

Intention Recognition

Generated by Doxygen 1.7.4

Sun Sep 15 2013 21:04:58

Contents

1	Intention Recognition	1
2	Module Index	3
2.1	Modules	3
3	Namespace Index	5
3.1	Packages	5
4	Class Index	7
4.1	Class List	7
5	File Index	9
5.1	File List	9
6	Module Documentation	11
6.1	Graphical User Interface	11
6.2	Intention Structure	11
7	Namespace Documentation	13
7.1	Package component	13
7.2	Package gui	13
7.3	Package intention	13
7.4	Package main	14
7.5	Package ontology	14
7.6	Package textfiles	14
7.7	Package tools	14
7.8	Package treecheckbox	15

8	Class Documentation	17
8.1	ontology.AnyOrder Class Reference	17
8.1.1	Constructor & Destructor Documentation	18
8.1.1.1	AnyOrder	18
8.1.2	Member Function Documentation	18
8.1.2.1	addCountToList	18
8.1.2.2	getCountList	18
8.1.2.3	getIndividual	18
8.1.2.4	getPosition	18
8.1.2.5	getTotalNumber	18
8.1.2.6	setIndividual	18
8.1.2.7	setPosition	18
8.1.2.8	setTotalNumber	18
8.1.3	Member Data Documentation	18
8.1.3.1	m_count_list	18
8.1.3.2	m_individual	18
8.1.3.3	m_position	18
8.1.3.4	m_total_number	18
8.2	gui.Chart Class Reference	18
8.2.1	Detailed Description	21
8.2.2	Constructor & Destructor Documentation	21
8.2.2.1	Chart	21
8.2.3	Member Function Documentation	21
8.2.3.1	createChart_likelihood	21
8.2.3.2	createChart_metrics	21
8.2.3.3	createLikelihoodDataset	21
8.2.3.4	createMetricsDataset	21
8.2.3.5	getChartTitle	21
8.2.3.6	getLegendItems	22
8.2.3.7	getMetricsChartTitle	22
8.2.3.8	setChartTitle	22
8.2.3.9	setMetricsChartTitle	22
8.2.3.10	updateChart_likelihood	22
8.2.3.11	updateChart_metrics	22

8.2.4	Member Data Documentation	22
8.2.4.1	m_colors	22
8.2.4.2	m_LH_Chart_Title	23
8.2.4.3	m_LH_chartPanel	23
8.2.4.4	m_LH_Dataset	23
8.2.4.5	m_Metric_Chart_Title	23
8.2.4.6	m_Metric_chartPanel	23
8.2.4.7	m_Metric_Dataset	23
8.2.4.8	m_tmp_LH_ChartPanel	23
8.2.4.9	m_tmp_Metric_ChartPanel	23
8.3	treecheckbox.CheckTreeCellRenderer Class Reference	24
8.3.1	Constructor & Destructor Documentation	25
8.3.1.1	CheckTreeCellRenderer	25
8.3.2	Member Function Documentation	25
8.3.2.1	getTreeCellRendererComponent	25
8.3.3	Member Data Documentation	25
8.3.3.1	checkBox	25
8.3.3.2	delegate	25
8.3.3.3	selectionModel	25
8.4	treecheckbox.CheckTreeManager Class Reference	25
8.4.1	Constructor & Destructor Documentation	26
8.4.1.1	CheckTreeManager	26
8.4.2	Member Function Documentation	26
8.4.2.1	addChildPaths	27
8.4.2.2	getAllCheckedPaths	27
8.4.2.3	getDescendants	27
8.4.2.4	getSelectionModel	27
8.4.2.5	mouseClicked	27
8.4.2.6	valueChanged	27
8.4.3	Member Data Documentation	27
8.4.3.1	hotspot	27
8.4.3.2	selectionModel	27
8.4.3.3	tree	27
8.5	treecheckbox.CheckTreeSelectionModel Class Reference	27

8.5.1	Constructor & Destructor Documentation	28
8.5.1.1	CheckTreeSelectionModel	28
8.5.2	Member Function Documentation	28
8.5.2.1	addSelectionPaths	28
8.5.2.2	areSiblingsSelected	29
8.5.2.3	isDescendant	29
8.5.2.4	isPartiallySelected	29
8.5.2.5	isPathSelected	29
8.5.2.6	removeSelectionPaths	29
8.5.2.7	setSelectionPaths	29
8.5.2.8	toggleRemoveSelection	29
8.5.3	Member Data Documentation	29
8.5.3.1	model	29
8.6	gui.CommonGUIComponents Class Reference	29
8.6.1	Detailed Description	31
8.6.2	Member Data Documentation	31
8.6.2.1	chartLeftPaneColor	31
8.6.2.2	chartPanelColor	31
8.6.2.3	intentionColor1	31
8.6.2.4	intentionColor10	31
8.6.2.5	intentionColor2	31
8.6.2.6	intentionColor3	31
8.6.2.7	intentionColor4	31
8.6.2.8	intentionColor5	32
8.6.2.9	intentionColor6	32
8.6.2.10	intentionColor7	32
8.6.2.11	intentionColor8	32
8.6.2.12	intentionColor9	32
8.6.2.13	JFreeChartBackground	32
8.6.2.14	JTabbedPaneColor	32
8.6.2.15	m_legendFont	32
8.6.2.16	m_part_green	32
8.6.2.17	m_part_orange	32
8.6.2.18	m_part_yellow	32

8.6.2.19	m_state	32
8.6.2.20	m_stateRelationFont	32
8.6.2.21	m_titleFont	32
8.6.2.22	menuBarColor	32
8.6.2.23	state_color	32
8.6.2.24	tabColor	33
8.7	tools.Configuration Class Reference	33
8.7.1	Member Function Documentation	33
8.7.1.1	isMac	33
8.7.1.2	isUnix	33
8.7.1.3	isWindows	33
8.7.2	Member Data Documentation	33
8.7.2.1	m_OS	33
8.8	ontology.Count Class Reference	34
8.8.1	Constructor & Destructor Documentation	35
8.8.1.1	Count	35
8.8.2	Member Function Documentation	35
8.8.2.1	getIndividual	35
8.8.2.2	getOccurence	35
8.8.2.3	getPredicate	35
8.8.2.4	setIndividual	35
8.8.2.5	setOccurrence	35
8.8.2.6	setPredicate	35
8.8.3	Member Data Documentation	35
8.8.3.1	m_individual	35
8.8.3.2	m_occurrence	35
8.8.3.3	m_predicate	35
8.9	gui.Chart.CustomRenderer Class Reference	36
8.9.1	Constructor & Destructor Documentation	37
8.9.1.1	CustomRenderer	37
8.9.2	Member Function Documentation	37
8.9.2.1	getItemPaint	37
8.9.3	Member Data Documentation	37
8.9.3.1	colors	37

8.9.3.2	serialVersionUID	37
8.10	gui.Chart.CustomRendererLine Class Reference	37
8.10.1	Constructor & Destructor Documentation	38
8.10.1.1	CustomRendererLine	38
8.10.2	Member Function Documentation	38
8.10.2.1	getItemPaint	38
8.10.3	Member Data Documentation	38
8.10.3.1	colors	38
8.10.3.2	serialVersionUID	39
8.11	gui.DemoPanel Class Reference	39
8.11.1	Constructor & Destructor Documentation	40
8.11.1.1	DemoPanel	40
8.11.2	Member Function Documentation	40
8.11.2.1	addChart	40
8.11.2.2	getCharts	40
8.11.3	Member Data Documentation	40
8.11.3.1	charts	40
8.12	gui.MainFrame.DisplayMetrics Class Reference	40
8.12.1	Detailed Description	41
8.12.2	Member Function Documentation	41
8.12.2.1	actionPerformed	41
8.13	DocumentFilter Class Reference	41
8.13.1	Detailed Description	41
8.14	gui.DrawStringPanel Class Reference	42
8.14.1	Constructor & Destructor Documentation	43
8.14.1.1	DrawStringPanel	43
8.14.2	Member Function Documentation	43
8.14.2.1	getFont	43
8.14.2.2	getPreferredSize	43
8.14.2.3	paintComponent	43
8.14.2.4	setAnchor	43
8.14.2.5	setAngle	43
8.14.2.6	setFont	43
8.14.2.7	setRotationAnchor	43

8.14.3	Member Data Documentation	43
8.14.3.1	anchor	44
8.14.3.2	angle	44
8.14.3.3	font	44
8.14.3.4	PREFERRED_SIZE	44
8.14.3.5	rotate	44
8.14.3.6	rotationAnchor	44
8.14.3.7	text	44
8.15	ontology.Exist Class Reference	44
8.15.1	Detailed Description	45
8.15.2	Constructor & Destructor Documentation	45
8.15.2.1	Exist	45
8.15.3	Member Function Documentation	45
8.15.3.1	getIndividual	45
8.15.3.2	getOccurrence	45
8.15.3.3	getPosition	45
8.15.3.4	getPredicate	45
8.15.3.5	getTotalNumber	46
8.15.3.6	setIndividual	46
8.15.3.7	setOccurrence	46
8.15.3.8	setPosition	46
8.15.3.9	setPredicate	46
8.15.3.10	setTotalNumber	46
8.15.4	Member Data Documentation	46
8.15.4.1	m_individual	46
8.15.4.2	m_position	46
8.15.4.3	m_predicate	46
8.15.4.4	m_total_number	46
8.15.4.5	occurrence	46
8.16	textfiles.FileOperator Class Reference	46
8.16.1	Detailed Description	47
8.16.2	Constructor & Destructor Documentation	47
8.16.2.1	FileOperator	47
8.16.3	Member Function Documentation	47

8.16.3.1	openFile	47
8.16.3.2	readLines	47
8.16.3.3	saveAllKitsData	47
8.16.3.4	translatePlanToStateRelation	47
8.17	intention.Intention Class Reference	47
8.17.1	Detailed Description	52
8.17.2	Constructor & Destructor Documentation	52
8.17.2.1	Intention	52
8.17.3	Member Function Documentation	52
8.17.3.1	builDetrimentalList	52
8.17.3.2	getDetrimentalList	52
8.17.3.3	getIndividual	52
8.17.3.4	getIntentionName	53
8.17.3.5	getM_am1	53
8.17.3.6	getM_am2	53
8.17.3.7	getM_am3	53
8.17.3.8	getM_am4	53
8.17.3.9	getM_am5	54
8.17.3.10	getM_built_kit	54
8.17.3.11	getM_found_detrimental_SR	54
8.17.3.12	getM_intention_number	54
8.17.3.13	getM_likelihood_observation	54
8.17.3.14	getM_map_SRirs	54
8.17.3.15	getM_mm1	55
8.17.3.16	getM_percentComplete_i_s	55
8.17.3.17	getM_percentProductive_i_s	55
8.17.3.18	getM_probability_kit_observation	55
8.17.3.19	getM_selected_plan	55
8.17.3.20	getM_Si	55
8.17.3.21	getM_SR_i_r_s	56
8.17.3.22	getM_SR_i_s	56
8.17.3.23	getM_SR_Total	56
8.17.3.24	getNumberStateRelation	56
8.17.3.25	getOrderedList	56

