `PSEUDO3D_SPATIAL_ENTITY_TYPE_CLUSTER` = "cluster"
- static const std::string `INPUT_FILES_EXTENSION` = ".xml"
- static const std::string `INPUT_FILES_SCHEMA_PATH` = "config/verification/spatial-temporal/schema/experiment.xsd"

7.146.1 Detailed Description

Class for reading spatial temporal trace data from input files.

Definition at line 20 of file SpatialTemporalDataReader.hpp.

7.146.2 Constructor & Destructor Documentation

7.146.2.1 SpatialTemporalDataReader::SpatialTemporalDataReader (const std::string & *folderPath*)

Definition at line 20 of file SpatialTemporalDataReader.cpp.

References initialise().

7.146.2.2 SpatialTemporalDataReader::~SpatialTemporalDataReader ()

Definition at line 24 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

7.146.3 Member Function Documentation

7.146.3.1 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>timePoint-Tree</i>	The given property tree
<i>timePoint</i>	The given timepoint

Definition at line 163 of file SpatialTemporalDataReader.cpp.

References addNumericStateVariableToTimePoint(), addSpatialEntityToTimePoint(), LABEL_NUMERIC_STATE_VARIABLE, and LABEL_SPATIAL_ENTITY.

Referenced by convertTimePointPropertyTreeToTrace().

7.146.3.2 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>numeric-State-VariableTree</i>	The provided numeric state variable property tree
<i>timePoint</i>	The given timepoint

Definition at line 174 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::addNumericStateVariable(), LABEL_NUMERIC_STATE_VARIABLE_NAME, and LABEL_NUMERIC_STATE_VARIABLE_VALUE.

Referenced by addEntitiesToTimePoint().

7.146.3.3 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 182 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::addSpatialEntity(), createDerivedSpatialEntity(), and setSpatialEntityValues().

Referenced by addEntitiesToTimePoint().

7.146.3.4 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 127 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::SpatialTemporalTrace::addTimePoint(), and convertTimePointPropertyTreeToTrace().

Referenced by constructSpatialTemporalTrace().

7.146.3.5 void SpatialTemporalDataReader::clearInputFilesSets () [private]

Clear the contents of the sets of processed and unprocessed input files.

Definition at line 66 of file SpatialTemporalDataReader.cpp.

References processedInputFiles, and unprocessedInputFiles.

Referenced by initialise().

**7.146.3.6 SpatialTemporalTrace SpatialTemporalDataReader::construct-SpatialTemporalTrace (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 109 of file SpatialTemporalDataReader.cpp.

Referenced by generateSpatialTemporalTrace().

**7.146.3.7 SpatialTemporalTrace SpatialTemporalDataReader-::constructSpatialTemporalTrace (const pt::ptree & *tree*)
[private]**

Construct the spatial temporal trace corresponding to the given property tree.

Definition at line 117 of file SpatialTemporalDataReader.cpp.

References addTimePointToTrace(), and LABEL_EXPERIMENT.

**7.146.3.8 void SpatialTemporalDataReader::convertTimePointProperty-TreeToTrace (const pt::ptree & *timePointTree*, TimePoint & *timePoint*)
[private]**

Convert a time point from a property tree to a timepoint representation.

Parameters

<i>timePoint-Tree</i>	Property tree representation of the timepoint
<i>timePoint</i>	The TimePoint representation of the timepoint

Definition at line 135 of file SpatialTemporalDataReader.cpp.

References addEntitiesToTimePoint(), and setTimePointValue().

Referenced by addTimePointToTrace().

7.146.3.9 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>spatialEntity-Tree</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 193 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::Clusters, ERR_UNDEFINED_SPATIAL_ENTITY_TYPE, LABEL_SPATIAL_ENTITY_PSEUDO3D_TYPE, MS_throw, PSEUDO3D_SPATIAL_ENTITY_TYPE_CLUSTER, PSEUDO3D_SPATIAL_ENTITY_TYPE_REGION, and multiscale::verification::Regions.

Referenced by addSpatialEntityToTimePoint().

7.146.3.10 **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 89 of file SpatialTemporalDataReader.cpp.

References constructSpatialTemporalTrace(), getRandomValidUnprocessedInputFilepath(), and processedInputFiles.

Referenced by getNextSpatialTemporalTrace().

7.146.3.11 **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 99 of file SpatialTemporalDataReader.cpp.

References constructSpatialTemporalTrace(), getRandomValidUnprocessedInputFilepath(), and processedInputFiles.

7.146.3.12 std::vector< std::string > SpatialTemporalDataReader::getFilesInFolder () [private]

Get the collection of files stored in the input folder.

Definition at line 270 of file SpatialTemporalDataReader.cpp.

References folderPath, and INPUT_FILES_EXTENSION.

Referenced by updateInputFilesSets().

7.146.3.13 std::string SpatialTemporalDataReader::getFirstValidUnprocessedInputFilepath () [private]

Get the first valid unprocessed input file.

Definition at line 224 of file SpatialTemporalDataReader.cpp.

References ERR_NO_VALID_INPUT_FILES_REMAINING, hasNext(), MS_throw, and unprocessedInputFiles.

7.146.3.14 SpatialTemporalTrace SpatialTemporalDataReader::getNextSpatialTemporalTrace ()

Return the next spatial temporal trace.

Definition at line 37 of file SpatialTemporalDataReader.cpp.

References ERR_NO_VALID_INPUT_FILES_REMAINING, generateSpatialTemporalTrace(), hasNext(), and MS_throw.

Referenced by multiscale::verification::ModelCheckingManager::getNextSpatialTemporalTrace(), and readValidXmlFiles().

7.146.3.15 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 45 of file SpatialTemporalDataReader.cpp.

References ERR_NO_VALID_INPUT_FILES_REMAINING, generateSpatialTemporalTrace(), hasNext(), and MS_throw.

7.146.3.16 std::string SpatialTemporalDataReader::getRandomValidUnprocessedInputFilepath() [private]

Get a random valid unprocessed input file.

Definition at line 239 of file SpatialTemporalDataReader.cpp.

References ERR_NO_VALID_INPUT_FILES_REMAINING, hasNext(), MS_throw, and unprocessedInputFiles.

Referenced by generateSpatialTemporalTrace().

7.146.3.17 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 29 of file SpatialTemporalDataReader.cpp.

References hasValidNext().

Referenced by getFirstValidUnprocessedInputFilepath(), getNextSpatialTemporalTrace(), getRandomValidUnprocessedInputFilepath(), readValidXmlFiles(), and multiscale::verification::ModelCheckingManager::runModelCheckersForCurrentlyExistingTraces().

7.146.3.18 bool SpatialTemporalDataReader::hasValidNext() [private]

Check if there are any remaining valid unprocessed traces in the given folder.

Definition at line 71 of file SpatialTemporalDataReader.cpp.

References isValidInputFile(), processedInputFiles, and unprocessedInputFiles.

Referenced by hasNext().

7.146.3.19 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 53 of file SpatialTemporalDataReader.cpp.

References folderPath, initialise(), and validateFolderPath().

Referenced by initialise().

7.146.3.20 void SpatialTemporalDataReader::initialise() [private]

Initialise the sets for storing processed and unprocessed input files.

Definition at line 61 of file SpatialTemporalDataReader.cpp.

References clearInputFilesSets(), and updateInputFilesSets().

Referenced by SpatialTemporalDataReader().

7.146.3.21 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 regions/clusters because the uniqueness of the regions/clusters 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 274 of file SpatialTemporalDataReader.cpp.

References INPUT_FILES_SCHEMA_PATH, and multiscale::XmlValidator::isValidXmlFile().

Referenced by hasValidNext().

7.146.3.22 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().

7.146.3.23 void SpatialTemporalDataReader::setSpatialEntityValues (const pt::ptree & *spatialEntityTree*, const std::shared_ptr< SpatialEntity > & *spatialEntity*) [private]

Initialise the spatial entity values using the given spatialEntityTree.

Parameters

<i>spatialEntity-Tree</i>	The spatial entity tree
<i>spatialEntity</i>	The spatial entity to be initialised

Definition at line 209 of file SpatialTemporalDataReader.cpp.

References 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 addSpatialEntityToTimePoint().

7.146.3.24 void SpatialTemporalDataReader::setTimePointValue (const pt::ptree & *timePointTree*, TimePoint & *timePoint*) [private]

Set the value of the timepoint considering the given timepoint tree.

Parameters

<i>timePoint-Tree</i>	Property tree representation of the timepoint
<i>timePoint</i>	The TimePoint representation of the timepoint

Definition at line 141 of file SpatialTemporalDataReader.cpp.

References multiscale::verification::TimePoint::setValue(), and timePointHasValue().

Referenced by convertTimePointPropertyTreeToTrace().

7.146.3.25 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 151 of file SpatialTemporalDataReader.cpp.

References LABEL_TIMEPOINT_VALUE.

Referenced by setTimePointValue().

7.146.3.26 void SpatialTemporalDataReader::updateInputFilesSets() [private]

Update the sets of processed and unprocessed files by checking if the folder contents have been updated.

Definition at line 259 of file SpatialTemporalDataReader.cpp.

References getFilesInFolder(), processedInputFiles, and unprocessedInputFiles.

Referenced by initialise(), and refresh().

7.146.3.27 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

<i>folderPath</i>

Definition at line 278 of file SpatialTemporalDataReader.cpp.

References ERR_INVALID_FOLDER_PATH, multiscale::Filesystem::isValidFolderPath(), and MS_throw.

Referenced by initialise().

7.146.4 Member Data Documentation

7.146.4.1 const std::string SpatialTemporalDataReader::ERR_INVALID_FOLDER_PATH = "The provided path does not point to a folder. Please change." [static, private]

Definition at line 204 of file SpatialTemporalDataReader.hpp.

Referenced by validateFolderPath().

7.146.4.2 const std::string SpatialTemporalDataReader::ERR_NO_VALID_INPUT_FILES_REMAINING = "There are no valid unprocessed input files remaining." [static, private]

Definition at line 205 of file SpatialTemporalDataReader.hpp.

Referenced by `getFirstValidUnprocessedInputFilepath()`, `getNextSpatialTemporalTrace()`, and `getRandomValidUnprocessedInputFilepath()`.

7.146.4.3 `const std::string SpatialTemporalDataReader::ERR_UNDEFINED_SPATIAL_ENTITY_TYPE = "The provided spatial entity type is invalid." [static, private]`

Definition at line 206 of file `SpatialTemporalDataReader.hpp`.

Referenced by `createDerivedSpatialEntity()`.

7.146.4.4 `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()`.

7.146.4.5 `const std::string SpatialTemporalDataReader::INPUT_FILES_EXTENSION = ".xml" [static, private]`

Definition at line 233 of file `SpatialTemporalDataReader.hpp`.

Referenced by `getFilesInFolder()`.

7.146.4.6 `const std::string SpatialTemporalDataReader::INPUT_FILES_SCHEMA_PATH = "config/verification/spatial-temporal/schema/experiment.xsd" [static, private]`

Definition at line 234 of file `SpatialTemporalDataReader.hpp`.

Referenced by `isValidInputFile()`.

7.146.4.7 `const std::string SpatialTemporalDataReader::LABEL_EXPERIMENT = "experiment" [static, private]`

Definition at line 208 of file `SpatialTemporalDataReader.hpp`.

Referenced by `constructSpatialTemporalTrace()`.

7.146.4.8 `const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE = "numericStateVariable" [static, private]`

Definition at line 211 of file `SpatialTemporalDataReader.hpp`.

Referenced by `addEntitiesToTimePoint()`.

```
7.146.4.9 const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE_NAME = "name" [static, private]
```

Definition at line 212 of file SpatialTemporalDataReader.hpp.

Referenced by addNumericStateVariableToTimePoint().

```
7.146.4.10 const std::string SpatialTemporalDataReader::LABEL_NUMERIC_STATE_VARIABLE_VALUE = "value" [static, private]
```

Definition at line 213 of file SpatialTemporalDataReader.hpp.

Referenced by addNumericStateVariableToTimePoint().

```
7.146.4.11 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY = "spatialEntity" [static, private]
```

Definition at line 215 of file SpatialTemporalDataReader.hpp.

Referenced by addEntitiesToTimePoint().

```
7.146.4.12 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_ANGLE = "pseudo3D.angle" [static, private]
```

Definition at line 223 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.13 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_AREA = "pseudo3D.area" [static, private]
```

Definition at line 220 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.14 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_CENTROID_X = "pseudo3D.centroid.x" [static, private]
```

Definition at line 227 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.15 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_CENTROID_Y = "pseudo3D.centroid.y" [static, private]
```

Definition at line 228 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.16 const std::string SpatialTemporalDataReader::LABEL_SPATIAL ENTITY_CIRCLE_MEASURE = "pseudo3D.circleMeasure" [static, private]
```

Definition at line 226 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.17 const std::string SpatialTemporalDataReader::LABEL_SPATIAL ENTITY_CLUSTEREDNESS = "pseudo3D.clusteredness" [static, private]
```

Definition at line 218 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.18 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_DENSITY = "pseudo3D.density" [static, private]
```

Definition at line 219 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.19 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_DISTANCE_FROM_ORIGIN = "pseudo3D.distanceFromOrigin" [static, private]
```

Definition at line 222 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.20 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_PERIMETER = "pseudo3D.perimeter" [static, private]
```

Definition at line 221 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.21 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_PSEUDO3D_TYPE = "pseudo3D.<xmattr>.type" [static, private]
```

Definition at line 216 of file SpatialTemporalDataReader.hpp.

Referenced by createDerivedSpatialEntity().

```
7.146.4.22 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_RECTANGLE_MEASURE = "pseudo3D.rectangleMeasure" [static, private]
```

Definition at line 225 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.23 const std::string SpatialTemporalDataReader::LABEL_SPATIAL_ENTITY_TRIANGLE_MEASURE = "pseudo3D.triangleMeasure" [static, private]
```

Definition at line 224 of file SpatialTemporalDataReader.hpp.

Referenced by setSpatialEntityValues().

```
7.146.4.24 const std::string SpatialTemporalDataReader::LABEL_TIMEPOINT_VALUE = "<xmattr>.value" [static, private]
```

Definition at line 209 of file SpatialTemporalDataReader.hpp.

Referenced by timePointHasValue().

```
7.146.4.25 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(), generateSpatialTemporalTrace(), hasValidNext(), updateInputFilesSets(), and ~SpatialTemporalDataReader().

```
7.146.4.26 const std::string SpatialTemporalDataReader::PSEUDO3D_SPATIAL_ENTITY_TYPE_CLUSTER = "cluster" [static, private]
```

Definition at line 231 of file SpatialTemporalDataReader.hpp.

Referenced by `createDerivedSpatialEntity()`.

```
7.146.4.27 const std::string SpatialTemporalDataReader::PSEUDO3D_-
SPATIAL_ENTITY_TYPE_REGION = "region" [static,
private]
```

Definition at line 230 of file `SpatialTemporalDataReader.hpp`.

Referenced by `createDerivedSpatialEntity()`.

```
7.146.4.28 std::set<std::string> multiscale::verification::Spatial-
TemporalDataReader::unprocessedInputFiles
[private]
```

The set of unprocessed input files

Definition at line 25 of file `SpatialTemporalDataReader.hpp`.

Referenced by `clearInputFilesSets()`, `getFirstValidUnprocessedInputFilepath()`, `getRandomValidUnprocessedInputFilepath()`, `isValidNext()`, `updateInputFilesSets()`, and `~SpatialTemporalDataReader()`.

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

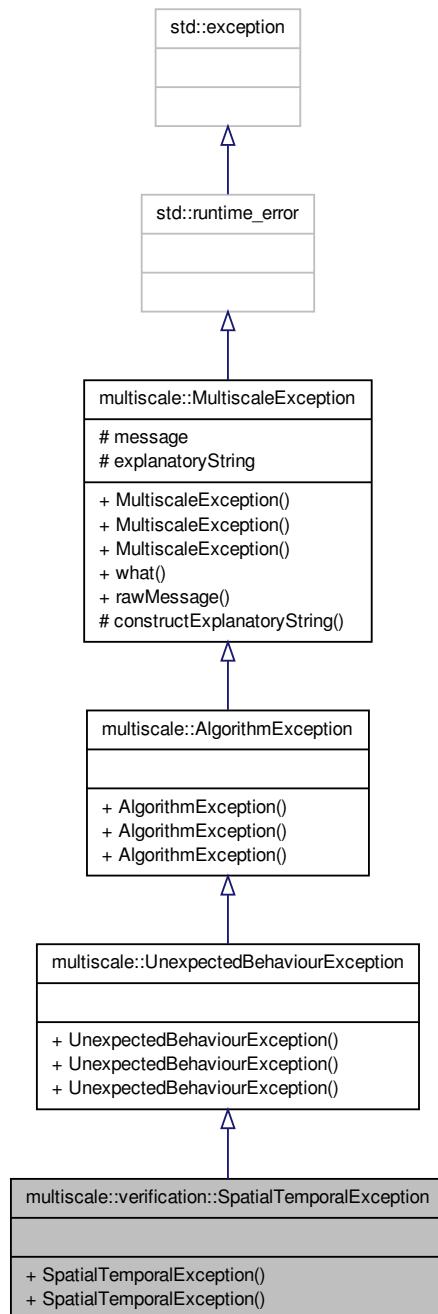
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/[SpatialTemporalDataReader.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/[SpatialTemporalDataReader.cpp](#)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

7.147 multiscale::verification::SpatialTemporalException Class - Reference

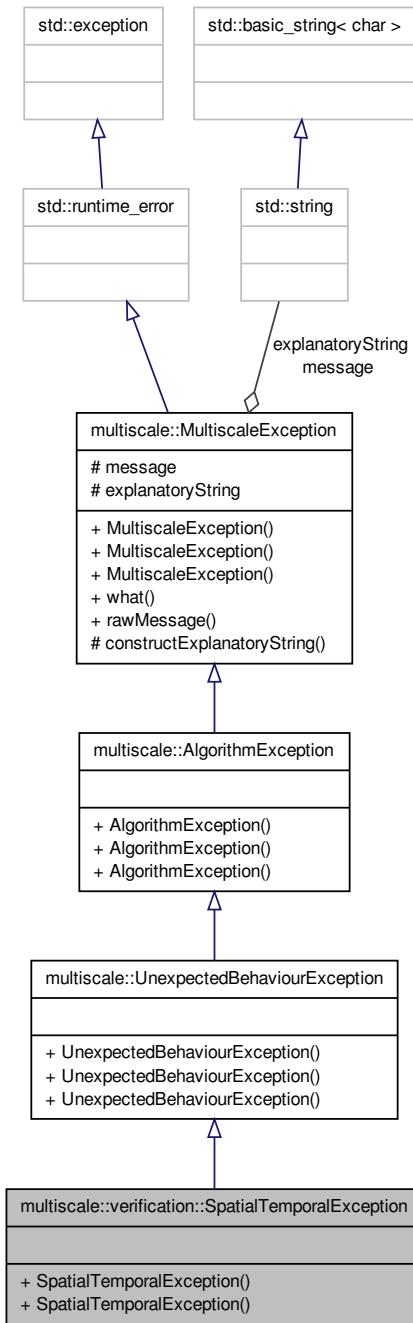
Class for representing a spatial temporal exception.

```
#include <SpatialTemporalException.hpp>
```

Inheritance diagram for multiscale::verification::SpatialTemporalException:



Collaboration diagram for multiscale::verification::SpatialTemporalException:



Public Member Functions

- [SpatialTemporalException](#) (const string &file, int line, const string &msg)
- [SpatialTemporalException](#) (const string &file, int line, const char *msg)

7.147.1 Detailed Description

Class for representing a spatial temporal exception.

Definition at line 12 of file SpatialTemporalException.hpp.

7.147.2 Constructor & Destructor Documentation

7.147.2.1 multiscale::verification::SpatialTemporalException::SpatialTemporalException (const string & file, int line, const string & msg) [inline]

Definition at line 16 of file SpatialTemporalException.hpp.

References multiscale::MultiscaleException::explanatoryString.

7.147.2.2 multiscale::verification::SpatialTemporalException::SpatialTemporalException (const string & file, int line, const char * msg) [inline]

Definition at line 20 of file SpatialTemporalException.hpp.

References multiscale::MultiscaleException::explanatoryString.

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

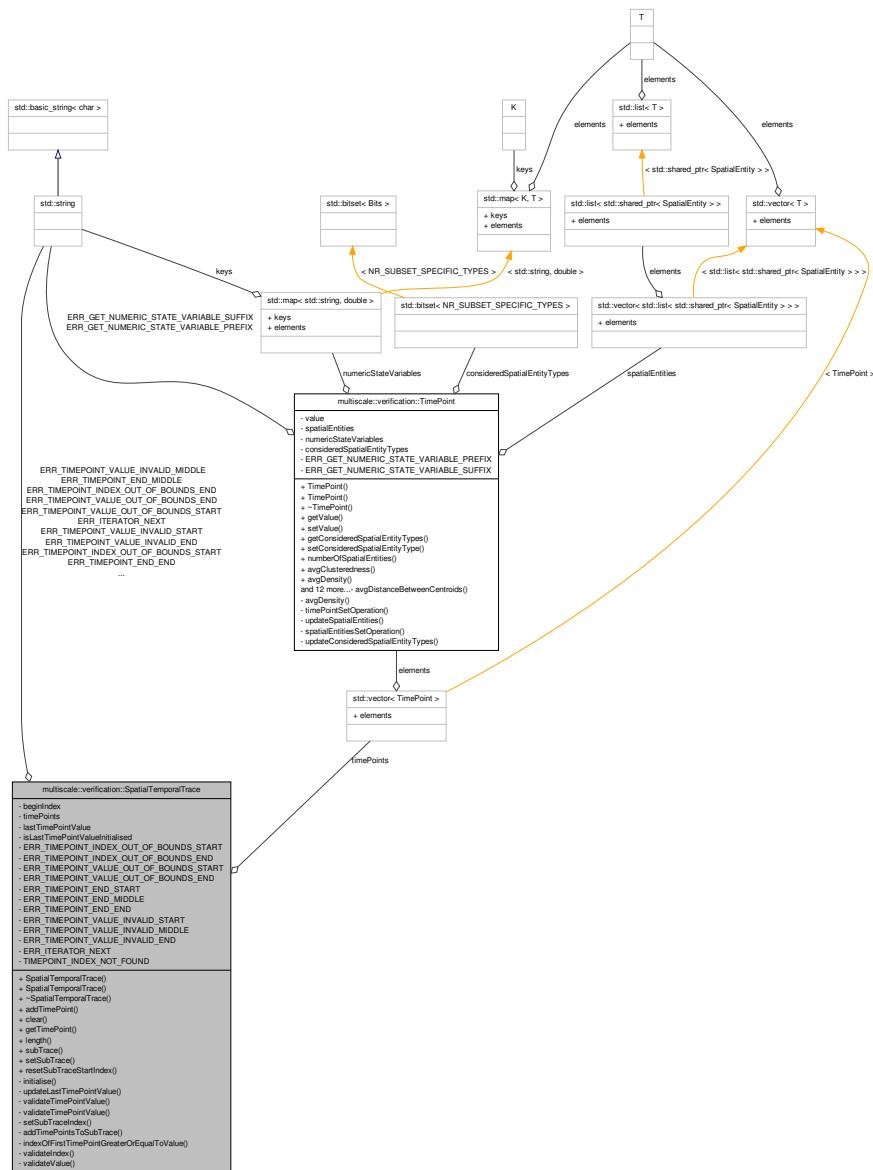
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/[SpatialTemporalException.hpp](#)

7.148 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)
- `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.
- 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_ITERATOR_NEXT` = " before calling the next() method."
- static const int `TIMEPOINT_INDEX_NOT_FOUND` = -1

7.148.1 Detailed Description

Class for representing a spatial temporal trace.

Definition at line 15 of file SpatialTemporalTrace.hpp.

7.148.2 Constructor & Destructor Documentation

7.148.2.1 SpatialTemporalTrace::SpatialTemporalTrace()

Definition at line 9 of file SpatialTemporalTrace.cpp.

References initialise().

7.148.2.2 SpatialTemporalTrace::SpatialTemporalTrace (const SpatialTemporalTrace & trace)

Definition at line 13 of file SpatialTemporalTrace.cpp.

7.148.2.3 SpatialTemporalTrace::~SpatialTemporalTrace()

Definition at line 18 of file SpatialTemporalTrace.cpp.

7.148.3 Member Function Documentation

7.148.3.1 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 20 of file SpatialTemporalTrace.cpp.

References timePoints, updateLastTimePointValue(), and validateTimePointValue().

Referenced by addTimePointsToSubTrace(), multiscale::verification::SpatialTemporalDataReader::addTimePointToTrace(), multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues(), and initialiseTrace().

7.148.3.2 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 110 of file SpatialTemporalTrace.cpp.

References addTimePoint(), and timePoints.

Referenced by subTrace().

7.148.3.3 void SpatialTemporalTrace::clear ()

Clear all the stored timepoints and reinitialise.

Definition at line 28 of file SpatialTemporalTrace.cpp.

References initialise().

7.148.3.4 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 32 of file SpatialTemporalTrace.cpp.

References beginIndex, timePoints, and validateIndex().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateNumericMeasure(), multiscale::verification::LogicPropertyVisitor::evaluateNumericSpatialMeasure(), and printTrace().

7.148.3.5 int SpatialTemporalTrace::indexOfFirstTimePoint-GreaterOrEqualToValue (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

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

Definition at line 117 of file SpatialTemporalTrace.cpp.

References beginIndex, TIMEPOINT_INDEX_NOT_FOUND, and timePoints.

Referenced by setSubTraceIndex().

7.148.3.6 void SpatialTemporalTrace::initialise () [private]

Initialise the member fields.

Definition at line 60 of file SpatialTemporalTrace.cpp.

References beginIndex, isLastTimePointValueInitialised, lastTimePointValue, and timePoints.

Referenced by clear(), and SpatialTemporalTrace().

7.148.3.7 unsigned int SpatialTemporalTrace::length () const

Get the length of the spatial temporal trace (i.e. number of timepoints)

Definition at line 38 of file SpatialTemporalTrace.cpp.

References beginIndex, and timePoints.

Referenced by multiscale::verification::ProbabilisticLogicPropertyAttribute::evaluate(), and printTrace().

7.148.3.8 void SpatialTemporalTrace::resetSubTraceStartIndex()

Reset the subtrace start index beginIndex to the value zero.

Definition at line 56 of file SpatialTemporalTrace.cpp.

References beginIndex.

7.148.3.9 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 51 of file SpatialTemporalTrace.cpp.

References setSubTraceIndex(), and validateValue().

Referenced by multiscale::verification::LogicPropertyVisitor::evaluateFutureLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluateGlobalLogicProperty(), multiscale::verification::LogicPropertyVisitor::evaluatePrecedingLogicProperties(), and multiscale::verification::LogicPropertyVisitor::evaluateUntilLogicProperty().

**7.148.3.10 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 100 of file SpatialTemporalTrace.cpp.

References beginIndex, indexOfFirstTimePointGreaterOrEqualToValue(), TIMEPOINT_INDEX_NOT_FOUND, and timePoints.

Referenced by setSubTrace().

7.148.3.11 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 42 of file SpatialTemporalTrace.cpp.

References addTimePointsToSubTrace(), beginIndex, timePoints, and validateIndex().

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

7.148.3.12 void SpatialTemporalTrace::updateLastTimePointValue (TimePoint & *timePoint*) [private]

Update the last timepoint value.

Parameters

<i>timePoint</i>	The last added timepoint
------------------	--------------------------

Definition at line 69 of file SpatialTemporalTrace.cpp.

References multiscale::verification::TimePoint::getValue(), lastTimePointValue, and multiscale::verification::TimePoint::setValue().

Referenced by addTimePoint().

7.148.3.13 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 132 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().

7.148.3.14 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 79 of file SpatialTemporalTrace.cpp.

References multiscale::verification::TimePoint::getValue().

Referenced by addTimePoint().

7.148.3.15 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 uninitialised 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-Value</i>	The value of the timepoint
------------------------	----------------------------

Definition at line 85 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.

7.148.3.16 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 139 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().

7.148.4 Member Data Documentation

7.148.4.1 unsigned int multiscale::verification::SpatialTemporalTrace::beginIndex [private]

The corresponding begin index

Definition at line 19 of file SpatialTemporalTrace.hpp.

Referenced by getTimePoint(), indexOfFirstTimePointGreaterOrEqualToValue(), ini-

tialise(), length(), resetSubTraceStartIndex(), setSubTraceIndex(), subTrace(), and validateIndex().

7.148.4.2 `const std::string SpatialTemporalTrace::ERR_ITERATOR_NEXT = "before calling the next() method."` [static, private]

Definition at line 144 of file SpatialTemporalTrace.hpp.

7.148.4.3 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_END = ")"` [static, private]

Definition at line 138 of file SpatialTemporalTrace.hpp.

7.148.4.4 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_MIDDLE = ")" should be greater or equal to the start timepoint ("` [static, private]

Definition at line 137 of file SpatialTemporalTrace.hpp.

7.148.4.5 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_END_START = "The provided end timepoint ("` [static, private]

Definition at line 136 of file SpatialTemporalTrace.hpp.

7.148.4.6 `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 131 of file SpatialTemporalTrace.hpp.

Referenced by validateIndex().

7.148.4.7 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_INDEX_OUT_OF_BOUNDS_START = "The provided timepoint index ("` [static, private]

Definition at line 130 of file SpatialTemporalTrace.hpp.

Referenced by validateIndex().

7.148.4.8 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_VALUE_INVALID_END = ")"` [static, private]

Definition at line 142 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.148.4.9 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_VALUE_INVALID_MIDDLE = ") should be greater than the previously added timepoint value ("` [static, private]

Definition at line 141 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.148.4.10 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_VALUE_INVALID_START = "The current timepoint value ("` [static, private]

Definition at line 140 of file SpatialTemporalTrace.hpp.

Referenced by validateTimePointValue().

7.148.4.11 `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 134 of file SpatialTemporalTrace.hpp.

Referenced by validateValue().

7.148.4.12 `const std::string SpatialTemporalTrace::ERR_TIMEPOINT_VALUE_OUT_OF_BOUNDS_START = "The provided timepoint value ("` [static, private]

Definition at line 133 of file SpatialTemporalTrace.hpp.

Referenced by validateValue().

7.148.4.13 `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().

7.148.4.14 `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().

7.148.4.15 **const int SpatialTemporalTrace::TIMEPOINT_INDEX_NOT_FOUND = -1**
[static, private]

Definition at line 146 of file SpatialTemporalTrace.hpp.

Referenced by `indexOfFirstTimePointGreaterOrEqualToValue()`, and `setSubTraceIndex()`.

7.148.4.16 **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()`, `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](#)

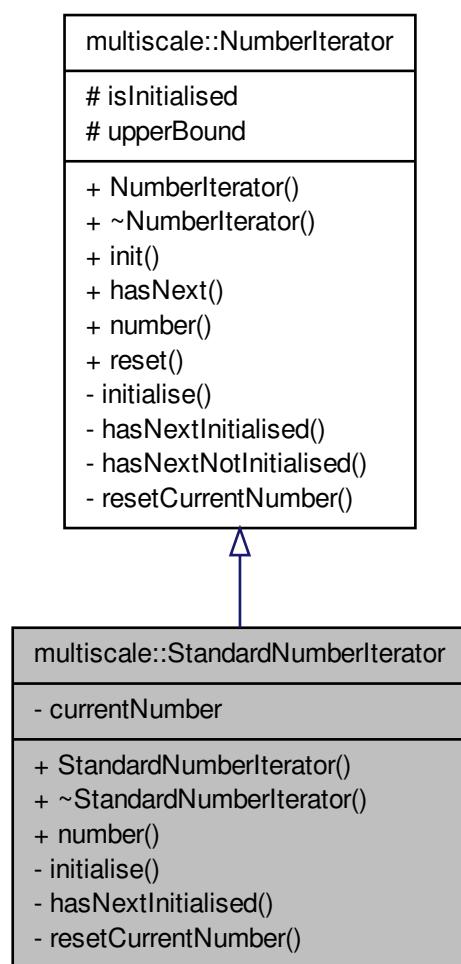
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/[SpatialTemporalTrace.cpp](#)

7.149 multiscale::StandardNumberIterator Class Reference

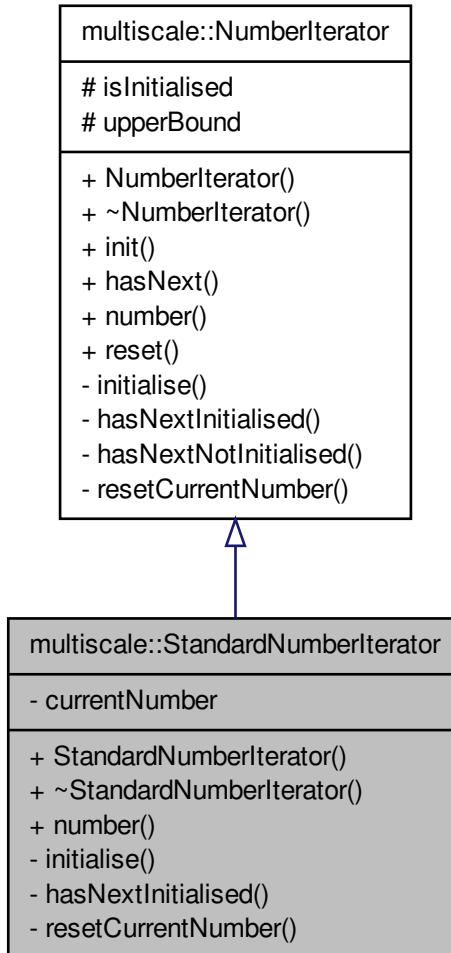
Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.

```
#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](#)

7.149.1 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.

7.149.2 Constructor & Destructor Documentation

7.149.2.1 StandardNumberIterator::StandardNumberIterator (`unsigned int upperBound`)

Definition at line 6 of file StandardNumberIterator.cpp.

References initialise(), and multiscale::NumberIterator::reset().

7.149.2.2 StandardNumberIterator::~StandardNumberIterator ()

Definition at line 11 of file StandardNumberIterator.cpp.

7.149.3 Member Function Documentation

7.149.3.1 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`.

7.149.3.2 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().

7.149.3.3 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.

7.149.3.4 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.

7.149.4 Member Data Documentation

7.149.4.1 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:

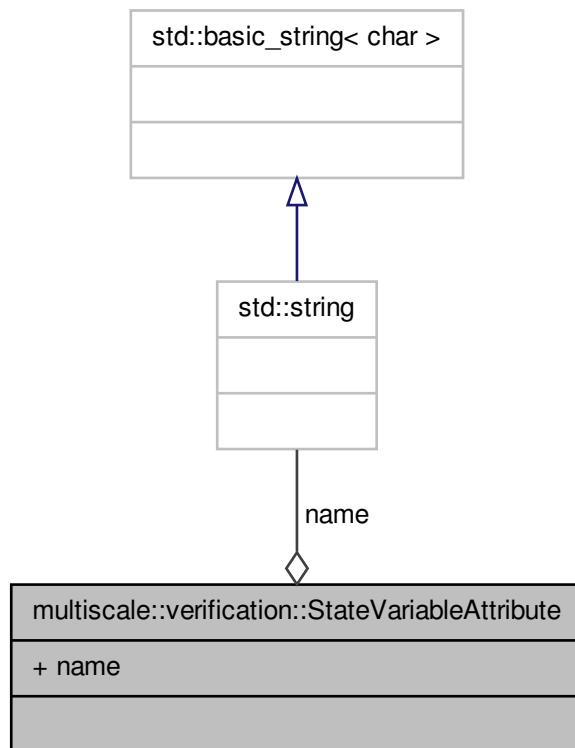
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/iterator/-
[StandardNumberIterator.hpp](#)
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/iterator/-
[StandardNumberIterator.cpp](#)

7.150 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`

7.150.1 Detailed Description

Class for representing a state variable attribute.

Definition at line 13 of file StateVariableAttribute.hpp.

7.150.2 Member Data Documentation

7.150.2.1 std::string multiscale::verification::StateVariableAttribute::name

Name of the state variable

Definition at line 17 of file StateVariableAttribute.hpp.

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

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

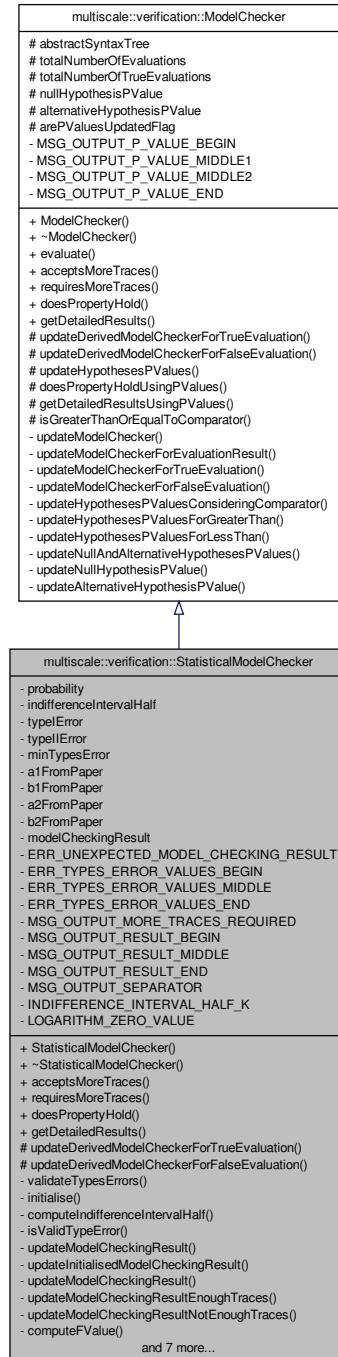
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[StateVariableAttribute.hpp](#)

7.151 multiscale::verification::StatisticalModelChecker Class - Reference

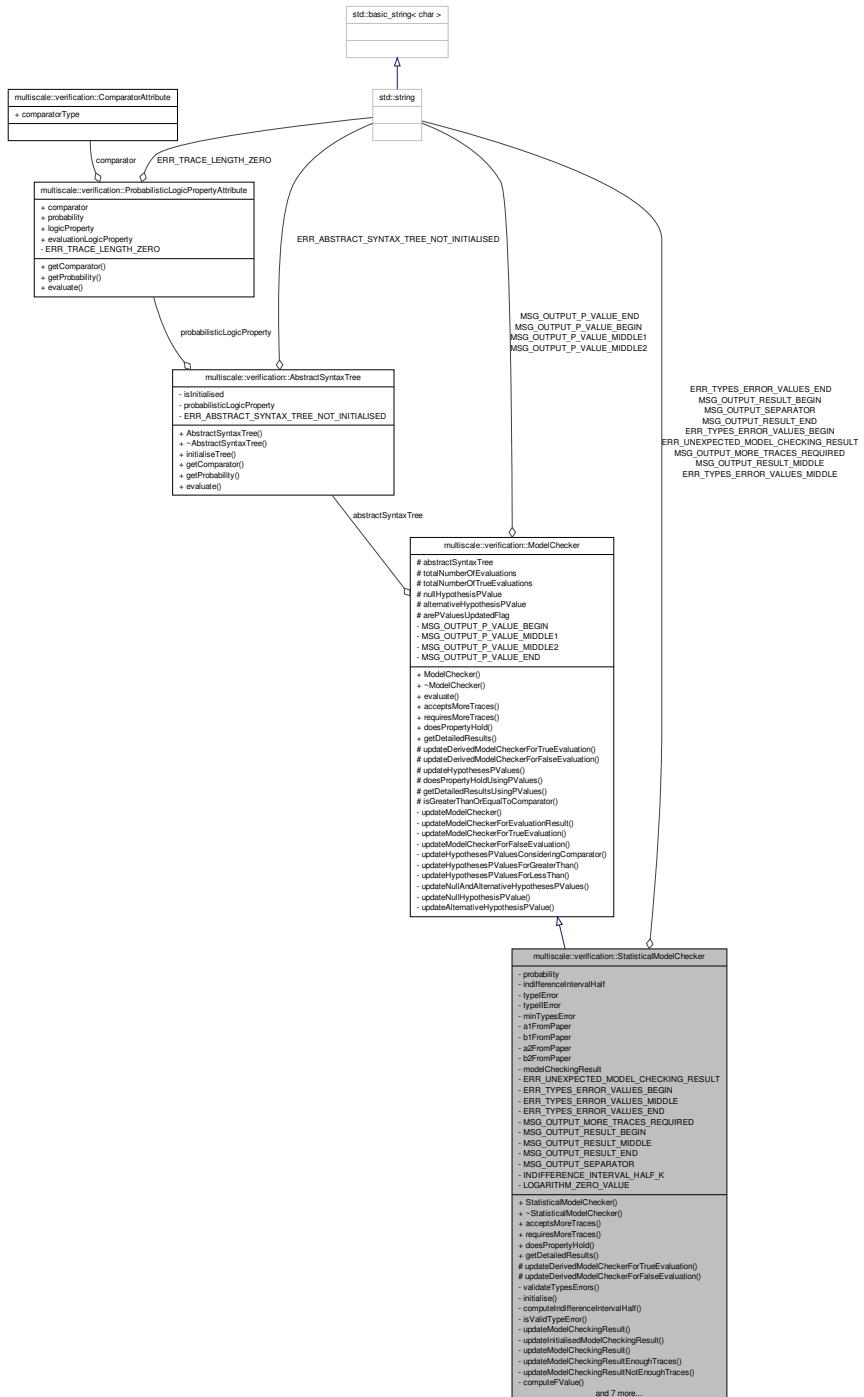
Class used to run statistical model checking tasks.

```
#include <StatisticalModelChecker.hpp>
```

Inheritance diagram for multiscale::verification::StatisticalModelChecker:



Collaboration diagram for multiscale::verification::StatisticalModelChecker:



Public Member Functions

- `StatisticalModelChecker (const AbstractSyntaxTree &abstractSyntaxTree, double typeIError, double typeIIError)`
Check if more traces are accepted for evaluating the logic property.
- `~StatisticalModelChecker ()`
- `bool acceptsMoreTraces () override`
Check if more traces are required for evaluating the logic property.
- `bool requiresMoreTraces () override`
Check if the given property holds.
- `bool doesPropertyHold () override`
Get the detailed description of the results.
- `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 validateTypeErrors (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 isNullHypothesis-True)`
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.

7.151.1 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 = θ`

`indifference = δ`

`typeIError = α`

`typeIIError = β`

`minTypesError = γ`

`totalNumberOfEvaluations = n`

`totalNumberOfTrueEvaluations = d`

Definition at line 49 of file StatisticalModelChecker.hpp.

7.151.2 Constructor & Destructor Documentation

7.151.2.1 StatisticalModelChecker::StatisticalModelChecker (const AbstractSyntaxTree & *abstractSyntaxTree*, double *typeIError*, double *typeIIError*)

Definition at line 13 of file StatisticalModelChecker.cpp.

References `initialise()`, `minTypesError`, `typeIError`, `typeIIError`, and `validateTypesErrors()`.

7.151.2.2 `StatisticalModelChecker::~StatisticalModelChecker()`

Definition at line 26 of file `StatisticalModelChecker.cpp`.

7.151.3 Member Function Documentation

7.151.3.1 `bool StatisticalModelChecker::acceptsMoreTraces() [override, virtual]`

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

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

Definition at line 28 of file `StatisticalModelChecker.cpp`.

References `modelCheckingResult`, `multiscale::verification::MORE_TRACES_REQUIRED`, and `updateModelCheckingResult()`.

Referenced by `requiresMoreTraces()`.

7.151.3.2 `double StatisticalModelChecker::computeFPrimeValue() [private]`

Compute the value of f' (from original paper)

Definition at line 161 of file `StatisticalModelChecker.cpp`.

References `computeFPrimeValueFirstTerm()`, `computeFPrimeValueSecondTerm()`, `multiscale::verification::ModelChecker::totalNumberOfEvaluations`, and `multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations`.

Referenced by `updateInitialisedModelCheckingResult()`.

7.151.3.3 `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 171 of file `StatisticalModelChecker.cpp`.

References `multiscale::Numeric::almostEqual()`, `indifferenceIntervalHalf`, `LOGARITHM_ZERO_VALUE`, and `probability`.

Referenced by `computeFPrimeValue()`.

7.151.3.4 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 180 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM_ZERO_VALUE, and probability.

Referenced by computeFPrimeValue().

7.151.3.5 double StatisticalModelChecker::computeFValue() [private]

Compute the value of f (from original paper)

Definition at line 133 of file StatisticalModelChecker.cpp.

References computeFValueFirstTerm(), computeFValueSecondTerm(), multiscale::verification::ModelChecker::totalNumberOfEvaluations, and multiscale::verification::ModelChecker::totalNumberOfTrueEvaluations.

Referenced by updateInitialisedModelCheckingResult().

7.151.3.6 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 143 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM_ZERO_VALUE, and probability.

Referenced by computeFValue().

7.151.3.7 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 152 of file StatisticalModelChecker.cpp.

References multiscale::Numeric::almostEqual(), indifferenceIntervalHalf, LOGARITHM_ZERO_VALUE, and probability.

Referenced by computeFValue().

7.151.3.8 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 75 of file StatisticalModelChecker.cpp.

References INDIFFERENCE_INTERVAL_HALF_K.

Referenced by initialise().

7.151.3.9 bool StatisticalModelChecker::doesPropertyHold () [override, virtual]

Check if the given property holds.

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

Definition at line 38 of file StatisticalModelChecker.cpp.

References doesPropertyHoldConsideringResult(), and updateModelCheckingResult().

7.151.3.10 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 208 of file StatisticalModelChecker.cpp.

References multiscale::verification::ModelChecker::isGreaterThanOrEqualToComparator().

Referenced by `doesPropertyHoldConsideringResult()`.

7.151.3.11 bool StatisticalModelChecker::doesPropertyHoldConsideringResult() [private]

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

Definition at line 189 of file `StatisticalModelChecker.cpp`.

References `doesPropertyHoldConsideringProbabilityComparator()`, `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()`.

7.151.3.12 std::string StatisticalModelChecker::getDetailedResults() [override, virtual]

Get the detailed description of the results.

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

Definition at line 44 of file `StatisticalModelChecker.cpp`.

References `getDetailedUpdatedResults()`, and `updateModelCheckingResult()`.

7.151.3.13 std::string StatisticalModelChecker::getDetailedUpdatedResults() [private]

Get the detailed description of the updated results.

Definition at line 216 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()`.

7.151.3.14 void StatisticalModelChecker::initialise() [private]

Initialisation of some of the class members.

Definition at line 65 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()`.

7.151.3.15 bool StatisticalModelChecker::isValidTypeIError (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

<code>typeError</code>	The probability of a type I/II error to occur
------------------------	---

Definition at line 82 of file `StatisticalModelChecker.cpp`.

Referenced by `validateTypeErrors()`.

7.151.3.16 bool StatisticalModelChecker::requiresMoreTraces () [override, virtual]

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

Implements `multiscale::verification::ModelChecker`.

Definition at line 34 of file `StatisticalModelChecker.cpp`.

References `acceptsMoreTraces()`.

7.151.3.17 void StatisticalModelChecker::updateDerivedModelChecker-ForFalseEvaluation () [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 52 of file `StatisticalModelChecker.cpp`.

7.151.3.18 void StatisticalModelChecker::updateDerivedModelChecker-ForTrueEvaluation () [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 50 of file StatisticalModelChecker.cpp.

7.151.3.19 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 97 of file StatisticalModelChecker.cpp.

References [computeFPrimeValue\(\)](#), [computeFValue\(\)](#), and [updateModelCheckingResult\(\)](#).

Referenced by [updateModelCheckingResult\(\)](#).

7.151.3.20 void StatisticalModelChecker::updateModelCheckingResult () [private]

Update the result of the model checking task.

Definition at line 89 of file StatisticalModelChecker.cpp.

References [modelCheckingResult](#), [multiscale::verification::UNDECIDED](#), and [updateInitialisedModelCheckingResult\(\)](#).

Referenced by [acceptsMoreTraces\(\)](#), [doesPropertyHold\(\)](#), [getDetailedResults\(\)](#), and [updateInitialisedModelCheckingResult\(\)](#).

7.151.3.21 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 f (from the original paper)
<i>fPrime</i>	The value of f' (from the original paper)

Definition at line 104 of file StatisticalModelChecker.cpp.

References [a1FromPaper](#), [a2FromPaper](#), [b1FromPaper](#), [b2FromPaper](#), [updateModelCheckingResultEnoughTraces\(\)](#), and [updateModelCheckingResultNotEnoughTraces\(\)](#).

7.151.3.22 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 <i>f'</i> (from the original paper)

Definition at line 117 of file StatisticalModelChecker.cpp.

References a1FromPaper, a2FromPaper, b1FromPaper, b2FromPaper, multiscale::verification::FALSE, indifferenceIntervalHalf, modelCheckingResult, multiscale::verification::TRUE, and multiscale::verification::UNDECIDED.

Referenced by updateModelCheckingResult().

7.151.3.23 void **StatisticalModelChecker::updateModelCheckingResultNotEnoughTraces** () [private]

Update the result of the model checking task when not enough traces were provided.

Definition at line 129 of file StatisticalModelChecker.cpp.

References modelCheckingResult, and multiscale::verification::MORE_TRACES_REQUIRED.

Referenced by updateModelCheckingResult().

7.151.3.24 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 54 of file StatisticalModelChecker.cpp.

References ERR_TYPES_ERROR_VALUES_BEGIN, ERR_TYPES_ERROR_VALUES_END, ERR_TYPES_ERROR_VALUES_MIDDLE, isValidTypeError(), MS_throw, and multiscale::StringManipulator::toString().

Referenced by StatisticalModelChecker().

7.151.4 Member Data Documentation

7.151.4.1 **double multiscale::verification::StatisticalModelChecker::a1FromPaper**
[private]

The variable A1 (from the original paper)

Definition at line 63 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.151.4.2 **double multiscale::verification::StatisticalModelChecker::a2FromPaper**
[private]

The variable A2 (from the original paper)

Definition at line 65 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.151.4.3 **double multiscale::verification::StatisticalModelChecker::b1FromPaper**
[private]

The variable B1 (from the original paper)

Definition at line 64 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.151.4.4 **double multiscale::verification::StatisticalModelChecker::b2FromPaper**
[private]

The variable B2 (from the original paper)

Definition at line 66 of file StatisticalModelChecker.hpp.

Referenced by initialise(), updateModelCheckingResult(), and updateModelCheckingResultEnoughTraces().

7.151.4.5 **const std::string StatisticalModelChecker::ERR_TYPES_ERROR_VALUE-S_BEGIN = "The provided probabilities of type I and type II errors ("** [static,
private]

Definition at line 213 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

```
7.151.4.6 const std::string StatisticalModelChecker::ERR_TYPES_ERROR_VALUE-
S_END = ") should be greater than zero and less or equal to 1. Please change."
[static, private]
```

Definition at line 215 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

```
7.151.4.7 const std::string StatisticalModelChecker::ERR_TYP-
ES_ERROR_VALUES_MIDDLE = "," [static,
private]
```

Definition at line 214 of file StatisticalModelChecker.hpp.

Referenced by validateTypesErrors().

```
7.151.4.8 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 211 of file StatisticalModelChecker.hpp.

Referenced by doesPropertyHoldConsideringResult().

```
7.151.4.9 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 225 of file StatisticalModelChecker.hpp.

Referenced by computeIndifferenceIntervalHalf().

```
7.151.4.10 double multiscale::verification::StatisticalModelChecker::indifference-
IntervalHalf [private]
```

Half of the size of the indifference interval

Definition at line 56 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(),
computeFValueFirstTerm(), computeFValueSecondTerm(), initialise(), and update-
ModelCheckingResultEnoughTraces().

```
7.151.4.11 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 227 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(), computeFValueFirstTerm(), and computeFValueSecondTerm().

7.151.4.12 `double multiscale::verification::StatisticalModelChecker::minTypes-Error` [private]

The minimum probability of type I and type II errors to occur

Definition at line 61 of file StatisticalModelChecker.hpp.

Referenced by initialise(), and StatisticalModelChecker().

7.151.4.13 `StatisticalModelCheckingResult multiscale::verification-::StatisticalModelChecker::modelCheckingResult` [private]

The result of the model checking task

Definition at line 68 of file StatisticalModelChecker.hpp.

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

7.151.4.14 `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 217 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

7.151.4.15 `const std::string StatisticalModelChecker::MSG_OUTPUT_RESULT_B-EGIN` = "The provided answer is given for the probability of type I errors = " [static, private]

Definition at line 219 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

```
7.151.4.16 const std::string StatisticalModelChecker::MSG_OUTPUT_RESULT_END  
= "" [static, private]
```

Definition at line 221 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

```
7.151.4.17 const std::string StatisticalModelChecker::MSG_OUTPUT_RESU-  
LT_MIDDLE = " and the probability of type II errors = " [static,  
private]
```

Definition at line 220 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

```
7.151.4.18 const std::string StatisticalModelChecker::MSG_OUTPUT_SEPARATOR =  
" " [static, private]
```

Definition at line 223 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults().

```
7.151.4.19 double multiscale::verification::StatisticalModelChecker::probability  
[private]
```

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

Definition at line 53 of file StatisticalModelChecker.hpp.

Referenced by computeFPrimeValueFirstTerm(), computeFPrimeValueSecondTerm(),
computeFValueFirstTerm(), computeFValueSecondTerm(), and initialise().

```
7.151.4.20 double multiscale::verification::StatisticalModelChecker::typeIError  
[private]
```

The probability of type I errors to occur

Definition at line 58 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults(), initialise(), and StatisticalModelChecker().

```
7.151.4.21 double multiscale::verification::StatisticalModelChecker::typeIIError  
[private]
```

The probability of type II errors to occur

Definition at line 59 of file StatisticalModelChecker.hpp.

Referenced by getDetailedUpdatedResults(), initialise(), and StatisticalModelChecker().

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

[7.152 multiscale::verification::StatisticalModelCheckerFactory Class Reference](#)

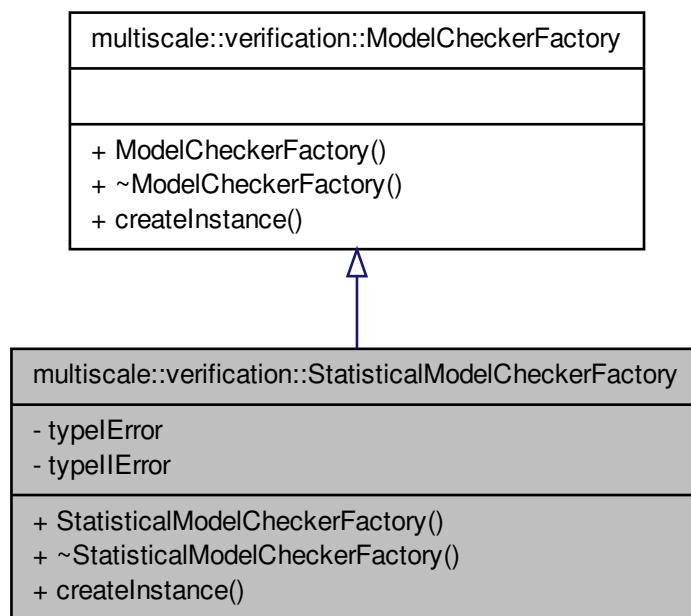
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/[StatisticalModelChecker.hpp](#)
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[StatisticalModelChecker.cpp](#)

[7.152 multiscale::verification::StatisticalModelCheckerFactory - Class Reference](#)

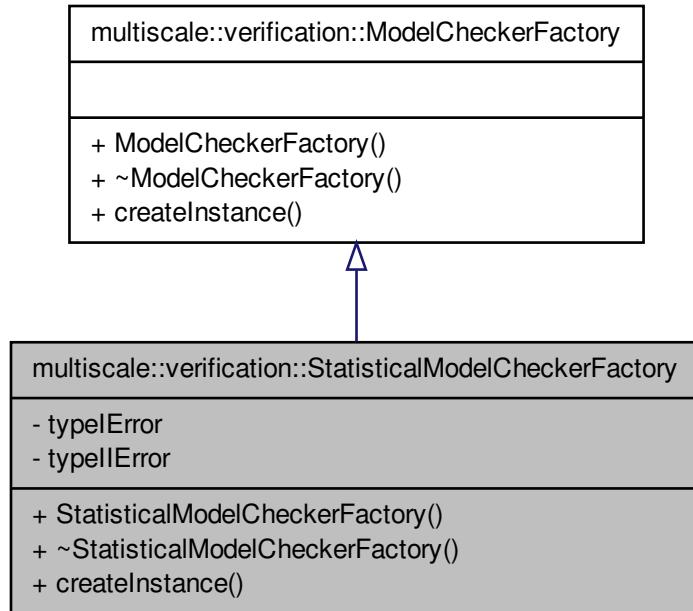
Class for creating [StatisticalModelChecker](#) instances.

```
#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](#)) override

Create an instance of [StatisticalModelChecker](#).

Private Attributes

- double `typeIError`
- double `typeIIError`

7.152.1 Detailed Description

Class for creating [StatisticalModelChecker](#) instances.

Definition at line 12 of file [StatisticalModelCheckerFactory.hpp](#).

7.152.2 Constructor & Destructor Documentation

7.152.2.1 **StatisticalModelCheckerFactory::StatisticalModelCheckerFactory (double typeIError, double typeIIError)**

Definition at line 7 of file StatisticalModelCheckerFactory.cpp.

7.152.2.2 **StatisticalModelCheckerFactory::~StatisticalModelCheckerFactory ()**

Definition at line 12 of file StatisticalModelCheckerFactory.cpp.

7.152.3 Member Function Documentation

7.152.3.1 **std::shared_ptr< ModelChecker > StatisticalModelCheckerFactory-::createInstance (const AbstractSyntaxTree & abstractSyntaxTree) [override, virtual]**

Create an instance of [StatisticalModelChecker](#).

Parameters

<i>abstract-SyntaxTree</i>	The abstract syntax tree representing the logic property to be checked
----------------------------	--

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

Definition at line 15 of file StatisticalModelCheckerFactory.cpp.

7.152.4 Member Data Documentation

7.152.4.1 **double multiscale::verification::StatisticalModelCheckerFactory::typeI-Error [private]**

The probability of a type I error

Definition at line 16 of file StatisticalModelCheckerFactory.hpp.

7.152.4.2 **double multiscale::verification::StatisticalModelCheckerFactory::typeII-Error [private]**

The probability of a type II error

Definition at line 17 of file StatisticalModelCheckerFactory.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/checking/[StatisticalModelCheckerFactory.hpp](#)

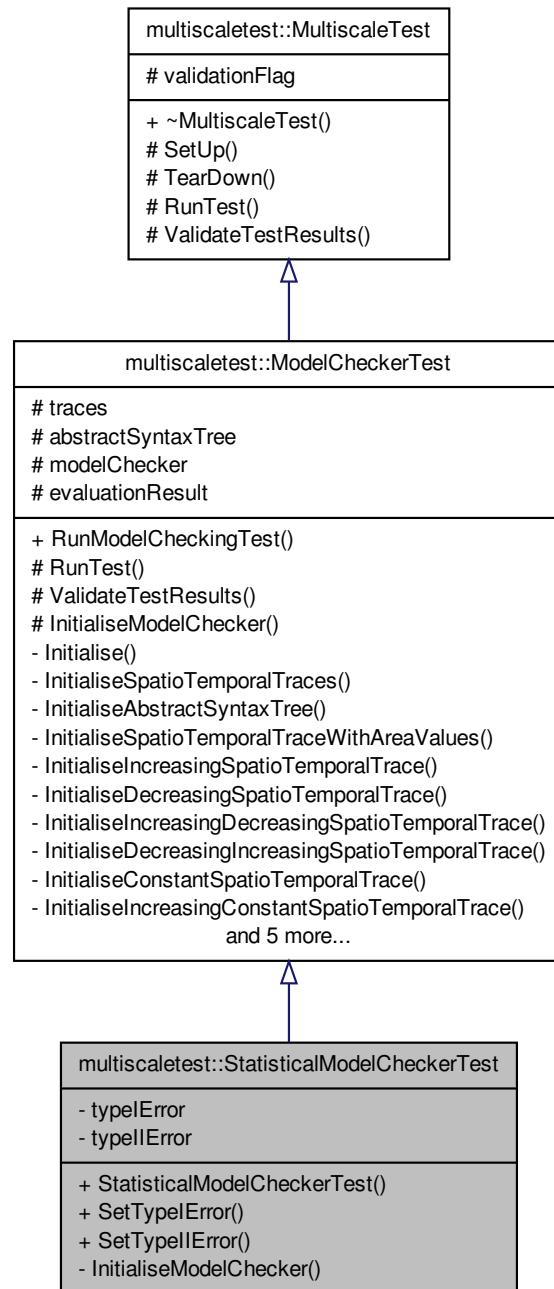
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/[StatisticalModelCheckerFactory.cpp](#)

7.153 multiscaletest::StatisticalModelCheckerTest Class Reference

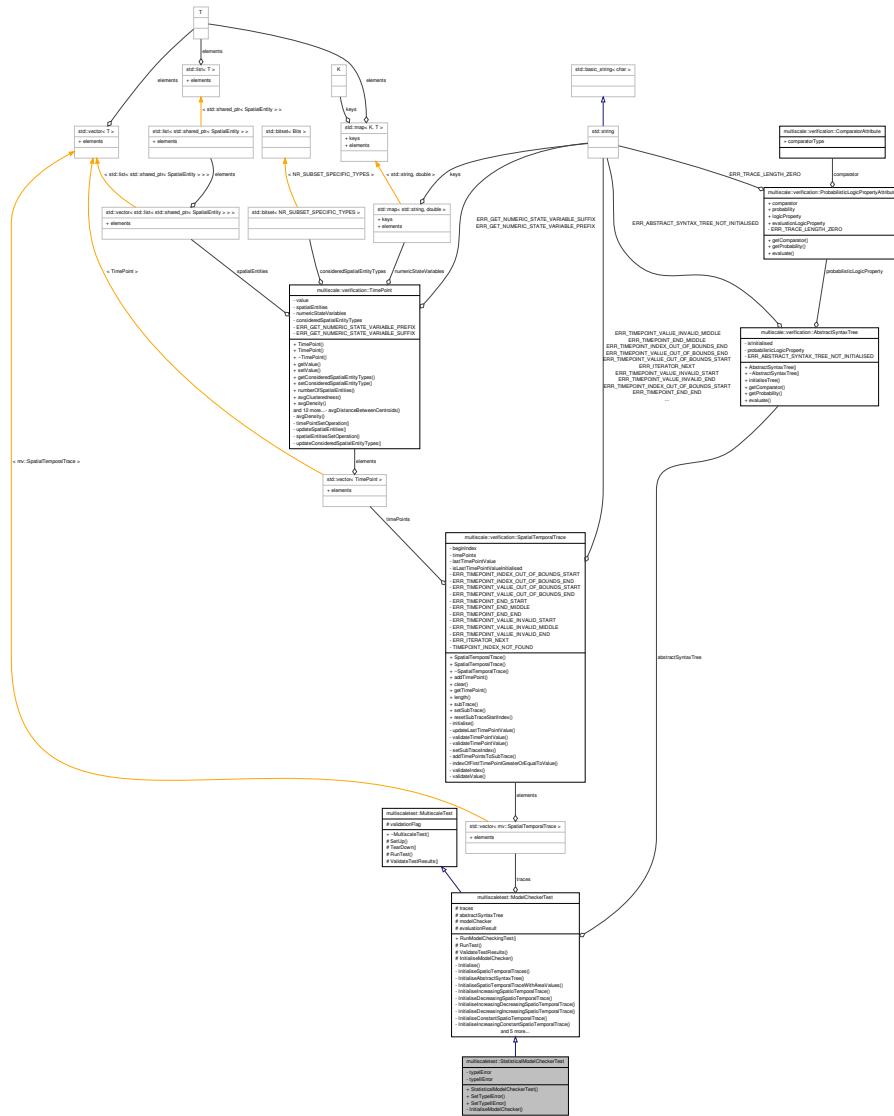
Class for testing the statistical model checker.

```
#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.

Private Member Functions

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

Initialise the model checker.

Private Attributes

- double [typeIError](#)
- double [typeIIError](#)

7.153.1 Detailed Description

Class for testing the statistical model checker.

Definition at line 15 of file StatisticalModelCheckerTest.hpp.

7.153.2 Constructor & Destructor Documentation

7.153.2.1 multiscaletest::StatisticalModelCheckerTest::StatisticalModelCheckerTest() [inline]

Definition at line 24 of file StatisticalModelCheckerTest.hpp.

7.153.3 Member Function Documentation

7.153.3.1 void multiscaletest::StatisticalModelCheckerTest::InitialiseModelChecker() [override, private, virtual]

Initialise the model checker.

Implements [multiscaletest::ModelCheckerTest](#).

Definition at line 55 of file StatisticalModelCheckerTest.hpp.

7.153.3.2 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.

7.153.3.3 void multiscaletest::StatisticalModelCheckerTest::SetTypeIIError (double typeIIError)

Set the value of the type II error.

Parameters

<i>typeIIError</i>	The probability of type II errors occurring
--------------------	---

Definition at line 51 of file StatisticalModelCheckerTest.hpp.

7.153.4 Member Data Documentation

7.153.4.1 double multiscaletest::StatisticalModelCheckerTest::typeIError [private]

The probability of type I errors

Definition at line 19 of file StatisticalModelCheckerTest.hpp.

7.153.4.2 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](#)

7.154 multiscale::StringManipulator Class Reference

Class for manipulating strings.

```
#include <StringManipulator.hpp>
```

Static Public Member Functions

- static string [filenameFromPath](#) (const string &filepath)
Obtain the file name from the given file path.
- static string [replace](#) (const string &initialString, const string &replaceWhat, const string &replaceTo)
Replace a substring of the given string with another string.
- static vector< string > [split](#) (const string &initialString, const string &delimiter)
Split the given string into a vector of strings considering the given delimiter.

- static string [trimRight](#) (string &inputString)
Remove the trailing "new line" characters from the end of the string.
- static string [trimRight](#) (const string &inputString)
Remove the trailing "new line" characters from the end of the string.
- template<typename T >
 static string [toString](#) (T variable)
Convert the variable to a string.

Static Public Attributes

- static const char [DIR_SEPARATOR](#) = '/'

7.154.1 Detailed Description

Class for manipulating strings.

Definition at line 14 of file StringManipulator.hpp.

7.154.2 Member Function Documentation

7.154.2.1 string StringManipulator::filenameFromPath (const string & *filepath*) [static]

Obtain the file name from the given file path.

Parameters

<i>filepath</i>	File path
-----------------	-----------

Definition at line 9 of file StringManipulator.cpp.

References [DIR_SEPARATOR](#).

Referenced by [multiscale::video::PolarGnuplotScriptGenerator::generateHeader\(\)](#), and [multiscale::video::RectangularGnuplotScriptGenerator::generateHeader\(\)](#).

7.154.2.2 string StringManipulator::replace (const string & *initialString*, const string & *replaceWhat*, const string & *replaceTo*) [static]

Replace a substring of the given string with another string.

Parameters

<i>initialString</i>	Initial string
<i>replaceWhat</i>	Substring which will be replaced
<i>replaceTo</i>	String which will be inserted instead of the replaceWhat string

Definition at line 19 of file StringManipulator.cpp.

Referenced by multiscale::video::PolarGnuplotScriptGenerator::outputContent(), multiscale::video::PolarGnuplotScriptGenerator::outputHeader(), and multiscale::video::RectangularGnuplotScriptGenerator::outputHeader().

7.154.2.3 `vector< string > StringManipulator::split (const string & initialString, const string & delimiter) [static]`

Split the given string into a vector of strings considering the given delimiter.

Parameters

<i>initialString</i>	Initial string
<i>delimiter</i>	Delimiter

Definition at line 32 of file StringManipulator.cpp.

Referenced by 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().

7.154.2.4 `template<typename T > static string multiscale::StringManipulator::toString (T variable) [inline, static]`

Convert the variable to a string.

Parameters

<i>variable</i>	Variable
-----------------	----------

Definition at line 58 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::initOutput-

File(), 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().

7.154.2.5 string StringManipulator::trimRight (string & *inputString*) [static]

Remove the trailing "new line" characters from the end of the string.

Parameters

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

Definition at line 39 of file StringManipulator.cpp.

Referenced by multiscale::verification::ModelCheckingOutputWriter::printLogicPropertyWithTag(), multiscale::verification::ModelCheckingOutputWriter::printParsingLogicPropertyMessage(), trimRight(), and multiscale::verification::ParserGrammarExceptionHandler::trimRight().

7.154.2.6 string StringManipulator::trimRight (const string & *inputString*) [static]

Remove the trailing "new line" characters from the end of the string.

Parameters

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

Definition at line 51 of file StringManipulator.cpp.

References trimRight().

7.154.3 Member Data Documentation

7.154.3.1 const char StringManipulator::DIR_SEPARATOR = '/' [static]

Definition at line 69 of file StringManipulator.hpp.

Referenced by filenameFromPath().

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/[String-](#)
[Manipulator.cpp](#)

7.155 multiscale::verification::SubsetAttribute Class Reference

Class for representing a subset attribute.

```
#include <SubsetAttribute.hpp>
```

Public Attributes

- [SubsetAttributeType subset](#)

7.155.1 Detailed Description

Class for representing a subset attribute.

Definition at line 29 of file [SubsetAttribute.hpp](#).

7.155.2 Member Data Documentation

7.155.2.1 SubsetAttributeType multiscale::verification::SubsetAttribute::subset

The subset

Definition at line 33 of file [SubsetAttribute.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/[SubsetAttribute.hpp](#)

7.156 multiscale::verification::SubsetOperationAttribute Class - Reference

Class for representing a subset operation attribute.

```
#include <SubsetOperationAttribute.hpp>
```

Public Attributes

- [SubsetOperationType subsetOperationType](#)

7.156.1 Detailed Description

Class for representing a subset operation attribute.

Definition at line 29 of file [SubsetOperationAttribute.hpp](#).

7.156.2 Member Data Documentation

7.156.2.1 [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/\[SubsetOperationAttribute.hpp\]\(#\)](#)

7.157 multiscale::verification::SubsetOperationTypeParser Struct - Reference

Symbol table and parser for the subset operation type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [SubsetOperationTypeParser \(\)](#)

7.157.1 Detailed Description

Symbol table and parser for the subset operation type.

Definition at line 85 of file [SymbolTables.hpp](#).

7.157.2 Constructor & Destructor Documentation

7.157.2.1 multiscale::verification::SubsetOperationTypeParser::SubsetOperationTypeParser() [inline]

Definition at line 87 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](#)

7.158 multiscale::verification::SubsetSpecificAttribute Class - Reference

Class for representing a subset specific attribute.

```
#include <SubsetSpecificAttribute.hpp>
```

Public Attributes

- [SubsetSpecificType subsetSpecificType](#)

7.158.1 Detailed Description

Class for representing a subset specific attribute.

Definition at line 61 of file SubsetSpecificAttribute.hpp.

7.158.2 Member Data Documentation

7.158.2.1 SubsetSpecificType multiscale::verification::SubsetSpecificAttribute::subsetSpecificType

The specific subset type

Definition at line 65 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](#)

7.159 multiscale::verification::SubsetSpecificTypeParser Struct - Reference

Symbol table and parser for a specific subset type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [SubsetSpecificTypeParser \(\)](#)

7.159.1 Detailed Description

Symbol table and parser for a specific subset type.

Definition at line 73 of file SymbolTables.hpp.

7.159.2 Constructor & Destructor Documentation

7.159.2.1 multiscale::verification::SubsetSpecificTypeParser::SubsetSpecificTypeParser() [inline]

Definition at line 75 of file SymbolTables.hpp.

References multiscale::verification::Clusters, and multiscale::verification::Regions.

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

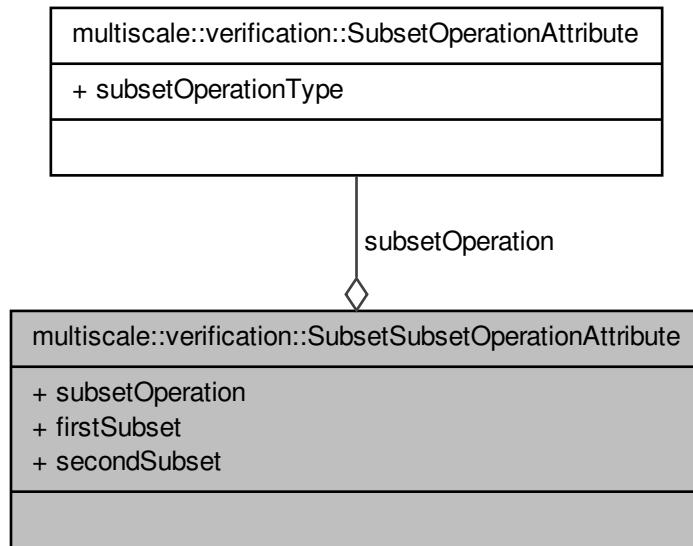
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp](#)

7.160 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`

7.160.1 Detailed Description

Class for representing a subset subset operation attribute.

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

7.160.2 Member Data Documentation

7.160.2.1 [SubsetAttributeType](#) `multiscale::verification::SubsetSubsetOperationAttribute::firstSubset`

The first considered subset

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

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

7.160.2.2 SubsetAttributeType multiscale::verification::SubsetSubsetOperation-Attribute::secondSubset

The second considered subset

Definition at line 21 of file SubsetSubsetOperationAttribute.hpp.

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

7.160.2.3 SubsetOperationAttribute multiscale::verification::SubsetSubset-OperationAttribute::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:

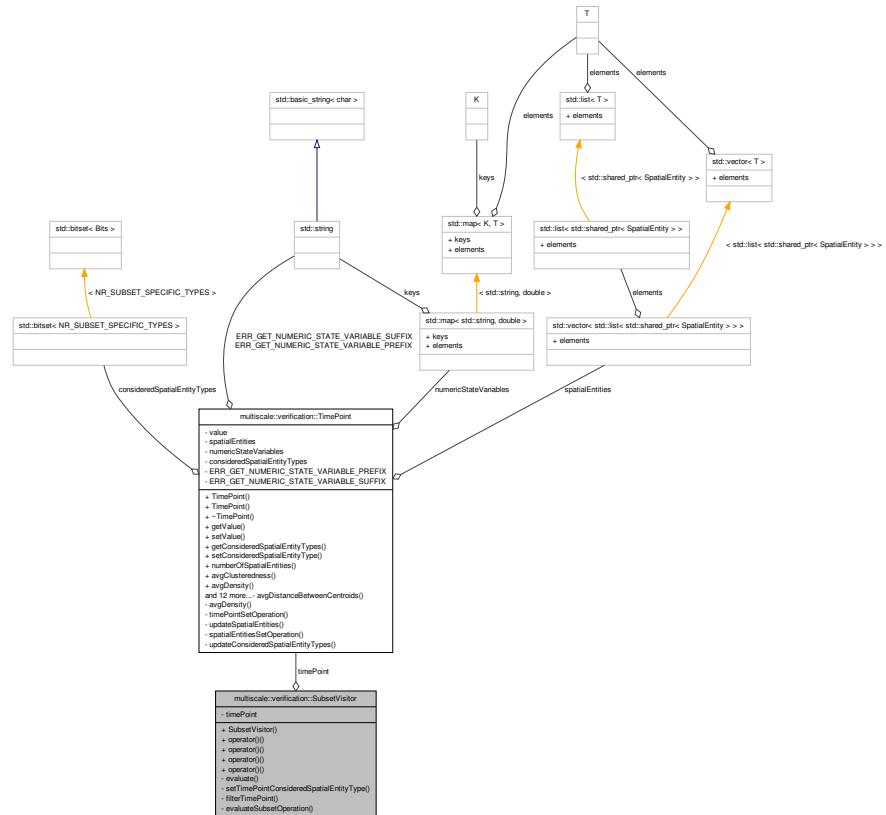
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[SubsetSubset-OperationAttribute.hpp](#)

7.161 multiscale::verification::SubsetVisitor Class Reference

Class used to evaluate subsets.

```
#include <SubsetVisitor.hpp>
```

Collaboration diagram for multiscale::verification::SubsetVisitor:



Public Member Functions

- `SubsetVisitor` (`const TimePoint &timePoint`)
 - `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 `ConstraintAttribute-Type &constraint`) const
Filter the given timepoint considering the provided constraint.

- `TimePoint evaluateSubsetOperation` (const `SubsetOperationType &subset-Operation`, const `TimePoint &firstSubsetTimePoint`, const `TimePoint &second-SubsetTimePoint`) const
Evaluate subsetOperation against the given subsets timepoints.

Private Attributes

- const `TimePoint & timePoint`

7.161.1 Detailed Description

Class used to evaluate subsets.

Definition at line 14 of file SubsetVisitor.hpp.

7.161.2 Constructor & Destructor Documentation

7.161.2.1 multiscale::verification::SubsetVisitor::SubsetVisitor (const `TimePoint & timePoint`) [inline]

Definition at line 22 of file SubsetVisitor.hpp.

Referenced by `evaluate()`.

7.161.3 Member Function Documentation

7.161.3.1 `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 77 of file SubsetVisitor.hpp.

References `SubsetVisitor()`.

Referenced by `operator()()`.

7.161.3.2 `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>subset-Operation</i>	The considered subset operation
<i>firstSubset-TimePoint</i>	The timepoint corresponding to the first subset
<i>second-SubsetTime-Point</i>	The timepoint corresponding to the second subset

Definition at line 107 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()()`.

7.161.3.3 `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 96 of file `SubsetVisitor.hpp`.

Referenced by `operator()()`.

7.161.3.4 `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 28 of file SubsetVisitor.hpp.

References evaluate(), multiscale::verification::SubsetAttribute::subset, and timePoint.

**7.161.3.5 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 36 of file SubsetVisitor.hpp.

References setTimePointConsideredSpatialEntityType(), multiscale::verification::-
SubsetSpecificAttribute::subsetSpecificType, and timePoint.

**7.161.3.6 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 48 of file SubsetVisitor.hpp.

References multiscale::verification::FilterSubsetAttribute::constraint, filterTimePoint(),
setTimePointConsideredSpatialEntityType(), multiscale::verification::FilterSubset-
Attribute::subsetSpecific, multiscale::verification::SubsetSpecificAttribute::subset-
SpecificType, and timePoint.

**7.161.3.7 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 61 of file SubsetVisitor.hpp.

References evaluate(), evaluateSubsetOperation(), multiscale::verification::SubsetSubsetOperationAttribute::firstSubset, multiscale::verification::SubsetSubsetOperationAttribute::secondSubset, multiscale::verification::SubsetSubsetOperationAttribute::subsetOperation, multiscale::verification::SubsetOperationAttribute::subsetOperationType, and timePoint.

7.161.3.8 void multiscale::verification::SubsetVisitor::setTimePointConsidered-SpatialEntityType (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 86 of file SubsetVisitor.hpp.

References multiscale::verification::TimePoint::setConsideredSpatialEntityType().

Referenced by operator()().

7.161.4 Member Data Documentation

7.161.4.1 const TimePoint& multiscale::verification::SubsetVisitor::timePoint [private]

The initial timepoint

Definition at line 18 of file SubsetVisitor.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/[SubsetVisitor.hpp](#)

7.162 multiscale::SubtractionOperation Class Reference

Functor representing a subtraction operation.

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

Public Member Functions

- template<typename Operand >
Operand **operator()** (Operand operand1, Operand operand2) const
Subtract the two operands.

7.162.1 Detailed Description

Functor representing a subtraction operation.

Definition at line 69 of file Numeric.hpp.

7.162.2 Member Function Documentation

7.162.2.1 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 79 of file Numeric.hpp.

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

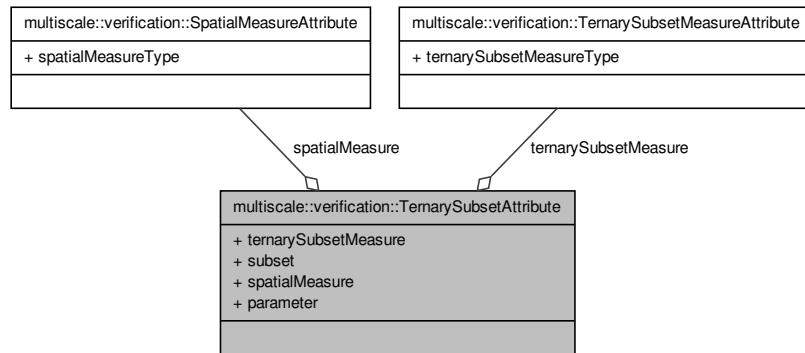
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
[Numeric.hpp](#)

7.163 multiscale::verification::TernarySubsetAttribute Class - Reference

Class for representing a ternary subset attribute.

```
#include <TernarySubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::TernarySubsetAttribute:



Public Attributes

- **TernarySubsetMeasureAttribute** `ternarySubsetMeasure`
- **SubsetAttributeType** `subset`
- **SpatialMeasureAttribute** `spatialMeasure`
- double `parameter`

7.163.1 Detailed Description

Class for representing a ternary subset attribute.

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

7.163.2 Member Data Documentation

7.163.2.1 double multiscale::verification::TernarySubsetAttribute::parameter

The considered parameter

Definition at line 23 of file `TernarySubsetAttribute.hpp`.

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

7.163.2.2 SpatialMeasureAttribute multiscale::verification::TernarySubsetAttribute::spatialMeasure

The considered spatial measure

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

7.164 multiscale::verification::TernarySubsetMeasureAttribute Class Reference

987

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

7.163.2.3 SubsetAttributeType multiscale::verification::TernarySubsetAttribute- ::subset

The considered subset

Definition at line 21 of file TernarySubsetAttribute.hpp.