8.17.3.26 setDetrimentalList	56
8.17.3.27 setExist	57
8.17.3.28 setIndividual	57
8.17.3.29 setIntentionName	57
8.17.3.30 setM_am1	57
8.17.3.31 setM_am2	57
8.17.3.32 setM_am3	58
8.17.3.33 setM_am4	58
8.17.3.34 setM_am5	58
8.17.3.35 setM_built_kit	58
8.17.3.36 setM_found_detrimental_SR	58
8.17.3.37 setM_intention_number	59
8.17.3.38 setM_likelihood_observation	59
8.17.3.39 setM_map_SRirs	59
8.17.3.40 setM_mm1	59
8.17.3.41 setM_percentComplete_i_s	59
8.17.3.42 setM_percentProductive_i_s	60
8.17.3.43 setM_probability_kit_observation	60
8.17.3.44 setM_selected_plan	60
8.17.3.45 setM_Si	60
8.17.3.46 setM_SR_i_r_s	60
8.17.3.47 setM_SR_i_s	61
8.17.3.48 setM_SR_Total	61
8.17.3.49 setNumberStateRelation	61
8.17.3.50 setOrderedList	61
8.17.4 Member Data Documentation	61
8.17.4.1 m_am1	61
8.17.4.2 m_am2	62
8.17.4.3 m_am3	62
8.17.4.4 m_am4	63
8.17.4.5 m_am5	63
8.17.4.6 m_anyorder	64
8.17.4.7 m_built_kit	64
8.17.4.8 m_detrimental_list	64

8.17.4.9	m_exist	64
8.17.4.10	m_found_detrimental_SR	64
8.17.4.11	m_individual	64
8.17.4.12	m_intention_name	64
8.17.4.13	m_intention_number	64
8.17.4.14	m_intention_orderingConstruct_list	64
8.17.4.15	m_likelihood_observation	64
8.17.4.16	m_mm1	65
8.17.4.17	m_number_state_relation	65
8.17.4.18	m_orderedlist	65
8.17.4.19	m_percentComplete_i_s	65
8.17.4.20	m_percentProductive_i_s	65
8.17.4.21	m_probability_kit_observation	65
8.17.4.22	m_selected_plan	65
8.17.4.23	m_Si	66
8.17.4.24	m_SR_i_r_s	66
8.17.4.25	m_SR_i_s	66
8.17.4.26	m_SR_Total	66
8.17.4.27	m_SRirs	66
8.18	component.IntFilter Class Reference	66
8.18.1	Member Function Documentation	67
8.18.1.1	insertString	67
8.18.1.2	remove	67
8.18.1.3	replace	67
8.18.1.4	test	67
8.19	main.Launcher Class Reference	68
8.19.1	Detailed Description	68
8.19.2	Constructor & Destructor Documentation	68
8.19.2.1	Launcher	68
8.19.3	Member Function Documentation	69
8.19.3.1	enabler	69
8.19.3.2	main	69
8.20	intention.Likelihood Class Reference	70
8.21	gui.MainFrame Class Reference	70

8.21.1	Constructor & Destructor Documentation	73
8.21.1.1	MainFrame	73
8.21.2	Member Function Documentation	73
8.21.2.1	actionPerformed	73
8.21.2.2	applyThemeToChart	73
8.21.2.3	attemptExit	73
8.21.2.4	buildLikelihoodPanel	73
8.21.2.5	buildMetricPanel	73
8.21.2.6	copyToClipboard	73
8.21.2.7	createContent	74
8.21.2.8	createLikelihoodChartPanel	74
8.21.2.9	createMenuBar	74
8.21.2.10	createMetricsChartPanel	74
8.21.2.11	exportToCSV	74
8.21.2.12	exportToPDF	74
8.21.2.13	lh_Button_ActionPerformed	74
8.21.2.14	saveToCSV	74
8.21.2.15	showErrorMessage	74
8.21.2.16	updateStateRelationPanel	74
8.21.3	Member Data Documentation	74
8.21.3.1	_sd	75
8.21.3.2	editorPane	75
8.21.3.3	EXIT_COMMAND	75
8.21.3.4	likelihood_bottomRightPane	75
8.21.3.5	likelihood_chartContainer	75
8.21.3.6	likelihood_displayPanel	75
8.21.3.7	likelihood_stateContainer	75
8.21.3.8	likelihoodLeft_JTextPane	75
8.21.3.9	m_csvitem	75
8.21.3.10	m_LH_Left_Panel	75
8.21.3.11	m_LH_ScrollPane	75
8.21.3.12	m_LH_Selection_Button	75
8.21.3.13	m_LH_SplitPane	75
8.21.3.14	m_likelihood_barchart	75

8.21.3.15	m_likelihoodCheckTreeManager	75
8.21.3.16	m_likelihoodTree	75
8.21.3.17	m_metricCheckTreeManager	75
8.21.3.18	m_metrics_barchart	75
8.21.3.19	m_metrics_jscrollpane	75
8.21.3.20	m_metricsSelectionButton	75
8.21.3.21	m_metricTree	76
8.21.3.22	metrics_chartContainer	76
8.21.3.23	metrics_jsplitpane	76
8.21.3.24	metrics_statePane	76
8.21.3.25	metricsLeft_JPanel	76
8.21.3.26	metricsLeft_JScrollPane	76
8.21.3.27	metricsLeft_JTextPane	76
8.21.3.28	serialVersionUID	76
8.22	intention.Metric Class Reference	76
8.22.1	Detailed Description	77
8.22.2	Member Function Documentation	77
8.22.2.1	get_AM1_Weight	77
8.22.2.2	get_AM2_Weight	77
8.22.2.3	get_AM3_Weight	77
8.22.2.4	get_AM4_Weight	78
8.22.2.5	get_AM5_Weight	78
8.22.2.6	get_MM1_Weight	78
8.22.2.7	set_AM1_Weight	78
8.22.2.8	set_AM2_Weight	78
8.22.2.9	set_AM3_Weight	78
8.22.2.10	set_AM4_Weight	79
8.22.2.11	set_AM5_Weight	79
8.22.2.12	set_MM1_Weight	79
8.22.3	Member Data Documentation	79
8.22.3.1	m_AM1_weight	79
8.22.3.2	m_AM2_weight	79
8.22.3.3	m_AM3_weight	79
8.22.3.4	m_AM4_weight	79

8.22.3.5	m_AM5_weight	79
8.22.3.6	m_MM1_weight	79
8.23	gui.OptionFrame.MouseHandler Class Reference	80
8.23.1	Constructor & Destructor Documentation	80
8.23.1.1	MouseHandler	80
8.23.2	Member Function Documentation	80
8.23.2.1	mousePressed	80
8.23.2.2	mouseReleased	80
8.23.3	Member Data Documentation	80
8.23.3.1	popupMenu	80
8.24	ontology.Ontology Class Reference	81
8.24.1	Constructor & Destructor Documentation	86
8.24.1.1	Ontology	86
8.24.2	Member Function Documentation	86
8.24.2.1	buildIntentionList	86
8.24.2.2	buildIntentionMetricsTree	87
8.24.2.3	buildIntentionTree	87
8.24.2.4	buildObservationList	87
8.24.2.5	buildStates	87
8.24.2.6	choose	87
8.24.2.7	chooseTest	88
8.24.2.8	cleanDataPropertyInteger	88
8.24.2.9	cleanIRI	88
8.24.2.10	compute_SR_all_s	88
8.24.2.11	compute_SR_i_r_s	88
8.24.2.12	compute_sum_observation	89
8.24.2.13	compute_sum_part_type	89
8.24.2.14	computeLikelihood	89
8.24.2.15	computeMetricAM1	89
8.24.2.16	computeMetricAM2	90
8.24.2.17	computeMetricAM3	91
8.24.2.18	computeMetricAM4	91
8.24.2.19	computeMetricAM5	91
8.24.2.20	computeMetricMM1	92

8.24.2.21 computeMetricsInformation	92
8.24.2.22 computeObservationLikelihood	93
8.24.2.23 computePercentComplete	93
8.24.2.24 computePercentProductive	94
8.24.2.25 computeProbabilityFromObservation	94
8.24.2.26 computeProceduresForAM5	94
8.24.2.27 getForEachIntentionTheNumberOfPartsForEachType	94
8.24.2.28 getIndividualClass	94
8.24.2.29 getIndividualClassString	94
8.24.2.30 getIndividualClassString	94
8.24.2.31 getManager	94
8.24.2.32 getOntology	95
8.24.2.33 getOntologyPath	95
8.24.2.34 getPartColor	95
8.24.2.35 getPartType	95
8.24.2.36 getPath	95
8.24.2.37 getReferenceObjectClass	95
8.24.2.38 getRootClass	96
8.24.2.39 getSeparator	96
8.24.2.40 getStateRelation	96
8.24.2.41 getStringHead	96
8.24.2.42 getStringTail	96
8.24.2.43 getSubclasses	96
8.24.2.44 getTargetObjectClass	96
8.24.2.45 hasProperty	97
8.24.2.46 initializeList	97
8.24.2.47 input	97
8.24.2.48 loadFromFile	97
8.24.2.49 loadOntologyFromPath	97
8.24.2.50 matchDetrimentalStateRelationToIntention	97
8.24.2.51 matchStateRelationToIntention	97
8.24.2.52 parseIntention	98
8.24.2.53 printProperties	98
8.24.2.54 readForEachIntentionTheNumberOfPartsForEachType	98

8.24.2.55	readIntentionList	98
8.24.2.56	readObservationList	98
8.24.2.57	removeDuplicates	99
8.24.2.58	roundTwoDecimals	99
8.24.2.59	searchList	99
8.24.2.60	setDataFactory	99
8.24.2.61	setInstanceFilePath	99
8.24.2.62	setManager	99
8.24.2.63	setManager	99
8.24.2.64	setOntology	99
8.24.2.65	setPath	100
8.24.2.66	setReasoner	100
8.24.2.67	setRootClass	100
8.24.2.68	showDialogBox	100
8.24.2.69	sortIntentionList	100
8.24.2.70	updateForEachIntentionTheNumberOfPartsForEachType	100
8.24.2.71	updateMainFrame	101
8.24.2.72	updateObservationList	101
8.24.3	Member Data Documentation	101
8.24.3.1	m_datafactory	101
8.24.3.2	m_hasCount_Occurrence	101
8.24.3.3	m_hasIntention_Name	101
8.24.3.4	m_hasIntention_OrderingConstruct	101
8.24.3.5	m_hasOrderingConstruct_OrderingConstruct	101
8.24.3.6	m_hasOrderingConstruct_Position	101
8.24.3.7	m_hasOrderingConstruct_Predicate	101
8.24.3.8	m_hasPredicate_ReferenceObject	101
8.24.3.9	m_hasPredicate_TargetObject	102
8.24.3.10	m_intention_list	102
8.24.3.11	m_kitToBuild	102
8.24.3.12	m_manager	102
8.24.3.13	m_observation_list	102
8.24.3.14	m_ontology	102
8.24.3.15	m_ontology_IRI	102

8.24.3.16 m_part_type_number_for_each_intention_list	102
8.24.3.17 m_path	102
8.24.3.18 m_planToBuild	102
8.24.3.19 m_progress_bar	102
8.24.3.20 m_progress_frame	102
8.24.3.21 m_reasoner	102
8.24.3.22 m_s_ontopath	102
8.24.3.23 m_s_rootClass	102
8.24.3.24 m_s_subClass	102
8.24.3.25 m_SEPARATOR	102
8.25 Ontology Class Reference	103
8.25.1 Detailed Description	103
8.26 gui.OntologyChooser Class Reference	103
8.26.1 Detailed Description	105
8.26.2 Constructor & Destructor Documentation	105
8.26.2.1 OntologyChooser	105
8.26.3 Member Function Documentation	105
8.26.3.1 actionPerformed	105
8.26.3.2 createAndShowGUI	105
8.26.3.3 createImageIcon	105
8.26.4 Member Data Documentation	105
8.26.4.1 fc	105
8.26.4.2 log	105
8.26.4.3 newline	105
8.26.4.4 openButton	105
8.26.4.5 saveButton	105
8.27 ontology.OnTopWithContactStateRelation Class Reference	105
8.27.1 Constructor & Destructor Documentation	106
8.27.1.1 OnTopWithContactStateRelation	106
8.27.2 Member Function Documentation	106
8.27.2.1 setName	106
8.27.3 Member Data Documentation	106
8.27.3.1 sr_name	106
8.28 gui.OptionFrame Class Reference	106

8.28.1	Constructor & Destructor Documentation	110
8.28.1.1	OptionFrame	110
8.28.2	Member Function Documentation	110
8.28.2.1	add	110
8.28.2.2	addComponentsToPane	110
8.28.2.3	createAndShowGUI	110
8.28.2.4	createAndUpdateConfigFile	110
8.28.2.5	createButtonFromTemplate	110
8.28.2.6	createPopupMenu	110
8.28.2.7	findFilesinDirectory	110
8.28.2.8	getKitList	110
8.28.2.9	getMetric	110
8.28.2.10	readConfigFile	110
8.28.2.11	setMetric	110
8.28.2.12	updateConfigFile	110
8.28.2.13	updateLabel	110
8.28.3	Member Data Documentation	110
8.28.3.1	allKitsButton	110
8.28.3.2	contentPane	111
8.28.3.3	CREATE_WINDOW	111
8.28.3.4	DEFAULT_ICON	111
8.28.3.5	FILE_ICON	111
8.28.3.6	LF_DECORATIONS	111
8.28.3.7	m_bool_allKits	111
8.28.3.8	m_browse_instance	111
8.28.3.9	m_browse_path	111
8.28.3.10	m_field_AM1	111
8.28.3.11	m_field_AM2	111
8.28.3.12	m_field_AM3	111
8.28.3.13	m_field_AM4	111
8.28.3.14	m_field_AM5	111
8.28.3.15	m_frame	111
8.28.3.16	m_instance_txt_field	111
8.28.3.17	m_kitComboBox	111

8.28.3.18	m_metric	111
8.28.3.19	m_planComboBox	111
8.28.3.20	m_planStrings	111
8.28.3.21	m_popupMenu	111
8.28.3.22	m_save_file_txt_field	111
8.28.3.23	m_subkitlist	112
8.28.3.24	m_textArea	112
8.28.3.25	m_validate	112
8.28.3.26	m_validate_button	112
8.28.3.27	NO_DECORATIONS	112
8.28.3.28	PAINT_ICON	112
8.28.3.29	picture	112
8.28.3.30	RIGHT_TO_LEFT	112
8.28.3.31	shouldFill	112
8.28.3.32	shouldWeightX	112
8.28.3.33	singleKitButton	112
8.28.3.34	WS_DECORATIONS	112
8.29	ontology.OrderedList Class Reference	112
8.29.1	Constructor & Destructor Documentation	113
8.29.1.1	OrderedList	113
8.29.2	Member Function Documentation	113
8.29.2.1	getAnyOrder	113
8.29.2.2	getExist	113
8.29.2.3	getIndividual	113
8.29.2.4	setAnyOrder	113
8.29.2.5	setExist	113
8.29.2.6	setIndividual	113
8.29.3	Member Data Documentation	113
8.29.3.1	m_anyorder	113
8.29.3.2	m_exist	113
8.29.3.3	m_individual	114
8.29.3.4	orderedlist_l	114
8.29.3.5	orderedlist_name	114
8.30	ontology.OrderingConstruct Class Reference	114

8.30.1	Constructor & Destructor Documentation	114
8.30.1.1	OrderingConstruct	114
8.31	ontology.PartiallyInStateRelation Class Reference	114
8.31.1	Constructor & Destructor Documentation	114
8.31.1.1	PartiallyInStateRelation	114
8.31.2	Member Function Documentation	114
8.31.2.1	setName	114
8.31.3	Member Data Documentation	115
8.31.3.1	sr_name	115
8.32	gui.PDFChartTransferable Class Reference	115
8.32.1	Constructor & Destructor Documentation	116
8.32.1.1	PDFChartTransferable	116
8.32.1.2	PDFChartTransferable	116
8.32.2	Member Function Documentation	116
8.32.2.1	getTransferData	116
8.32.2.2	getTransferDataFlavors	116
8.32.2.3	isDataFlavorSupported	116
8.32.2.4	writeChartAsPDF	116
8.32.3	Member Data Documentation	116
8.32.3.1	chart	117
8.32.3.2	height	117
8.32.3.3	pdfFlavor	117
8.32.3.4	width	117
8.33	gui.OptionFrame.PlanComboBoxListener Class Reference	117
8.33.1	Member Function Documentation	118
8.33.1.1	actionPerformed	118
8.34	ontology.Predicate Class Reference	118
8.34.1	Constructor & Destructor Documentation	119
8.34.1.1	Predicate	119
8.34.2	Member Function Documentation	119
8.34.2.1	getIndividual	119
8.34.2.2	getReference	119
8.34.2.3	getReferenceObjectClass	119
8.34.2.4	getStateRelation	119

8.34.2.5	getTarget	119
8.34.2.6	getTargetObjectClass	119
8.34.2.7	setIndividual	119
8.34.2.8	setReference	119
8.34.2.9	setReferenceObjectClass	119
8.34.2.10	setStateRelation	119
8.34.2.11	setTarget	119
8.34.2.12	setTargetObjectClass	119
8.34.3	Member Data Documentation	119
8.34.3.1	m_individual	120
8.34.3.2	m_reference	120
8.34.3.3	m_reference_class	120
8.34.3.4	m_staterelation	120
8.34.3.5	m_target	120
8.34.3.6	m_target_class	120
8.35	gui.ProgressBar Class Reference	120
8.35.1	Constructor & Destructor Documentation	121
8.35.1.1	ProgressBar	121
8.35.2	Member Function Documentation	121
8.35.2.1	updateBar	121
8.35.3	Member Data Documentation	121
8.35.3.1	MY_MAXIMUM	121
8.35.3.2	MY_MINIMUM	122
8.35.3.3	pbar	122
8.36	treecheckbox.TreeExample Class Reference	122
8.36.1	Constructor & Destructor Documentation	123
8.36.1.1	TreeExample	123
8.36.2	Member Data Documentation	123
8.36.2.1	scrollPane	123
8.36.2.2	serialVersionUID	123
8.36.2.3	topPanel	123
8.36.2.4	tree	123
8.37	treecheckbox.TristateCheckBox Class Reference	124
8.37.1	Detailed Description	125

8.37.2	Constructor & Destructor Documentation	125
8.37.2.1	TristateCheckBox	125
8.37.2.2	TristateCheckBox	125
8.37.2.3	TristateCheckBox	125
8.37.2.4	TristateCheckBox	125
8.37.3	Member Function Documentation	125
8.37.3.1	addMouseListener	125
8.37.3.2	getState	125
8.37.3.3	setState	126
8.37.4	Member Data Documentation	126
8.37.4.1	model	126
8.38	treecheckbox.TristateCheckBox.TristateDecorator Class Reference . . .	126
8.38.1	Detailed Description	128
8.38.2	Constructor & Destructor Documentation	128
8.38.2.1	TristateDecorator	128
8.38.3	Member Function Documentation	128
8.38.3.1	addActionListener	128
8.38.3.2	addChangeListener	128
8.38.3.3	addItemListener	128
8.38.3.4	getActionCommand	128
8.38.3.5	getMnemonic	128
8.38.3.6	getSelectedObjects	128
8.38.3.7	getState	128
8.38.3.8	isArmed	129
8.38.3.9	isEnabled	129
8.38.3.10	isFocusTraversable	129
8.38.3.11	isPressed	129
8.38.3.12	isRollover	129
8.38.3.13	isSelected	129
8.38.3.14	nextState	129
8.38.3.15	removeActionListener	129
8.38.3.16	removeChangeListener	129
8.38.3.17	removeItemListener	129
8.38.3.18	setActionCommand	129