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

7.163.2.4 TernarySubsetMeasureAttribute multiscale::verification::Ternary- SubsetAttribute::ternarySubsetMeasure

The ternary subset measure

Definition at line 20 of file TernarySubsetAttribute.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/[TernarySubsetAttribute.hpp](#)

7.164 multiscale::verification::TernarySubsetMeasureAttribute - Class Reference

Class for representing a ternary subset measure attribute.

```
#include <TernarySubsetMeasureAttribute.hpp>
```

Public Attributes

- [TernarySubsetMeasureType ternarySubsetMeasureType](#)

7.164.1 Detailed Description

Class for representing a ternary subset measure attribute.

Definition at line 28 of file TernarySubsetMeasureAttribute.hpp.

7.164.2 Member Data Documentation

7.164.2.1 **TernarySubsetMeasureType multiscale::verification::TernarySubsetMeasureAttribute::ternarySubsetMeasureType**

The ternary subset measure type

Definition at line 32 of file `TernarySubsetMeasureAttribute.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/TernarySubsetMeasureAttribute.hpp`

7.165 **multiscale::verification::TernarySubsetMeasureTypeParser - Struct Reference**

Symbol table and parser for the ternary subset measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [TernarySubsetMeasureTypeParser \(\)](#)

7.165.1 Detailed Description

Symbol table and parser for the ternary subset measure type.

Definition at line 143 of file `SymbolTables.hpp`.

7.165.2 Constructor & Destructor Documentation

7.165.2.1 **multiscale::verification::TernarySubsetMeasureTypeParser::TernarySubsetMeasureTypeParser () [inline]**

Definition at line 145 of file `SymbolTables.hpp`.

References `multiscale::verification::Percentile`, and `multiscale::verification::Quartile`.

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

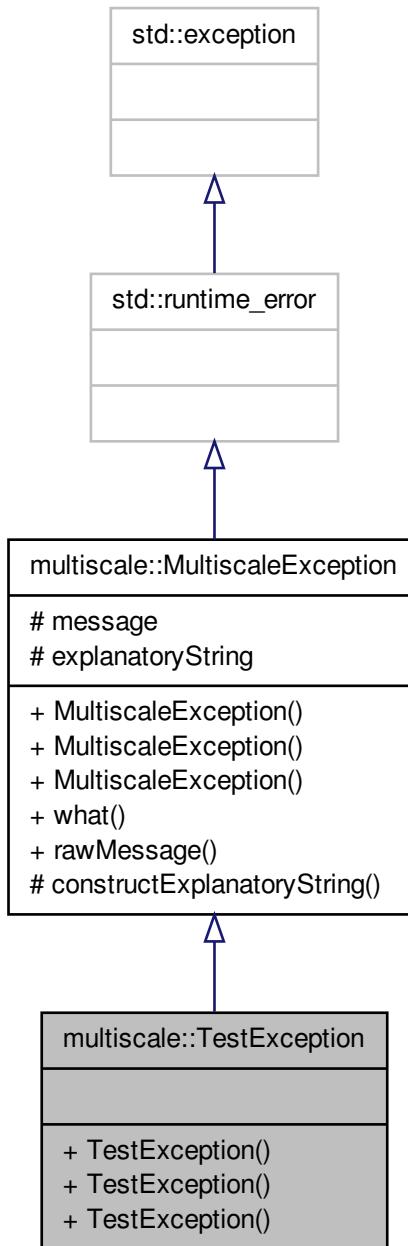
- `/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp`

7.166 multiscale::TestException Class Reference

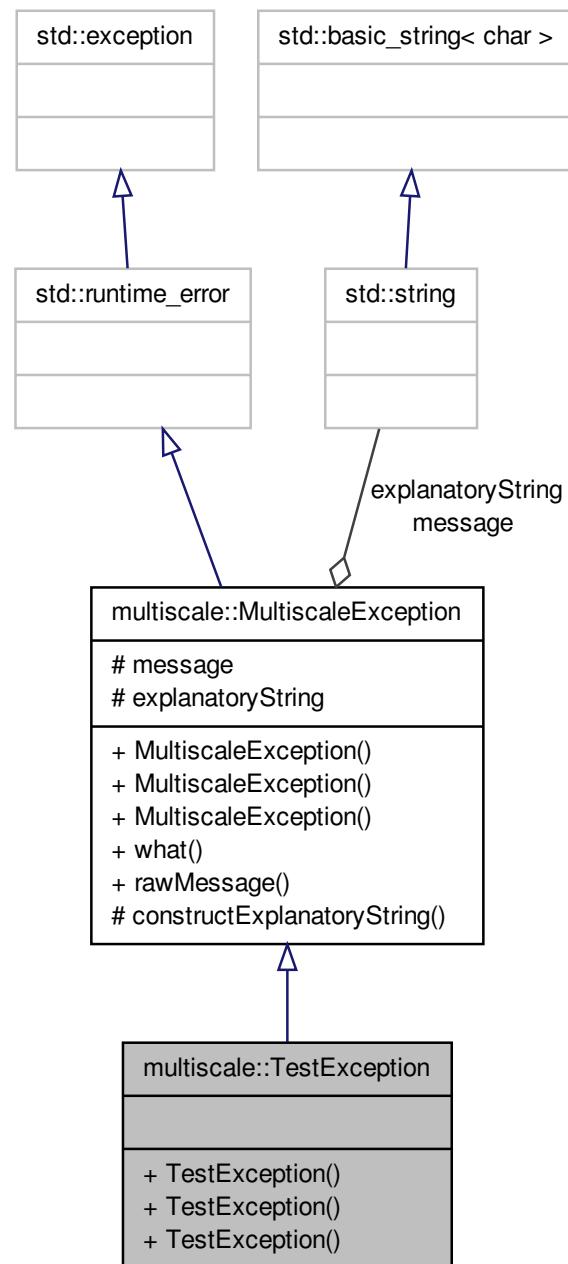
Class for representing testing exceptions.

```
#include <TestException.hpp>
```

Inheritance diagram for multiscale::TestException:



Collaboration diagram for multiscale::TestException:



Public Member Functions

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

7.166.1 Detailed Description

Class for representing testing exceptions.

Definition at line 14 of file `TestException.hpp`.

7.166.2 Constructor & Destructor Documentation

7.166.2.1 multiscale::TestException::TestException() [inline]

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

7.166.2.2 multiscale::TestException::TestException(const string & file, int line, const string & msg) [inline, explicit]

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

7.166.2.3 multiscale::TestException::TestException(const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 of file `TestException.hpp`.

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

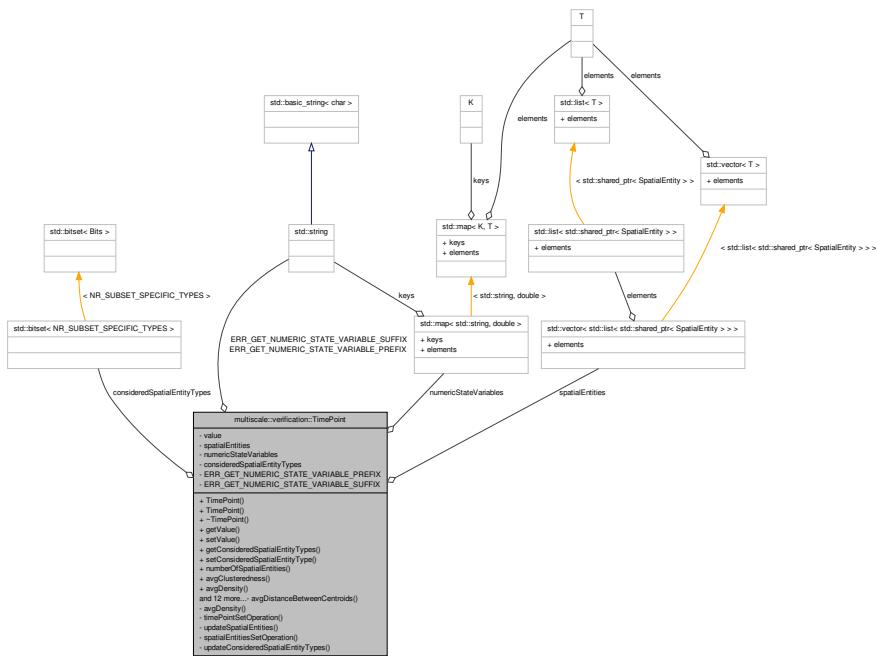
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-
`TestException.hpp`](#)

7.167 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::set<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.
- `double avgClusteredness () const`

Get the clusteredness of the considered collection of spatial entities.
- `double avgDensity () const`

Get the density of the considered collection of 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 std::string &name, double value)`

Add a numeric state variable to the map.
- `bool existsNumericStateVariable (const std::string &name)`

Check if the numeric state variable with the given name 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 std::string &name) const`

Get the value of the numeric state variable with the given name if it exists and throw an exception otherwise.
- `void timePointDifference (const TimePoint &timePoint)`

Compute the difference of this timepoint and the given timepoint.
- `void timePointIntersection (const TimePoint &timePoint)`

Compute the intersection of this timepoint and the given timepoint.
- `void timePointUnion (const TimePoint &timePoint)`

Compute the union of this timepoint and the given timepoint.
- `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

- `double avgDistanceBetweenCentroids (const std::vector< std::shared_ptr< - SpatialEntity >> &spatialEntities) const`

Compute the average Euclidean distance between the centroids of the given collection of spatial entities.
- `double avgDensity (const std::vector< std::shared_ptr< SpatialEntity >> &spatialEntities) const`

Compute the density of the given collection of spatial entities.

- void `timePointSetOperation` (const `TimePoint` &timePoint, const `SetOperationType` &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 `SetOperationType` &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 `SetOperationType` &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 `SetOperationType` &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< std::string, 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 identified by the given name ("")
- static const std::string `ERR_GET_NUMERIC_STATE_VARIABLE_SUFFIX` = ")"
does not exist."

7.167.1 Detailed Description

Class for representing a timepoint.

Definition at line 30 of file TimePoint.hpp.

7.167.2 Constructor & Destructor Documentation

7.167.2.1 **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.

7.167.2.2 **TimePoint::TimePoint (const TimePoint & *timePoint*)**

Definition at line 20 of file TimePoint.cpp.

7.167.2.3 **TimePoint::~TimePoint ()**

Definition at line 24 of file TimePoint.cpp.

7.167.3 Member Function Documentation

7.167.3.1 **void TimePoint::addNumericStateVariable (const std::string & *name*, double *value*)**

Add a numeric state variable to the map.

If a numeric state variable with the same name exists then the value of the existing numeric state variable will be replaced by the provided new value.

Parameters

<i>name</i>	The name of the numeric state variable
<i>value</i>	The value of the numeric state variable

Definition at line 82 of file TimePoint.cpp.

References numericStateVariables, and value.

Referenced by multiscale::verification::SpatialTemporalDataReader::addNumericStateVariableToTimePoint().

7.167.3.2 **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 73 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::SpatialTemporalDataReader::addSpatialEntityToTimePoint(), and multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues().

7.167.3.3 double TimePoint::avgClusteredness () const

Get the clusteredness of the considered collection of spatial entities.

Definition at line 57 of file TimePoint.cpp.

References avgDistanceBetweenCentroids(), and getConsideredSpatialEntities().

Referenced by multiscale::verification::NumericEvaluator::evaluate().

7.167.3.4 double TimePoint::avgDensity () const

Get the density of the considered collection of spatial entities.

Definition at line 63 of file TimePoint.cpp.

References avgDistanceBetweenCentroids(), and getConsideredSpatialEntities().

Referenced by multiscale::verification::NumericEvaluator::evaluate().

7.167.3.5 double TimePoint::avgDensity (const std::vector< std::shared_ptr< SpatialEntity >> & spatialEntities) const [private]

Compute the density of the given collection of spatial entities.

Parameters

<i>spatial- Entities</i>	The collection of considered spatial entities
------------------------------	---

Definition at line 193 of file TimePoint.cpp.

References spatialEntities.

7.167.3.6 double TimePoint::avgDistanceBetweenCentroids (const std::vector< std::shared_ptr< SpatialEntity >> & spatialEntities) const [private]

Compute the average Euclidean distance between the centroids of the given collection of spatial entities.

The average Euclidean distance between one centroid c_1 and all other centroids is computed as below: $AED(c_1) = \sum_{c \in \text{centroids}} \frac{\text{distance}(c, c_1)}{|\text{centroids}|}$.

The average Euclidean distance between all centroids is computed as below: $AEDC = \sum_{c \in \text{centroids}} \frac{AED(c)}{|\text{centroids}|}$.

Parameters

<i>spatial- Entities</i>	The collection of considered spatial entities
------------------------------	---

Definition at line 178 of file TimePoint.cpp.

References multiscale::Geometry2D::distanceBtwPoints(), and spatialEntities.

Referenced by avgClusteredness(), and avgDensity().

7.167.3.7 bool TimePoint::existsNumericStateVariable (const std::string & name)

Check if the numeric state variable with the given name exists.

Parameters

<i>name</i>	The name of the numeric state variable
-------------	--

Definition at line 86 of file TimePoint.cpp.

References numericStateVariables.

**7.167.3.8 std::vector< std::shared_ptr< SpatialEntity > >
TimePoint::getConsideredSpatialEntities () const**

Get the collection of considered spatial entities.

Definition at line 134 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::NR_SUBSET_SPECIFIC_TYPES, and spatialEntities.

Referenced by avgClusteredness(), avgDensity(), and multiscale::verification::TimePointEvaluator::getSpatialMeasureValues().

**7.167.3.9 std::bitset< NR_SUBSET_SPECIFIC_TYPES >
TimePoint::getConsideredSpatialEntityTypes ()**

Get the considered spatial entity type.

Definition at line 34 of file TimePoint.cpp.

References consideredSpatialEntityTypes.

Referenced by multiscale::verification::ConstraintEvaluator::evalSpatialMeasureConstraint(), and multiscale::verification::ConstraintEvaluator::evalTypeConstraint().

7.167.3.10 double TimePoint::getNumericStateVariable (const std::string & *name*)
const

Get the value of the numeric state variable with the given name if it exists and throw an exception otherwise.

Parameters

<i>name</i>	The name of the numeric state variable
-------------	--

Definition at line 147 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::NumericVisitor::operator()().

7.167.3.11 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 90 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtType(), multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues(), printSpatialEntities(), and spatialEntitiesSetOperation().

7.167.3.12 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 103 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

```
7.167.3.13 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 116 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtType(), printSpatialEntities(), and spatialEntitiesSetOperation().

```
7.167.3.14 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 125 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

7.167.3.15 unsigned long TimePoint::getValue() const

Get the value of the timepoint.

Definition at line 26 of file TimePoint.cpp.

References value.

Referenced by printTimePoint(), multiscale::verification::SpatialTemporalTrace::updateLastTimePointValue(), and multiscale::verification::SpatialTemporalTrace::validateTimePointValue().

7.167.3.16 double TimePoint::numberOfSpatialEntities() const

Get the number of considered spatial entities.

Definition at line 45 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::NR_SUBSET_SPECIFIC_TYPES, and spatialEntities.

Referenced by multiscale::verification::NumericEvaluator::evaluate().

```
7.167.3.17 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 171 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), spatialEntities, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtSpatialMeasure(), multiscale::verification::ConstraintEvaluator::filterSpatialEntitiesWrtType(), and multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues().

```
7.167.3.18 void TimePoint::setConsideredSpatialEntityType( const
    SubsetSpecificType & consideredSpatialEntityType )
```

Set the considered spatial entity type to the given type.

Parameters

<i>considered- Spatial- EntityType</i>	The considered type of the spatial entities
--	---

Definition at line 38 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificTypeIndex(), consideredSpatialEntityTypes, and multiscale::verification::subsetspecific::validateSubsetSpecificType().

Referenced by multiscaletest::ModelCheckerTest::InitialiseSpatioTemporalTraceWithAreaValues(), and multiscale::verification::SubsetVisitor::setTimePointConsideredSpatialEntityType().

7.167.3.19 void TimePoint::setValue (unsigned long *value*)

Set the value of the timepoint.

Parameters

<i>value</i>	The value of the timepoint
--------------	----------------------------

Definition at line 30 of file TimePoint.cpp.

References value.

Referenced by multiscale::verification::SpatialTemporalDataReader::setTimePointValue(), and multiscale::verification::SpatialTemporalTrace::updateLastTimePointValue().

7.167.3.20 std::list< std::shared_ptr< SpatialEntity > > TimePoint::spatialEntities- SetOperation (const TimePoint & *timePoint*, const SetOperationType & *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>set- Operation- Type</i>	The considered set operation type
<i>spatialEntity- TypeIndex</i>	The considered spatial entity type index

Definition at line 219 of file TimePoint.cpp.

References multiscale::verification::Difference, getSpatialEntitiesBeginIterator(), getSpatialEntitiesEndIterator(), multiscale::verification::Intersection, and multiscale::verification::Union.

Referenced by updateSpatialEntities().

7.167.3.21 void TimePoint::timePointDifference (const TimePoint & *timePoint*)

Compute the difference of this timepoint and the given timepoint.

Compute the difference of this timepoint and the given timepoint by taking into account the value of consideredSpatialEntityType

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 considered-SpatialEntityType of this timepoint.

Parameters

<i>timePoint</i>	The given timepoint
------------------	---------------------

Definition at line 158 of file TimePoint.cpp.

References multiscale::verification::Difference, and timePointSetOperation().

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::ConstraintVisitor::operator()().

7.167.3.22 void TimePoint::timePointIntersection (const TimePoint & *timePoint*)

Compute the intersection of this timepoint and the given timepoint.

Compute the intersection of this timepoint and the given timepoint by taking into account the value of consideredSpatialEntityType

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' consideredSpatialEntityTypes.

Parameters

<i>timePoint</i>	The given timepoint
------------------	---------------------

Definition at line 162 of file TimePoint.cpp.

References multiscale::verification::Intersection, and timePointSetOperation().

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::ConstraintVisitor::operator()().

7.167.3.23 void TimePoint::timePointSetOperation (const TimePoint & *timePoint*,
const SetOperationType & *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>set- Operation- Type</i>	The considered set operation type

Definition at line 205 of file TimePoint.cpp.

References consideredSpatialEntityTypes, updateConsideredSpatialEntityTypes(), and updateSpatialEntities().

Referenced by timePointDifference(), timePointIntersection(), and timePointUnion().

7.167.3.24 void TimePoint::timePointUnion (const TimePoint & *timePoint*)

Compute the union of this timepoint and the given timepoint.

Compute the union of this timepoint and the given timepoint by taking into account the value of consideredSpatialEntityType.

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 166 of file TimePoint.cpp.

References timePointSetOperation(), and multiscale::verification::Union.

Referenced by multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), and multiscale::verification::ConstraintVisitor::operator()().

7.167.3.25 void TimePoint::updateConsideredSpatialEntityTypes (const std::bitset< NR_SUBSET_SPECIFIC_TYPES > & *consideredSpatialEntityTypes*, const SetOperationType & *setOperationType*) [private]

Update the considered spatial entity type of this timepoint considering the given setOperationType and consideredSpatialEntityTypes.

Definition at line 252 of file TimePoint.cpp.

References consideredSpatialEntityTypes, multiscale::verification::Difference, multiscale::verification::Intersection, and multiscale::verification::Union.

Referenced by timePointSetOperation().

7.167.3.26 void TimePoint::updateSpatialEntities (const TimePoint & *timePoint*, const SetOperationType & *setOperationType*) [private]

Apply the set operation to the collection of spatial entities from this and the given time-point.

Parameters

<i>timePoint</i>	The given timepoint
<i>set-Operation-Type</i>	The considered set operation type

Definition at line 210 of file TimePoint.cpp.

References multiscale::verification::subsetspecific::computeSubsetSpecificType(), multiscale::verification::NR_SUBSET_SPECIFIC_TYPES, and spatialEntitiesSetOperation().

Referenced by timePointSetOperation().

7.167.4 Member Data Documentation

7.167.4.1 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 52 of file TimePoint.hpp.

Referenced by getConsideredSpatialEntities(), getConsideredSpatialEntityTypes(), getSpatialEntitiesBeginIterator(), numberOfSpatialEntities(), setConsideredSpatialEntityType(), timePointSetOperation(), and updateConsideredSpatialEntityTypes().

7.167.4.2 const std::string TimePoint::ERR_GET_NUMERIC_STATE_VARIABLE_PRFIX = "The numeric state variable identified by the given name (" [static, private]

Definition at line 274 of file TimePoint.hpp.

Referenced by getNumericStateVariable().

7.167.4.3 const std::string TimePoint::ERR_GET_NUMERIC_STATE_VARIABLE_SUFFIX = ") does not exist." [static, private]

Definition at line 275 of file TimePoint.hpp.

Referenced by getNumericStateVariable().

7.167.4.4 std::map<std::string, double> multiscale::verification::TimePoint::numericStateVariables [private]

The associative map for storing numeric state variables

Definition at line 50 of file TimePoint.hpp.

Referenced by addNumericStateVariable(), existsNumericStateVariable(), and getNumericStateVariable().

7.167.4.5 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 45 of file TimePoint.hpp.

Referenced by addSpatialEntity(), avgDensity(), avgDistanceBetweenCentroids(), getConsideredSpatialEntities(), getSpatialEntitiesBeginIterator(), getSpatialEntitiesEndIterator(), numberOfSpatialEntities(), and removeSpatialEntity().

7.167.4.6 unsigned long multiscale::verification::TimePoint::value [private]

The value of the timepoint within a simulation/experiment

Definition at line 34 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](#)

7.168 multiscale::verification::TimePointEvaluator Class Reference

Class used to evaluate timepoints.

```
#include <TimePointEvaluator.hpp>
```

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 time-point.
- 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 time-point.

7.168.1 Detailed Description

Class used to evaluate timepoints.

Definition at line 12 of file TimePointEvaluator.hpp.

7.168.2 Member Function Documentation

7.168.2.1 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 time-point.

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::NumericVisitor::operator()()`.

7.168.2.2 static void multiscale::verification::TimePointEvaluator::getSpatialMeasureValues (const std::vector< std::shared_ptr< `SpatialEntity` >> & `consideredSpatialEntities`, std::vector< double > & `spatialMeasureValues`, const SpatialMeasureType & `spatialMeasure`) [inline, static]

Return the spatial measure values for all considered spatial entities in the given time-point.

Parameters

<i>considered-Spatial-Entities</i>	The considered spatial entities
<i>spatial-Measure-Values</i>	The collection of values for the given spatial measure
<i>spatial-Measure</i>	The considered spatial measure

Definition at line 38 of file TimePointEvaluator.hpp.

References multiscale::verification::SpatialEntityEvaluator::evaluate().

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

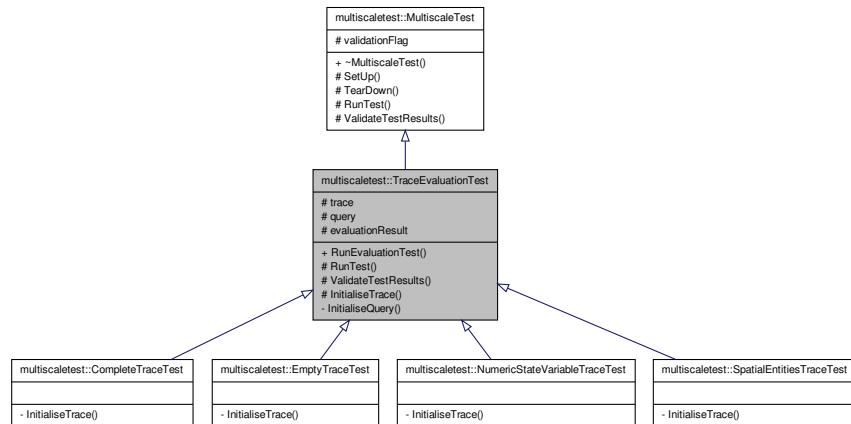
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/[TimePoint-Evaluator.hpp](#)

7.169 multiscaletest::TraceEvaluationTest Class Reference

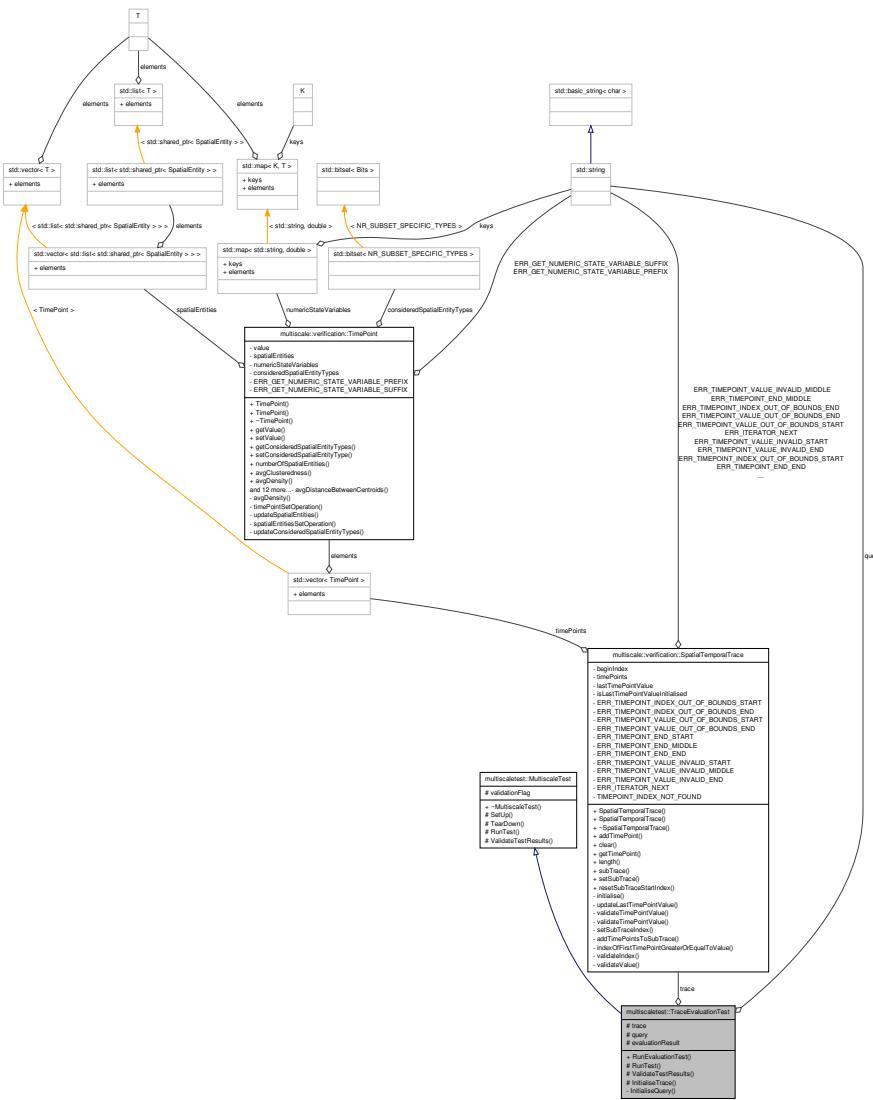
Class for testing evaluation of traces.

```
#include <TraceEvaluationTest.hpp>
```

Inheritance diagram for multiscaletest::TraceEvaluationTest:



Collaboration diagram for multiscaletest::TraceEvaluationTest:



Public Member Functions

- bool `RunEvaluationTest` (const std::string &`query`)
Run the test with the given string.

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

- [mv::SpatialTemporalTrace trace](#)
- std::string [query](#)
- bool [evaluationResult](#)

Private Member Functions

- void [InitialiseQuery \(const std::string &query\)](#)

Initialise the query.

7.169.1 Detailed Description

Class for testing evaluation of traces.

Definition at line 20 of file TraceEvaluationTest.hpp.

7.169.2 Member Function Documentation

7.169.2.1 void multiscaletest::TraceEvaluationTest::InitialiseQuery (const std::string & query) [private]

Initialise the query.

Parameters

query	The given query
-----------------------	-----------------

Definition at line 81 of file TraceEvaluationTest.hpp.

References query.

Referenced by RunEvaluationTest().

7.169.2.2 virtual void multiscaletest::TraceEvaluationTest::InitialiseTrace () [protected, pure virtual]

Initialise the trace.

Implemented in [multiscaletest::CompleteTraceTest](#), [multiscaletest::EmptyTraceTest](#),

[multiscaletest::NumericStateVariableTraceTest](#), and [multiscaletest::SpatialEntities-TraceTest](#).

Referenced by [RunEvaluationTest\(\)](#).

7.169.2.3 bool multiscaletest::TraceEvaluationTest::RunEvaluationTest (const std::string & query)

Run the test with the given string.

Parameters

<code>query</code>	The given query
--------------------	-----------------

Definition at line 58 of file [TraceEvaluationTest.hpp](#).

References [evaluationResult](#), [InitialiseQuery\(\)](#), [InitialiseTrace\(\)](#), [RunTest\(\)](#), and [ValidateTestResults\(\)](#).

7.169.2.4 void multiscaletest::TraceEvaluationTest::RunTest () [override, protected, virtual]

Run the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 68 of file [TraceEvaluationTest.hpp](#).

References [ERR_MSG_TEST](#), [multiscale::verification::AbstractSyntaxTree::evaluate\(\)](#), [evaluationResult](#), [MS_throw](#), [multiscale::verification::Parser::parse\(\)](#), [query](#), and [trace](#).

Referenced by [RunEvaluationTest\(\)](#).

7.169.2.5 void multiscaletest::TraceEvaluationTest::ValidateTestResults () [override, protected, virtual]

Validate the results of the test.

Implements [multiscaletest::MultiscaleTest](#).

Definition at line 79 of file [TraceEvaluationTest.hpp](#).

Referenced by [RunEvaluationTest\(\)](#).

7.169.3 Member Data Documentation

7.169.3.1 bool multiscaletest::TraceEvaluationTest::evaluationResult [protected]

The result of the evaluation

Definition at line 27 of file [TraceEvaluationTest.hpp](#).

Referenced by RunEvaluationTest(), and RunTest().

7.169.3.2 std::string multiscaletest::TraceEvaluationTest::query [protected]

The query to be checked

Definition at line 25 of file TraceEvaluationTest.hpp.

Referenced by InitialiseQuery(), and RunTest().

7.169.3.3 mv::SpatialTemporalTrace multiscaletest::TraceEvaluationTest::trace [protected]

The spatial temporal trace

Definition at line 24 of file TraceEvaluationTest.hpp.

Referenced by RunTest().

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

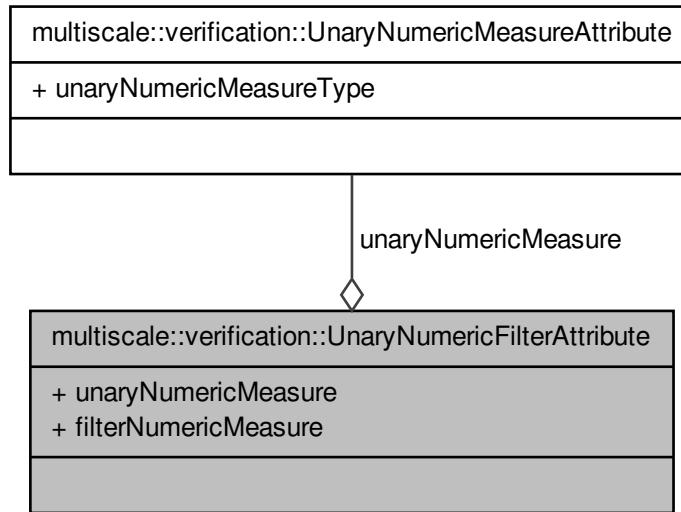
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/[TraceEvaluationTest.hpp](#)

7.170 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`

7.170.1 Detailed Description

Class for representing a unary numeric filter attribute.

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

7.170.2 Member Data Documentation

7.170.2.1 FilterNumericMeasureAttributeType multiscale::verification::UnaryNumericFilterAttribute::filterNumericMeasure

The considered filter numeric measure

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

Referenced by `multiscale::verification::FilterNumericVisitor::operator()()`.

7.170.2.2 **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](#)

7.171 **multiscale::verification::UnaryNumericMeasureAttribute** - Class Reference

Class for representing a unary numeric measure attribute.

```
#include <UnaryNumericMeasureAttribute.hpp>
```

Public Attributes

- [UnaryNumericMeasureType unaryNumericMeasureType](#)

7.171.1 Detailed Description

Class for representing a unary numeric measure attribute.

Definition at line 33 of file UnaryNumericMeasureAttribute.hpp.

7.171.2 Member Data Documentation

7.171.2.1 **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](#)

7.172 multiscale::verification::UnaryNumericMeasureTypeParser Struct Reference

Symbol table and parser for the unary numeric measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [UnaryNumericMeasureTypeParser \(\)](#)

7.172.1 Detailed Description

Symbol table and parser for the unary numeric measure type.

Definition at line 115 of file `SymbolTables.hpp`.

7.172.2 Constructor & Destructor Documentation

7.172.2.1 multiscale::verification::UnaryNumericMeasureTypeParser::UnaryNumericMeasureTypeParser () [inline]

Definition at line 117 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:

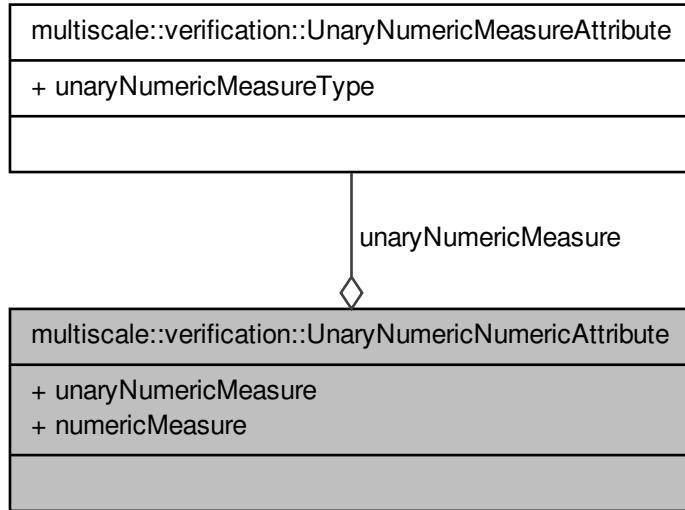
- [/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/`SymbolTables.hpp`](#)

7.173 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](#)
- [NumericMeasureAttributeType numericMeasure](#)

7.173.1 Detailed Description

Class for representing a unary numeric numeric measure attribute.

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

7.173.2 Member Data Documentation

7.173.2.1 NumericMeasureAttributeType multiscale::verification::UnaryNumericNumericAttribute::numericMeasure

The considered numeric measure

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

Referenced by `multiscale::verification::NumericVisitor::operator()()`.

7.174 multiscale::verification::UnarySpatialConstraintAttribute Class Reference

7.173.2.2 UnaryNumericMeasureAttribute multiscale::verification::Unary- NumericNumericAttribute::unaryNumericMeasure

The unary numeric measure

Definition at line 19 of file UnaryNumericNumericAttribute.hpp.

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

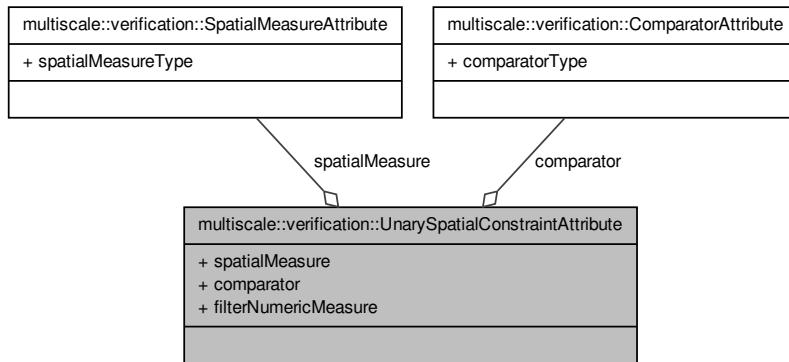
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnaryNumericNumericAttribute.hpp](#)

7.174 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`

7.174.1 Detailed Description

Class for representing a "unary" spatial constraint attribute.

Definition at line 16 of file UnarySpatialConstraintAttribute.hpp.

7.174.2 Member Data Documentation

7.174.2.1 ComparatorAttribute multiscale::verification::UnarySpatialConstraintAttribute::comparator

The comparator

Definition at line 21 of file UnarySpatialConstraintAttribute.hpp.

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

7.174.2.2 FilterNumericMeasureAttributeType multiscale::verification::UnarySpatialConstraintAttribute::filterNumericMeasure

The filter numeric measure

Definition at line 22 of file UnarySpatialConstraintAttribute.hpp.

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

7.174.2.3 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:

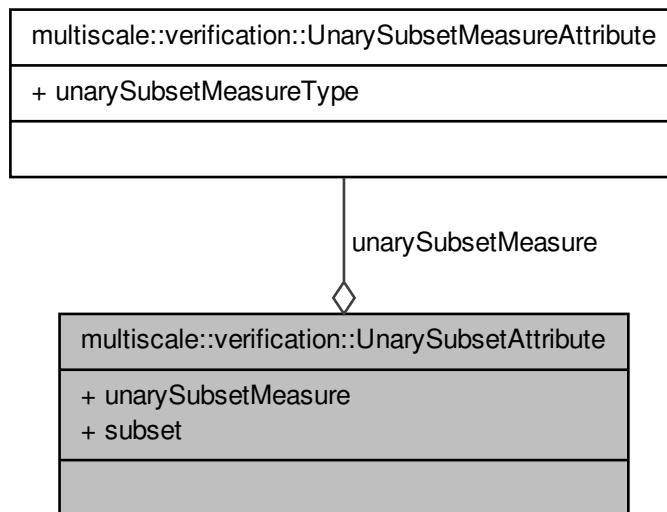
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnarySpatialConstraintAttribute.hpp](#)

7.175 multiscale::verification::UnarySubsetAttribute Class - Reference

Class for representing a unary subset attribute.

```
#include <UnarySubsetAttribute.hpp>
```

Collaboration diagram for multiscale::verification::UnarySubsetAttribute:



Public Attributes

- [UnarySubsetMeasureAttribute unarySubsetMeasure](#)
- [SubsetAttributeType subset](#)

7.175.1 Detailed Description

Class for representing a unary subset attribute.

Definition at line 15 of file UnarySubsetAttribute.hpp.

7.175.2 Member Data Documentation

7.175.2.1 SubsetAttributeType multiscale::verification::UnarySubsetAttribute- ::subset

The considered subset

Definition at line 20 of file UnarySubsetAttribute.hpp.

Referenced by [multiscale::verification::NumericVisitor::operator\(\)\(\)](#).

7.175.2.2 **UnarySubsetMeasureAttribute multiscale::verification::UnarySubsetAttribute::unarySubsetMeasure**

The unary subset measure

Definition at line 19 of file UnarySubsetAttribute.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/[UnarySubsetAttribute.hpp](#)

7.176 **multiscale::verification::UnarySubsetMeasureAttribute** - Class Reference

Class for representing a unary subset measure attribute.

```
#include <UnarySubsetMeasureAttribute.hpp>
```

Public Attributes

- [UnarySubsetMeasureType unarySubsetMeasureType](#)

7.176.1 Detailed Description

Class for representing a unary subset measure attribute.

Definition at line 29 of file UnarySubsetMeasureAttribute.hpp.

7.176.2 Member Data Documentation

7.176.2.1 **UnarySubsetMeasureType multiscale::verification::UnarySubsetMeasureAttribute::unarySubsetMeasureType**

The unary subset measure type

Definition at line 33 of file UnarySubsetMeasureAttribute.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/[UnarySubsetMeasureAttribute.hpp](#)

7.177 multiscale::verification::UnarySubsetMeasureTypeParser Struct Reference

7.177 multiscale::verification::UnarySubsetMeasureTypeParser - Struct Reference

Symbol table and parser for the unary subset measure type.

```
#include <SymbolTables.hpp>
```

Public Member Functions

- [UnarySubsetMeasureTypeParser \(\)](#)

7.177.1 Detailed Description

Symbol table and parser for the unary subset measure type.

Definition at line 178 of file [SymbolTables.hpp](#).

7.177.2 Constructor & Destructor Documentation

7.177.2.1 multiscale::verification::UnarySubsetMeasureTypeParser::UnarySubsetMeasureTypeParser () [inline]

Definition at line 180 of file [SymbolTables.hpp](#).

References [multiscale::verification::Clusteredness](#), [multiscale::verification::Count](#), and [multiscale::verification::Density](#).

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

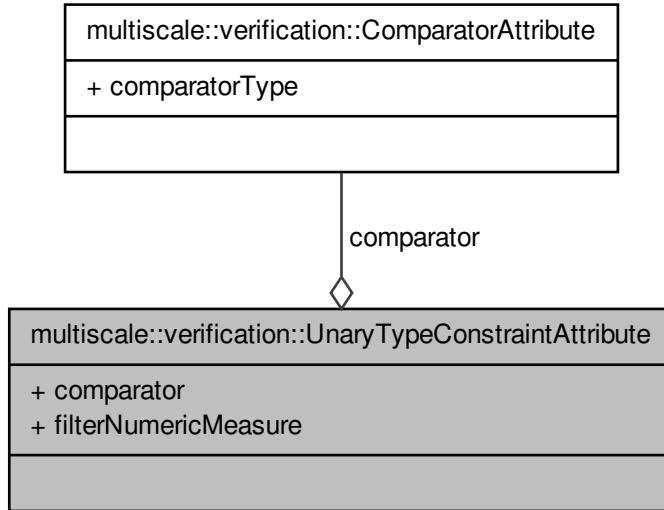
- [/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/\[SymbolTables.hpp\]\(#\)](#)

7.178 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](#)
- [FilterNumericMeasureAttributeType filterNumericMeasure](#)

7.178.1 Detailed Description

Class for representing a "unary" type constraint attribute.

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

7.178.2 Member Data Documentation

7.178.2.1 ComparatorAttribute multiscale::verification::UnaryTypeConstraintAttribute::comparator

The comparator

Definition at line 19 of file `UnaryTypeConstraintAttribute.hpp`.

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

7.178.2.2 FilterNumericMeasureAttributeType multiscale::verification::UnaryTypeConstraintAttribute::filterNumericMeasure

The filter numeric measure

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:

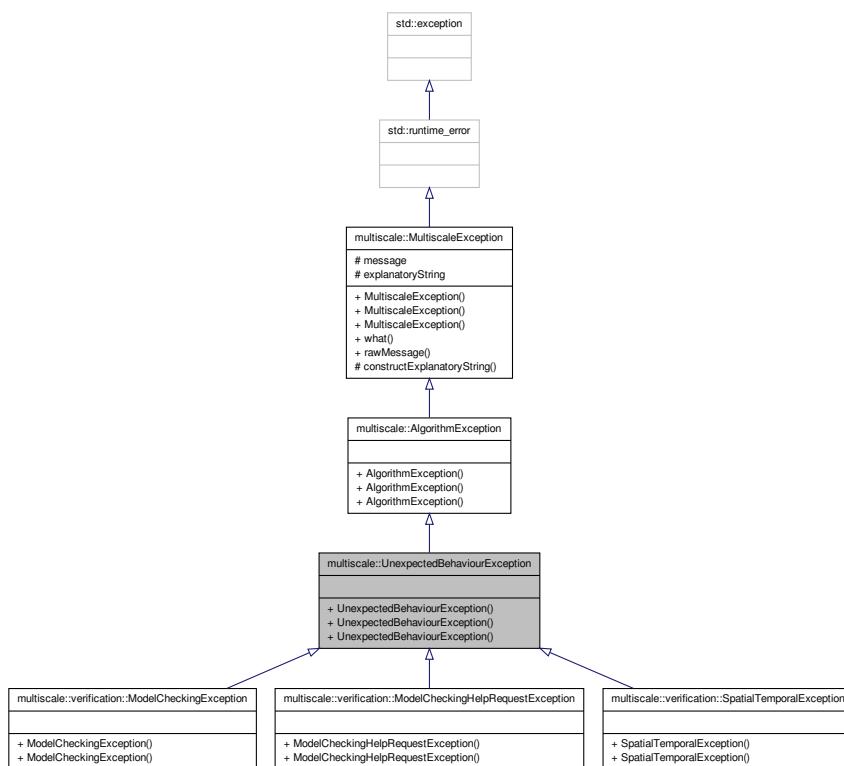
- /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UnaryTypeConstraintAttribute.hpp](#)

7.179 multiscale::UnexpectedBehaviourException Class Reference

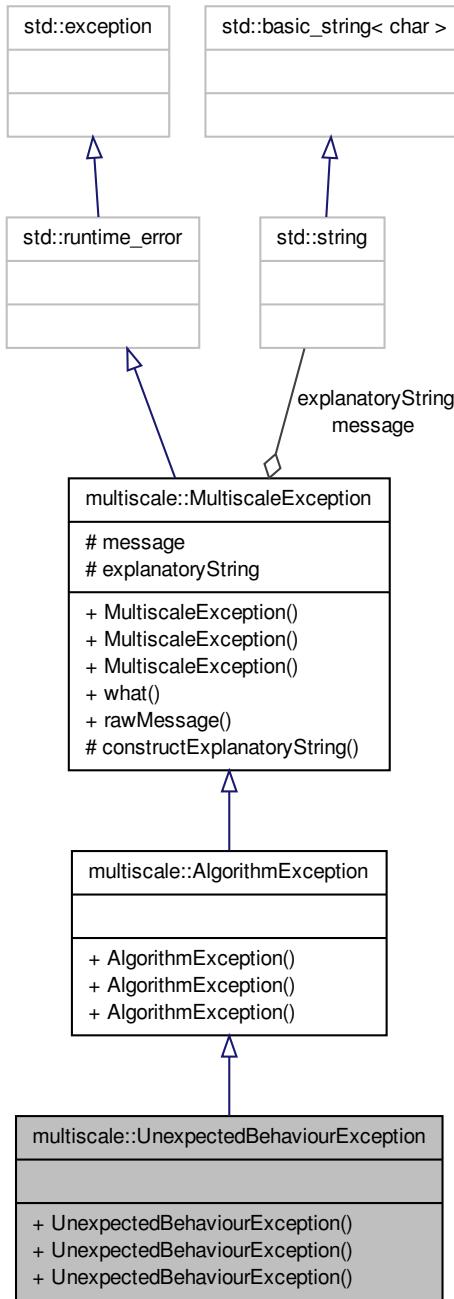
Class for representing unexpected behaviour exceptions.

```
#include <UnexpectedBehaviourException.hpp>
```

Inheritance diagram for multiscale::UnexpectedBehaviourException:



Collaboration diagram for multiscale::UnexpectedBehaviourException:



7.180 multiscale::verification::UnexpectedErrorHandler Struct Reference

Public Member Functions

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

7.179.1 Detailed Description

Class for representing unexpected behaviour exceptions.

Definition at line 14 of file UnexpectedBehaviourException.hpp.

7.179.2 Constructor & Destructor Documentation

7.179.2.1 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException () [inline]

Definition at line 18 of file UnexpectedBehaviourException.hpp.

7.179.2.2 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException (const string & file, int line, const string & msg) [inline, explicit]

Definition at line 20 of file UnexpectedBehaviourException.hpp.

7.179.2.3 multiscale::UnexpectedBehaviourException::UnexpectedBehaviourException (const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 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](#)

7.180 multiscale::verification::UnexpectedErrorHandler - Struct Reference

Structure for defining the error handler for unexpected token errors.

```
#include <UnexpectedErrorHandler.hpp>
```

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.

7.180.1 Detailed Description

Structure for defining the error handler for unexpected token errors.

Definition at line 17 of file UnexpectedTokenErrorHandler.hpp.

7.180.2 Member Function Documentation

- 7.180.2.1 **std::string multiscale::verification::UnexpectedTokenErrorHandler::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

expected- Token	The expected token (not a std::string)
----------------------------	--

Definition at line 46 of file UnexpectedTokenErrorHandler.hpp.

Referenced by **operator()()**.

- 7.180.2.2 **template<typename Iterator> void multiscale::verification::UnexpectedTokenErrorHandler::operator() (qi::info const & expectedToken, Iterator errorPosition, Iterator last) const [inline]**

Overloaded operator.

Parameters

<i>expected- Token</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:

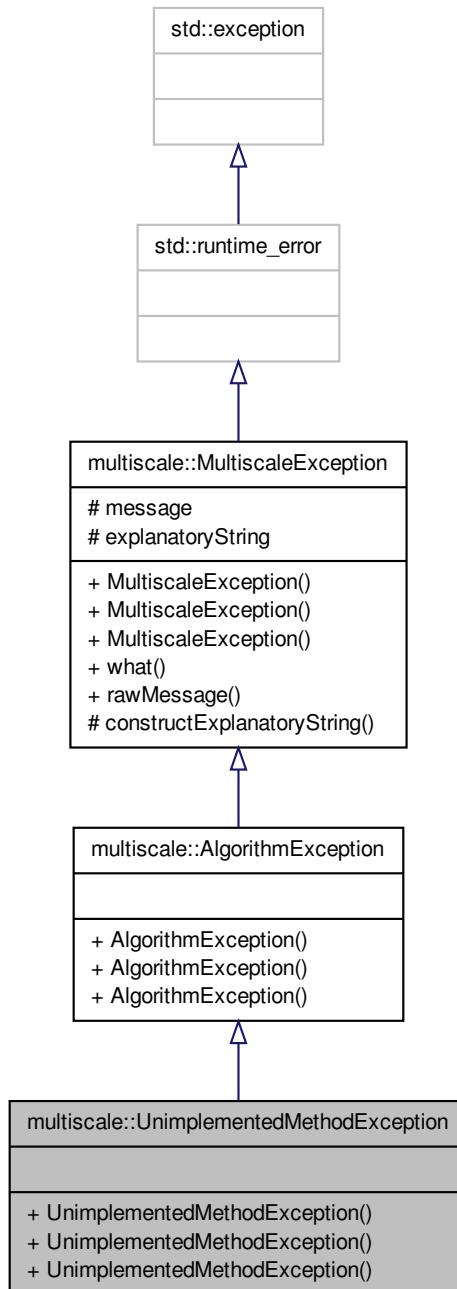
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/[UnexpectedErrorHandler.hpp](#)

7.181 multiscale::UnimplementedMethodException Class Reference

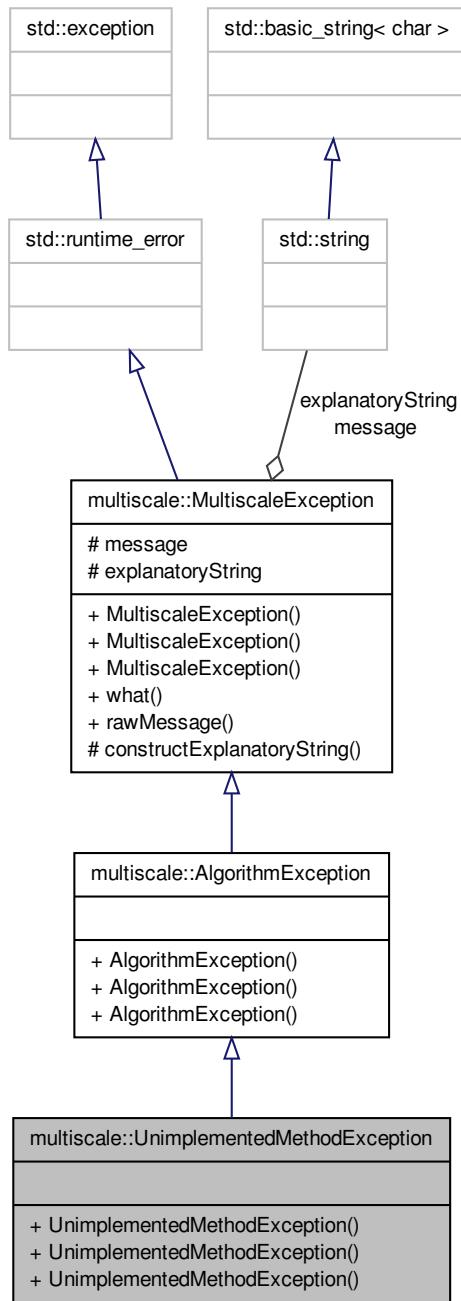
Class for representing unimplemented method exceptions.

```
#include <UnimplementedMethodException.hpp>
```

Inheritance diagram for multiscale::UnimplementedMethodException:



Collaboration diagram for multiscale::UnimplementedMethodException:



Public Member Functions

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

7.181.1 Detailed Description

Class for representing unimplemented method exceptions.

Definition at line 14 of file UnimplementedMethodException.hpp.

7.181.2 Constructor & Destructor Documentation

7.181.2.1 multiscale::UnimplementedMethodException::UnimplementedMethodException() [inline]

Definition at line 18 of file UnimplementedMethodException.hpp.

7.181.2.2 multiscale::UnimplementedMethodException::UnimplementedMethodException(const string & file, int line, const string & msg) [inline, explicit]

Definition at line 20 of file UnimplementedMethodException.hpp.

7.181.2.3 multiscale::UnimplementedMethodException::UnimplementedMethodException(const string & file, int line, const char * msg) [inline, explicit]

Definition at line 24 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](#)

7.182 multiscale::verification::UntilLogicPropertyAttribute Class - Reference

Class for representing an "until" logic property attribute.

```
#include <UntilLogicPropertyAttribute.hpp>
```

Public Attributes

- unsigned long [startTimepoint](#)
- unsigned long [endTimepoint](#)
- [LogicPropertyAttributeType logicProperty](#)

7.182.1 Detailed Description

Class for representing an "until" logic property attribute.

Definition at line 14 of file UntilLogicPropertyAttribute.hpp.

7.182.2 Member Data Documentation

7.182.2.1 unsigned long multiscale::verification::UntilLogicPropertyAttribute::endTimepoint

The considered end timepoint

Definition at line 19 of file UntilLogicPropertyAttribute.hpp.

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

7.182.2.2 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().

7.182.2.3 unsigned long multiscale::verification::UntilLogicPropertyAttribute::startTimepoint

The considered start timepoint

Definition at line 18 of file UntilLogicPropertyAttribute.hpp.

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

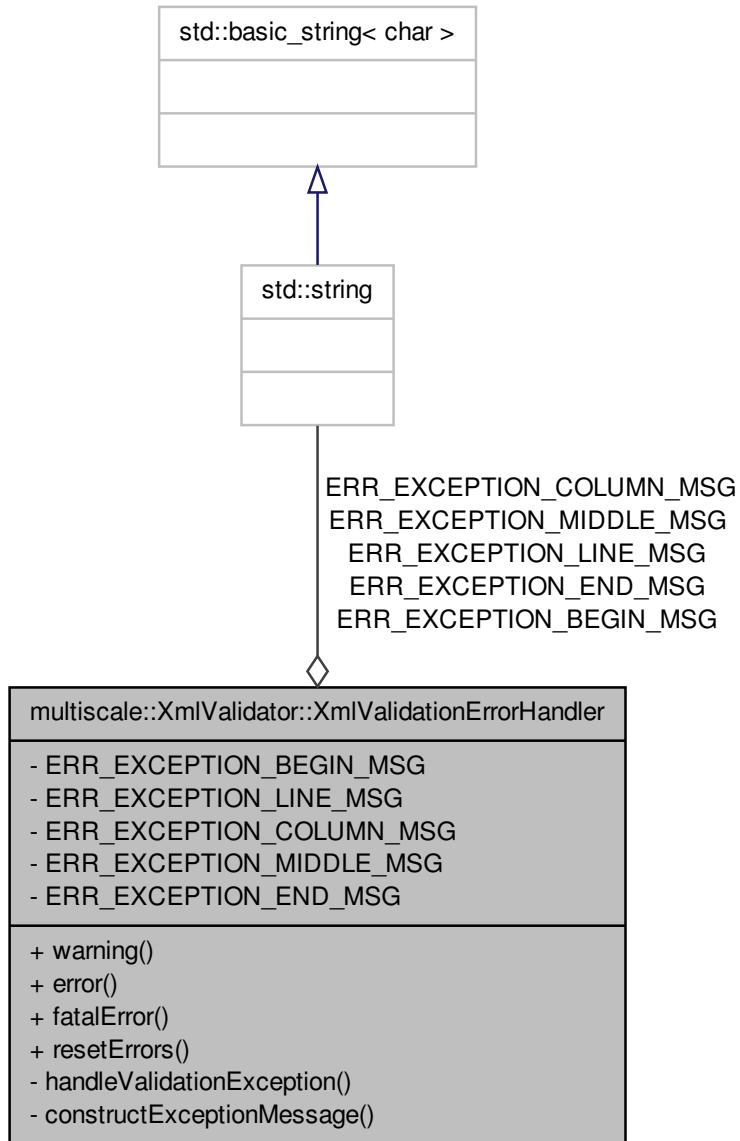
The documentation for this class was generated from the following file:

- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/[UntilLogicPropertyAttribute.hpp](#)

7.183 multiscale::XmlValidator::XmlValidationErrorHandler Class - Reference

Class used for handling errors during the xml file validation process.

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` = "The provided xml file is invalid. 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` = "\."

7.183.1 Detailed Description

Class used for handling errors during the xml file validation process.

Definition at line 91 of file XmlValidator.hpp.

7.183.2 Member Function Documentation

7.183.2.1 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 97 of file XmlValidator.cpp.

7.183 multiscale::XmlValidator::XmlValidationErrorHandler Class Reference 985

References multiscale::StringManipulator::toString().

7.183.2.2 void XmlValidator::XmlValidationErrorHandler::error (const SAXParseException & ex) [override]

Handle recoverable error messages.

Parameters

ex	The exception thrown during the validation process
-----------	--

Definition at line 81 of file XmlValidator.cpp.

7.183.2.3 void XmlValidator::XmlValidationErrorHandler::fatalError (const SAXParseException & ex) [override]

Handle non-recoverable error messages.

Parameters

ex	The exception thrown during the validation process
-----------	--

Definition at line 85 of file XmlValidator.cpp.

7.183.2.4 void XmlValidator::XmlValidationErrorHandler::handleValidationException (const SAXParseException & ex) [private]

Handle the exception thrown during the validation process.

Parameters

ex	The exception thrown during the validation process
-----------	--

Definition at line 91 of file XmlValidator.cpp.

References MS_throw.

Referenced by warning().

7.183.2.5 void XmlValidator::XmlValidationErrorHandler::resetErrors () [override]

Reinitialise the error handler.

Definition at line 89 of file XmlValidator.cpp.

```
7.183.2.6 void XmlValidator::XmlValidationErrorHandler::warning ( const  
SAXParseException & ex ) [override]
```

Handle warning messages.

Parameters

ex	The exception thrown during the validation process
----	--

Definition at line 77 of file XmlValidator.cpp.

References handleValidationException().

7.183.3 Member Data Documentation

```
7.183.3.1 const std::string XmlValidator::XmlValidationErrorHandler::ERR_EXCEPT-  
ION_BEGIN_MSG = "The provided xml file is invalid. An error occurred at "  
[static, private]
```

Definition at line 131 of file XmlValidator.hpp.

```
7.183.3.2 const std::string XmlValidator::XmlValidationErrorHandler::-  
ERR_EXCEPTION_COLUMN_MSG = ", column " [static,  
private]
```

Definition at line 134 of file XmlValidator.hpp.

```
7.183.3.3 const std::string XmlValidator::XmlValidationErrorHandler-  
::ERR_EXCEPTION_END_MSG = "\n" [static,  
private]
```

Definition at line 137 of file XmlValidator.hpp.

```
7.183.3.4 const std::string XmlValidator::XmlValidationErrorHandler-  
::ERR_EXCEPTION_LINE_MSG = "line " [static,  
private]
```

Definition at line 133 of file XmlValidator.hpp.

```
7.183.3.5 const std::string XmlValidator::XmlValidationErrorHandler::ERR_EXCE-  
PTION_MIDDLE_MSG = " and the error message is \\" [static,  
private]
```

Definition at line 135 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](#)

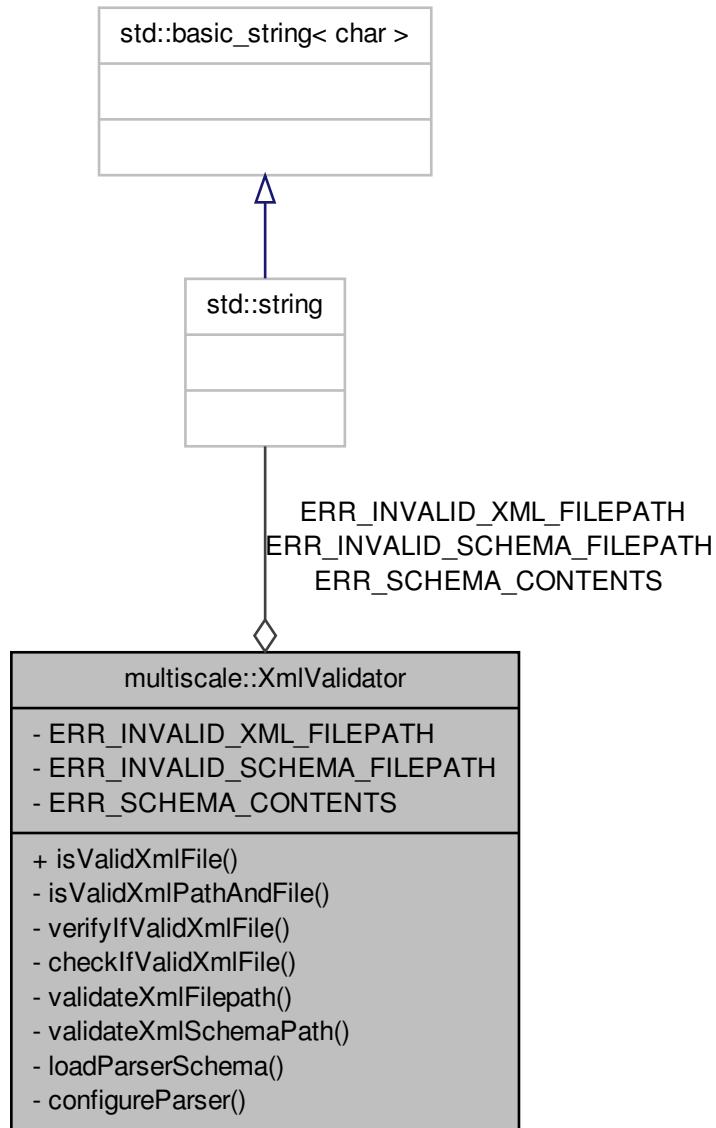
- /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/[XmlValidator.-cpp](#)

7.184 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 Private Member Functions

- static bool [isValidXmlPathAndFile](#) (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 [verifyIfValidXmlFile](#) (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 [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."

7.184.1 Detailed Description

Class used to validate xml files.

Definition at line 18 of file XmlValidator.hpp.

7.184.2 Member Function Documentation

7.184.2.1 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

<i>xmlFilepath</i>	The path to the xml file
<i>xmlSchema-Path</i>	The path to the xml schema file

Definition at line 37 of file XmlValidator.cpp.

References configureParser(), and loadParserSchema().

Referenced by verifyIfValidXmlFile().

7.184.2.2 void XmlValidator::configureParser (XercesDOMParser & *parser*) [static, private]

Configure the given parser.

Parameters

<i>parser</i>	The given xml DOM parser
---------------	--------------------------

Definition at line 66 of file XmlValidator.cpp.

Referenced by checkIfValidXmlFile().

7.184.2.3 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

<i>xmlFilepath</i>	The path to the xml file
<i>xmlSchema-Path</i>	The path to the xml schema file

Definition at line 12 of file XmlValidator.cpp.

References isValidXmlPathAndFile().

Referenced by multiscale::verification::SpatialTemporalDataReader::isValidInputFile().

7.184.2.4 bool XmlValidator::isValidXmlPathAndFile (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

<i>xmlfilepath</i>	The path to the xml file
<i>xmlSchema-Path</i>	The path to the xml schema file

Definition at line 22 of file XmlValidator.cpp.

References validateXmlFilepath(), validateXmlSchemaPath(), and verifyIfValidXmlFile().

Referenced by isValidXmlFile().

7.184.2.5 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>xmlSchema-Path</i>	The file path to the xml schema

Definition at line 60 of file XmlValidator.cpp.

References ERR_SCHEMA_CONTENTS, and MS_throw.

Referenced by checkIfValidXmlFile().

7.184.2.6 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 48 of file XmlValidator.cpp.

References ERR_INVALID_XML_FILEPATH, multiscale::Filesystem::isValidFilePath(), and MS_throw.

Referenced by isValidXmlPathAndFile().

7.184.2.7 void XmlValidator::validateXmlSchemaPath (const std::string & *xmlSchemaPath*) [static, private]

Check if the provided xml schema file path is valid.

Parameters

<i>xmlSchema- Path</i>	The path to the xml schema
----------------------------	----------------------------

Definition at line 54 of file XmlValidator.cpp.

References `ERR_INVALID_SCHEMA_FILEPATH`, `multiscale::Filesystem::isValidFilePath()`, and `MS_throw`.

Referenced by `isValidXmlPathAndFile()`.

7.184.2.8 bool XmlValidator::verifyIfValidXmlFile (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

<i>xmlFilepath</i>	The path to the xml file
<i>xmlSchema- Path</i>	The path to the xml schema file

Definition at line 29 of file XmlValidator.cpp.

References `checkIfValidXmlFile()`.

Referenced by `isValidXmlPathAndFile()`.

7.184.3 Member Data Documentation

**7.184.3.1 const std::string XmlValidator::ERR_INVALID_SCHEMA_FILEPATH
= "The provided xml schema file path is invalid. Please change." [static, private]**

Definition at line 84 of file XmlValidator.hpp.

Referenced by `validateXmlSchemaPath()`.

**7.184.3.2 const std::string XmlValidator::ERR_INVALID_XML_FILEPATH = "The
provided xml file path is invalid. Please change." [static, private]**

Definition at line 83 of file XmlValidator.hpp.

Referenced by validateXmlFilepath().

```
7.184.3.3 const std::string XmlValidator::ERR_SCHEMA_CONTENTS = "The provided  
xml schema is invalid. Please verify the xml schema contents." [static,  
private]
```

Definition at line 86 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

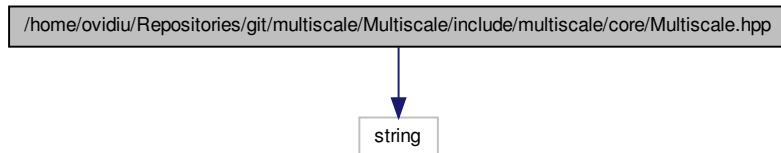
Chapter 8

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

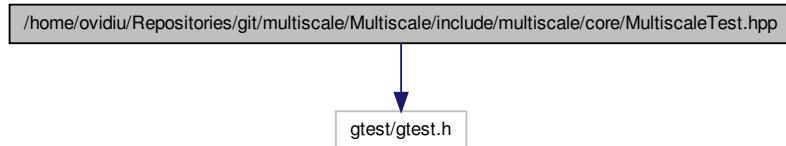
- namespace [multiscale](#)

Variables

- const int [multiscale::EXEC_SUCCESS_CODE](#) = 0
- const int [multiscale::EXEC_ERR_CODE](#) = 1
- const std::string [multiscale::ERR_MSG](#) = "An error occurred: "

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

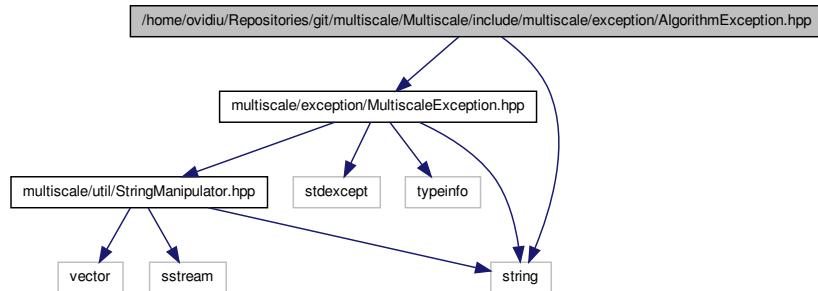
- class [multiscaletest::MultiscaleTest](#)

Namespaces

- namespace [multiscaletest](#)

8.4 /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:



- class [multiscale::AlgorithmException](#)

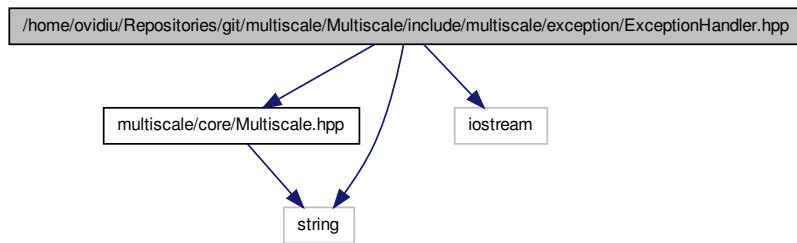
Class for representing algorithm exceptions.

Namespaces

- namespace [multiscale](#)

8.5 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/- ExceptionHandler.hpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include <iostream> x
#include <string> Include dependency graph for ExceptionHandler.hpp:
```



Classes

- class [multiscale::ExceptionHandler](#)
Exception handler class.

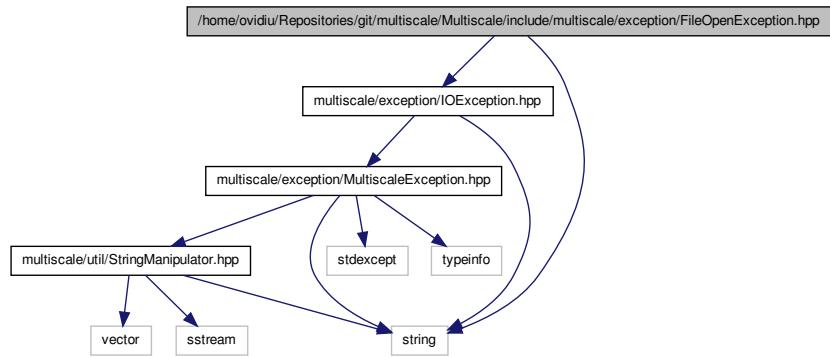
Namespaces

- namespace [multiscale](#)

8.6 /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](#)

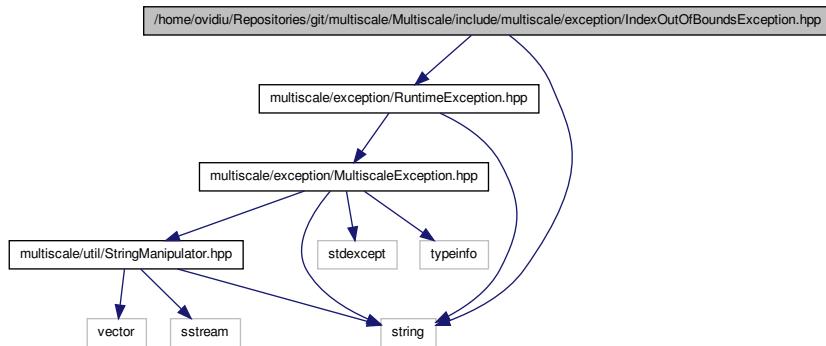
Class for representing exceptions when opening a file.

Namespaces

- namespace [multiscale](#)

8.7 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception-/IndexOutOfBoundsException.hpp File Reference

```
#include "multiscale/exception/RuntimeException.hpp" x
#include <string> Include dependency graph for IndexOutOfBoundsException-
```



Classes

- class [multiscale::IndexOutOfBoundsException](#)

Class for representing an index out of bounds exception.

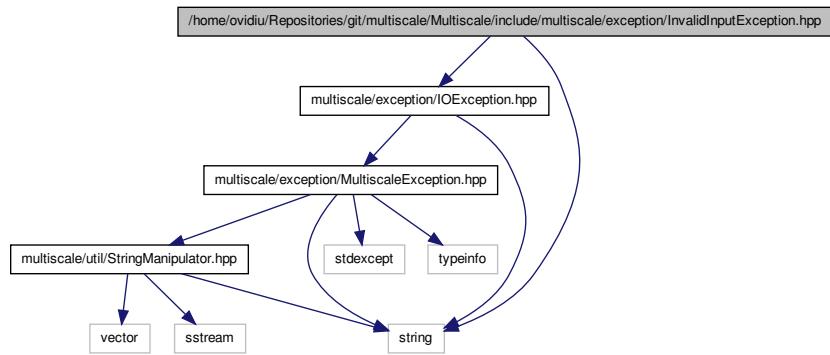
Namespaces

- namespace [multiscale](#)

8.8 /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

- namespace [multiscale](#)

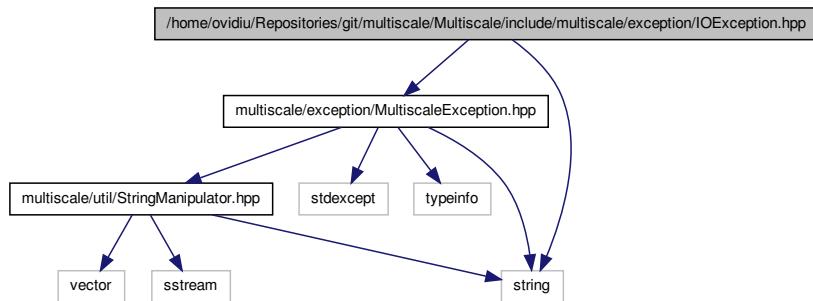
8.9 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/IOException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp" x
```

**8.10 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/include/multiscale/exception/MultiscaleException.hpp File
Reference**

1001

#include <string> Include dependency graph for IOException.hpp:



Classes

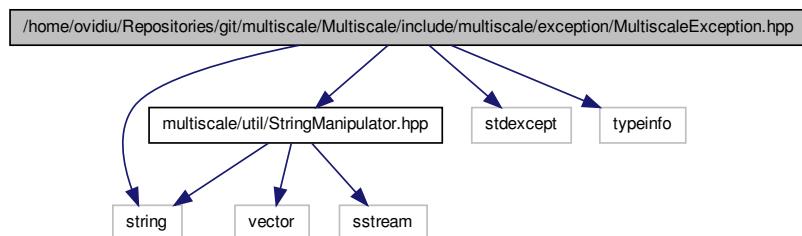
- class [multiscale::IOException](#)
Class for representing input and output exceptions.

Namespaces

- namespace [multiscale](#)

**8.10 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-
MultiscaleException.hpp File Reference**

#include "multiscale/util/StringManipulator.hpp" #include
<stdexcept> #include <string> #include <typeinfo> Include
dependency graph for MultiscaleException.hpp:



Classes

- class [multiscale::MultiscaleException](#)

Parent exception class for the project.

Namespaces

- namespace [multiscale](#)

Defines

- #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))

Variables

- const std::string [multiscale::ERR_UNDEFINED_ENUM_VALUE](#) = "The provided enumeration value is invalid. Please use one of the available enumeration values instead."

8.10.1 Define Documentation

8.10.1.1 #define MS_throw(ex, msg) (throw ex(__FILE__, __LINE__, msg))

Definition at line 12 of file MultiscaleException.hpp.

Referenced by multiscale::Numeric::applyOperation(), multiscale::verification::CommandLineModelChecking::areInvalidModelCheckingArguments(), multiscale::verification::CommandLineModelChecking::areModelCheckingTypeSpecificArgumentsPresent(), multiscale::Numeric::combinations(), multiscale::video::RectangularEntity-CsvToInputFilesConverter::computeCoordinate(), multiscale::video::RectangularEntity-CsvToInputFilesConverter::computeSimulationTime(), multiscale::video::Rectangular-CsvToInputFilesConverter::computeSimulationTime(), multiscale::video::PolarCsv-ToInputFilesConverter::computeSimulationTime(), multiscale::analysis::MatFactory::create(), multiscale::verification::SpatialTemporalDataReader::createDerivedSpatial-Entity(), multiscale::analysis::Detector::detect(), multiscale::verification::Approximate-BayesianModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::BayesianModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::StatisticalModelChecker::doesPropertyHoldConsideringResult(), multiscale::verification::SpatialEntityEvaluator::evaluate(), multiscale::verification::SpatialMeasureEvaluator::evaluate(), multiscale::verification::NumericEvaluator::evaluate(), multiscale::verification::ComparatorEvaluator::evaluate(), multiscale::verification::ProbabilisticLogicProperty-Attribute::evaluate(), multiscale::verification::AbstractSyntaxTree::evaluate(), multiscale::verification::SubsetVisitor::evaluateSubsetOperation(), multiscale::OperatingSystem::executeProgramAndVerifyPath(), multiscale::MinEnclosingTriangleFinder::find(),

**8.10 /home/ovidiu/Repositories/git/multiscale/-
Multiscale/include/multiscale/exception/MultiscaleException.hpp File**

Reference	1003
------------------	-------------

```
multiscale::MinEnclosingTriangleFinder::findVertexCOnSideB(), multiscale::verification-  
::SpatialTemporalDataReader::getFirstValidUnprocessedInputFilepath(), multiscale-  
::verification::SpatialTemporalDataReader::getNextSpatialTemporalTrace(), multiscale-  
::verification::SpatialTemporalDataReader::getRandomValidUnprocessedInputFilepath(),  
multiscale::verification::ParserGrammarExceptionHandler::handleExtraInputException(),  
multiscale::verification::CommandLineModelChecking::handleHelpRequest(), multiscale-  
::verification::ParserGrammarExceptionHandler::handleProbabilityException(), multiscale-  
::verification::ParserGrammarExceptionHandler::handleUnexpectedTokenException(),  
multiscale::verification::ParserGrammarExceptionHandler::handleUnparseableInput-  
Exception(), multiscale::XmlValidator::XmlValidationErrorHandler::handleValidation-  
Exception(), multiscale::verification::CommandLineModelChecking::initialise(), multiscale-  
::verification::CommandLineModelChecking::initialiseModelChecker(), multiscale-  
::video::RectangularEntityCsvToInputFilesConverter::initInputFile(), multiscale::analysis-  
::MatFactory::initInputFile(), multiscale::video::PolarCsvToInputFilesConverter::init-  
InputFile(), multiscale::video::RectangularCsvToInputFilesConverter::initInputFile(),  
multiscale::video::PolarCsvToInputFilesConverter::initMaximumConcentration(), multiscale-  
::video::RectangularCsvToInputFilesConverter::initMaximumConcentration(), multiscale-  
::analysis::CircularMatFactory::isValidViewerImage(), multiscale::analysis::Rectangular-  
MatFactory::isValidViewerImage(), multiscale::XmlValidator::loadParserSchema(),  
multiscale::Filesystem::nativeFormatFilePath(), multiscale::analysis::SimulationCluster-  
Detector::outputClusterShape(), multiscale::analysis::Detector::outputResultsToCsv-  
File(), multiscale::analysis::CircularMatFactory::processConcentrations(), multiscale-  
::analysis::RectangularMatFactory::processConcentrations(), multiscale::video::-  
CartesianToConcentrationsConverter::readConcentrations(), multiscale::video::-  
CartesianToPolarConverter::readConcentrations(), multiscale::video::Cartesian-  
ToConcentrationsConverter::readHeaderLine(), multiscale::video::CartesianTo-  
PolarConverter::readHeaderLine(), multiscale::video::CartesianToConcentrations-  
Converter::readInputData(), multiscale::video::CartesianToPolarConverter::read-  
InputData(), multiscale::verification::LogicPropertyDataReader::readLogicProperties-  
FromFile(), multiscale::verification::LogicPropertyDataReader::readLogicProperties-  
FromValidFilepath(), multiscale::verification::CommandLineModelChecking::remove-  
ModelCheckingTypeSpecificArguments(), multiscaletest::TraceEvaluationTest::Run-  
Test(), multiscale::analysis::SpatialEntityPseudo3D::typeAsString(), multiscale::Min-  
EnclosingTriangleFinder::updateSideB(), multiscale::verification::BayesianModel-  
Checker::validateBayesFactorThreshold(), multiscale::video::RectangularEntityCsv-  
ToInputFilesConverter::validateCoordinate(), multiscale::video::RectangularEntity-  
CsvToInputFilesConverter::validateEntitiesGrid(), multiscale::verification::Spatial-  
TemporalDataReader::validateFolderPath(), multiscale::verification::Approximate-  
ProbabilisticModelChecker::validateInput(), multiscale::video::RectangularEntity-  
CsvToInputFilesConverter::validateInput(), multiscale::video::PolarCsvToInputFiles-  
Converter::validateInput(), multiscale::video::RectangularCsvToInputFilesConverter-  
::validateInput(), multiscale::video::RectangularEntityCsvToInputFilesConverter-  
::validateInputLine(), multiscale::video::PolarCsvToInputFilesConverter::validate-  
InputLine(), multiscale::video::RectangularCsvToInputFilesConverter::validateInput-  
Line(), multiscale::analysis::Region::validateInputValues(), multiscale::analysis::Entity-  
::validateInputValues(), multiscale::video::RectangularEntityCsvToInputFilesConverter-  
::validateMaxNrOfEntitiesPerPosition(), multiscale::BinomialDistribution::validate-  
NrOfSuccesses(), multiscale::analysis::Cluster::validateOriginDependentValues(),  
multiscale::Distribution::validateProbability(), multiscale::video::PolarCsvToInputFiles-  
Converter::validateSelectedConcentrationIndex(), multiscale::video::Rectangular-
```

CsvToInputFilesConverter::validateSelectedConcentrationIndex(), multiscale::BetaDistribution::validateShapeParameters(), multiscale::verification::BayesianModelChecker::validateShapeParameters(), multiscale::verification::ApproximateBayesianModelChecker::validateShapeParameters(), multiscale::video::RectangularEntityCsvToInputFilesConverter::validateSimulationTime(), multiscale::verification::subsetsspecific::validateSubsetSpecificType(), multiscale::verification::subsetsspecific::validateSubsetSpecificTypeIndex(), multiscale::verification::StatisticalModelChecker::validateTypesErrors(), multiscale::verification::ApproximateBayesianModelChecker::validateVarianceThreshold(), multiscale::XmlValidator::validateXmlFilepath(), and multiscale::XmlValidator::validateXmlSchemaPath().

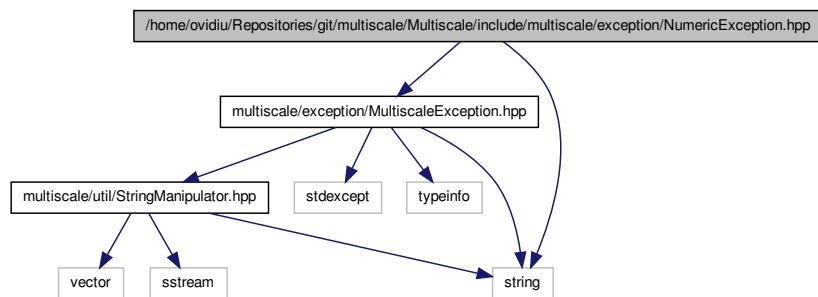
8.10.1.2 #define MS_throw_detailed(ex, startMsg, msg, endMsg) (throw ex(_FILE_, _LINE_, startMsg + msg + endMsg))

Definition at line 13 of file MultiscaleException.hpp.

Referenced by multiscale::verification::TimePoint::getNumericStateVariable(), multiscale::verification::SpatialEntity::validateAngleValue(), multiscale::verification::SpatialTemporalTrace::validateIndex(), multiscale::Numeric::validateLogBase(), multiscale::Numeric::validateLogNumber(), multiscale::Numeric::validatePercentile(), multiscale::Numeric::validateQuartile(), multiscale::verification::SpatialEntity::validateRealNonNegativeValue(), multiscale::verification::SpatialEntity::validateRealValueBtwZeroAndOne(), multiscale::verification::SpatialTemporalTrace::validateTimePointValue(), and multiscale::verification::SpatialTemporalTrace::validateValue().

8.11 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/- NumericException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp" x
#include <string> Include dependency graph for NumericException.hpp:
```



- class [multiscale::NumericException](#)

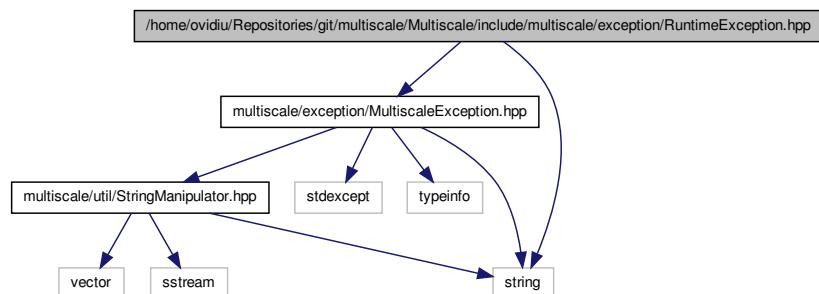
Class for representing algorithm exceptions.