8.38.3.19	setArmed	129
8.38.3.20	setEnabled	129
8.38.3.21	setGroup	129
8.38.3.22	setMnemonic	129
8.38.3.23	setPressed	129
8.38.3.24	setRollover	129
8.38.3.25	setSelected	130
8.38.3.26	setState	130
8.38.4	Member Data Documentation	130
8.38.4.1	other	130
8.39	ontology.UnderWithContactStateRelation Class Reference	130
8.39.1	Constructor & Destructor Documentation	130
8.39.1.1	UnderWithContactStateRelation	130
8.39.2	Member Function Documentation	130
8.39.2.1	setName	130
8.39.3	Member Data Documentation	130
8.39.3.1	sr_name	130
9	File Documentation	131
9.1	src/component/IntFilter.java File Reference	131
9.2	src/gui/Chart.java File Reference	131
9.3	src/gui/CommonGUIComponents.java File Reference	131
9.4	src/gui/DemoPanel.java File Reference	132
9.5	src/gui/DrawStringPanel.java File Reference	132
9.6	src/gui/MainFrame.java File Reference	132
9.7	src/gui/OntologyChooser.java File Reference	133
9.8	src/gui/OptionFrame.java File Reference	133
9.9	src/gui/PDFChartTransferable.java File Reference	133
9.10	src/gui/ProgressBar.java File Reference	133
9.11	src/intention/Intention.java File Reference	134
9.12	src/intention/Likelihood.java File Reference	134
9.13	src/intention/Metric.java File Reference	134
9.14	src/main/Launcher.java File Reference	134
9.14.1	Detailed Description	135

9.15	src/ontology/AnyOrder.java File Reference	135
9.16	src/ontology/Count.java File Reference	135
9.17	src/ontology/Exist.java File Reference	136
9.18	src/ontology/Ontology.java File Reference	136
9.19	src/ontology/OnTopWithContactStateRelation.java File Reference	136
9.20	src/ontology/OrderedList.java File Reference	136
9.21	src/ontology/OrderingConstruct.java File Reference	137
9.22	src/ontology/PartiallyInStateRelation.java File Reference	137
9.23	src/ontology/Predicate.java File Reference	137
9.24	src/ontology/UnderWithContactStateRelation.java File Reference	137
9.25	src/textfiles/FileOperator.java File Reference	138
9.26	src/tools/Configuration.java File Reference	138
9.27	src/treecheckbox/CheckTreeCellRenderer.java File Reference	138
9.28	src/treecheckbox/CheckTreeManager.java File Reference	138
9.29	src/treecheckbox/CheckTreeSelectionModel.java File Reference	139
9.30	src/treecheckbox/TreeExample.java File Reference	139
9.31	src/treecheckbox/TristateCheckBox.java File Reference	139

Chapter 1

Intention Recognition

This tool is capable of reading a plan file to 'recognize' which kit is being built.

Chapter 2

Module Index

2.1 Modules

Here is a list of all modules:

Graphical User Interface	11
Intention Structure	11

Chapter 3

Namespace Index

3.1 Packages

Here are the packages with brief descriptions (if available):

component	13
gui	13
intention	13
main	14
ontology	14
textfiles	14
tools	14
treecheckbox	15

Chapter 4

Class Index

4.1 Class List

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

ontology.AnyOrder	17
gui.Chart (Chart display for metrics and likelihoods)	18
treecheckbox.CheckTreeCellRenderer	24
treecheckbox.CheckTreeManager	25
treecheckbox.CheckTreeSelectionModel	27
gui.CommonGUIComponents (Common GUI components used across different files of the project)	29
tools.Configuration	33
ontology.Count	34
gui.Chart.CustomRenderer	36
gui.Chart.CustomRendererLine	37
gui.DemoPanel	39
gui.MainFrame.DisplayMetrics	40
DocumentFilter (Force a GUI component to contain a certain type of data (text, Integer, Double, etc))	41
gui.DrawStringPanel	42
ontology.Exist	44
textfiles.FileOperator	46
intention.Intention (Representation of intentions from their definition in the ontology)	47
component.IntFilter	66
main.Launcher (Main class of the tool)	68
intention.Likelihood	70
gui.MainFrame	70
intention.Metric (Definition of additive and multiplicative metrics)	76
gui.OptionFrame.MouseHandler	80
ontology.Ontology	81
Ontology (Class for the ontology)	103
gui.OntologyChooser	103

ontology.OnTopWithContactStateRelation	105
gui.OptionFrame	106
ontology.OrderedList	112
ontology.OrderingConstruct	114
ontology.PartiallyInStateRelation	114
gui.PDFChartTransferable	115
gui.OptionFrame.PlanComboBoxListener	117
ontology.Predicate	118
gui.ProgressBar	120
treecheckbox.TreeExample	122
treecheckbox.TristateCheckBox	124
treecheckbox.TristateCheckBox.TristateDecorator	126
ontology.UnderWithContactStateRelation	130

Chapter 5

File Index

5.1 File List

Here is a list of all files with brief descriptions:

src/component/IntFilter.java	131
src/gui/Chart.java	131
src/gui/CommonGUIComponents.java	131
src/gui/DemoPanel.java	132
src/gui/DrawStringPanel.java	132
src/gui/MainFrame.java	132
src/gui/OntologyChooser.java	133
src/gui/OptionFrame.java	133
src/gui/PDFChartTransferable.java	133
src/gui/ProgressBar.java	133
src/intention/Intention.java	134
src/intention/Likelihood.java	134
src/intention/Metric.java	134
src/main/Launcher.java (Contains the main of the program)	134
src/ontology/AnyOrder.java	135
src/ontology/Count.java	135
src/ontology/Exist.java	136
src/ontology/Ontology.java	136
src/ontology/OnTopWithContactStateRelation.java	136
src/ontology/OrderedList.java	136
src/ontology/OrderingConstruct.java	137
src/ontology/PartiallyInStateRelation.java	137
src/ontology/Predicate.java	137
src/ontology/UnderWithContactStateRelation.java	137
src/textfiles/FileOperator.java	138
src/tools/Configuration.java	138
src/treecheckbox/CheckTreeCellRenderer.java	138
src/treecheckbox/CheckTreeManager.java	138
src/treecheckbox/CheckTreeSelectionModel.java	139

src/treecheckbox/ TreeExample.java	139
src/treecheckbox/ TristateCheckBox.java	139

Chapter 6

Module Documentation

6.1 Graphical User Interface

Classes

- class [gui.Chart](#)
Chart display for metrics and likelihoods.
- class [gui.CommonGUIComponents](#)
Common GUI components used across different files of the project.

6.2 Intention Structure

Classes

- class [intention.Intention](#)
Representation of intentions from their definition in the ontology.
- class [intention.Metric](#)
Definition of additive and multiplicative metrics.

Chapter 7

Namespace Documentation

7.1 Package component

Classes

- class [IntFilter](#)

7.2 Package gui

Classes

- class [Chart](#)
Chart display for metrics and likelihoods.
- class [CommonGUIComponents](#)
Common GUI components used across different files of the project.
- class [DemoPanel](#)
- class [DrawStringPanel](#)
- class [MainFrame](#)
- class [OntologyChooser](#)
- class [OptionFrame](#)
- class [PDFChartTransferable](#)
- class [ProgressBar](#)

7.3 Package intention

Classes

- class [Intention](#)
Representation of intentions from their definition in the ontology.

- class [Likelihood](#)
- class [Metric](#)

Definition of additive and multiplicative metrics.

7.4 Package main

Classes

- class [Launcher](#)

Main class of the tool.

7.5 Package ontology

Classes

- class [AnyOrder](#)
- class [Count](#)
- class [Exist](#)
- class [Ontology](#)
- class [OnTopWithContactStateRelation](#)
- class [OrderedList](#)
- class [OrderingConstruct](#)
- class [PartiallyInStateRelation](#)
- class [Predicate](#)
- class [UnderWithContactStateRelation](#)

7.6 Package textfiles

Classes

- class [FileOperator](#)

7.7 Package tools

Classes

- class [Configuration](#)

7.8 Package treecheckbox

Classes

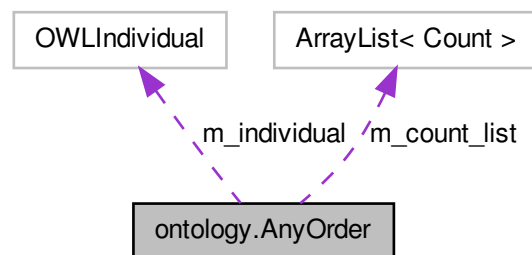
- class [CheckTreeCellRenderer](#)
- class [CheckTreeManager](#)
- class [CheckTreeSelectionModel](#)
- class [TreeExample](#)
- class [TristateCheckBox](#)

Chapter 8

Class Documentation

8.1 ontology.AnyOrder Class Reference

Collaboration diagram for ontology.AnyOrder:



Public Member Functions

- [AnyOrder](#) ()
- void [addCountToList](#) ([Count](#) c)
- [ArrayList<Count>](#) [getCountList](#) ()
- [OWLIndividual](#) [getIndividual](#) ()
- int [getPosition](#) ()
- int [getTotalNumber](#) ()
- void [setIndividual](#) ([OWLIndividual](#) i)
- void [setPosition](#) (int p)
- void [setTotalNumber](#) (int i)

Private Attributes

- `ArrayList< Count > m_count_list`
- `OWLIndividual m_individual`
- `int m_position`
- `int m_total_number`

8.1.1 Constructor & Destructor Documentation

8.1.1.1 `ontology.AnyOrder.AnyOrder ()`

8.1.2 Member Function Documentation

8.1.2.1 `void ontology.AnyOrder.addCountToList (Count c)`

8.1.2.2 `ArrayList<Count> ontology.AnyOrder.getCountList ()`

8.1.2.3 `OWLIndividual ontology.AnyOrder.getIndividual ()`

8.1.2.4 `int ontology.AnyOrder.getPosition ()`

8.1.2.5 `int ontology.AnyOrder.getTotalNumber ()`

8.1.2.6 `void ontology.AnyOrder.setIndividual (OWLIndividual i)`

8.1.2.7 `void ontology.AnyOrder.setPosition (int p)`

8.1.2.8 `void ontology.AnyOrder.setTotalNumber (int i)`

8.1.3 Member Data Documentation

8.1.3.1 `ArrayList<Count> ontology.AnyOrder.m_count_list` [private]

8.1.3.2 `OWLIndividual ontology.AnyOrder.m_individual` [private]

8.1.3.3 `int ontology.AnyOrder.m_position` [private]

8.1.3.4 `int ontology.AnyOrder.m_total_number` [private]

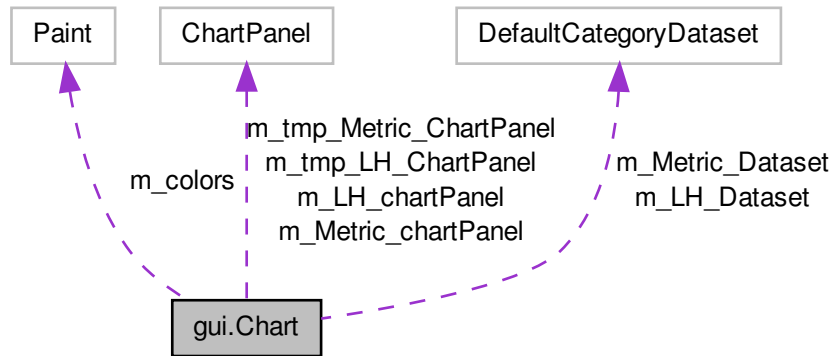
The documentation for this class was generated from the following file:

- `src/ontology/AnyOrder.java`

8.2 gui.Chart Class Reference

[Chart](#) display for metrics and likelihoods.

Collaboration diagram for gui.Chart:



Classes

- class [CustomRenderer](#)
- class [CustomRendererLine](#)

Public Member Functions

- [Chart](#) ()
Class constructor.
- LegendItemCollection [getLegendItems](#) (CombinedDomainCategoryPlot combined-domaincategoryplot)
Returns a collection of legend items for one of the subplot of a CombinedDomainCategoryPlot.
- ChartPanel [updateChart_metrics](#) (DefaultCategoryDataset metricsDataset)

Static Public Member Functions

- static void [setChartTitle](#) (String title_)
Set the title for the chart displaying intention likelihoods.
- static String [getChartTitle](#) ()
Get the title for the chart displaying intention likelihoods.
- static void [setMetricsChartTitle](#) (String title_)
Set the title for the chart displaying intention metrics.
- static String [getMetricsChartTitle](#) ()

Get the title for the chart displaying intention metrics.

- static ChartPanel [createChart_metrics](#) ()
- static void [createLikelihoodDataset](#) (double _likelihood, [Intention](#) _intention, int _state)
- static void [createMetricsDataset](#) ([Intention](#) _intention, int _state)

Create a data set for metrics.

Public Attributes

- ChartPanel [m_LH_chartPanel](#) = [createChart_likelihood](#)()
- ChartPanel [m_Metric_chartPanel](#) = [createChart_metrics](#)()
- ChartPanel [m_tmp_Metric_ChartPanel](#)
- ChartPanel [m_tmp_LH_ChartPanel](#)

Static Public Attributes

- static DefaultCategoryDataset [m_LH_Dataset](#) = new DefaultCategoryDataset()
Data set for likelihoods.
- static DefaultCategoryDataset [m_Metric_Dataset](#) = new DefaultCategoryDataset()

Data set for metrics.

Package Functions

- ChartPanel [updateChart_likelihood](#) (DefaultCategoryDataset dataset)

Static Package Attributes

- static Paint[] [m_colors](#)
Array of colors.

Private Member Functions

- ChartPanel [createChart_likelihood](#) ()

Static Private Attributes

- static String [m_LH_Chart_Title](#)
Title of the chart window.
- static String [m_Metric_Chart_Title](#)
Title of the metrics chart.

8.2.1 Detailed Description

[Chart](#) display for metrics and likelihoods.

This class consists of components that allow the display and selection of metrics and likelihoods

Author

[Zeid Kootbally](#) zeid.kootbally@nist.gov

Date

September 2013

8.2.2 Constructor & Destructor Documentation

8.2.2.1 `gui.Chart.Chart ()`

Class constructor.

- Allow the use of the mouse wheel to zoom in and out on the charts
- Allow the use of autoscroll on the charts

8.2.3 Member Function Documentation

8.2.3.1 `ChartPanel gui.Chart.createChart_likelihood () [private]`

8.2.3.2 `static ChartPanel gui.Chart.createChart_metrics () [static]`

8.2.3.3 `static void gui.Chart.createLikelihoodDataset (double _likelihood, Intention _intention, int _state) [static]`

8.2.3.4 `static void gui.Chart.createMetricsDataset (Intention _intention, int _state) [static]`

Create a data set for metrics.

Parameters

<code>_intention</code>	The intention from which we will retrieve the metrics
<code>_state</code>	The current state

8.2.3.5 `static String gui.Chart.getChartTitle () [static]`

Get the title for the chart displaying intention likelihoods.

8.2.3.6 `LegendItemCollection` `gui.Chart.getLegendItems (CombinedDomainCategoryPlot combineddomaincategoryplot)`

Returns a collection of legend items for one of the subplot of a `CombinedDomainCategoryPlot`.

Parameters

<i>combined-domaincategoryplot</i>	Instance of <code>CombinedDomainCategoryPlot</code>
------------------------------------	---

Returns

The legend items.

8.2.3.7 `static String` `gui.Chart.getMetricsChartTitle () [static]`

Get the title for the chart displaying intention metrics.

8.2.3.8 `static void` `gui.Chart.setChartTitle (String title_) [static]`

Set the title for the chart displaying intention likelihoods.

Parameters

<i>title_</i>	Title of the Likelihood chart
---------------	-------------------------------

8.2.3.9 `static void` `gui.Chart.setMetricsChartTitle (String title_) [static]`

Set the title for the chart displaying intention metrics.

Parameters

<i>title_</i>	Title of the Metrics chart
---------------	----------------------------

8.2.3.10 `ChartPanel` `gui.Chart.updateChart_likelihood (DefaultCategoryDataset dataset) [package]`

8.2.3.11 `ChartPanel` `gui.Chart.updateChart_metrics (DefaultCategoryDataset metricsDataset)`

8.2.4 Member Data Documentation

8.2.4.1 `Paint[]` `gui.Chart.m_colors [static, package]`

Initial value:


```
new Paint[] { CommonGUIComponents.intentionColor9,  
              CommonGUIComponents.intentionColor2, CommonGUIComponents.intentionColor3,  
              CommonGUIComponents.intentionColor10,  
              CommonGUIComponents.intentionColor1, CommonGUIComponents.intentionColor6,  
              CommonGUIComponents.intentionColor7,  
              CommonGUIComponents.intentionColor8 }
```

Array of colors.

8.2.4.2 String gui.Chart.m_LH_Chart_Title [static, private]

Title of the chart window.

8.2.4.3 ChartPanel gui.Chart.m_LH_chartPanel = createChart.likelihood()

8.2.4.4 DefaultCategoryDataset gui.Chart.m_LH_Dataset = new DefaultCategoryDataset() [static]

Data set for likelihoods.

8.2.4.5 String gui.Chart.m_Metric_Chart_Title [static, private]

Title of the metrics chart.

8.2.4.6 ChartPanel gui.Chart.m_Metric_chartPanel = createChart_metrics()

8.2.4.7 DefaultCategoryDataset gui.Chart.m_Metric_Dataset = new DefaultCategoryDataset() [static]

Data set for metrics.

8.2.4.8 ChartPanel gui.Chart.m_tmp_LH_ChartPanel

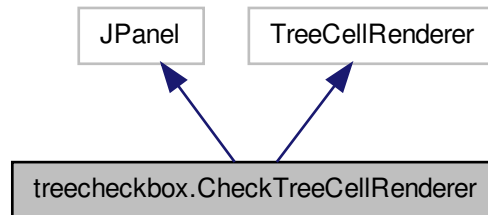
8.2.4.9 ChartPanel gui.Chart.m_tmp_Metric_ChartPanel

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

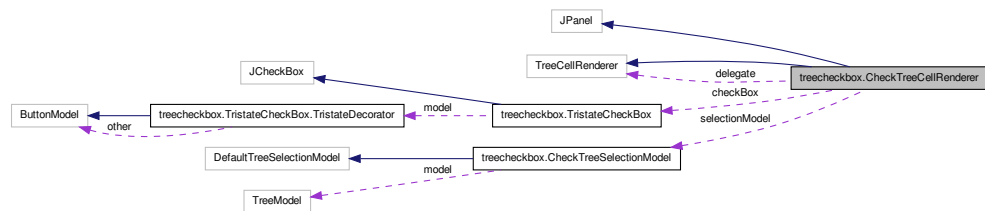
- src/gui/[Chart.java](#)

8.3 treecheckbox.CheckTreeCellRenderer Class Reference

Inheritance diagram for treecheckbox.CheckTreeCellRenderer:



Collaboration diagram for treecheckbox.CheckTreeCellRenderer:



Public Member Functions

- [CheckTreeCellRenderer](#) ([TreeCellRenderer](#) `delegate`, [CheckTreeSelectionModel](#) `selectionModel`)
- Component [getTreeCellRendererComponent](#) (`JTree` `tree`, `Object` `value`, `boolean` `selected`, `boolean` `expanded`, `boolean` `leaf`, `int` `row`, `boolean` `hasFocus`)

Private Attributes

- [CheckTreeSelectionModel](#) `selectionModel`
- `TreeCellRenderer` `delegate`
- [TristateCheckBox](#) `checkBox` = new [TristateCheckBox](#)()

8.3.1 Constructor & Destructor Documentation

8.3.1.1 treecheckbox.CheckTreeCellRenderer.CheckTreeCellRenderer (*TreeCellRenderer delegate*, *CheckTreeSelectionModel selectionModel*)

8.3.2 Member Function Documentation

8.3.2.1 Component treecheckbox.CheckTreeCellRenderer.getTreeCellRendererComponent (*JTree tree*, *Object value*, *boolean selected*, *boolean expanded*, *boolean leaf*, *int row*, *boolean hasFocus*)

8.3.3 Member Data Documentation

8.3.3.1 TristateCheckBox treecheckbox.CheckTreeCellRenderer.checkBox = new TristateCheckBox() [private]

8.3.3.2 TreeCellRenderer treecheckbox.CheckTreeCellRenderer.delegate [private]

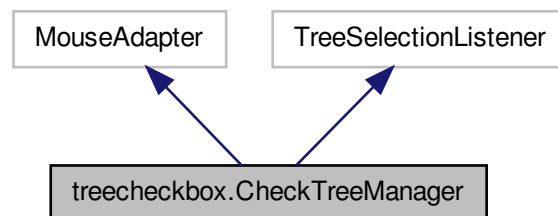
8.3.3.3 CheckTreeSelectionModel treecheckbox.CheckTreeCellRenderer.selectionModel [private]