Namespaces

- namespace [multiscale](#)

8.12 /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](#)

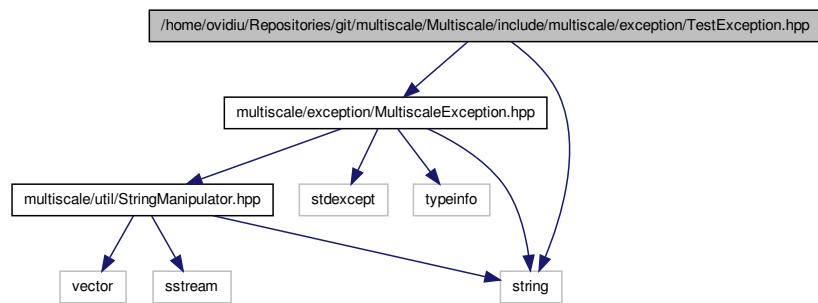
Class for representing runtime exceptions.

Namespaces

- namespace [multiscale](#)

8.13 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/TestException.hpp File Reference

```
#include "multiscale/exception/MultiscaleException.hpp" x
#include <string> Include dependency graph for TestException.hpp:
```



Classes

- class [multiscale::TestException](#)

Class for representing testing exceptions.

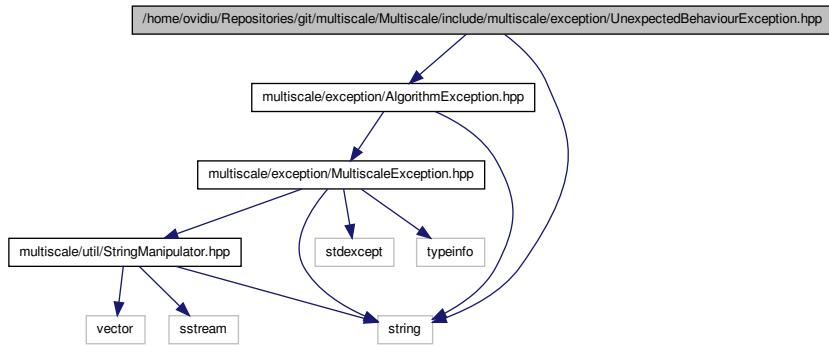
Namespaces

- namespace [multiscale](#)

8.14 /home/ovidiu/Repositories/git/multiscale/Multiscale/include/multiscale/exception/UnexpectedBehaviourException.hpp File Reference

```
#include "multiscale/exception/AlgorithmException.hpp" x
#include <string> Include dependency graph for UnexpectedBehaviour-
```

**8.15 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/include/multiscale/exception/UnimplementedMethodException.hpp**
File Reference **1007**
Exception.hpp:



Classes

- class [multiscale::UnexpectedBehaviourException](#)

Class for representing unexpected behaviour exceptions.

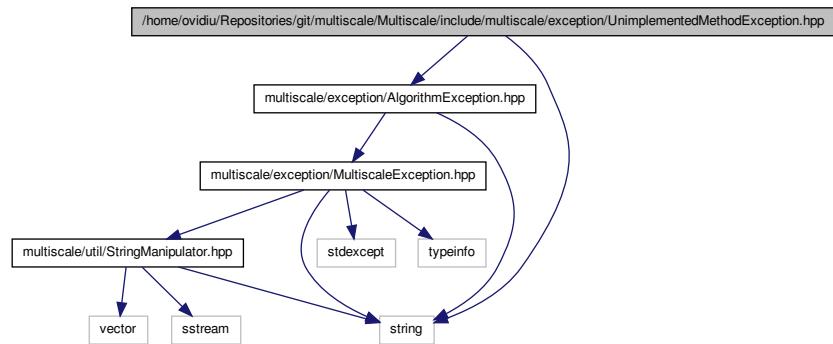
Namespaces

- namespace [multiscale](#)

**8.15 /home/ovidiu.Repositories/git/multiscale/Multiscale/include/multiscale/exception/-
UnimplementedMethodException.hpp File Reference**

```
#include "multiscale/exception/AlgorithmException.hpp" x
#include <string> Include dependency graph for UnimplementedMethod-
```

Exception.hpp:



Classes

- class [multiscale::UnimplementedMethodException](#)

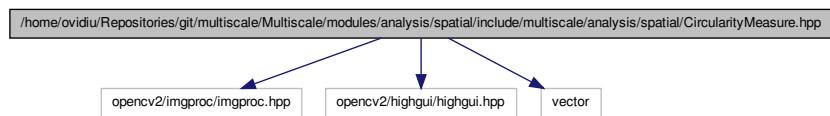
Class for representing unimplemented method exceptions.

Namespaces

- namespace [multiscale](#)

8.16 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/CircularityMeasure.hpp File Reference

```
#include "opencv2/imgproc/imgproc.hpp" #include "opencv2/highgui/highgui.hpp" #include <vector> Include dependency graph for CircularityMeasure.hpp:
```



- class [multiscale::analysis::CircularityMeasure](#)

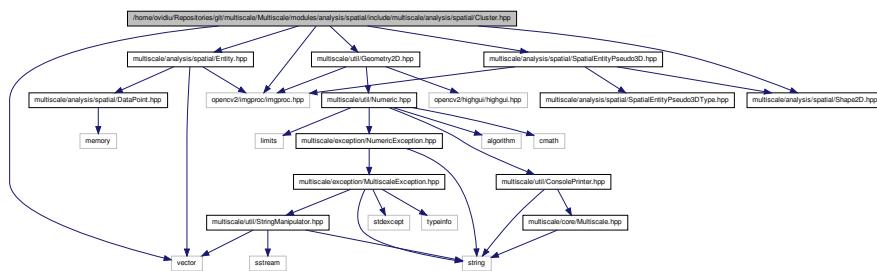
Class for computing the circularity measure for the given collection of points.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.17 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/Cluster.hpp File Reference

```
#include "opencv2/imgproc/imgproc.hpp" #include "multiscale/analysis/spatial/-  
Entity.hpp" #include "multiscale/analysis/spatial/Shape2-  
D.hpp" #include "multiscale/analysis/spatial/Spatial-  
EntityPseudo3D.hpp" #include "multiscale/util/Geometry2-  
D.hpp" #include <vector> Include dependency graph for Cluster.hpp:
```



Classes

- class [multiscale::analysis::Cluster](#)

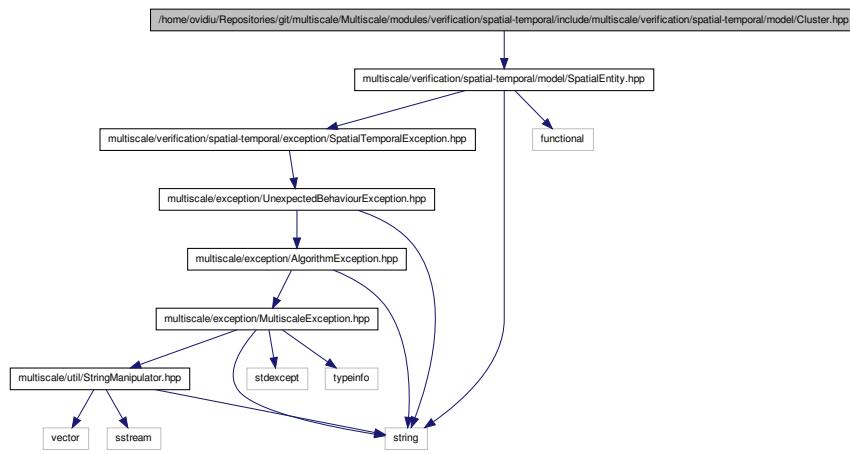
Class for representing a cluster of entities in an image.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.18 /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" Include dependency graph for Cluster.hpp:



Classes

- class [multiscale::verification::Cluster](#)

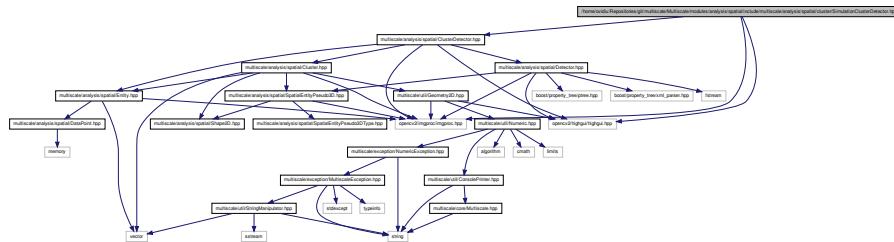
Class for representing a cluster.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.19 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/SimulationClusterDetector.hpp File Reference

#include "multiscale/analysis/spatial/ClusterDetector.-hpp" #include "opencv2/imgproc/imgproc.hpp" #include "opencv2/highgui/highgui.hpp"



Classes

- class [multiscale::analysis::SimulationClusterDetector](#)

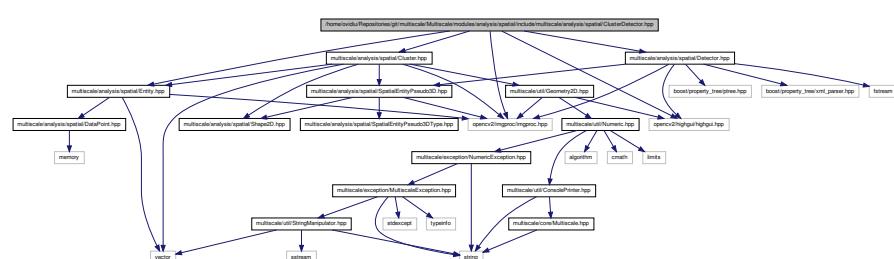
Class for detecting clusters in 2D images obtained from simulations.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.20 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/ClusterDetector.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 Cluster-
Detector.hpp:
```



Classes

- class [multiscale::analysis::ClusterDetector](#)

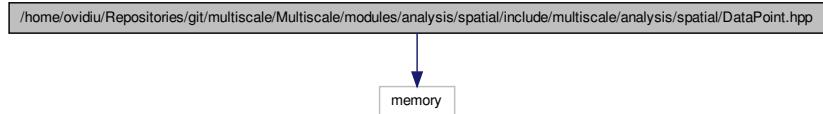
Class for detecting clusters in 2D images.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.21 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/DataPoint.hpp File Reference

#include <memory> Include dependency graph for DataPoint.hpp:



Classes

- class [multiscale::analysis::DataPoint](#)

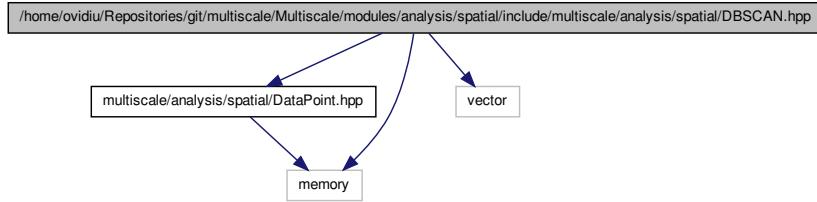
Class for representing a data point.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.22 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/DBSCAN.hpp File Reference

#include "multiscale/analysis/spatial/DataPoint.hpp" ×
#include <memory> #include <vector> Include dependency graph for



Classes

- class [multiscale::analysis::DBSCAN](#)

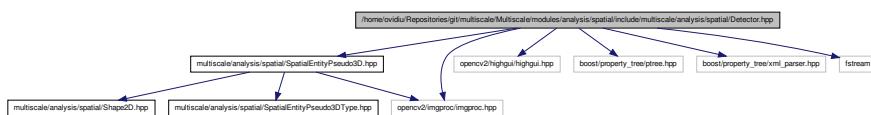
Class which implements an improved version of the [DBSCAN](#) algorithm.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.23 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/include/multiscale/analysis/spatial/Detector.hpp File Reference

```
#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp" #include "opencv2/imgproc/imgproc.hpp" #include "opencv2/highgui/highgui.hpp" #include <boost/property_tree/ptree.hpp> #include <boost/property_tree/xml_parser.hpp> #include <fstream> Include dependency graph for Detector.hpp:
```



Classes

- class [multiscale::analysis::Detector](#)

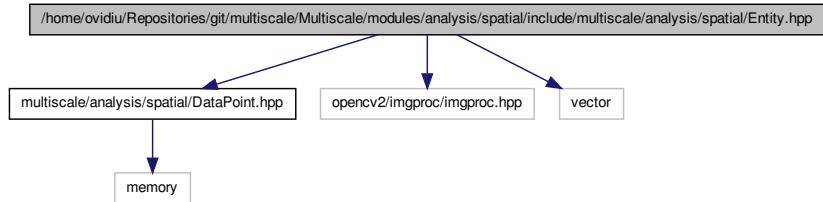
Abstract class for detecting entities of interest in images.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.24 /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](#)

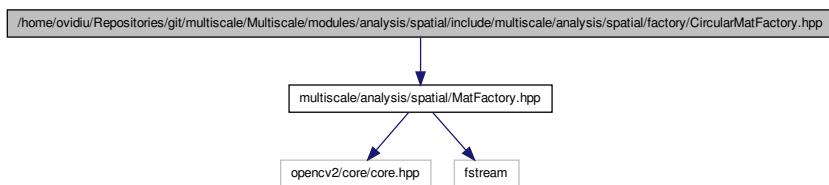
Class for representing an entity in an image (e.g. cell, organism etc.)

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.25 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/CircularMatFactory.hpp File Reference

```
#include "multiscale/analysis/spatial/MatFactory.hpp" ×
```



Classes

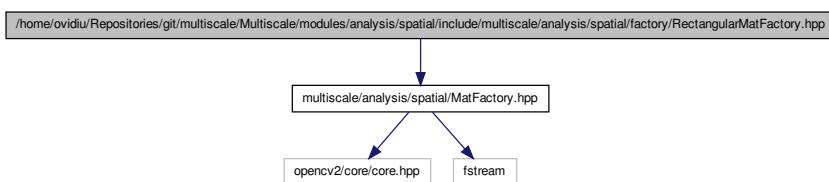
- class [multiscale::analysis::CircularMatFactory](#)
Class for creating a Mat object considering a circular grid.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.26 /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

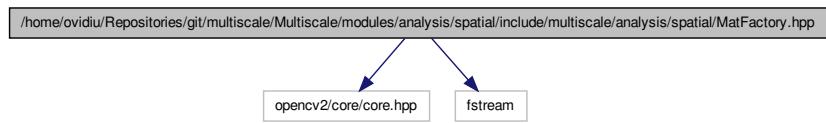
- class [multiscale::analysis::RectangularMatFactory](#)
Class for creating a Mat object considering a rectangular grid.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.27 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/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 Mat object.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

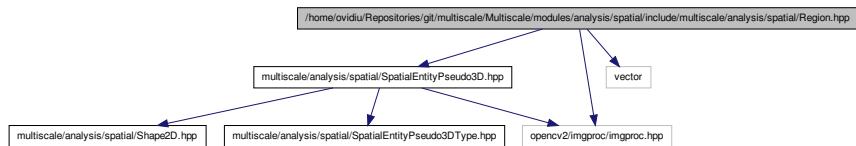
8.28 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/Region.hpp File Reference

#include "multiscale/analysis/spatial/SpatialEntityPseudo3D.hpp" #include "opencv2/imgproc/imgproc.hpp" #include

8.29

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/Region.hpp File Reference > Include dependency graph for Region.hpp:

1017



Classes

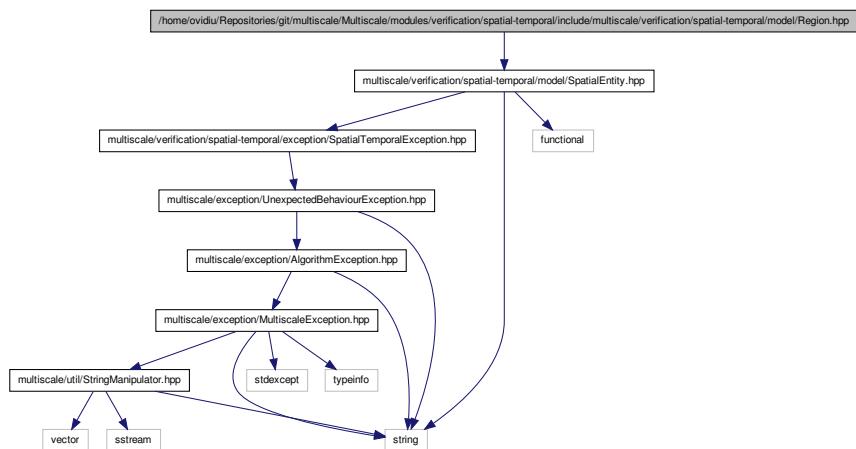
- class multiscale::analysis::Region
Class for representing a region.

Namespaces

- namespace multiscale
- namespace multiscale::analysis

8.29 /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](#)

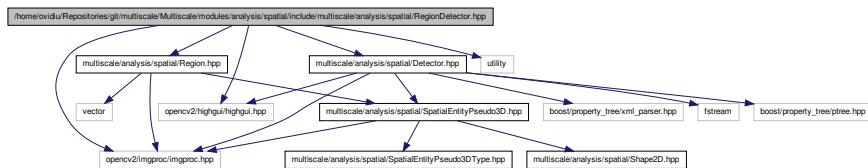
Class for representing a region.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.30 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/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> x
Include dependency graph for RegionDetector.hpp:
```



Classes

- class [multiscale::analysis::RegionDetector](#)

Class for detecting regions of high intensity in grayscale images.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

Typedefs

- `typedef std::pair< std::vector < Point >, std::vector < std::vector < Point > >>`
[multiscale::analysis::Polygon](#)

8.31 /home/ovidiu.Repositories/git/multiscale/-

Multiscale/modules/analysis/spatial/include/multiscale/analysis/spatial/Shape2-

D.hpp File

8.31 /h
Reference

rules/a

Shape2D.hpp File Reference

Namespaces

- namespace multiscale
 - namespace multiscale::analysis

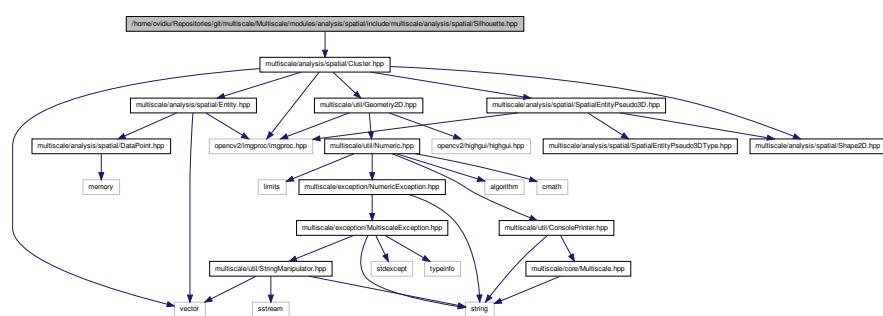
Enumerations

- enum multiscale::analysis::Shape2D { multiscale::analysis::Triangle = 1, multiscale::analysis::Rectangle = 2, multiscale::analysis::Circle = 3, multiscale::analysis::Undefined = 4 }

Enumeration for determining the type of a 2D shape.

8.32 /home/ovidiu/Repositories/git/Silhouette.hpp File Reference

```
#include "multiscale/analysis/spatial/Cluster.hpp"    Include  
dependency graph for Silhouette.hpp:
```



Glasses

- class `multiscale::analysis::Silhouette`

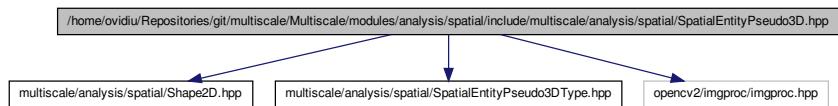
Class for computing the "Silhouette" clustering index.

Namespaces

- namespace multiscale
 - namespace multiscale::analysis

8.33 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/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

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

8.34 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/include/SpatialEntityPseudo3DType.hpp File Reference

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::analysis](#)

Enumerations

- enum [multiscale::analysis::SpatialEntityPseudo3DType](#) { [multiscale::analysis::Cluster](#) = 1, [multiscale::analysis::Region](#) = 2 }
- Enumeration for determining the type of a pseudo 3D entity.*

8.35 /home/ovidiu/Repositories/git/multiscale/-

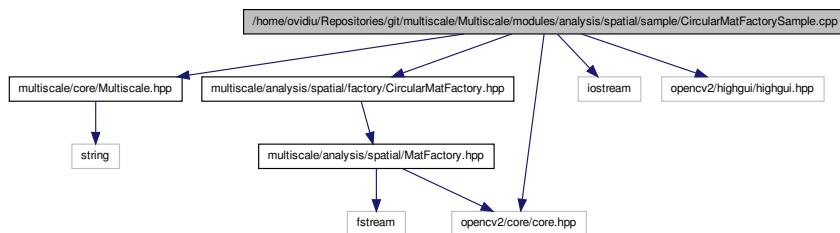
Multiscale/modules/analysis/spatial/sample/CircularMatFactorySample.cpp File

Reference

1021

8.35 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/-
CircularMatFactorySample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/analysis/spatial/fact  
CircularMatFactory.hpp" #include <iostream> #include <opencv2/core/core.-  
hpp> #include <opencv2/highgui/highgui.hpp> Include dependency  
graph for CircularMatFactorySample.cpp:
```



Functions

- int main ()

8.35.1 Function Documentation

8.35.1.1 int main ()

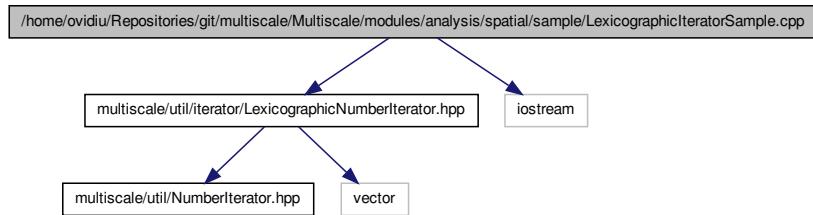
Definition at line 14 of file CircularMatFactorySample.cpp.

References multiscale::analysis::CircularMatFactory::createFromViewerImage(), multiscale::EXEC_SUCCESS_CODE, and multiscale::analysis::CircularMatFactory::maxColourBarIntensityFromViewerImage().

8.36 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/-
LexicographicIteratorSample.cpp File Reference

```
#include "multiscale/util/iterator/LexicographicNumber-  
Iterator.hpp" #include <iostream> Include dependency graph for -
```

LexicographicIteratorSample.cpp:



Functions

- int `main ()`

8.36.1 Function Documentation

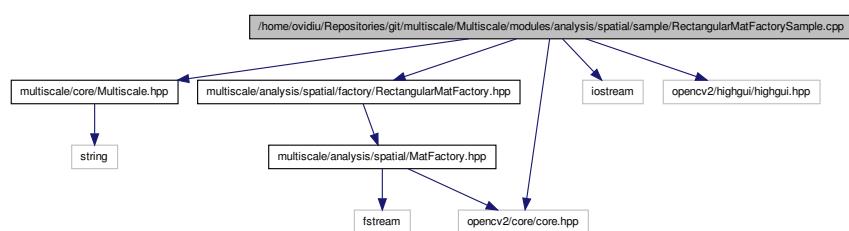
8.36.1.1 int main ()

Definition at line 8 of file LexicographicIteratorSample.cpp.

References `multiscale::NumberIterator::hasNext()`, and `multiscale::LexicographicNumberIterator::number()`.

8.37 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/sample/RectangularMatFactorySample.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/analysis/spatial/RectangularMatFactory.hpp" #include <iostream> #include <opencv2/core/core.hpp> #include <opencv2/highgui/highgui.hpp> Include dependency graph for RectangularMatFactorySample.cpp:
```



- int `main ()`

8.37.1 Function Documentation

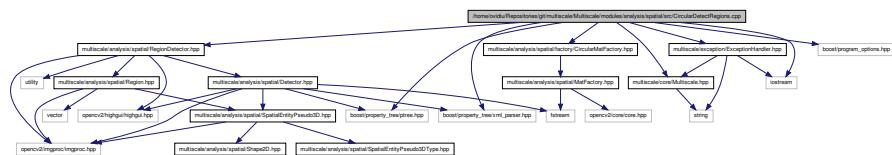
8.37.1.1 int main ()

Definition at line 14 of file RectangularMatFactorySample.cpp.

References `multiscale::analysis::RectangularMatFactory::createFromViewerImage()`, `multiscale::EXEC_SUCCESS_CODE`, and `multiscale::analysis::RectangularMatFactory::maxColourBarIntensityFromViewerImage()`.

8.38 /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 de-  
pendency 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 (string &inputFilepath, 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 string `CONFIG_FILE` = "config/analysis/spatial/circular_region_detector.xml"
- const string `LABEL_ROOT_COMMENT` = "<xmlcomment>"
- const string `LABEL_ALPHA` = "detector.alpha"
- const string `LABEL_BETA` = "detector.beta"
- const string `LABEL_BLUR_KERNEL_SIZE` = "detector.blurKernelSize"
- const string `LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS` = "detector.morphologicalCloseIterations"
- const string `LABEL_EPSILON` = "detector.epsilon"
- const string `LABEL_REGION_AREA_THRESH` = "detector.regionAreaThresh"
- const string `LABEL_THRESHOLD_VALUE` = "detector.thresholdValue"
- const string `ROOT_COMMENT` = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

8.38.1 Function Documentation

8.38.1.1 `bool areValidParameters (string & inputFilepath, string & outputFilename, bool & debugFlag, int argc, char ** argv)`

Definition at line 72 of file CircularDetectRegions.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

Referenced by `main()`.

8.38.1.2 `po::variables_map initArgumentsConfig (po::options_description & usageDescription, int argc, char ** argv)`

Definition at line 48 of file CircularDetectRegions.cpp.

Referenced by `areValidParameters()`.

8.38.1.3 `void loadDetectorParameterValues (RegionDetector & detector)`

Definition at line 100 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_VALUE`, `multiscale::analysis::RegionDetector::setAlpha()`, `multiscale::analysis::RegionDetector::setBeta()`,

**8.38 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/CircularDetectRegions.cpp File**

Reference **1025**

multiscale::analysis::RegionDetector::setBlurKernelSize(), multiscale::analysis::RegionDetector::setEpsilon(), multiscale::analysis::RegionDetector::setMorphologicalCloselterations(), multiscale::analysis::RegionDetector::setRegionAreaThresh(), and multiscale::analysis::RegionDetector::setThresholdValue().

Referenced by loadDetectorParameterValues(), and main().

**8.38.1.4 void loadDetectorParameterValues (RegionDetector & detector, bool
debugMode)**

Definition at line 134 of file CircularDetectRegions.cpp.

References loadDetectorParameterValues().

8.38.1.5 int main (int argc, char ** argv)

Definition at line 146 of file CircularDetectRegions.cpp.

References areValidParameters(), multiscale::analysis::CircularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC_E-RR_CODE, multiscale::EXEC_SUCCESS_CODE, loadDetectorParameterValues(), multiscale::analysis::Detector::outputResults(), printWrongParameters(), and saveDetectorParameterValues().

**8.38.1.6 void printHelpInformation (const po::variables_map & vm, const
po::options_description & usageDescription)**

Definition at line 61 of file CircularDetectRegions.cpp.

Referenced by areValidParameters().

8.38.1.7 void printWrongParameters ()

Definition at line 66 of file CircularDetectRegions.cpp.

References multiscale::ERR_MSG.

Referenced by main().

8.38.1.8 void saveDetectorParameterValues (RegionDetector & detector)

Definition at line 115 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::getMorphologicalCloselterations(), 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().

8.38.1.9 void saveDetectorParameterValues (RegionDetector & detector, bool debugMode)

Definition at line 139 of file CircularDetectRegions.cpp.

References saveDetectorParameterValues().

8.38.2 Variable Documentation

8.38.2.1 const string CONFIG_FILE = "config/analysis/spatial/circular_region_detector.xml"

Definition at line 33 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.2 const string LABEL_ALPHA = "detector.alpha"

Definition at line 36 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.3 const string LABEL_BETA = "detector.beta"

Definition at line 37 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.4 const string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"

Definition at line 38 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.5 const string LABEL_EPSILON = "detector.epsilon"

Definition at line 40 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

**8.39 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/CircularDetectRegions.in.cpp File
Reference** **1027**

**8.38.2.6 const string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS =
"detector.morphologicalCloselterations"**

Definition at line 39 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.7 const string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"

Definition at line 41 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.38.2.8 const string LABEL_ROOT_COMMENT = "<xmlcomment>"

Definition at line 35 of file CircularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

8.38.2.9 const string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"

Definition at line 42 of file CircularDetectRegions.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

**8.38.2.10 const string ROOT_COMMENT = "Warning! This xml file was automatically
generated by a C++ program using the Boost PropertyTree library."**

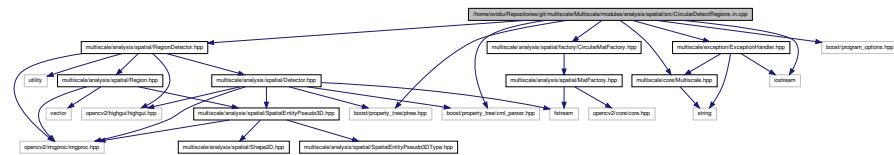
Definition at line 44 of file CircularDetectRegions.cpp.

Referenced by saveDetectorParameterValues().

8.39 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- CircularDetectRegions.in.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 de-
```

pendency graph for CircularDetectRegions.in.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 [isValidParameters](#) (string &inputFilepath, 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 string **CONFIG_FILE** = "\${CIRCULAR_REGION_DETECTOR_CONFIG_PATH}"
 - const string **LABEL_ROOT_COMMENT** = "<xmlcomment>"
 - const string **LABEL_ALPHA** = "detector.alpha"
 - const string **LABEL_BETA** = "detector.beta"
 - const string **LABEL_BLUR_KERNEL_SIZE** = "detector.blurKernelSize"
 - const string **LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS** = "detector.morphologicalCloseIterations"
 - const string **LABEL_EPSILON** = "detector.epsilon"
 - const string **LABEL_REGION_AREA_THRESH** = "detector.regionAreaThresh"
 - const string **LABEL_THRESHOLD_VALUE** = "detector.thresholdValue"
 - const 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 (string & *inputFilepath*, string & *outputFilename*, bool & *debugFlag*, int *argc*, char ** *argv*)

Definition at line 72 of file CircularDetectRegions.in.cpp.

References initArgumentsConfig(), and printHelpInformation().

8.39.1.2 po::variables_map initArgumentsConfig (po::options_description & *usageDescription*, int *argc*, char ** *argv*)

Definition at line 48 of file CircularDetectRegions.in.cpp.

8.39.1.3 void loadDetectorParameterValues (RegionDetector & *detector*)

Definition at line 100 of file CircularDetectRegions.in.cpp.

References CONFIG_FILE, LABEL_ALPHA, LABEL_BETA, LABEL_BLUR_KERNEL_SIZE, LABEL_EPSILON, LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS, LABEL_REGION_AREA_THRESH, LABEL_THRESHOLD_VALUE, multiscale::analysis::RegionDetector::setAlpha(), multiscale::analysis::RegionDetector::setBeta(), multiscale::analysis::RegionDetector::setBlurKernelSize(), multiscale::analysis::RegionDetector::setEpsilon(), multiscale::analysis::RegionDetector::setMorphologicalCloselterations(), multiscale::analysis::RegionDetector::setRegionAreaThresh(), and multiscale::analysis::RegionDetector::setThresholdValue().

8.39.1.4 void loadDetectorParameterValues (RegionDetector & *detector*, bool *debugMode*)

Definition at line 134 of file CircularDetectRegions.in.cpp.

References loadDetectorParameterValues().

8.39.1.5 int main (int *argc*, char ** *argv*)

Definition at line 146 of file CircularDetectRegions.in.cpp.

References areValidParameters(), multiscale::analysis::CircularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC_ERROR_CODE, multiscale::EXEC_SUCCESS_CODE, loadDetectorParameterValues(), multiscale::analysis::Detector::outputResults(), printWrongParameters(), and saveDetectorParameterValues().

8.39.1.6 void printHelpInformation (const po::variables_map & *vm*, const po::options_description & *usageDescription*)

Definition at line 61 of file CircularDetectRegions.in.cpp.

8.39.1.7 void printWrongParameters()

Definition at line 66 of file CircularDetectRegions.in.cpp.

References multiscale::ERR_MSG.

8.39.1.8 void saveDetectorParameterValues(RegionDetector & detector)

Definition at line 115 of file CircularDetectRegions.in.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.

8.39.1.9 void saveDetectorParameterValues(RegionDetector & detector, bool debugMode)

Definition at line 139 of file CircularDetectRegions.in.cpp.

References saveDetectorParameterValues().

8.39.2 Variable Documentation

8.39.2.1 const string CONFIG_FILE = "\${CIRCULAR_REGION_DETECTOR_CONFIG_PATH}"

Definition at line 33 of file CircularDetectRegions.in.cpp.

8.39.2.2 const string LABEL_ALPHA = "detector.alpha"

Definition at line 36 of file CircularDetectRegions.in.cpp.

8.39.2.3 const string LABEL_BETA = "detector.beta"

Definition at line 37 of file CircularDetectRegions.in.cpp.

8.39.2.4 const string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"

Definition at line 38 of file CircularDetectRegions.in.cpp.

**8.40 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/CircularityMeasure.cpp File
Reference** **1031**
8.39.2.5 const string LABEL_EPSILON = "detector.epsilon"

Definition at line 40 of file CircularDetectRegions.in.cpp.

**8.39.2.6 const string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS =
"detector.morphologicalCloselterations"**

Definition at line 39 of file CircularDetectRegions.in.cpp.

8.39.2.7 const string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"

Definition at line 41 of file CircularDetectRegions.in.cpp.

8.39.2.8 const string LABEL_ROOT_COMMENT = "<xmlcomment>"

Definition at line 35 of file CircularDetectRegions.in.cpp.

8.39.2.9 const string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"

Definition at line 42 of file CircularDetectRegions.in.cpp.

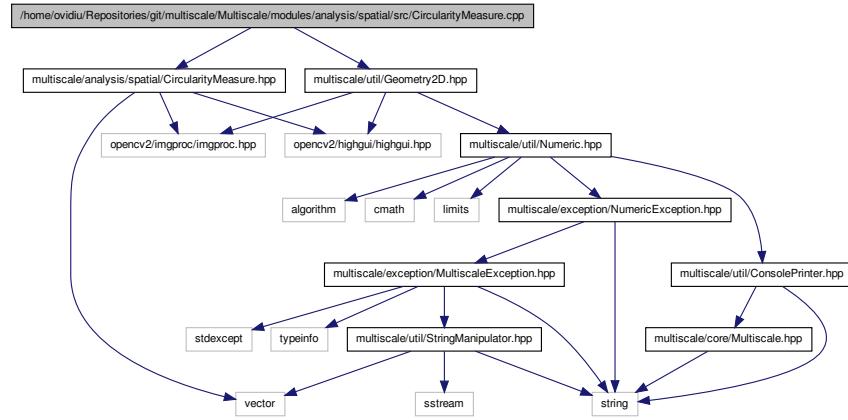
**8.39.2.10 const string ROOT_COMMENT = "Warning! This xml file was automatically
generated by a C++ program using the Boost PropertyTree library."**

Definition at line 44 of file CircularDetectRegions.in.cpp.

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 depen-
```

dency graph for CircularityMeasure.cpp:

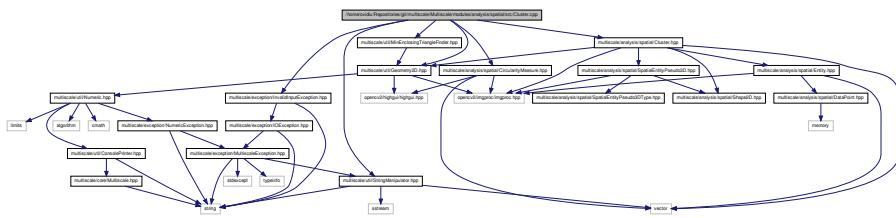


8.41 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/Cluster.cpp File Reference

```

#include "multiscale/analysis/spatial/CircularityMeasure.h"
#include "multiscale/analysis/spatial/Cluster.h"
#include "multiscale/exception/InvalidInputException.h"
#include "multiscale/util/Geometry2D.h"      #include
#include "multiscale/util/MinEnclosingTriangleFinder.h" #include
#include "multiscale/util/StringManipulator.h" Include dependency graph
for Cluster.cpp:

```



8.42 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/c SimulationClusterDetector.cpp File Reference

```

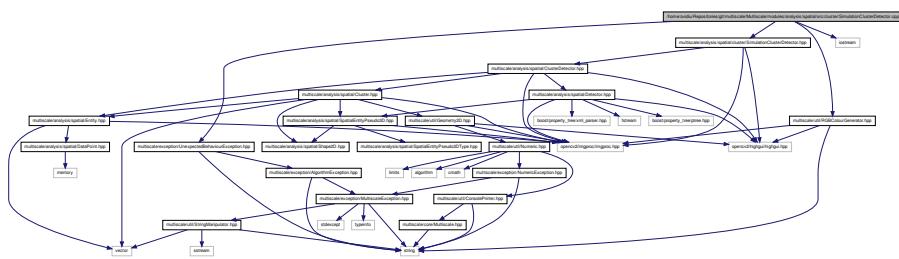
#include "multiscale/analysis/spatial/cluster/Simulation-
ClusterDetector.hpp"      #include "multiscale/exception/-

```

8.43 /home/ovidiu/Repositories/git/multiscale-/Multiscale/modules/analysis/spatial/src/ClusterDetector.cpp File

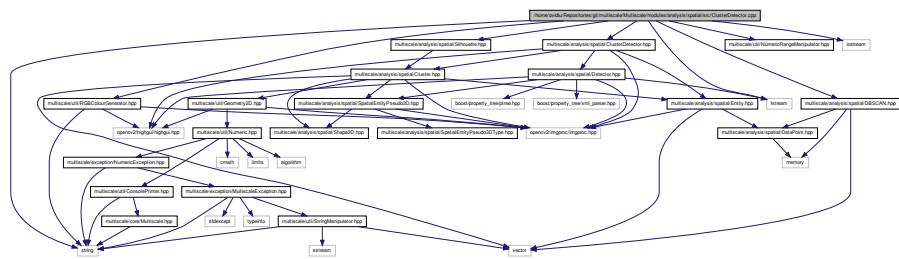
1033

```
UnexpectedBehaviourException.hpp" #include "multiscale/util/-  
RGBColourGenerator.hpp" #include <iostream> Include dependency  
graph for SimulationClusterDetector.cpp:
```



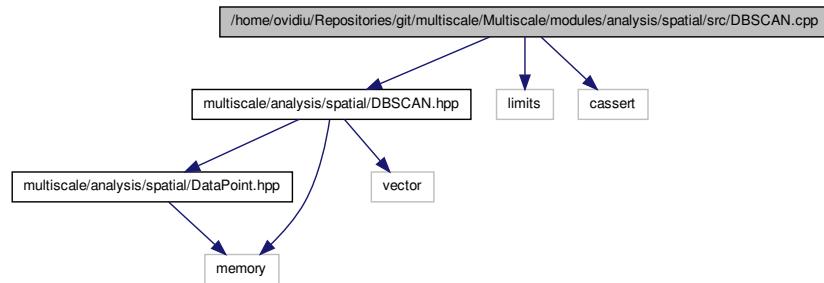
8.43 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-
ClusterDetector.cpp File Reference

```
#include "multiscale/analysis/spatial/ClusterDetector.-  
hpp" #include "multiscale/analysis/spatial/DBSCAN.hpp" x  
#include "multiscale/analysis/spatial/Silhouette.hpp" x  
#include "multiscale/util/NumericRangeManipulator.hpp" x  
#include "multiscale/util/RGBColourGenerator.hpp" #include  
<iostream> #include <fstream> #include <string> Include de-  
pendency graph for ClusterDetector.cpp:
```



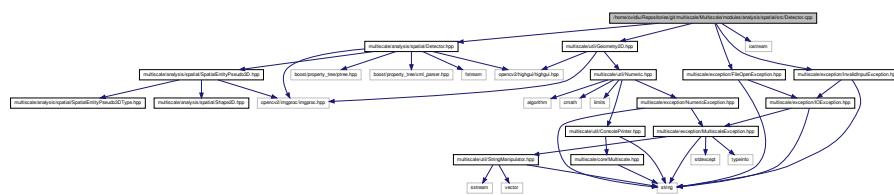
8.44 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-DBSCAN.cpp File Reference

```
#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:
```



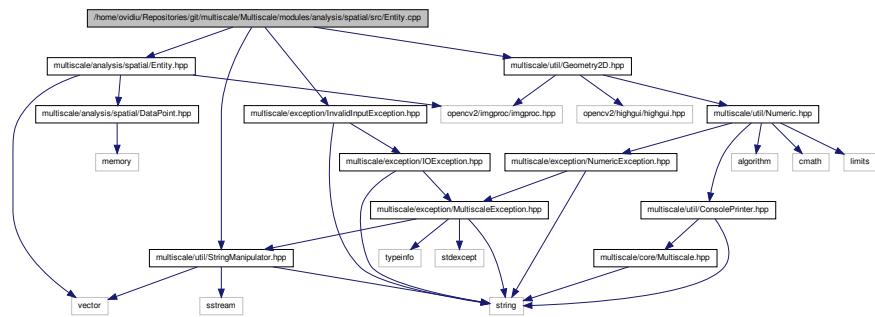
8.46 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Entity.cpp File Reference

```
#include "multiscale/analysis/spatial/Entity.hpp" #include
"multiscale/exception/InvalidInputException.hpp" #include
```

8.47 /home/ovidiu.Repositories/git/multiscale/- Multiscale/modules/analysis/spatial/src/factory/CircularMatFactory.cpp File Reference

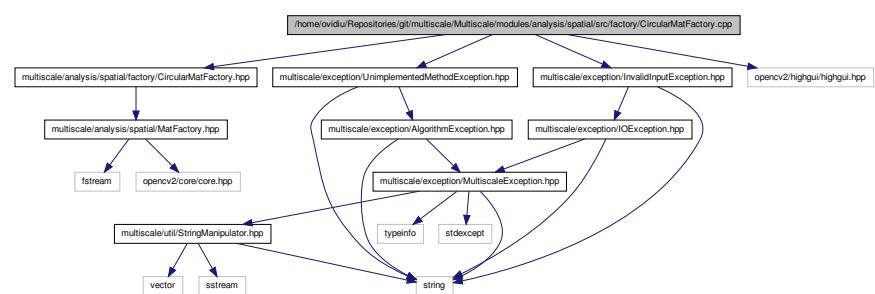
1035

#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

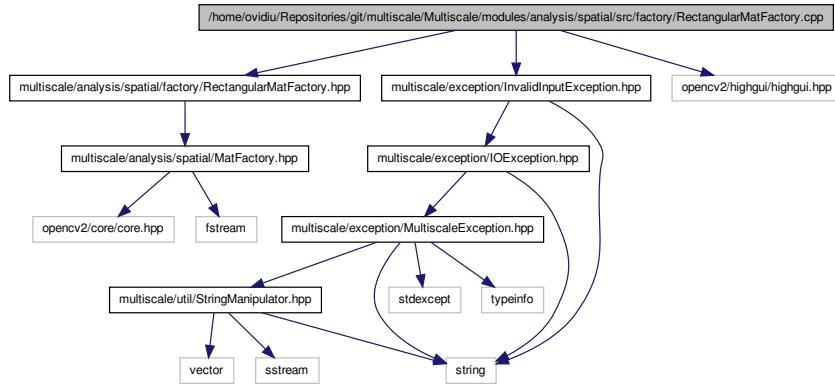
#include "multiscale/analysis/spatial/factory/CircularMatFactory.hpp" #include "multiscale/exception/InvalidInputException.hpp" #include "multiscale/exception/UnimplementedMethodException.hpp" #include "opencv2/highgui/highgui.hpp" Include dependency graph for CircularMatFactory.cpp:



8.48 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/factory/- RectangularMatFactory.cpp File Reference

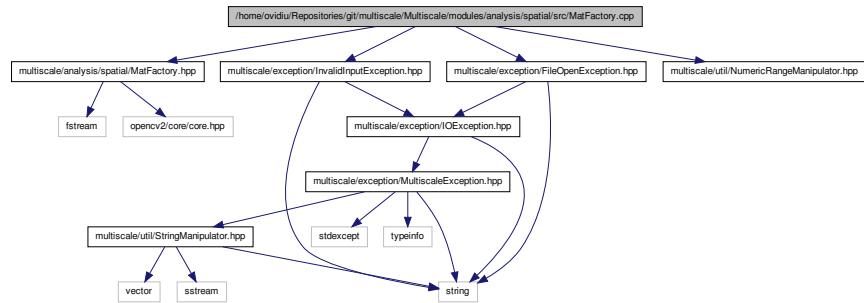
#include "multiscale/analysis/spatial/factory/RectangularMatFactory.hpp" #include "multiscale/exception/InvalidInputException.hpp" #include "opencv2/highgui/highgui.h-

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" x
#include "multiscale/exception/FileOpenException.hpp" x
#include "multiscale/exception/InvalidInputException.-
.hpp" #include "multiscale/util/NumericRangeManipulator.-
.hpp" Include dependency graph for MatFactory.cpp:
```



8.50 /home/ovidiu/Repositories/git/multiscale/-

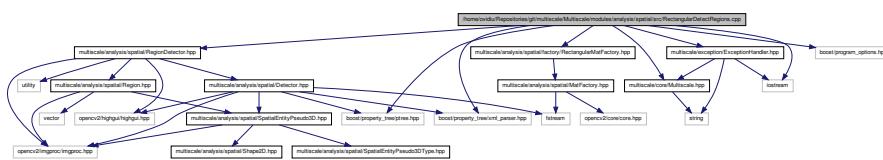
Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.cpp File

Reference

1037

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 de-  
pendency graph for RectangularDetectRegions.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 (string &inputFilepath, 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 string CONFIG_FILE = "config/analysis/spatial/rectangular_region_-
detector.xml"`
- `const string LABEL_ROOT_COMMENT = "<xmlcomment>"`
- `const string LABEL_ALPHA = "detector.alpha"`
- `const string LABEL_BETA = "detector.beta"`
- `const string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"`
- `const string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS = "detector.-
morphologicalCloseIterations"`
- `const string LABEL_EPSILON = "detector.epsilon"`

- const string `LABEL_REGION_AREA_THRESH` = "detector.regionAreaThresh"
- const string `LABEL_THRESHOLD_VALUE` = "detector.thresholdValue"
- const 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 (string & *inputfilepath*, string & *outputfilename*, bool & *debugFlag*, int *argc*, char ** *argv*)

Definition at line 72 of file RectangularDetectRegions.cpp.

References initArgumentsConfig(), and printHelpInformation().

8.50.1.2 po::variables_map initArgumentsConfig (po::options_description & *usageDescription*, int *argc*, char ** *argv*)

Definition at line 48 of file RectangularDetectRegions.cpp.

8.50.1.3 void loadDetectorParameterValues (RegionDetector & *detector*)

Definition at line 100 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_VALUE, multiscale::analysis::RegionDetector::setAlpha(), multiscale::analysis::RegionDetector::setBeta(), multiscale::analysis::RegionDetector::setBlurKernelSize(), multiscale::analysis::RegionDetector::setEpsilon(), multiscale::analysis::RegionDetector::setMorphologicalCloselterations(), multiscale::analysis::RegionDetector::setRegionAreaThresh(), and multiscale::analysis::RegionDetector::setThresholdValue().

8.50.1.4 void loadDetectorParameterValues (RegionDetector & *detector*, bool *debugMode*)

Definition at line 134 of file RectangularDetectRegions.cpp.

References loadDetectorParameterValues().

8.50.1.5 int main (int *argc*, char ** *argv*)

Definition at line 146 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(), printWrongParameters(), and saveDetectorParameterValues().

**8.50 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.cpp File
Reference** **1039**

**8.50.1.6 void printHelpInformation (const po::variables_map & vm, const
po::options_description & usageDescription)**

Definition at line 61 of file RectangularDetectRegions.cpp.

8.50.1.7 void printWrongParameters ()

Definition at line 66 of file RectangularDetectRegions.cpp.

References multiscale::ERR_MSG.

8.50.1.8 void saveDetectorParameterValues (RegionDetector & detector)

Definition at line 115 of file RectangularDetectRegions.cpp.

References CONFIG_FILE, multiscale::analysis::RegionDetector::getAlpha(), multiscale-
::analysis::RegionDetector::getBeta(), multiscale::analysis::RegionDetector::getBlur-
KernelSize(), multiscale::analysis::RegionDetector::getEpsilon(), multiscale::analysis-
::RegionDetector::getMorphologicalCloselterations(), multiscale::analysis::Region-
Detector::getRegionAreaThresh(), multiscale::analysis::RegionDetector::getThreshold-
Value(), 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.

**8.50.1.9 void saveDetectorParameterValues (RegionDetector & detector, bool
debugMode)**

Definition at line 139 of file RectangularDetectRegions.cpp.

References saveDetectorParameterValues().

8.50.2 Variable Documentation

**8.50.2.1 const string CONFIG_FILE = "config/analysis/spatial/rectangular_region_detector.-
xml"**

Definition at line 33 of file RectangularDetectRegions.cpp.

8.50.2.2 const string LABEL_ALPHA = "detector.alpha"

Definition at line 36 of file RectangularDetectRegions.cpp.

8.50.2.3 const string LABEL_BETA = "detector.beta"

Definition at line 37 of file RectangularDetectRegions.cpp.

8.50.2.4 const string **LABEL_BLUR_KERNEL_SIZE** = "detector.blurKernelSize"

Definition at line 38 of file RectangularDetectRegions.cpp.

8.50.2.5 const string **LABEL_EPSILON** = "detector.epsilon"

Definition at line 40 of file RectangularDetectRegions.cpp.

8.50.2.6 const string **LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS** =
"detector.morphologicalCloseIterations"

Definition at line 39 of file RectangularDetectRegions.cpp.

8.50.2.7 const string **LABEL_REGION_AREA_THRESH** = "detector.regionAreaThresh"

Definition at line 41 of file RectangularDetectRegions.cpp.

8.50.2.8 const string **LABEL_ROOT_COMMENT** = "<xmlcomment>"

Definition at line 35 of file RectangularDetectRegions.cpp.

8.50.2.9 const string **LABEL_THRESHOLD_VALUE** = "detector.thresholdValue"

Definition at line 42 of file RectangularDetectRegions.cpp.

8.50.2.10 const string **ROOT_COMMENT** = "Warning! This xml file was automatically
generated by a C++ program using the Boost PropertyTree library."

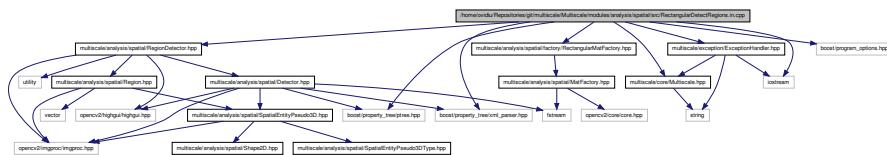
Definition at line 44 of file RectangularDetectRegions.cpp.

8.51 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- RectangularDetectRegions.in.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/analysis/spa-  
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
```

**8.51 /home/ovidiu/Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.in.cpp File
Reference** **1041**

<boost/program_options.hpp> #include <iostream> Include dependency graph for RectangularDetectRegions.in.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** (string &inputFilepath, 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 string **CONFIG_FILE** = "\${RECTANGULAR_REGION_DETECTOR_CONFIG_PATH}"
- const string **LABEL_ROOT_COMMENT** = "<xmlcomment>"
- const string **LABEL_ALPHA** = "detector.alpha"
- const string **LABEL_BETA** = "detector.beta"
- const string **LABEL_BLUR_KERNEL_SIZE** = "detector.blurKernelSize"
- const string **LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS** = "detector.morphologicalCloseIterations"
- const string **LABEL_EPSILON** = "detector.epsilon"
- const string **LABEL_REGION_AREA_THRESH** = "detector.regionAreaThresh"
- const string **LABEL_THRESHOLD_VALUE** = "detector.thresholdValue"
- const string **ROOT_COMMENT** = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

8.51.1 Function Documentation

8.51.1.1 `bool areValidParameters (string & inputfilepath, string & outputfilename, bool & debugFlag, int argc, char ** argv)`

Definition at line 72 of file RectangularDetectRegions.in.cpp.

References initArgumentsConfig(), and printHelpInformation().

8.51.1.2 `po::variables_map initArgumentsConfig (po::options_description & usageDescription, int argc, char ** argv)`

Definition at line 48 of file RectangularDetectRegions.in.cpp.

8.51.1.3 `void loadDetectorParameterValues (RegionDetector & detector)`

Definition at line 100 of file RectangularDetectRegions.in.cpp.

References CONFIG_FILE, LABEL_ALPHA, LABEL_BETA, LABEL_BLUR_KERNEL_SIZE, LABEL_EPSILON, LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS, LABEL_REGION_AREA_THRESH, LABEL_THRESHOLD_VALUE, multiscale::analysis::RegionDetector::setAlpha(), multiscale::analysis::RegionDetector::setBeta(), multiscale::analysis::RegionDetector::setBlurKernelSize(), multiscale::analysis::RegionDetector::setEpsilon(), multiscale::analysis::RegionDetector::setMorphologicalCloselterations(), multiscale::analysis::RegionDetector::setRegionAreaThresh(), and multiscale::analysis::RegionDetector::setThresholdValue().

8.51.1.4 `void loadDetectorParameterValues (RegionDetector & detector, bool debugMode)`

Definition at line 134 of file RectangularDetectRegions.in.cpp.

References loadDetectorParameterValues().

8.51.1.5 `int main (int argc, char ** argv)`

Definition at line 146 of file RectangularDetectRegions.in.cpp.

References areValidParameters(), multiscale::analysis::RectangularMatFactory::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, loadDetectorParameterValues(), multiscale::analysis::Detector::outputResults(), printWrongParameters(), and saveDetectorParameterValues().

8.51.1.6 `void printHelpInformation (const po::variables_map & vm, const po::options_description & usageDescription)`

Definition at line 61 of file RectangularDetectRegions.in.cpp.

**8.51 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/RectangularDetectRegions.in.cpp File
Reference 1043**
8.51.1.7 void printWrongParameters()

Definition at line 66 of file RectangularDetectRegions.in.cpp.

References multiscale::ERR_MSG.

8.51.1.8 void saveDetectorParameterValues(RegionDetector & detector)

Definition at line 115 of file RectangularDetectRegions.in.cpp.

References CONFIG_FILE, multiscale::analysis::RegionDetector::getAlpha(), multiscale::analysis::RegionDetector::getBeta(), multiscale::analysis::RegionDetector::getBlurKernelSize(), multiscale::analysis::RegionDetector::getEpsilon(), multiscale::analysis::RegionDetector::getMorphologicalCloselterations(), 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.

8.51.1.9 void saveDetectorParameterValues(RegionDetector & detector, bool debugMode)

Definition at line 139 of file RectangularDetectRegions.in.cpp.

References saveDetectorParameterValues().

8.51.2 Variable Documentation

**8.51.2.1 const string CONFIG_FILE = "\${RECTANGULAR_REGION_DETECTOR_CONFIG_PA-
TH}"**

Definition at line 33 of file RectangularDetectRegions.in.cpp.

8.51.2.2 const string LABEL_ALPHA = "detector.alpha"

Definition at line 36 of file RectangularDetectRegions.in.cpp.

8.51.2.3 const string LABEL_BETA = "detector.beta"

Definition at line 37 of file RectangularDetectRegions.in.cpp.

8.51.2.4 const string LABEL_BLUR_KERNEL_SIZE = "detector.blurKernelSize"

Definition at line 38 of file RectangularDetectRegions.in.cpp.

8.51.2.5 `const string LABEL_EPSILON = "detector.epsilon"`

Definition at line 40 of file RectangularDetectRegions.in.cpp.

8.51.2.6 `const string LABEL_MORPHOLOGICAL_CLOSE_ITERATIONS = "detector.morphologicalCloseIterations"`

Definition at line 39 of file RectangularDetectRegions.in.cpp.

8.51.2.7 `const string LABEL_REGION_AREA_THRESH = "detector.regionAreaThresh"`

Definition at line 41 of file RectangularDetectRegions.in.cpp.

8.51.2.8 `const string LABEL_ROOT_COMMENT = "<xmlcomment>"`

Definition at line 35 of file RectangularDetectRegions.in.cpp.

8.51.2.9 `const string LABEL_THRESHOLD_VALUE = "detector.thresholdValue"`

Definition at line 42 of file RectangularDetectRegions.in.cpp.

8.51.2.10 `const string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."`

Definition at line 44 of file RectangularDetectRegions.in.cpp.

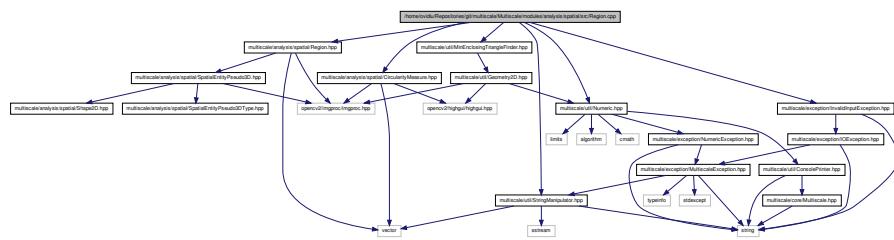
8.52 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Region.cpp File Reference

```
#include "multiscale/analysis/spatial/CircularityMeasure.-  
hpp" #include "multiscale/analysis/spatial/Region.hpp" x  
#include "multiscale/exception/InvalidInputException.-  
hpp" #include "multiscale/util/MinEnclosingTriangleFinder.-  
hpp" #include "multiscale/util/Numeric.hpp" #include "multiscale/util/-
```

8.53 /home/ovidiu/Repositories/git/multiscale-/Multiscale/modules/analysis/spatial/src/RegionDetector.cpp File Reference

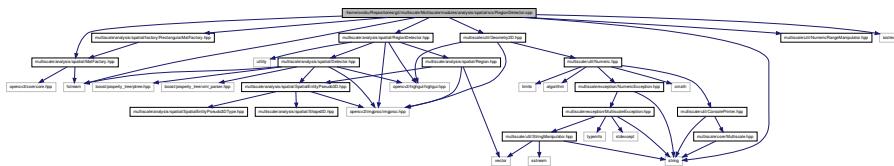
1045

`StringManipulator.hpp` Include dependency graph for `Region.cpp`:



8.53 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-RegionDetector.cpp File Reference

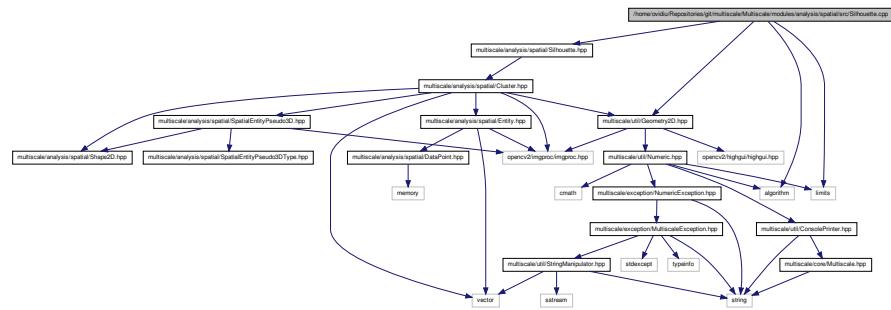
```
#include "multiscale/analysis/spatial/MatFactory.hpp"
#include "multiscale/analysis/spatial/factory/Rectangular-
MatFactory.hpp"    #include "multiscale/analysis/spatial/-
RegionDetector.hpp"    #include "multiscale/util/Numeric-
RangeManipulator.hpp" #include "multiscale/util/Geometry2-
D.hpp"  #include <iostream>  #include <fstream>  #include
<string> Include dependency graph for RegionDetector.cpp:
```



8.54 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/-Silhouette.cpp File Reference

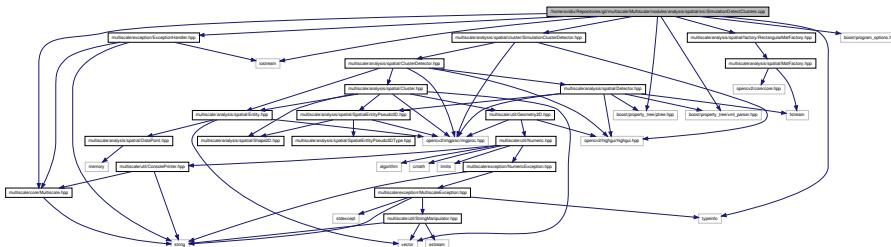
```
#include "multiscale/analysis/spatial/Silhouette.hpp"
#include "multiscale/util/Geometry2D.hpp" #include <algorithm> x
```

```
#include <limits> Include dependency graph for Silhouette.cpp:
```



8.55 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src/- SimulationDetectClusters.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/analysis/spa  
SimulationClusterDetector.hpp" #include "multiscale/analysis/spatial/factor  
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 `isValidParameters` (string &inputFilepath, string &outputFilename, bool &debugFlag, unsigned int &height, unsigned int &width, unsigned int &maxPileup, int argc, char **argv)

**8.55 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/SimulationDetectClusters.cpp File
Reference**

1047

-
- 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 string `CONFIG_FILE` = "config/analysis/spatial/simulation_cluster_-
detector.xml"
- const string `LABEL_ROOT_COMMENT` = "<xmlcomment>"
- const string `LABEL_EPS` = "detector.eps"
- const string `LABEL_MINPOINTS` = "detector.minPoints"
- const string `ROOT_COMMENT` = "Warning! This xml file was automatically gen-
erated by a C++ program using the Boost PropertyTree library."

8.55.1 Function Documentation

**8.55.1.1 bool areValidParameters (string & *inputFilepath*, string & *outputFilename*, bool &
debugFlag, unsigned int & *height*, unsigned int & *width*, unsigned int & *maxPileup*, int
argc, char ** *argv*)**

Definition at line 71 of file SimulationDetectClusters.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

**8.55.1.2 po::variables_map initArgumentsConfig (po::options_description &
usageDescription, int *argc*, char ** *argv*)**

Definition at line 44 of file SimulationDetectClusters.cpp.

**8.55.1.3 void loadDetectorParameterValues (SimulationClusterDetector & *detector*
)**

Definition at line 105 of file SimulationDetectClusters.cpp.

References `CONFIG_FILE`, `LABEL_EPS`, `LABEL_MINPOINTS`, `multiscale::analysis::-`
`ClusterDetector::setEps()`, and `multiscale::analysis::ClusterDetector::setMinPoints()`.

**8.55.1.4 void loadDetectorParameterValues (SimulationClusterDetector & *detector*,
bool *debugMode*)**

Definition at line 129 of file SimulationDetectClusters.cpp.

References `loadDetectorParameterValues()`.

8.55.1.5 int main (int argc, char ** argv)

Definition at line 141 of file SimulationDetectClusters.cpp.

References `isValidParameters()`, `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()`, `printWrongParameters()`, and `saveDetectorParameterValues()`.

8.55.1.6 void printHelpInformation (const po::variables_map & vm, const po::options_description & usageDescription)

Definition at line 60 of file SimulationDetectClusters.cpp.

8.55.1.7 void printWrongParameters ()

Definition at line 65 of file SimulationDetectClusters.cpp.

References `multiscale::ERR_MSG`.

8.55.1.8 void saveDetectorParameterValues (SimulationClusterDetector & detector)

Definition at line 115 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`.

8.55.1.9 void saveDetectorParameterValues (SimulationClusterDetector & detector, bool debugMode)

Definition at line 134 of file SimulationDetectClusters.cpp.

References `saveDetectorParameterValues()`.

8.55.2 Variable Documentation

8.55.2.1 const string CONFIG_FILE = "config/analysis/spatial/simulation_cluster_detector.xml"

Definition at line 34 of file SimulationDetectClusters.cpp.

8.55.2.2 const string LABEL_EPS = "detector.eps"

Definition at line 37 of file SimulationDetectClusters.cpp.

Referenced by loadDetectorParameterValues(), and saveDetectorParameterValues().

8.55.2.3 const string LABEL_MINPOINTS = "detector.minPoints"

Definition at line 38 of file SimulationDetectClusters.cpp.

Referenced by `loadDetectorParameterValues()`, and `saveDetectorParameterValues()`.

8.55.2.4 const string LABEL_ROOT_COMMENT = "<xmlcomment>"

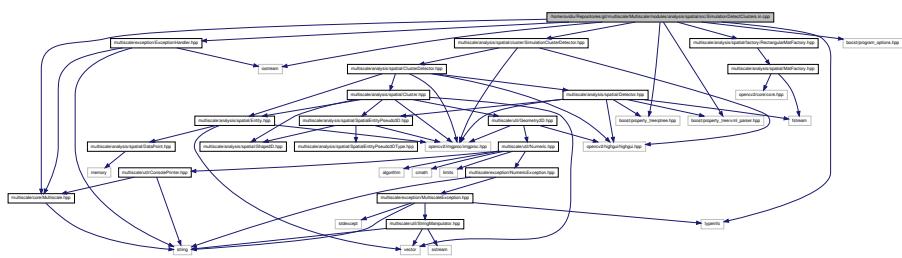
Definition at line 36 of file SimulationDetectClusters.cpp.

8.55.2.5 const string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

Definition at line 40 of file SimulationDetectClusters.cpp.

8.56 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/src-SimulationDetectClusters.in.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.in.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 `isValidParameters (string &inputfilepath, 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 string `CONFIG_FILE` = "\${SIMULATION_CLUSTER_DETECTOR_CONFIG_PATH}"
- const string `LABEL_ROOT_COMMENT` = "<xmlcomment>"
- const string `LABEL_EPS` = "detector.eps"
- const string `LABEL_MINPOINTS` = "detector.minPoints"
- const string `ROOT_COMMENT` = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

8.56.1 Function Documentation

8.56.1.1 bool `isValidParameters (string & inputfilepath, string & outputfilename, bool & debugFlag, unsigned int & height, unsigned int & width, unsigned int & maxPileup, int argc, char ** argv)`

Definition at line 71 of file SimulationDetectClusters.in.cpp.

References `initArgumentsConfig()`, and `printHelpInformation()`.

8.56.1.2 po::variables_map `initArgumentsConfig (po::options_description & usageDescription, int argc, char ** argv)`

Definition at line 44 of file SimulationDetectClusters.in.cpp.

8.56.1.3 void `loadDetectorParameterValues (SimulationClusterDetector & detector)`

Definition at line 105 of file SimulationDetectClusters.in.cpp.

References `CONFIG_FILE`, `LABEL_EPS`, `LABEL_MINPOINTS`, `multiscale::analysis::ClusterDetector::setEps()`, and `multiscale::analysis::ClusterDetector::setMinPoints()`.

**8.56 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/analysis/spatial/src/SimulationDetectClusters.in.cpp File
Reference 1051**
**8.56.1.4 void loadDetectorParameterValues (SimulationClusterDetector & detector,
bool debugMode)**

Definition at line 129 of file SimulationDetectClusters.in.cpp.

References loadDetectorParameterValues().

8.56.1.5 int main (int argc, char ** argv)

Definition at line 141 of file SimulationDetectClusters.in.cpp.

References areValidParameters(), multiscale::analysis::RectangularMatFactory-
::createFromViewerImage(), multiscale::analysis::Detector::detect(), multiscale::EXEC-
_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, loadDetectorParameterValues(),
multiscale::analysis::RectangularMatFactory::maxColourBarIntensityFromViewer-
Image(), multiscale::analysis::Detector::outputResults(), printWrongParameters(), and
saveDetectorParameterValues().

**8.56.1.6 void printHelpInformation (const po::variables_map & vm, const
po::options_description & usageDescription)**

Definition at line 60 of file SimulationDetectClusters.in.cpp.

8.56.1.7 void printWrongParameters ()

Definition at line 65 of file SimulationDetectClusters.in.cpp.

References multiscale::ERR_MSG.

**8.56.1.8 void saveDetectorParameterValues (SimulationClusterDetector & detector
)**

Definition at line 115 of file SimulationDetectClusters.in.cpp.

References CONFIG_FILE, multiscale::analysis::ClusterDetector::getEps(), multiscale-
::analysis::ClusterDetector::getMinPoints(), LABEL_EPS, LABEL_MINPOINTS, LABE-
L_ROOT_COMMENT, and ROOT_COMMENT.

**8.56.1.9 void saveDetectorParameterValues (SimulationClusterDetector & detector,
bool debugMode)**

Definition at line 134 of file SimulationDetectClusters.in.cpp.

References saveDetectorParameterValues().

8.56.2 Variable Documentation

8.56.2.1 const string CONFIG_FILE = "\${SIMULATION_CLUSTER_DETECTOR_CONFIG_PATH}"

Definition at line 34 of file SimulationDetectClusters.in.cpp.

8.56.2.2 const string LABEL_EPS = "detector.eps"

Definition at line 37 of file SimulationDetectClusters.in.cpp.

8.56.2.3 const string LABEL_MINPOINTS = "detector.minPoints"

Definition at line 38 of file SimulationDetectClusters.in.cpp.

8.56.2.4 const string LABEL_ROOT_COMMENT = "<xmlcomment>"

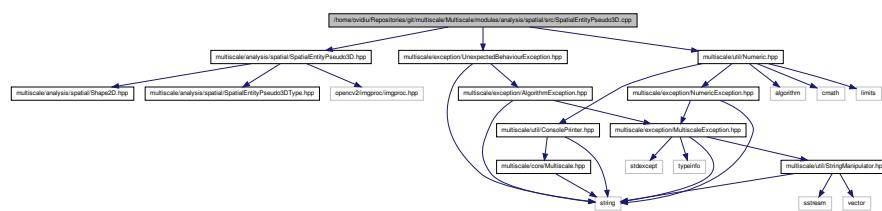
Definition at line 36 of file SimulationDetectClusters.in.cpp.

8.56.2.5 const string ROOT_COMMENT = "Warning! This xml file was automatically generated by a C++ program using the Boost PropertyTree library."

Definition at line 40 of file SimulationDetectClusters.in.cpp.

8.57 /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:
```



8.58 /home/ovidiu/Repositories/git/multiscale/-

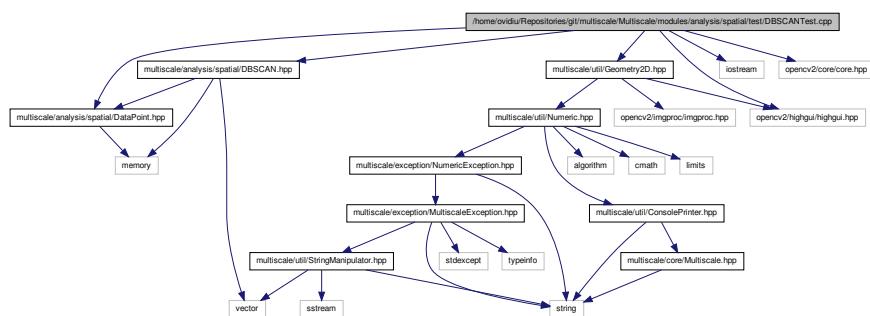
Multiscale/modules/analysis/spatial/test/DBSCANTest.cpp File

Reference

1053

8.58 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/analysis/spatial/test/-
DBSCANTest.cpp File Reference

```
#include "multiscale/analysis/spatial/DataPoint.hpp" x
#include "multiscale/analysis/spatial/DBSCAN.hpp" #include
"multiscale/util/Geometry2D.hpp"      #include <iostream>x
#include <opencv2/core/core.hpp>#include <opencv2/highgui/highgui.-
.hpp> Include dependency graph for DBSCANTest.cpp:
```



Classes

- class [EuclideanDataPoint](#)

Functions

- `vector< shared_ptr< DataPoint > > convertPoints (vector< EuclideanDataPoint > &points)`
- `void printResults (const vector< int > &clusterIndexes)`
- `void runTest (vector< EuclideanDataPoint > &points, double eps, int minPoints)`
- `void runTest1 ()`
- `void runTest2 ()`
- `void runTest3 ()`
- `void runTest4 ()`
- `void runTest5 ()`
- `void runTests ()`
- `int main ()`

8.58.1 Function Documentation

8.58.1.1 `vector<shared_ptr<DataPoint> > convertPoints (vector< EuclideanDataPoint > & points)`

Definition at line 37 of file DBSCANTest.cpp.

Referenced by runTest().

8.58.1.2 int main()

Definition at line 133 of file DBSCANTest.cpp.

References multiscale::EXEC_SUCCESS_CODE, and runTests().

8.58.1.3 void printResults(const vector< int > & clusterIndexes)

Definition at line 48 of file DBSCANTest.cpp.

Referenced by runTest().

8.58.1.4 void runTest(vector< EuclideanDataPoint > & points, double eps, int minPoints)

Definition at line 57 of file DBSCANTest.cpp.

References convertPoints(), printResults(), and multiscale::analysis::DBSCAN::run().

Referenced by runTest1(), runTest2(), runTest3(), runTest4(), and runTest5().

8.58.1.5 void runTest1()

Definition at line 67 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.58.1.6 void runTest2()

Definition at line 78 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.58.1.7 void runTest3()

Definition at line 95 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.58.1.8 void runTest4()

Definition at line 106 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.58.1.9 void runTest5()

Definition at line 113 of file DBSCANTest.cpp.

References runTest().

Referenced by runTests().

8.58.1.10 void runTests()

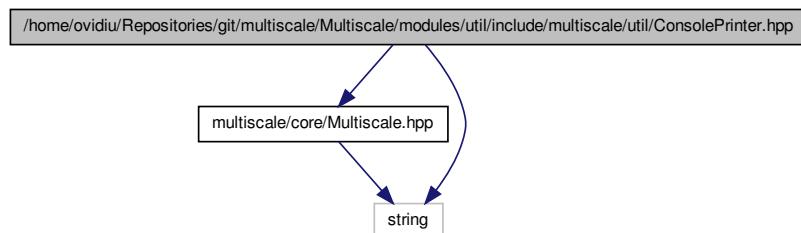
Definition at line 124 of file DBSCANTest.cpp.

References runTest1(), runTest2(), runTest3(), runTest4(), and runTest5().

Referenced by main().

8.59 /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

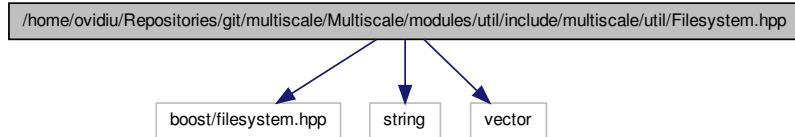
- namespace [multiscale](#)

Enumerations

- enum multiscale::UnixColourCode { multiscale::BLACK = 0, multiscale::RED = 1, multiscale::GREEN = 2, multiscale::YELLOW = 3, multiscale::BLUE = 4, multiscale::MAGENTA = 5, multiscale::CYAN = 6, multiscale::WHITE = 7 }
- enum multiscale::WindowsColourCode { multiscale::BLACK = 0, multiscale::DARK_BLUE = 1, multiscale::DARK_GREEN = 2, multiscale::DARK_CYAN = 3, multiscale::DARK_RED = 4, multiscale::DARK_MAGENTA = 5, multiscale::DARK_YELLOW = 6, multiscale::DARK_WHITE = 7, multiscale::GRAY = 8, multiscale::BLUE = 4, multiscale::GREEN = 2, multiscale::CYAN = 6, multiscale::RED = 1, multiscale::MAGENTA = 5, multiscale::YELLOW = 3, multiscale::WHITE = 7 }
- enum multiscale::ColourCode { multiscale::BLACK = 0, multiscale::RED = 1, multiscale::GREEN = 2, multiscale::YELLOW = 3, multiscale::BLUE = 4, multiscale::MAGENTA = 5, multiscale::CYAN = 6, multiscale::WHITE = 7 }

8.60 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/ filesystem.hpp 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.

Namespaces

- namespace [multiscale](#)

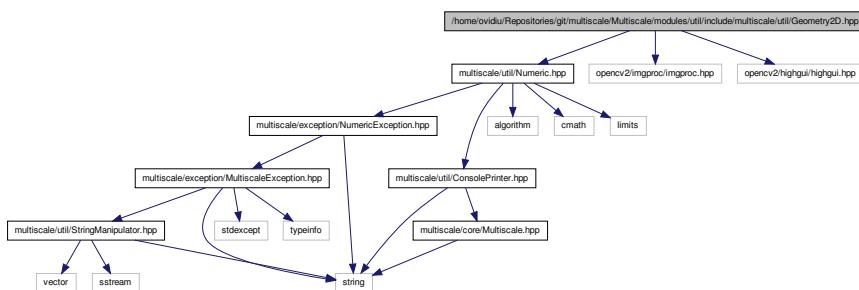
8.61 /home/ovidiu.Repositories/git/multiscale/-

Multiscale/modules/util/include/multiscale/util/Geometry2D.hpp File

Reference

8.61 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- Geometry2D.hpp File Reference 1057

```
#include "multiscale/util/Numeric.hpp" #include "opencv2/imgproc/imgproc.h"
#include "opencv2/highgui/highgui.hpp" Include dependency
graph for Geometry2D.hpp:
```



Classes

- class [multiscale::Geometry2D](#)

Two-dimensional geometric operations.

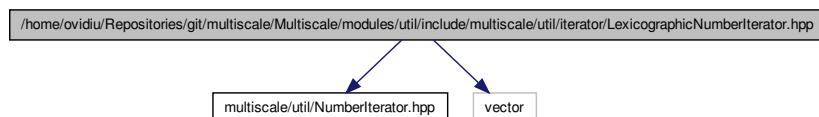
Namespaces

- namespace [multiscale](#)

8.62 /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

- namespace [multiscale](#)

8.63 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/NumberIteratorType.hpp File Reference

Namespaces

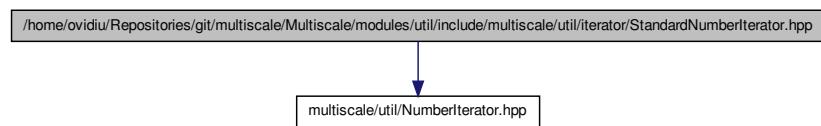
- namespace [multiscale](#)

Enumerations

- enum [multiscale::NumberIteratorType](#) { [multiscale::STANDARD](#) = 1, [multiscale::LEXICOGRAPHIC](#) = 2 }
The type of the number iterator.

8.64 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/StandardNumberIterator.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp" Include dependency graph for StandardNumberIterator.hpp:
```



Classes

- class [multiscale::StandardNumberIterator](#)

**8.65 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/util/include/multiscale/util/MinEnclosingTriangleFinder.hpp
File Reference** 1059

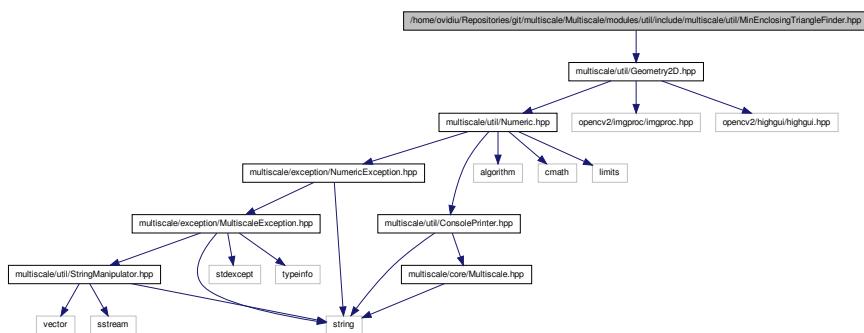
Iterator class starting at 1 and iterating over all natural numbers until the provided upper bound is reached.

Namespaces

- namespace [multiscale](#)

8.65 /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

- namespace [multiscale](#)

8.66 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- NumberIterator.hpp File Reference

Classes

- class [multiscale::NumberIterator](#)

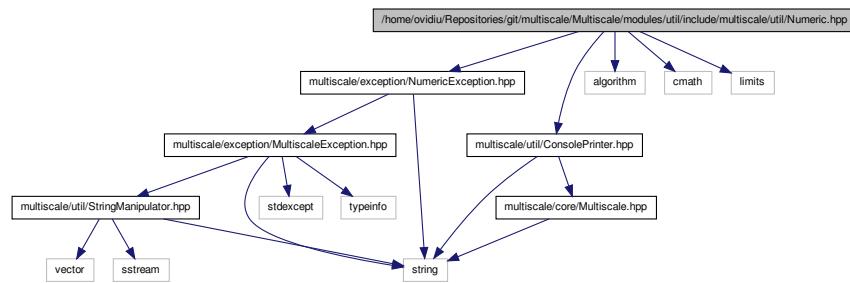
Abstract class representing a number iterator.

Namespaces

- namespace [multiscale](#)

8.67 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/Numeric.hpp File Reference

```
#include "multiscale/exception/NumericException.hpp" x
#include "multiscale/util/ConsolePrinter.hpp"      #include
<algorithm> #include <cmath> #include <limits> Include de-
pendency 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

- namespace [multiscale](#)

**8.68 /home/ovidiu.Repositories/git/multiscale/-
Multiscale/modules/util/include/multiscale/util/NumericRangeManipulator.hpp**
File Reference
**8.68 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
NumericRangeManipulator.hpp File Reference** 1061

Classes

- class [multiscale::NumericRangeManipulator](#)

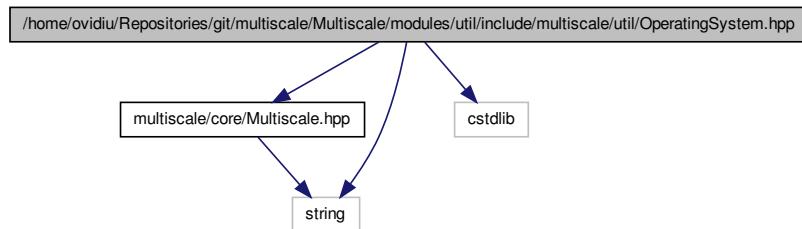
Operations for ranges of numeric values.

Namespaces

- namespace [multiscale](#)

**8.69 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/-
OperatingSystem.hpp File Reference**

```
#include "multiscale/core/Multiscale.hpp" #include <cstdlib> x
#include <string> Include dependency graph for OperatingSystem.hpp:
```



Classes

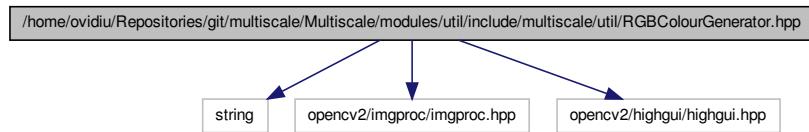
- class [multiscale::OperatingSystem](#)
Class for executing operating system related functions.

Namespaces

- namespace [multiscale](#)

8.70 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/RGBColourGenerator.hpp File Reference

```
#include <string> #include "opencv2/imgproc/imgproc.hpp" ×
#include "opencv2/highgui/highgui.hpp" Include dependency graph
for RGBColourGenerator.hpp:
```



Classes

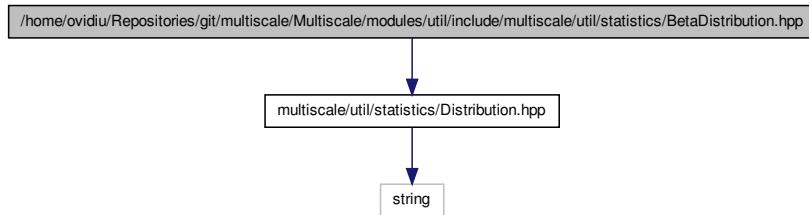
- class [multiscale::RGBColourGenerator](#)
Generate a RGB colour.

Namespaces

- namespace [multiscale](#)

8.71 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/BetaDistribution.hpp File Reference

```
#include "multiscale/util/statistics/Distribution.hpp" ×
Include dependency graph for BetaDistribution.hpp:
```



- class [multiscale::BetaDistribution](#)

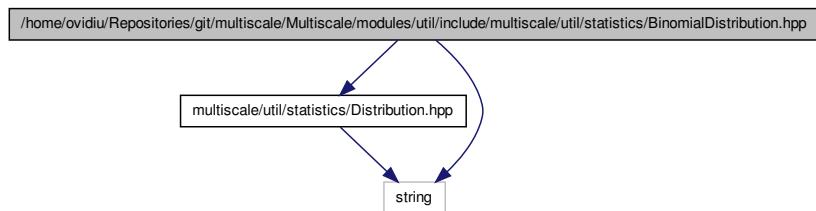
Class for analysing Beta distributed data.

Namespaces

- namespace [multiscale](#)

8.72 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/statistics/BinomialDistribution.hpp File Reference

```
#include "multiscale/util/statistics/Distribution.hpp" x
#include <string> Include dependency graph for BinomialDistribution.hpp:
```



Classes

- class [multiscale::BinomialDistribution](#)

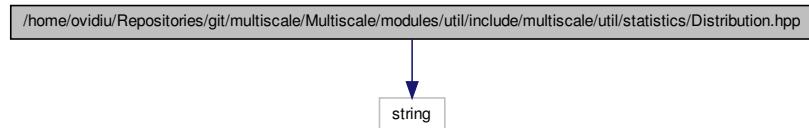
Class for analysing Binomial distributed data.

Namespaces

- namespace [multiscale](#)

8.73 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/Distribution.hpp File Reference

#include <string> Include dependency graph for Distribution.hpp:



Classes

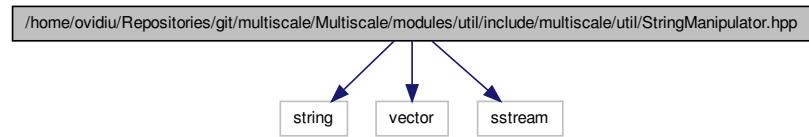
- class [multiscale::Distribution](#)

Namespaces

- namespace [multiscale](#)

8.74 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/StringManipulator.hpp File Reference

#include <string> #include <vector> #include <sstream> X
Include dependency graph for StringManipulator.hpp:



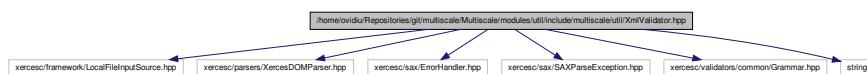
Classes

- class [multiscale::StringManipulator](#)
Class for manipulating strings.

- namespace multiscale

8.75 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/include/multiscale/util/- XmlValidator.hpp File Reference

```
#include <xercesc/framework/LocalFileInputSource.hpp> x
#include <xercesc/parsers/XercesDOMParser.hpp> #include
<xercesc/sax/Error Handler.hpp> #include <xercesc/sax/SAX-
ParseException.hpp> #include <xercesc/validators/common/-
Grammar.hpp> #include <string> Include dependency graph for Xml-
Validator.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.

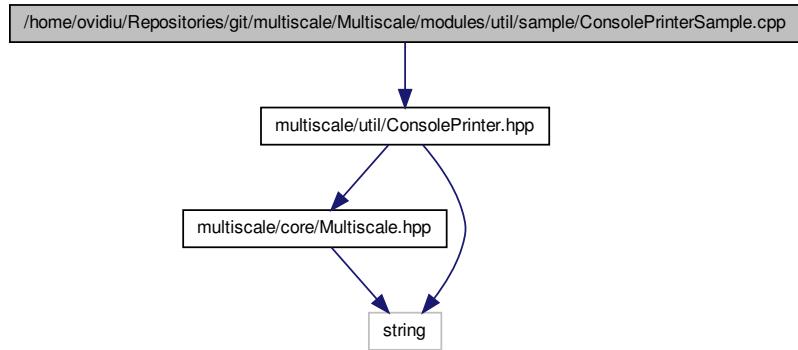
Namespaces

- namespace multiscale

8.76 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/- ConsolePrinterSample.cpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp" Include depen-
```

dency graph for ConsolePrinterSample.cpp:



Functions

- int `main ()`

Variables

- const std::string `SAMPLE_TAG` = "[SAMPLE]"
- const std::string `SAMPLE_MSG` = "This is a sample message."

8.76.1 Function Documentation

8.76.1.1 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.

8.76.2 Variable Documentation

8.76.2.1 const std::string SAMPLE_MSG = "This is a sample message."

Definition at line 6 of file ConsolePrinterSample.cpp.

Referenced by `main()`.

**8.77 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-
ExecuteProgramSample.cpp File**

Reference

1067

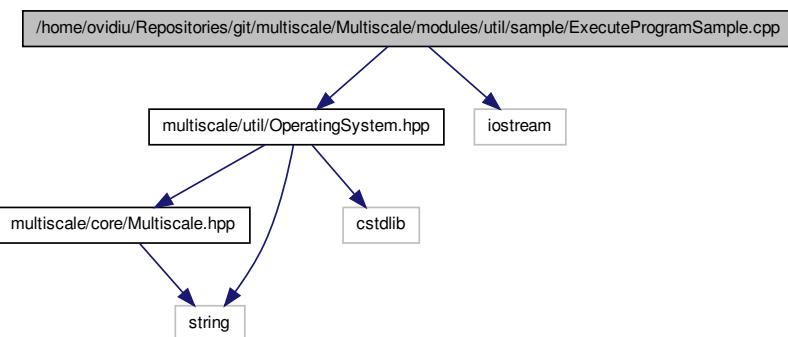
8.76.2.2 const std::string SAMPLE_TAG = "[SAMPLE]"

Definition at line 5 of file ConsolePrinterSample.cpp.

Referenced by main().

**8.77 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-
ExecuteProgramSample.cpp File Reference**

```
#include "multiscale/util/OperatingSystem.hpp"      #include  
<iostream> Include dependency graph for ExecuteProgram.cpp:
```



Functions

- int [main](#) (int argc, char **argv)

8.77.1 Function Documentation

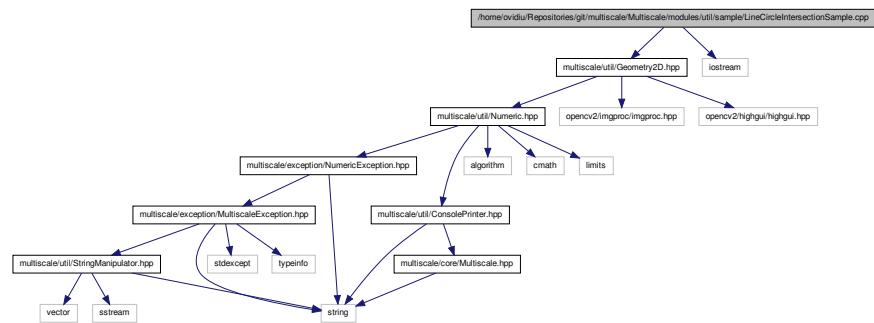
8.77.1.1 int [main](#) (int *argc*, char ** *argv*)

Definition at line 9 of file ExecuteProgramSample.cpp.

References multiscale::EXEC_SUCCESS_CODE, and multiscale::OperatingSystem::executeProgram().

8.78 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-LineCircleIntersectionSample.cpp File Reference

```
#include "multiscale/util/Geometry2D.hpp" #include <iostream> x
Include dependency graph for LineCircleIntersectionSample.cpp:
```



Functions

- void [printPoints](#) (const vector< Point2f > &points)
- int [main](#) ()

8.78.1 Function Documentation

8.78.1.1 int [main](#) ()

Definition at line 22 of file LineCircleIntersectionSample.cpp.

References multiscale::EXEC_SUCCESS_CODE, and printPoints().

8.78.1.2 void [printPoints](#) (const vector< Point2f > & *points*)

Definition at line 12 of file LineCircleIntersectionSample.cpp.

Referenced by main().

8.79 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/sample/-MinEnclosingTriangleFinderSample.cpp File Reference

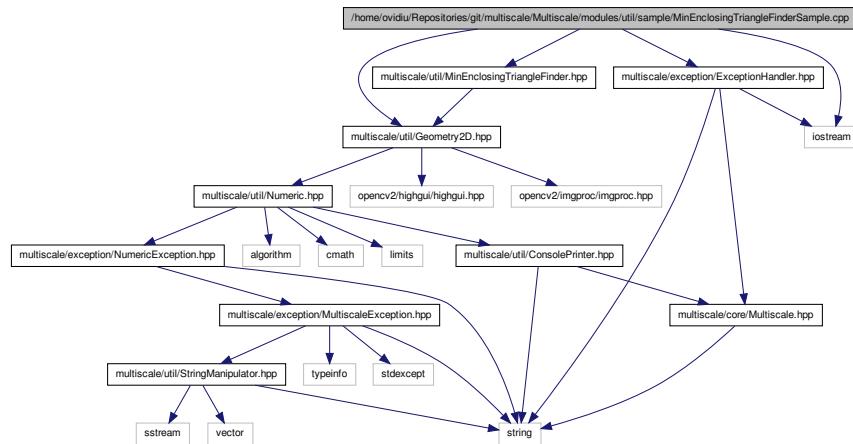
```
#include "multiscale/exception/ExceptionHandler.hpp" x
#include "multiscale/util/Geometry2D.hpp" #include "multiscale/util/-
```

8.79 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/sample/-MinEnclosingTriangleFinderSample.cpp File

Reference

1069

MinEnclosingTriangleFinder.hpp" #include <iostream> include dependency graph for MinEnclosingTriangleFinderSample.cpp:



Functions

- vector< Point2f > [generateRandomSetOf2DPoints](#) (int nrOfPoints)
- void [printPolygon](#) (const vector< Point2f > &points)
- void [outputMinEnclosingTriangleFinderResults](#) (const vector< Point2f > &min-
EnclosingTriangle, const vector< Point2f > &points)
- bool [arePointsEnclosed](#) (const vector< Point2f > &points, const vector< Point2f > &triangle)
- bool [isTriangleTouchingPolygon](#) (const vector< Point2f > &convexPolygon, const
vector< Point2f > &triangle)
- bool [isOneEdgeFlush](#) (const vector< Point2f > &convexPolygon, const vector<
Point2f > &triangle)
- bool [isValidTriangle](#) (const vector< Point2f > &points, const vector< Point2f >
&triangle)
- void [runMinEnclosingTriangleFinder](#) (const vector< Point2f > &points)
- void [runMinEnclosingTriangleFinderUsingRandomPolygons](#) ()
- void [runMinEnclosingTriangleFinder](#) ()
- int [main](#) (int argc, char **argv)

Variables

- const 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

8.79.1 Function Documentation

8.79.1.1 bool arePointsEnclosed (const vector< Point2f > & *points*, const vector< Point2f > & *triangle*)

Definition at line 78 of file MinEnclosingTriangleFinderSample.cpp.

References POINT_IN_TRIANGLE_THRESH.

Referenced by isValidTriangle().

8.79.1.2 vector<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().

8.79.1.3 bool isOneEdgeFlush (const vector< Point2f > & *convexPolygon*, const vector< Point2f > & *triangle*)

Definition at line 116 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by isValidTriangle().

8.79.1.4 bool isTriangleTouchingPolygon (const vector< Point2f > & *convexPolygon*, const vector< Point2f > & *triangle*)

Definition at line 93 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by isValidTriangle().

8.79.1.5 bool isValidTriangle (const vector< Point2f > & *points*, const vector< Point2f > & *triangle*)

Definition at line 134 of file MinEnclosingTriangleFinderSample.cpp.

References arePointsEnclosed(), isOneEdgeFlush(), and isTriangleTouchingPolygon().

Referenced by runMinEnclosingTriangleFinder().

8.79.1.6 `int main (int argc, char ** argv)`

Definition at line 189 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::EXEC_SUCCESS_CODE, and runMinEnclosingTriangleFinder().

8.79.1.7 `void outputMinEnclosingTriangleFinderResults (const vector< Point2f > & minEnclosingTriangle, const vector< 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().

8.79.1.8 `void printPolygon (const vector< Point2f > & points)`

Definition at line 38 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

8.79.1.9 `void runMinEnclosingTriangleFinder (const vector< Point2f > & points)`

Definition at line 147 of file MinEnclosingTriangleFinderSample.cpp.

References multiscale::MinEnclosingTriangleFinder::find(), isValidTriangle(), and outputMinEnclosingTriangleFinderResults().

Referenced by main(), and runMinEnclosingTriangleFinderUsingRandomPolygons().

8.79.1.10 `void runMinEnclosingTriangleFinder ()`

Definition at line 184 of file MinEnclosingTriangleFinderSample.cpp.

References runMinEnclosingTriangleFinderUsingRandomPolygons().

8.79.1.11 `void runMinEnclosingTriangleFinderUsingRandomPolygons ()`

Definition at line 163 of file MinEnclosingTriangleFinderSample.cpp.

References generateRandomSetOf2DPoints(), KEY_ESC, MAX_POLYGON_POINTS, and runMinEnclosingTriangleFinder().

Referenced by runMinEnclosingTriangleFinder().

8.79.2 Variable Documentation

8.79.2.1 const int KEY_ESC = 27

Definition at line 13 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by multiscale::analysis::Detector::detectInDebugMode(), and runMinEnclosingTriangleFinderUsingRandomPolygons().

8.79.2.2 const int LINE_THICKNESS = 50

Definition at line 16 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

8.79.2.3 const int MAX_POLYGON_POINTS = 100

Definition at line 18 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by runMinEnclosingTriangleFinderUsingRandomPolygons().

8.79.2.4 const int NR_RAND_POLYGONS = 50

Definition at line 17 of file MinEnclosingTriangleFinderSample.cpp.

8.79.2.5 const double POINT_IN_TRIANGLE_THRESH = 1E-4

Definition at line 22 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by arePointsEnclosed(), and multiscaletest::MinEnclosingTriangleFinderTest::ArePointsEnclosed().

8.79.2.6 const int POLYGON_POINT_X_MAX = 500

Definition at line 19 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by generateRandomSetOf2DPoints(), and outputMinEnclosingTriangleFinderResults().

8.79.2.7 const int POLYGON_POINT_Y_MAX = 500

Definition at line 20 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by generateRandomSetOf2DPoints(), and outputMinEnclosingTriangleFinderResults().

8.79.2.8 const int RADIUS = 1

Definition at line 15 of file MinEnclosingTriangleFinderSample.cpp.

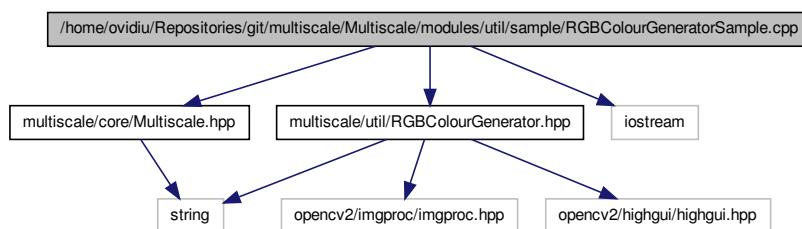
8.79.2.9 const string WIN_MIN_AREA_TRIANGLE = "Minimum area enclosing triangle"

Definition at line 11 of file MinEnclosingTriangleFinderSample.cpp.

Referenced by outputMinEnclosingTriangleFinderResults().

8.80 /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 ()`

8.80.1 Function Documentation

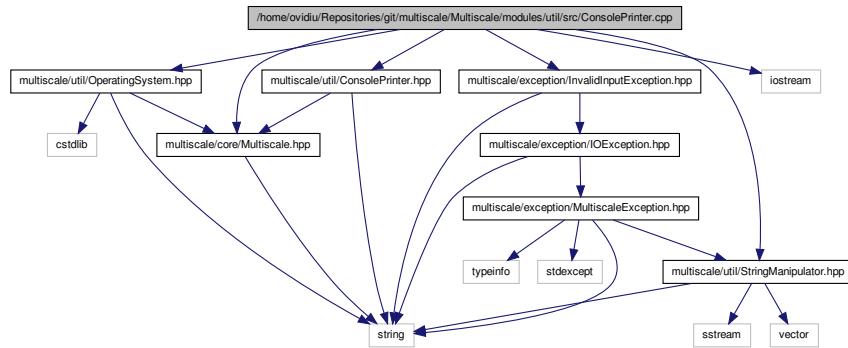
8.80.1.1 int main ()

Definition at line 9 of file RGBColourGeneratorSample.cpp.

References `multiscale::EXEC_SUCCESS_CODE`, and `multiscale::RGBColourGenerator::generate()`.

8.81 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/- ConsolePrinter.cpp File Reference

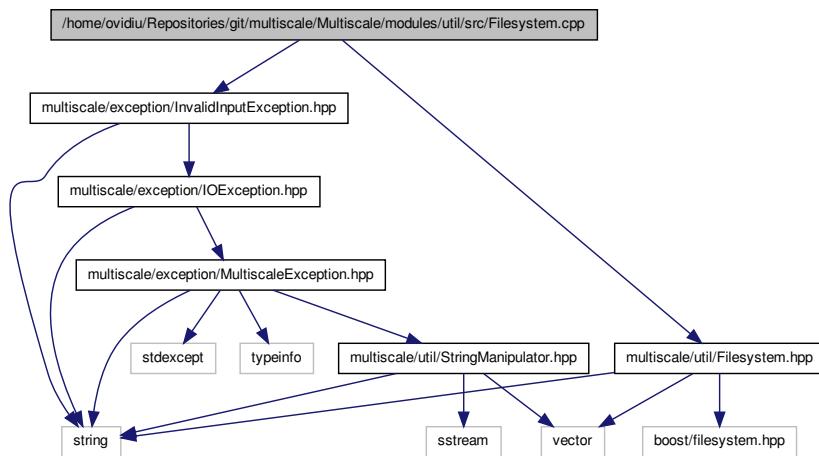
```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/exception/-  
InvalidInputException.hpp"      #include "multiscale/util/-  
ConsolePrinter.hpp"    #include "multiscale/util/Operating-  
System.hpp"   #include "multiscale/util/StringManipulator.-  
hpp" #include <iostream> Include dependency graph for ConsolePrinter.cpp:
```



8.82 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/- Filesystem.cpp File Reference

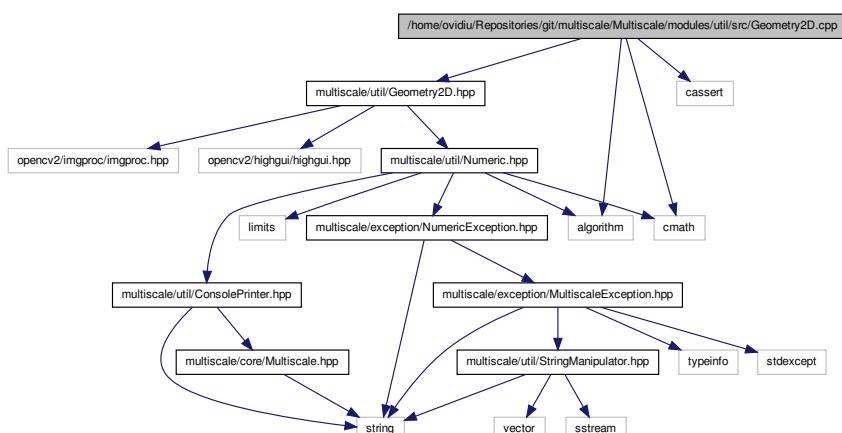
```
#include "multiscale/exception/InvalidInputException.-  
hpp" #include "multiscale/util/Filesystem.hpp" Include depen-
```

dependency graph for Filesystem.cpp:



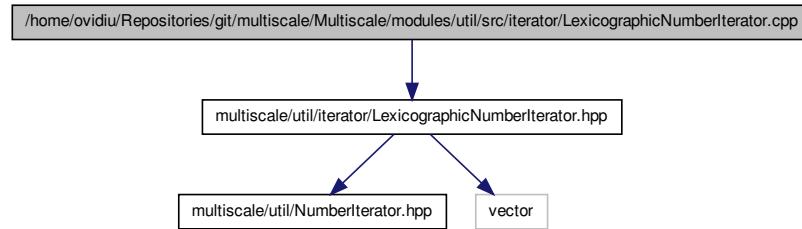
8.83 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/src/- Geometry2D.cpp File Reference

```
#include "multiscale/util/Geometry2D.hpp" #include <algorithm>
#include <cassert> #include <cmath> Include dependency graph for
Geometry2D.cpp:
```



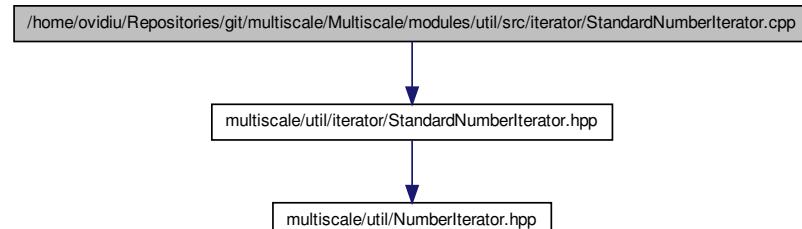
8.84 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/iterator/- LexicographicNumberIterator.cpp File Reference

```
#include "multiscale/util/iterator/LexicographicNumber-
Iterator.hpp" Include dependency graph for LexicographicNumberIterator.cpp:
```



8.85 /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:
```



8.86 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/- MinEnclosingTriangleFinder.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/exception/UnexpectedBehaviour-
```

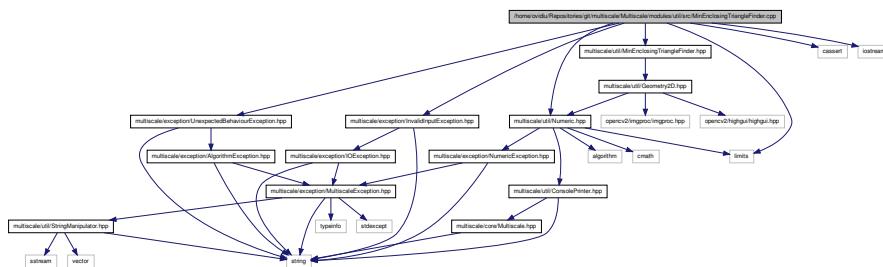
8.87 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/-

NumberIterator.cpp File

Reference

1077

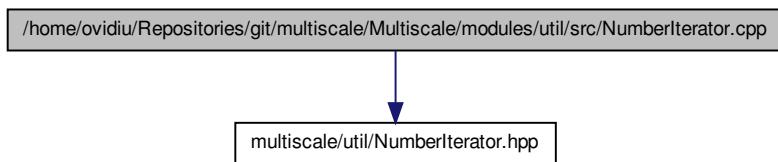
```
Exception.hpp" #include "multiscale/util/MinEnclosing-
TriangleFinder.hpp" #include "multiscale/util/Numeric.-
hpp" #include <cassert> #include <iostream> #include
<limits> Include dependency graph for MinEnclosingTriangleFinder.cpp:
```



8.87 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/-

NumberIterator.cpp File Reference

```
#include "multiscale/util/NumberIterator.hpp" Include dependency graph for NumberIterator.cpp:
```

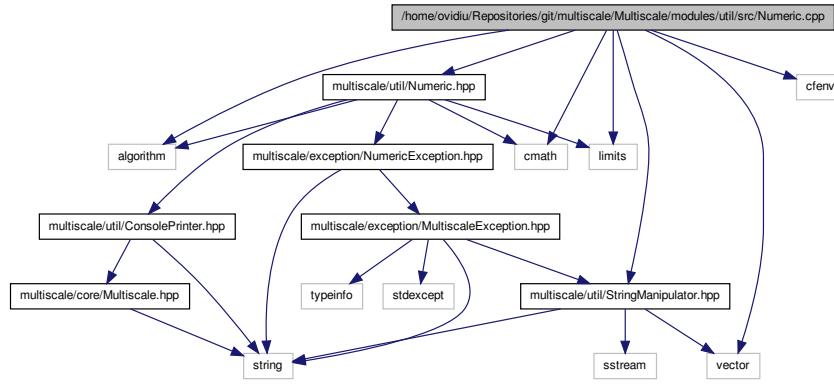


8.88 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/-

Numeric.cpp File Reference

```
#include "multiscale/util/Numeric.hpp" #include "multiscale/util/-
StringManipulator.hpp" #include <algorithm> #include <cfenv> x
#include <cmath> #include <limits> #include <vector> x
```

Include dependency graph for Numeric.cpp:

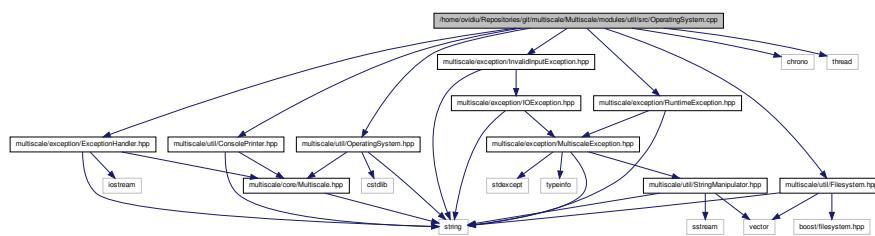


8.89 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/-OperatingSystem.cpp File Reference

```

#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:

```



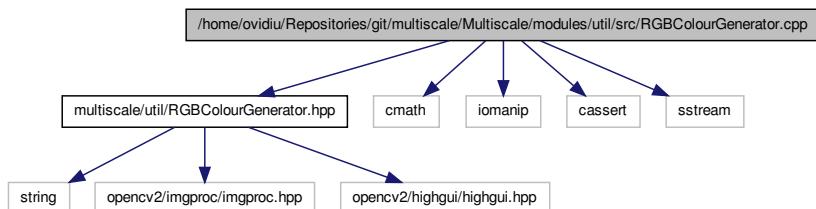
8.90 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/-RGBColourGenerator.cpp File Reference

```
#include "multiscale/util/RGBColourGenerator.hpp" #include
```

8.91 /home/ovidiu/Repositories/git/multiscale/- Multiscale/modules/util/src/statistics/BetaDistribution.cpp File Reference

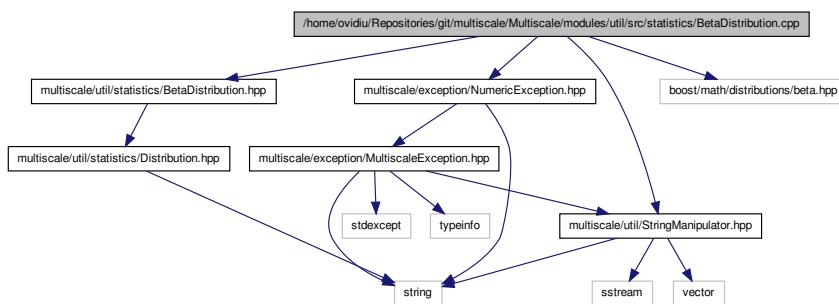
1079

```
<cmath> #include <iomanip> #include <cassert> #include  
<sstream> Include dependency graph for RGBColourGenerator.cpp:
```



8.91 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/- BetaDistribution.cpp File Reference

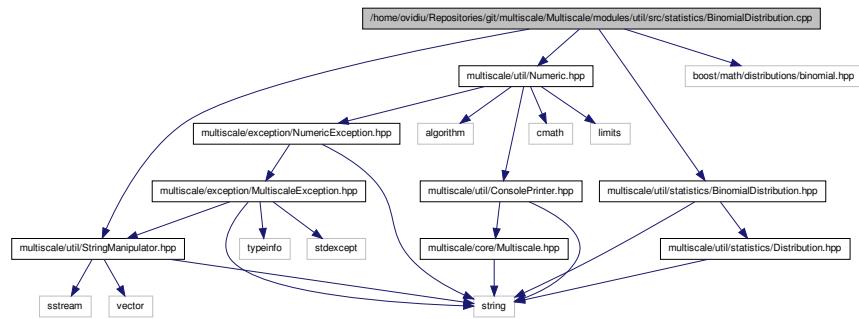
```
#include "multiscale/exception/NumericException.hpp" x  
#include "multiscale/util/StringManipulator.hpp" #include  
"multiscale/util/statistics/BetaDistribution.hpp" #include  
<boost/math/distributions/beta.hpp> Include dependency graph for  
BetaDistribution.cpp:
```



8.92 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/- BinomialDistribution.cpp File Reference

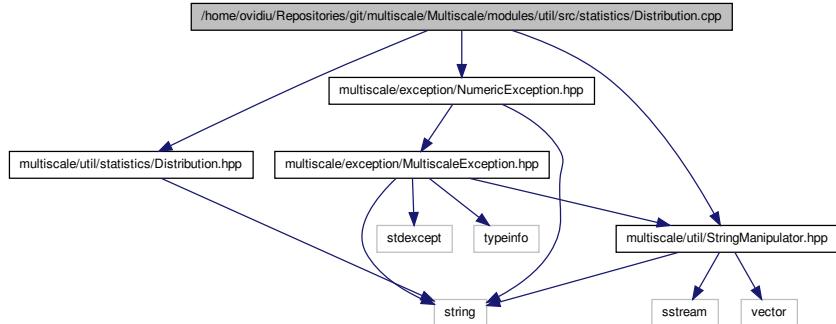
```
#include "multiscale/util/Numeric.hpp" #include "multiscale/util/-  
StringManipulator.hpp" #include "multiscale/util/statistics/-  
BinomialDistribution.hpp" #include <boost/math/distributions/binomial.-
```

hpp> Include dependency graph for BinomialDistribution.cpp:



8.93 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src/statistics/-Distribution.cpp File Reference

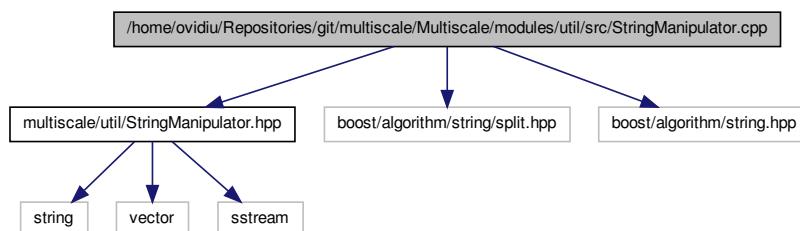
```
#include "multiscale/exception/NumericException.hpp" 
#include "multiscale/util/StringManipulator.hpp" #include 
"multiscale/util/statistics/Distribution.hpp" Include dependency graph for Distribution.cpp:
```



8.94 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src-/StringManipulator.cpp File Reference

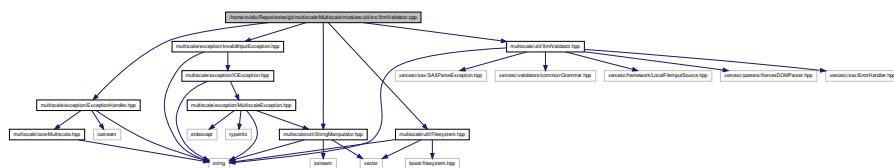
```
#include "multiscale/util/StringManipulator.hpp" #include 
<boost/algorithm/string/split.hpp> #include <boost/algorithm/string.-
```

hpp> Include dependency graph for StringManipulator.cpp:



8.95 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/util/src-/XmlValidator.cpp File Reference

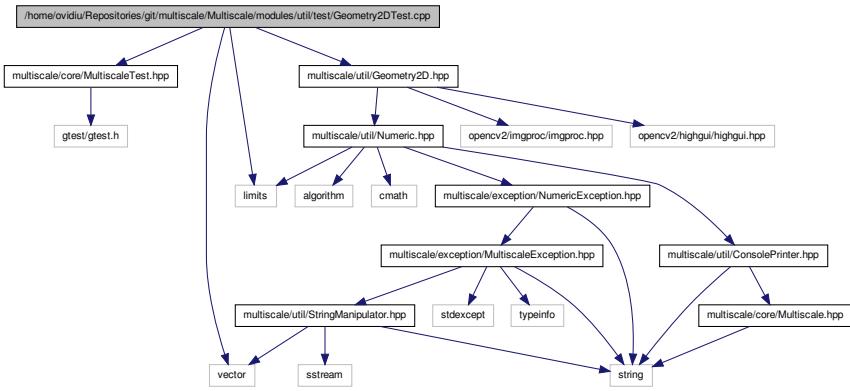
```
#include "multiscale/exception/ExceptionHandler.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/util/Filesystem.hpp" #include
"multiscale/util/StringManipulator.hpp" #include "multiscale/util/-
XmlValidator.hpp" Include dependency graph for XmlValidator.cpp:
```



8.96 /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](#)

8.96.1 Function Documentation

8.96.1.1 int main (int argc, char ** argv)

Definition at line 33 of file Geometry2DTest.cpp.

8.96.1.2 TEST (Geometry2D , TriangleArea)

Definition at line 16 of file Geometry2DTest.cpp.

References DOUBLE_COMP_ERROR.

8.96.1.3 TEST (Geometry2D , PointOnLineSegment)

Definition at line 23 of file Geometry2DTest.cpp.

8.96.2 Variable Documentation

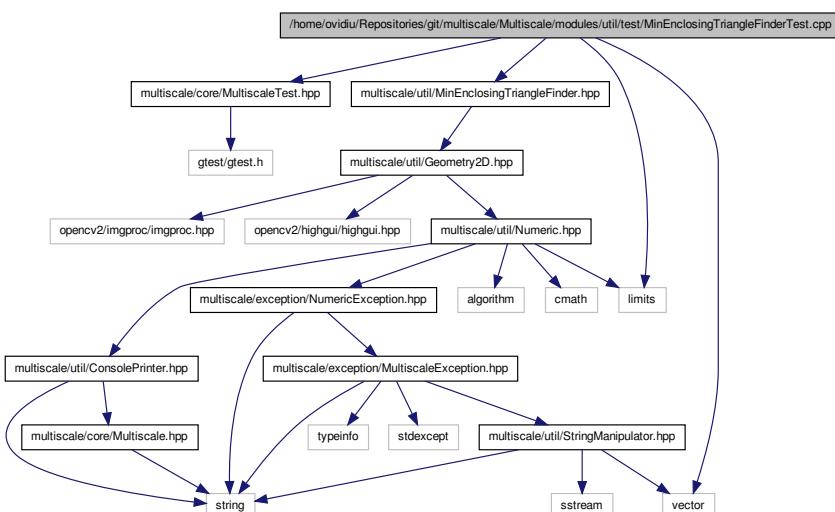
8.96.2.1 const double DOUBLE_COMP_ERROR = 1E-6

Definition at line 12 of file Geometry2DTest.cpp.

Referenced by TEST().

8.97 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/- MinEnclosingTriangleFinderTest.cpp File Reference

```
#include "multiscale/core/MultiscaleTest.hpp"      #include
#include "multiscale/util/MinEnclosingTriangleFinder.hpp" #include
<limits> #include <vector> Include dependency graph for MinEnclosing-
TriangleFinderTest.cpp:
```



Classes

- class [multiscaletest::MinEnclosingTriangleFinderTest](#)

Class for testing the minimum enclosing triangle algorithm.

Namespaces

- namespace [multiscaletest](#)

Functions

- [TEST_F \(MinEnclosingTriangleFinderTest, TestNoPoints\)](#)
- [TEST_F \(MinEnclosingTriangleFinderTest, TestNegativeCoordinates\)](#)
- [TEST_F \(MinEnclosingTriangleFinderTest, TestVaryingNumberOfPoints\)](#)
- [TEST_F \(MinEnclosingTriangleFinderTest, TestRandomPoints\)](#)
- int [main \(int argc, char **argv\)](#)

8.97.1 Function Documentation

8.97.1.1 int main (int argc, char ** argv)

Definition at line 334 of file MinEnclosingTriangleFinderTest.cpp.

8.97.1.2 TEST_F (MinEnclosingTriangleFinderTest , TestNoPoints)

Definition at line 310 of file MinEnclosingTriangleFinderTest.cpp.

8.97.1.3 TEST_F (MinEnclosingTriangleFinderTest , TestNegativeCoordinates)

Definition at line 314 of file MinEnclosingTriangleFinderTest.cpp.

8.97.1.4 TEST_F (MinEnclosingTriangleFinderTest , TestVaryingNumberOfPoints)

Definition at line 320 of file MinEnclosingTriangleFinderTest.cpp.

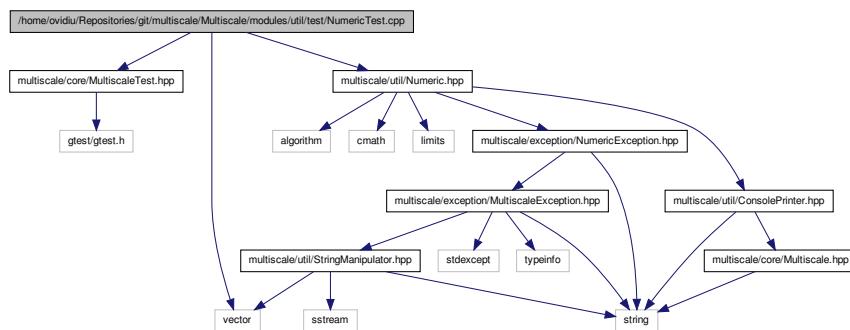
8.97.1.5 TEST_F (MinEnclosingTriangleFinderTest , TestRandomPoints)

Definition at line 328 of file MinEnclosingTriangleFinderTest.cpp.

8.98 /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 de-
```

pendency 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

8.98.1 Function Documentation

8.98.1.1 `int main(int argc, char ** argv)`

Definition at line 247 of file NumericTest.cpp.

8.98.1.2 `TEST(Numeric , GreaterOrEqual)`

Definition at line 14 of file NumericTest.cpp.

8.98.1.3 `TEST(Numeric , LessOrEqual)`

Definition at line 26 of file NumericTest.cpp.

8.98.1.4 `TEST(Numeric , AlmostEqual)`

Definition at line 38 of file NumericTest.cpp.

8.98.1.5 `TEST(Numeric , Average)`

Definition at line 51 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.6 `TEST(Numeric , Combinations)`

Definition at line 60 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.7 `TEST(Numeric , Covariance)`

Definition at line 67 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.8 `TEST(Numeric , Factorial)`

Definition at line 76 of file NumericTest.cpp.

8.98.1.9 `TEST(Numeric , GeometricMean)`

Definition at line 83 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.10 TEST(Numeric , HarmonicMean)

Definition at line 91 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.11 TEST(Numeric , Kurtosis)

Definition at line 99 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.12 TEST(Numeric , Maximum)

Definition at line 107 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.13 TEST(Numeric , Median)

Definition at line 117 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.14 TEST(Numeric , Minimum)

Definition at line 128 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.15 TEST(Numeric , Mode)

Definition at line 139 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.16 TEST(Numeric , Percentile)

Definition at line 152 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.17 TEST(Numeric , Product)

Definition at line 165 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.18 TEST(Numeric , Quartile)

Definition at line 181 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.19 TEST(Numeric , Skew)

Definition at line 196 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.20 TEST(Numeric , StandardDeviation)

Definition at line 208 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.21 TEST(Numeric , Sum)

Definition at line 220 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.1.22 TEST(Numeric , Variance)

Definition at line 233 of file NumericTest.cpp.

References DOUBLE_COMP_ERROR.

8.98.2 Variable Documentation**8.98.2.1 const double DOUBLE_COMP_ERROR = 1E-6**

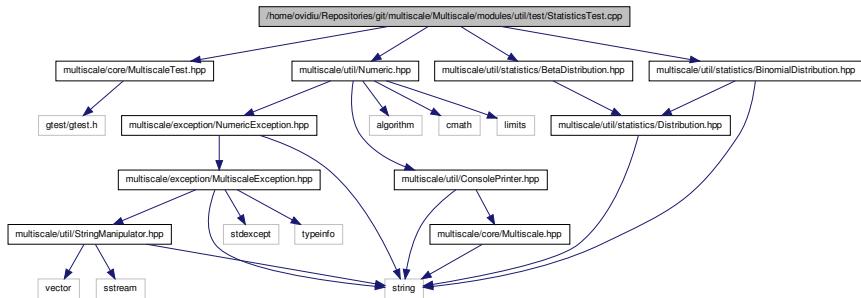
Definition at line 10 of file NumericTest.cpp.

**8.99 /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/-
```

**8.99 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/util/test/-
StatisticsTest.cpp File**

Reference **1089**
BinomialDistribution.hpp" Include dependency graph for StatisticsTest.cpp:



Functions

- [TEST](#) (Statistics, BinomialPDF)
- [TEST](#) (Statistics, BinomialCDF)
- [TEST](#) (Statistics, BetaCDF)

8.99.1 Function Documentation

8.99.1.1 TEST (Statistics , BinomialPDF)

Definition at line 11 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BinomialDistribution::pdf().

8.99.1.2 TEST (Statistics , BinomialCDF)

Definition at line 21 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BinomialDistribution::cdf().

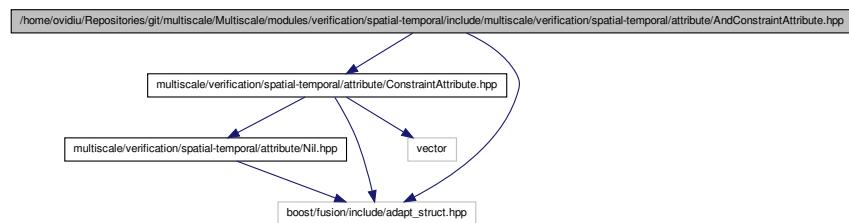
8.99.1.3 TEST (Statistics , BetaCDF)

Definition at line 30 of file StatisticsTest.cpp.

References multiscale::Numeric::almostEqual(), and multiscale::BetaDistribution::cdf().

8.100 /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" #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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

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

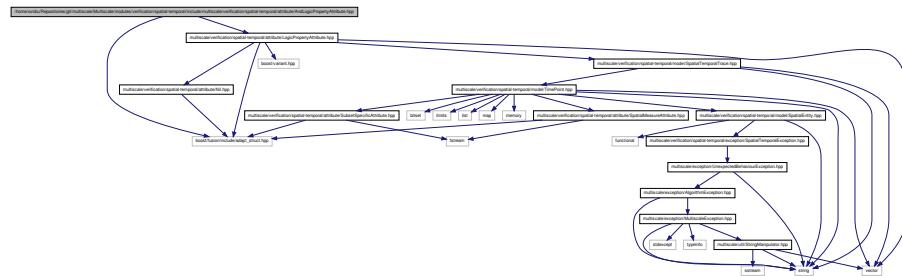
```
#include "multiscale/verification/spatial-temporal/attribute-  
LogicPropertyAttribute.hpp" #include <boost/fusion/include/adapt-
```

8.102

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Binary-NumericFilterAttribute.hpp Pre-
Name: /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Binary-NumericFilterAttribute.hpp dependency graph for AndLogicPropertyAttribute.hpp:

Reference

1091



Classes

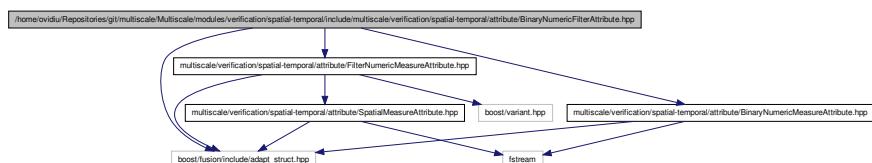
- class `multiscale::verification::AndLogicPropertyAttribute`
Class for representing an "and" logic property attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.102 /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-tempora  
FilterNumericMeasureAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for BinaryNumericFilterAttribute.hpp:
```



Classes

- class multiscale::verification::BinaryNumericFilterAttribute

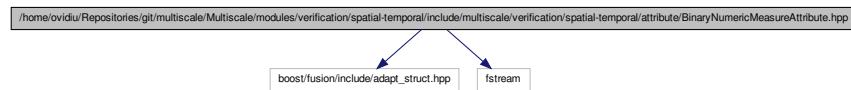
Class for representing a binary numeric filter attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.103 /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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::BinaryNumericMeasureType](#) { [multiscale::verification::Add](#) = 1, [multiscale::verification::Div](#) = 2, [multiscale::verification::Log](#) = 3, [multiscale::verification::Mod](#) = 4, [multiscale::verification::Multiply](#) = 5, [multiscale::verification::Power](#) = 6, [multiscale::verification::Subtract](#) = 7 }

Enumeration for representing a binary numeric measure type.

8.104

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

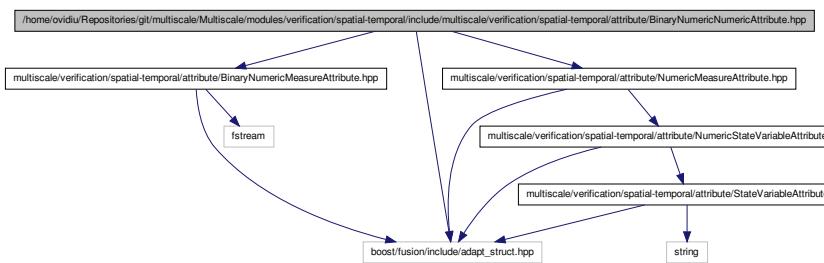
1093

- std::ostream & [multiscale::verification::operator<<\(std::ostream &out, const BinaryNumericMeasureType &binaryNumericMeasureType\)](#)

Overload the output stream operator for the enumeration.

8.104 /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-tempora  
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.

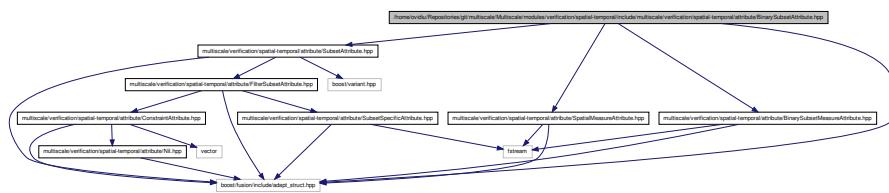
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.105 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/BinarySubsetAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-
```

```
BinarySubsetMeasureAttribute.hpp #include "multiscale/verification/spatial-
SubsetAttribute.hpp" #include "multiscale/verification/spatial-temporal/att-
SpatialMeasureAttribute.hpp" #include <boost/fusion/include/adapt-
_struct.hpp> Include dependency graph for BinarySubsetAttribute.hpp:
```



Classes

- class [multiscale::verification::BinarySubsetAttribute](#)

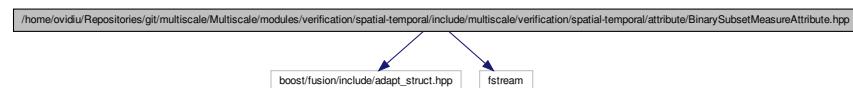
Class for representing a binary subset attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.106 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-/BinarySubsetMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp> #include
<iostream> Include dependency graph for BinarySubsetMeasureAttribute.hpp:
```



Classes

- class [multiscale::verification::BinarySubsetMeasureAttribute](#)

Class for representing a binary subset measure attribute.

8.107

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

Reference

1095

- namespace multiscale
- namespace multiscale::verification

Enumerations

- enum multiscale::verification::BinarySubsetMeasureType { multiscale::verification::Avg = 1, multiscale::verification::Geomean = 2, multiscale::verification::Harmean = 3, multiscale::verification::Kurt = 4, multiscale::verification::Max = 5, multiscale::verification::Median = 6, multiscale::verification::Min = 7, multiscale::verification::Mode = 8, multiscale::verification::Product = 9, multiscale::verification::Skew = 10, multiscale::verification::Stdev = 11, multiscale::verification::Sum = 12, multiscale::verification::Var = 13 }

Enumeration for representing a binary subset measure type.

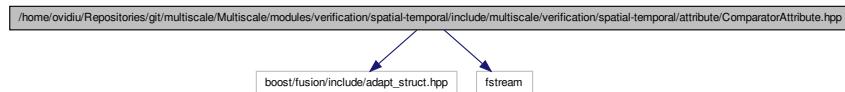
Functions

- std::ostream & multiscale::verification::operator<< (std::ostream &out, const BinarySubsetMeasureType &binarySubsetMeasureType)

Overload the output stream operator for the enumeration.

8.107 /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

- namespace multiscale
- namespace multiscale::verification

Enumerations

- enum multiscale::verification::ComparatorType { multiscale::verification::GreaterThan = 1, multiscale::verification::GreaterThanOrEqual = 2, multiscale::verification::LessThan = 3, multiscale::verification::LessThanOrEqual = 4, multiscale::verification::Equal = 5 }

Enumeration for representing a comparator type.

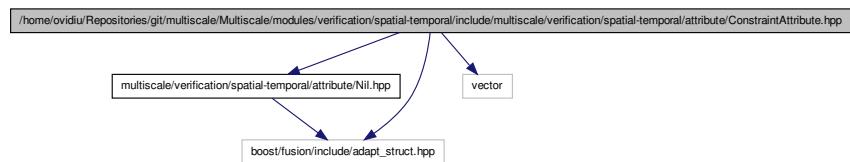
Functions

- std::ostream & multiscale::verification::operator<< (std::ostream &out, const ComparatorType &comparatorType)

Overload the output stream operator for the enumeration.

8.108 /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.

8.109

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

Reference

1097

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

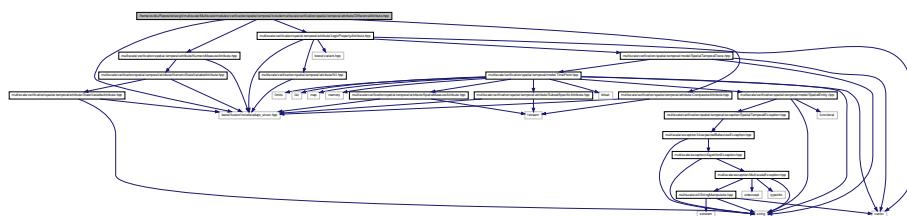
Typedefs

- `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 > > multiscale::verification::ConstraintAttributeType`

Variant for a constraint attribute type.

8.109 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-DifferenceAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribut  
LogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal/attri  
NumericMeasureAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for DifferenceAttribute.hpp:
```



Classes

- class [multiscale::verification::DifferenceAttribute](#)

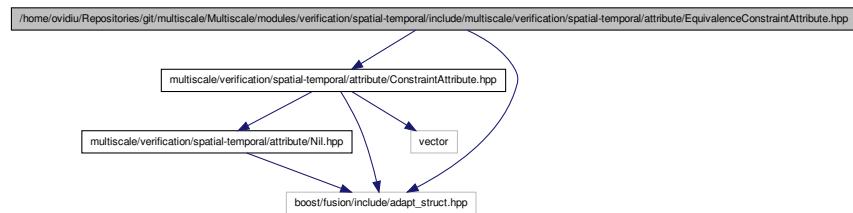
Class for representing a difference attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.110 /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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.111 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-EquivalenceLogicPropertyAttribute.hpp File Reference

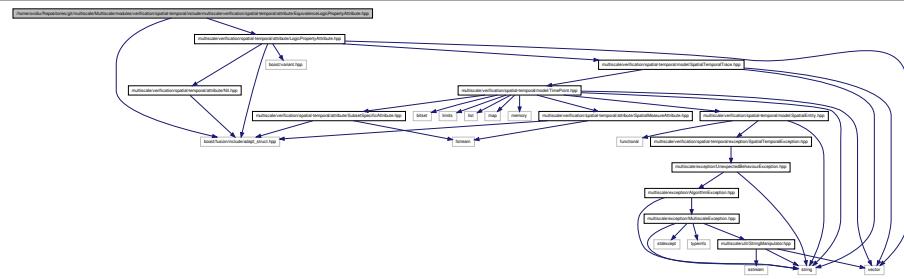
```
#include "multiscale/verification/spatial-temporal/attribute/-
LogicPropertyAttribute.hpp" #include <boost/fusion/include/adapt-
_struct.hpp> Include dependency graph for EquivalenceLogicPropertyAttribute.-
```

8.112

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

Reference

1099



Classes

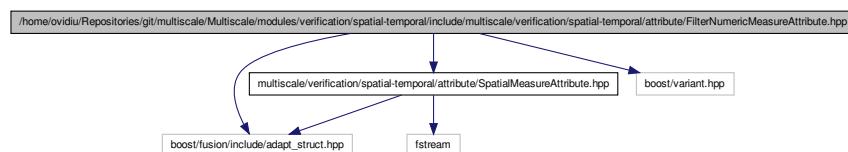
- class [multiscale::verification::EquivalenceLogicPropertyAttribute](#)
Class for representing an "equivalence" logic property attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.112 /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 LR; FilterNumericMeasureAttribute.hpp --> SpatialMeasureAttribute.hpp;
```



Classes

- class multiscale::verification::FilterNumericMeasureAttribute

Class for representing a filter numeric measure.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

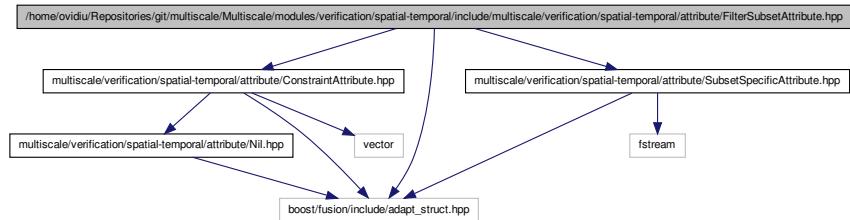
TypeDefs

- 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.

8.113 /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/  
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.

8.114

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

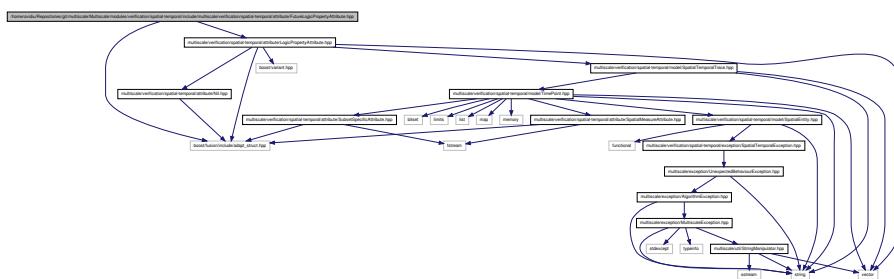
1101

- namespace [multiscale](#)

- namespace [multiscale::verification](#)

8.114 /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.

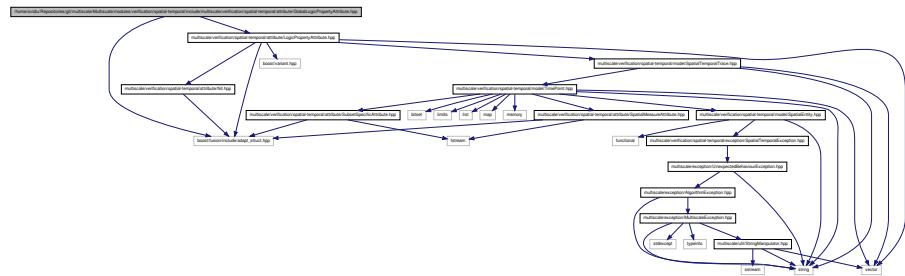
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.115 /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> Include dependency graph for GlobalLogicPropertyAttribute.hpp:



Classes

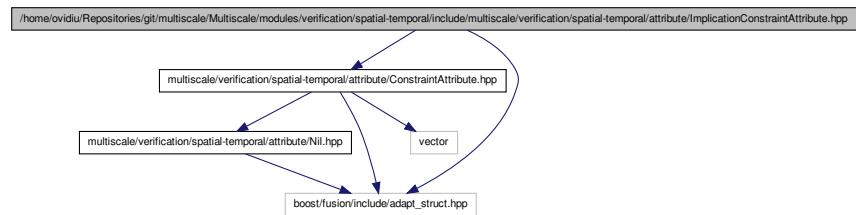
- class `multiscale::verification::GlobalLogicPropertyAttribute`
Class for representing a "globally" logic property attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.116 /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:
```



8.117

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

Reference

1103

- class `multiscale::verification::ImplicationConstraintAttribute`

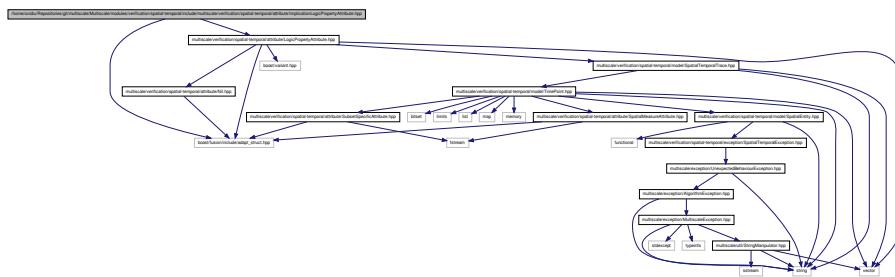
Class for representing an "implication" constraint attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.117 /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

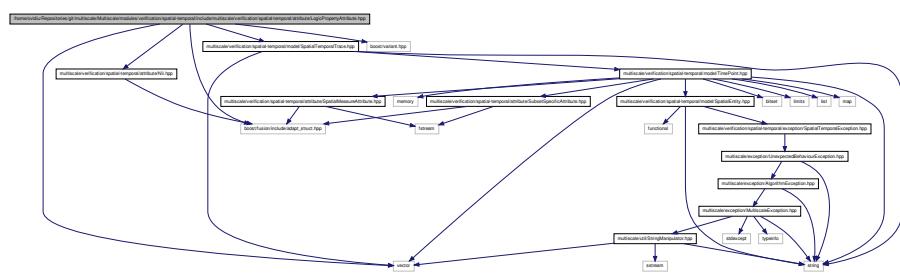
Class for representing an "implication" logic property attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.118 /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> x  
Include dependency graph for LogicPropertyAttribute.hpp:
```



Classes

- class `multiscale::verification::LogicPropertyAttribute`
Class for representing a logic property attribute.

Namespaces

- namespace multiscale
 - namespace 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`

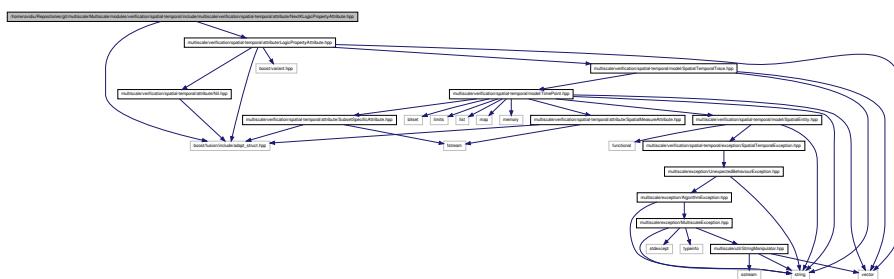
Variant for the logic property attribute:

8.119

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextKLogicPropertyAttribute.hpp File Reference
#include "multiscale/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/NextKLogicPropertyAttribute.hpp" #include <boost/fusion/include/adapt_struct.hpp> #include dependency graph for NextKLogicPropertyAttribute.hpp:

NextKLogicPropertyAttribute.hpp File Reference

```
#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.

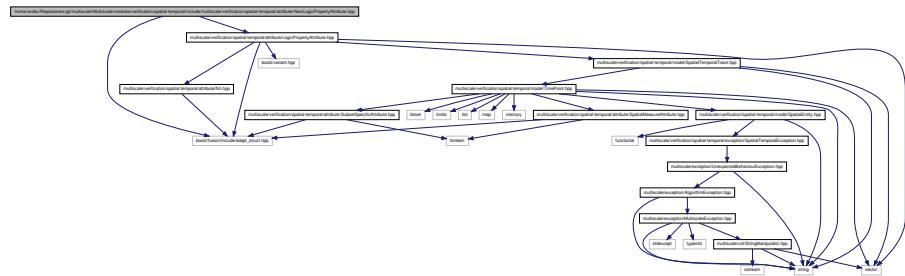
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.120 /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

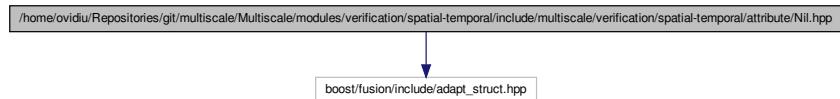
- class `multiscale::verification::NextLogicPropertyAttribute`
Class for representing a "next" logic property attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.121 /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.

8.122

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

Reference

1107

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Functions

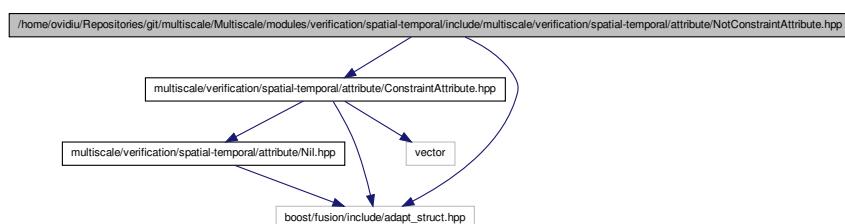
- [BOOST_FUSION_ADAPT_STRUCT \(multiscale::verification::Nil,\)](#)

8.121.1 Function Documentation

8.121.1.1 [BOOST_FUSION_ADAPT_STRUCT \(multiscale::verification::Nil \)](#)

8.122 /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

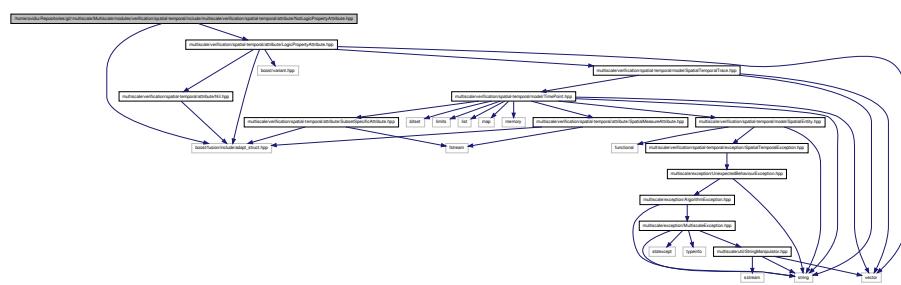
- class [multiscale::verification::NotConstraintAttribute](#)
Class for representing a "not" constraint attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.123 /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

- namespace multiscale
 - namespace multiscale::verification

8.124 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-NumericMeasureAttribute.hpp File Reference

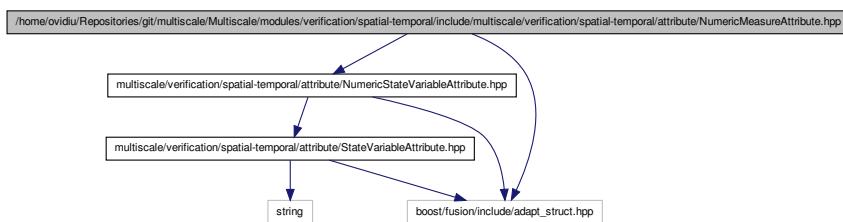
```
#include "multiscale/verification/spatial-temporal/attribute/-  
NumericStateVariableAttribute.hpp" #include <boost/fusion/include/adapt-
```

8.125

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

Reference

1109



Classes

- class [multiscale::verification::NumericMeasureAttribute](#)
Class for representing a numeric measure attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Typedefs

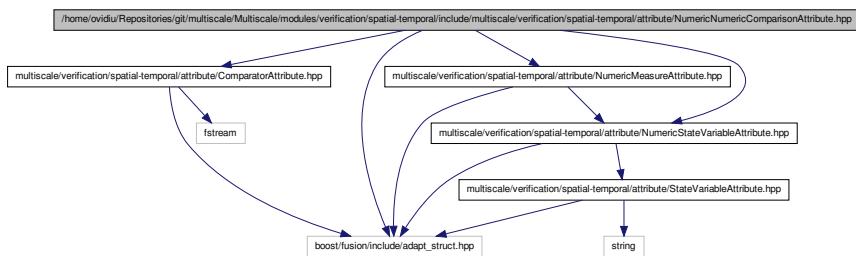
- typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive_wrapper < NumericSpatialAttribute >, boost::recursive_wrapper < PrimaryNumericMeasureAttribute >, boost::recursive_wrapper < UnaryNumericNumericAttribute >, boost::recursive_wrapper < BinaryNumericNumericAttribute >, boost::recursive_wrapper < NumericMeasureAttribute > > [multiscale::verification::NumericMeasureAttributeType](#)

Variant for the numeric measure attribute.

8.125 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericNumericComparisonAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribut  
NumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attrib  
NumericStateVariableAttribute.hpp" #include <boost/fusion/include/adapt-
```

_struct.hpp> Include dependency graph for NumericNumericComparisonAttribute.hpp:



Classes

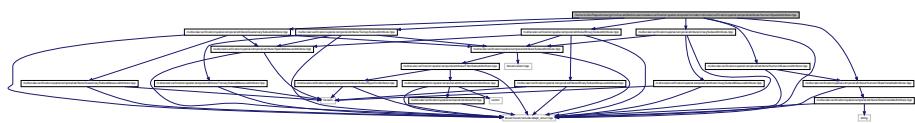
- class [multiscale::verification::NumericNumericComparisonAttribute](#)
Class for representing a numeric numeric comparison attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.126 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericSpatialAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
BinarySubsetAttribute.hpp" #include "multiscale/verification/spatial-tempor  
NumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-temp  
NumericStateVariableAttribute.hpp" #include "multiscale/verification/spatia  
QuaternarySubsetAttribute.hpp" #include "multiscale/verification/spatial-te  
TernarySubsetAttribute.hpp" #include "multiscale/verification/spatial-tempo  
UnarySubsetAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for NumericSpatialAttribute.hpp:
```



8.127

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

Reference

1111

- class [multiscale::verification::NumericSpatialAttribute](#)

Class for representing a numeric spatial attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Typedefs

- typedef boost::variant < UnarySubsetAttribute, BinarySubsetAttribute, Ternary-SubsetAttribute, QuaternarySubsetAttribute, boost::recursive_wrapper < - NumericSpatialAttribute > > [multiscale::verification::NumericSpatialAttribute-Type](#)

Variant for a numeric spatial attribute.

8.127 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- NumericSpatialNumericComparisonAttribute.hpp File - Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribut  
NumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attr  
NumericSpatialAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for NumericSpatialNumericComparison-  
Attribute.hpp:
```



Classes

- class [multiscale::verification::NumericSpatialNumericComparisonAttribute](#)

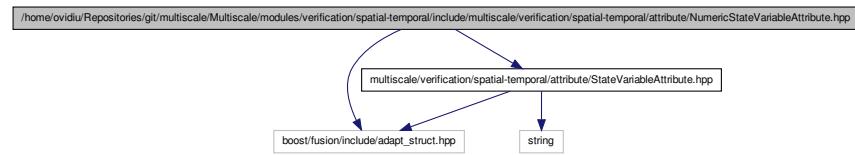
Class for representing a numeric spatial numeric comparison attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.128 /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 <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for NumericStateVariableAttribute.hpp:
```



Classes

- class [multiscale::verification::NumericStateVariableAttribute](#)
Class for representing a numeric state variable attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.129 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- OrConstraintAttribute.hpp File Reference

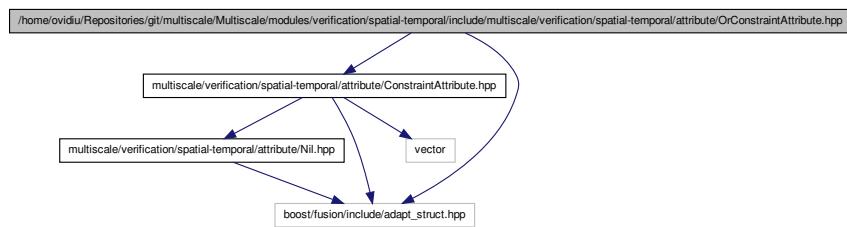
```
#include "multiscale/verification/spatial-temporal/attribute/-  
ConstraintAttribute.hpp" #include <boost/fusion/include/adapt-
```

8.130

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

Reference

1113



Classes

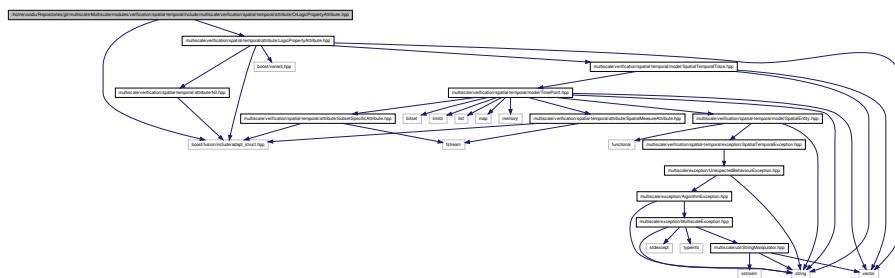
- class multiscale::verification::OrConstraintAttribute
Class for representing an "or" constraint attribute.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.130 /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](#)

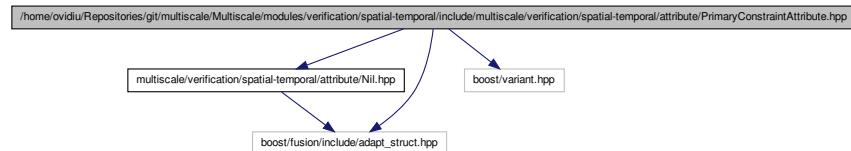
Class for representing an "or" logic property attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.131 /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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Typedefs

- typedef boost::variant< Nil, boost::recursive_wrapper < ConstraintAttribute > , boost::recursive_wrapper < NotConstraintAttribute > , boost::recursive_wrapper < UnarySpatialConstraintAttribute > , boost::recursive_wrapper <

8.132

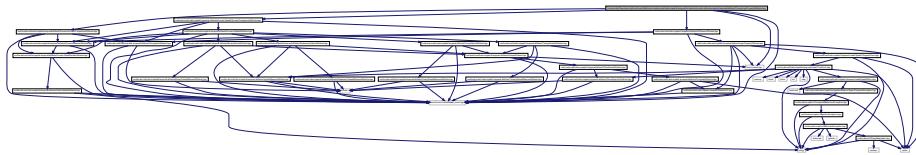
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/PrimaryLogicPropertyAttribute.hpp File Reference > > multiscale::verification::PrimaryConstraintAttributeType

1115

Variant for a primary constraint attribute.

8.132 /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/-DifferenceAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/NumericNumericComparisonAttribute.hpp" #include "multiscale/verification/spatial-temporal/NumericSpatialNumericComparisonAttribute.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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

TypeDefs

- typedef boost::variant < DifferenceAttribute, NumericSpatialNumericComparisonAttribute, NumericNumericComparisonAttribute, 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

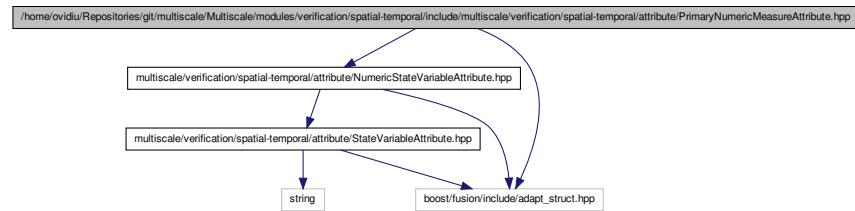
- `BOOST_FUSION_ADAPT_STRUCT (multiscale::verification::PrimaryLogicPropertyAttribute,(multiscale::verification::PrimaryLogicPropertyAttributeType, primaryLogicProperty))`

8.132.1 Function Documentation

8.132.1.1 `BOOST_FUSION_ADAPT_STRUCT (multiscale::verification::PrimaryLogicPropertyAttribute , (multiscale::verification::PrimaryLogicPropertyAttributeType, primaryLogicProperty))`

8.133 /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

- namespace `multiscale`
- namespace `multiscale::verification`

8.134

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

Reference

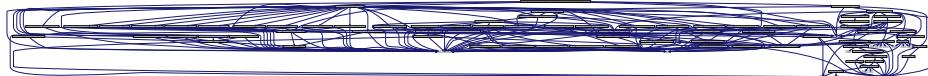
1117

- `typedef boost::variant< double, NumericStateVariableAttribute, boost::recursive_wrapper < NumericSpatialAttribute >, boost::recursive_wrapper < PrimaryNumericMeasureAttribute > > multiscale::verification::PrimaryNumericMeasureAttributeType`

Variant for the primary numeric measure attribute.

8.134 /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/attribut  
LogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal/attri  
SynthesizedAttribute.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

- namespace `multiscale`
- namespace `multiscale::verification`

Functions

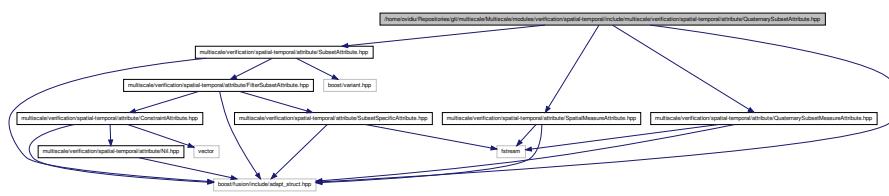
- `BOOST_FUSION_ADAPT_STRUCT (multiscale::verification::Probabilistic-
LogicPropertyAttribute,(multiscale::verification::ComparatorAttribute, compara-
tor)(double, probability)(multiscale::verification::LogicPropertyAttributeType,
logicProperty))`

8.134.1 Function Documentation

```
8.134.1.1 BOOST_FUSION_ADAPT_STRUCT ( multiscale-
    ::verification::ProbabilisticLogicPropertyAttribute ,
    (multiscale::verification::ComparatorAttribute, comparator)(double,
    probability)(multiscale::verification::LogicPropertyAttributeType,
    logicProperty) )
```

8.135 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-QuaternarySubsetAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute-QuaternarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatialSubsetAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute-SpatialMeasureAttribute.hpp" #include <boost/fusion/include/adapt_struct.hpp> Include dependency graph for QuaternarySubsetAttribute.hpp:
```



Classes

- class [multiscale::verification::QuaternarySubsetAttribute](#)
Class for representing a quaternary subset attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

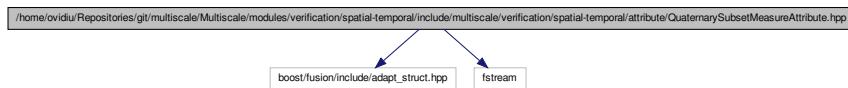
8.136 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-QuaternarySubsetMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp> #include
```

8.137

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

1119



Classes

- class [multiscale::verification::QuaternarySubsetMeasureAttribute](#)

Class for representing a quaternary subset measure attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::QuaternarySubsetMeasureType](#) { multiscale-::verification::Covar = 1 }

Enumeration for representing a quaternary subset measure type.

Functions

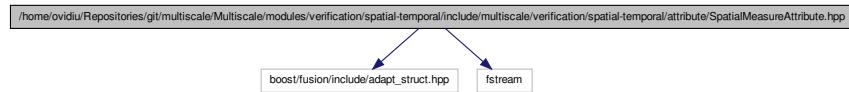
- std::ostream & [multiscale::verification::operator<<](#) (std::ostream &out, const [QuaternarySubsetMeasureType](#) &quaternarySubsetMeasureType)

Overload the output stream operator for the enumeration.

8.137 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/- SpatialMeasureAttribute.hpp File Reference

```
#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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::SpatialMeasureType](#) { [multiscale::verification::Clusteredness](#) = 1, [multiscale::verification::Density](#) = 2, [multiscale::verification::Area](#) = 3, [multiscale::verification::Perimeter](#) = 4, [multiscale::verification::DistanceFromOrigin](#) = 5, [multiscale::verification::Angle](#) = 6, [multiscale::verification::TriangleMeasure](#) = 7, [multiscale::verification::RectangleMeasure](#) = 8, [multiscale::verification::CircleMeasure](#) = 9, [multiscale::verification::CentroidX](#) = 10, [multiscale::verification::CentroidY](#) = 11 }
- Enumeration for representing the types of spatial measures.*

Functions

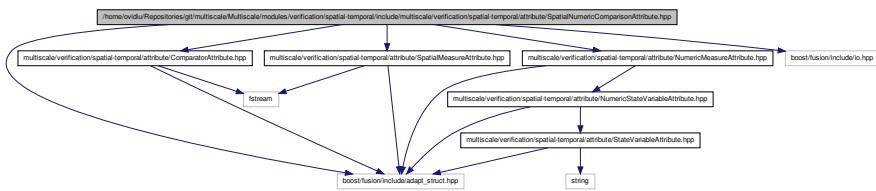
- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const - SpatialMeasureType &spatialMeasureType)`
Overload the output stream operator for the enumeration.

8.138 /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-temp-
```

8.139

```
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/StateVariable-Attribute.hpp File Reference #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](#)

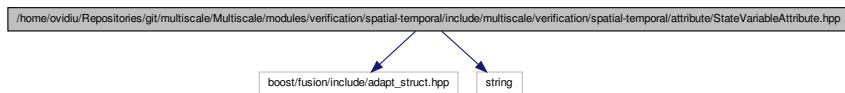
Class for representing a spatial numeric comparison attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.139 /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](#)

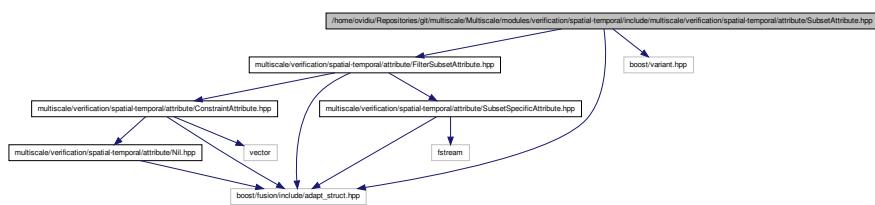
Class for representing a state variable attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.140 /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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Typedefs

- typedef `boost::variant < SubsetSpecificAttribute, FilterSubsetAttribute, boost::recursive_wrapper < SubsetSubsetOperationAttribute >, boost::recursive_wrapper < SubsetAttribute > >` [multiscale::verification::SubsetAttributeType](#)

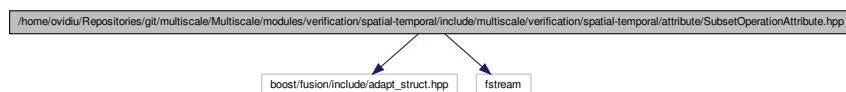
Variant for a subset attribute.