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

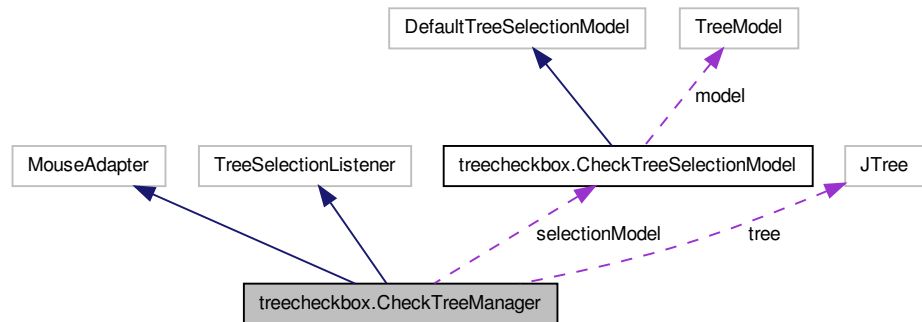
- src/treecheckbox/[CheckTreeCellRenderer.java](#)

8.4 treecheckbox.CheckTreeManager Class Reference

Inheritance diagram for treecheckbox.CheckTreeManager:



Collaboration diagram for `treecheckbox.CheckTreeManager`:



Public Member Functions

- [CheckTreeManager](#) ([JTree](#) `tree`)
- void [addChildPaths](#) ([TreePath](#) `path`, [TreeModel](#) `model`, [List](#) `result`)
- [ArrayList](#) [getDescendants](#) ([TreePath](#) `paths[]`, [TreeModel](#) `model`)
- [ArrayList](#)< [Object](#) > [getAllCheckedPaths](#) ([CheckTreeManager](#) `manager`, [JTree](#) `tree`)
- void [mouseClicked](#) ([MouseEvent](#) `me`)
- [CheckTreeSelectionModel](#) [getSelectionModel](#) ()
- void [valueChanged](#) ([TreeSelectionEvent](#) `e`)

Package Attributes

- int [hotspot](#) = new [JCheckBox](#)().getPreferredSize().width

Private Attributes

- [CheckTreeSelectionModel](#) [selectionModel](#)
- [JTree](#) [tree](#) = new [JTree](#)()

8.4.1 Constructor & Destructor Documentation

8.4.1.1 `treecheckbox.CheckTreeManager.CheckTreeManager (JTree tree)`

8.4.2 Member Function Documentation

- 8.4.2.1 void treecheckbox.CheckTreeManager.addChildPaths (*TreePath path*, *TreeModel model*, *List result*)
- 8.4.2.2 ArrayList<Object> treecheckbox.CheckTreeManager.getAllCheckedPaths (*CheckTreeManager manager*, *JTree tree*)
- 8.4.2.3 ArrayList treecheckbox.CheckTreeManager.getDescendants (*TreePath paths[]*, *TreeModel model*)
- 8.4.2.4 CheckTreeSelectionModel treecheckbox.CheckTreeManager.getSelectionModel ()
- 8.4.2.5 void treecheckbox.CheckTreeManager.mouseClicked (*MouseEvent me*)
- 8.4.2.6 void treecheckbox.CheckTreeManager.valueChanged (*TreeSelectionEvent e*)

8.4.3 Member Data Documentation

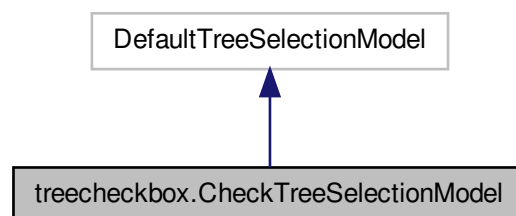
- 8.4.3.1 int treecheckbox.CheckTreeManager.hotspot = new JCheckBox().getPreferredSize().width [package]
- 8.4.3.2 CheckTreeSelectionModel treecheckbox.CheckTreeManager.selectionModel [private]
- 8.4.3.3 JTree treecheckbox.CheckTreeManager.tree = new JTree() [private]

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

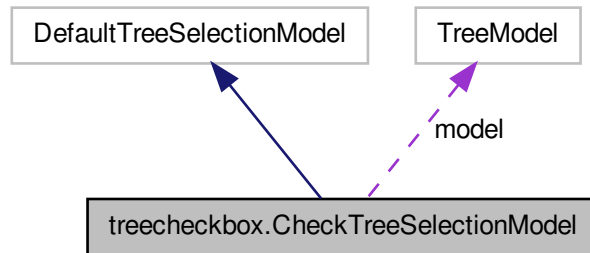
- src/treecheckbox/[CheckTreeManager.java](#)

8.5 treecheckbox.CheckTreeSelectionModel Class Reference

Inheritance diagram for treecheckbox.CheckTreeSelectionModel:



Collaboration diagram for `treecheckbox.CheckTreeSelectionModel`:



Public Member Functions

- [CheckTreeSelectionModel](#) (`TreeModel model`)
- boolean [isPartiallySelected](#) (`TreePath path`)
- boolean [isPathSelected](#) (`TreePath path`, boolean `dig`)
- void [setSelectionPaths](#) (`TreePath[] pPaths`)
- void [addSelectionPaths](#) (`TreePath[] paths`)
- void [removeSelectionPaths](#) (`TreePath[] paths`)

Private Member Functions

- boolean [isDescendant](#) (`TreePath path1`, `TreePath path2`)
- boolean [areSiblingsSelected](#) (`TreePath path`)
- void [toggleRemoveSelection](#) (`TreePath path`)

Private Attributes

- `TreeModel model`

8.5.1 Constructor & Destructor Documentation

8.5.1.1 `treecheckbox.CheckTreeSelectionModel.CheckTreeSelectionModel (TreeModel model)`

8.5.2 Member Function Documentation

8.5.2.1 `void treecheckbox.CheckTreeSelectionModel.addSelectionPaths (TreePath[] paths)`

- 8.5.2.2 `boolean treecheckbox.CheckTreeSelectionModel.areSiblingsSelected (TreePath path)`
[private]
- 8.5.2.3 `boolean treecheckbox.CheckTreeSelectionModel.isDescendant (TreePath path1,
TreePath path2)` [private]
- 8.5.2.4 `boolean treecheckbox.CheckTreeSelectionModel.isPartiallySelected (TreePath path)`
- 8.5.2.5 `boolean treecheckbox.CheckTreeSelectionModel.isPathSelected (TreePath path,
boolean dig)`
- 8.5.2.6 `void treecheckbox.CheckTreeSelectionModel.removeSelectionPaths (TreePath[] paths)`
- 8.5.2.7 `void treecheckbox.CheckTreeSelectionModel.setSelectionPaths (TreePath[] pPaths)`
- 8.5.2.8 `void treecheckbox.CheckTreeSelectionModel.toggleRemoveSelection (TreePath path)`
[private]

8.5.3 Member Data Documentation

- 8.5.3.1 `TreeModel treecheckbox.CheckTreeSelectionModel.model` [private]

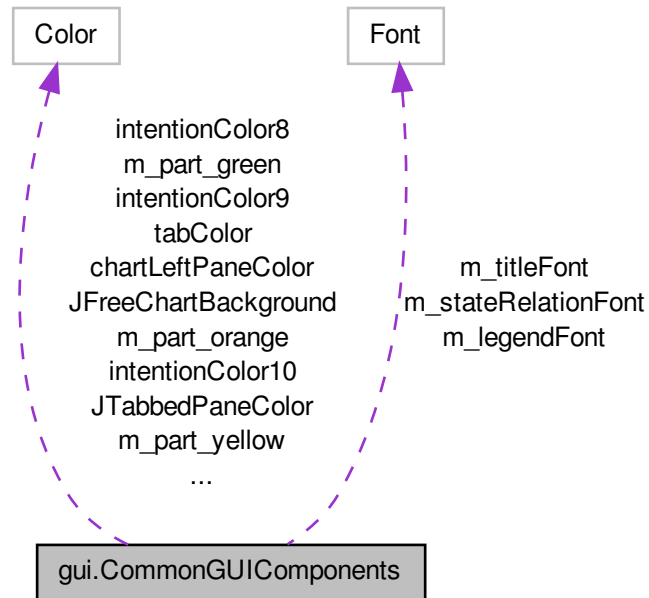
The documentation for this class was generated from the following file:

- `src/treecheckbox/CheckTreeSelectionModel.java`

8.6 gui.CommonGUIComponents Class Reference

Common GUI components used across different files of the project.

Collaboration diagram for gui.CommonGUIComponents:



Static Public Attributes

- static Color `chartPanelColor` = new Color(160, 188, 136)
- static Color `menuBarColor` = new Color(75, 148, 48)
- static Color `tabColor` = new Color(255, 255, 255)
- static Color `chartLeftPaneColor` = new Color(255, 255, 255)
- static Color `intentionColor1` = new Color(255, 0, 200)
- static Color `intentionColor2` = new Color(255, 145, 0)
- static Color `intentionColor3` = new Color(118, 175, 60)
- static Color `intentionColor6` = new Color(255, 255, 255)
- static Color `intentionColor7` = new Color(255, 111, 0)
- static Color `intentionColor8` = new Color(0, 255, 34)
- static Color `intentionColor9` = new Color(0, 137, 255)
- static Color `intentionColor10` = new Color(64, 73, 81)
- static Color `state_color` = new Color(255, 0, 0)
- static Color `JTabbedPaneColor` = new Color(255, 145, 0)
- static Color `JFreeChartBackground` = new Color(214, 224, 219)
- static Color `m_part_orange` = new Color(233, 148, 0)

- static Color `m_part_green` = new Color(71, 144, 30)
- static Color `m_part_yellow` = new Color(236, 236, 43)
- static Color `m_state` = new Color(129, 129, 120)
- static Font `m_titleFont` = new Font("Times",Font.PLAIN, 18)
- static Font `m_legendFont` = new Font("Times",Font.PLAIN, 13)
- static Font `m_stateRelationFont` = new Font("Times",Font.PLAIN, 14)

Static Private Attributes

- static Color `intentionColor4` = new Color(50, 75, 156)
- static Color `intentionColor5` = new Color(242, 9, 9)

8.6.1 Detailed Description

Common GUI components used across different files of the project.

This class defines colors and fonts that are used in multiple source files.