8.141

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

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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::SubsetOperationType](#) { [multiscale::verification::Difference](#) = 1, [multiscale::verification::Intersection](#) = 2, [multiscale::verification::Union](#) = 3 }

Enumeration for representing the types of subset operations.

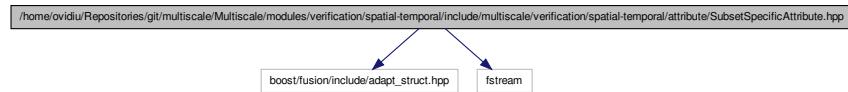
Functions

- std::ostream & [multiscale::verification::operator<<](#) (std::ostream &out, const [SubsetOperationType](#) &subsetOperationType)
Overload the output stream operator for the enumeration.

8.142 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp File Reference

```
#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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)
- namespace [multiscale::verification::subsetspecific](#)

Enumerations

- enum [multiscale::verification::SubsetSpecificType](#) { [multiscale::verification::Clusters](#) = 0, [multiscale::verification::Regions](#), [multiscale::verification::NrOfSubsetSpecificTypeEntries](#) }
Enumeration for representing a specific subset type.

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.

8.143

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/SubsetSubsetVariables.h

SubsetSubsetOperationAttribute.hpp File

Reference

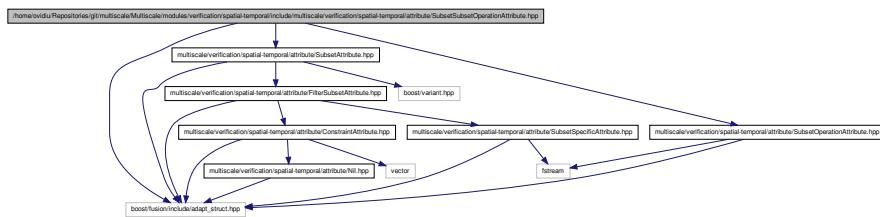
1125

- static const std::size_t [multiscale::verification::NR_SUBSET_SPECIFIC_TYPE_ENTRIES](#) = static_cast<std::size_t>(SubsetSpecificType::NrOfSubsetSpecificTypeEntries)

An std::size_t constant which stores the number of subset specific type entries.

8.143 /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.

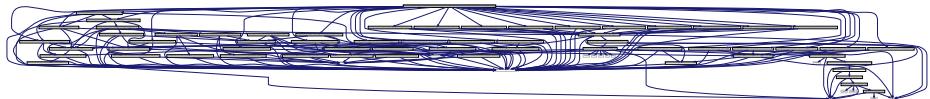
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.144 /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/-  
PrimaryLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-tempora
```

```
DifferenceAttribute.hpp" #include "multiscale/verification/spatial-temporal
NumericNumericComparisonAttribute.hpp" #include "multiscale/verification/spatial-temporal
NumericSpatialNumericComparisonAttribute.hpp"      #include
"multiscale/verification/spatial-temporal/attribute/Not-
LogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
FutureLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
GlobalLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
NextLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
NextKLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
OrLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
AndLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
ImplicationLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
EquivalenceLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
UntilLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal
PrimaryNumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal
UnaryNumericNumericAttribute.hpp" #include "multiscale/verification/spatial-temporal
BinaryNumericNumericAttribute.hpp" #include "multiscale/verification/spatial-temporal
PrimaryConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
NotConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
OrConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
AndConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
ImplicationConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
EquivalenceConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
UnarySpatialConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
UnaryTypeConstraintAttribute.hpp" #include "multiscale/verification/spatial-temporal
SubsetSubsetOperationAttribute.hpp" #include "multiscale/verification/spatial-temporal
UnaryNumericFilterAttribute.hpp" #include "multiscale/verification/spatial-temporal
BinaryNumericFilterAttribute.hpp" Include dependency graph for -
SynthesizedAttribute.hpp:
```



8.145 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TernarySubsetAttribute.hpp File Reference

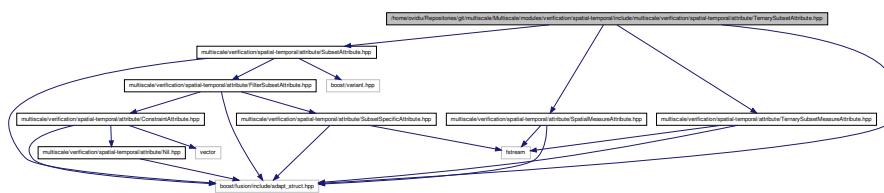
```
#include "multiscale/verification/spatial-temporal/attribute/-
TernarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal
SubsetAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/-
SpatialMeasureAttribute.hpp" #include <boost/fusion/include/adapt-
```

8.146

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/TernarySubsetAttribute.hpp File Reference dependency graph for TernarySubsetAttribute.hpp:

SubsetMeasureAttribute.hpp

1127



Classes

- class [multiscale::verification::TernarySubsetAttribute](#)

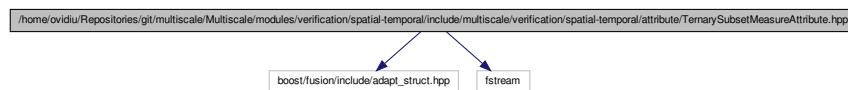
Class for representing a ternary subset attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.146 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-TernarySubsetMeasureAttribute.hpp File Reference

#include <boost/fusion/include/adapt_struct.hpp> #include
<fstream> Include dependency graph for TernarySubsetMeasureAttribute.hpp:



Classes

- class [multiscale::verification::TernarySubsetMeasureAttribute](#)

Class for representing a ternary subset measure attribute.

Namespaces

- namespace `multiscale`
- namespace `multiscale::verification`

Enumerations

- enum `multiscale::verification::TernarySubsetMeasureType` { `multiscale::verification::Percentile` = 1, `multiscale::verification::Quartile` = 2 }

Enumeration for representing a ternary subset measure type.

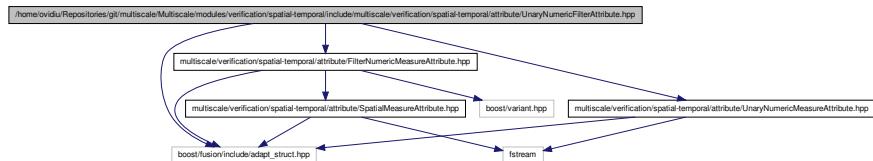
Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const TernarySubsetMeasureType &ternarySubsetMeasureType)`

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-/UnaryNumericFilterAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
FilterNumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-  
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

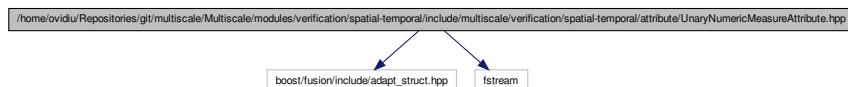
- namespace `multiscale`
- namespace `multiscale::verification`

8.148

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/Unary-NumericMeasureAttribute.hpp File Reference

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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::UnaryNumericMeasureType](#) { [multiscale::verification::Abs](#) = 1, [multiscale::verification::Ceil](#) = 2, [multiscale::verification::Floor](#) = 3, [multiscale::verification::Round](#) = 4, [multiscale::verification::Sign](#) = 5, [multiscale::verification::Sqrt](#) = 6, [multiscale::verification::Trunc](#) = 7 }

Enumeration for representing a unary numeric measure type.

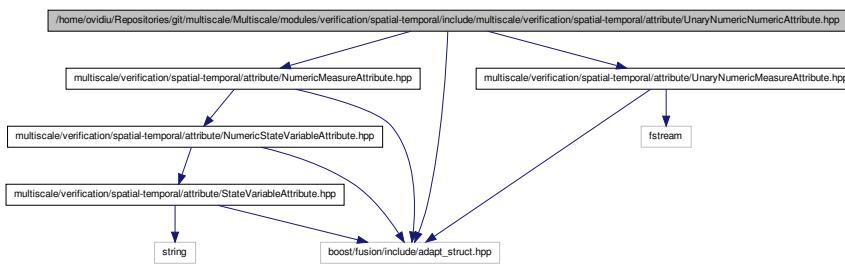
Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const UnaryNumericMeasureType &unaryNumericMeasureType)`

Overload the output stream operator for the enumeration.

8.149 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-/UnaryNumericNumericAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
NumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-tempo-  
laryNumericMeasureAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for UnaryNumericNumericAttribute.hpp:
```



Classes

- class [multiscale::verification::UnaryNumericNumericAttribute](#)

Class for representing a unary numeric numeric measure attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

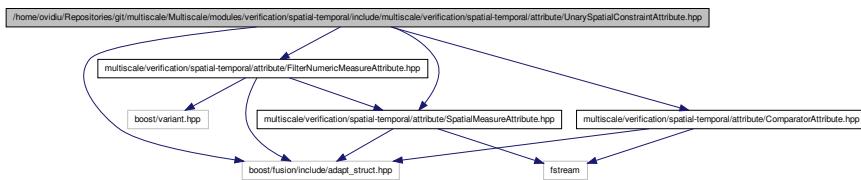
8.150 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-/UnarySpatialConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/  
FilterNumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-  
SpatialMeasureAttribute.hpp" #include <boost/fusion/include/adapt-
```

8.151

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

1131



Classes

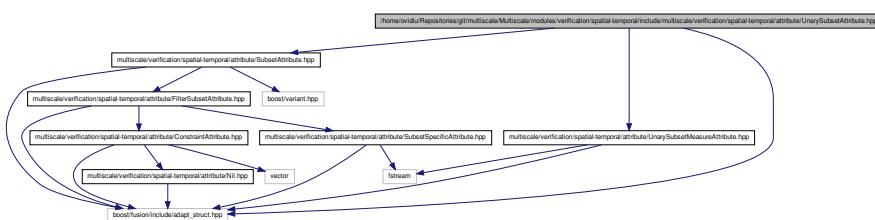
- class [multiscale::verification::UnarySpatialConstraintAttribute](#)
Class for representing a "unary" spatial constraint attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.151 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute/-UnarySubsetAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
UnarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/  
SubsetAttribute.hpp" #include <boost/fusion/include/adapt-  
_struct.hpp> Include dependency graph for UnarySubsetAttribute.hpp:
```



Classes

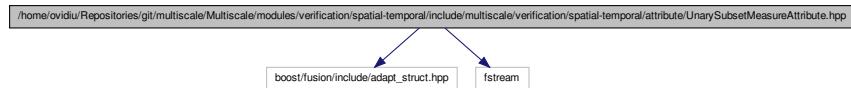
- class [multiscale::verification::UnarySubsetAttribute](#)
Class for representing a unary subset attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.152 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/attribute-UnarySubsetMeasureAttribute.hpp File Reference

```
#include <boost/fusion/include/adapt_struct.hpp> #include
<fstream> Include dependency graph for UnarySubsetMeasureAttribute.hpp:
```



Classes

- class [multiscale::verification::UnarySubsetMeasureAttribute](#)
Class for representing a unary subset measure attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::UnarySubsetMeasureType](#) { [multiscale::verification::Count](#) = 1, [multiscale::verification::Clusteredness](#) = 1, [multiscale::verification::Density](#) = 2 }
Enumeration for representing a unary subset measure type.

Functions

- `std::ostream & multiscale::verification::operator<< (std::ostream &out, const UnarySubsetMeasureType &unarySubsetMeasureType)`
Overload the output stream operator for the enumeration.

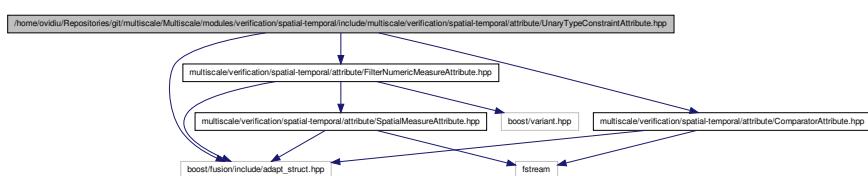
8.153

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

Reference [1133](#)

UnaryTypeConstraintAttribute.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribut  
FilterNumericMeasureAttribute.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.

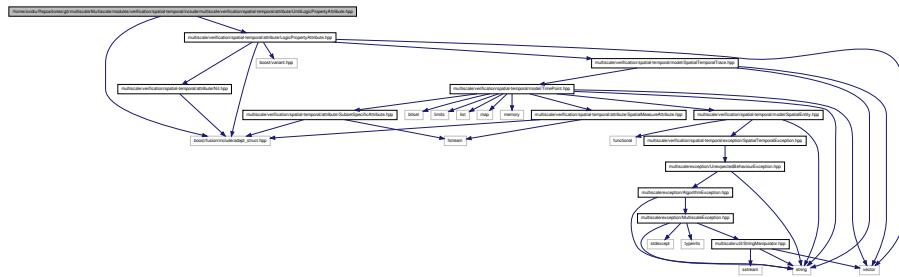
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.154 /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> Include dependency graph for UntilLogicPropertyAttribute.hpp:



Classes

- class [multiscale::verification::UntilLogicPropertyAttribute](#)

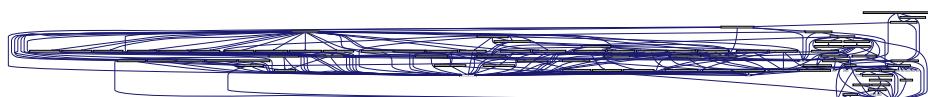
Class for representing an "until" logic property attribute.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.155 /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.

8.156

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ApproximateBayesianModelCheckerFactory.hpp File

Reference

1135

- namespace multiscale
- namespace multiscale::verification

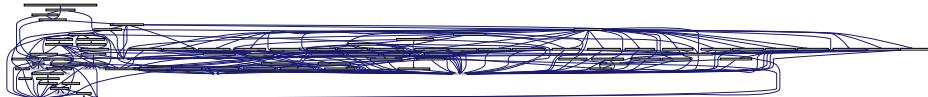
Enumerations

- enum multiscale::verification::ApproximateBayesianModelCheckingResult {
 multiscale::verification::TRUE, multiscale::verification::FALSE, multiscale-
 ::verification::MORE_TRACES_REQUIRED }

Enumeration for representing the model checking result.

8.156 /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 Approximate-  
BayesianModelCheckerFactory.hpp:
```



Classes

- class multiscale::verification::ApproximateBayesianModelCheckerFactory
Class for creating ApproximateBayesianModelChecker instances.

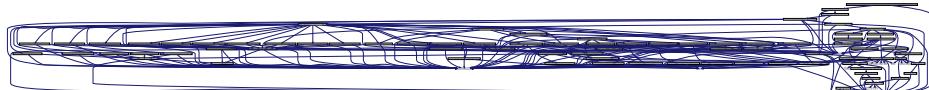
Namespaces

- namespace multiscale
- namespace multiscale::verification

8.157 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/- ApproximateProbabilisticModelChecker.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/-
```

`ModelChecker.hpp" #include <string> Include dependency graph for ApproximateProbabilisticModelChecker.hpp:`



Classes

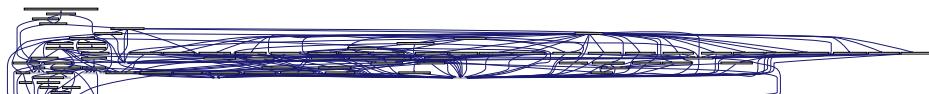
- class [multiscale::verification::ApproximateProbabilisticModelChecker](#)
Class used to run approximate probabilistic model checking tasks.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.158 /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

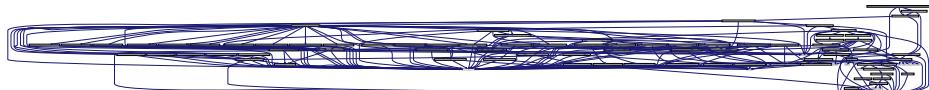
- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.159

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/BayesianModelChecker.hpp File Reference
8.159 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/-
Reference BayesianModelChecker.hpp File Reference

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

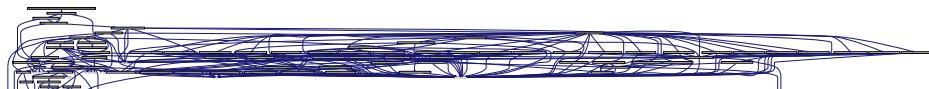
- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::BayesianModelCheckingResult](#) { [multiscale-::verification::TRUE](#), [multiscale::verification::FALSE](#), [multiscale::verification::MORE_TRACES_REQUIRED](#) }
Enumeration for representing the model checking result.

**8.160 /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 BayesianModel-  
CheckerFactory.hpp:
```



Classes

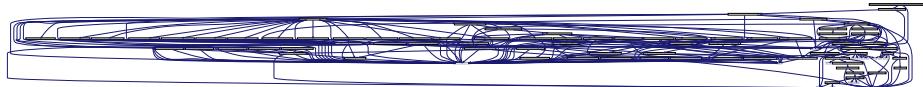
- class [multiscale::verification::BayesianModelCheckerFactory](#)
Class for creating [BayesianModelChecker](#) instances.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.161 /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/-
SpatialTemporalTrace.hpp" Include dependency graph for ModelChecker.-
hpp:
```



Classes

- class [multiscale::verification::ModelChecker](#)
Abstract class representing a generic model checker.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.162 /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/
```

8.163

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ModelCheckingManager.hpp File Reference #include <memory> Include dependency graph for ModelCheckerFactory.hpp:

1139



Classes

- class [multiscale::verification::ModelCheckerFactory](#)

Interface for different model checker factories.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.163 /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/-
AbstractSyntaxTree.hpp" #include "multiscale/verification/spatial-temporal/parsing/-
Parser.hpp" #include <chrono> #include <ctime> #include
<string> #include <thread> #include <vector> Include dependency graph for ModelCheckingManager.hpp:
```



Classes

- class [multiscale::verification::ModelCheckingManager](#)

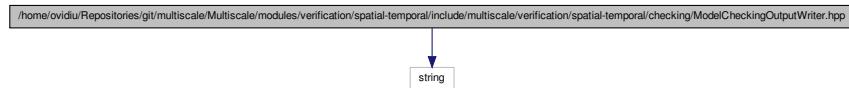
Class for managing the model checking processes.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.164 /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

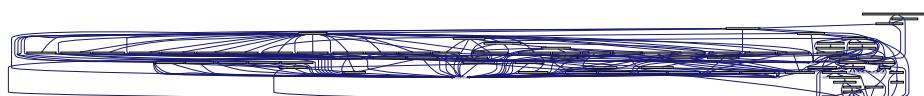
- class [multiscale::verification::ModelCheckingOutputWriter](#)
Class used to output the model checkers progress.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.165 /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 ProbabilisticBlack-  
BoxModelChecker.hpp:
```



8.166

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/checking/ProbabilisticBlackBoxModelCheckerFactory.hpp File
Reference

1141

- class [multiscale::verification::ProbabilisticBlackBoxModelChecker](#)

Class used to run probabilistic black-box model checking tasks.

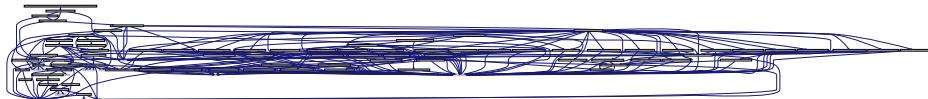
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.166 /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" Include dependency graph for ProbabilisticBlack-  
BoxModelCheckerFactory.hpp:
```



Classes

- class [multiscale::verification::ProbabilisticBlackBoxModelCheckerFactory](#)
Class for creating [ProbabilisticBlackBoxModelChecker](#) instances.

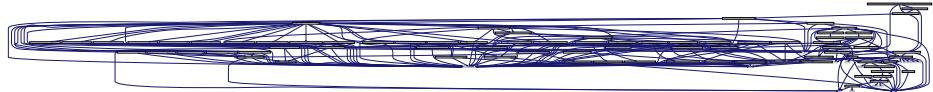
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.167 /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

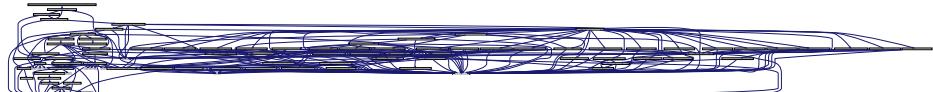
- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::StatisticalModelCheckingResult](#) { [multiscale::verification::TRUE](#), [multiscale::verification::FALSE](#), [multiscale::verification::UNDECIDED](#), [multiscale::verification::MORE_TRACES_REQUIRED](#) }
Enumeration for representing the model checking result.

8.168 /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.

8.169

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/data/LogicPropertyNamespaces.hpp File

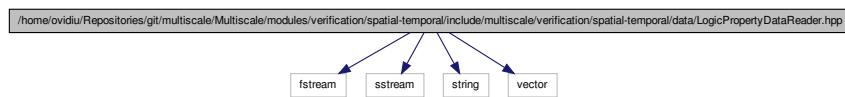
Reference

1143

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.169 /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> x
#include <vector> Include dependency graph for LogicPropertyDataReader.-hpp:
```



Classes

- class [multiscale::verification::LogicPropertyDataReader](#)

Class used to input logic properties.

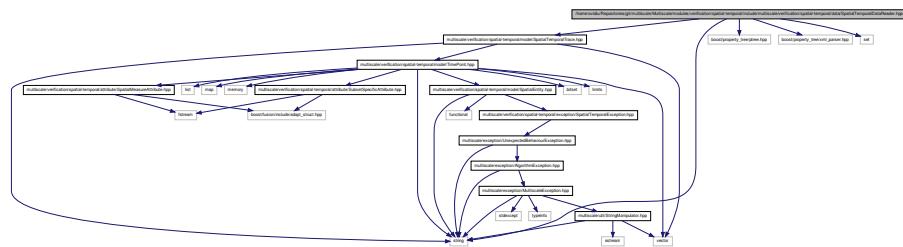
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.170 /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 SpatialTemporal-  
DataReader.hpp:
```



Classes

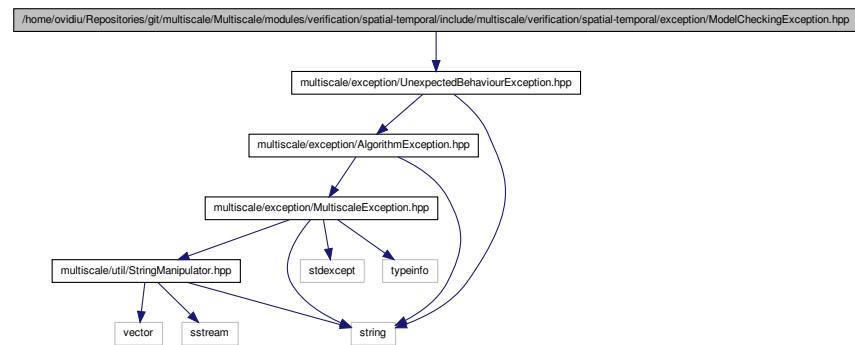
- class `multiscale::verification::SpatialTemporalDataReader`
Class for reading spatial temporal trace data from input files.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.171 /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 ModelCheckingException.hpp:
```



8.172

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ModelCheckingHelpRequestException.hpp File

Reference

1145

- class [multiscale::verification::ModelCheckingException](#)

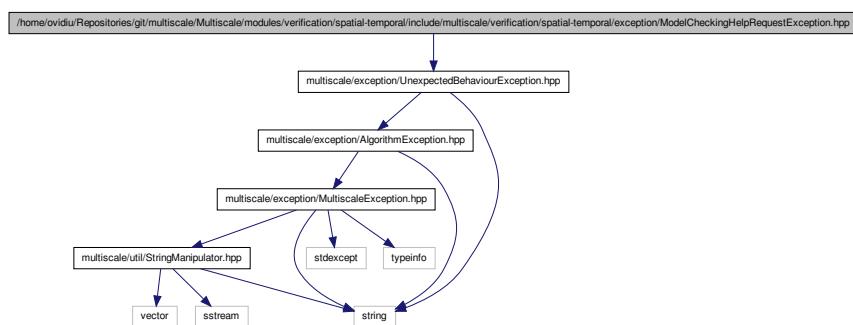
Class for representing a model checking exception.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.172 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/-ModelCheckingHelpRequestException.hpp File Reference

#include "multiscale/exception/UnexpectedBehaviourException.-
hpp" Include dependency graph for ModelCheckingHelpRequestException.hpp:



Classes

- class [multiscale::verification::ModelCheckingHelpRequestException](#)

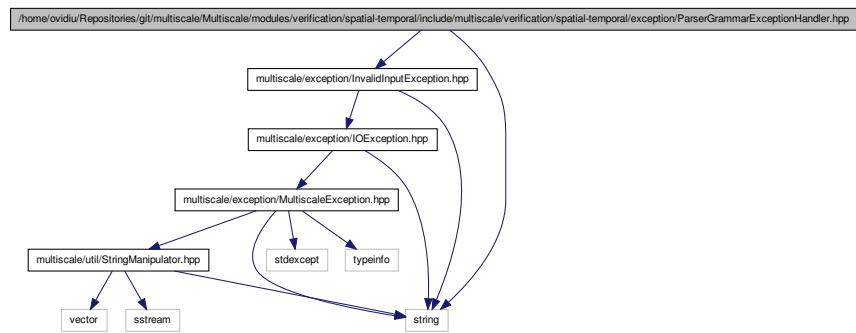
Class for representing a model checking help request exception.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.173 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception-/ParserGrammarExceptionHandler.hpp File Reference

```
#include "multiscale/exception/InvalidInputException.-  
hpp" #include <string> Include dependency graph for ParserGrammar-  
ExceptionHandler.hpp:
```



Classes

- class [multiscale::verification::ParserGrammarExceptionHandler](#)

Class for handling parser grammar exceptions.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.174 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception-/ParserGrammarExtraInputException.hpp File Reference

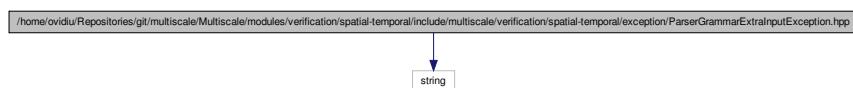
```
#include <string> Include dependency graph for ParserGrammarExtraInput-  
Exception.hpp:
```

8.175

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/ParserException.hpp
GrammarProbabilityException.hpp File

Reference

1147



Classes

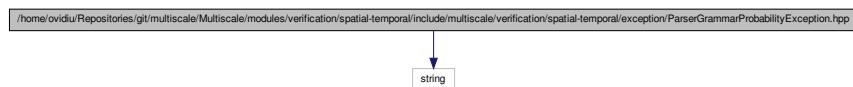
- class [multiscale::verification::ParserGrammarExtraInputException](#)
Class for representing "extra input" exceptions in the parsing process.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.175 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception-/ParserGrammarProbabilityException.hpp File Reference

#include <string> Include dependency graph for ParserGrammarProbabilityException.hpp:



Classes

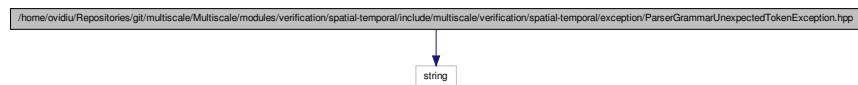
- class [multiscale::verification::ParserGrammarProbabilityException](#)
Class for representing "probability" exceptions in the parsing process.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.176 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception-/ParserGrammarUnexpectedTokenException.hpp File - Reference

```
#include <string> Include dependency graph for ParserGrammarUnexpectedTokenException.hpp:
```



Classes

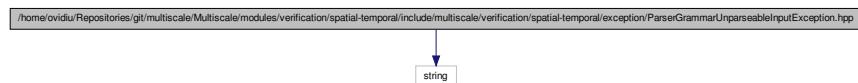
- class [multiscale::verification::ParserGrammarUnexpectedTokenException](#)
Class for representing "unexpected token" exceptions in the parsing process.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.177 /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:
```



8.178

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/exception/SpatialTemporalException.hpp File
Reference

1149

- class [multiscale::verification::ParserGrammarUnparseableInputException](#)

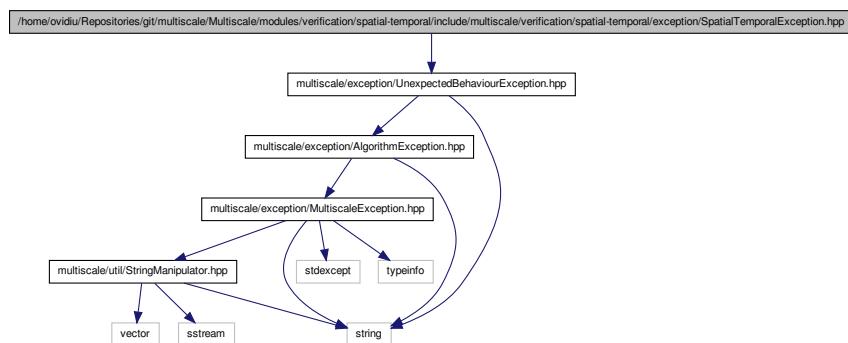
Class for representing "unparseable input" exceptions in the parsing process.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.178 /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 dependency graph for SpatialTemporalException.hpp:



Classes

- class [multiscale::verification::SpatialTemporalException](#)

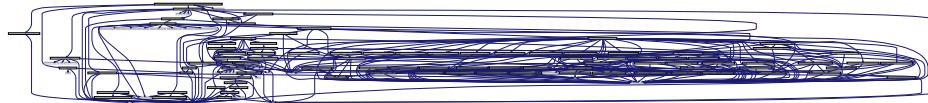
Class for representing a spatial temporal exception.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.179 /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/chModelCheckingManager.hpp" #include "multiscale/verification/spatial-temporaModelCheckingOutputWriter.hpp" #include <boost/program_-options.hpp> Include dependency graph for CommandLineModelChecking.hpp:
```



Classes

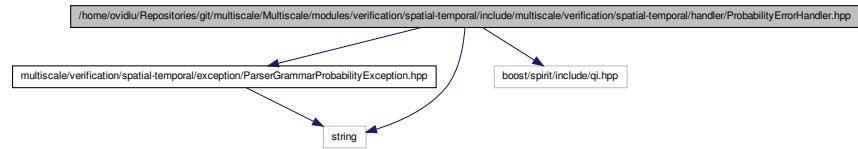
- class [multiscale::verification::CommandLineModelChecking](#)
Class for running model checkers from the command line.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.180 /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:
```



8.181

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/UnexpectedTokenErrorHandler.hpp File

Reference

1151

- struct [multiscale::verification::ProbabilityErrorHandler](#)

Structure for defining the error handler for invalid probability errors.

- struct [multiscale::verification::ProbabilityErrorHandler::result< typename, typename, typename >](#)

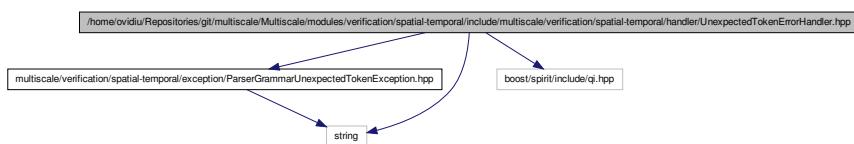
Structure for specifying the type of the result.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.181 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/handler/-UnexpectedTokenErrorHandler.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/-  
ParserGrammarUnexpectedTokenException.hpp" #include <boost/spirit/include/qi.-  
hpp> #include <string> Include dependency graph for UnexpectedToken-  
ErrorHandler.hpp:
```



Classes

- struct [multiscale::verification::UnexpectedTokenErrorHandler](#)

Structure for defining the error handler for unexpected token errors.
- struct [multiscale::verification::UnexpectedTokenErrorHandler::result< typename, typename, typename >](#)

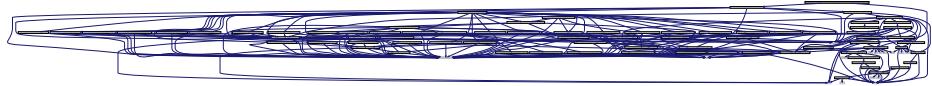
Structure for specifying the type of the result.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.182 /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/  
SpatialTemporalTrace.hpp" Include dependency graph for AbstractSyntax-  
Tree.hpp:
```



Classes

- class [multiscale::verification::AbstractSyntaxTree](#)

Class used for representing an abstract syntax tree.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

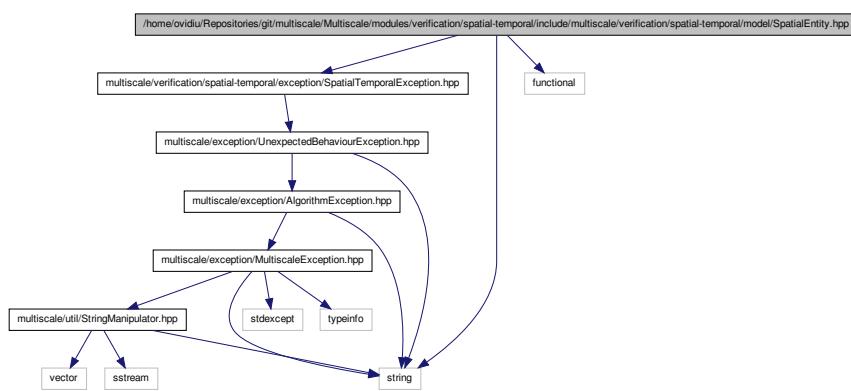
8.183 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model-/SpatialEntity.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/-  
SpatialTemporalException.hpp"      #include <functional> x
```

8.184

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/model/SpatialTemporalTrace.hpp File Reference

1153



Classes

- class [multiscale::verification::SpatialEntity](#)

Class for representing a pseudo-3D spatial entity.

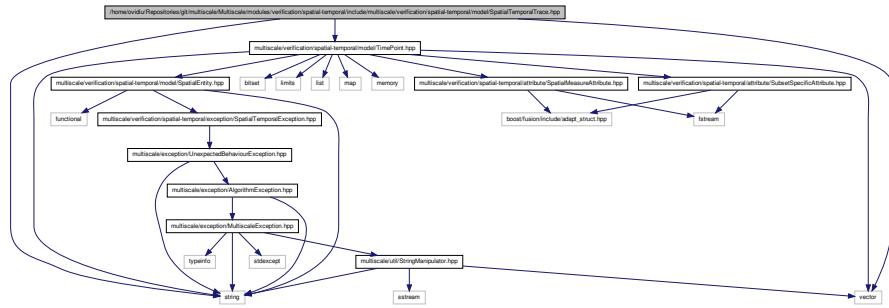
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.184 /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

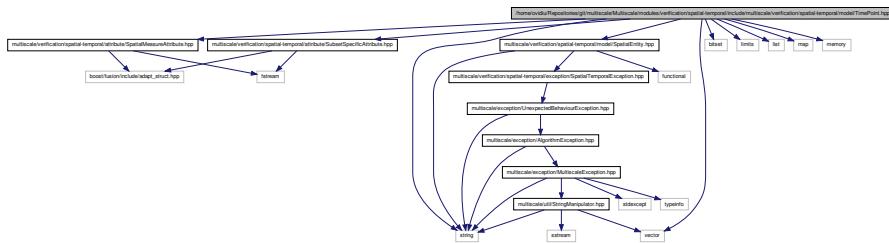
- class multiscale::verification::SpatialTemporalTrace
Class for representing a spatial temporal trace.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.185 /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-temp-  
SubsetSpecificAttribute.hpp" #include "multiscale/verification/spatial-temp-  
SpatialEntity.hpp" #include <bitset> #include <limits> ×  
#include <list> #include <map> #include <memory> #include  
<string> #include <vector> Include dependency graph for TimePoint.hpp:
```



8.186

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/Parser.hpp File
Classes Reference 1155

- class [multiscale::verification::TimePoint](#)

Class for representing a timepoint.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Enumerations

- enum [multiscale::verification::SetOperationType](#) { [multiscale::verification::Difference](#) = 1, [multiscale::verification::Intersection](#) = 2, [multiscale::verification::Union](#) = 3 }

Enumeration for representing the set operation type(s)

8.186 /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/-ParserGrammar.hpp" #include <string> Include dependency graph for Parser.hpp:
```



Classes

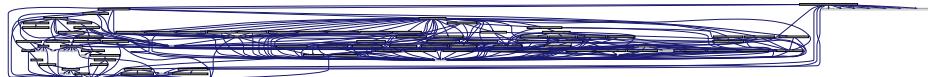
- class [multiscale::verification::Parser](#)
Class used for parsing (P)BLSTL logical queries.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.187 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing-/ParserGrammar.hpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
ProbabilisticLogicPropertyAttribute.hpp" #include "multiscale/verification/  
ProbabilityErrorHandler.hpp" #include "multiscale/verification/spatial-tempo-  
UnexpectedTokenErrorHandler.hpp" #include "multiscale/verification/spatial-  
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/pho-  
_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> x  
#include <string> Include dependency graph for ParserGrammar.hpp:
```



Classes

- class [multiscale::verification::ParserGrammar< Iterator >](#)
The grammar for parsing (P)BLSTL spatial-temporal logical queries.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

Variables

- phoenix::function < UnexpectedTokenErrorHandler > const [multiscale-
::verification::handleUnexpectedTokenError](#) = UnexpectedTokenErrorHandler()
- phoenix::function < ProbabilityErrorHandler > const [multiscale::verification-
::handleProbabilityError](#) = ProbabilityErrorHandler()

8.188 /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/-
```

8.188

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/parsing/SymbolTables.hpp File Reference

BinarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/BinarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/1157
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/QuaternarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/QuaternarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/SpatialMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/SubsetSpecificAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/SubsetOperationAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/TernarySubsetMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/UnaryNumericMeasureAttribute.hpp" #include "multiscale/verification/spatial-temporal/attribute/UnarySubsetMeasureAttribute.hpp" #include <boost/spirit/include/qi-symbols.hpp> **Include dependency graph for SymbolTables.hpp:**



Classes

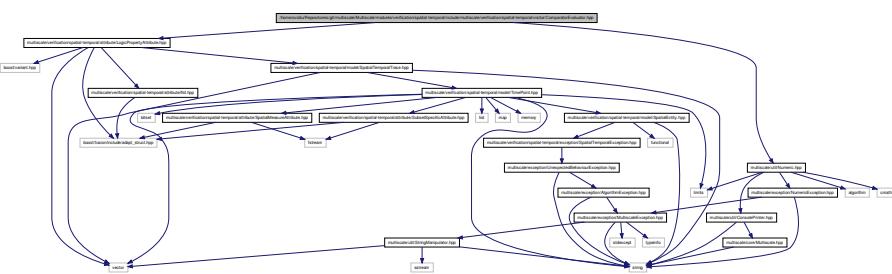
- struct [multiscale::verification::ComparatorTypeParser](#)
Symbol table and parser for the comparator type.
- struct [multiscale::verification::ComparatorNonEqualTypeParser](#)
Symbol table and parser for the comparator type which does not accept the "=" symbol.
- struct [multiscale::verification::SpatialMeasureTypeParser](#)
Symbol table and parser for the spatial measure type.
- struct [multiscale::verification::SubsetSpecificTypeParser](#)
Symbol table and parser for a specific subset type.
- struct [multiscale::verification::SubsetOperationTypeParser](#)
Symbol table and parser for the subset operation type.
- struct [multiscale::verification::BinaryNumericMeasureTypeParser](#)
Symbol table and parser for the binary numeric measure type.
- struct [multiscale::verification::UnaryNumericMeasureTypeParser](#)
Symbol table and parser for the unary numeric measure type.
- struct [multiscale::verification::QuaternarySubsetMeasureTypeParser](#)
Symbol table and parser for the quaternary subset measure type.
- struct [multiscale::verification::TernarySubsetMeasureTypeParser](#)
Symbol table and parser for the ternary subset measure type.
- struct [multiscale::verification::BinarySubsetMeasureTypeParser](#)
Symbol table and parser for the binary subset measure type.
- struct [multiscale::verification::UnarySubsetMeasureTypeParser](#)
Symbol table and parser for the unary subset measure type.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.189 /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/sp  
LogicPropertyAttribute.hpp" Include dependency graph for Comparator-  
Evaluator.hpp:
```



Classes

- class `multiscale::verification::ComparatorEvaluator`

Class for evaluating comparison expressions.

Namespaces

- namespace multiscale
 - namespace multiscale::verification

8.190 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor-
ConstraintEvaluator.hpp File Reference

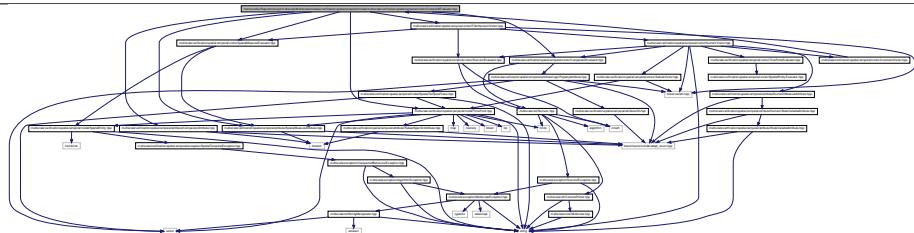
```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" #include "multiscale/verification/spatial-temporal/  
SpatialMeasureAttribute.hpp" #include "multiscale/verification/spatial-tempo-  
TimePoint.hpp" #include "multiscale/verification/spatial-temporal/visitor/-  
ComparatorEvaluator.hpp" #include "multiscale/verification/spatial-temporal/  
FilterNumericVisitor.hpp" #include "multiscale/verification/spatial-tempora-  
SpatialMeasureEvaluator.hpp" Include dependency graph for Constraint-
```

8.191

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/ConstraintEvaluator.hpp
Visitor.hpp File

Reference

1159



Classes

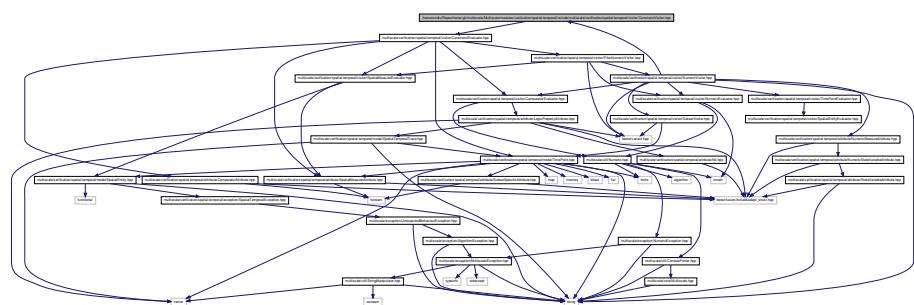
- class [multiscale::verification::ConstraintEvaluator](#)
Class for evaluating constraint expressions.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.191 /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/visitor/-  
ConstraintEvaluator.hpp" Include dependency graph for ConstraintVisitor.-  
hpp:
```



Classes

- class [multiscale::verification::ConstraintVisitor](#)

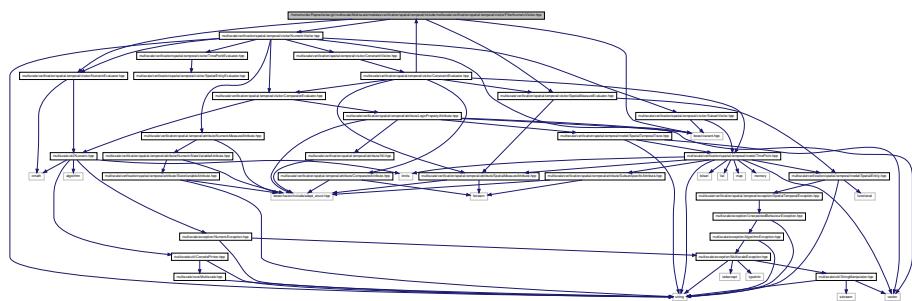
Class used to evaluate constraints.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.192 /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/visi  
NumericVisitor.hpp" #include "multiscale/verification/spatial-temporal/visi  
SpatialMeasureEvaluator.hpp"      #include <boost/variant.-  
hpp> Include dependency graph for FilterNumericVisitor.hpp:
```



Classes

- class [multiscale::verification::FilterNumericVisitor](#)

Class for evaluating filter numeric measures.

Namespaces

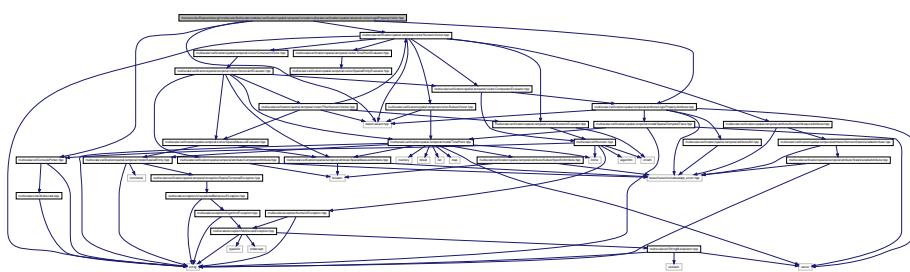
- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.193

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/LogicPropertyVisitor.hpp File Reference
8.193 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/-
[LogicPropertyVisitor.hpp File Reference](#)

LogicPropertyVisitor.hpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp"      #include
"multiscale/verification/spatial-temporal/attribute/-
LogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-temporal/visit-
NumericVisitor.hpp" #include <boost/variant.hpp> Include de-
pendency graph for LogicPropertyVisitor.hpp:
```



Classes

- class [multiscale::verification::LogicPropertyVisitor](#)
Class used to evaluate logic properties.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

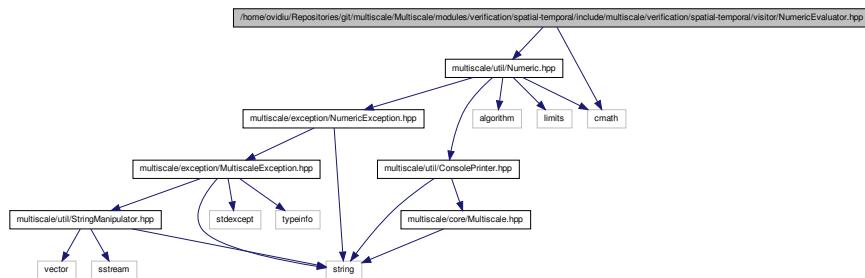
Variables

- static const std::string [multiscale::verification::WRN_LOGIC_PROPERTY_EVALUATOR_FALSE](#) = "The enclosing logic property was evaluated to the default value \"false\"."

**8.194 /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

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.195 /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-tempo  

Comparatorevaluator.hpp" #include "multiscale/verification/spatial-temporal/  

NumericEvaluator.hpp" #include "multiscale/verification/spatial-temporal/vi  

TimePointEvaluator.hpp"      #include <boost/variant.hpp> x  

#include <string> #include "multiscale/verification/spatial-temporal/visito  

ConstraintVisitor.hpp" #include "multiscale/verification/spatial-temporal/vi

```

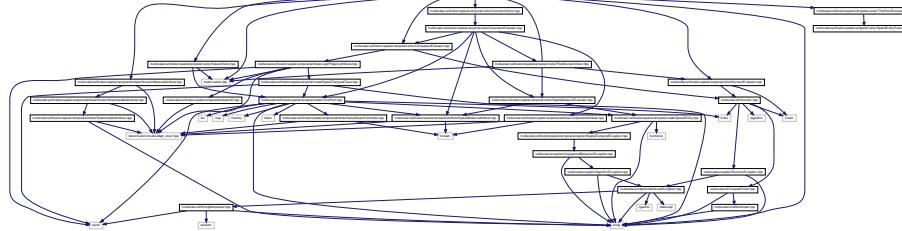
8.196

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/SpatialEntityEvaluator.hpp File Reference

Include dependency graph for NumericVisitor.hpp:

Reference

1163



Classes

- class [multiscale::verification::NumericVisitor](#)
Class for evaluating numeric measures.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.196 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/- SpatialEntityEvaluator.hpp File Reference

Classes

- class [multiscale::verification::SpatialEntityEvaluator](#)
Class used to evaluate spatial entities.

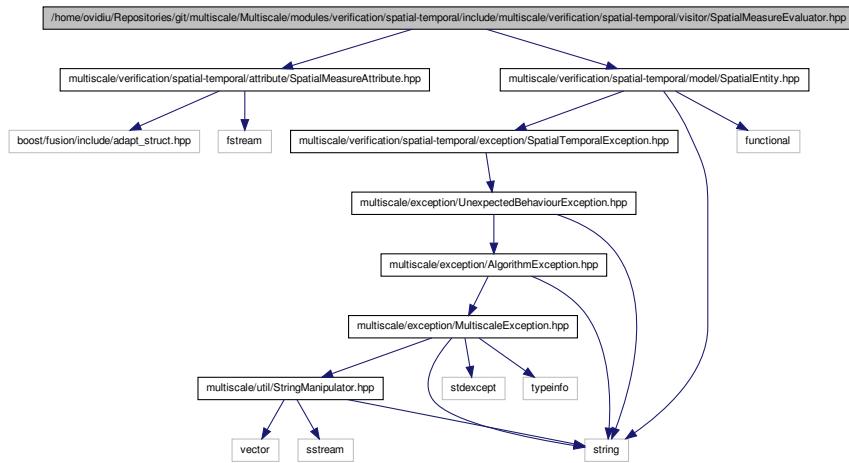
Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.197 /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/mod
```

`SpatialEntity.hpp`" Include dependency graph for `SpatialMeasureEvaluator.hpp`:



Classes

- class `multiscale::verification::SpatialMeasureEvaluator`

Class for evaluating spatial measures.

Namespaces

- namespace `multiscale`
- namespace `multiscale::verification`

8.198 /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 dependency
```

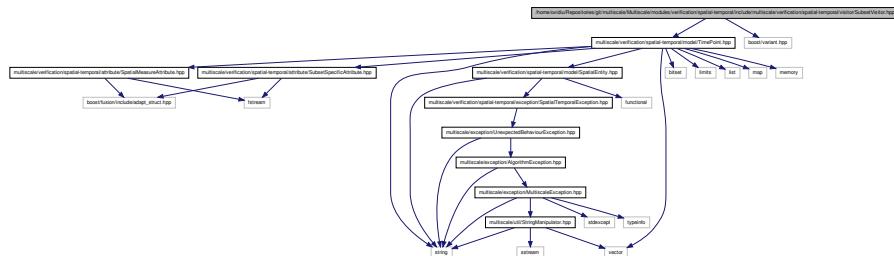
8.199

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/include/multiscale/verification/spatial-temporal/visitor/TimePointGraphForSubsetVisitor.hpp:

Evaluator.hpp File

Reference

1165



Classes

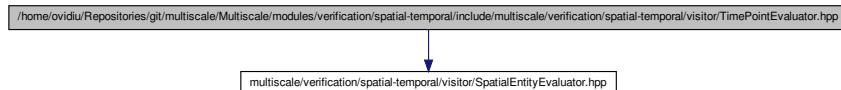
- class [multiscale::verification::SubsetVisitor](#)
Class used to evaluate subsets.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::verification](#)

8.199 /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/-SpatialEntityEvaluator.hpp" Include dependency graph for TimePointEvaluator.hpp:
```



Classes

- class [multiscale::verification::TimePointEvaluator](#)
Class used to evaluate timepoints.

Namespaces

- namespace `multiscale`
- namespace `multiscale::verification`

8.200 /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-temp-
AbstractSyntaxTree.hpp" #include "multiscale/verification/spatial-temporal/
Parser.hpp" #include <iostream> #include <string> #include
<vector> Include dependency graph for LogicPropertyDataReader.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)`

8.200.1 Function Documentation

8.200.1.1 int main (int argc, char ** argv)

Definition at line 61 of file LogicPropertyDataReaderSample.cpp.

References `multiscale::EXEC_SUCCESS_CODE`, and `readQueriesFromFile()`.

8.200.1.2 void printParsingResult (Parser & parser, AbstractSyntaxTree & parsingResult)

Definition at line 17 of file LogicPropertyDataReaderSample.cpp.

References `multiscale::verification::Parser::parse()`, and `multiscale::ExceptionHandler::printErrorMessage()`.

8.201

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserEvaluationSample.cpp File

Referenced by printQueries().

1167

8.200.1.3 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().

8.200.1.4 void readQueriesFromFile (const std::string & path)

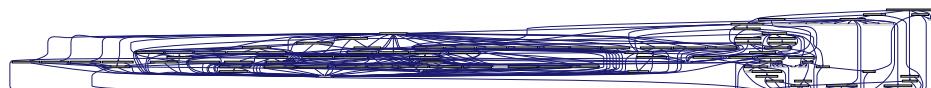
Definition at line 48 of file LogicPropertyDataReaderSample.cpp.

References multiscale::ExceptionHandler::printErrorMessage(), printQueries(), and multiscale::verification::LogicPropertyDataReader::readLogicPropertiesFromFile().

Referenced by main().

8.201 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserEvaluationSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/verification/spatial-temporal/attribute/-_
ProbabilisticLogicPropertyAttribute.hpp" #include "multiscale/verification/spatial-t-
Parser.hpp" #include <iostream> Include dependency graph for Parser-
EvaluationSample.cpp:
```



Functions

- void [initialiseTrace \(SpatialTemporalTrace &trace\)](#)
- int [main \(int argc, char **argv\)](#)

8.201.1 Function Documentation

8.201.1.1 void initialiseTrace (SpatialTemporalTrace & trace)

Definition at line 13 of file ParserEvaluationSample.cpp.

References multiscale::verification::SpatialTemporalTrace::addTimePoint().

Referenced by main().

8.201.1.2 int main (int argc, char ** argv)

Definition at line 40 of file ParserEvaluationSample.cpp.

References multiscale::verification::AbstractSyntaxTree::evaluate(), multiscale::EXEC_C_ERR_CODE, initialiseTrace(), multiscale::verification::Parser::parse(), and multiscale::ExceptionHandler::printErrorMessage().

8.202 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/ParserSample.cpp File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/verification/spatial-temporal/attribute/-
ProbabilisticLogicPropertyAttribute.hpp" #include "multiscale/verification/
Parser.hpp" #include <iostream> Include dependency graph for Parser-
Sample.cpp:
```



Functions

- int [main](#) (int argc, char **argv)

8.202.1 Function Documentation

8.202.1.1 int main (int argc, char ** argv)

Definition at line 13 of file ParserSample.cpp.

References multiscale::verification::AbstractSyntaxTree::evaluate(), multiscale::EXEC_C_ERR_CODE, multiscale::verification::Parser::parse(), and multiscale::ExceptionHandler::printErrorMessage().

8.203

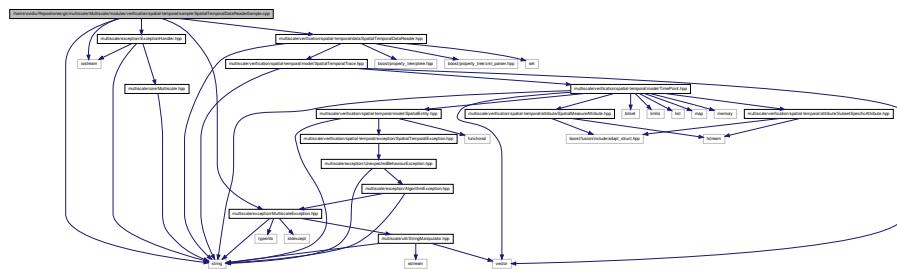
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/sample/SpatialTemporalDataReaderSample.cpp File

8.203 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/sample/SpatialTemporalDataReaderSample.cpp -

File Reference

```
#include "multiscale/exception/ExceptionHandler.hpp" x
#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)

8.203.1 Function Documentation

8.203.1.1 int main (int argc, char ** argv)

Definition at line 78 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::EXEC_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, and `readValidXmlFilesFromFolder()`.

8.203.1.2 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().

8.203.1.3 void printTimePoint (TimePoint & timePoint)

Definition at line 33 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::Clusters, multiscale::verification::TimePoint::getValue(), printSpatialEntities(), and multiscale::verification::Regions.

Referenced by printTrace().

8.203.1.4 void printTrace (SpatialTemporalTrace & trace)

Definition at line 45 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::SpatialTemporalTrace::getTimePoint(), multiscale::verification::SpatialTemporalTrace::length(), and printTimePoint().

Referenced by readValidXmlFiles().

8.203.1.5 void readValidXmlFiles (SpatialTemporalDataReader & reader)

Definition at line 58 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::verification::SpatialTemporalDataReader::getNextSpatialTemporalTrace(), multiscale::verification::SpatialTemporalDataReader::hasNext(), and printTrace().

Referenced by readValidXmlFilesFromFolder().

8.203.1.6 void readValidXmlFilesFromFolder (const std::string & path)

Definition at line 67 of file SpatialTemporalDataReaderSample.cpp.

References multiscale::ExceptionHandler::printErrorMessage(), and readValidXmlFiles().

Referenced by main().

8.204 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinaryNumericMeasureAttribute.cpp

File Reference

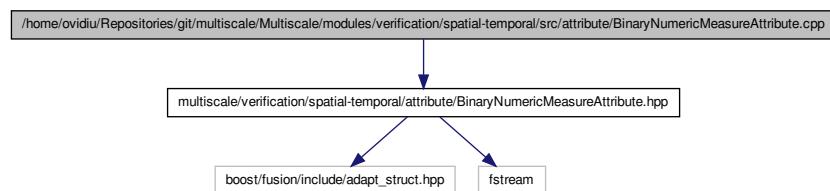
```
#include "multiscale/verification/spatial-temporal/attribute/-  
BinaryNumericMeasureAttribute.hpp" Include dependency graph for
```

8.205

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinarySubsetMeasureAttribute.cpp File

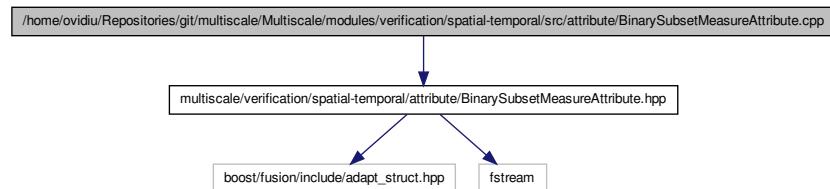
BinaryNumericMeasureAttribute.cpp:

1171



8.205 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/BinarySubsetMeasureAttribute.cpp - File Reference

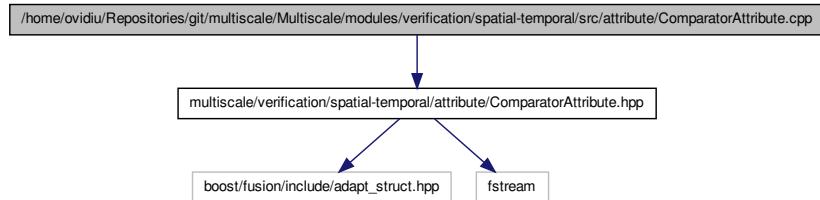
```
#include "multiscale/verification/spatial-temporal/attribute/-  
BinarySubsetMeasureAttribute.hpp" Include dependency graph for -  
BinarySubsetMeasureAttribute.cpp:
```



8.206 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/ComparatorAttribute.cpp File Reference

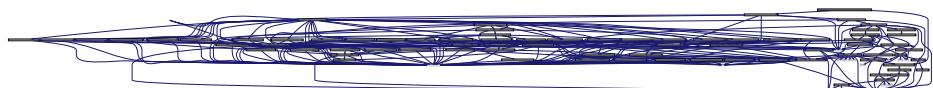
```
#include "multiscale/verification/spatial-temporal/attribute/-  
ComparatorAttribute.hpp" Include dependency graph for Comparator-
```

Attribute.cpp:



8.207 /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/  
SpatialTemporalException.hpp" Include dependency graph for Probabilistic-  
LogicPropertyAttribute.cpp:
```



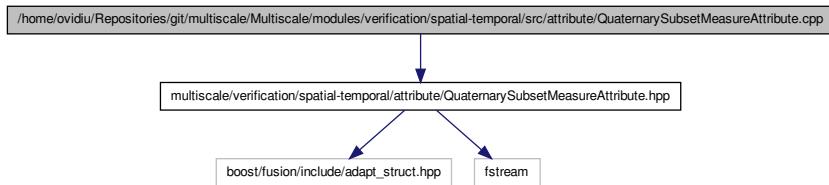
8.208 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/QuaternarySubsetMeasureAttribute.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-  
QuaternarySubsetMeasureAttribute.hpp" Include dependency graph for
```

8.209

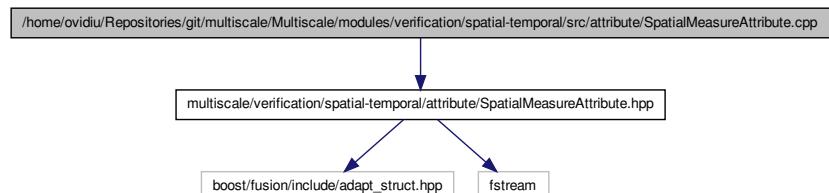
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SpatialMeasureAttribute.cpp File
QuaternarySubsetMeasureAttribute.cpp:

1173



8.209 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/SpatialMeasureAttribute.cpp File - Reference

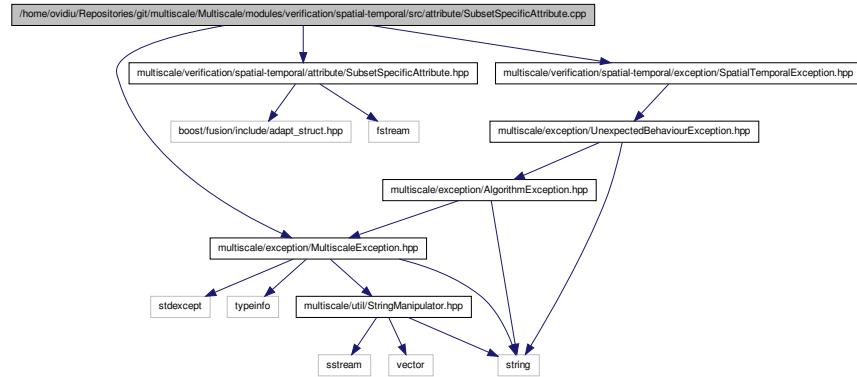
```
#include "multiscale/verification/spatial-temporal/attribute/-  
SpatialMeasureAttribute.hpp" Include dependency graph for Spatial-  
MeasureAttribute.cpp:
```



8.210 /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/exce-  
SpatialTemporalException.hpp" Include dependency graph for Subset-
```

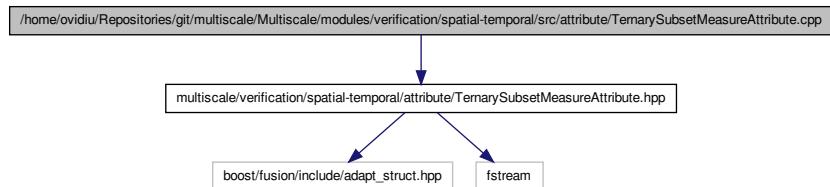
SpecificAttribute.cpp:



8.211 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/TernarySubsetMeasureAttribute.cpp

File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-TernarySubsetMeasureAttribute.hpp" Include dependency graph for TernarySubsetMeasureAttribute.cpp:
```



8.212 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnaryNumericMeasureAttribute.cpp

File Reference

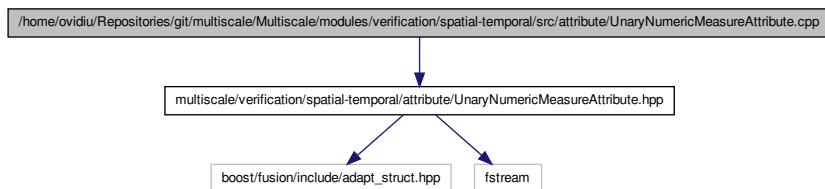
```
#include "multiscale/verification/spatial-temporal/attribute/-UnaryNumericMeasureAttribute.hpp" Include dependency graph for -
```

8.213

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/attribute/UnarySubsetMeasureAttribute.cpp File

File Reference

1175

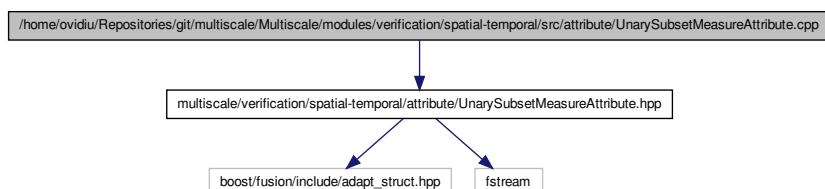


8.213 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/attribute/UnarySubsetMeasureAttribute.cpp -

File Reference

```
#include "multiscale/verification/spatial-temporal/attribute/-
UnarySubsetMeasureAttribute.hpp" Include dependency graph for Unary-
SubsetMeasureAttribute.cpp:
```



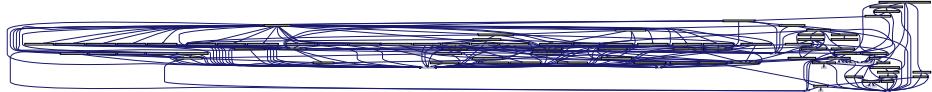
8.214 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/checking/ApproximateBayesianModelChecker.cpp

File Reference

```
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/exception/UnexpectedBehaviour-
Exception.hpp" #include "multiscale/util/Numeric.hpp" ×
#include "multiscale/util/StringManipulator.hpp" #include
"multiscale/verification/spatial-temporal/checking/Approximate-
BayesianModelChecker.hpp" Include dependency graph for Approximate-
```

BayesianModelChecker.cpp:



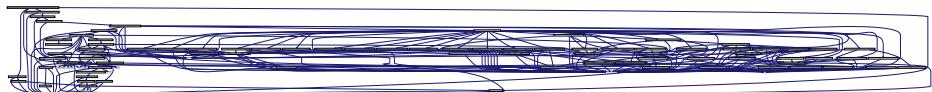
8.215 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateBayesianModelChecker-Factory.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/-  
ApproximateBayesianModelChecker.hpp" #include "multiscale/verification/spat  
ApproximateBayesianModelCheckerFactory.hpp" Include dependency  
graph for ApproximateBayesianModelCheckerFactory.cpp:
```



8.216 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModel-Checker.cpp File Reference

```
#include "multiscale/exception/InvalidInputException.-  
hpp" #include "multiscale/util/Numeric.hpp" #include "multiscale/util/-  
StringManipulator.hpp" #include "multiscale/verification/spatial-temporal/c  
ApproximateProbabilisticModelChecker.hpp" Include dependency  
graph for ApproximateProbabilisticModelChecker.cpp:
```



8.217 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ApproximateProbabilisticModel-CheckerFactory.cpp File Reference

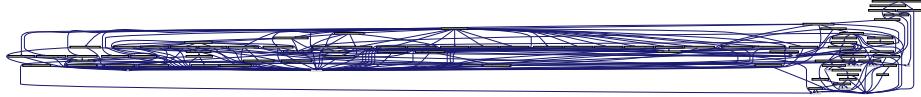
```
#include "multiscale/verification/spatial-temporal/checking/-
```

8.218

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/BayesianModelChecker.cpp File

Reference

ApproximateProbabilisticModelChecker.hpp" #include "multiscale/verification/spatial-temporal/ApproximateProbabilisticModelCheckerFactory.hpp" Include dependency graph for ApproximateProbabilisticModelCheckerFactory.cpp:

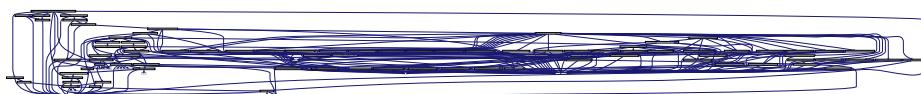


8.218 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/checking/BayesianModelChecker.cpp File -

Reference

```
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/exception/UnexpectedBehaviours-
Exception.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:
```

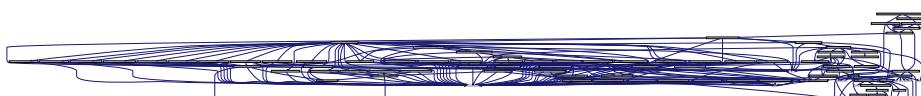


8.219 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/checking/BayesianModelCheckerFactory.cpp

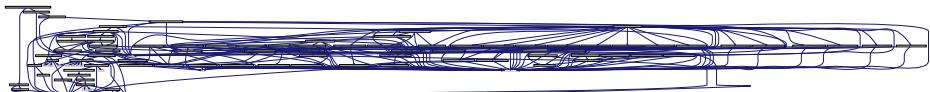
File Reference

```
#include "multiscale/verification/spatial-temporal/checking/-
BayesianModelChecker.hpp" #include "multiscale/verification/spatial-temporal/checkin-
BayesianModelCheckerFactory.hpp" Include dependency graph for -
BayesianModelCheckerFactory.cpp:
```



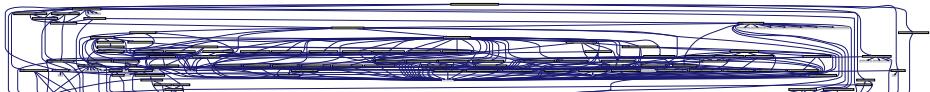
8.220 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelChecker.cpp File Reference

```
#include "multiscale/util/statistics/BinomialDistribution.-  
hpp" #include "multiscale/verification/spatial-temporal/checking/-  
ModelChecker.hpp" Include dependency graph for ModelChecker.cpp:
```



8.221 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingManager.cpp File - Reference

```
#include "multiscale/exception/ExceptionHandler.hpp" x  
#include "multiscale/exception/InvalidInputException.-  
hpp" #include "multiscale/util/OperatingSystem.hpp" #include  
"multiscale/verification/spatial-temporal/checking/Probabilistic-  
BlackBoxModelChecker.hpp" #include "multiscale/verification/spatial-temporal/  
ModelCheckingOutputWriter.hpp" #include "multiscale/verification/spatial-te  
ModelCheckingManager.hpp" Include dependency graph for ModelChecking-  
Manager.cpp:
```



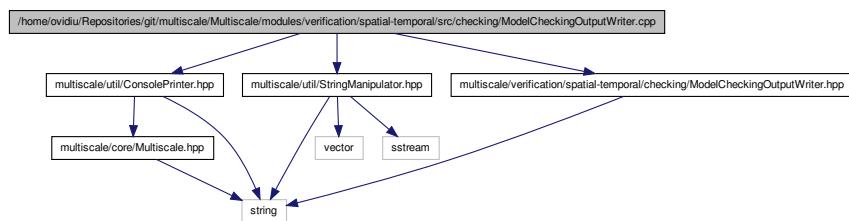
8.222 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.cpp File Reference

```
#include "multiscale/util/ConsolePrinter.hpp"      #include  
"multiscale/util/StringManipulator.hpp" #include "multiscale/verification/s  
ModelCheckingOutputWriter.hpp" Include dependency graph for Model-
```

8.223

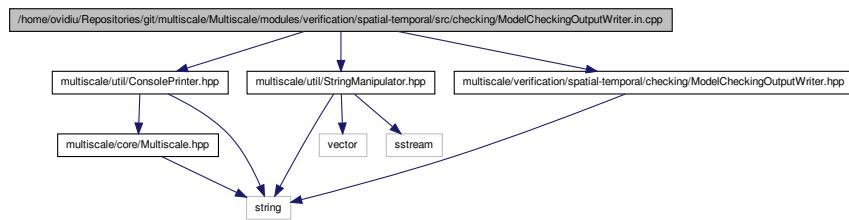
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.in.cpp File
File Reference

1179



8.223 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ModelCheckingOutputWriter.in.cpp
File Reference

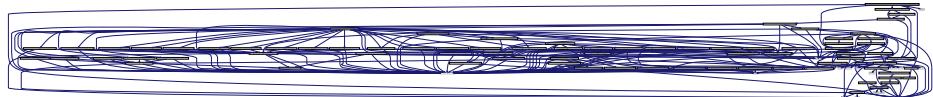
```
#include "multiscale/util/ConsolePrinter.hpp"      #include
"multiscale/util/StringManipulator.hpp" #include "multiscale/verification/spatial-te
ModelCheckingOutputWriter.hpp" Include dependency graph for Model-
CheckingOutputWriter.in.cpp:
```



8.224 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelChecker.cpp
File Reference

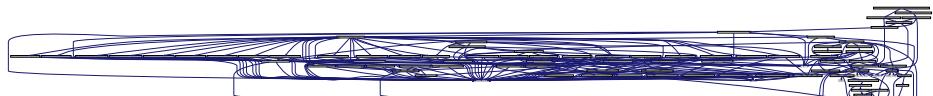
```
#include "multiscale/util/Numeric.hpp" #include "multiscale/util/-
StringManipulator.hpp" #include "multiscale/verification/spatial-temporal/checking/-
ProbabilisticBlackBoxModelChecker.hpp" #include "multiscale/verification/spatial-tem
ComparatorEvaluator.hpp" #include <iostream> Include depen-
```

dependency graph for ProbabilisticBlackBoxModelChecker.cpp:



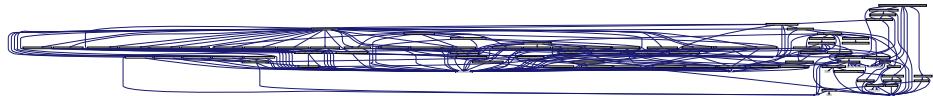
8.225 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/ProbabilisticBlackBoxModelChecker-Factory.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/checking/-
ProbabilisticBlackBoxModelChecker.hpp" #include "multiscale/verification/sp-
ProbabilisticBlackBoxModelCheckerFactory.hpp" Include dependency graph for ProbabilisticBlackBoxModelCheckerFactory.cpp:
```



8.226 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/checking/StatisticalModelChecker.cpp - Reference

```
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/exception/UnexpectedBehaviour-
Exception.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:
```



8.227

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/checking/StatisticalModelCheckerFactory.cpp File

Reference

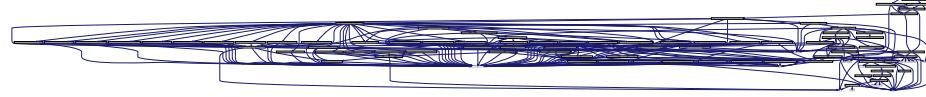
8.227 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/src/checking/StatisticalModelCheckerFactory.cpp File

Reference

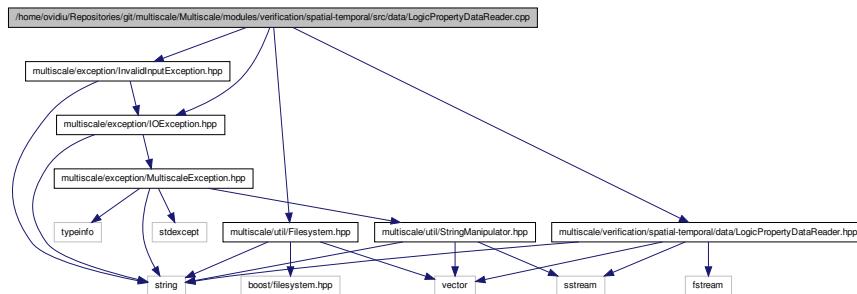
File Reference

```
#include "multiscale/verification/spatial-temporal/checking/-  
StatisticalModelChecker.hpp" #include "multiscale/verification/spatial-temporal/che  
StatisticalModelCheckerFactory.hpp" Include dependency graph for  
StatisticalModelCheckerFactory.cpp:
```



**8.228 /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" x  
#include "multiscale/util/Filesystem.hpp" #include "multiscale/verification/spatial-  
LogicPropertyDataReader.hpp" Include dependency graph for Logic-  
PropertyDataReader.cpp:
```



**8.229 /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/exception/UnexpectedBehaviour-  

Exception.hpp"      #include "multiscale/util/Filesystem.-  

.hpp" #include "multiscale/util/XmlValidator.hpp" #include  

"multiscale/verification/spatial-temporal/data/Spatial-  

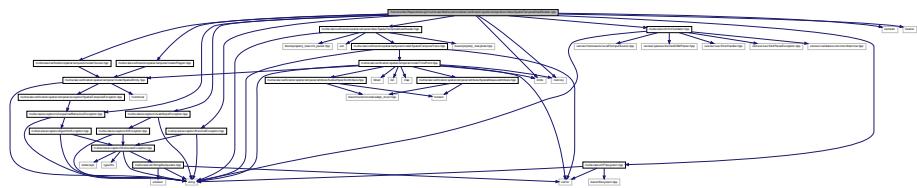
TemporalDataReader.hpp" #include "multiscale/verification/spatial-temporal/  

Cluster.hpp" #include "multiscale/verification/spatial-temporal/model/-  

Region.hpp"      #include <iostream>    #include <iterator>x  

#include <limits> #include <memory> Include dependency graph for  

SpatialTemporalDataReader.cpp:
```



8.230 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/data/SpatialTemporalDataReader.in.cpp File - Reference

```

#include "multiscale/exception/InvalidInputException.-  

.hpp"      #include "multiscale/exception/RuntimeException.-  

.hpp"      #include "multiscale/exception/UnexpectedBehaviour-  

Exception.hpp"      #include "multiscale/util/Filesystem.-  

.hpp" #include "multiscale/util/XmlValidator.hpp" #include  

"multiscale/verification/spatial-temporal/data/Spatial-  

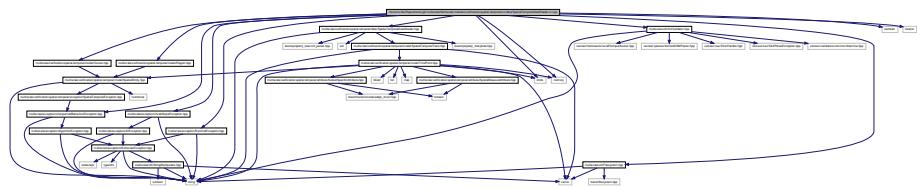
TemporalDataReader.hpp" #include "multiscale/verification/spatial-temporal/  

Cluster.hpp" #include "multiscale/verification/spatial-temporal/model/-  

Region.hpp"      #include <iostream>    #include <iterator>x  

#include <limits> #include <memory> Include dependency graph for  

SpatialTemporalDataReader.in.cpp:
```



8.231

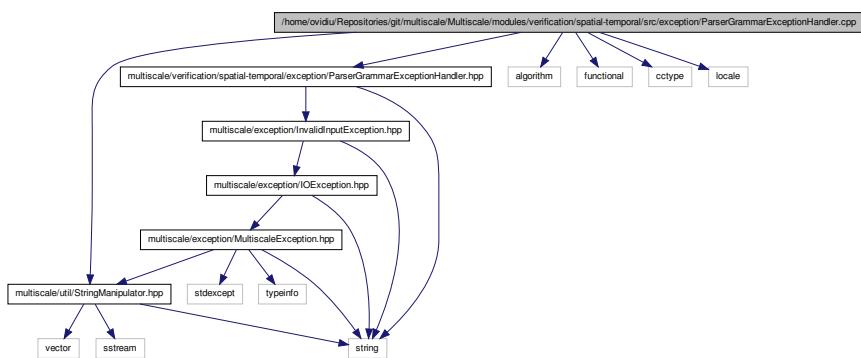
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ParserGrammarExceptionHandler.cpp File

8.231 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/exception/ParserGrammarExceptionHandler.cpp

File Reference

File Reference

```
#include "multiscale/util/StringManipulator.hpp" #include  
"multiscale/verification/spatial-temporal/exception/-  
ParserGrammarExceptionHandler.hpp" #include <algorithm>  
#include <functional> #include <cctype> #include <locale> x  
Include dependency graph for ParserGrammarExceptionHandler.cpp:
```



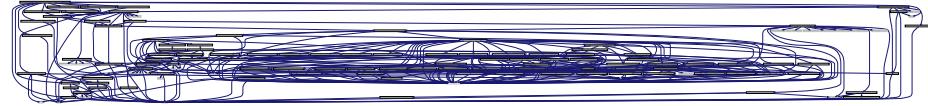
8.232 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

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-  
ApproximateProbabilisticModelCheckerFactory.hpp" #include  
"multiscale/verification/spatial-temporal/checking/Bayesian-  
ModelCheckerFactory.hpp" #include "multiscale/verification/spatial-temporal/checking/  
ProbabilisticBlackBoxModelCheckerFactory.hpp" #include  
"multiscale/verification/spatial-temporal/checking/Statistical-  
ModelCheckerFactory.hpp" #include "multiscale/verification/spatial-temporal/exception/  
ModelCheckingException.hpp" #include "multiscale/verification/spatial-temporal/exception/  
ModelCheckingHelpRequestException.hpp" #include "multiscale/verification/spatial-tem-  
CommandLineModelChecking.hpp" #include <iostream> Include de-
```

pendency graph for CommandLineModelChecking.cpp:



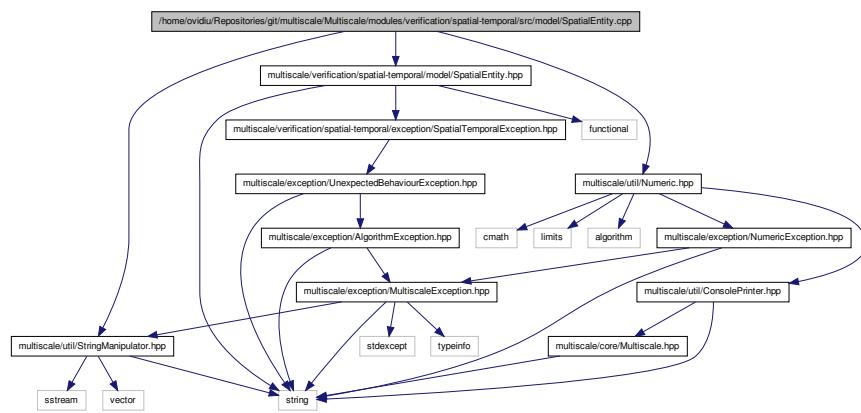
8.233 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/AbstractSyntaxTree.cpp File Reference

```
#include "multiscale/exception/UnexpectedBehaviourException.-  
hpp" #include "multiscale/verification/spatial-temporal/model/-  
AbstractSyntaxTree.hpp" Include dependency graph for AbstractSyntaxTree.-  
cpp:
```



8.234 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialEntity.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/model/-  
SpatialEntity.hpp" #include "multiscale/util/Numeric.-  
hpp" #include "multiscale/util/StringManipulator.hpp" Include  
dependency graph for SpatialEntity.cpp:
```



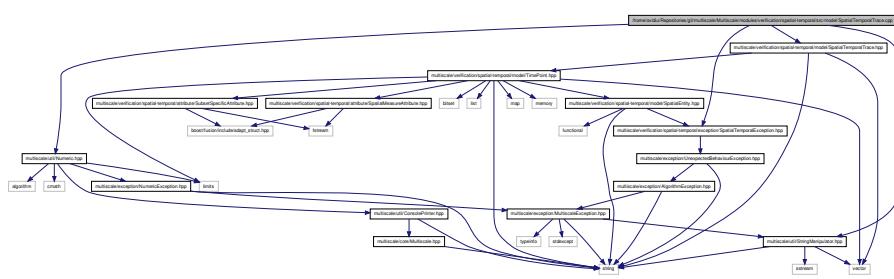
8.235

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/model/SpatialTemporalTrace.cpp File

8.235 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-
Reference 183

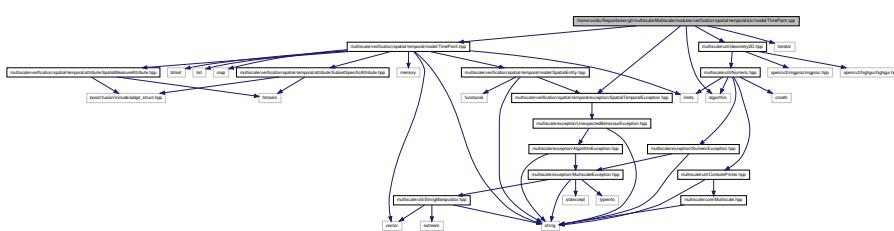
temporal/src/model/SpatialTemporalTrace.cpp File Reference

```
#include "multiscale/util/Numeric.hpp" #include "multiscale/util/-  
StringManipulator.hpp" #include "multiscale/verification/spatial-temporal/exception/  
SpatialTemporalException.hpp" #include "multiscale/verification/spatial-temporal/mod-  
SpatialTemporalTrace.hpp" Include dependency graph for SpatialTemporal-  
Trace.cpp:
```



8.236 /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-  
SpatialTemporalException.hpp" #include "multiscale/verification/spatial-temporal/mod  
TimePoint.hpp" #include <algorithm> #include <iterator>  
Include dependency graph for TimePoint.cpp:
```



8.237 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/src/Mule.cpp File Reference

```
#include "multiscale/verification/spatial-temporal/exception/-  
ModelCheckingHelpRequestException.hpp" #include "multiscale/ver
```

CommandLineModelChecking.hpp" Include dependency graph for Mule.cpp:



Functions

- void [runModelCheckingTask](#) (int argc, char **argv)
- int [main](#) (int argc, char **argv)

8.237.1 Function Documentation

8.237.1.1 int main (int argc, char ** argv)

Definition at line 23 of file Mule.cpp.

References multiscale::EXEC_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, multiscale::ExceptionHandler::printErrorMessage(), and runModelCheckingTask().

8.237.1.2 void runModelCheckingTask (int argc, char ** argv)

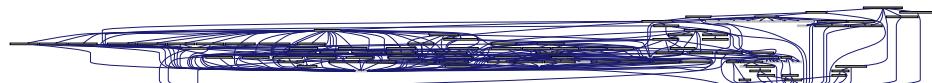
Definition at line 15 of file Mule.cpp.

References multiscale::verification::CommandLineModelChecking::execute(), and multiscale::verification::CommandLineModelChecking::initialise().

Referenced by main().

8.238 /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-
ParserGrammarExtraInputException.hpp" #include "multiscale/verification/spatial-
ParserGrammarProbabilityException.hpp" #include "multiscale/verification/spatial-
ParserGrammarUnexpectedTokenException.hpp" #include "multiscale/verification/spatial-
ParserGrammarUnparseableInputException.hpp" Include dependency
graph for Parser.cpp:
```

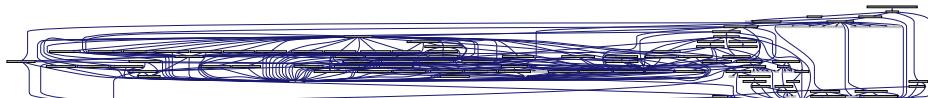


8.239

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericFilterTest.hpp File
8.239 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-
Reference 187

temporal/test/attribute/BinaryNumericFilterTest.hpp File - Reference

```
#include "parsing/InputStreamParser.hpp"  Include dependency  
graph for BinaryNumericFilterTest.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 \(BinaryNumericFilter, Correct\)](#)

8.239.1 Function Documentation

8.239.1.1 TEST (BinaryNumericFilter , IncorrectInputMissingParameterOne)

Definition at line 12 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputStream().

8.239.1.2 TEST (BinaryNumericFilter , IncorrectInputMissingParameterTwo)

Definition at line 16 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputStream().

8.239.1.3 TEST (BinaryNumericFilter , IncorrectInputMissingParametersOneTwo)

Definition at line 20 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputStream().

8.239.1.4 TEST (BinaryNumericFilter , IncorrectInputBeforeStartBracket)

Definition at line 24 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.5 TEST (BinaryNumericFilter , IncorrectInputAfterStartBracket)

Definition at line 28 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.6 TEST (BinaryNumericFilter , InvalidFirstParameter)

Definition at line 32 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.7 TEST (BinaryNumericFilter , MissingParametersComma)

Definition at line 36 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.8 TEST (BinaryNumericFilter , InvalidSecondParameter)

Definition at line 40 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.9 TEST (BinaryNumericFilter , IncorrectInputBeforeEndBracket)

Definition at line 44 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.10 TEST (BinaryNumericFilter , IncorrectInputAfterEndBracket)

Definition at line 48 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.239.1.11 TEST (BinaryNumericFilter , Correct)

Definition at line 52 of file BinaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.240

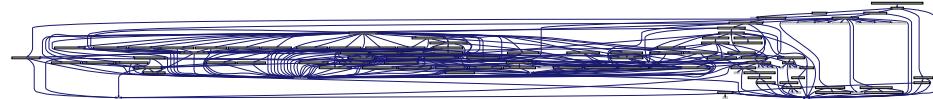
/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericMeasureTest.hpp File
8.240 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericMeasureTest.hpp File

Reference

temporal/test/attribute/BinaryNumericMeasureTest.hpp File

Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency  
graph for BinaryNumericMeasureTest.hpp:
```



Functions

- [TEST \(BinaryNumericMeasure, IncorrectBinaryNumericMeasure\)](#)
- [TEST \(BinaryNumericMeasure, CorrectAdd\)](#)
- [TEST \(BinaryNumericMeasure, CorrectDiv\)](#)
- [TEST \(BinaryNumericMeasure, CorrectLog\)](#)
- [TEST \(BinaryNumericMeasure, CorrectMod\)](#)
- [TEST \(BinaryNumericMeasure, CorrectMultiply\)](#)
- [TEST \(BinaryNumericMeasure, CorrectPower\)](#)
- [TEST \(BinaryNumericMeasure, CorrectSubtract\)](#)

8.240.1 Function Documentation

8.240.1.1 TEST (BinaryNumericMeasure , IncorrectBinaryNumericMeasure)

Definition at line 13 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.2 TEST (BinaryNumericMeasure , CorrectAdd)

Definition at line 17 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.3 TEST (BinaryNumericMeasure , CorrectDiv)

Definition at line 21 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.4 TEST (BinaryNumericMeasure , CorrectLog)

Definition at line 25 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.5 TEST (BinaryNumericMeasure , CorrectMod)

Definition at line 29 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.6 TEST (BinaryNumericMeasure , CorrectMultiply)

Definition at line 33 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.240.1.7 TEST (BinaryNumericMeasure , CorrectPower)

Definition at line 37 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

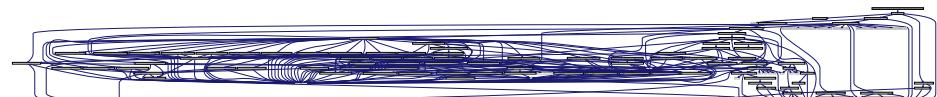
8.240.1.8 TEST (BinaryNumericMeasure , CorrectSubtract)

Definition at line 41 of file BinaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.241 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericNumericTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"    Include dependency
graph for BinaryNumericNumericTest.hpp:
```



Functions

- [TEST \(BinaryNumericNumeric, IncorrectInputMissingParameterOne\)](#)

8.241

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinaryNumericNumericTest.hpp File

Reference [TEST \(BinaryNumericNumeric, IncorrectInputMissingParameterTwo\)](#) 1191

- [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 \(BinaryNumericNumeric, Correct\)](#)

8.241.1 Function Documentation

8.241.1.1 TEST (BinaryNumericNumeric , IncorrectInputMissingParameterOne)

Definition at line 12 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.2 TEST (BinaryNumericNumeric , IncorrectInputMissingParameterTwo)

Definition at line 16 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.3 TEST (BinaryNumericNumeric , IncorrectInputMissingParametersOneTwo)

Definition at line 20 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.4 TEST (BinaryNumericNumeric , IncorrectInputBeforeStartBracket)

Definition at line 24 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.5 TEST (BinaryNumericNumeric , IncorrectInputAfterStartBracket)

Definition at line 28 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.6 TEST (BinaryNumericNumeric , InvalidFirstParameter)

Definition at line 32 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.7 TEST (BinaryNumericNumeric , MissingParametersComma)

Definition at line 36 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.8 TEST (BinaryNumericNumeric , InvalidSecondParameter)

Definition at line 40 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.9 TEST (BinaryNumericNumeric , IncorrectInputBeforeEndBracket)

Definition at line 44 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.241.1.10 TEST (BinaryNumericNumeric , IncorrectInputAfterEndBracket)

Definition at line 48 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

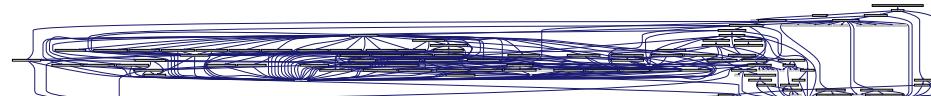
8.241.1.11 TEST (BinaryNumericNumeric , Correct)

Definition at line 52 of file BinaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.242 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetMeasureTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for BinarySubsetMeasureTest.hpp:



Functions

- [TEST](#) (BinarySubsetMeasure, IncorrectBinarySubsetMeasure)

8.242

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetMeasureTest.hpp File

Reference [TEST \(BinarySubsetMeasure, CorrectAvg\)](#)

1193

- [TEST \(BinarySubsetMeasure, CorrectGeomean\)](#)
- [TEST \(BinarySubsetMeasure, CorrectHarmean\)](#)
- [TEST \(BinarySubsetMeasure, CorrectKurt\)](#)
- [TEST \(BinarySubsetMeasure, CorrectMax\)](#)
- [TEST \(BinarySubsetMeasure, CorrectMedian\)](#)
- [TEST \(BinarySubsetMeasure, CorrectMin\)](#)
- [TEST \(BinarySubsetMeasure, CorrectMode\)](#)
- [TEST \(BinarySubsetMeasure, CorrectProduct\)](#)
- [TEST \(BinarySubsetMeasure, CorrectSkew\)](#)
- [TEST \(BinarySubsetMeasure, CorrectStddev\)](#)
- [TEST \(BinarySubsetMeasure, CorrectSum\)](#)
- [TEST \(BinarySubsetMeasure, CorrectVar\)](#)

8.242.1 Function Documentation

8.242.1.1 TEST (BinarySubsetMeasure , IncorrectBinarySubsetMeasure)

Definition at line 12 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.2 TEST (BinarySubsetMeasure , CorrectAvg)

Definition at line 16 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.3 TEST (BinarySubsetMeasure , CorrectGeomean)

Definition at line 20 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.4 TEST (BinarySubsetMeasure , CorrectHarmean)

Definition at line 24 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.5 TEST (BinarySubsetMeasure , CorrectKurt)

Definition at line 28 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.6 TEST (BinarySubsetMeasure , CorrectMax)

Definition at line 32 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.7 TEST (BinarySubsetMeasure , CorrectMedian)

Definition at line 36 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.8 TEST (BinarySubsetMeasure , CorrectMin)

Definition at line 40 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.9 TEST (BinarySubsetMeasure , CorrectMode)

Definition at line 44 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.10 TEST (BinarySubsetMeasure , CorrectProduct)

Definition at line 48 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.11 TEST (BinarySubsetMeasure , CorrectSkew)

Definition at line 52 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.12 TEST (BinarySubsetMeasure , CorrectStdev)

Definition at line 56 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.242.1.13 TEST (BinarySubsetMeasure , CorrectSum)

Definition at line 60 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.243

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetTest.hpp File

8.243.1.1 TEST (BinarySubsetMeasure , CorrectVar)

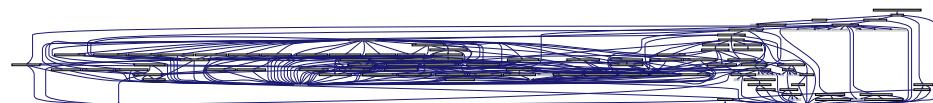
1195

Definition at line 64 of file BinarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.243 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/BinarySubsetTest.hpp File Reference

#include "parsing/InputStreamParser.hpp" Include dependency graph for BinarySubsetTest.hpp:



Functions

- [TEST](#) (BinarySubset, IncorrectMissingFirstParameter)
- [TEST](#) (BinarySubset, IncorrectMissingSecondParameter)
- [TEST](#) (BinarySubset, IncorrectMissingParameters)
- [TEST](#) (BinarySubset, IncorrectInputBeforeStartBracket)
- [TEST](#) (BinarySubset, IncorrectInputAfterStartBracket)
- [TEST](#) (BinarySubset, MissingComma)
- [TEST](#) (BinarySubset, InvalidSpatialMeasure)
- [TEST](#) (BinarySubset, IncorrectInputBeforeEndBracket)
- [TEST](#) (BinarySubset, IncorrectInputAfterEndBracket)
- [TEST](#) (BinarySubset, Correct)

8.243.1 Function Documentation

8.243.1.1 TEST (BinarySubset , IncorrectMissingFirstParameter)

Definition at line 12 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.2 TEST (BinarySubset , IncorrectMissingSecondParameter)

Definition at line 16 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.3 TEST (BinarySubset , IncorrectMissingParameters)

Definition at line 20 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.4 TEST (BinarySubset , IncorrectInputBeforeStartBracket)

Definition at line 24 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.5 TEST (BinarySubset , IncorrectInputAfterStartBracket)

Definition at line 28 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.6 TEST (BinarySubset , MissingComma)

Definition at line 32 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.7 TEST (BinarySubset , InvalidSpatialMeasure)

Definition at line 36 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.8 TEST (BinarySubset , IncorrectInputBeforeEndBracket)

Definition at line 40 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.9 TEST (BinarySubset , IncorrectInputAfterEndBracket)

Definition at line 44 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.243.1.10 TEST (BinarySubset , Correct)

Definition at line 48 of file BinarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.244

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

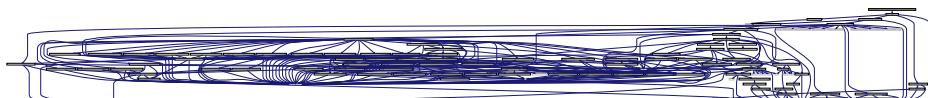
temporal/test/attribute/ComparatorTest.hpp File

Reference

8.244 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/attribute/ComparatorTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency  
graph for ComparatorTest.hpp:
```



Functions

- [TEST \(Comparator, IncorrectEqual\)](#)
- [TEST \(Comparator, IncorrectDifferent1\)](#)
- [TEST \(Comparator, IncorrectDifferent2\)](#)
- [TEST \(Comparator, CorrectGreaterThan\)](#)
- [TEST \(Comparator, CorrectLessThan\)](#)
- [TEST \(Comparator, CorrectGreaterThanOrEqual\)](#)
- [TEST \(Comparator, CorrectLessThanOrEqual\)](#)
- [TEST \(Comparator, CorrectEqual\)](#)

8.244.1 Function Documentation

8.244.1.1 TEST (Comparator , IncorrectEqual)

Definition at line 12 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.2 TEST (Comparator , IncorrectDifferent1)

Definition at line 16 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.3 TEST (Comparator , IncorrectDifferent2)

Definition at line 20 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.4 TEST (Comparator , CorrectGreaterThan)

Definition at line 24 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.5 TEST (Comparator , CorrectLessThan)

Definition at line 28 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.6 TEST (Comparator , CorrectGreaterThanOrEqualTo)

Definition at line 32 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.244.1.7 TEST (Comparator , CorrectLessThanOrEqualTo)

Definition at line 36 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

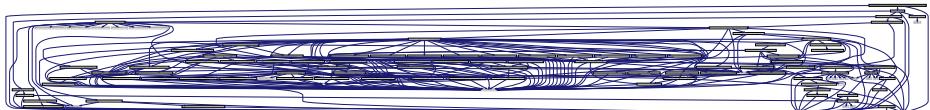
8.244.1.8 TEST (Comparator , CorrectEqual)

Definition at line 40 of file ComparatorTest.hpp.

References multiscaletest::verification::parseInputString().

8.245 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundConstraintTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp" #include <string>
#include <vector> Include dependency graph for CompoundConstraintTest.hpp:
```



Functions

- [TEST](#) (CompoundConstraint, MissingBinaryOperator)
- [TEST](#) (CompoundConstraint, MissingConstraints)
- [TEST](#) (CompoundConstraint, MissingFirstConstraint)
- [TEST](#) (CompoundConstraint, MissingSecondConstraint)
- [TEST](#) (CompoundConstraint, BinaryOperatorAsUnaryBefore)
- [TEST](#) (CompoundConstraint, BinaryOperatorAsUnaryAfter)

8.245

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundConstraintTest.hpp File

Reference [TEST \(CompoundConstraint, NumericNumericComparisonBeforeBinaryOperator\)](#)

Operator)

- [TEST \(CompoundConstraint, UnaryNumericMeasureAfterBinaryOperator\)](#)
- [TEST \(CompoundConstraint, AdditionalOperatorBeforeBinaryOperator\)](#)
- [TEST \(CompoundConstraint, AdditionalOperatorAfterBinaryOperator\)](#)
- [TEST \(CompoundConstraint, Correct\)](#)
- [TEST \(CompoundConstraint, MultipleCorrect\)](#)

Variables

- static const std::vector < std::string > [CONSTRAINTS_BINARY_OPERATORS](#)
= std::vector<std::string>({“^”, “V”, “=>”, “<=>”})

8.245.1 Function Documentation

8.245.1.1 TEST (CompoundConstraint , MissingBinaryOperator)

Definition at line 19 of file CompoundConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.245.1.2 TEST (CompoundConstraint , MissingConstraints)

Definition at line 23 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.245.1.3 TEST (CompoundConstraint , MissingFirstConstraint)

Definition at line 29 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.245.1.4 TEST (CompoundConstraint , MissingSecondConstraint)

Definition at line 35 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.245.1.5 TEST (CompoundConstraint , BinaryOperatorAsUnaryBefore)

Definition at line 41 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.245.1.6 TEST (CompoundConstraint , BinaryOperatorAsUnaryAfter)

Definition at line 47 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.7 TEST (CompoundConstraint , NumericNumericComparisonBeforeBinaryOperator)

Definition at line 53 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.8 TEST (CompoundConstraint , UnaryNumericMeasureAfterBinaryOperator)

Definition at line 59 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.9 TEST (CompoundConstraint , AdditionalOperatorBeforeBinaryOperator)

Definition at line 65 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.10 TEST (CompoundConstraint , AdditionalOperatorAfterBinaryOperator)

Definition at line 71 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.11 TEST (CompoundConstraint , Correct)

Definition at line 77 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.245.1.12 TEST (CompoundConstraint , MultipleCorrect)

Definition at line 83 of file CompoundConstraintTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS, and multiscaletest::verification-::parseInputString().

8.246

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundLogicPropertyTest.hpp File
8.245.2 Variable Documentation

1201

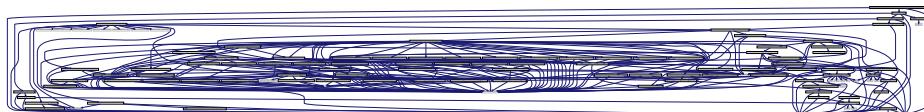
8.245.2.1 const std::vector<std::string> CONSTRAINTS_BINARY_OPERATORS =
std::vector<std::string>({" \wedge ", " \vee ", " \Rightarrow ", " $\Leftarrow\Rightarrow$ "}) [static]

Definition at line 14 of file CompoundConstraintTest.hpp.

Referenced by TEST(), and TEST_F().

8.246 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundLogicPropertyTest.hpp - File Reference

```
#include "parsing/InputStringParser.hpp" #include <string>
#include <vector> Include dependency graph for CompoundLogicProperty-
Test.hpp:
```



Functions

- [TEST](#) (CompoundLogicProperty, MissingBinaryOperator)
- [TEST](#) (CompoundLogicProperty, MissingConstraints)
- [TEST](#) (CompoundLogicProperty, MissingFirstConstraint)
- [TEST](#) (CompoundLogicProperty, MissingSecondConstraint)
- [TEST](#) (CompoundLogicProperty, BinaryOperatorAsUnaryBefore)
- [TEST](#) (CompoundLogicProperty, BinaryOperatorAsUnaryAfter)
- [TEST](#) (CompoundLogicProperty, UnarySubsetMeasureBeforeBinaryOperator)
- [TEST](#) (CompoundLogicProperty, UnaryNumericMeasureAfterBinaryOperator)
- [TEST](#) (CompoundLogicProperty, AdditionalOperatorBeforeBinaryOperator)
- [TEST](#) (CompoundLogicProperty, AdditionalOperatorAfterBinaryOperator)
- [TEST](#) (CompoundLogicProperty, Correct)
- [TEST](#) (CompoundLogicProperty, MultipleCorrect)

Variables

- static const std::vector < std::string > [LOGIC_PROPERTIES_BINARY_OPERATORS](#) = std::vector<std::string>({" \wedge ", " \vee ", " \Rightarrow ", " $\Leftarrow\Rightarrow$ "})

8.246.1 Function Documentation

8.246.1.1 TEST (CompoundLogicProperty , MissingBinaryOperator)

Definition at line 19 of file CompoundLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.246.1.2 TEST (CompoundLogicProperty , MissingConstraints)

Definition at line 23 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246.1.3 TEST (CompoundLogicProperty , MissingFirstConstraint)

Definition at line 29 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246.1.4 TEST (CompoundLogicProperty , MissingSecondConstraint)

Definition at line 35 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246.1.5 TEST (CompoundLogicProperty , BinaryOperatorAsUnaryBefore)

Definition at line 41 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246.1.6 TEST (CompoundLogicProperty , BinaryOperatorAsUnaryAfter)

Definition at line 47 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246.1.7 TEST (CompoundLogicProperty , UnarySubsetMeasureBeforeBinaryOperator)

Definition at line 53 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest::verification::parseInputString().

8.246

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/CompoundLogicPropertyTest.hpp File
8.246.1.8 TEST (CompoundLogicProperty , UnaryNumericMeasureAfterBinaryOperator)
[Reference](#) [1203](#)

Definition at line 59 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest-::verification::parseInputString().

8.246.1.9 TEST (CompoundLogicProperty , AdditionalOperatorBeforeBinaryOperator)

Definition at line 65 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest-::verification::parseInputString().

8.246.1.10 TEST (CompoundLogicProperty , AdditionalOperatorAfterBinaryOperator)

Definition at line 71 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest-::verification::parseInputString().

8.246.1.11 TEST (CompoundLogicProperty , Correct)

Definition at line 77 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest-::verification::parseInputString().

8.246.1.12 TEST (CompoundLogicProperty , MultipleCorrect)

Definition at line 83 of file CompoundLogicPropertyTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS, and multiscaletest-::verification::parseInputString().

8.246.2 Variable Documentation

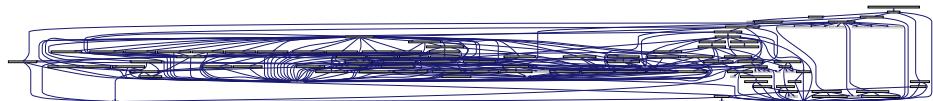
8.246.2.1 const std::vector<std::string> LOGIC_PROPERTIES_BINARY_OPERATORS = std::vector<std::string>({”^”, ”V”, ”=>”, ”<=>”})
[static]

Definition at line 14 of file CompoundLogicPropertyTest.hpp.

Referenced by TEST(), and TEST_F().

8.247 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ConstraintParenthesesTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for ConstraintParenthesesTest.hpp:



Functions

- [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)

8.247.1 Function Documentation

8.247.1.1 TEST (ConstraintEnclosedByParentheses , MissingParenthesisRight)

Definition at line 12 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.2 TEST (ConstraintEnclosedByParentheses , MissingParenthesisLeft)

Definition at line 16 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.3 TEST (ConstraintEnclosedByParentheses , ExtraParenthesisLeft)

Definition at line 20 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.248

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ConstraintTest.hpp File

8.247.1.4 TEST (ConstraintEnclosedByParentheses , ExtraParenthesisRight)

1205

Definition at line 24 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.5 TEST (ConstraintEnclosedByParentheses , InvertedParentheses)

Definition at line 28 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.6 TEST (ConstraintEnclosedByParentheses , ExtraParenthesesBothSides)

Definition at line 32 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.7 TEST (ConstraintEnclosedByParentheses , ParenthesesInWrongOrder)

Definition at line 36 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.8 TEST (ConstraintEnclosedByParentheses , Correct)

Definition at line 40 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.9 TEST (ConstraintEnclosedByParentheses , CorrectDoubled)

Definition at line 44 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.247.1.10 TEST (ConstraintEnclosedByParentheses , CorrectQuadrupled)

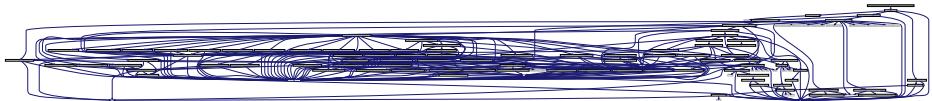
Definition at line 48 of file ConstraintParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.248 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ConstraintTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
```

graph for ConstraintTest.hpp:



Functions

- [TEST \(Constraint, ExtraInputBeforeConstraint\)](#)
- [TEST \(Constraint, ExtraInputAfterConstraint\)](#)
- [TEST \(Constraint, Correct\)](#)

8.248.1 Function Documentation

8.248.1.1 TEST (Constraint , ExtraInputBeforeConstraint)

Definition at line 12 of file ConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.248.1.2 TEST (Constraint , ExtraInputAfterConstraint)

Definition at line 16 of file ConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

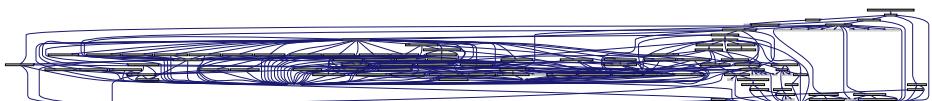
8.248.1.3 TEST (Constraint , Correct)

Definition at line 20 of file ConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.249 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/DifferenceTest.hpp File Reference

#include "parsing/InputStreamParser.hpp" Include dependency
graph for DifferenceTest.hpp:



- [TEST \(Difference, IncorrectInputMissingDifferenceNumericMeasure\)](#)
- [TEST \(Difference, IncorrectInputMissingDifferenceNumericMeasureAndBrackets\)](#)
- [TEST \(Difference, IncorrectDifferenceSymbol\)](#)
- [TEST \(Difference, IncorrectOpeningBracket\)](#)
- [TEST \(Difference, IncorrectClosingBracket\)](#)
- [TEST \(Difference, IncorrectBrackets\)](#)
- [TEST \(Difference, IncorrectBracketsInverted\)](#)
- [TEST \(Difference, IncorrectBracketsDoubled\)](#)
- [TEST \(Difference, Correct\)](#)

8.249.1 Function Documentation

8.249.1.1 TEST (Difference , IncorrectInputMissingDifferenceNumericMeasure)

Definition at line 12 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.2 TEST (Difference , IncorrectInputMissingDifferenceNumericMeasureAndBrackets)

Definition at line 16 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.3 TEST (Difference , IncorrectDifferenceSymbol)

Definition at line 20 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.4 TEST (Difference , IncorrectOpeningBracket)

Definition at line 24 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.5 TEST (Difference , IncorrectClosingBracket)

Definition at line 28 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.6 TEST (Difference , IncorrectBrackets)

Definition at line 32 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.7 TEST (Difference , IncorrectBracketsInverted)

Definition at line 36 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.249.1.8 TEST (Difference , IncorrectBracketsDoubled)

Definition at line 40 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

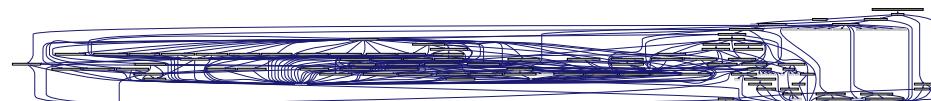
8.249.1.9 TEST (Difference , Correct)

Definition at line 44 of file DifferenceTest.hpp.

References multiscaletest::verification::parseInputString().

8.250 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FilterNumericMeasureTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
graph for FilterNumericMeasureTest.hpp:
```



Functions

- [TEST](#) (FilterSubset, IncorrectAlternative)
- [TEST](#) (FilterSubset, CorrectSpatialMeasureRealValue)
- [TEST](#) (FilterSubset, CorrectSpatialMeasures)
- [TEST](#) (FilterSubset, CorrectMultiple)
- [TEST](#) (FilterSubset, CorrectMultipleComplex)

8.251

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FilterSubsetTest.hpp File

8.250.1 Function Documentation

1209

8.250.1.1 TEST (FilterSubset , IncorrectAlternative)

Definition at line 12 of file FilterNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.250.1.2 TEST (FilterSubset , CorrectSpatialMeasureRealValue)

Definition at line 16 of file FilterNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.250.1.3 TEST (FilterSubset , CorrectSpatialMeasures)

Definition at line 20 of file FilterNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.250.1.4 TEST (FilterSubset , CorrectMultiple)

Definition at line 24 of file FilterNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

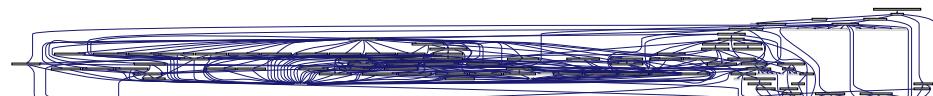
8.250.1.5 TEST (FilterSubset , CorrectMultipleComplex)

Definition at line 28 of file FilterNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.251 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FilterSubsetTest.hpp File Reference

#include "parsing/InputStreamParser.hpp" Include dependency graph for FilterSubsetTest.hpp:



Functions

- [TEST \(FilterSubset, IncorrectInputMisspelledFilter\)](#)

- [TEST](#) (FilterSubset, IncorrectInputBeforeStartBracket)
- [TEST](#) (FilterSubset, IncorrectInputAfterStartBracket)
- [TEST](#) (FilterSubset, IncorrectInputMissingComma)
- [TEST](#) (FilterSubset, IncorrectInputBeforeEndBracket)
- [TEST](#) (FilterSubset, IncorrectInputAfterEndBracket)
- [TEST](#) (FilterSubset, Correct)

8.251.1 Function Documentation

8.251.1.1 TEST (FilterSubset , IncorrectInputMisspelledFilter)

Definition at line 12 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.251.1.2 TEST (FilterSubset , IncorrectInputBeforeStartBracket)

Definition at line 16 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.251.1.3 TEST (FilterSubset , IncorrectInputAfterStartBracket)

Definition at line 20 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.251.1.4 TEST (FilterSubset , IncorrectInputMissingComma)

Definition at line 24 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.251.1.5 TEST (FilterSubset , IncorrectInputBeforeEndBracket)

Definition at line 28 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.251.1.6 TEST (FilterSubset , IncorrectInputAfterEndBracket)

Definition at line 32 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.252

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FutureLogicPropertyTest.hpp File

8.251.1.7 TEST (FilterSubset , Correct)

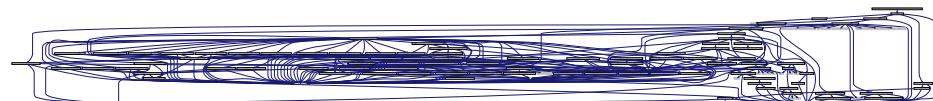
1211

Definition at line 36 of file FilterSubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.252 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/FutureLogicPropertyTest.hpp File - Reference

#include "parsing/InputStreamParser.hpp" Include dependency graph for FutureLogicPropertyTest.hpp:



Functions

- [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 \(FutureLogicProperty, Correct\)](#)

8.252.1 Function Documentation

8.252.1.1 TEST (FutureLogicProperty , WrongInputMissingStartTimepoint)

Definition at line 12 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.2 TEST (FutureLogicProperty , WrongInputMissingEndTimepoint)

Definition at line 16 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.3 TEST (FutureLogicProperty , WrongInputMissingTimepoints)

Definition at line 20 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.4 TEST (FutureLogicProperty , WrongInputMissingTimepointsAndBrackets)

Definition at line 24 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.5 TEST (FutureLogicProperty , WrongInputBeforeStartParanthesis)

Definition at line 28 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.6 TEST (FutureLogicProperty , WrongInputAfterStartParanthesis)

Definition at line 32 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.7 TEST (FutureLogicProperty , MissingTimepointComma)

Definition at line 36 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.8 TEST (FutureLogicProperty , InvalidStartTimepoint)

Definition at line 40 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.9 TEST (FutureLogicProperty , InvalidEndTimepoint)

Definition at line 44 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/GlobalLogicPropertyTest.hpp File

8.252.1.10 TEST (FutureLogicProperty , InvalidTimepoints)

1213

Definition at line 48 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.11 TEST (FutureLogicProperty , WrongInputBeforeEndParanthesis)

Definition at line 52 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.252.1.12 TEST (FutureLogicProperty , WrongInputAfterEndParanthesis)

Definition at line 56 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

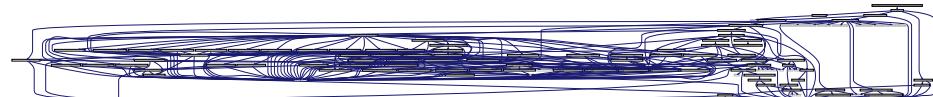
8.252.1.13 TEST (FutureLogicProperty , Correct)

Definition at line 60 of file FutureLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/GlobalLogicPropertyTest.hpp File -
Reference

#include "parsing/InputStringParser.hpp" Include dependency
graph for GlobalLogicPropertyTest.hpp:



Functions

- [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)

8.253.1 Function Documentation

8.253.1.1 TEST (GlobalLogicProperty , WrongInputMissingStartTimepoint)

Definition at line 12 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.2 TEST (GlobalLogicProperty , WrongInputMissingEndTimepoint)

Definition at line 16 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.3 TEST (GlobalLogicProperty , WrongInputMissingTimepoints)

Definition at line 20 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.4 TEST (GlobalLogicProperty , WrongInputMissingTimepointsAndBrackets)

Definition at line 24 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.5 TEST (GlobalLogicProperty , WrongInputBeforeStartParanthesis)

Definition at line 28 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.6 TEST (GlobalLogicProperty , WrongInputAfterStartParanthesis)

Definition at line 32 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/GlobalLogicPropertyTest.hpp File

8.253.1.7 TEST (GlobalLogicProperty , MissingTimepointComma)

1215

Definition at line 36 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.8 TEST (GlobalLogicProperty , InvalidStartTimepoint)

Definition at line 40 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.9 TEST (GlobalLogicProperty , InvalidEndTimepoint)

Definition at line 44 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.10 TEST (GlobalLogicProperty , InvalidTimepoints)

Definition at line 48 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.11 TEST (GlobalLogicProperty , WrongInputBeforeEndParanthesis)

Definition at line 52 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.12 TEST (GlobalLogicProperty , WrongInputAfterEndParanthesis)

Definition at line 56 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.253.1.13 TEST (GlobalLogicProperty , Correct)

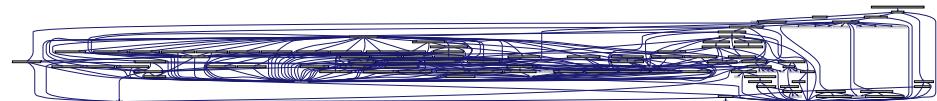
Definition at line 60 of file GlobalLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.254 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/LogicPropertyParenthesesTest.hpp

File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
graph for LogicPropertyParenthesesTest.hpp:
```



Functions

- [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`)

8.254.1 Function Documentation

8.254.1.1 TEST (`LogicPropertyEnclosedByParentheses` , `MissingParenthesisRight`)

Definition at line 12 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.2 TEST (`LogicPropertyEnclosedByParentheses` , `MissingParenthesisLeft`)

Definition at line 16 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.3 TEST (`LogicPropertyEnclosedByParentheses` , `ExtraParenthesisLeft`)

Definition at line 20 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.255

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/LogicPropertyTest.hpp File
8.254.1.4 TEST (LogicPropertyEnclosedByParentheses , ExtraParenthesisRight) **1217**

Definition at line 24 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.5 TEST (LogicPropertyEnclosedByParentheses , InvertedParentheses)

Definition at line 28 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.6 TEST (LogicPropertyEnclosedByParentheses , ExtraParenthesesBothSides)

Definition at line 32 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.7 TEST (LogicPropertyEnclosedByParentheses , ParenthesesInWrongOrder)

Definition at line 36 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.8 TEST (LogicPropertyEnclosedByParentheses , Correct)

Definition at line 40 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.9 TEST (LogicPropertyEnclosedByParentheses , CorrectDoubled)

Definition at line 44 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.254.1.10 TEST (LogicPropertyEnclosedByParentheses , CorrectQuadrupled)

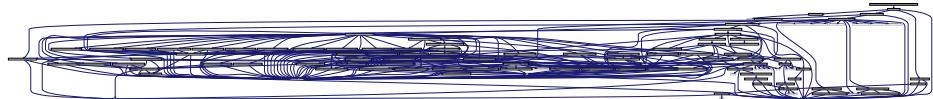
Definition at line 48 of file LogicPropertyParenthesesTest.hpp.

References multiscaletest::verification::parseInputString().

8.255 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/LogicPropertyTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
```

graph for LogicPropertyTest.hpp:



Functions

- [TEST \(LogicProperty, ExtralnputBeforeLogicProperty\)](#)
- [TEST \(LogicProperty, ExtralnputInsideLogicProperty\)](#)
- [TEST \(LogicProperty, ExtralnputAfterLogicProperty\)](#)
- [TEST \(LogicProperty, Correct\)](#)

8.255.1 Function Documentation

8.255.1.1 TEST (LogicProperty , ExtralnputBeforeLogicProperty)

Definition at line 12 of file LogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.255.1.2 TEST (LogicProperty , ExtralnputInsideLogicProperty)

Definition at line 16 of file LogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.255.1.3 TEST (LogicProperty , ExtralnputAfterLogicProperty)

Definition at line 20 of file LogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.255.1.4 TEST (LogicProperty , Correct)

Definition at line 24 of file LogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

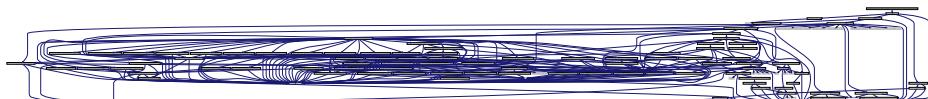
8.256 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/MultipleLogicPropertiesTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
```

8.257

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextKLogicPropertyTest.hpp File
graph for MultipleLogicPropertiesTest.hpp:
Reference

1219



Functions

- [TEST](#) (MultipleLogicProperties, Correct1)
- [TEST](#) (MultipleLogicProperties, Correct2)

8.256.1 Function Documentation

8.256.1.1 TEST (MultipleLogicProperties , Correct1)

Definition at line 12 of file MultipleLogicPropertiesTest.hpp.

References multiscaletest::verification::parseInputString().

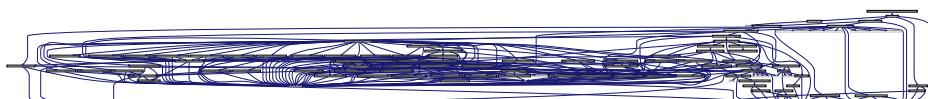
8.256.1.2 TEST (MultipleLogicProperties , Correct2)

Definition at line 16 of file MultipleLogicPropertiesTest.hpp.

References multiscaletest::verification::parseInputString().

8.257 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextKLogicPropertyTest.hpp File - Reference

#include "parsing/InputStreamParser.hpp" Include dependency
graph for NextKLogicPropertyTest.hpp:



Functions

- [TEST](#) (NextKLogicProperty, IncorrectInputMissingTimepoint)
- [TEST](#) (NextKLogicProperty, IncorrectInputAfterNextSymbol)
- [TEST](#) (NextKLogicProperty, IncorrectValueForNextTimepoints)

- [TEST](#) (NextKLogicProperty, RealValueForNextTimepoints)
- [TEST](#) (NextKLogicProperty, IncorrectInputBeforeLogicProperty)
- [TEST](#) (NextKLogicProperty, Correct)

8.257.1 Function Documentation

8.257.1.1 TEST (NextKLogicProperty , IncorrectInputMissingTimepoint)

Definition at line 12 of file NextKLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.257.1.2 TEST (NextKLogicProperty , IncorrectInputAfterNextSymbol)

Definition at line 16 of file NextKLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.257.1.3 TEST (NextKLogicProperty , IncorrectValueForNextTimepoints)

Definition at line 20 of file NextKLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.257.1.4 TEST (NextKLogicProperty , RealValueForNextTimepoints)

Definition at line 24 of file NextKLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.257.1.5 TEST (NextKLogicProperty , IncorrectInputBeforeLogicProperty)

Definition at line 28 of file NextKLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.257.1.6 TEST (NextKLogicProperty , Correct)

Definition at line 32 of file NextKLogicPropertyTest.hpp.

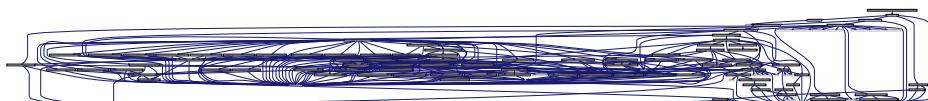
References multiscaletest::verification::parseInputString().

8.258

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NextLogicPropertyTest.hpp File Reference

temporal/test/attribute/NextLogicPropertyTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for NextLogicPropertyTest.hpp:



Functions

- **TEST** (NextLogicProperty, IncorrectInputAfterNextSymbol)
- **TEST** (NextLogicProperty, Correct)

8.258.1 Function Documentation

8.258.1.1 TEST (NextLogicProperty , IncorrectInputAfterNextSymbol)

Definition at line 12 of file NextLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

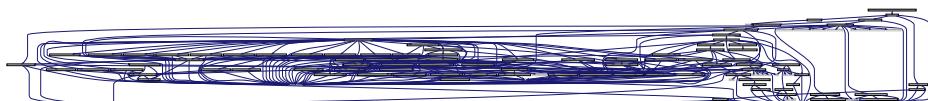
8.258.1.2 TEST (NextLogicProperty , Correct)

Definition at line 16 of file NextLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.259 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotConstraintTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for NotConstraintTest.hpp:



Functions

- [TEST \(NotConstraint, IncorrectOperator\)](#)
- [TEST \(NotConstraint, OperatorAfterConstraint\)](#)
- [TEST \(NotConstraint, OperatorAfterConstraintAndExtraConstraint\)](#)
- [TEST \(NotConstraint, OperatorBeforeConstraintAndExtraConstraint\)](#)
- [TEST \(NotConstraint, Correct\)](#)

8.259.1 Function Documentation

8.259.1.1 TEST (NotConstraint , IncorrectOperator)

Definition at line 12 of file NotConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.259.1.2 TEST (NotConstraint , OperatorAfterConstraint)

Definition at line 16 of file NotConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.259.1.3 TEST (NotConstraint , OperatorAfterConstraintAndExtraConstraint)

Definition at line 20 of file NotConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.259.1.4 TEST (NotConstraint , OperatorBeforeConstraintAndExtraConstraint)

Definition at line 24 of file NotConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.259.1.5 TEST (NotConstraint , Correct)

Definition at line 28 of file NotConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.260 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NotLogicPropertyTest.hpp File - Reference

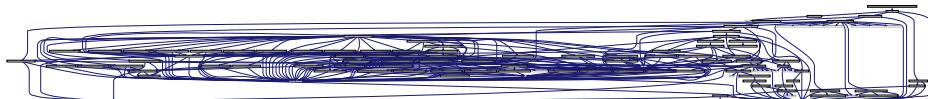
```
#include "parsing/InputStringParser.hpp"  Include dependency
```

8.261

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericMeasureTest.hpp File
Graph for NotLogicPropertyTest.hpp:

Reference

1223



Functions

- [TEST](#) (NotLogicProperty, OperatorAfterLogicProperty)
- [TEST](#) (NotLogicProperty, OperatorAfterLogicPropertyAndExtraLogicProperty)
- [TEST](#) (NotLogicProperty, OperatorBeforeLogicPropertyAndExtraLogicProperty)
- [TEST](#) (NotLogicProperty, Correct)

8.260.1 Function Documentation

8.260.1.1 TEST (NotLogicProperty , OperatorAfterLogicProperty)

Definition at line 12 of file NotLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.260.1.2 TEST (NotLogicProperty , OperatorAfterLogicPropertyAndExtraLogicProperty)

Definition at line 16 of file NotLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.260.1.3 TEST (NotLogicProperty , OperatorBeforeLogicPropertyAndExtraLogicProperty)

Definition at line 20 of file NotLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.260.1.4 TEST (NotLogicProperty , Correct)

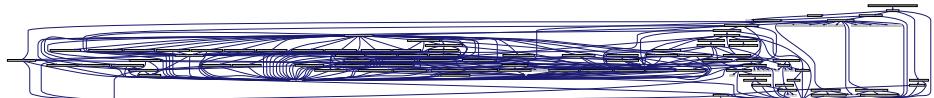
Definition at line 24 of file NotLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.261 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericMeasureTest.hpp File - Reference

```
#include "parsing/InputStringParser.hpp"    Include dependency
```

graph for NumericMeasureTest.hpp:



Functions

- [TEST](#) (NumericMeasure, WrongAlternative)
- [TEST](#) (NumericMeasure, Correct)

8.261.1 Function Documentation

8.261.1.1 TEST (NumericMeasure , WrongAlternative)

Definition at line 12 of file NumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.261.1.2 TEST (NumericMeasure , Correct)

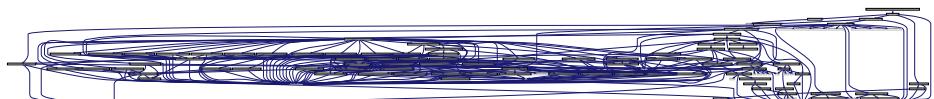
Definition at line 16 of file NumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.262 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericNumericComparisonTest.hpp

File Reference

#include "parsing/InputStreamParser.hpp" Include dependency
graph for NumericNumericComparisonTest.hpp:



Functions

- [TEST](#) (NumericNumericComparison, NumericMeasureFirst1)
- [TEST](#) (NumericNumericComparison, NumericMeasureFirst2)
- [TEST](#) (NumericNumericComparison, ComparatorFirst1)

8.262

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericNumericComparisonTest.hpp File
Reference [TEST](#) (NumericNumericComparison, ComparatorFirst2)

1225

- [TEST](#) (NumericNumericComparison, IncorrectOrder)
- [TEST](#) (NumericNumericComparison, Correct)

8.262.1 Function Documentation

8.262.1.1 TEST (NumericNumericComparison , NumericMeasureFirst1)

Definition at line 12 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.262.1.2 TEST (NumericNumericComparison , NumericMeasureFirst2)

Definition at line 16 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.262.1.3 TEST (NumericNumericComparison , ComparatorFirst1)

Definition at line 20 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.262.1.4 TEST (NumericNumericComparison , ComparatorFirst2)

Definition at line 24 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.262.1.5 TEST (NumericNumericComparison , IncorrectOrder)

Definition at line 28 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

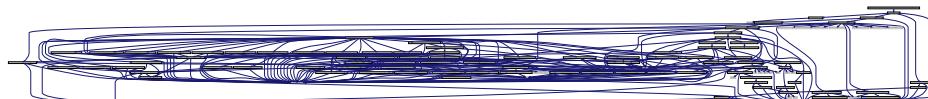
8.262.1.6 TEST (NumericNumericComparison , Correct)

Definition at line 32 of file NumericNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.263 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialMeasureTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for NumericSpatialMeasureTest.hpp:



Functions

- [TEST](#) (NumericSpatialMeasure, IncorrectAlternative)
- [TEST](#) (NumericSpatialMeasure, Correct)

8.263.1 Function Documentation

8.263.1.1 TEST (NumericSpatialMeasure , IncorrectAlternative)

Definition at line 12 of file NumericSpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

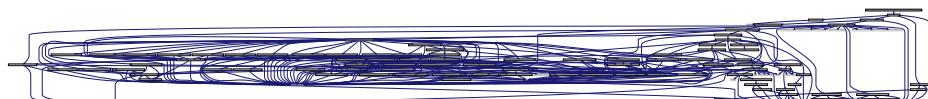
8.263.1.2 TEST (NumericSpatialMeasure , Correct)

Definition at line 16 of file NumericSpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.264 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericSpatialNumericComparisonTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for NumericSpatialNumericComparisonTest.hpp:



- [TEST](#) (NumericSpatialNumericComparison, NumericMeasureFirst1)
- [TEST](#) (NumericSpatialNumericComparison, NumericMeasureFirst2)
- [TEST](#) (NumericSpatialNumericComparison, ComparatorFirst1)
- [TEST](#) (NumericSpatialNumericComparison, ComparatorFirst2)
- [TEST](#) (NumericSpatialNumericComparison, IncorrectOrder)
- [TEST](#) (NumericSpatialNumericComparison, Correct)

8.264.1 Function Documentation

8.264.1.1 TEST (NumericSpatialNumericComparison , NumericMeasureFirst1)

Definition at line 12 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.264.1.2 TEST (NumericSpatialNumericComparison , NumericMeasureFirst2)

Definition at line 16 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.264.1.3 TEST (NumericSpatialNumericComparison , ComparatorFirst1)

Definition at line 20 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.264.1.4 TEST (NumericSpatialNumericComparison , ComparatorFirst2)

Definition at line 24 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.264.1.5 TEST (NumericSpatialNumericComparison , IncorrectOrder)

Definition at line 28 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

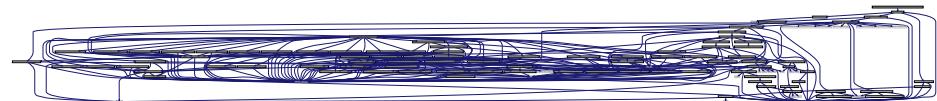
8.264.1.6 TEST (NumericSpatialNumericComparison , Correct)

Definition at line 32 of file NumericSpatialNumericComparisonTest.hpp.

References multiscaletest::verification::parseInputString().

8.265 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericStateVariableTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for NumericStateVariableTest.hpp:



Functions

- [TEST \(NumericStateVariable, MissingLeftCurlyBrace\)](#)
- [TEST \(NumericStateVariable, MissingRightCurlyBrace\)](#)
- [TEST \(NumericStateVariable, ExtraLeftCurlyBrace\)](#)
- [TEST \(NumericStateVariable, ExtraRightCurlyBrace\)](#)
- [TEST \(NumericStateVariable, InvertedCurlyBraces\)](#)
- [TEST \(NumericStateVariable, DoubleCurlyBraces\)](#)
- [TEST \(NumericStateVariable, TripleCurlyBraces\)](#)
- [TEST \(NumericStateVariable, Correct1\)](#)
- [TEST \(NumericStateVariable, Correct2\)](#)
- [TEST \(NumericStateVariable, Correct3\)](#)

8.265.1 Function Documentation

8.265.1.1 TEST (NumericStateVariable , MissingLeftCurlyBrace)

Definition at line 12 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.2 TEST (NumericStateVariable , MissingRightCurlyBrace)

Definition at line 16 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.3 TEST (NumericStateVariable , ExtraLeftCurlyBrace)

Definition at line 20 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/NumericStateVariableTest.hpp File

8.265.1.4 TEST (NumericStateVariable , ExtraRightCurlyBrace)

1229

Definition at line 24 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.5 TEST (NumericStateVariable , InvertedCurlyBraces)

Definition at line 28 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.6 TEST (NumericStateVariable , DoubleCurlyBraces)

Definition at line 32 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.7 TEST (NumericStateVariable , TripleCurlyBraces)

Definition at line 36 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.8 TEST (NumericStateVariable , Correct1)

Definition at line 40 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.9 TEST (NumericStateVariable , Correct2)

Definition at line 44 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.265.1.10 TEST (NumericStateVariable , Correct3)

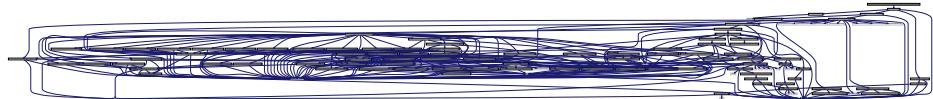
Definition at line 48 of file NumericStateVariableTest.hpp.

References multiscaletest::verification::parseInputString().

8.266 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ProbabilisticLogicPropertyTest.hpp

File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
graph for ProbabilisticLogicPropertyTest.hpp:
```



Functions

- [TEST \(ProbabilisticLogicProperty, IncorrectProbabilitySymbol\)](#)
- [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 \(ProbabilisticLogicProperty, ProbabilityHigh\)](#)

8.266.1 Function Documentation

8.266.1.1 TEST (ProbabilisticLogicProperty , IncorrectProbabilitySymbol)

Definition at line 12 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/ProbabilisticLogicPropertyTest.hpp File

8.266.1.2 TEST (ProbabilisticLogicProperty , IncorrectComparator)

1231

Definition at line 16 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.3 TEST (ProbabilisticLogicProperty , IncorrectEqualComparator)

Definition at line 20 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.4 TEST (ProbabilisticLogicProperty , InvalidProbabilityValueTooLow)

Definition at line 24 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.5 TEST (ProbabilisticLogicProperty , InvalidProbabilityValueTooLowMinor)

Definition at line 28 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.6 TEST (ProbabilisticLogicProperty , InvalidProbabilityValueTooHigh)

Definition at line 32 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.7 TEST (ProbabilisticLogicProperty , InvalidProbabilityValueTooHighMinor)

Definition at line 36 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.8 TEST (ProbabilisticLogicProperty , IncorrectLogicProperty)

Definition at line 40 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.9 TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftMissing)

Definition at line 44 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.10 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesRightMissing)**

Definition at line 48 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.11 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftExtra)**

Definition at line 52 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.12 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesRightExtra)**

Definition at line 56 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.13 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesLeftRightExtra)**

Definition at line 60 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.14 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesInverted)**

Definition at line 64 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.15 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParanthesesClosing)**

Definition at line 68 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.16 **TEST (ProbabilisticLogicProperty , IncorrectlyEnclosingParantheses)**

Definition at line 72 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.17 **TEST (ProbabilisticLogicProperty , Correct)**

Definition at line 76 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.267

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetMeasureTest.hpp File
8.266.1.18 TEST (ProbabilisticLogicProperty , ProbabilityMin)

1233

Definition at line 80 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.19 TEST (ProbabilisticLogicProperty , ProbabilityMax)

Definition at line 84 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.266.1.20 TEST (ProbabilisticLogicProperty , ProbabilityLow)

Definition at line 88 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

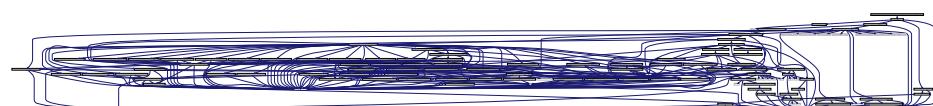
8.266.1.21 TEST (ProbabilisticLogicProperty , ProbabilityHigh)

Definition at line 92 of file ProbabilisticLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.267 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetMeasureTest.hpp
File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for QuaternarySubsetMeasureTest.hpp:



Functions

- [TEST](#) (QuaternarySubsetMeasure, IncorrectQuaternarySubsetMeasure)
- [TEST](#) (QuaternarySubset, CorrectCovar)

8.267.1 Function Documentation

8.267.1.1 TEST (QuaternarySubsetMeasure , IncorrectQuaternarySubsetMeasure)

Definition at line 12 of file QuaternarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

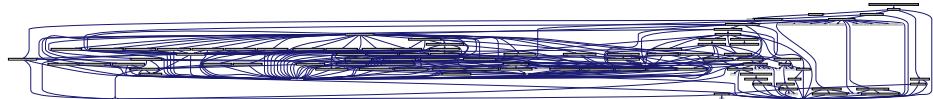
8.267.1.2 TEST (QuaternarySubset , CorrectCovar)

Definition at line 16 of file QuaternarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.268 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for QuaternarySubsetTest.hpp:



Functions

- [TEST](#) (QuaternarySubset, IncorrectInputMissingParameterOne)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParameterTwo)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParameterThree)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParameterFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneThree)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersTwoThree)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersTwoFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersThreeFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneTwoThree)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneTwoFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersTwoThreeFour)
- [TEST](#) (QuaternarySubset, IncorrectInputMissingParametersOneTwoThreeFour)
- [TEST](#) (QuaternarySubset, IncorrectInputBeforeStartBracket)
- [TEST](#) (QuaternarySubset, IncorrectInputAfterStartBracket)
- [TEST](#) (QuaternarySubset, MissingFirstComma)
- [TEST](#) (QuaternarySubset, MissingSecondComma)
- [TEST](#) (QuaternarySubset, MissingThirdComma)

8.268

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/QuaternarySubsetTest.hpp File

Reference [TEST](#) (QuaternarySubset, IncorrectInputBeforeEndBracket)

1235

- [TEST](#) (QuaternarySubset, IncorrectInputAfterEndBracket)
- [TEST](#) (QuaternarySubset, Correct)

8.268.1 Function Documentation

8.268.1.1 TEST (QuaternarySubset , IncorrectInputMissingParameterOne)

Definition at line 12 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.2 TEST (QuaternarySubset , IncorrectInputMissingParameterTwo)

Definition at line 16 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.3 TEST (QuaternarySubset , IncorrectInputMissingParameterThree)

Definition at line 20 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.4 TEST (QuaternarySubset , IncorrectInputMissingParameterFour)

Definition at line 24 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.5 TEST (QuaternarySubset , IncorrectInputMissingParametersOneTwo)

Definition at line 28 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.6 TEST (QuaternarySubset , IncorrectInputMissingParametersOneThree)

Definition at line 32 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.7 TEST (QuaternarySubset , IncorrectInputMissingParametersOneFour)

Definition at line 36 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.8 TEST (QuaternarySubset , IncorrectInputMissingParametersTwoThree)

Definition at line 40 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.9 TEST (QuaternarySubset , IncorrectInputMissingParametersTwoFour)

Definition at line 44 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.10 TEST (QuaternarySubset , IncorrectInputMissingParametersThreeFour)

Definition at line 48 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.11 TEST (QuaternarySubset , IncorrectInputMissingParametersOneTwoThree)

Definition at line 52 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.12 TEST (QuaternarySubset , IncorrectInputMissingParametersOneTwoFour)

Definition at line 56 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.13 TEST (QuaternarySubset , IncorrectInputMissingParametersTwoThreeFour)

Definition at line 60 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.14 TEST (QuaternarySubset , IncorrectInputMissingParametersOneTwoThreeFour)

Definition at line 64 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.15 TEST (QuaternarySubset , IncorrectInputBeforeStartBracket)

Definition at line 68 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.269

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/attribute/SpatialMeasureTest.hpp File

8.268.1.16 TEST (QuaternarySubset , IncorrectInputAfterStartBracket)

1237

Definition at line 72 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.17 TEST (QuaternarySubset , MissingFirstComma)

Definition at line 76 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.18 TEST (QuaternarySubset , MissingSecondComma)

Definition at line 80 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.19 TEST (QuaternarySubset , MissingThirdComma)

Definition at line 84 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.20 TEST (QuaternarySubset , IncorrectInputBeforeEndBracket)

Definition at line 88 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.21 TEST (QuaternarySubset , IncorrectInputAfterEndBracket)

Definition at line 92 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.268.1.22 TEST (QuaternarySubset , Correct)

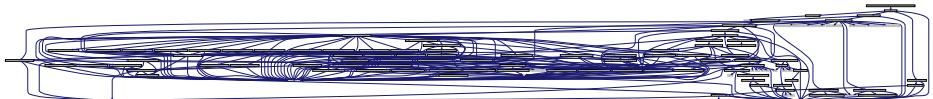
Definition at line 96 of file QuaternarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.269 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SpatialMeasureTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
```

graph for SpatialMeasureTest.hpp:



Functions

- [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 \(SpatialMeasure, CorrectCentroidY\)](#)

8.269.1 Function Documentation

8.269.1.1 TEST (SpatialMeasure , IncorrectSpatialMeasure)

Definition at line 13 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.2 TEST (SpatialMeasure , CorrectClusteredness)

Definition at line 17 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.3 TEST (SpatialMeasure , CorrectDensity)

Definition at line 21 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.4 TEST (SpatialMeasure , CorrectArea)

Definition at line 25 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SpatialMeasureTest.hpp File

8.269.1.5 TEST (SpatialMeasure , CorrectPerimeter)

1239

Definition at line 29 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.6 TEST (SpatialMeasure , CorrectDistanceFromOrigin)

Definition at line 33 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.7 TEST (SpatialMeasure , CorrectAngle)

Definition at line 37 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.8 TEST (SpatialMeasure , CorrectTriangleMeasure)

Definition at line 41 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.9 TEST (SpatialMeasure , CorrectRectangleMeasure)

Definition at line 45 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.10 TEST (SpatialMeasure , CorrectCircleMeasure)

Definition at line 49 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.269.1.11 TEST (SpatialMeasure , CorrectCentroidX)

Definition at line 53 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

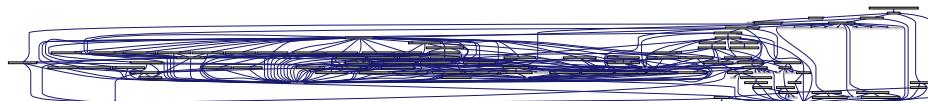
8.269.1.12 TEST (SpatialMeasure , CorrectCentroidY)

Definition at line 57 of file SpatialMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.270 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetOperationTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for SubsetOperationTest.hpp:



Functions

- [TEST](#) (SubsetOperation, IncorrectInputWrongSubsetOperationAlternative)
- [TEST](#) (SubsetOperation, CorrectDifference)
- [TEST](#) (SubsetOperation, CorrectIntersection)
- [TEST](#) (SubsetOperation, CorrectUnion)

8.270.1 Function Documentation

8.270.1.1 TEST (SubsetOperation , IncorrectInputWrongSubsetOperationAlternative)

Definition at line 12 of file SubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.270.1.2 TEST (SubsetOperation , CorrectDifference)

Definition at line 16 of file SubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.270.1.3 TEST (SubsetOperation , CorrectIntersection)

Definition at line 20 of file SubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.270.1.4 TEST (SubsetOperation , CorrectUnion)

Definition at line 24 of file SubsetOperationTest.hpp.

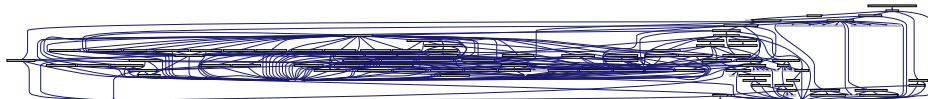
References multiscaletest::verification::parseInputString().

8.271

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSpecificTest.hpp File
8.271 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSpecificTest.hpp File Reference [24]

temporal/test/attribute/SubsetSpecificTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
graph for SubsetSpecificTest.hpp:
```



Functions

- [TEST](#) (SubsetSpecific, IncorrectInputWrongSubsetAlternative)
- [TEST](#) (SubsetSpecific, CorrectClusters)
- [TEST](#) (SubsetSpecific, CorrectRegions)

8.271.1 Function Documentation

8.271.1.1 TEST (SubsetSpecific , IncorrectInputWrongSubsetAlternative)

Definition at line 12 of file SubsetSpecificTest.hpp.

References multiscaletest::verification::parseInputString().

8.271.1.2 TEST (SubsetSpecific , CorrectClusters)

Definition at line 16 of file SubsetSpecificTest.hpp.

References multiscaletest::verification::parseInputString().

8.271.1.3 TEST (SubsetSpecific , CorrectRegions)

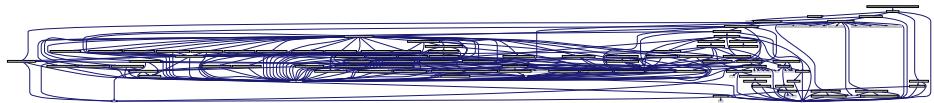
Definition at line 20 of file SubsetSpecificTest.hpp.

References multiscaletest::verification::parseInputString().

8.272 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetSubsetOperationTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
```

graph for SubsetSubsetOperationTest.hpp:



Functions

- [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 \(SubsetSubsetOperation, Correct\)](#)

8.272.1 Function Documentation

8.272.1.1 TEST (SubsetSubsetOperation , IncorrectInputWrongAlternative)

Definition at line 12 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.2 TEST (SubsetSubsetOperation , IncorrectInputBeforeStartParanthesis)

Definition at line 16 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.3 TEST (SubsetSubsetOperation , IncorrectInputAfterStartParanthesis)

Definition at line 20 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.4 TEST (SubsetSubsetOperation , IncorrectInputMissingFirstArgument)

Definition at line 24 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

**8.273 /home/ovidiu/Repositories/git/multiscale-/
Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetTest.hpp**
File Reference **1243**
8.272.1.5 TEST (SubsetSubsetOperation , IncorrectInputMissingSeparatorComma)

Definition at line 28 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.6 TEST (SubsetSubsetOperation , IncorrectInputMissingCommaAndArgument)

Definition at line 32 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.7 TEST (SubsetSubsetOperation , IncorrectInputMissingSecondArgument)

Definition at line 36 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.8 TEST (SubsetSubsetOperation , IncorrectInputBeforeEndParanthesis)

Definition at line 40 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.272.1.9 TEST (SubsetSubsetOperation , IncorrectInputAfterEndParanthesis)

Definition at line 44 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

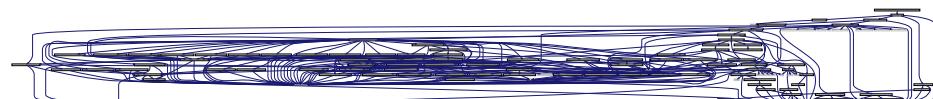
8.272.1.10 TEST (SubsetSubsetOperation , Correct)

Definition at line 48 of file SubsetSubsetOperationTest.hpp.

References multiscaletest::verification::parseInputString().

8.273 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/SubsetTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency
graph for SubsetTest.hpp:



Functions

- [TEST](#) (Subset, IncorrectInputWrongSubsetAlternativeRegion)
- [TEST](#) (Subset, IncorrectInputWrongSubsetAlternativeCluster)
- [TEST](#) (Subset, Correct)

8.273.1 Function Documentation

8.273.1.1 TEST (Subset , IncorrectInputWrongSubsetAlternativeRegion)

Definition at line 12 of file SubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.273.1.2 TEST (Subset , IncorrectInputWrongSubsetAlternativeCluster)

Definition at line 16 of file SubsetTest.hpp.

References multiscaletest::verification::parseInputString().

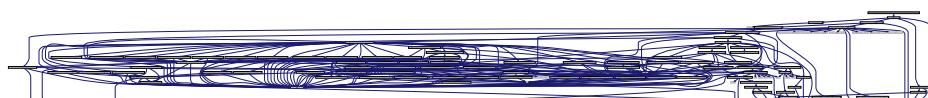
8.273.1.3 TEST (Subset , Correct)

Definition at line 20 of file SubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.274 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetMeasureTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"    Include dependency
graph for TernarySubsetMeasureTest.hpp:
```



Functions

- [TEST](#) (TernarySubsetMeasure, IncorrectTernarySubsetMeasure)
- [TEST](#) (TernarySubsetMeasure, CorrectPercentile)
- [TEST](#) (TernarySubsetMeasure, CorrectQuartile)

8.275

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetTest.hpp File

8.274.1 Function Documentation

1245

8.274.1.1 TEST (TernarySubsetMeasure , IncorrectTernarySubsetMeasure)

Definition at line 12 of file TernarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.274.1.2 TEST (TernarySubsetMeasure , CorrectPercentile)

Definition at line 16 of file TernarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

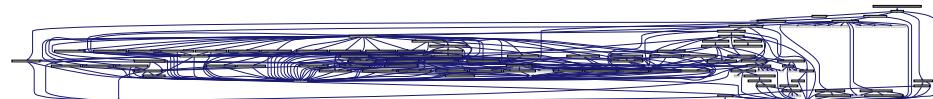
8.274.1.3 TEST (TernarySubsetMeasure , CorrectQuartile)

Definition at line 20 of file TernarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.275 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency
graph for TernarySubsetTest.hpp:
```



Functions

- [TEST](#) (TernarySubset, IncorrectInputMissingParameterOne)
- [TEST](#) (TernarySubset, IncorrectInputMissingParameterTwo)
- [TEST](#) (TernarySubset, IncorrectInputMissingParameterThree)
- [TEST](#) (TernarySubset, IncorrectInputMissingParametersOneTwo)
- [TEST](#) (TernarySubset, IncorrectInputMissingParametersOneThree)
- [TEST](#) (TernarySubset, IncorrectInputMissingParametersTwoThree)
- [TEST](#) (TernarySubset, IncorrectInputMissingAllParameters)
- [TEST](#) (TernarySubset, IncorrectInputBeforeStartBracket)
- [TEST](#) (TernarySubset, IncorrectInputAfterStartBracket)
- [TEST](#) (TernarySubset, MissingFirstComma)
- [TEST](#) (TernarySubset, InvalidSpatialMeasure)
- [TEST](#) (TernarySubset, MissingSecondComma)

- [TEST](#) (TernarySubset, IncorrectInputBeforeEndBracket)
- [TEST](#) (TernarySubset, IncorrectInputAfterEndBracket)
- [TEST](#) (TernarySubset, Correct)

8.275.1 Function Documentation

8.275.1.1 TEST (TernarySubset , IncorrectInputMissingParameterOne)

Definition at line 12 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.2 TEST (TernarySubset , IncorrectInputMissingParameterTwo)

Definition at line 16 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.3 TEST (TernarySubset , IncorrectInputMissingParameterThree)

Definition at line 20 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.4 TEST (TernarySubset , IncorrectInputMissingParametersOneTwo)

Definition at line 24 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.5 TEST (TernarySubset , IncorrectInputMissingParametersOneThree)

Definition at line 28 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.6 TEST (TernarySubset , IncorrectInputMissingParametersTwoThree)

Definition at line 32 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.7 TEST (TernarySubset , IncorrectInputMissingAllParameters)

Definition at line 36 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/TernarySubsetTest.hpp File

8.275.1.8 TEST (TernarySubset , IncorrectInputBeforeStartBracket)

1247

Definition at line 40 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.9 TEST (TernarySubset , IncorrectInputAfterStartBracket)

Definition at line 44 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.10 TEST (TernarySubset , MissingFirstComma)

Definition at line 48 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.11 TEST (TernarySubset , InvalidSpatialMeasure)

Definition at line 52 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.12 TEST (TernarySubset , MissingSecondComma)

Definition at line 56 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.13 TEST (TernarySubset , IncorrectInputBeforeEndBracket)

Definition at line 60 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.275.1.14 TEST (TernarySubset , IncorrectInputAfterEndBracket)

Definition at line 64 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

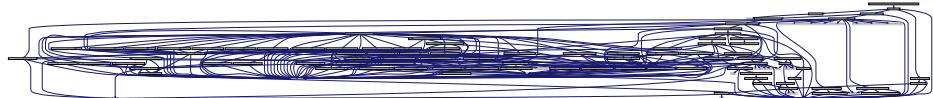
8.275.1.15 TEST (TernarySubset , Correct)

Definition at line 68 of file TernarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.276 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericFilterTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for UnaryNumericFilterTest.hpp:



Functions

- [TEST \(UnaryNumericFilter, IncorrectInputMissingParameter\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputAfterStartBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(UnaryNumericFilter, IncorrectInputAfterEndBracket\)](#)
- [TEST \(UnaryNumericFilter, Correct\)](#)

8.276.1 Function Documentation

8.276.1.1 TEST (UnaryNumericFilter , IncorrectInputMissingParameter)

Definition at line 12 of file UnaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.276.1.2 TEST (UnaryNumericFilter , IncorrectInputBeforeStartBracket)

Definition at line 16 of file UnaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.276.1.3 TEST (UnaryNumericFilter , IncorrectInputAfterStartBracket)

Definition at line 20 of file UnaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.276.1.4 TEST (UnaryNumericFilter , IncorrectInputBeforeEndBracket)

Definition at line 24 of file UnaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.277

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericMeasureTest.hpp File

8.276.1.5 TEST (UnaryNumericFilter , IncorrectInputAfterEndBracket)

1249

Definition at line 28 of file UnaryNumericFilterTest.hpp.

References multiscaletest::verification::parseInputString().

8.276.1.6 TEST (UnaryNumericFilter , Correct)

Definition at line 32 of file UnaryNumericFilterTest.hpp.

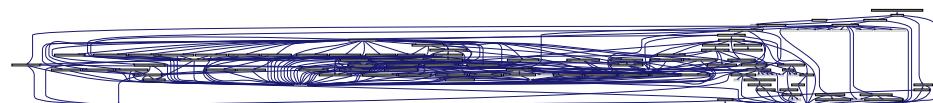
References multiscaletest::verification::parseInputString().

8.277 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/attribute/UnaryNumericMeasureTest.hpp File

Reference

```
#include "parsing/InputStreamParser.hpp"  Include dependency
graph for UnaryNumericMeasureTest.hpp:
```



Functions

- [TEST](#) (UnaryNumericMeasure, IncorrectUnaryNumericMeasure)
- [TEST](#) (UnaryNumericMeasure, CorrectAbs)
- [TEST](#) (UnaryNumericMeasure, CorrectCeil)
- [TEST](#) (UnaryNumericMeasure, CorrectFloor)
- [TEST](#) (UnaryNumericMeasure, CorrectRound)
- [TEST](#) (UnaryNumericMeasure, CorrectSign)
- [TEST](#) (UnaryNumericMeasure, CorrectSqrt)
- [TEST](#) (UnaryNumericMeasure, CorrectTrunc)

8.277.1 Function Documentation

8.277.1.1 TEST (UnaryNumericMeasure , IncorrectUnaryNumericMeasure)

Definition at line 12 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.2 TEST (UnaryNumericMeasure , CorrectAbs)

Definition at line 16 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.3 TEST (UnaryNumericMeasure , CorrectCeil)

Definition at line 20 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.4 TEST (UnaryNumericMeasure , CorrectFloor)

Definition at line 24 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.5 TEST (UnaryNumericMeasure , CorrectRound)

Definition at line 28 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.6 TEST (UnaryNumericMeasure , CorrectSign)

Definition at line 32 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.7 TEST (UnaryNumericMeasure , CorrectSqrt)

Definition at line 36 of file UnaryNumericMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.277.1.8 TEST (UnaryNumericMeasure , CorrectTrunc)

Definition at line 40 of file UnaryNumericMeasureTest.hpp.

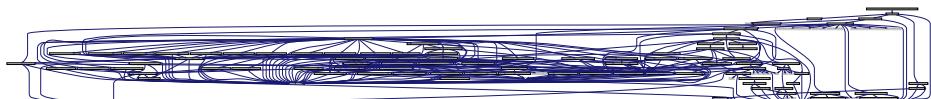
References multiscaletest::verification::parseInputString().

8.278

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryNumericNumericTest.hpp File
8.278 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-
temporal/test/attribute/UnaryNumericNumericTest.hpp File
Reference 1251

temporal/test/attribute/UnaryNumericNumericTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency
graph for UnaryNumericNumericTest.hpp:



Functions

- [TEST](#) (UnaryNumericNumeric, IncorrectInputMissingParameter)
- [TEST](#) (UnaryNumericNumeric, IncorrectInputBeforeStartBracket)
- [TEST](#) (UnaryNumericNumeric, IncorrectInputAfterStartBracket)
- [TEST](#) (UnaryNumericNumeric, IncorrectInputBeforeEndBracket)
- [TEST](#) (UnaryNumericNumeric, IncorrectInputAfterEndBracket)
- [TEST](#) (UnaryNumericNumeric, Correct)

8.278.1 Function Documentation

8.278.1.1 TEST (UnaryNumericNumeric , IncorrectInputMissingParameter)

Definition at line 12 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.278.1.2 TEST (UnaryNumericNumeric , IncorrectInputBeforeStartBracket)

Definition at line 16 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.278.1.3 TEST (UnaryNumericNumeric , IncorrectInputAfterStartBracket)

Definition at line 20 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.278.1.4 TEST (UnaryNumericNumeric , IncorrectInputBeforeEndBracket)

Definition at line 24 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.278.1.5 TEST (UnaryNumericNumeric , IncorrectInputAfterEndBracket)

Definition at line 28 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

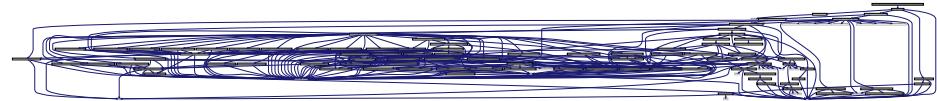
8.278.1.6 TEST (UnaryNumericNumeric , Correct)

Definition at line 32 of file UnaryNumericNumericTest.hpp.

References multiscaletest::verification::parseInputString().

8.279 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySpatialConstraintTest.hpp File Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for UnarySpatialConstraintTest.hpp:



Functions

- [TEST](#) (UnarySpatialConstraint, IncorrectSpatialMeasureBeforeConstraint)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingSpatialMeasure)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingComparator)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingNumericMeasure)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingComparatorNumericMeasure)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingSpatialMeasureNumericMeasure)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputMissingSpatialMeasureComparator)
- [TEST](#) (UnarySpatialConstraint, IncorrectInputEmptyConstraint)
- [TEST](#) (UnarySpatialConstraint, Correct)

8.279.1 Function Documentation

8.279.1.1 TEST (UnarySpatialConstraint , IncorrectSpatialMeasureBeforeConstraint)

Definition at line 12 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySpatialConstraintTest.hpp File
8.279.1.2 TEST (UnarySpatialConstraint , IncorrectInputMissingSpatialMeasure) **1253**

Definition at line 16 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.3 TEST (UnarySpatialConstraint , IncorrectInputMissingComparator)

Definition at line 20 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.4 TEST (UnarySpatialConstraint , IncorrectInputMissingNumericMeasure)

Definition at line 24 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.5 TEST (UnarySpatialConstraint , IncorrectInputMissingComparatorNumericMeasure)

Definition at line 28 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.6 TEST (UnarySpatialConstraint , IncorrectInputMissingSpatialMeasureNumericMeasure)

Definition at line 32 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.7 TEST (UnarySpatialConstraint , IncorrectInputMissingSpatialMeasureComparator)

Definition at line 36 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.279.1.8 TEST (UnarySpatialConstraint , IncorrectInputEmptyConstraint)

Definition at line 40 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

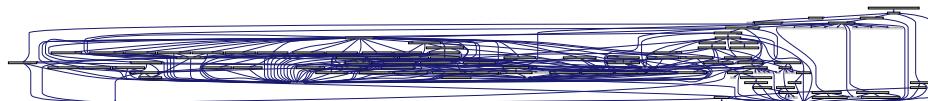
8.279.1.9 TEST (UnarySpatialConstraint , Correct)

Definition at line 44 of file UnarySpatialConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.280 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetMeasureTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for UnarySubsetMeasureTest.hpp:



Functions

- [TEST](#) (UnarySubsetMeasure, IncorrectUnarySubsetMeasure)
- [TEST](#) (UnarySubsetMeasure, CorrectCount)
- [TEST](#) (UnarySubsetMeasure, CorrectClusteredness)
- [TEST](#) (UnarySubsetMeasure, CorrectDensity)

8.280.1 Function Documentation

8.280.1.1 TEST (UnarySubsetMeasure , IncorrectUnarySubsetMeasure)

Definition at line 12 of file UnarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.280.1.2 TEST (UnarySubsetMeasure , CorrectCount)

Definition at line 16 of file UnarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.280.1.3 TEST (UnarySubsetMeasure , CorrectClusteredness)

Definition at line 20 of file UnarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.280.1.4 TEST (UnarySubsetMeasure , CorrectDensity)

Definition at line 24 of file UnarySubsetMeasureTest.hpp.

References multiscaletest::verification::parseInputString().

8.281

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnarySubsetTest.hpp File

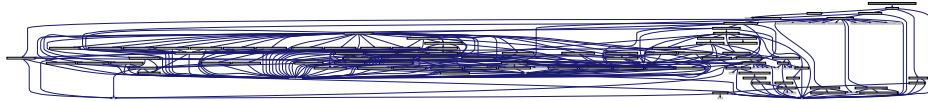
8.281 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/attribute/UnarySubsetTest.hpp File Reference

1255

temporal/test/attribute/UnarySubsetTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp"  Include dependency  
graph for UnarySubsetTest.hpp:
```



Functions

- [TEST \(UnarySubset, IncorrectInputNoSubset\)](#)
- [TEST \(UnarySubset, IncorrectInputBeforeStartBracket\)](#)
- [TEST \(UnarySubset, IncorrectInputAfterStartBracket\)](#)
- [TEST \(UnarySubset, IncorrectInputBeforeEndBracket\)](#)
- [TEST \(UnarySubset, IncorrectInputAfterEndBracket\)](#)
- [TEST \(UnarySubset, Correct\)](#)

8.281.1 Function Documentation

8.281.1.1 TEST (UnarySubset , IncorrectInputNoSubset)

Definition at line 12 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.281.1.2 TEST (UnarySubset , IncorrectInputBeforeStartBracket)

Definition at line 16 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.281.1.3 TEST (UnarySubset , IncorrectInputAfterStartBracket)

Definition at line 20 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.281.1.4 TEST (UnarySubset , IncorrectInputBeforeEndBracket)

Definition at line 24 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.281.1.5 TEST (UnarySubset , IncorrectInputAfterEndBracket)

Definition at line 28 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

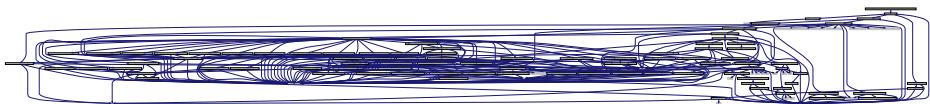
8.281.1.6 TEST (UnarySubset , Correct)

Definition at line 32 of file UnarySubsetTest.hpp.