Author

Zeid Kootbally *zeid.kootbally@nist.gov*

Date

September 2013

8.6.2 Member Data Documentation

- 8.6.2.1 Color `gui.CommonGUIComponents.chartLeftPaneColor` = new Color(255, 255, 255) [static]
- 8.6.2.2 Color `gui.CommonGUIComponents.chartPanelColor` = new Color(160, 188, 136) [static]
- 8.6.2.3 Color `gui.CommonGUIComponents.intentionColor1` = new Color(255, 0, 200) [static]
- 8.6.2.4 Color `gui.CommonGUIComponents.intentionColor10` = new Color(64, 73, 81) [static]
- 8.6.2.5 Color `gui.CommonGUIComponents.intentionColor2` = new Color(255, 145, 0) [static]
- 8.6.2.6 Color `gui.CommonGUIComponents.intentionColor3` = new Color(118, 175, 60) [static]
- 8.6.2.7 Color `gui.CommonGUIComponents.intentionColor4` = new Color(50, 75, 156) [static, private]

- 8.6.2.8 **Color** `gui.CommonGUIComponents.intentionColor5 = new Color(242, 9, 9)`
[static, private]
- 8.6.2.9 **Color** `gui.CommonGUIComponents.intentionColor6 = new Color(255, 255, 255)`
[static]
- 8.6.2.10 **Color** `gui.CommonGUIComponents.intentionColor7 = new Color(255, 111, 0)`
[static]
- 8.6.2.11 **Color** `gui.CommonGUIComponents.intentionColor8 = new Color(0, 255, 34)`
[static]
- 8.6.2.12 **Color** `gui.CommonGUIComponents.intentionColor9 = new Color(0, 137, 255)`
[static]
- 8.6.2.13 **Color** `gui.CommonGUIComponents.JFreeChartBackground = new Color(214,224,219)` [static]
- 8.6.2.14 **Color** `gui.CommonGUIComponents.JTabbedPaneColor = new Color(255, 145, 0)` [static]
- 8.6.2.15 **Font** `gui.CommonGUIComponents.m_legendFont = new Font("Times",Font.PLAIN, 13)` [static]
- 8.6.2.16 **Color** `gui.CommonGUIComponents.m_part_green = new Color(71, 144, 30)`
[static]
- 8.6.2.17 **Color** `gui.CommonGUIComponents.m_part_orange = new Color(233, 148, 0)`
[static]
- 8.6.2.18 **Color** `gui.CommonGUIComponents.m_part_yellow = new Color(236, 236, 43)`
[static]
- 8.6.2.19 **Color** `gui.CommonGUIComponents.m_state = new Color(129, 129, 120)`
[static]
- 8.6.2.20 **Font** `gui.CommonGUIComponents.m_stateRelationFont = new Font("Times",Font.PLAIN, 14)` [static]
- 8.6.2.21 **Font** `gui.CommonGUIComponents.m_titleFont = new Font("Times",Font.PLAIN, 18)` [static]
- 8.6.2.22 **Color** `gui.CommonGUIComponents.menuBarColor = new Color(75, 148, 48)`
[static]
- 8.6.2.23 **Color** `gui.CommonGUIComponents.state_color = new Color(255, 0, 0)`
[static]

8.6.2.24 **Color** `gui.CommonGUIComponents.tabColor = new Color(255, 255, 255)`
[static]

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

- `src/gui/CommonGUIComponents.java`

8.7 tools.Configuration Class Reference

Static Public Member Functions

- static boolean `isWindows` ()
- static boolean `isMac` ()
- static boolean `isUnix` ()

Static Public Attributes

- static String `m_OS` = `System.getProperty("os.name").toLowerCase()`

8.7.1 Member Function Documentation

8.7.1.1 static boolean `tools.Configuration.isMac` () [static]

8.7.1.2 static boolean `tools.Configuration.isUnix` () [static]

8.7.1.3 static boolean `tools.Configuration.isWindows` () [static]

8.7.2 Member Data Documentation

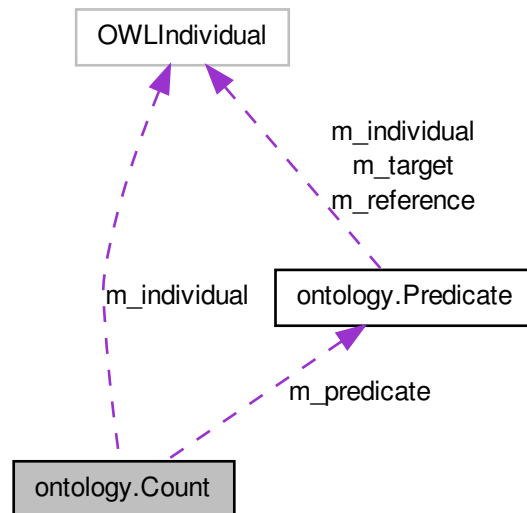
8.7.2.1 String `tools.Configuration.m_OS` = `System.getProperty("os.name").toLowerCase()`
[static]

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

- `src/tools/Configuration.java`

8.8 ontology.Count Class Reference

Collaboration diagram for ontology.Count:



Public Member Functions

- [Count](#) ()
- [Predicate](#) [getPredicate](#) ()
- `OWLIndividual` [getIndividual](#) ()
- `Integer` [getOccurrence](#) ()
- `void` [setPredicate](#) ([Predicate](#) p)
- `void` [setIndividual](#) (`OWLIndividual` i)
- `void` [setOccurrence](#) (`Integer` o)

Private Attributes

- [Predicate](#) [m_predicate](#)
- `Integer` [m_occurrence](#)
- `OWLIndividual` [m_individual](#)

8.8.1 Constructor & Destructor Documentation

8.8.1.1 ontology.Count.Count ()

8.8.2 Member Function Documentation

8.8.2.1 OWLIndividual ontology.Count.getIndividual ()

8.8.2.2 Integer ontology.Count.getOccurrence ()

8.8.2.3 Predicate ontology.Count.getPredicate ()

8.8.2.4 void ontology.Count.setIndividual (OWLIndividual *i*)

8.8.2.5 void ontology.Count.setOccurrence (Integer *o*)

8.8.2.6 void ontology.Count.setPredicate (Predicate *p*)

8.8.3 Member Data Documentation

8.8.3.1 OWLIndividual ontology.Count.m_individual [private]

8.8.3.2 Integer ontology.Count.m_occurrence [private]

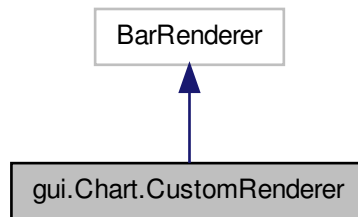
8.8.3.3 Predicate ontology.Count.m_predicate [private]

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

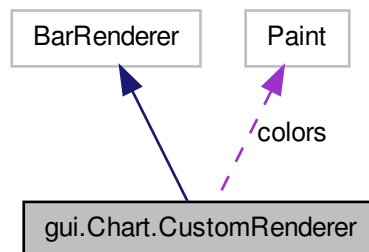
- src/ontology/Count.java

8.9 gui.Chart.CustomRenderer Class Reference

Inheritance diagram for gui.Chart.CustomRenderer:



Collaboration diagram for gui.Chart.CustomRenderer:



Public Member Functions

- `CustomRenderer ()`
Read an array of colors and select each color in order they are in the array.
- `Paint getItemPaint (final int row, final int column)`

Private Attributes

- `Paint[] colors`

Static Private Attributes

- static final long [serialVersionUID](#) = 1L

8.9.1 Constructor & Destructor Documentation

8.9.1.1 `gui.Chart.CustomRenderer.CustomRenderer ()`

Read an array of colors and select each color in order they are in the array.

8.9.2 Member Function Documentation

8.9.2.1 `Paint gui.Chart.CustomRenderer.getItemPaint (final int row, final int column)`

8.9.3 Member Data Documentation

8.9.3.1 `Paint[] gui.Chart.CustomRenderer.colors [private]`

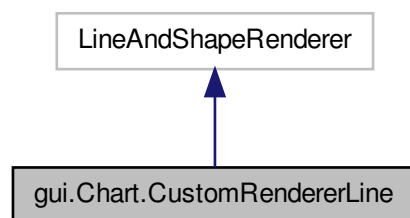
8.9.3.2 `final long gui.Chart.CustomRenderer.serialVersionUID = 1L [static, private]`

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

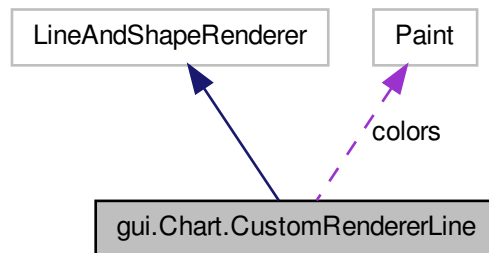
- `src/gui/Chart.java`

8.10 `gui.Chart.CustomRendererLine` Class Reference

Inheritance diagram for `gui.Chart.CustomRendererLine`:



Collaboration diagram for gui.Chart.CustomRendererLine:



Public Member Functions

- [CustomRendererLine](#) ()
- [Paint getItemPaint](#) (final int row, final int column)

Private Attributes

- [Paint\[\] colors](#)

Static Private Attributes

- static final long [serialVersionUID](#) = 1L

8.10.1 Constructor & Destructor Documentation

8.10.1.1 [gui.Chart.CustomRendererLine.CustomRendererLine](#) ()

8.10.2 Member Function Documentation

8.10.2.1 [Paint gui.Chart.CustomRendererLine.getItemPaint](#) (final int row, final int column)

8.10.3 Member Data Documentation

8.10.3.1 [Paint \[\] gui.Chart.CustomRendererLine.colors](#) [private]

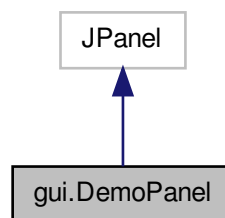
8.10.3.2 `final long gui.Chart.CustomRendererLine.serialVersionUID = 1L`
[static, private]

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

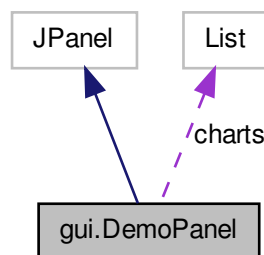
- `src/gui/Chart.java`

8.11 gui.DemoPanel Class Reference

Inheritance diagram for gui.DemoPanel:



Collaboration diagram for gui.DemoPanel:



Public Member Functions

- [DemoPanel](#) (java.awt.LayoutManager layoutmanager)
- void [addChart](#) (JFreeChart jfreechart)
- JFreeChart[] [getCharts](#) ()