References multiscaletest::verification::parseInputString().

8.282 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UnaryTypeConstraintTest.hpp File - Reference

#include "parsing/InputStringParser.hpp" Include dependency graph for UnaryTypeConstraintTest.hpp:



Functions

- [TEST \(UnaryTypeConstraint, IncorrectInputWrongTypeKeywordExtraLetterAfter\)](#)
- [TEST \(UnaryTypeConstraint, IncorrectInputWrongTypeKeywordExtraLetterBefore\)](#)
- [TEST \(UnaryTypeConstraint, IncorrectInputBeforeTypeKeyword\)](#)
- [TEST \(UnaryTypeConstraint, IncorrectInputAfterTypeKeyword\)](#)
- [TEST \(UnaryTypeConstraint, IncorrectInputAfterComparator\)](#)
- [TEST \(UnaryTypeConstraint, IncorrectInputAfterFilterNumericMeasure\)](#)
- [TEST \(UnaryTypeConstraint, Correct\)](#)

8.282.1 Function Documentation

8.282.1.1 TEST (UnaryTypeConstraint , IncorrectInputWrongTypeKeywordExtraLetterAfter)

Definition at line 12 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.283

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UntilLogicPropertyTest.hpp File
8.282.1.2 TEST (UnaryTypeConstraint , IncorrectInputWrongTypeKeywordExtraLetterBefore)
Reference 1257

Definition at line 16 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.282.1.3 TEST (UnaryTypeConstraint , IncorrectInputBeforeTypeKeyword)

Definition at line 20 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.282.1.4 TEST (UnaryTypeConstraint , IncorrectInputAfterTypeKeyword)

Definition at line 24 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.282.1.5 TEST (UnaryTypeConstraint , IncorrectInputAfterComparator)

Definition at line 28 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.282.1.6 TEST (UnaryTypeConstraint , IncorrectInputAfterFilterNumericMeasure)

Definition at line 32 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.282.1.7 TEST (UnaryTypeConstraint , Correct)

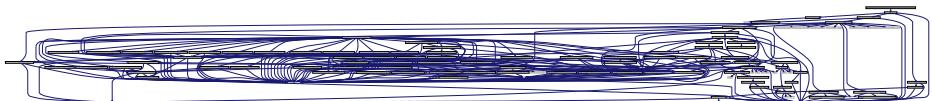
Definition at line 36 of file UnaryTypeConstraintTest.hpp.

References multiscaletest::verification::parseInputString().

8.283 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UntilLogicPropertyTest.hpp File - Reference

```
#include "parsing/InputStringParser.hpp"    Include dependency
```

graph for UntilLogicPropertyTest.hpp:



Functions

- [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 \(UntilLogicProperty, MultipleCorrect\)](#)

8.283.1 Function Documentation

8.283.1.1 TEST (UntilLogicProperty , IncorrectInputMissingStartTimepoint)

Definition at line 12 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.2 TEST (UntilLogicProperty , IncorrectInputMissingEndTimepoint)

Definition at line 16 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/attribute/UntilLogicPropertyTest.hpp File

8.283.1.3 TEST (UntilLogicProperty , IncorrectInputMissingTimepoints)

1259

Definition at line 20 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.4 TEST (UntilLogicProperty , IncorrectInputMissingTimepointsAndBrackets)

Definition at line 24 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.5 TEST (UntilLogicProperty , UntilOperatorAsUnaryBefore)

Definition at line 28 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.6 TEST (UntilLogicProperty , UntilOperatorAsUnaryAfter)

Definition at line 32 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.7 TEST (UntilLogicProperty , IncorrectInputBeforeUntilOperator)

Definition at line 36 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.8 TEST (UntilLogicProperty , AdditionalOperatorBeforeUntilOperator)

Definition at line 40 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.9 TEST (UntilLogicProperty , IncorrectInputAfterUntilOperator)

Definition at line 44 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.10 TEST (UntilLogicProperty , AdditionalOperatorAfterUntilOperator)

Definition at line 48 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.11 TEST (UntilLogicProperty , WrongInputBeforeStartParenthesis)

Definition at line 52 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.12 TEST (UntilLogicProperty , WrongInputAfterStartParenthesis)

Definition at line 56 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.13 TEST (UntilLogicProperty , MissingTimepointsComma)

Definition at line 60 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.14 TEST (UntilLogicProperty , StartTimepointInvalid)

Definition at line 64 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.15 TEST (UntilLogicProperty , StartTimepointRealNumber)

Definition at line 68 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.16 TEST (UntilLogicProperty , EndTimepointInvalid)

Definition at line 72 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.17 TEST (UntilLogicProperty , EndTimepointRealNumber)

Definition at line 76 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.18 TEST (UntilLogicProperty , TimepointsInvalid)

Definition at line 80 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.284

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

8.283.1.19 TEST (UntilLogicProperty , TimepointsRealNumber)

1261

Definition at line 84 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.20 TEST (UntilLogicProperty , WrongInputBeforeEndParenthesis)

Definition at line 88 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.21 TEST (UntilLogicProperty , WrongInputAfterEndParenthesis)

Definition at line 92 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.283.1.22 TEST (UntilLogicProperty , Correct)

Definition at line 96 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

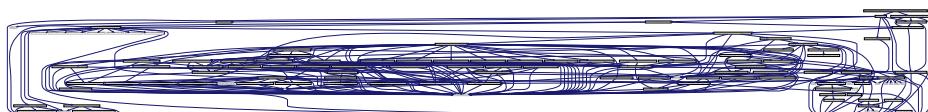
8.283.1.23 TEST (UntilLogicProperty , MultipleCorrect)

Definition at line 100 of file UntilLogicPropertyTest.hpp.

References multiscaletest::verification::parseInputString().

8.284 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateBayesianModelChecker-Test.hpp File Reference

```
#include "ModelCheckerTest.hpp" #include "multiscale/verification/spatial-temporal/ApproximateBayesianModelChecker.hpp" Include dependency graph for ApproximateBayesianModelCheckerTest.hpp:
```



Classes

- class [multiscaletest::ApproximateBayesianModelCheckerTest](#)

Class for testing the approximate Bayesian model checker.

Namespaces

- namespace [multiscaletest](#)

Functions

- [TEST_F \(ApproximateBayesianModelCheckerTest, CaseTrue\)](#)
- [TEST_F \(ApproximateBayesianModelCheckerTest, CaseFalse\)](#)

8.284.1 Function Documentation

8.284.1.1 TEST_F (ApproximateBayesianModelCheckerTest , CaseTrue)

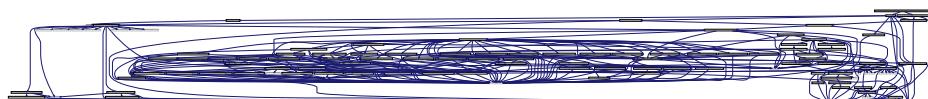
Definition at line 85 of file ApproximateBayesianModelCheckerTest.hpp.

8.284.1.2 TEST_F (ApproximateBayesianModelCheckerTest , CaseFalse)

Definition at line 93 of file ApproximateBayesianModelCheckerTest.hpp.

8.285 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ApproximateProbabilisticModel-CheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp" #include "multiscale/verification/spatial-temporal/test/checking/ApproximateProbabilisticModelChecker.hpp" Include dependency graph for ApproximateProbabilisticModelCheckerTest.hpp:
```



Classes

- class [multiscaletest::ApproximateProbabilisticModelCheckerTest](#)

Class for testing the approximate probabilistic model checker.

8.286

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

Namespaces

Reference 1263

- namespace [multiscaletest](#)

Functions

- [TEST_F \(ApproximateProbabilisticModelCheckerTest, CaseFalse\)](#)

8.285.1 Function Documentation

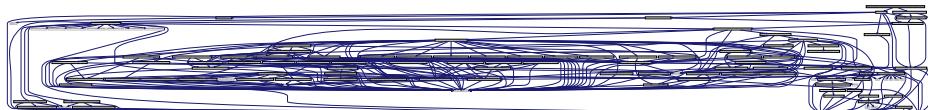
8.285.1.1 TEST_F (ApproximateProbabilisticModelCheckerTest , CaseFalse)

Definition at line 78 of file ApproximateProbabilisticModelCheckerTest.hpp.

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

Reference

```
#include "ModelCheckerTest.hpp" #include "multiscale/verification/spatial-temporal/...
BayesianModelChecker.hpp" Include dependency graph for BayesianModel-
CheckerTest.hpp:
```



Classes

- class [multiscaletest::BayesianModelCheckerTest](#)

Class for testing the Bayesian model checker.

Namespaces

- namespace [multiscaletest](#)

Functions

- [TEST_F \(BayesianModelCheckerTest, CaseTrue\)](#)
- [TEST_F \(BayesianModelCheckerTest, CaseFalse\)](#)

8.286.1 Function Documentation

8.286.1.1 TEST_F(BayesianModelCheckerTest , CaseTrue)

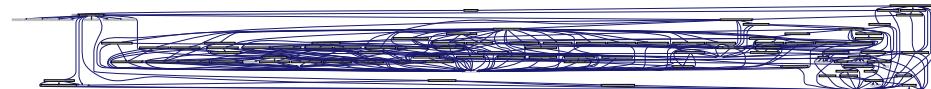
Definition at line 85 of file BayesianModelCheckerTest.hpp.

8.286.1.2 TEST_F(BayesianModelCheckerTest , CaseFalse)

Definition at line 93 of file BayesianModelCheckerTest.hpp.

8.287 /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/
ModelChecker.hpp" #include "multiscale/verification/spatial-temporal/model/
Region.hpp" #include "multiscale/verification/spatial-temporal/model/-
SpatialTemporalTrace.hpp" #include "multiscale/verification/spatial-temporal/
Parser.hpp" #include <string> Include dependency graph for Model-
CheckerTest.hpp:
```



Classes

- class [multiscaletest::ModelCheckerTest](#)
Class for testing model checkers.

Namespaces

- namespace [multiscaletest](#)

Variables

- const std::string [INPUT_LOGIC_PROPERTY](#) = "P > 0.6 [F [0, 3] (avg(regions, area) > 20)]"

8.287.1 Variable Documentation

8.288

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckingTest.cpp File
Reference 6 const std::string INPUT_LOGIC_PROPERTY = "P > 0.6 [F [0, 3] (avg(region \$1265
area) > 20)]"

Definition at line 15 of file ModelCheckerTest.hpp.

Referenced by multiscaletest::ModelCheckerTest::InitialiseAbstractSyntaxTree().

8.288 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ModelCheckingTest.cpp File Reference

```
#include "ApproximateBayesianModelCheckerTest.hpp" #include
#include "ApproximateProbabilisticModelCheckerTest.hpp"    #include
#include "BayesianModelCheckerTest.hpp"      #include "Probabilistic-
#include "BlackBoxModelCheckerTest.hpp"  #include "StatisticalModel-
#include "ModelCheckerTest.hpp" Include dependency graph for ModelCheckingTest.cpp:
```



Functions

- int [main](#) (int argc, char **argv)

8.288.1 Function Documentation

8.288.1.1 int main (int argc, char ** argv)

Definition at line 9 of file ModelCheckingTest.cpp.

8.289 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/ProbabilisticBlackBoxModelChecker-Test.hpp File Reference

```
#include "ModelCheckerTest.hpp" #include "multiscale/verification/spatial-temporal/
#include "ProbabilisticBlackBoxModelChecker.hpp" Include dependency graph
for ProbabilisticBlackBoxModelCheckerTest.hpp:
```



Classes

- class [multiscaletest::ProbabilisticBlackBoxModelCheckerTest](#)

Class for testing the probabilistic black-box model checker.

Namespaces

- namespace [multiscaletest](#)

Functions

- [TEST_F \(ProbabilisticBlackBoxModelCheckerTest, CaseFalse\)](#)

8.289.1 Function Documentation

8.289.1.1 TEST_F (ProbabilisticBlackBoxModelCheckerTest , CaseFalse)

Definition at line 41 of file ProbabilisticBlackBoxModelCheckerTest.hpp.

8.290 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/checking/StatisticalModelCheckerTest.hpp File Reference

```
#include "ModelCheckerTest.hpp" #include "multiscale/verification/spatial-temporal/test/checking/StatisticalModelChecker.hpp" Include dependency graph for StatisticalModelCheckerTest.hpp:
```



Classes

- class [multiscaletest::StatisticalModelCheckerTest](#)

Class for testing the statistical model checker.

Namespaces

- namespace [multiscaletest](#)

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File
Functions Reference

1267

- [TEST_F \(StatisticalModelCheckerTest, CaseTrue\)](#)
- [TEST_F \(StatisticalModelCheckerTest, CaseFalse\)](#)

8.290.1 Function Documentation

8.290.1.1 TEST_F (StatisticalModelCheckerTest , CaseTrue)

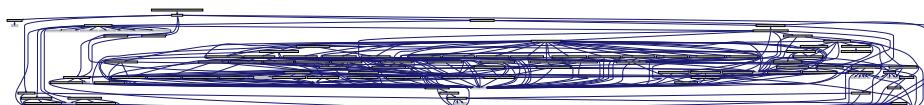
Definition at line 71 of file StatisticalModelCheckerTest.hpp.

8.290.1.2 TEST_F (StatisticalModelCheckerTest , CaseFalse)

Definition at line 78 of file StatisticalModelCheckerTest.hpp.

8.291 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File - Reference

#include "TraceEvaluationTest.hpp" Include dependency graph for -
CompleteTraceTest.hpp:



Classes

- class [multiscaletest::CompleteTraceTest](#)

Class for testing evaluation of complete traces containing both numeric state variables and spatial entities.

Namespaces

- namespace [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 \(CompleteTraceTest, BinarySubsetMeasureAvg\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureGeomean\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureHarmean\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureKurt\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureMax\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureMedian\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureMin\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureMode\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureProduct\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureSkew\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureStdev\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureSum\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubsetMeasureVar\)](#)
- [TEST_F \(CompleteTraceTest, BinarySubset\)](#)
- [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 \(CompleteTraceTest, Difference\)](#)
- [TEST_F \(CompleteTraceTest, FilterNumericMeasure\)](#)
- [TEST_F \(CompleteTraceTest, FilterSubset\)](#)
- [TEST_F \(CompleteTraceTest, FutureLogicProperty\)](#)
- [TEST_F \(CompleteTraceTest, GlobalLogicProperty\)](#)
- [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 \(CompleteTraceTest, NextLogicProperty\)](#)

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File

Reference TEST_F (CompleteTraceTest, NotConstraint)

1269

- TEST_F (CompleteTraceTest, NotLogicProperty)
- TEST_F (CompleteTraceTest, NumericMeasure)
- TEST_F (CompleteTraceTest, NumericNumericComparison)
- TEST_F (CompleteTraceTest, NumericSpatialMeasure)
- TEST_F (CompleteTraceTest, NumericSpatialNumericComparison)
- TEST_F (CompleteTraceTest, NumericStateVariable1)
- TEST_F (CompleteTraceTest, NumericStateVariable2)
- TEST_F (CompleteTraceTest, NumericStateVariable3)
- TEST_F (CompleteTraceTest, ProbabilisticLogicProperty)
- TEST_F (CompleteTraceTest, QuaternarySubsetCovar)
- TEST_F (CompleteTraceTest, QuaternarySubset)
- 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, SubsetOperationDifference)
- TEST_F (CompleteTraceTest, SubsetOperationIntersection)
- TEST_F (CompleteTraceTest, SubsetOperationUnion)
- TEST_F (CompleteTraceTest, SubsetSpecificClusters)
- TEST_F (CompleteTraceTest, SubsetSpecificRegions)
- TEST_F (CompleteTraceTest, SubsetSubsetOperation)
- TEST_F (CompleteTraceTest, Subset)
- TEST_F (CompleteTraceTest, TernarySubsetMeasurePercentile)
- TEST_F (CompleteTraceTest, TernarySubsetMeasureQuartile)
- TEST_F (CompleteTraceTest, TernarySubset)
- TEST_F (CompleteTraceTest, UnaryTypeConstraint)
- TEST_F (CompleteTraceTest, UnarySpatialConstraint)
- 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, UnarySubsetMeasureCount)
- TEST_F (CompleteTraceTest, UnarySubsetMeasureClusteredness)

- [TEST_F \(CompleteTraceTest, UnarySubsetMeasureDensity\)](#)
- [TEST_F \(CompleteTraceTest, UnarySubset\)](#)
- [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 \(CompleteTraceTest, GlobalConstantValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalConstantValueBinarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalConstantValueTernarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalConstantValueQuaternarySubset\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueReal\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueBinarySubset\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueTernarySubset\)](#)
- [TEST_F \(CompleteTraceTest, FutureIncreasingValueQuaternarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueBinarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueTernarySubset\)](#)
- [TEST_F \(CompleteTraceTest, GlobalDecreasingValueQuaternarySubset\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueNumericStateVariable2\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueBinarySubset\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueTernarySubset\)](#)
- [TEST_F \(CompleteTraceTest, IncreasingUntilDecreasingValueQuaternarySubset\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueNumericStateVariable\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueBinarySubset\)](#)

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File

Reference: [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueTernarySubset\)](#) 1271

- [TEST_F \(CompleteTraceTest, DecreasingUntilIncreasingValueQuaternarySubset\)](#)
- [TEST_F \(CompleteTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueUnarySubset\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueBinarySubset\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueTernarySubset\)](#)
- [TEST_F \(CompleteTraceTest, OscillationsValueQuaternarySubset\)](#)
- [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 \(CompleteTraceTest, ConstraintsCombinationNary\)](#)

8.291.1 Function Documentation

8.291.1.1 TEST_F (CompleteTraceTest , BinaryNumericFilter)

Definition at line 106 of file CompleteTraceTest.hpp.

8.291.1.2 TEST_F (CompleteTraceTest , BinaryNumericMeasureAdd)

Definition at line 119 of file CompleteTraceTest.hpp.

8.291.1.3 TEST_F (CompleteTraceTest , BinaryNumericMeasureDiv)

Definition at line 123 of file CompleteTraceTest.hpp.

8.291.1.4 TEST_F (CompleteTraceTest , BinaryNumericMeasureLog)

Definition at line 127 of file CompleteTraceTest.hpp.

8.291.1.5 TEST_F (CompleteTraceTest , BinaryNumericMeasureMod)

Definition at line 131 of file CompleteTraceTest.hpp.

8.291.1.6 TEST_F (CompleteTraceTest , BinaryNumericMeasureMultiply)

Definition at line 135 of file CompleteTraceTest.hpp.

8.291.1.7 TEST_F(CompleteTraceTest , BinaryNumericMeasurePower)

Definition at line 139 of file CompleteTraceTest.hpp.

8.291.1.8 TEST_F(CompleteTraceTest , BinaryNumericMeasureSubtract)

Definition at line 143 of file CompleteTraceTest.hpp.

8.291.1.9 TEST_F(CompleteTraceTest , BinaryNumericNumeric)

Definition at line 156 of file CompleteTraceTest.hpp.

8.291.1.10 TEST_F(CompleteTraceTest , BinarySubsetMeasureAvg)

Definition at line 169 of file CompleteTraceTest.hpp.

8.291.1.11 TEST_F(CompleteTraceTest , BinarySubsetMeasureGeomean)

Definition at line 173 of file CompleteTraceTest.hpp.

8.291.1.12 TEST_F(CompleteTraceTest , BinarySubsetMeasureHarmean)

Definition at line 177 of file CompleteTraceTest.hpp.

8.291.1.13 TEST_F(CompleteTraceTest , BinarySubsetMeasureKurt)

Definition at line 181 of file CompleteTraceTest.hpp.

8.291.1.14 TEST_F(CompleteTraceTest , BinarySubsetMeasureMax)

Definition at line 185 of file CompleteTraceTest.hpp.

8.291.1.15 TEST_F(CompleteTraceTest , BinarySubsetMeasureMedian)

Definition at line 189 of file CompleteTraceTest.hpp.

8.291.1.16 TEST_F(CompleteTraceTest , BinarySubsetMeasureMin)

Definition at line 193 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/CompleteTraceTest.hpp File

8.291.1.17 TEST_F(CompleteTraceTest , BinarySubsetMeasureMode)

1273

Definition at line 197 of file CompleteTraceTest.hpp.

8.291.1.18 TEST_F(CompleteTraceTest , BinarySubsetMeasureProduct)

Definition at line 201 of file CompleteTraceTest.hpp.

8.291.1.19 TEST_F(CompleteTraceTest , BinarySubsetMeasureSkew)

Definition at line 205 of file CompleteTraceTest.hpp.

8.291.1.20 TEST_F(CompleteTraceTest , BinarySubsetMeasureStdev)

Definition at line 209 of file CompleteTraceTest.hpp.

8.291.1.21 TEST_F(CompleteTraceTest , BinarySubsetMeasureSum)

Definition at line 213 of file CompleteTraceTest.hpp.

8.291.1.22 TEST_F(CompleteTraceTest , BinarySubsetMeasureVar)

Definition at line 217 of file CompleteTraceTest.hpp.

8.291.1.23 TEST_F(CompleteTraceTest , BinarySubset)

Definition at line 230 of file CompleteTraceTest.hpp.

8.291.1.24 TEST_F(CompleteTraceTest , ComparatorGreaterThan)

Definition at line 243 of file CompleteTraceTest.hpp.

8.291.1.25 TEST_F(CompleteTraceTest , ComparatorLessThan)

Definition at line 247 of file CompleteTraceTest.hpp.

8.291.1.26 TEST_F(CompleteTraceTest , ComparatorGreaterThanOrEqual)

Definition at line 251 of file CompleteTraceTest.hpp.

8.291.1.27 TEST_F(CompleteTraceTest , ComparatorLessThanOrEqual)

Definition at line 255 of file CompleteTraceTest.hpp.

8.291.1.28 TEST_F(CompleteTraceTest , ComparatorEqual)

Definition at line 259 of file CompleteTraceTest.hpp.

8.291.1.29 TEST_F(CompleteTraceTest , CompoundConstraint)

Definition at line 272 of file CompleteTraceTest.hpp.

8.291.1.30 TEST_F(CompleteTraceTest , CompoundConstraintMultiple)

Definition at line 279 of file CompleteTraceTest.hpp.

8.291.1.31 TEST_F(CompleteTraceTest , CompoundLogicProperty)

Definition at line 295 of file CompleteTraceTest.hpp.

8.291.1.32 TEST_F(CompleteTraceTest , CompoundLogicPropertyMultiple)

Definition at line 302 of file CompleteTraceTest.hpp.

8.291.1.33 TEST_F(CompleteTraceTest , ConstraintEnclosedByParentheses)

Definition at line 318 of file CompleteTraceTest.hpp.

8.291.1.34 TEST_F(CompleteTraceTest , ConstraintEnclosedByParenthesesDoubled)

Definition at line 322 of file CompleteTraceTest.hpp.

8.291.1.35 TEST_F(CompleteTraceTest , ConstraintEnclosedByParenthesesQuadrupled)

Definition at line 326 of file CompleteTraceTest.hpp.

8.291.1.36 TEST_F(CompleteTraceTest , Constraint)

Definition at line 339 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File

8.291.1.37 TEST_F(CompleteTraceTest , Difference)

1275

Definition at line 352 of file CompleteTraceTest.hpp.

8.291.1.38 TEST_F(CompleteTraceTest , FilterNumericMeasure)

Definition at line 365 of file CompleteTraceTest.hpp.

8.291.1.39 TEST_F(CompleteTraceTest , FilterSubset)

Definition at line 378 of file CompleteTraceTest.hpp.

8.291.1.40 TEST_F(CompleteTraceTest , FutureLogicProperty)

Definition at line 391 of file CompleteTraceTest.hpp.

8.291.1.41 TEST_F(CompleteTraceTest , GlobalLogicProperty)

Definition at line 404 of file CompleteTraceTest.hpp.

8.291.1.42 TEST_F(CompleteTraceTest , LogicPropertyEnclosedByParentheses)

Definition at line 417 of file CompleteTraceTest.hpp.

8.291.1.43 TEST_F(CompleteTraceTest , LogicPropertyEnclosedByParenthesesDoubled)

Definition at line 421 of file CompleteTraceTest.hpp.

8.291.1.44 TEST_F(CompleteTraceTest , LogicPropertyEnclosedByParentheses-Quadrupled)

Definition at line 425 of file CompleteTraceTest.hpp.

8.291.1.45 TEST_F(CompleteTraceTest , LogicProperty)

Definition at line 438 of file CompleteTraceTest.hpp.

8.291.1.46 TEST_F(CompleteTraceTest , MultipleLogicProperties1)

Definition at line 451 of file CompleteTraceTest.hpp.

8.291.1.47 TEST_F(CompleteTraceTest , MultipleLogicProperties2)

Definition at line 455 of file CompleteTraceTest.hpp.

8.291.1.48 TEST_F(CompleteTraceTest , NextKLogicProperty)

Definition at line 468 of file CompleteTraceTest.hpp.

8.291.1.49 TEST_F(CompleteTraceTest , NextLogicProperty)

Definition at line 481 of file CompleteTraceTest.hpp.

8.291.1.50 TEST_F(CompleteTraceTest , NotConstraint)

Definition at line 494 of file CompleteTraceTest.hpp.

8.291.1.51 TEST_F(CompleteTraceTest , NotLogicProperty)

Definition at line 507 of file CompleteTraceTest.hpp.

8.291.1.52 TEST_F(CompleteTraceTest , NumericMeasure)

Definition at line 520 of file CompleteTraceTest.hpp.

8.291.1.53 TEST_F(CompleteTraceTest , NumericNumericComparison)

Definition at line 533 of file CompleteTraceTest.hpp.

8.291.1.54 TEST_F(CompleteTraceTest , NumericSpatialMeasure)

Definition at line 546 of file CompleteTraceTest.hpp.

8.291.1.55 TEST_F(CompleteTraceTest , NumericSpatialNumericComparison)

Definition at line 559 of file CompleteTraceTest.hpp.

8.291.1.56 TEST_F(CompleteTraceTest , NumericStateVariable1)

Definition at line 572 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File

8.291.1.57 TEST_F(CompleteTraceTest , NumericStateVariable2)

1277

Definition at line 576 of file CompleteTraceTest.hpp.

8.291.1.58 TEST_F(CompleteTraceTest , NumericStateVariable3)

Definition at line 580 of file CompleteTraceTest.hpp.

8.291.1.59 TEST_F(CompleteTraceTest , ProbabilisticLogicProperty)

Definition at line 593 of file CompleteTraceTest.hpp.

8.291.1.60 TEST_F(CompleteTraceTest , QuaternarySubsetCovar)

Definition at line 606 of file CompleteTraceTest.hpp.

8.291.1.61 TEST_F(CompleteTraceTest , QuaternarySubset)

Definition at line 619 of file CompleteTraceTest.hpp.

8.291.1.62 TEST_F(CompleteTraceTest , SpatialMeasureClusteredness)

Definition at line 632 of file CompleteTraceTest.hpp.

8.291.1.63 TEST_F(CompleteTraceTest , SpatialMeasureDensity)

Definition at line 636 of file CompleteTraceTest.hpp.

8.291.1.64 TEST_F(CompleteTraceTest , SpatialMeasureArea)

Definition at line 640 of file CompleteTraceTest.hpp.

8.291.1.65 TEST_F(CompleteTraceTest , SpatialMeasurePerimeter)

Definition at line 644 of file CompleteTraceTest.hpp.

8.291.1.66 TEST_F(CompleteTraceTest , SpatialMeasureDistanceFromOrigin)

Definition at line 648 of file CompleteTraceTest.hpp.

8.291.1.67 TEST_F(CompleteTraceTest , SpatialMeasureAngle)

Definition at line 652 of file CompleteTraceTest.hpp.

8.291.1.68 TEST_F(CompleteTraceTest , SpatialMeasureTriangleMeasure)

Definition at line 656 of file CompleteTraceTest.hpp.

8.291.1.69 TEST_F(CompleteTraceTest , SpatialMeasureRectangleMeasure)

Definition at line 660 of file CompleteTraceTest.hpp.

8.291.1.70 TEST_F(CompleteTraceTest , SpatialMeasureCircleMeasure)

Definition at line 664 of file CompleteTraceTest.hpp.

8.291.1.71 TEST_F(CompleteTraceTest , SpatialMeasureCentroidX)

Definition at line 668 of file CompleteTraceTest.hpp.

8.291.1.72 TEST_F(CompleteTraceTest , SpatialMeasureCentroidY)

Definition at line 672 of file CompleteTraceTest.hpp.

8.291.1.73 TEST_F(CompleteTraceTest , SubsetOperationDifference)

Definition at line 685 of file CompleteTraceTest.hpp.

8.291.1.74 TEST_F(CompleteTraceTest , SubsetOperationIntersection)

Definition at line 689 of file CompleteTraceTest.hpp.

8.291.1.75 TEST_F(CompleteTraceTest , SubsetOperationUnion)

Definition at line 693 of file CompleteTraceTest.hpp.

8.291.1.76 TEST_F(CompleteTraceTest , SubsetSpecificClusters)

Definition at line 706 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File
8.291.1.77 **TEST_F(CompleteTraceTest , SubsetSpecificRegions)** **1279**

Definition at line 710 of file CompleteTraceTest.hpp.

8.291.1.78 TEST_F(CompleteTraceTest , SubsetSubsetOperation)

Definition at line 723 of file CompleteTraceTest.hpp.

8.291.1.79 TEST_F(CompleteTraceTest , Subset)

Definition at line 736 of file CompleteTraceTest.hpp.

8.291.1.80 TEST_F(CompleteTraceTest , TernarySubsetMeasurePercentile)

Definition at line 749 of file CompleteTraceTest.hpp.

8.291.1.81 TEST_F(CompleteTraceTest , TernarySubsetMeasureQuartile)

Definition at line 753 of file CompleteTraceTest.hpp.

8.291.1.82 TEST_F(CompleteTraceTest , TernarySubset)

Definition at line 766 of file CompleteTraceTest.hpp.

8.291.1.83 TEST_F(CompleteTraceTest , UnaryTypeConstraint)

Definition at line 779 of file CompleteTraceTest.hpp.

8.291.1.84 TEST_F(CompleteTraceTest , UnarySpatialConstraint)

Definition at line 792 of file CompleteTraceTest.hpp.

8.291.1.85 TEST_F(CompleteTraceTest , UnaryNumericFilter)

Definition at line 805 of file CompleteTraceTest.hpp.

8.291.1.86 TEST_F(CompleteTraceTest , UnaryNumericMeasureAbs)

Definition at line 818 of file CompleteTraceTest.hpp.

8.291.1.87 **TEST_F(CompleteTraceTest , UnaryNumericMeasureCeil)**

Definition at line 822 of file CompleteTraceTest.hpp.

8.291.1.88 **TEST_F(CompleteTraceTest , UnaryNumericMeasureFloor)**

Definition at line 826 of file CompleteTraceTest.hpp.

8.291.1.89 **TEST_F(CompleteTraceTest , UnaryNumericMeasureRound)**

Definition at line 830 of file CompleteTraceTest.hpp.

8.291.1.90 **TEST_F(CompleteTraceTest , UnaryNumericMeasureSign)**

Definition at line 834 of file CompleteTraceTest.hpp.

8.291.1.91 **TEST_F(CompleteTraceTest , UnaryNumericMeasureSqrt)**

Definition at line 838 of file CompleteTraceTest.hpp.

8.291.1.92 **TEST_F(CompleteTraceTest , UnaryNumericMeasureTrunc)**

Definition at line 842 of file CompleteTraceTest.hpp.

8.291.1.93 **TEST_F(CompleteTraceTest , UnaryNumericNumeric)**

Definition at line 855 of file CompleteTraceTest.hpp.

8.291.1.94 **TEST_F(CompleteTraceTest , UnarySubsetMeasureCount)**

Definition at line 868 of file CompleteTraceTest.hpp.

8.291.1.95 **TEST_F(CompleteTraceTest , UnarySubsetMeasureClusteredness)**

Definition at line 872 of file CompleteTraceTest.hpp.

8.291.1.96 **TEST_F(CompleteTraceTest , UnarySubsetMeasureDensity)**

Definition at line 876 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/CompleteTraceTest.hpp File

8.291.1.97 TEST_F(CompleteTraceTest , UnarySubset)

1281

Definition at line 889 of file CompleteTraceTest.hpp.

8.291.1.98 TEST_F(CompleteTraceTest , UntilLogicProperty)

Definition at line 902 of file CompleteTraceTest.hpp.

8.291.1.99 TEST_F(CompleteTraceTest , UntilLogicPropertyMultiple)

Definition at line 906 of file CompleteTraceTest.hpp.

8.291.1.100 TEST_F(CompleteTraceTest , GlobalConstantValueReal)

Definition at line 919 of file CompleteTraceTest.hpp.

8.291.1.101 TEST_F(CompleteTraceTest , GlobalConstantValueNumericStateVariable)

Definition at line 923 of file CompleteTraceTest.hpp.

8.291.1.102 TEST_F(CompleteTraceTest , GlobalConstantValueUnaryNumeric)

Definition at line 927 of file CompleteTraceTest.hpp.

8.291.1.103 TEST_F(CompleteTraceTest , GlobalConstantValueBinaryNumeric)

Definition at line 931 of file CompleteTraceTest.hpp.

8.291.1.104 TEST_F(CompleteTraceTest , GlobalConstantValueUnarySubset)

Definition at line 935 of file CompleteTraceTest.hpp.

8.291.1.105 TEST_F(CompleteTraceTest , GlobalConstantValueBinarySubset)

Definition at line 939 of file CompleteTraceTest.hpp.

8.291.1.106 TEST_F(CompleteTraceTest , GlobalConstantValueTernarySubset)

Definition at line 943 of file CompleteTraceTest.hpp.

8.291.1.107 **TEST_F(CompleteTraceTest , GlobalConstantValueQuaternarySubset)**

Definition at line 947 of file CompleteTraceTest.hpp.

8.291.1.108 **TEST_F(CompleteTraceTest , FutureIncreasingValueReal)**

Definition at line 960 of file CompleteTraceTest.hpp.

8.291.1.109 **TEST_F(CompleteTraceTest , FutureIncreasingValueNumericStateVariable)**

Definition at line 964 of file CompleteTraceTest.hpp.

8.291.1.110 **TEST_F(CompleteTraceTest , FutureIncreasingValueUnaryNumeric)**

Definition at line 968 of file CompleteTraceTest.hpp.

8.291.1.111 **TEST_F(CompleteTraceTest , FutureIncreasingValueBinaryNumeric)**

Definition at line 972 of file CompleteTraceTest.hpp.

8.291.1.112 **TEST_F(CompleteTraceTest , FutureIncreasingValueUnarySubset)**

Definition at line 976 of file CompleteTraceTest.hpp.

8.291.1.113 **TEST_F(CompleteTraceTest , FutureIncreasingValueBinarySubset)**

Definition at line 980 of file CompleteTraceTest.hpp.

8.291.1.114 **TEST_F(CompleteTraceTest , FutureIncreasingValueTernarySubset)**

Definition at line 984 of file CompleteTraceTest.hpp.

8.291.1.115 **TEST_F(CompleteTraceTest , FutureIncreasingValueQuaternarySubset)**

Definition at line 988 of file CompleteTraceTest.hpp.

8.291.1.116 **TEST_F(CompleteTraceTest , GlobalDecreasingValueReal)**

Definition at line 1001 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File
8.291.1.117 TEST_F(CompleteTraceTest , GlobalDecreasingValueNumericStateVariable)
1283

Definition at line 1005 of file CompleteTraceTest.hpp.

8.291.1.118 TEST_F(CompleteTraceTest , GlobalDecreasingValueUnaryNumeric)

Definition at line 1009 of file CompleteTraceTest.hpp.

8.291.1.119 TEST_F(CompleteTraceTest , GlobalDecreasingValueBinaryNumeric)

Definition at line 1013 of file CompleteTraceTest.hpp.

8.291.1.120 TEST_F(CompleteTraceTest , GlobalDecreasingValueUnarySubset)

Definition at line 1017 of file CompleteTraceTest.hpp.

8.291.1.121 TEST_F(CompleteTraceTest , GlobalDecreasingValueBinarySubset)

Definition at line 1021 of file CompleteTraceTest.hpp.

8.291.1.122 TEST_F(CompleteTraceTest , GlobalDecreasingValueTernarySubset)

Definition at line 1025 of file CompleteTraceTest.hpp.

8.291.1.123 TEST_F(CompleteTraceTest , GlobalDecreasingValueQuaternarySubset)

Definition at line 1029 of file CompleteTraceTest.hpp.

8.291.1.124 TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueReal)

Definition at line 1042 of file CompleteTraceTest.hpp.

8.291.1.125 TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueNumericState-Variable)

Definition at line 1046 of file CompleteTraceTest.hpp.

8.291.1.126 TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueNumericState-Variable2)

Definition at line 1050 of file CompleteTraceTest.hpp.

8.291.1.127 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueUnaryNumeric)**

Definition at line 1054 of file CompleteTraceTest.hpp.

8.291.1.128 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueBinaryNumeric)**

Definition at line 1058 of file CompleteTraceTest.hpp.

8.291.1.129 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueUnarySubset)**

Definition at line 1062 of file CompleteTraceTest.hpp.

8.291.1.130 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueBinarySubset)**

Definition at line 1066 of file CompleteTraceTest.hpp.

8.291.1.131 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueTernarySubset)**

Definition at line 1070 of file CompleteTraceTest.hpp.

8.291.1.132 **TEST_F(CompleteTraceTest , IncreasingUntilDecreasingValueQuaternary-Subset)**

Definition at line 1074 of file CompleteTraceTest.hpp.

8.291.1.133 **TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueReal)**

Definition at line 1087 of file CompleteTraceTest.hpp.

8.291.1.134 **TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueNumericState-Variable)**

Definition at line 1091 of file CompleteTraceTest.hpp.

8.291.1.135 **TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueUnaryNumeric)**

Definition at line 1095 of file CompleteTraceTest.hpp.

8.291

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/CompleteTraceTest.hpp File
Reference TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueBinaryNumeric)
1285

Definition at line 1099 of file CompleteTraceTest.hpp.

8.291.1.137 TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueUnarySubset)

Definition at line 1103 of file CompleteTraceTest.hpp.

8.291.1.138 TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueBinarySubset)

Definition at line 1107 of file CompleteTraceTest.hpp.

8.291.1.139 TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueTernarySubset)

Definition at line 1111 of file CompleteTraceTest.hpp.

8.291.1.140 TEST_F(CompleteTraceTest , DecreasingUntilIncreasingValueQuaternarySubset)

Definition at line 1115 of file CompleteTraceTest.hpp.

8.291.1.141 TEST_F(CompleteTraceTest , OscillationValueNumericStateVariable)

Definition at line 1128 of file CompleteTraceTest.hpp.

8.291.1.142 TEST_F(CompleteTraceTest , OscillationsValueUnaryNumeric)

Definition at line 1132 of file CompleteTraceTest.hpp.

8.291.1.143 TEST_F(CompleteTraceTest , OscillationsValueBinaryNumeric)

Definition at line 1136 of file CompleteTraceTest.hpp.

8.291.1.144 TEST_F(CompleteTraceTest , OscillationsValueUnarySubset)

Definition at line 1140 of file CompleteTraceTest.hpp.

8.291.1.145 TEST_F(CompleteTraceTest , OscillationsValueBinarySubset)

Definition at line 1144 of file CompleteTraceTest.hpp.

8.291.1.146 TEST_F(CompleteTraceTest , OscillationsValueTernarySubset)

Definition at line 1148 of file CompleteTraceTest.hpp.

8.291.1.147 TEST_F(CompleteTraceTest , OscillationsValueQuaternarySubset)

Definition at line 1152 of file CompleteTraceTest.hpp.

8.291.1.148 TEST_F(CompleteTraceTest , EnclosingWithParenthesesDifferently1)

Definition at line 1165 of file CompleteTraceTest.hpp.

8.291.1.149 TEST_F(CompleteTraceTest , EnclosingWithParenthesesDifferently2)

Definition at line 1169 of file CompleteTraceTest.hpp.

8.291.1.150 TEST_F(CompleteTraceTest , TimeIntervalExceedsTraceEndTime)

Definition at line 1182 of file CompleteTraceTest.hpp.

8.291.1.151 TEST_F(CompleteTraceTest , TimeIntervalExceedsTraceStartTime)

Definition at line 1186 of file CompleteTraceTest.hpp.

8.291.1.152 TEST_F(CompleteTraceTest , ConstraintsCombinationUnary)

Definition at line 1199 of file CompleteTraceTest.hpp.

8.291.1.153 TEST_F(CompleteTraceTest , ConstraintsCombinationBinary)

Definition at line 1203 of file CompleteTraceTest.hpp.

8.291.1.154 TEST_F(CompleteTraceTest , ConstraintsCombinationNary)

Definition at line 1207 of file CompleteTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp File

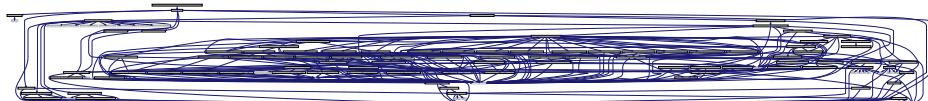
8.292 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File Reference

1287

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

- namespace [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 \(EmptyTraceTest, BinarySubsetMeasureAvg\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureGeomean\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureHarmean\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureKurt\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureMax\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureMedian\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureMin\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureMode\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureProduct\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureSkew\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureStdev\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureSum\)](#)
- [TEST_F \(EmptyTraceTest, BinarySubsetMeasureVar\)](#)

- [TEST_F \(EmptyTraceTest, BinarySubset\)](#)
- [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 \(EmptyTraceTest, Difference\)](#)
- [TEST_F \(EmptyTraceTest, FilterNumericMeasure\)](#)
- [TEST_F \(EmptyTraceTest, FilterSubset\)](#)
- [TEST_F \(EmptyTraceTest, FutureLogicProperty\)](#)
- [TEST_F \(EmptyTraceTest, GlobalLogicProperty\)](#)
- [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 \(EmptyTraceTest, NumericNumericComparison\)](#)
- [TEST_F \(EmptyTraceTest, NumericSpatialMeasure\)](#)
- [TEST_F \(EmptyTraceTest, NumericSpatialNumericComparison\)](#)
- [TEST_F \(EmptyTraceTest, NumericStateVariable1\)](#)
- [TEST_F \(EmptyTraceTest, NumericStateVariable2\)](#)
- [TEST_F \(EmptyTraceTest, NumericStateVariable3\)](#)
- [TEST_F \(EmptyTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST_F \(EmptyTraceTest, QuaternarySubsetCovar\)](#)
- [TEST_F \(EmptyTraceTest, QuaternarySubset\)](#)
- [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 \(EmptyTraceTest, SpatialMeasureTriangleMeasure\)](#)

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp File

Reference [TEST_F \(EmptyTraceTest, SpatialMeasureRectangleMeasure\)](#)

1289

- [TEST_F \(EmptyTraceTest, SpatialMeasureCircleMeasure\)](#)
- [TEST_F \(EmptyTraceTest, SpatialMeasureCentroidX\)](#)
- [TEST_F \(EmptyTraceTest, SpatialMeasureCentroidY\)](#)
- [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 \(EmptyTraceTest, Subset\)](#)
- [TEST_F \(EmptyTraceTest, TernarySubsetMeasurePercentile\)](#)
- [TEST_F \(EmptyTraceTest, TernarySubsetMeasureQuartile\)](#)
- [TEST_F \(EmptyTraceTest, TernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, UnaryTypeConstraint\)](#)
- [TEST_F \(EmptyTraceTest, UnarySpatialConstraint\)](#)
- [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 \(EmptyTraceTest, UnarySubsetMeasureCount\)](#)
- [TEST_F \(EmptyTraceTest, UnarySubsetMeasureClusteredness\)](#)
- [TEST_F \(EmptyTraceTest, UnarySubsetMeasureDensity\)](#)
- [TEST_F \(EmptyTraceTest, UnarySubset\)](#)
- [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 \(EmptyTraceTest, GlobalConstantValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalConstantValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalConstantValueTernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalConstantValueQuaternarySubset\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueReal\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, FutureIncreasingValueTernarySubset\)](#)

- [TEST_F \(EmptyTraceTest, FutureIncreasingValueQuaternarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueTernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, GlobalDecreasingValueQuaternarySubset\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueNumericState-Variable\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueTernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, IncreasingUntilDecreasingValueQuaternarySubset\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueNumericState-Variable\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueUnaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueBinaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueTernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, DecreasingUntilIncreasingValueQuaternarySubset\)](#)
- [TEST_F \(EmptyTraceTest, OscillationValueNumericStateVariable\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueUnarySubset\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueBinarySubset\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueTernarySubset\)](#)
- [TEST_F \(EmptyTraceTest, OscillationsValueQuaternarySubset\)](#)
- [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 \(EmptyTraceTest, ConstraintsCombinationNary\)](#)

8.292.1 Function Documentation

8.292.1.1 TEST_F (EmptyTraceTest , BinaryNumericFilter)

Definition at line 37 of file `EmptyTraceTest.hpp`.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.2 TEST_F(EmptyTraceTest , BinaryNumericMeasureAdd)

1291

Definition at line 50 of file EmptyTraceTest.hpp.

8.292.1.3 TEST_F(EmptyTraceTest , BinaryNumericMeasureDiv)

Definition at line 54 of file EmptyTraceTest.hpp.

8.292.1.4 TEST_F(EmptyTraceTest , BinaryNumericMeasureLog)

Definition at line 58 of file EmptyTraceTest.hpp.

8.292.1.5 TEST_F(EmptyTraceTest , BinaryNumericMeasureMod)

Definition at line 62 of file EmptyTraceTest.hpp.

8.292.1.6 TEST_F(EmptyTraceTest , BinaryNumericMeasureMultiply)

Definition at line 66 of file EmptyTraceTest.hpp.

8.292.1.7 TEST_F(EmptyTraceTest , BinaryNumericMeasurePower)

Definition at line 70 of file EmptyTraceTest.hpp.

8.292.1.8 TEST_F(EmptyTraceTest , BinaryNumericMeasureSubtract)

Definition at line 74 of file EmptyTraceTest.hpp.

8.292.1.9 TEST_F(EmptyTraceTest , BinaryNumericNumeric)

Definition at line 87 of file EmptyTraceTest.hpp.

8.292.1.10 TEST_F(EmptyTraceTest , BinarySubsetMeasureAvg)

Definition at line 100 of file EmptyTraceTest.hpp.

8.292.1.11 TEST_F(EmptyTraceTest , BinarySubsetMeasureGeomean)

Definition at line 104 of file EmptyTraceTest.hpp.

8.292.1.12 TEST_F(EmptyTraceTest , BinarySubsetMeasureHarmean)

Definition at line 108 of file EmptyTraceTest.hpp.

8.292.1.13 TEST_F(EmptyTraceTest , BinarySubsetMeasureKurt)

Definition at line 112 of file EmptyTraceTest.hpp.

8.292.1.14 TEST_F(EmptyTraceTest , BinarySubsetMeasureMax)

Definition at line 116 of file EmptyTraceTest.hpp.

8.292.1.15 TEST_F(EmptyTraceTest , BinarySubsetMeasureMedian)

Definition at line 120 of file EmptyTraceTest.hpp.

8.292.1.16 TEST_F(EmptyTraceTest , BinarySubsetMeasureMin)

Definition at line 124 of file EmptyTraceTest.hpp.

8.292.1.17 TEST_F(EmptyTraceTest , BinarySubsetMeasureMode)

Definition at line 128 of file EmptyTraceTest.hpp.

8.292.1.18 TEST_F(EmptyTraceTest , BinarySubsetMeasureProduct)

Definition at line 132 of file EmptyTraceTest.hpp.

8.292.1.19 TEST_F(EmptyTraceTest , BinarySubsetMeasureSkew)

Definition at line 136 of file EmptyTraceTest.hpp.

8.292.1.20 TEST_F(EmptyTraceTest , BinarySubsetMeasureStdev)

Definition at line 140 of file EmptyTraceTest.hpp.

8.292.1.21 TEST_F(EmptyTraceTest , BinarySubsetMeasureSum)

Definition at line 144 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.22 TEST_F(EmptyTraceTest , BinarySubsetMeasureVar)

1293

Definition at line 148 of file EmptyTraceTest.hpp.

8.292.1.23 TEST_F(EmptyTraceTest , BinarySubset)

Definition at line 161 of file EmptyTraceTest.hpp.

8.292.1.24 TEST_F(EmptyTraceTest , ComparatorGreaterThan)

Definition at line 174 of file EmptyTraceTest.hpp.

8.292.1.25 TEST_F(EmptyTraceTest , ComparatorLessThan)

Definition at line 178 of file EmptyTraceTest.hpp.

8.292.1.26 TEST_F(EmptyTraceTest , ComparatorGreaterThanOrEqual)

Definition at line 182 of file EmptyTraceTest.hpp.

8.292.1.27 TEST_F(EmptyTraceTest , ComparatorLessThanOrEqual)

Definition at line 186 of file EmptyTraceTest.hpp.

8.292.1.28 TEST_F(EmptyTraceTest , ComparatorEqual)

Definition at line 190 of file EmptyTraceTest.hpp.

8.292.1.29 TEST_F(EmptyTraceTest , CompoundConstraint)

Definition at line 203 of file EmptyTraceTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS.

8.292.1.30 TEST_F(EmptyTraceTest , CompoundConstraintMultiple)

Definition at line 211 of file EmptyTraceTest.hpp.

References CONSTRAINTS_BINARY_OPERATORS.

8.292.1.31 TEST_F(EmptyTraceTest , CompoundLogicProperty)

Definition at line 228 of file EmptyTraceTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS.

8.292.1.32 TEST_F(EmptyTraceTest , CompoundLogicPropertyMultiple)

Definition at line 236 of file EmptyTraceTest.hpp.

References LOGIC_PROPERTIES_BINARY_OPERATORS.

8.292.1.33 TEST_F(EmptyTraceTest , ConstraintEnclosedByParentheses)

Definition at line 253 of file EmptyTraceTest.hpp.

8.292.1.34 TEST_F(EmptyTraceTest , ConstraintEnclosedByParenthesesDoubled)

Definition at line 257 of file EmptyTraceTest.hpp.

8.292.1.35 TEST_F(EmptyTraceTest , ConstraintEnclosedByParenthesesQuadrupled)

Definition at line 261 of file EmptyTraceTest.hpp.

8.292.1.36 TEST_F(EmptyTraceTest , Constraint)

Definition at line 274 of file EmptyTraceTest.hpp.

8.292.1.37 TEST_F(EmptyTraceTest , Difference)

Definition at line 287 of file EmptyTraceTest.hpp.

8.292.1.38 TEST_F(EmptyTraceTest , FilterNumericMeasure)

Definition at line 300 of file EmptyTraceTest.hpp.

8.292.1.39 TEST_F(EmptyTraceTest , FilterSubset)

Definition at line 313 of file EmptyTraceTest.hpp.

8.292.1.40 TEST_F(EmptyTraceTest , FutureLogicProperty)

Definition at line 326 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.41 TEST_F(EmptyTraceTest , GlobalLogicProperty)

1295

Definition at line 339 of file EmptyTraceTest.hpp.

8.292.1.42 TEST_F(EmptyTraceTest , LogicPropertyEnclosedByParentheses)

Definition at line 352 of file EmptyTraceTest.hpp.

8.292.1.43 TEST_F(EmptyTraceTest , LogicPropertyEnclosedByParenthesesDoubled)

Definition at line 356 of file EmptyTraceTest.hpp.

8.292.1.44 TEST_F(EmptyTraceTest , LogicPropertyEnclosedByParenthesesQuadrupled)

Definition at line 360 of file EmptyTraceTest.hpp.

8.292.1.45 TEST_F(EmptyTraceTest , LogicProperty)

Definition at line 373 of file EmptyTraceTest.hpp.

8.292.1.46 TEST_F(EmptyTraceTest , MultipleLogicProperties1)

Definition at line 386 of file EmptyTraceTest.hpp.

8.292.1.47 TEST_F(EmptyTraceTest , MultipleLogicProperties2)

Definition at line 390 of file EmptyTraceTest.hpp.

8.292.1.48 TEST_F(EmptyTraceTest , NextKLogicProperty)

Definition at line 403 of file EmptyTraceTest.hpp.

8.292.1.49 TEST_F(EmptyTraceTest , NextLogicProperty)

Definition at line 416 of file EmptyTraceTest.hpp.

8.292.1.50 TEST_F(EmptyTraceTest , NotConstraint)

Definition at line 429 of file EmptyTraceTest.hpp.

8.292.1.51 TEST_F(EmptyTraceTest , NotLogicProperty)

Definition at line 442 of file EmptyTraceTest.hpp.

8.292.1.52 TEST_F(EmptyTraceTest , NumericMeasure)

Definition at line 455 of file EmptyTraceTest.hpp.

8.292.1.53 TEST_F(EmptyTraceTest , NumericNumericComparison)

Definition at line 468 of file EmptyTraceTest.hpp.

8.292.1.54 TEST_F(EmptyTraceTest , NumericSpatialMeasure)

Definition at line 481 of file EmptyTraceTest.hpp.

8.292.1.55 TEST_F(EmptyTraceTest , NumericSpatialNumericComparison)

Definition at line 494 of file EmptyTraceTest.hpp.

8.292.1.56 TEST_F(EmptyTraceTest , NumericStateVariable1)

Definition at line 507 of file EmptyTraceTest.hpp.

8.292.1.57 TEST_F(EmptyTraceTest , NumericStateVariable2)

Definition at line 511 of file EmptyTraceTest.hpp.

8.292.1.58 TEST_F(EmptyTraceTest , NumericStateVariable3)

Definition at line 515 of file EmptyTraceTest.hpp.

8.292.1.59 TEST_F(EmptyTraceTest , ProbabilisticLogicProperty)

Definition at line 528 of file EmptyTraceTest.hpp.

8.292.1.60 TEST_F(EmptyTraceTest , QuaternarySubsetCovar)

Definition at line 541 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.61 TEST_F(EmptyTraceTest , QuaternarySubset)

1297

Definition at line 554 of file EmptyTraceTest.hpp.

8.292.1.62 TEST_F(EmptyTraceTest , SpatialMeasureClusteredness)

Definition at line 567 of file EmptyTraceTest.hpp.

8.292.1.63 TEST_F(EmptyTraceTest , SpatialMeasureDensity)

Definition at line 571 of file EmptyTraceTest.hpp.

8.292.1.64 TEST_F(EmptyTraceTest , SpatialMeasureArea)

Definition at line 575 of file EmptyTraceTest.hpp.

8.292.1.65 TEST_F(EmptyTraceTest , SpatialMeasurePerimeter)

Definition at line 579 of file EmptyTraceTest.hpp.

8.292.1.66 TEST_F(EmptyTraceTest , SpatialMeasureDistanceFromOrigin)

Definition at line 583 of file EmptyTraceTest.hpp.

8.292.1.67 TEST_F(EmptyTraceTest , SpatialMeasureAngle)

Definition at line 587 of file EmptyTraceTest.hpp.

8.292.1.68 TEST_F(EmptyTraceTest , SpatialMeasureTriangleMeasure)

Definition at line 591 of file EmptyTraceTest.hpp.

8.292.1.69 TEST_F(EmptyTraceTest , SpatialMeasureRectangleMeasure)

Definition at line 595 of file EmptyTraceTest.hpp.

8.292.1.70 TEST_F(EmptyTraceTest , SpatialMeasureCircleMeasure)

Definition at line 599 of file EmptyTraceTest.hpp.

8.292.1.71 TEST_F(EmptyTraceTest , SpatialMeasureCentroidX)

Definition at line 603 of file EmptyTraceTest.hpp.

8.292.1.72 TEST_F(EmptyTraceTest , SpatialMeasureCentroidY)

Definition at line 607 of file EmptyTraceTest.hpp.

8.292.1.73 TEST_F(EmptyTraceTest , SubsetOperationDifference)

Definition at line 620 of file EmptyTraceTest.hpp.

8.292.1.74 TEST_F(EmptyTraceTest , SubsetOperationIntersection)

Definition at line 624 of file EmptyTraceTest.hpp.

8.292.1.75 TEST_F(EmptyTraceTest , SubsetOperationUnion)

Definition at line 628 of file EmptyTraceTest.hpp.

8.292.1.76 TEST_F(EmptyTraceTest , SubsetSpecificClusters)

Definition at line 641 of file EmptyTraceTest.hpp.

8.292.1.77 TEST_F(EmptyTraceTest , SubsetSpecificRegions)

Definition at line 645 of file EmptyTraceTest.hpp.

8.292.1.78 TEST_F(EmptyTraceTest , SubsetSubsetOperation)

Definition at line 658 of file EmptyTraceTest.hpp.

8.292.1.79 TEST_F(EmptyTraceTest , Subset)

Definition at line 671 of file EmptyTraceTest.hpp.

8.292.1.80 TEST_F(EmptyTraceTest , TernarySubsetMeasurePercentile)

Definition at line 684 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.81 TEST_F(EmptyTraceTest , TernarySubsetMeasureQuartile)

1299

Definition at line 688 of file EmptyTraceTest.hpp.

8.292.1.82 TEST_F(EmptyTraceTest , TernarySubset)

Definition at line 701 of file EmptyTraceTest.hpp.

8.292.1.83 TEST_F(EmptyTraceTest , UnaryTypeConstraint)

Definition at line 714 of file EmptyTraceTest.hpp.

8.292.1.84 TEST_F(EmptyTraceTest , UnarySpatialConstraint)

Definition at line 727 of file EmptyTraceTest.hpp.

8.292.1.85 TEST_F(EmptyTraceTest , UnaryNumericFilter)

Definition at line 740 of file EmptyTraceTest.hpp.

8.292.1.86 TEST_F(EmptyTraceTest , UnaryNumericMeasureAbs)

Definition at line 753 of file EmptyTraceTest.hpp.

8.292.1.87 TEST_F(EmptyTraceTest , UnaryNumericMeasureCeil)

Definition at line 757 of file EmptyTraceTest.hpp.

8.292.1.88 TEST_F(EmptyTraceTest , UnaryNumericMeasureFloor)

Definition at line 761 of file EmptyTraceTest.hpp.

8.292.1.89 TEST_F(EmptyTraceTest , UnaryNumericMeasureRound)

Definition at line 765 of file EmptyTraceTest.hpp.

8.292.1.90 TEST_F(EmptyTraceTest , UnaryNumericMeasureSign)

Definition at line 769 of file EmptyTraceTest.hpp.

8.292.1.91 TEST_F(EmptyTraceTest , UnaryNumericMeasureSqrt)

Definition at line 773 of file EmptyTraceTest.hpp.

8.292.1.92 TEST_F(EmptyTraceTest , UnaryNumericMeasureTrunc)

Definition at line 777 of file EmptyTraceTest.hpp.

8.292.1.93 TEST_F(EmptyTraceTest , UnaryNumericNumeric)

Definition at line 790 of file EmptyTraceTest.hpp.

8.292.1.94 TEST_F(EmptyTraceTest , UnarySubsetMeasureCount)

Definition at line 803 of file EmptyTraceTest.hpp.

8.292.1.95 TEST_F(EmptyTraceTest , UnarySubsetMeasureClusteredness)

Definition at line 807 of file EmptyTraceTest.hpp.

8.292.1.96 TEST_F(EmptyTraceTest , UnarySubsetMeasureDensity)

Definition at line 811 of file EmptyTraceTest.hpp.

8.292.1.97 TEST_F(EmptyTraceTest , UnarySubset)

Definition at line 824 of file EmptyTraceTest.hpp.

8.292.1.98 TEST_F(EmptyTraceTest , UntilLogicProperty)

Definition at line 837 of file EmptyTraceTest.hpp.

8.292.1.99 TEST_F(EmptyTraceTest , UntilLogicPropertyMultiple)

Definition at line 841 of file EmptyTraceTest.hpp.

8.292.1.100 TEST_F(EmptyTraceTest , GlobalConstantValueReal)

Definition at line 854 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.101 TEST_F(EmptyTraceTest , GlobalConstantValueNumericStateVariable)1301

Definition at line 858 of file EmptyTraceTest.hpp.

8.292.1.102 TEST_F(EmptyTraceTest , GlobalConstantValueUnaryNumeric)

Definition at line 862 of file EmptyTraceTest.hpp.

8.292.1.103 TEST_F(EmptyTraceTest , GlobalConstantValueBinaryNumeric)

Definition at line 866 of file EmptyTraceTest.hpp.

8.292.1.104 TEST_F(EmptyTraceTest , GlobalConstantValueUnarySubset)

Definition at line 870 of file EmptyTraceTest.hpp.

8.292.1.105 TEST_F(EmptyTraceTest , GlobalConstantValueBinarySubset)

Definition at line 874 of file EmptyTraceTest.hpp.

8.292.1.106 TEST_F(EmptyTraceTest , GlobalConstantValueTernarySubset)

Definition at line 878 of file EmptyTraceTest.hpp.

8.292.1.107 TEST_F(EmptyTraceTest , GlobalConstantValueQuaternarySubset)

Definition at line 882 of file EmptyTraceTest.hpp.

8.292.1.108 TEST_F(EmptyTraceTest , FutureIncreasingValueReal)

Definition at line 895 of file EmptyTraceTest.hpp.

8.292.1.109 TEST_F(EmptyTraceTest , FutureIncreasingValueNumericStateVariable)

Definition at line 899 of file EmptyTraceTest.hpp.

8.292.1.110 TEST_F(EmptyTraceTest , FutureIncreasingValueUnaryNumeric)

Definition at line 903 of file EmptyTraceTest.hpp.

8.292.1.111 TEST_F(EmptyTraceTest , FutureIncreasingValueBinaryNumeric)

Definition at line 907 of file EmptyTraceTest.hpp.

8.292.1.112 TEST_F(EmptyTraceTest , FutureIncreasingValueUnarySubset)

Definition at line 911 of file EmptyTraceTest.hpp.

8.292.1.113 TEST_F(EmptyTraceTest , FutureIncreasingValueBinarySubset)

Definition at line 915 of file EmptyTraceTest.hpp.

8.292.1.114 TEST_F(EmptyTraceTest , FutureIncreasingValueTernarySubset)

Definition at line 919 of file EmptyTraceTest.hpp.

8.292.1.115 TEST_F(EmptyTraceTest , FutureIncreasingValueQuaternarySubset)

Definition at line 923 of file EmptyTraceTest.hpp.

8.292.1.116 TEST_F(EmptyTraceTest , GlobalDecreasingValueReal)

Definition at line 936 of file EmptyTraceTest.hpp.

8.292.1.117 TEST_F(EmptyTraceTest , GlobalDecreasingValueNumericStateVariable)

Definition at line 940 of file EmptyTraceTest.hpp.

8.292.1.118 TEST_F(EmptyTraceTest , GlobalDecreasingValueUnaryNumeric)

Definition at line 944 of file EmptyTraceTest.hpp.

8.292.1.119 TEST_F(EmptyTraceTest , GlobalDecreasingValueBinaryNumeric)

Definition at line 948 of file EmptyTraceTest.hpp.

8.292.1.120 TEST_F(EmptyTraceTest , GlobalDecreasingValueUnarySubset)

Definition at line 952 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.121 TEST_F(EmptyTraceTest , GlobalDecreasingValueBinarySubset)

1303

Definition at line 956 of file EmptyTraceTest.hpp.

8.292.1.122 TEST_F(EmptyTraceTest , GlobalDecreasingValueTernarySubset)

Definition at line 960 of file EmptyTraceTest.hpp.

8.292.1.123 TEST_F(EmptyTraceTest , GlobalDecreasingValueQuaternarySubset)

Definition at line 964 of file EmptyTraceTest.hpp.

8.292.1.124 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueReal)

Definition at line 977 of file EmptyTraceTest.hpp.

8.292.1.125 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueNumericState-
Variable)

Definition at line 981 of file EmptyTraceTest.hpp.

8.292.1.126 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueUnaryNumeric)

Definition at line 985 of file EmptyTraceTest.hpp.

8.292.1.127 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueBinaryNumeric)

Definition at line 989 of file EmptyTraceTest.hpp.

8.292.1.128 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueUnarySubset)

Definition at line 993 of file EmptyTraceTest.hpp.

8.292.1.129 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueBinarySubset)

Definition at line 997 of file EmptyTraceTest.hpp.

8.292.1.130 TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueTernarySubset)

Definition at line 1001 of file EmptyTraceTest.hpp.

8.292.1.131 **TEST_F(EmptyTraceTest , IncreasingUntilDecreasingValueQuaternarySubset)**

Definition at line 1005 of file EmptyTraceTest.hpp.

8.292.1.132 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueReal)**

Definition at line 1018 of file EmptyTraceTest.hpp.

8.292.1.133 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueNumericStateVariable)**

Definition at line 1022 of file EmptyTraceTest.hpp.

8.292.1.134 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueUnaryNumeric)**

Definition at line 1026 of file EmptyTraceTest.hpp.

8.292.1.135 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueBinaryNumeric)**

Definition at line 1030 of file EmptyTraceTest.hpp.

8.292.1.136 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueUnarySubset)**

Definition at line 1034 of file EmptyTraceTest.hpp.

8.292.1.137 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueBinarySubset)**

Definition at line 1038 of file EmptyTraceTest.hpp.

8.292.1.138 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueTernarySubset)**

Definition at line 1042 of file EmptyTraceTest.hpp.

8.292.1.139 **TEST_F(EmptyTraceTest , DecreasingUntilIncreasingValueQuaternarySubset)**

Definition at line 1046 of file EmptyTraceTest.hpp.

8.292.1.140 **TEST_F(EmptyTraceTest , OscillationValueNumericStateVariable)**

Definition at line 1059 of file EmptyTraceTest.hpp.

8.292

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/EmptyTraceTest.hpp File

8.292.1.141 TEST_F(EmptyTraceTest , OscillationsValueUnaryNumeric)

1305

Definition at line 1063 of file EmptyTraceTest.hpp.

8.292.1.142 TEST_F(EmptyTraceTest , OscillationsValueBinaryNumeric)

Definition at line 1067 of file EmptyTraceTest.hpp.

8.292.1.143 TEST_F(EmptyTraceTest , OscillationsValueUnarySubset)

Definition at line 1071 of file EmptyTraceTest.hpp.

8.292.1.144 TEST_F(EmptyTraceTest , OscillationsValueBinarySubset)

Definition at line 1075 of file EmptyTraceTest.hpp.

8.292.1.145 TEST_F(EmptyTraceTest , OscillationsValueTernarySubset)

Definition at line 1079 of file EmptyTraceTest.hpp.

8.292.1.146 TEST_F(EmptyTraceTest , OscillationsValueQuaternarySubset)

Definition at line 1083 of file EmptyTraceTest.hpp.

8.292.1.147 TEST_F(EmptyTraceTest , EnclosingWithParenthesesDifferently1)

Definition at line 1096 of file EmptyTraceTest.hpp.

8.292.1.148 TEST_F(EmptyTraceTest , EnclosingWithParenthesesDifferently2)

Definition at line 1100 of file EmptyTraceTest.hpp.

8.292.1.149 TEST_F(EmptyTraceTest , TimeIntervalExceedsTraceEndTime)

Definition at line 1113 of file EmptyTraceTest.hpp.

8.292.1.150 TEST_F(EmptyTraceTest , TimeIntervalExceedsTraceStartTime)

Definition at line 1117 of file EmptyTraceTest.hpp.

8.292.1.151 TEST_F(EmptyTraceTest , ConstraintsCombinationUnary)

Definition at line 1130 of file EmptyTraceTest.hpp.

8.292.1.152 TEST_F(EmptyTraceTest , ConstraintsCombinationBinary)

Definition at line 1134 of file EmptyTraceTest.hpp.

8.292.1.153 TEST_F(EmptyTraceTest , ConstraintsCombinationNary)

Definition at line 1138 of file EmptyTraceTest.hpp.

8.293 /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

- namespace [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 \(NumericStateVariableTraceTest, BinaryNumericMeasurePower\)](#)

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File

Reference TEST_F (NumericStateVariableTraceTest, BinaryNumericMeasureSubtract) 1307

- TEST_F (NumericStateVariableTraceTest, BinaryNumericNumeric)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureAvg)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureGeomean)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureHarmean)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureKurt)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureMax)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureMedian)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureMin)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureMode)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureProduct)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureSkew)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureStdev)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureSum)
- TEST_F (NumericStateVariableTraceTest, BinarySubsetMeasureVar)
- TEST_F (NumericStateVariableTraceTest, BinarySubset)
- TEST_F (NumericStateVariableTraceTest, ComparatorGreaterThan)
- TEST_F (NumericStateVariableTraceTest, ComparatorLessThan)
- TEST_F (NumericStateVariableTraceTest, ComparatorGreaterThanOrEqual)
- TEST_F (NumericStateVariableTraceTest, ComparatorLessThanOrEqual)
- 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, ConstraintEnclosedByParentheses-Doubled)
- TEST_F (NumericStateVariableTraceTest, ConstraintEnclosedByParentheses-Quadrupled)
- TEST_F (NumericStateVariableTraceTest, Constraint)
- TEST_F (NumericStateVariableTraceTest, Difference)
- TEST_F (NumericStateVariableTraceTest, FilterNumericMeasure)
- TEST_F (NumericStateVariableTraceTest, FilterSubset)
- TEST_F (NumericStateVariableTraceTest, FutureLogicProperty)
- TEST_F (NumericStateVariableTraceTest, GlobalLogicProperty)
- TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParentheses)
- TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParentheses-Doubled)
- TEST_F (NumericStateVariableTraceTest, LogicPropertyEnclosedByParentheses-Quadrupled)
- TEST_F (NumericStateVariableTraceTest, LogicProperty)
- TEST_F (NumericStateVariableTraceTest, MultipleLogicProperties1)
- 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 \(NumericStateVariableTraceTest, NumericNumericComparison\)](#)
- [TEST_F \(NumericStateVariableTraceTest, NumericSpatialMeasure\)](#)
- [TEST_F \(NumericStateVariableTraceTest, NumericSpatialNumericComparison\)](#)
- [TEST_F \(NumericStateVariableTraceTest, NumericStateVariable1\)](#)
- [TEST_F \(NumericStateVariableTraceTest, NumericStateVariable2\)](#)
- [TEST_F \(NumericStateVariableTraceTest, NumericStateVariable3\)](#)
- [TEST_F \(NumericStateVariableTraceTest, ProbabilisticLogicProperty\)](#)
- [TEST_F \(NumericStateVariableTraceTest, QuaternarySubsetCovar\)](#)
- [TEST_F \(NumericStateVariableTraceTest, QuaternarySubset\)](#)
- [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 \(NumericStateVariableTraceTest, SubsetOperationDifference\)](#)
- [TEST_F \(NumericStateVariableTraceTest, SubsetOperationIntersection\)](#)
- [TEST_F \(NumericStateVariableTraceTest, SubsetOperationUnion\)](#)
- [TEST_F \(NumericStateVariableTraceTest, SubsetSpecificClusters\)](#)
- [TEST_F \(NumericStateVariableTraceTest, SubsetSpecificRegions\)](#)
- [TEST_F \(NumericStateVariableTraceTest, SubsetSubsetOperation\)](#)
- [TEST_F \(NumericStateVariableTraceTest, Subset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, TernarySubsetMeasurePercentile\)](#)
- [TEST_F \(NumericStateVariableTraceTest, TernarySubsetMeasureQuartile\)](#)
- [TEST_F \(NumericStateVariableTraceTest, TernarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, UnaryTypeConstraint\)](#)
- [TEST_F \(NumericStateVariableTraceTest, UnarySpatialConstraint\)](#)
- [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 \(NumericStateVariableTraceTest, UnarySubsetMeasureCount\)](#)
- [TEST_F \(NumericStateVariableTraceTest, UnarySubsetMeasureClusteredness\)](#)
- [TEST_F \(NumericStateVariableTraceTest, UnarySubsetMeasureDensity\)](#)

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File

Reference TEST_F (NumericStateVariableTraceTest, UnarySubset)

1309

- TEST_F (NumericStateVariableTraceTest, UntilLogicProperty)
- TEST_F (NumericStateVariableTraceTest, UntilLogicPropertyMultiple)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueReal)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueNumericState-Variable)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueUnaryNumeric)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueBinaryNumeric)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueUnarySubset)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueBinarySubset)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueTernarySubset)
- TEST_F (NumericStateVariableTraceTest, GlobalConstantValueQuaternary-Subset)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueReal)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueNumericState-Variable)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueUnaryNumeric)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueBinary-
Numeric)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueUnarySubset)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueBinarySubset)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueTernarySubset)
- TEST_F (NumericStateVariableTraceTest, FutureIncreasingValueQuaternary-Subset)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueReal)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueNumericState-Variable)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueUnary-
Numeric)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueBinary-
Numeric)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueUnarySubset)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueBinarySubset)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueTernary-Subset)
- TEST_F (NumericStateVariableTraceTest, GlobalDecreasingValueQuaternary-Subset)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValueReal)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-
NumericStateVariable)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-
UnaryNumeric)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-
BinaryNumeric)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-
UnarySubset)
- TEST_F (NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-
BinarySubset)

- [TEST_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-TernarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, IncreasingUntilDecreasingValue-QuaternarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValueReal\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-NumericStateVariable\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-UnaryNumeric\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-BinaryNumeric\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-UnarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-BinarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-TernarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, DecreasingUntilIncreasingValue-QuaternarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationValueNumericState-Variable\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueUnaryNumeric\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueBinaryNumeric\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueUnarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueBinarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueTernarySubset\)](#)
- [TEST_F \(NumericStateVariableTraceTest, OscillationsValueQuaternarySubset\)](#)
- [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 \(NumericStateVariableTraceTest, ConstraintsCombinationNary\)](#)

8.293.1 Function Documentation

8.293.1.1 TEST_F(NumericStateVariableTraceTest , BinaryNumericFilter)

Definition at line 61 of file NumericStateVariableTraceTest.hpp.

8.293.1.2 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureAdd)

Definition at line 74 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.3 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureDiv)
Reference [1311](#)

Definition at line 78 of file NumericStateVariableTraceTest.hpp.

8.293.1.4 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureLog)

Definition at line 82 of file NumericStateVariableTraceTest.hpp.

8.293.1.5 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureMod)

Definition at line 86 of file NumericStateVariableTraceTest.hpp.

8.293.1.6 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureMultiply)

Definition at line 90 of file NumericStateVariableTraceTest.hpp.

8.293.1.7 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasurePower)

Definition at line 94 of file NumericStateVariableTraceTest.hpp.

8.293.1.8 TEST_F(NumericStateVariableTraceTest , BinaryNumericMeasureSubtract)

Definition at line 98 of file NumericStateVariableTraceTest.hpp.

8.293.1.9 TEST_F(NumericStateVariableTraceTest , BinaryNumericNumeric)

Definition at line 111 of file NumericStateVariableTraceTest.hpp.

8.293.1.10 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureAvg)

Definition at line 124 of file NumericStateVariableTraceTest.hpp.

8.293.1.11 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureGeomean)

Definition at line 128 of file NumericStateVariableTraceTest.hpp.

8.293.1.12 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureHarmean)

Definition at line 132 of file NumericStateVariableTraceTest.hpp.

8.293.1.13 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureKurt)

Definition at line 136 of file NumericStateVariableTraceTest.hpp.

8.293.1.14 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureMax)

Definition at line 140 of file NumericStateVariableTraceTest.hpp.

8.293.1.15 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureMedian)

Definition at line 144 of file NumericStateVariableTraceTest.hpp.

8.293.1.16 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureMin)

Definition at line 148 of file NumericStateVariableTraceTest.hpp.

8.293.1.17 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureMode)

Definition at line 152 of file NumericStateVariableTraceTest.hpp.

8.293.1.18 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureProduct)

Definition at line 156 of file NumericStateVariableTraceTest.hpp.

8.293.1.19 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureSkew)

Definition at line 160 of file NumericStateVariableTraceTest.hpp.

8.293.1.20 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureStdev)

Definition at line 164 of file NumericStateVariableTraceTest.hpp.

8.293.1.21 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureSum)

Definition at line 168 of file NumericStateVariableTraceTest.hpp.

8.293.1.22 TEST_F(NumericStateVariableTraceTest , BinarySubsetMeasureVar)

Definition at line 172 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.23 TEST_F(NumericStateVariableTraceTest , BinarySubset) **1313**

Definition at line 185 of file NumericStateVariableTraceTest.hpp.

8.293.1.24 TEST_F(NumericStateVariableTraceTest , ComparatorGreaterThan)

Definition at line 198 of file NumericStateVariableTraceTest.hpp.

8.293.1.25 TEST_F(NumericStateVariableTraceTest , ComparatorLessThan)

Definition at line 202 of file NumericStateVariableTraceTest.hpp.

8.293.1.26 TEST_F(NumericStateVariableTraceTest , ComparatorGreaterThanOrEqual)

Definition at line 206 of file NumericStateVariableTraceTest.hpp.

8.293.1.27 TEST_F(NumericStateVariableTraceTest , ComparatorLessThanOrEqual)

Definition at line 210 of file NumericStateVariableTraceTest.hpp.

8.293.1.28 TEST_F(NumericStateVariableTraceTest , ComparatorEqual)

Definition at line 214 of file NumericStateVariableTraceTest.hpp.

8.293.1.29 TEST_F(NumericStateVariableTraceTest , CompoundConstraint)

Definition at line 227 of file NumericStateVariableTraceTest.hpp.

8.293.1.30 TEST_F(NumericStateVariableTraceTest , CompoundConstraintMultiple)

Definition at line 234 of file NumericStateVariableTraceTest.hpp.

8.293.1.31 TEST_F(NumericStateVariableTraceTest , CompoundLogicProperty)

Definition at line 250 of file NumericStateVariableTraceTest.hpp.

8.293.1.32 TEST_F(NumericStateVariableTraceTest , CompoundLogicPropertyMultiple)

Definition at line 257 of file NumericStateVariableTraceTest.hpp.

8.293.1.33 **TEST_F** (**NumericStateVariableTraceTest** ,
 ConstraintEnclosedByParentheses)

Definition at line 273 of file NumericStateVariableTraceTest.hpp.

8.293.1.34 **TEST_F** (**NumericStateVariableTraceTest** ,
 ConstraintEnclosedByParenthesesDoubled)

Definition at line 277 of file NumericStateVariableTraceTest.hpp.

8.293.1.35 **TEST_F** (**NumericStateVariableTraceTest** ,
 ConstraintEnclosedByParenthesesQuadrupled)

Definition at line 281 of file NumericStateVariableTraceTest.hpp.

8.293.1.36 **TEST_F** (**NumericStateVariableTraceTest** , **Constraint**)

Definition at line 294 of file NumericStateVariableTraceTest.hpp.

8.293.1.37 **TEST_F** (**NumericStateVariableTraceTest** , **Difference**)

Definition at line 307 of file NumericStateVariableTraceTest.hpp.

8.293.1.38 **TEST_F** (**NumericStateVariableTraceTest** , **FilterNumericMeasure**)

Definition at line 320 of file NumericStateVariableTraceTest.hpp.

8.293.1.39 **TEST_F** (**NumericStateVariableTraceTest** , **FilterSubset**)

Definition at line 333 of file NumericStateVariableTraceTest.hpp.

8.293.1.40 **TEST_F** (**NumericStateVariableTraceTest** , **FutureLogicProperty**)

Definition at line 346 of file NumericStateVariableTraceTest.hpp.

8.293.1.41 **TEST_F** (**NumericStateVariableTraceTest** , **GlobalLogicProperty**)

Definition at line 359 of file NumericStateVariableTraceTest.hpp.

8.293.1.42 **TEST_F** (**NumericStateVariableTraceTest** ,
 LogicPropertyEnclosedByParentheses)

Definition at line 372 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.43 TEST_F (NumericStateVariableTraceTest ,
Reference LogicPropertyEnclosedByParenthesesDoubled)

1315

Definition at line 376 of file NumericStateVariableTraceTest.hpp.

**8.293.1.44 TEST_F (NumericStateVariableTraceTest ,
LogicPropertyEnclosedByParenthesesQuadrupled)**

Definition at line 380 of file NumericStateVariableTraceTest.hpp.

8.293.1.45 TEST_F (NumericStateVariableTraceTest , LogicProperty)

Definition at line 393 of file NumericStateVariableTraceTest.hpp.

8.293.1.46 TEST_F (NumericStateVariableTraceTest , MultipleLogicProperties1)

Definition at line 406 of file NumericStateVariableTraceTest.hpp.

8.293.1.47 TEST_F (NumericStateVariableTraceTest , MultipleLogicProperties2)

Definition at line 410 of file NumericStateVariableTraceTest.hpp.

8.293.1.48 TEST_F (NumericStateVariableTraceTest , NextKLogicProperty)

Definition at line 423 of file NumericStateVariableTraceTest.hpp.

8.293.1.49 TEST_F (NumericStateVariableTraceTest , NextLogicProperty)

Definition at line 436 of file NumericStateVariableTraceTest.hpp.

8.293.1.50 TEST_F (NumericStateVariableTraceTest , NotConstraint)

Definition at line 449 of file NumericStateVariableTraceTest.hpp.

8.293.1.51 TEST_F (NumericStateVariableTraceTest , NotLogicProperty)

Definition at line 462 of file NumericStateVariableTraceTest.hpp.

8.293.1.52 TEST_F (NumericStateVariableTraceTest , NumericMeasure)

Definition at line 475 of file NumericStateVariableTraceTest.hpp.

8.293.1.53 TEST_F(NumericStateVariableTraceTest , NumericNumericComparison)

Definition at line 488 of file NumericStateVariableTraceTest.hpp.

8.293.1.54 TEST_F(NumericStateVariableTraceTest , NumericSpatialMeasure)

Definition at line 501 of file NumericStateVariableTraceTest.hpp.

8.293.1.55 TEST_F(NumericStateVariableTraceTest , NumericSpatialNumericComparison)

Definition at line 514 of file NumericStateVariableTraceTest.hpp.

8.293.1.56 TEST_F(NumericStateVariableTraceTest , NumericStateVariable1)

Definition at line 527 of file NumericStateVariableTraceTest.hpp.

8.293.1.57 TEST_F(NumericStateVariableTraceTest , NumericStateVariable2)

Definition at line 531 of file NumericStateVariableTraceTest.hpp.

8.293.1.58 TEST_F(NumericStateVariableTraceTest , NumericStateVariable3)

Definition at line 535 of file NumericStateVariableTraceTest.hpp.

8.293.1.59 TEST_F(NumericStateVariableTraceTest , ProbabilisticLogicProperty)

Definition at line 548 of file NumericStateVariableTraceTest.hpp.

8.293.1.60 TEST_F(NumericStateVariableTraceTest , QuaternarySubsetCovar)

Definition at line 561 of file NumericStateVariableTraceTest.hpp.

8.293.1.61 TEST_F(NumericStateVariableTraceTest , QuaternarySubset)

Definition at line 574 of file NumericStateVariableTraceTest.hpp.

8.293.1.62 TEST_F(NumericStateVariableTraceTest , SpatialMeasureClusteredness)

Definition at line 587 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.63 TEST_F(NumericStateVariableTraceTest , SpatialMeasureDensity) 1317

Definition at line 591 of file NumericStateVariableTraceTest.hpp.

8.293.1.64 TEST_F(NumericStateVariableTraceTest , SpatialMeasureArea)

Definition at line 595 of file NumericStateVariableTraceTest.hpp.

8.293.1.65 TEST_F(NumericStateVariableTraceTest , SpatialMeasurePerimeter)

Definition at line 599 of file NumericStateVariableTraceTest.hpp.

**8.293.1.66 TEST_F(NumericStateVariableTraceTest ,
SpatialMeasureDistanceFromOrigin)**

Definition at line 603 of file NumericStateVariableTraceTest.hpp.

8.293.1.67 TEST_F(NumericStateVariableTraceTest , SpatialMeasureAngle)

Definition at line 607 of file NumericStateVariableTraceTest.hpp.

**8.293.1.68 TEST_F(NumericStateVariableTraceTest , SpatialMeasureTriangleMeasure
)**

Definition at line 611 of file NumericStateVariableTraceTest.hpp.

**8.293.1.69 TEST_F(NumericStateVariableTraceTest , SpatialMeasureRectangleMeasure
)**

Definition at line 615 of file NumericStateVariableTraceTest.hpp.

8.293.1.70 TEST_F(NumericStateVariableTraceTest , SpatialMeasureCircleMeasure)

Definition at line 619 of file NumericStateVariableTraceTest.hpp.

8.293.1.71 TEST_F(NumericStateVariableTraceTest , SpatialMeasureCentroidX)

Definition at line 623 of file NumericStateVariableTraceTest.hpp.

8.293.1.72 TEST_F(NumericStateVariableTraceTest , SpatialMeasureCentroidY)

Definition at line 627 of file NumericStateVariableTraceTest.hpp.

8.293.1.73 TEST_F(NumericStateVariableTraceTest , SubsetOperationDifference)

Definition at line 640 of file NumericStateVariableTraceTest.hpp.

8.293.1.74 TEST_F(NumericStateVariableTraceTest , SubsetOperationIntersection)

Definition at line 644 of file NumericStateVariableTraceTest.hpp.

8.293.1.75 TEST_F(NumericStateVariableTraceTest , SubsetOperationUnion)

Definition at line 648 of file NumericStateVariableTraceTest.hpp.

8.293.1.76 TEST_F(NumericStateVariableTraceTest , SubsetSpecificClusters)

Definition at line 661 of file NumericStateVariableTraceTest.hpp.

8.293.1.77 TEST_F(NumericStateVariableTraceTest , SubsetSpecificRegions)

Definition at line 665 of file NumericStateVariableTraceTest.hpp.

8.293.1.78 TEST_F(NumericStateVariableTraceTest , SubsetSubsetOperation)

Definition at line 678 of file NumericStateVariableTraceTest.hpp.

8.293.1.79 TEST_F(NumericStateVariableTraceTest , Subset)

Definition at line 691 of file NumericStateVariableTraceTest.hpp.

8.293.1.80 TEST_F(NumericStateVariableTraceTest , TernarySubsetMeasurePercentile)

Definition at line 704 of file NumericStateVariableTraceTest.hpp.

8.293.1.81 TEST_F(NumericStateVariableTraceTest , TernarySubsetMeasureQuartile)

Definition at line 708 of file NumericStateVariableTraceTest.hpp.

8.293.1.82 TEST_F(NumericStateVariableTraceTest , TernarySubset)

Definition at line 721 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.83 TEST_F(NumericStateVariableTraceTest , UnaryTypeConstraint) 1319

Definition at line 734 of file NumericStateVariableTraceTest.hpp.

8.293.1.84 TEST_F(NumericStateVariableTraceTest , UnarySpatialConstraint)

Definition at line 747 of file NumericStateVariableTraceTest.hpp.

8.293.1.85 TEST_F(NumericStateVariableTraceTest , UnaryNumericFilter)

Definition at line 760 of file NumericStateVariableTraceTest.hpp.

8.293.1.86 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureAbs)

Definition at line 773 of file NumericStateVariableTraceTest.hpp.

8.293.1.87 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureCeil)

Definition at line 777 of file NumericStateVariableTraceTest.hpp.

8.293.1.88 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureFloor)

Definition at line 781 of file NumericStateVariableTraceTest.hpp.

8.293.1.89 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureRound)

Definition at line 785 of file NumericStateVariableTraceTest.hpp.

8.293.1.90 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureSign)

Definition at line 789 of file NumericStateVariableTraceTest.hpp.

8.293.1.91 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureSqrt)

Definition at line 793 of file NumericStateVariableTraceTest.hpp.

8.293.1.92 TEST_F(NumericStateVariableTraceTest , UnaryNumericMeasureTrunc)

Definition at line 797 of file NumericStateVariableTraceTest.hpp.

8.293.1.93 **TEST_F(NumericStateVariableTraceTest , UnaryNumericNumeric)**

Definition at line 810 of file NumericStateVariableTraceTest.hpp.

8.293.1.94 **TEST_F(NumericStateVariableTraceTest , UnarySubsetMeasureCount)**

Definition at line 823 of file NumericStateVariableTraceTest.hpp.

8.293.1.95 **TEST_F(NumericStateVariableTraceTest ,
UnarySubsetMeasureClusteredness)**

Definition at line 827 of file NumericStateVariableTraceTest.hpp.

8.293.1.96 **TEST_F(NumericStateVariableTraceTest , UnarySubsetMeasureDensity)**

Definition at line 831 of file NumericStateVariableTraceTest.hpp.

8.293.1.97 **TEST_F(NumericStateVariableTraceTest , UnarySubset)**

Definition at line 844 of file NumericStateVariableTraceTest.hpp.

8.293.1.98 **TEST_F(NumericStateVariableTraceTest , UntilLogicProperty)**

Definition at line 857 of file NumericStateVariableTraceTest.hpp.

8.293.1.99 **TEST_F(NumericStateVariableTraceTest , UntilLogicPropertyMultiple)**

Definition at line 861 of file NumericStateVariableTraceTest.hpp.

8.293.1.100 **TEST_F(NumericStateVariableTraceTest , GlobalConstantValueReal)**

Definition at line 874 of file NumericStateVariableTraceTest.hpp.

8.293.1.101 **TEST_F(NumericStateVariableTraceTest ,
GlobalConstantValueNumericStateVariable)**

Definition at line 878 of file NumericStateVariableTraceTest.hpp.

8.293.1.102 **TEST_F(NumericStateVariableTraceTest ,
GlobalConstantValueUnaryNumeric)**

Definition at line 882 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.103 TEST_F (NumericStateVariableTraceTest ,
Reference GlobalConstantValueBinaryNumeric)

1321

Definition at line 886 of file NumericStateVariableTraceTest.hpp.

**8.293.1.104 TEST_F (NumericStateVariableTraceTest ,
GlobalConstantValueUnarySubset)**

Definition at line 890 of file NumericStateVariableTraceTest.hpp.

**8.293.1.105 TEST_F (NumericStateVariableTraceTest ,
GlobalConstantValueBinarySubset)**

Definition at line 894 of file NumericStateVariableTraceTest.hpp.

**8.293.1.106 TEST_F (NumericStateVariableTraceTest ,
GlobalConstantValueTernarySubset)**

Definition at line 898 of file NumericStateVariableTraceTest.hpp.

**8.293.1.107 TEST_F (NumericStateVariableTraceTest ,
GlobalConstantValueQuaternarySubset)**

Definition at line 902 of file NumericStateVariableTraceTest.hpp.

8.293.1.108 TEST_F (NumericStateVariableTraceTest , FutureIncreasingValueReal)

Definition at line 915 of file NumericStateVariableTraceTest.hpp.

**8.293.1.109 TEST_F (NumericStateVariableTraceTest ,
FutureIncreasingValueNumericStateVariable)**

Definition at line 919 of file NumericStateVariableTraceTest.hpp.

**8.293.1.110 TEST_F (NumericStateVariableTraceTest ,
FutureIncreasingValueUnaryNumeric)**

Definition at line 923 of file NumericStateVariableTraceTest.hpp.

**8.293.1.111 TEST_F (NumericStateVariableTraceTest ,
FutureIncreasingValueBinaryNumeric)**

Definition at line 927 of file NumericStateVariableTraceTest.hpp.

8.293.1.112 **TEST_F**(**NumericStateVariableTraceTest** ,
 FutureIncreasingValueUnarySubset)

Definition at line 931 of file NumericStateVariableTraceTest.hpp.

8.293.1.113 **TEST_F**(**NumericStateVariableTraceTest** ,
 FutureIncreasingValueBinarySubset)

Definition at line 935 of file NumericStateVariableTraceTest.hpp.

8.293.1.114 **TEST_F**(**NumericStateVariableTraceTest** ,
 FutureIncreasingValueTernarySubset)

Definition at line 939 of file NumericStateVariableTraceTest.hpp.

8.293.1.115 **TEST_F**(**NumericStateVariableTraceTest** ,
 FutureIncreasingValueQuaternarySubset)

Definition at line 943 of file NumericStateVariableTraceTest.hpp.

8.293.1.116 **TEST_F**(**NumericStateVariableTraceTest** , **GlobalDecreasingValueReal**)

Definition at line 956 of file NumericStateVariableTraceTest.hpp.

8.293.1.117 **TEST_F**(**NumericStateVariableTraceTest** ,
 GlobalDecreasingValueNumericStateVariable)

Definition at line 960 of file NumericStateVariableTraceTest.hpp.

8.293.1.118 **TEST_F**(**NumericStateVariableTraceTest** ,
 GlobalDecreasingValueUnaryNumeric)

Definition at line 964 of file NumericStateVariableTraceTest.hpp.

8.293.1.119 **TEST_F**(**NumericStateVariableTraceTest** ,
 GlobalDecreasingValueBinaryNumeric)

Definition at line 968 of file NumericStateVariableTraceTest.hpp.

8.293.1.120 **TEST_F**(**NumericStateVariableTraceTest** ,
 GlobalDecreasingValueUnarySubset)

Definition at line 972 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/NumericStateVariableTraceTest.hpp File
8.293.1.121 TEST_F (NumericStateVariableTraceTest ,
Reference GlobalDecreasingValueBinarySubset) 1323

Definition at line 976 of file NumericStateVariableTraceTest.hpp.

**8.293.1.122 TEST_F (NumericStateVariableTraceTest ,
GlobalDecreasingValueTernarySubset)**

Definition at line 980 of file NumericStateVariableTraceTest.hpp.

**8.293.1.123 TEST_F (NumericStateVariableTraceTest ,
GlobalDecreasingValueQuaternarySubset)**

Definition at line 984 of file NumericStateVariableTraceTest.hpp.

**8.293.1.124 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueReal)**

Definition at line 997 of file NumericStateVariableTraceTest.hpp.

**8.293.1.125 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueNumericStateVariable)**

Definition at line 1001 of file NumericStateVariableTraceTest.hpp.

**8.293.1.126 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueUnaryNumeric)**

Definition at line 1005 of file NumericStateVariableTraceTest.hpp.

**8.293.1.127 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueBinaryNumeric)**

Definition at line 1009 of file NumericStateVariableTraceTest.hpp.

**8.293.1.128 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueUnarySubset)**

Definition at line 1013 of file NumericStateVariableTraceTest.hpp.

**8.293.1.129 TEST_F (NumericStateVariableTraceTest ,
IncreasingUntilDecreasingValueBinarySubset)**

Definition at line 1017 of file NumericStateVariableTraceTest.hpp.

8.293.1.130 **TEST_F** (**NumericStateVariableTraceTest** ,
 IncreasingUntilDecreasingValueTernarySubset)

Definition at line 1021 of file NumericStateVariableTraceTest.hpp.

8.293.1.131 **TEST_F** (**NumericStateVariableTraceTest** ,
 IncreasingUntilDecreasingValueQuaternarySubset)

Definition at line 1025 of file NumericStateVariableTraceTest.hpp.

8.293.1.132 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueReal)

Definition at line 1038 of file NumericStateVariableTraceTest.hpp.

8.293.1.133 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueNumericStateVariable)

Definition at line 1042 of file NumericStateVariableTraceTest.hpp.

8.293.1.134 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueUnaryNumeric)

Definition at line 1046 of file NumericStateVariableTraceTest.hpp.

8.293.1.135 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueBinaryNumeric)

Definition at line 1050 of file NumericStateVariableTraceTest.hpp.

8.293.1.136 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueUnarySubset)

Definition at line 1054 of file NumericStateVariableTraceTest.hpp.

8.293.1.137 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueBinarySubset)

Definition at line 1058 of file NumericStateVariableTraceTest.hpp.

8.293.1.138 **TEST_F** (**NumericStateVariableTraceTest** ,
 DecreasingUntilIncreasingValueTernarySubset)

Definition at line 1062 of file NumericStateVariableTraceTest.hpp.

8.293

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/NumericStateVariableTraceTest.hpp File

8.293.1.139 **TEST_F (NumericStateVariableTraceTest ,**

1325

DecreasingUntilIncreasingValueQuaternarySubset)

Definition at line 1066 of file NumericStateVariableTraceTest.hpp.

8.293.1.140 **TEST_F (NumericStateVariableTraceTest ,**
OscillationValueNumericStateVariable)

Definition at line 1079 of file NumericStateVariableTraceTest.hpp.

8.293.1.141 **TEST_F (NumericStateVariableTraceTest , OscillationsValueUnaryNumeric**
)

Definition at line 1083 of file NumericStateVariableTraceTest.hpp.

8.293.1.142 **TEST_F (NumericStateVariableTraceTest , OscillationsValueBinaryNumeric**
)

Definition at line 1087 of file NumericStateVariableTraceTest.hpp.

8.293.1.143 **TEST_F (NumericStateVariableTraceTest , OscillationsValueUnarySubset**
)

Definition at line 1091 of file NumericStateVariableTraceTest.hpp.

8.293.1.144 **TEST_F (NumericStateVariableTraceTest , OscillationsValueBinarySubset**
)

Definition at line 1095 of file NumericStateVariableTraceTest.hpp.

8.293.1.145 **TEST_F (NumericStateVariableTraceTest , OscillationsValueTernarySubset**
)

Definition at line 1099 of file NumericStateVariableTraceTest.hpp.

8.293.1.146 **TEST_F (NumericStateVariableTraceTest ,**
OscillationsValueQuaternarySubset)

Definition at line 1103 of file NumericStateVariableTraceTest.hpp.

8.293.1.147 **TEST_F (NumericStateVariableTraceTest ,**
EnclosingWithParenthesesDifferently1)

Definition at line 1116 of file NumericStateVariableTraceTest.hpp.

8.293.1.148 **TEST_F**(**NumericStateVariableTraceTest** ,
 EnclosingWithParenthesesDifferently2)

Definition at line 1120 of file **NumericStateVariableTraceTest.hpp**.

8.293.1.149 **TEST_F**(**NumericStateVariableTraceTest** ,
 TimeIntervalExceedsTraceEndTime)

Definition at line 1133 of file **NumericStateVariableTraceTest.hpp**.

8.293.1.150 **TEST_F**(**NumericStateVariableTraceTest** ,
 TimeIntervalExceedsTraceStartTime)

Definition at line 1137 of file **NumericStateVariableTraceTest.hpp**.

8.293.1.151 **TEST_F**(**NumericStateVariableTraceTest** , **ConstraintsCombinationUnary**
)

Definition at line 1150 of file **NumericStateVariableTraceTest.hpp**.

8.293.1.152 **TEST_F**(**NumericStateVariableTraceTest** , **ConstraintsCombinationBinary**
)

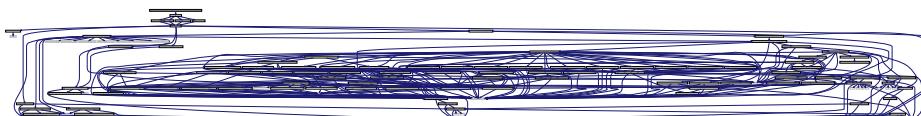
Definition at line 1154 of file **NumericStateVariableTraceTest.hpp**.

8.293.1.153 **TEST_F**(**NumericStateVariableTraceTest** , **ConstraintsCombinationNary**)

Definition at line 1158 of file **NumericStateVariableTraceTest.hpp**.

8.294 /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:



8.295

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/ParserEvaluationTest.hpp File

Functions

Reference

1327

- int [main](#) (int argc, char **argv)

8.294.1 Function Documentation

8.294.1.1 int main (int argc, char ** argv)

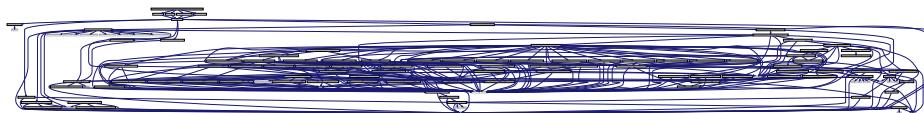
Definition at line 4 of file ParserEvaluationTest.cpp.

8.295 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/ParserEvaluationTest.hpp File -

Reference

```
#include "CompleteTraceTest.hpp"      #include "EmptyTrace-
Test.hpp"    #include "NumericStateVariableTraceTest.hpp" ×
#include "SpatialEntitiesTraceTest.hpp" Include dependency graph
for ParserEvaluationTest.hpp:
```

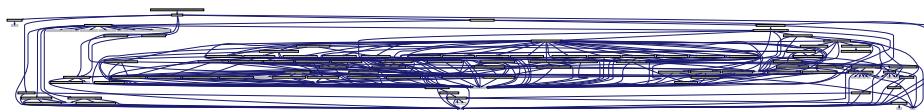


8.296 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File

Reference

```
#include "TraceEvaluationTest.hpp" Include dependency graph for -
SpatialEntitiesTraceTest.hpp:
```



Classes

- class [multiscaletest::SpatialEntitiesTraceTest](#)

Class for testing evaluation of spatial entities-only traces.

Namespaces

- namespace [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 \(SpatialEntitiesTraceTest, BinarySubsetMeasureAvg\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureGeomean\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureHarmean\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureKurt\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureMax\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureMedian\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureMin\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureMode\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureProduct\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureSkew\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureStdev\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureSum\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubsetMeasureVar\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, BinarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, ComparatorGreaterThan\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, ComparatorLessThan\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, ComparatorGreaterThanOrEqual\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, ComparatorLessThanOrEqual\)](#)
- [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 \(SpatialEntitiesTraceTest, ConstraintEnclosedByParenthesesQuadrupled\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, Constraint\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, Difference\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FilterNumericMeasure\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FilterSubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureLogicProperty\)](#)

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File

Reference TEST_F (SpatialEntitiesTraceTest, GlobalLogicProperty)

1329

- TEST_F (SpatialEntitiesTraceTest, LogicPropertyEnclosedByParentheses)
- TEST_F (SpatialEntitiesTraceTest, LogicPropertyEnclosedByParentheses-Doubled)
- TEST_F (SpatialEntitiesTraceTest, LogicPropertyEnclosedByParentheses-Quadrupled)
- 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, NumericNumericComparison)
- TEST_F (SpatialEntitiesTraceTest, NumericSpatialMeasure)
- TEST_F (SpatialEntitiesTraceTest, NumericSpatialNumericComparison)
- TEST_F (SpatialEntitiesTraceTest, NumericStateVariable1)
- TEST_F (SpatialEntitiesTraceTest, NumericStateVariable2)
- TEST_F (SpatialEntitiesTraceTest, NumericStateVariable3)
- TEST_F (SpatialEntitiesTraceTest, ProbabilisticLogicProperty)
- TEST_F (SpatialEntitiesTraceTest, QuaternarySubsetCovar)
- TEST_F (SpatialEntitiesTraceTest, QuaternarySubset)
- 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, SubsetOperationDifference)
- TEST_F (SpatialEntitiesTraceTest, SubsetOperationIntersection)
- TEST_F (SpatialEntitiesTraceTest, SubsetOperationUnion)
- TEST_F (SpatialEntitiesTraceTest, SubsetSpecificClusters)
- TEST_F (SpatialEntitiesTraceTest, SubsetSpecificRegions)
- TEST_F (SpatialEntitiesTraceTest, SubsetSubsetOperation)
- TEST_F (SpatialEntitiesTraceTest, Subset)
- TEST_F (SpatialEntitiesTraceTest, TernarySubsetMeasurePercentile)
- TEST_F (SpatialEntitiesTraceTest, TernarySubsetMeasureQuartile)
- TEST_F (SpatialEntitiesTraceTest, TernarySubset)
- TEST_F (SpatialEntitiesTraceTest, UnaryTypeConstraint)
- TEST_F (SpatialEntitiesTraceTest, UnarySpatialConstraint)

- [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 \(SpatialEntitiesTraceTest, UnarySubsetMeasureCount\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, UnarySubsetMeasureClusteredness\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, UnarySubsetMeasureDensity\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, UnarySubset\)](#)
- [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 \(SpatialEntitiesTraceTest, GlobalConstantValueUnarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalConstantValueBinarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalConstantValueTernarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalConstantValueQuaternarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueReal\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueNumericStateVariable\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueUnaryNumeric\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueBinaryNumeric\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueUnarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueBinarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueTernarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, FutureIncreasingValueQuaternarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueReal\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueNumericStateVariable\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueBinaryNumeric\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueUnarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueBinarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueTernarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, GlobalDecreasingValueQuaternarySubset\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueReal\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueNumericStateVariable\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueUnaryNumeric\)](#)
- [TEST_F \(SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueBinaryNumeric\)](#)

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File

Reference	TEST_F (SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueUnarySubset)	1331
	• TEST_F (SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueBinarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueTernarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, IncreasingUntilDecreasingValueQuaternarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueReal)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueNumericStateVariable)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueUnaryNumeric)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueBinaryNumeric)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueUnarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueBinarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueTernarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, DecreasingUntilIncreasingValueQuaternarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationValueNumericStateVariable)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueUnaryNumeric)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueBinaryNumeric)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueUnarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueBinarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueTernarySubset)	
	• TEST_F (SpatialEntitiesTraceTest, OscillationsValueQuaternarySubset)	
	• 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)	

8.296.1 Function Documentation

8.296.1.1 TEST_F (SpatialEntitiesTraceTest , BinaryNumericFilter)

Definition at line 91 of file SpatialEntitiesTraceTest.hpp.

8.296.1.2 TEST_F (SpatialEntitiesTraceTest , BinaryNumericMeasureAdd)

Definition at line 104 of file SpatialEntitiesTraceTest.hpp.

8.296.1.3 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasureDiv)

Definition at line 108 of file SpatialEntitiesTraceTest.hpp.

8.296.1.4 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasureLog)

Definition at line 112 of file SpatialEntitiesTraceTest.hpp.

8.296.1.5 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasureMod)

Definition at line 116 of file SpatialEntitiesTraceTest.hpp.

8.296.1.6 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasureMultiply)

Definition at line 120 of file SpatialEntitiesTraceTest.hpp.

8.296.1.7 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasurePower)

Definition at line 124 of file SpatialEntitiesTraceTest.hpp.

8.296.1.8 TEST_F(SpatialEntitiesTraceTest , BinaryNumericMeasureSubtract)

Definition at line 128 of file SpatialEntitiesTraceTest.hpp.

8.296.1.9 TEST_F(SpatialEntitiesTraceTest , BinaryNumericNumeric)

Definition at line 141 of file SpatialEntitiesTraceTest.hpp.

8.296.1.10 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureAvg)

Definition at line 154 of file SpatialEntitiesTraceTest.hpp.

8.296.1.11 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureGeomean)

Definition at line 158 of file SpatialEntitiesTraceTest.hpp.

8.296.1.12 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureHarmean)

Definition at line 162 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.13 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureKurt) 1333

Definition at line 166 of file SpatialEntitiesTraceTest.hpp.

8.296.1.14 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureMax)

Definition at line 170 of file SpatialEntitiesTraceTest.hpp.

8.296.1.15 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureMedian)

Definition at line 174 of file SpatialEntitiesTraceTest.hpp.

8.296.1.16 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureMin)

Definition at line 178 of file SpatialEntitiesTraceTest.hpp.

8.296.1.17 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureMode)

Definition at line 182 of file SpatialEntitiesTraceTest.hpp.

8.296.1.18 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureProduct)

Definition at line 186 of file SpatialEntitiesTraceTest.hpp.

8.296.1.19 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureSkew)

Definition at line 190 of file SpatialEntitiesTraceTest.hpp.

8.296.1.20 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureStdev)

Definition at line 194 of file SpatialEntitiesTraceTest.hpp.

8.296.1.21 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureSum)

Definition at line 198 of file SpatialEntitiesTraceTest.hpp.

8.296.1.22 TEST_F(SpatialEntitiesTraceTest , BinarySubsetMeasureVar)

Definition at line 202 of file SpatialEntitiesTraceTest.hpp.

8.296.1.23 TEST_F(SpatialEntitiesTraceTest , BinarySubset)

Definition at line 215 of file SpatialEntitiesTraceTest.hpp.

8.296.1.24 TEST_F(SpatialEntitiesTraceTest , ComparatorGreaterThan)

Definition at line 228 of file SpatialEntitiesTraceTest.hpp.

8.296.1.25 TEST_F(SpatialEntitiesTraceTest , ComparatorLessThan)

Definition at line 232 of file SpatialEntitiesTraceTest.hpp.

8.296.1.26 TEST_F(SpatialEntitiesTraceTest , ComparatorGreaterThanOrEqual)

Definition at line 236 of file SpatialEntitiesTraceTest.hpp.

8.296.1.27 TEST_F(SpatialEntitiesTraceTest , ComparatorLessThanOrEqual)

Definition at line 240 of file SpatialEntitiesTraceTest.hpp.

8.296.1.28 TEST_F(SpatialEntitiesTraceTest , ComparatorEqual)

Definition at line 244 of file SpatialEntitiesTraceTest.hpp.

8.296.1.29 TEST_F(SpatialEntitiesTraceTest , CompoundConstraint)

Definition at line 257 of file SpatialEntitiesTraceTest.hpp.

8.296.1.30 TEST_F(SpatialEntitiesTraceTest , CompoundConstraintMultiple)

Definition at line 264 of file SpatialEntitiesTraceTest.hpp.

8.296.1.31 TEST_F(SpatialEntitiesTraceTest , CompoundLogicProperty)

Definition at line 280 of file SpatialEntitiesTraceTest.hpp.

8.296.1.32 TEST_F(SpatialEntitiesTraceTest , CompoundLogicPropertyMultiple)

Definition at line 287 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.33 TEST_F(SpatialEntitiesTraceTest , ConstraintEnclosedByParentheses) 1335

Definition at line 303 of file SpatialEntitiesTraceTest.hpp.

8.296.1.34 TEST_F(SpatialEntitiesTraceTest , ConstraintEnclosedByParentheses-Doubled)

Definition at line 307 of file SpatialEntitiesTraceTest.hpp.

8.296.1.35 TEST_F(SpatialEntitiesTraceTest , ConstraintEnclosedByParentheses-Quadrupled)

Definition at line 311 of file SpatialEntitiesTraceTest.hpp.

8.296.1.36 TEST_F(SpatialEntitiesTraceTest , Constraint)

Definition at line 324 of file SpatialEntitiesTraceTest.hpp.

8.296.1.37 TEST_F(SpatialEntitiesTraceTest , Difference)

Definition at line 337 of file SpatialEntitiesTraceTest.hpp.

8.296.1.38 TEST_F(SpatialEntitiesTraceTest , FilterNumericMeasure)

Definition at line 350 of file SpatialEntitiesTraceTest.hpp.

8.296.1.39 TEST_F(SpatialEntitiesTraceTest , FilterSubset)

Definition at line 363 of file SpatialEntitiesTraceTest.hpp.

8.296.1.40 TEST_F(SpatialEntitiesTraceTest , FutureLogicProperty)

Definition at line 376 of file SpatialEntitiesTraceTest.hpp.

8.296.1.41 TEST_F(SpatialEntitiesTraceTest , GlobalLogicProperty)

Definition at line 389 of file SpatialEntitiesTraceTest.hpp.

8.296.1.42 TEST_F(SpatialEntitiesTraceTest , LogicPropertyEnclosedByParentheses)

Definition at line 402 of file SpatialEntitiesTraceTest.hpp.

8.296.1.43 **TEST_F(SpatialEntitiesTraceTest , LogicPropertyEnclosedByParentheses-Doubled)**

Definition at line 406 of file SpatialEntitiesTraceTest.hpp.

8.296.1.44 **TEST_F(SpatialEntitiesTraceTest , LogicPropertyEnclosedByParentheses-Quadrupled)**

Definition at line 410 of file SpatialEntitiesTraceTest.hpp.

8.296.1.45 **TEST_F(SpatialEntitiesTraceTest , LogicProperty)**

Definition at line 423 of file SpatialEntitiesTraceTest.hpp.

8.296.1.46 **TEST_F(SpatialEntitiesTraceTest , MultipleLogicProperties1)**

Definition at line 436 of file SpatialEntitiesTraceTest.hpp.

8.296.1.47 **TEST_F(SpatialEntitiesTraceTest , MultipleLogicProperties2)**

Definition at line 440 of file SpatialEntitiesTraceTest.hpp.

8.296.1.48 **TEST_F(SpatialEntitiesTraceTest , NextKLogicProperty)**

Definition at line 453 of file SpatialEntitiesTraceTest.hpp.

8.296.1.49 **TEST_F(SpatialEntitiesTraceTest , NextLogicProperty)**

Definition at line 466 of file SpatialEntitiesTraceTest.hpp.

8.296.1.50 **TEST_F(SpatialEntitiesTraceTest , NotConstraint)**

Definition at line 479 of file SpatialEntitiesTraceTest.hpp.

8.296.1.51 **TEST_F(SpatialEntitiesTraceTest , NotLogicProperty)**

Definition at line 492 of file SpatialEntitiesTraceTest.hpp.

8.296.1.52 **TEST_F(SpatialEntitiesTraceTest , NumericMeasure)**

Definition at line 505 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.53 TEST_F(SpatialEntitiesTraceTest , NumericNumericComparison) **1337**

Definition at line 518 of file SpatialEntitiesTraceTest.hpp.

8.296.1.54 TEST_F(SpatialEntitiesTraceTest , NumericSpatialMeasure)

Definition at line 531 of file SpatialEntitiesTraceTest.hpp.

8.296.1.55 TEST_F(SpatialEntitiesTraceTest , NumericSpatialNumericComparison)

Definition at line 544 of file SpatialEntitiesTraceTest.hpp.

8.296.1.56 TEST_F(SpatialEntitiesTraceTest , NumericStateVariable1)

Definition at line 557 of file SpatialEntitiesTraceTest.hpp.

8.296.1.57 TEST_F(SpatialEntitiesTraceTest , NumericStateVariable2)

Definition at line 561 of file SpatialEntitiesTraceTest.hpp.

8.296.1.58 TEST_F(SpatialEntitiesTraceTest , NumericStateVariable3)

Definition at line 565 of file SpatialEntitiesTraceTest.hpp.

8.296.1.59 TEST_F(SpatialEntitiesTraceTest , ProbabilisticLogicProperty)

Definition at line 578 of file SpatialEntitiesTraceTest.hpp.

8.296.1.60 TEST_F(SpatialEntitiesTraceTest , QuaternarySubsetCovar)

Definition at line 591 of file SpatialEntitiesTraceTest.hpp.

8.296.1.61 TEST_F(SpatialEntitiesTraceTest , QuaternarySubset)

Definition at line 604 of file SpatialEntitiesTraceTest.hpp.

8.296.1.62 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureClusteredness)

Definition at line 617 of file SpatialEntitiesTraceTest.hpp.

8.296.1.63 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureDensity)

Definition at line 621 of file SpatialEntitiesTraceTest.hpp.

8.296.1.64 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureArea)

Definition at line 625 of file SpatialEntitiesTraceTest.hpp.

8.296.1.65 TEST_F(SpatialEntitiesTraceTest , SpatialMeasurePerimeter)

Definition at line 629 of file SpatialEntitiesTraceTest.hpp.

8.296.1.66 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureDistanceFromOrigin)

Definition at line 633 of file SpatialEntitiesTraceTest.hpp.

8.296.1.67 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureAngle)

Definition at line 637 of file SpatialEntitiesTraceTest.hpp.

8.296.1.68 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureTriangleMeasure)

Definition at line 641 of file SpatialEntitiesTraceTest.hpp.

8.296.1.69 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureRectangleMeasure)

Definition at line 645 of file SpatialEntitiesTraceTest.hpp.

8.296.1.70 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureCircleMeasure)

Definition at line 649 of file SpatialEntitiesTraceTest.hpp.

8.296.1.71 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureCentroidX)

Definition at line 653 of file SpatialEntitiesTraceTest.hpp.

8.296.1.72 TEST_F(SpatialEntitiesTraceTest , SpatialMeasureCentroidY)

Definition at line 657 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.73 TEST_F(SpatialEntitiesTraceTest , SubsetOperationDifference) **1339**

Definition at line 670 of file SpatialEntitiesTraceTest.hpp.

8.296.1.74 TEST_F(SpatialEntitiesTraceTest , SubsetOperationIntersection)

Definition at line 674 of file SpatialEntitiesTraceTest.hpp.

8.296.1.75 TEST_F(SpatialEntitiesTraceTest , SubsetOperationUnion)

Definition at line 678 of file SpatialEntitiesTraceTest.hpp.

8.296.1.76 TEST_F(SpatialEntitiesTraceTest , SubsetSpecificClusters)

Definition at line 691 of file SpatialEntitiesTraceTest.hpp.

8.296.1.77 TEST_F(SpatialEntitiesTraceTest , SubsetSpecificRegions)

Definition at line 695 of file SpatialEntitiesTraceTest.hpp.

8.296.1.78 TEST_F(SpatialEntitiesTraceTest , SubsetSubsetOperation)

Definition at line 708 of file SpatialEntitiesTraceTest.hpp.

8.296.1.79 TEST_F(SpatialEntitiesTraceTest , Subset)

Definition at line 721 of file SpatialEntitiesTraceTest.hpp.

8.296.1.80 TEST_F(SpatialEntitiesTraceTest , TernarySubsetMeasurePercentile)

Definition at line 734 of file SpatialEntitiesTraceTest.hpp.

8.296.1.81 TEST_F(SpatialEntitiesTraceTest , TernarySubsetMeasureQuartile)

Definition at line 738 of file SpatialEntitiesTraceTest.hpp.

8.296.1.82 TEST_F(SpatialEntitiesTraceTest , TernarySubset)

Definition at line 751 of file SpatialEntitiesTraceTest.hpp.

8.296.1.83 TEST_F(SpatialEntitiesTraceTest , UnaryTypeConstraint)

Definition at line 764 of file SpatialEntitiesTraceTest.hpp.

8.296.1.84 TEST_F(SpatialEntitiesTraceTest , UnarySpatialConstraint)

Definition at line 777 of file SpatialEntitiesTraceTest.hpp.

8.296.1.85 TEST_F(SpatialEntitiesTraceTest , UnaryNumericFilter)

Definition at line 790 of file SpatialEntitiesTraceTest.hpp.

8.296.1.86 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureAbs)

Definition at line 803 of file SpatialEntitiesTraceTest.hpp.

8.296.1.87 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureCeil)

Definition at line 807 of file SpatialEntitiesTraceTest.hpp.

8.296.1.88 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureFloor)

Definition at line 811 of file SpatialEntitiesTraceTest.hpp.

8.296.1.89 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureRound)

Definition at line 815 of file SpatialEntitiesTraceTest.hpp.

8.296.1.90 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureSign)

Definition at line 819 of file SpatialEntitiesTraceTest.hpp.

8.296.1.91 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureSqrt)

Definition at line 823 of file SpatialEntitiesTraceTest.hpp.

8.296.1.92 TEST_F(SpatialEntitiesTraceTest , UnaryNumericMeasureTrunc)

Definition at line 827 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.93 TEST_F(SpatialEntitiesTraceTest , UnaryNumericNumeric)

1341

Definition at line 840 of file SpatialEntitiesTraceTest.hpp.

8.296.1.94 TEST_F(SpatialEntitiesTraceTest , UnarySubsetMeasureCount)

Definition at line 853 of file SpatialEntitiesTraceTest.hpp.

8.296.1.95 TEST_F(SpatialEntitiesTraceTest , UnarySubsetMeasureClusteredness)

Definition at line 857 of file SpatialEntitiesTraceTest.hpp.

8.296.1.96 TEST_F(SpatialEntitiesTraceTest , UnarySubsetMeasureDensity)

Definition at line 861 of file SpatialEntitiesTraceTest.hpp.

8.296.1.97 TEST_F(SpatialEntitiesTraceTest , UnarySubset)

Definition at line 874 of file SpatialEntitiesTraceTest.hpp.

8.296.1.98 TEST_F(SpatialEntitiesTraceTest , UntilLogicProperty)

Definition at line 887 of file SpatialEntitiesTraceTest.hpp.

8.296.1.99 TEST_F(SpatialEntitiesTraceTest , UntilLogicPropertyMultiple)

Definition at line 891 of file SpatialEntitiesTraceTest.hpp.

8.296.1.100 TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueReal)

Definition at line 904 of file SpatialEntitiesTraceTest.hpp.

8.296.1.101 TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueNumericState-Variable)

Definition at line 908 of file SpatialEntitiesTraceTest.hpp.

8.296.1.102 TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueUnaryNumeric)

Definition at line 912 of file SpatialEntitiesTraceTest.hpp.

8.296.1.103 **TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueBinaryNumeric)**

Definition at line 916 of file SpatialEntitiesTraceTest.hpp.

8.296.1.104 **TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueUnarySubset)**

Definition at line 920 of file SpatialEntitiesTraceTest.hpp.

8.296.1.105 **TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueBinarySubset)**

Definition at line 924 of file SpatialEntitiesTraceTest.hpp.

8.296.1.106 **TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueTernarySubset)**

Definition at line 928 of file SpatialEntitiesTraceTest.hpp.

8.296.1.107 **TEST_F(SpatialEntitiesTraceTest , GlobalConstantValueQuaternarySubset)**

Definition at line 932 of file SpatialEntitiesTraceTest.hpp.

8.296.1.108 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueReal)**

Definition at line 945 of file SpatialEntitiesTraceTest.hpp.

8.296.1.109 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueNumericState-Variable)**

Definition at line 949 of file SpatialEntitiesTraceTest.hpp.

8.296.1.110 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueUnaryNumeric)**

Definition at line 953 of file SpatialEntitiesTraceTest.hpp.

8.296.1.111 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueBinaryNumeric)**

Definition at line 957 of file SpatialEntitiesTraceTest.hpp.

8.296.1.112 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueUnarySubset)**

Definition at line 961 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-

temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File

8.296.1.113 **TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueBinarySubset)**

1343

Definition at line 965 of file SpatialEntitiesTraceTest.hpp.

8.296.1.114 TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueTernarySubset)

Definition at line 969 of file SpatialEntitiesTraceTest.hpp.

8.296.1.115 TEST_F(SpatialEntitiesTraceTest , FutureIncreasingValueQuaternarySubset)

Definition at line 973 of file SpatialEntitiesTraceTest.hpp.

8.296.1.116 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueReal)

Definition at line 986 of file SpatialEntitiesTraceTest.hpp.

8.296.1.117 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueNumericState-Variable)

Definition at line 990 of file SpatialEntitiesTraceTest.hpp.

8.296.1.118 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueUnaryNumeric)

Definition at line 994 of file SpatialEntitiesTraceTest.hpp.

8.296.1.119 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueBinaryNumeric)

Definition at line 998 of file SpatialEntitiesTraceTest.hpp.

8.296.1.120 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueUnarySubset)

Definition at line 1002 of file SpatialEntitiesTraceTest.hpp.

8.296.1.121 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueBinarySubset)

Definition at line 1006 of file SpatialEntitiesTraceTest.hpp.

8.296.1.122 TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueTernarySubset)

Definition at line 1010 of file SpatialEntitiesTraceTest.hpp.

8.296.1.123 **TEST_F(SpatialEntitiesTraceTest , GlobalDecreasingValueQuaternarySubset)**

Definition at line 1014 of file SpatialEntitiesTraceTest.hpp.

8.296.1.124 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueReal)**

Definition at line 1027 of file SpatialEntitiesTraceTest.hpp.

8.296.1.125 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueNumericStateVariable)**

Definition at line 1031 of file SpatialEntitiesTraceTest.hpp.

8.296.1.126 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueUnaryNumeric)**

Definition at line 1035 of file SpatialEntitiesTraceTest.hpp.

8.296.1.127 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueBinaryNumeric)**

Definition at line 1039 of file SpatialEntitiesTraceTest.hpp.

8.296.1.128 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueUnarySubset)**

Definition at line 1043 of file SpatialEntitiesTraceTest.hpp.

8.296.1.129 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueBinarySubset)**

Definition at line 1047 of file SpatialEntitiesTraceTest.hpp.

8.296.1.130 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueTernarySubset)**

Definition at line 1051 of file SpatialEntitiesTraceTest.hpp.

8.296.1.131 **TEST_F(SpatialEntitiesTraceTest , IncreasingUntilDecreasingValueQuaternarySubset)**

Definition at line 1055 of file SpatialEntitiesTraceTest.hpp.

8.296

/home/ovidiu/Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/SpatialEntitiesTraceTest.hpp File
8.296.1.132 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueReal)
Reference 1345

Definition at line 1068 of file SpatialEntitiesTraceTest.hpp.

8.296.1.133 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueNumericStateVariable)

Definition at line 1072 of file SpatialEntitiesTraceTest.hpp.

8.296.1.134 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueUnaryNumeric)

Definition at line 1076 of file SpatialEntitiesTraceTest.hpp.

8.296.1.135 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueBinaryNumeric)

Definition at line 1080 of file SpatialEntitiesTraceTest.hpp.

8.296.1.136 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueUnarySubset)

Definition at line 1084 of file SpatialEntitiesTraceTest.hpp.

8.296.1.137 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueBinarySubset)

Definition at line 1088 of file SpatialEntitiesTraceTest.hpp.

8.296.1.138 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueTernarySubset)

Definition at line 1092 of file SpatialEntitiesTraceTest.hpp.

8.296.1.139 TEST_F(SpatialEntitiesTraceTest , DecreasingUntilIncreasingValueQuaternarySubset)

Definition at line 1096 of file SpatialEntitiesTraceTest.hpp.

8.296.1.140 TEST_F(SpatialEntitiesTraceTest , OscillationValueNumericStateVariable)

Definition at line 1109 of file SpatialEntitiesTraceTest.hpp.

8.296.1.141 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueUnaryNumeric)**

Definition at line 1113 of file SpatialEntitiesTraceTest.hpp.

8.296.1.142 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueBinaryNumeric)**

Definition at line 1117 of file SpatialEntitiesTraceTest.hpp.

8.296.1.143 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueUnarySubset)**

Definition at line 1121 of file SpatialEntitiesTraceTest.hpp.

8.296.1.144 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueBinarySubset)**

Definition at line 1125 of file SpatialEntitiesTraceTest.hpp.

8.296.1.145 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueTernarySubset)**

Definition at line 1129 of file SpatialEntitiesTraceTest.hpp.

8.296.1.146 **TEST_F(SpatialEntitiesTraceTest , OscillationsValueQuaternarySubset)**

Definition at line 1133 of file SpatialEntitiesTraceTest.hpp.

8.296.1.147 **TEST_F(SpatialEntitiesTraceTest , EnclosingWithParenthesesDifferently1)**

Definition at line 1146 of file SpatialEntitiesTraceTest.hpp.

8.296.1.148 **TEST_F(SpatialEntitiesTraceTest , EnclosingWithParenthesesDifferently2)**

Definition at line 1150 of file SpatialEntitiesTraceTest.hpp.

8.296.1.149 **TEST_F(SpatialEntitiesTraceTest , TimeIntervalExceedsTraceEndTime)**

Definition at line 1163 of file SpatialEntitiesTraceTest.hpp.

8.296.1.150 **TEST_F(SpatialEntitiesTraceTest , TimeIntervalExceedsTraceStartTime)**

Definition at line 1167 of file SpatialEntitiesTraceTest.hpp.

8.297

/home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/evaluation/TraceEvaluationTest.hpp File
8.296.1.151 TEST_F(SpatialEntitiesTraceTest , ConstraintsCombinationUnary) 1347

Definition at line 1180 of file SpatialEntitiesTraceTest.hpp.

8.296.1.152 TEST_F(SpatialEntitiesTraceTest , ConstraintsCombinationBinary)

Definition at line 1184 of file SpatialEntitiesTraceTest.hpp.

8.296.1.153 TEST_F(SpatialEntitiesTraceTest , ConstraintsCombinationNary)

Definition at line 1188 of file SpatialEntitiesTraceTest.hpp.

8.297 /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/verification/spatial-t
Cluster.hpp" #include "multiscale/verification/spatial-temporal/model/-
Region.hpp" #include "multiscale/verification/spatial-temporal/parsing/-
Parser.hpp" #include <string> Include dependency graph for Trace-
EvaluationTest.hpp:
```



Classes

- class [multiscaletest::TraceEvaluationTest](#)

Class for testing evaluation of traces.

Namespaces

- namespace [multiscaletest](#)

Variables

- static const std::string [ERR_MSG_TEST](#) = "The given input string could not be successfully parsed."

8.297.1 Variable Documentation

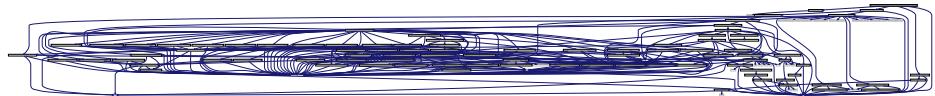
8.297.1.1 `const std::string ERR_MSG_TEST = "The given input string could not be successfully parsed." [static]`

Definition at line 14 of file TraceEvaluationTest.hpp.

Referenced by `multiscaletest::TraceEvaluationTest::RunTest()`.

8.298 /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

- namespace `multiscaletest`
- namespace `multiscaletest::verification`

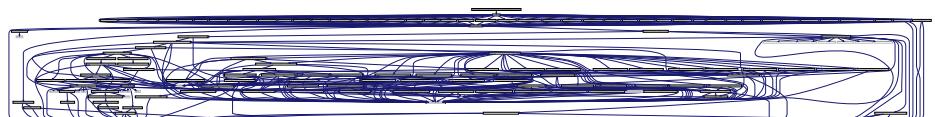
Functions

- bool `multiscaletest::verification::parseInputString (const std::string &inputString)`

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

8.299 /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:
```



- int **main** (int argc, char **argv)

8.299.1 Function Documentation

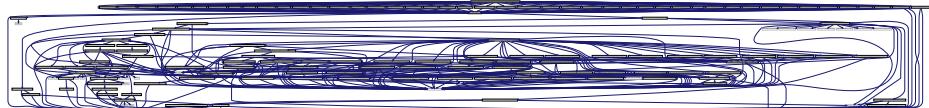
8.299.1.1 int main (int argc, char ** argv)

Definition at line 6 of file ParserTest.cpp.

8.300 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/verification/spatial-temporal/test/parsing/ParserTest.hpp File Reference

```
#include "parsing/InputStringParser.hpp" #include "attribute/-  
BinaryNumericFilterTest.hpp" #include "attribute/Binary-  
NumericMeasureTest.hpp" #include "attribute/BinaryNumeric-  
NumericTest.hpp" #include "attribute/BinarySubsetMeasure-  
Test.hpp" #include "attribute/BinarySubsetTest.hpp" #include  
"attribute/ComparatorTest.hpp" #include "attribute/Compound-  
ConstraintTest.hpp" #include "attribute/CompoundLogic-  
PropertyTest.hpp" #include "attribute/ConstraintParentheses-  
Test.hpp" #include "attribute/ConstraintTest.hpp" #include  
"attribute/DifferenceTest.hpp" #include "attribute/Filter-  
NumericMeasureTest.hpp" #include "attribute/FilterSubset-  
Test.hpp" #include "attribute/FutureLogicPropertyTest.-  
hpp" #include "attribute/GlobalLogicPropertyTest.hpp" ×  
#include "attribute/LogicPropertyParenthesesTest.hpp"  
#include "attribute/LogicPropertyTest.hpp" #include "attribute/-  
MultipleLogicPropertiesTest.hpp" #include "attribute/-  
NextKLogicPropertyTest.hpp" #include "attribute/Next-  
LogicPropertyTest.hpp" #include "attribute/NotConstraint-  
Test.hpp" #include "attribute/NotLogicPropertyTest.hpp"  
#include "attribute/NumericMeasureTest.hpp" #include "attribute/-  
NumericNumericComparisonTest.hpp" #include "attribute/-  
NumericSpatialMeasureTest.hpp" #include "attribute/Numeric-  
SpatialNumericComparisonTest.hpp" #include "attribute/-  
NumericStateVariableTest.hpp" #include "attribute/Probabilistic-  
LogicPropertyTest.hpp" #include "attribute/Quaternary-  
SubsetMeasureTest.hpp" #include "attribute/Quaternary-  
SubsetTest.hpp" #include "attribute/SpatialMeasureTest.-  
hpp" #include "attribute/SubsetOperationTest.hpp" #include  
"attribute/SubsetSpecificTest.hpp" #include "attribute/-  
SubsetSubsetOperationTest.hpp" #include "attribute/Subset-  
Test.hpp" #include "attribute/TernarySubsetMeasureTest.-  
hpp" #include "attribute/TernarySubsetTest.hpp" #include  
"attribute/UnaryTypeConstraintTest.hpp" #include "attribute/-
```

```
UnarySpatialConstraintTest.hpp" #include "attribute/Unary-
NumericFilterTest.hpp" #include "attribute/UnaryNumeric-
MeasureTest.hpp" #include "attribute/UnaryNumericNumeric-
Test.hpp" #include "attribute/UnarySubsetMeasureTest.-
hpp" #include "attribute/UnarySubsetTest.hpp" #include
"attribute/UntilLogicPropertyTest.hpp" Include dependency graph
for ParserTest.hpp:
```



Functions

- [TEST \(Input, IncorrectTrueInput\)](#)
- [TEST \(Input, IncorrectTInput\)](#)
- [TEST \(Input, IncorrectFalseInput\)](#)
- [TEST \(Input, IncorrectFInput\)](#)

8.300.1 Function Documentation

8.300.1.1 TEST (Input , IncorrectTrueInput)

Definition at line 21 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.300.1.2 TEST (Input , IncorrectTInput)

Definition at line 25 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.300.1.3 TEST (Input , IncorrectFalseInput)

Definition at line 29 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.300.1.4 TEST (Input , IncorrectFInput)

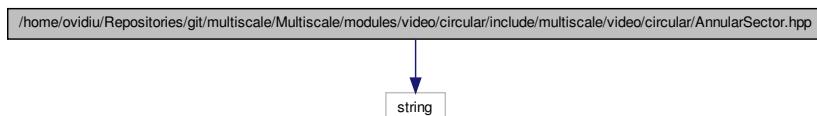
Definition at line 33 of file ParserTest.hpp.

References multiscaletest::verification::parseInputString().

8.301 /home/ovidiu.Repositories/git/multiscale-/
Multiscale/modules/video/circular/include/multiscale/video/circular/Annular-
Sector.hpp File Reference

AnnularSector.hpp File Reference

#include <string> Include dependency graph for AnnularSector.hpp:



Classes

- class [multiscale::video::AnnularSector](#)

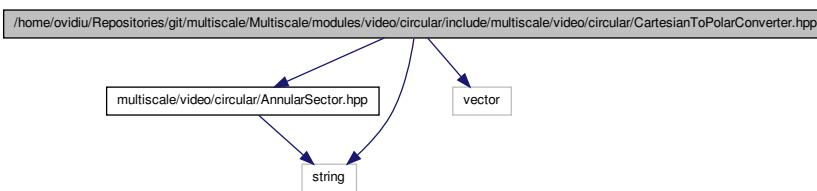
An annular sector is the basic element in the considered circular geometry.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.302 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/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](#)

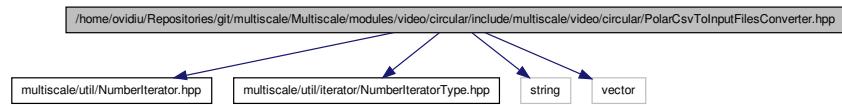
Converter from the rectangular geometry grid cells to annular sectors.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.303 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/PolarCsvToInputFilesConverter.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"      #include
"multiscale/util/iterator/NumberIteratorType.hpp" #include
<string> #include <vector> Include dependency graph for PolarCsvTo-
InputFilesConverter.hpp:
```



Classes

- class [multiscale::video::PolarCsvToInputFilesConverter](#)

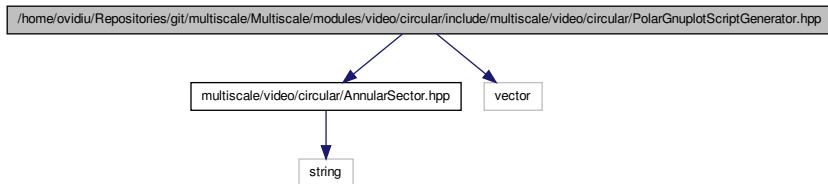
Csv file to input file converter considering polar coordinates.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.304 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/include/multiscale/video/circular/AnnularSector.hpp File Reference

```
#include "multiscale/video/circular/AnnularSector.hpp" x
#include <vector> Include dependency graph for PolarGnuplotScript-
```



Classes

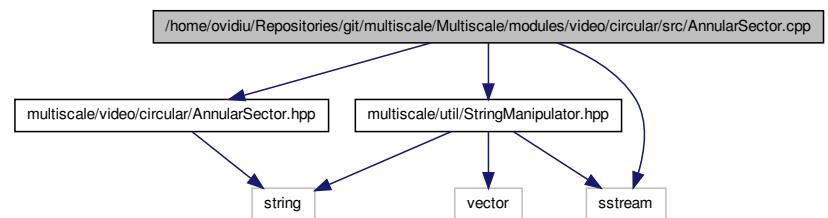
- class [multiscale::video::PolarGnuplotScriptGenerator](#)
Gnuplot script generator from the provided annular sectors.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.305 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- AnnularSector.cpp File Reference

```
#include "multiscale/video/circular/AnnularSector.hpp" x
#include "multiscale/util/StringManipulator.hpp" #include
<sstream> Include dependency graph for AnnularSector.cpp:
```



Variables

- const string [SEPARATOR = " "](#)

8.305.1 Variable Documentation

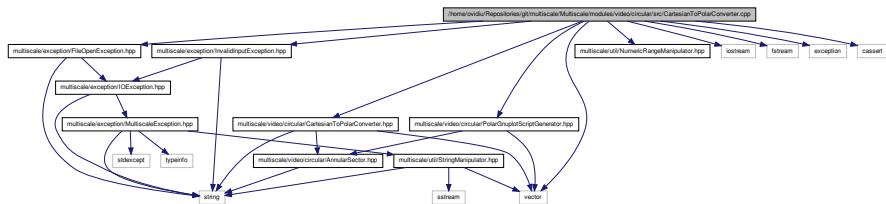
8.305.1.1 const string SEPARATOR = " "

Definition at line 6 of file AnnularSector.cpp.

Referenced by multiscale::video::AnnularSector::toString().

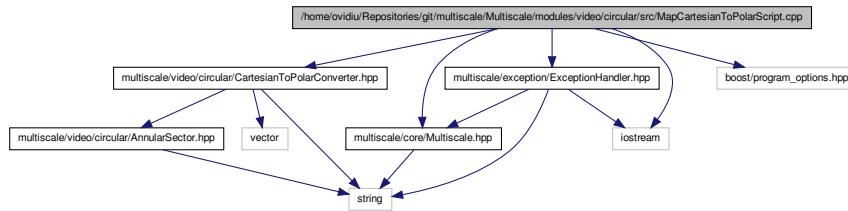
8.306 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/circular/src/CartesianToPolarConverter.cpp File Reference

```
#include "multiscale/exception/FileOpenException.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/video/circular/CartesianToPolar-
Converter.hpp" #include "multiscale/video/circular/Polar-
GnuplotScriptGenerator.hpp" #include "multiscale/util/-_
NumericRangeManipulator.hpp" #include <iostream> #include
<fstream> #include <exception> #include <cassert> #include
<vector> Include dependency graph for CartesianToPolarConverter.cpp:
```



8.307 /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 MapCartesian-
```



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 (string &inputFilepath, string &outputFilename, bool &isScript, int argc, char **argv)`
- `int main (int argc, char **argv)`

8.307.1 Function Documentation

8.307.1.1 `bool areValidParameters (string & inputFilepath, string & outputFilename, bool & isScript, int argc, char ** argv)`

Definition at line 100 of file `MapCartesianToPolarScript.cpp`.

References `initArgumentsConfig()`, `isValidOutputType()`, and `printHelpInformation()`.

8.307.1.2 `po::variables_map initArgumentsConfig (po::options_description & usageDescription, int argc, char ** argv)`

Definition at line 52 of file `MapCartesianToPolarScript.cpp`.

8.307.1.3 `bool isValidOutputType (const po::variables_map & vm, bool & isScript)`

Definition at line 79 of file `MapCartesianToPolarScript.cpp`.

References `multiscale::ERR_MSG`.

Referenced by `areValidParameters()`.

8.307.1.4 int main (int argc, char ** argv)

Definition at line 126 of file MapCartesianToPolarScript.cpp.

References `isValidParameters()`, `multiscale::video::CartesianToPolarConverter::convert()`, `multiscale::EXEC_ERR_CODE`, `multiscale::EXEC_SUCCESS_CODE`, and `printWrongParameters()`.

8.307.1.5 void printHelpInformation (const po::variables_map & vm, const po::options_description & usageDescription)

Definition at line 68 of file MapCartesianToPolarScript.cpp.

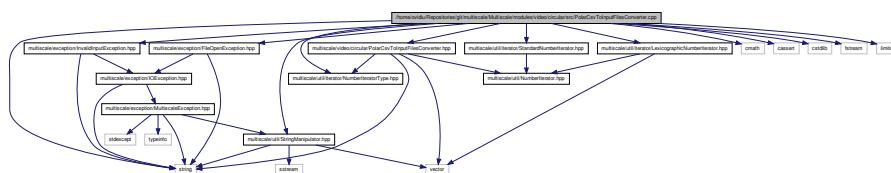
8.307.1.6 void printWrongParameters ()

Definition at line 73 of file MapCartesianToPolarScript.cpp.

References `multiscale::ERR_MSG`.

8.308 /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/PolarCsvToInput-
FilesConverter.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 PolarCsvToInputFiles-
Converter.cpp:
```



8.309 /home/ovidiu/Repositories/git/multiscale/-

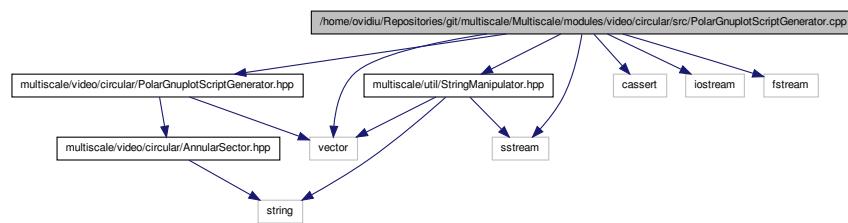
Multiscale/modules/video/circular/src/PolarGnuplotScriptGenerator.cpp File

Reference

1357

8.309 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/- PolarGnuplotScriptGenerator.cpp File Reference

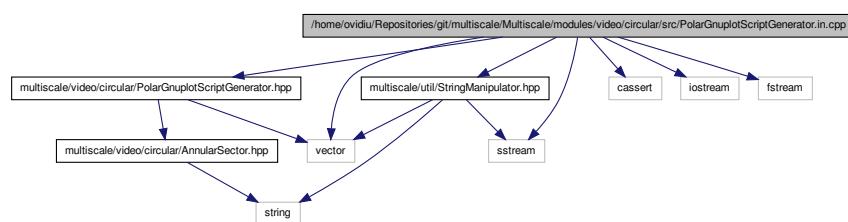
```
#include "multiscale/video/circular/PolarGnuplotScript-
Generator.hpp" #include "multiscale/util/StringManipulator.-
hpp" #include <cassert> #include <iostream> #include
<vector> #include <sstream> #include <fstream> Include de-
pendency graph for PolarGnuplotScriptGenerator.cpp:
```



8.310 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-

PolarGnuplotScriptGenerator.in.cpp File Reference

```
#include "multiscale/video/circular/PolarGnuplotScript-
Generator.hpp" #include "multiscale/util/StringManipulator.-
hpp" #include <cassert> #include <iostream> #include
<vector> #include <sstream> #include <fstream> Include de-
pendency graph for PolarGnuplotScriptGenerator.in.cpp:
```

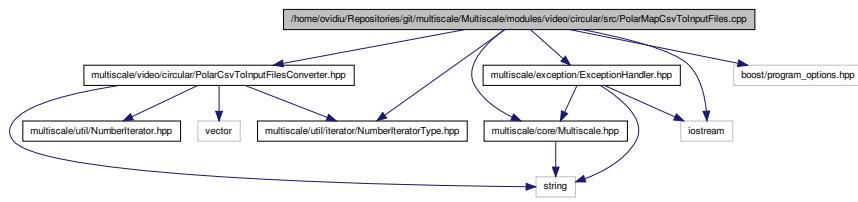


8.311 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/circular/src/-

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 PolarMapCsv-
ToInputFiles.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 (string &inputFilepath, 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)`

8.311.1 Function Documentation

8.311.1.1 `bool areValidParameters (string & inputFilepath, 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 91 of file `PolarMapCsvToInputFiles.cpp`.

8.311 /home/ovidiu/Repositories/git/multiscale-/Multiscale/modules/video/circular/src/PolarMapCsvToInputFiles.cpp File Reference 1359
References `initArgumentsConfig()`, `isValidNrOfConcentrationsForPosition()`, `printHelpInformation()`, `setLogScaling()`, `setNumberIteratorType()`, and `setSelectedConcentrationIndex()`.

8.311.1.2 po::variables_map initArgumentsConfig (po::options_description & *usageDescription*, int *argc*, char ** *argv*)

Definition at line 32 of file PolarMapCsvToInputFiles.cpp.

8.311.1.3 bool isValidNrOfConcentrationsForPosition (const po::variables_map & *vm*, unsigned int & *nrOfConcentrationsForPosition*)

Definition at line 76 of file PolarMapCsvToInputFiles.cpp.

References multiscale::ERR_MSG.

Referenced by `areValidParameters()`.

8.311.1.4 int main (int *argc*, char ** *argv*)

Definition at line 138 of file PolarMapCsvToInputFiles.cpp.

References `areValidParameters()`, `multiscale::video::PolarCsvToInputFilesConverter::convert()`, `multiscale::EXEC_ERR_CODE`, `multiscale::EXEC_SUCCESS_CODE`, `printWrongParameters()`, and `multiscale::STANDARD`.

8.311.1.5 void printHelpInformation (const po::variables_map & *vm*, const po::options_description & *usageDescription*)

Definition at line 50 of file PolarMapCsvToInputFiles.cpp.

8.311.1.6 void printWrongParameters ()

Definition at line 55 of file PolarMapCsvToInputFiles.cpp.

References multiscale::ERR_MSG.

8.311.1.7 void setLogScaling (const po::variables_map & *vm*, bool & *useLogScaling*)

Definition at line 71 of file PolarMapCsvToInputFiles.cpp.

Referenced by `areValidParameters()`.

8.311.1.8 void setNumberIteratorType (const po::variables_map & *vm*, NumberIteratorType & *numberIteratorType*)

Definition at line 61 of file PolarMapCsvToInputFiles.cpp.

References multiscale::LEXICOGRAPHIC.

Referenced by areValidParameters().

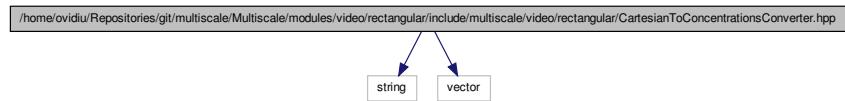
8.311.1.9 void setSelectedConcentrationIndex (const po::variables_map & vm, unsigned int & selectedConcentrationIndex)

Definition at line 66 of file PolarMapCsvToInputFiles.cpp.

Referenced by areValidParameters().

8.312 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/in 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

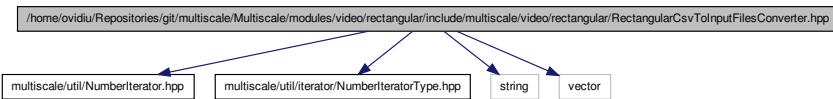
- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.313 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/in RectangularCsvToInputFilesConverter.hpp File Reference

```
#include "multiscale/util/NumberIterator.hpp"      #include
"multiscale/util/iterator/NumberIteratorType.hpp" #include
```

**8.314 /home/ovidiu/Repositories/git/multiscale-/
Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/-
RectangularEntityCsvToInputFilesConverter.hpp File**

Reference [#include <vector>](#) [Include dependency graph for RectangularEntityCsvToInputFilesConverter.hpp:](#) [1561](#)



Classes

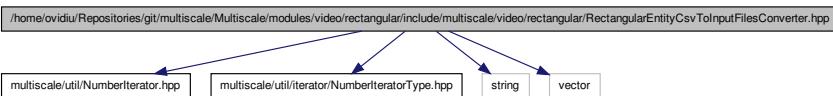
- class [multiscale::video::RectangularCsvToInputFilesConverter](#)
Csv file to input file converter considering cartesian coordinates.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.314 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/RectangularEntityCsvToInputFilesConverter.hpp File - Reference

```
#include "multiscale/util/NumberIterator.hpp"      #include
"multiscale/util/iterator/NumberIteratorType.hpp" #include
<string> #include <vector> Include dependency graph for Rectangular-
EntityCsvToInputFilesConverter.hpp:
```



Classes

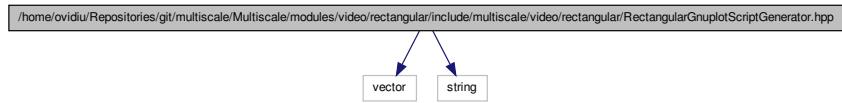
- class [multiscale::video::RectangularEntityCsvToInputFilesConverter](#)
Csv entity file to input file converter considering cartesian coordinates.

Namespaces

- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.315 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/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

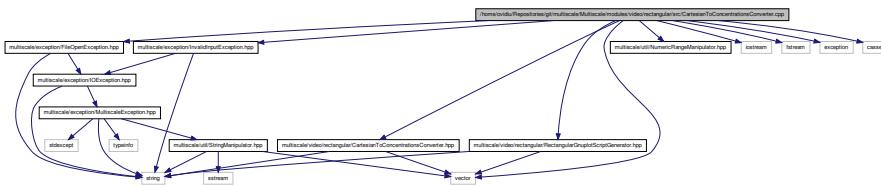
- namespace [multiscale](#)
- namespace [multiscale::video](#)

8.316 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/include/multiscale/video/rectangular/CartesianToConcentrationsConverter.hpp File Reference

```
#include "multiscale/exception/FileOpenException.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/video/rectangular/CartesianTo-
ConcentrationsConverter.hpp" #include "multiscale/video/rectangular/-
RectangularGnuplotScriptGenerator.hpp" #include "multiscale/util/-
NumericRangeManipulator.hpp" #include <iostream> #include
```

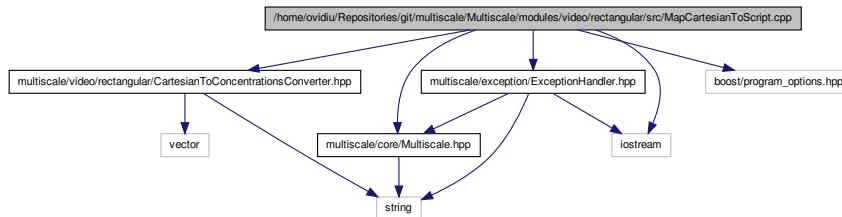
**8.317 /home/ovidiu/Repositories/git/multiscale/-
Multiscale/modules/video/rectangular/src/MapCartesianToScript.cpp File
Reference** 1363

```
<fstream> #include <exception> #include <cassert> #include
<vector> Include dependency graph for CartesianToConcentrationsConverter.cpp:
```



8.317 /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 MapCartesian-
ToScript.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 (string &inputFilepath, string &outputFilename, int argc, char **argv)`
- `int main (int argc, char **argv)`

8.317.1 Function Documentation

8.317.1.1 bool areValidParameters (string & *inputFilepath*, string & *outputFilename*, int *argc*, char ** *argv*)

Definition at line 60 of file MapCartesianToScript.cpp.

References initArgumentsConfig(), and printHelpInformation().

8.317.1.2 po::variables_map initArgumentsConfig (po::options_description & *usageDescription*, int *argc*, char ** *argv*)

Definition at line 37 of file MapCartesianToScript.cpp.

8.317.1.3 int main (int *argc*, char ** *argv*)

Definition at line 84 of file MapCartesianToScript.cpp.

References areValidParameters(), multiscale::video::CartesianToConcentrations-Converter::convert(), multiscale::EXEC_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, and printWrongParameters().

8.317.1.4 void printHelpInformation (const po::variables_map & *vm*, const po::options_description & *usageDescription*)

Definition at line 49 of file MapCartesianToScript.cpp.

8.317.1.5 void printWrongParameters ()

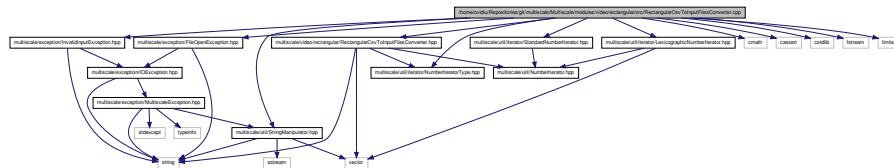
Definition at line 54 of file MapCartesianToScript.cpp.

References multiscale::ERR_MSG.

8.318 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/RectangularCsvToInputFilesConverter.cpp File Reference

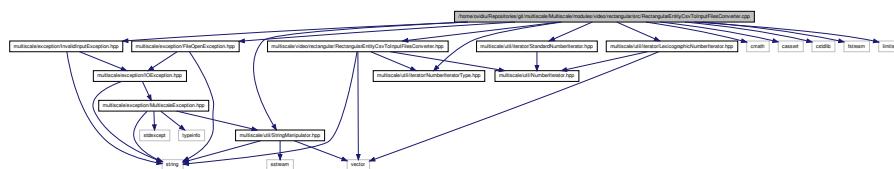
```
#include "multiscale/exception/FileOpenException.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp" #include "multiscale/video/rectangular/Rectangular-
CsvToInputFilesConverter.hpp" #include "multiscale/util/iterator/-.
NumberIteratorType.hpp" #include "multiscale/util/iterator/-.
LexicographicNumberIterator.hpp" #include "multiscale/util/iterator/-.
StandardNumberIterator.hpp" #include "multiscale/util/-.
StringManipulator.hpp" #include <cmath> #include <cassert> x
#include <cstdlib> #include <fstream> #include <limits>
```

8.319 /home/ovidiu/Repositories/git/multiscale-/
Multiscale/modules/video/rectangular/src/RectangularEntityCsvToInputFiles-
Converter.cpp File
Include dependency graph for RectangularEntityCsvToInputFilesConverter.cpp: 1365



8.319 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/src/-
RectangularEntityCsvToInputFilesConverter.cpp File -
Reference

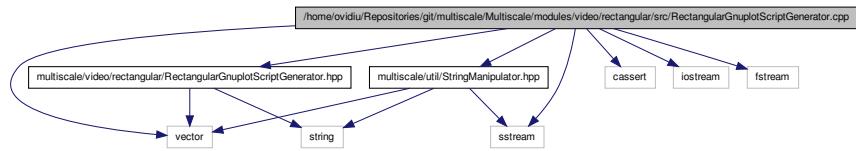
```
#include "multiscale/exception/FileOpenException.hpp" x
#include "multiscale/exception/InvalidInputException.-
hpp"    #include "multiscale/video/rectangular/Rectangular-
EntityCsvToInputFilesConverter.hpp" #include "multiscale/util/iterator/-_
NumberIteratorType.hpp" #include "multiscale/util/iterator/-_
LexicographicNumberIterator.hpp" #include "multiscale/util/iterator/-_
StandardNumberIterator.hpp"    #include "multiscale/util/-_
StringManipulator.hpp" #include <cmath> #include <cassert> x
#include <cstdlib> #include <fstream> #include <limits>
Include dependency graph for RectangularEntityCsvToInputFilesConverter.cpp:
```



8.320 /home/ovidiu.Repositories/git/multiscale/Multiscale/modules/video/rectangular/src-RectangularGnuplotScriptGenerator.cpp File Reference

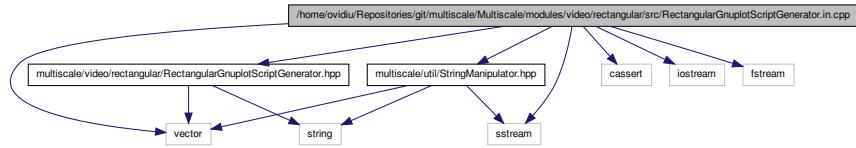
```
#include "multiscale/video/rectangular/RectangularGnuplot-  
ScriptGenerator.hpp"      #include "multiscale/util/String-  
Manipulator.hpp" #include <cassert> #include <iostream>  
#include <vector> #include <sstream> #include <fstream>
```

Include dependency graph for RectangularGnuplotScriptGenerator.cpp:



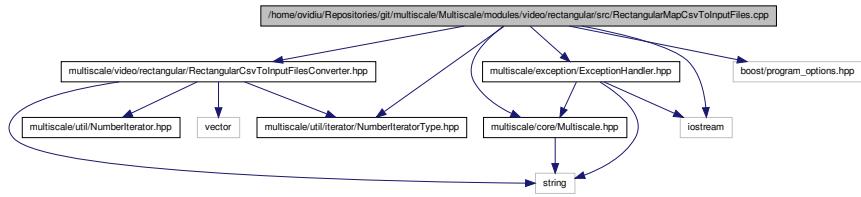
8.321 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/RectangularGnuplotScriptGenerator.in.cpp File Reference

```
#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.in.cpp:
```



8.322 /home/ovidiu/Repositories/git/multiscale/Multiscale/modules/video/rectangular/RectangularMapCsvToInputFiles.cpp File Reference

```
#include "multiscale/core/Multiscale.hpp" #include "multiscale/video/rectangular/CsvToInputFilesConverter.hpp" #include "multiscale/util/iterator/NumberIteratorType.hpp" #include "multiscale/exception/-ExceptionHandler.hpp" #include <boost/program_options.-hpp> #include <iostream>
Include dependency graph for RectangularMap-
```



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 (string &inputFilepath, 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)`

8.322.1 Function Documentation

8.322.1.1 `bool areValidParameters (string & inputFilepath, 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 91 of file RectangularMapCsvToInputFiles.cpp.

References `initArgumentsConfig()`, `isValidNrOfConcentrationsForPosition()`, `printHelpInformation()`, `setLogScaling()`, `setNumberIteratorType()`, and `setSelectedConcentrationIndex()`.

8.322.1.2 **po::variables_map initArgumentsConfig (po::options_description & usageDescription, int argc, char ** argv)**

Definition at line 32 of file RectangularMapCsvToInputFiles.cpp.

8.322.1.3 **bool isValidNrOfConcentrationsForPosition (const po::variables_map & vm, unsigned int & nrOfConcentrationsForPosition)**

Definition at line 76 of file RectangularMapCsvToInputFiles.cpp.

References multiscale::ERR_MSG.

8.322.1.4 **int main (int argc, char ** argv)**

Definition at line 138 of file RectangularMapCsvToInputFiles.cpp.

References areValidParameters(), multiscale::video::RectangularCsvToInputFiles-Converter::convert(), multiscale::EXEC_ERR_CODE, multiscale::EXEC_SUCCESS_CODE, printWrongParameters(), and multiscale::STANDARD.

8.322.1.5 **void printHelpInformation (const po::variables_map & vm, const po::options_description & usageDescription)**

Definition at line 50 of file RectangularMapCsvToInputFiles.cpp.

8.322.1.6 **void printWrongParameters ()**

Definition at line 55 of file RectangularMapCsvToInputFiles.cpp.

References multiscale::ERR_MSG.

8.322.1.7 **void setLogScaling (const po::variables_map & vm, bool & useLogScaling)**

Definition at line 71 of file RectangularMapCsvToInputFiles.cpp.

8.322.1.8 **void setNumberIteratorType (const po::variables_map & vm, NumberIteratorType & numberIteratorType)**

Definition at line 61 of file RectangularMapCsvToInputFiles.cpp.

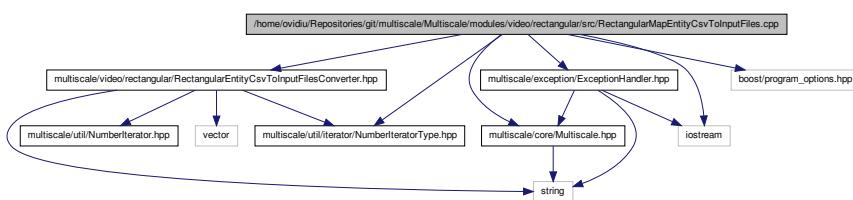
References multiscale::LEXICOGRAPHIC.

8.322.1.9 **void setSelectedConcentrationIndex (const po::variables_map & vm, unsigned int & selectedConcentrationIndex)**

Definition at line 66 of file RectangularMapCsvToInputFiles.cpp.

8.323 /home/ovidiu/Repositories/git/multiscale-/
Multiscale/modules/video/rectangular/src/RectangularMapEntityCsvToInput-
Files.cpp File
8.323 /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** (string &inputFilepath, 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)

8.323.1 Function Documentation

8.323.1.1 bool areValidParameters (string & *inputFilepath*, string & *outputFilename*, unsigned int & *height*, unsigned int & *width*, unsigned int & *nrOfEntities*, unsigned int & *maxPileup*, NumberIteratorType & *numberIteratorType*, int *argc*, char ** *argv*)

Definition at line 65 of file RectangularMapEntityCsvToInputFiles.cpp.

References initArgumentsConfig(), printHelpInformation(), and setNumberIteratorType().

```
8.323.1.2 po::variables_map initArgumentsConfig ( po::options_description &  
usageDescription, int argc, char ** argv )
```

Definition at line 32 of file RectangularMapEntityCsvToInputFiles.cpp.

```
8.323.1.3 int main ( int argc, char ** argv )
```

Definition at line 103 of file RectangularMapEntityCsvToInputFiles.cpp.

References `isValidParameters()`, `multiscale::video::RectangularEntityCsvToInputFilesConverter::convert()`, `multiscale::EXEC_ERR_CODE`, `multiscale::EXEC_SUCCESS_CODE`, `printWrongParameters()`, and `multiscale::STANDARD`.

```
8.323.1.4 void printHelpInformation ( const po::variables_map & vm, const  
po::options_description & usageDescription )
```

Definition at line 49 of file RectangularMapEntityCsvToInputFiles.cpp.

```
8.323.1.5 void printWrongParameters ( )
```

Definition at line 54 of file RectangularMapEntityCsvToInputFiles.cpp.

References `multiscale::ERR_MSG`.

```
8.323.1.6 void setNumberIteratorType ( const po::variables_map & vm,  
NumberIteratorType & numberIteratorType )
```

Definition at line 60 of file RectangularMapEntityCsvToInputFiles.cpp.

References `multiscale::LEXICOGRAPHIC`.