Package Attributes

- List [charts](#)

8.11.1 Constructor & Destructor Documentation

8.11.1.1 `gui.DemoPanel.DemoPanel (java.awt.LayoutManager layoutmanager)`

8.11.2 Member Function Documentation

8.11.2.1 `void gui.DemoPanel.addChart (JFreeChart jfreechart)`

8.11.2.2 `JFreeChart [] gui.DemoPanel.getCharts ()`

8.11.3 Member Data Documentation

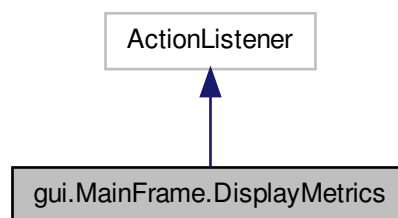
8.11.3.1 List `gui.DemoPanel.charts` [package]

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

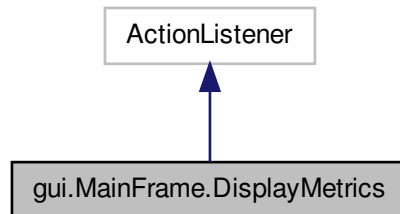
- src/gui/[DemoPanel.java](#)

8.12 gui.MainFrame.DisplayMetrics Class Reference

Inheritance diagram for gui.MainFrame.DisplayMetrics:



Collaboration diagram for gui.MainFrame.DisplayMetrics:



Public Member Functions

- void [actionPerformed](#) (ActionEvent e)

8.12.1 Detailed Description

Listener for the metrics selection in the JTree of checkboxes

8.12.2 Member Function Documentation

8.12.2.1 void `gui.MainFrame.DisplayMetrics.actionPerformed` (ActionEvent *e*)

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

- [src/gui/MainFrame.java](#)

8.13 DocumentFilter Class Reference

Force a GUI component to contain a certain type of data (text, Integer, Double, etc)

8.13.1 Detailed Description

Force a GUI component to contain a certain type of data (text, Integer, Double, etc)

The user will not be able to enter any type of data other than the one specified. For instance, if the GUI component can contain only Integer and the user wants to type 3.14, the "." will not be enabled.

Author

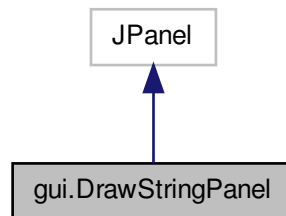
Zeid Kootbally *zeid.kootbally@nist.gov*

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

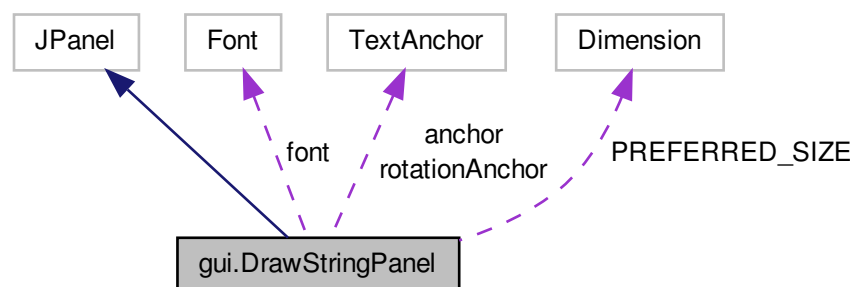
- `src/component/IntFilter.java`

8.14 gui.DrawStringPanel Class Reference

Inheritance diagram for gui.DrawStringPanel:



Collaboration diagram for gui.DrawStringPanel:



Public Member Functions

- [DrawStringPanel](#) (String s, boolean flag)
- Dimension [getPreferredSize](#) ()
- void [setAnchor](#) (TextAnchor textanchor)
- void [setRotationAnchor](#) (TextAnchor textanchor)
- void [setAngle](#) (double d)
- Font [getFont](#) ()
- void [setFont](#) (Font font1)
- void [paintComponent](#) (Graphics g)

Private Attributes

- boolean [rotate](#)
- String [text](#)
- TextAnchor [anchor](#)
- TextAnchor [rotationAnchor](#)
- Font [font](#)
- double [angle](#)

Static Private Attributes

- static final Dimension [PREFERRED_SIZE](#) = new Dimension(500, 300)

8.14.1 Constructor & Destructor Documentation

8.14.1.1 `gui.DrawStringPanel.DrawStringPanel (String s, boolean flag)`

8.14.2 Member Function Documentation

8.14.2.1 `Font gui.DrawStringPanel.getFont ()`

8.14.2.2 `Dimension gui.DrawStringPanel.getPreferredSize ()`

8.14.2.3 `void gui.DrawStringPanel.paintComponent (Graphics g)`

8.14.2.4 `void gui.DrawStringPanel.setAnchor (TextAnchor textanchor)`

8.14.2.5 `void gui.DrawStringPanel.setAngle (double d)`

8.14.2.6 `void gui.DrawStringPanel.setFont (Font font1)`

8.14.2.7 `void gui.DrawStringPanel.setRotationAnchor (TextAnchor textanchor)`

8.14.3 Member Data Documentation

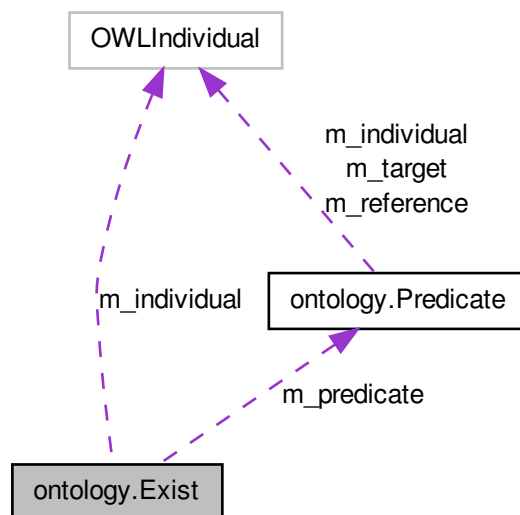
- 8.14.3.1 `TextAnchor gui.DrawStringPanel.anchor` [private]
- 8.14.3.2 `double gui.DrawStringPanel.angle` [private]
- 8.14.3.3 `Font gui.DrawStringPanel.font` [private]
- 8.14.3.4 `final Dimension gui.DrawStringPanel.PREFERRED_SIZE = new Dimension(500, 300)` [static, private]
- 8.14.3.5 `boolean gui.DrawStringPanel.rotate` [private]
- 8.14.3.6 `TextAnchor gui.DrawStringPanel.rotationAnchor` [private]
- 8.14.3.7 `String gui.DrawStringPanel.text` [private]

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

- [src/gui/DrawStringPanel.java](#)

8.15 ontology.Exist Class Reference

Collaboration diagram for ontology.Exist:



Public Member Functions

- [Exist](#) ()
- int [getTotalNumber](#) ()
- void [setTotalNumber](#) (int i)
- OWLIndividual [getIndividual](#) ()
- void [setIndividual](#) (OWLIndividual i)
- int [getOccurrence](#) ()
- void [setOccurrence](#) (int o)
- int [getPosition](#) ()
- void [setPosition](#) (int p)
- [Predicate](#) [getPredicate](#) ()
- void [setPredicate](#) ([Predicate](#) p)

Private Attributes

- int [occurrence](#)
- OWLIndividual [m_individual](#)
- int [m_position](#)
- int [m_total_number](#)
- [Predicate](#) [m_predicate](#)

8.15.1 Detailed Description

This class manages the Ordering Construct [Exist](#)

Author

Zeid Kootbally

8.15.2 Constructor & Destructor Documentation

8.15.2.1 ontology.Exist.Exist ()

Default Constructor

8.15.3 Member Function Documentation

8.15.3.1 OWLIndividual ontology.Exist.getIndividual ()

8.15.3.2 int ontology.Exist.getOccurrence ()

8.15.3.3 int ontology.Exist.getPosition ()

8.15.3.4 Predicate ontology.Exist.getPredicate ()

8.15.3.5 `int ontology.Exist.getTotalNumber ()`

8.15.3.6 `void ontology.Exist.setIndividual (OWLIndividual i)`

8.15.3.7 `void ontology.Exist.setOccurrence (int o)`

8.15.3.8 `void ontology.Exist.setPosition (int p)`

8.15.3.9 `void ontology.Exist.setPredicate (Predicate p)`

8.15.3.10 `void ontology.Exist.setTotalNumber (int i)`

8.15.4 Member Data Documentation

8.15.4.1 `OWLIndividual ontology.Exist.m_individual` [private]

The OWLIndividual for an Instance [Exist](#) from the ontology

8.15.4.2 `int ontology.Exist.m_position` [private]

8.15.4.3 `Predicate ontology.Exist.m_predicate` [private]

8.15.4.4 `int ontology.Exist.m_total_number` [private]

8.15.4.5 `int ontology.Exist.occurrence` [private]

Occurrence of an instance of [Exist](#) in an Intention

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

- `src/ontology/Exist.java`

8.16 textfiles.FileOperator Class Reference

Public Member Functions

- [FileOperator](#) ()
- `String[] openFile` (String path) throws IOException
- `int readLines` (String path) throws IOException
- `ArrayList< ArrayList< ArrayList< String > > > translatePlanToStateRelation` (String[] plan)

Static Public Member Functions

- static void [saveAllKitsData](#) (String file_path_, FileWriter writer, String kit_, String plan_) throws IOException

Public Member Functions

- [Intention](#) ()
Class constructor.
- void [builDetrimentalList](#) ([Exist](#) _exist)
Add an ordering construct of type Exist to the list of detrimental state relations.
- [ArrayList](#)< [Exist](#) > [getDetrimentalList](#) ()
Return the list of detrimental state relations for an intention.
- [OWLIndividual](#) [getIndividual](#) ()
Return an intention of type OWLIndividual.
- [String](#) [getIntentionName](#) ()
Return the name of the intention.
- int [getNumberStateRelation](#) ()
Return the number of state relations that constitute an intention.
- [OrderedList](#) [getOrderedList](#) ()
Return the ordering construct OrderedList for an intention.
- double [getM_am1](#) ()
Return the value of AM_1 .
- double [getM_am2](#) ()
Return the value of AM_2 .
- double [getM_am3](#) ()
Return the value of AM_3 .
- double [getM_am4](#) ()
Return the value of AM_4 .
- double [getM_am5](#) ()
Return the value of AM_5 .
- double [getM_mm1](#) ()
Return the value of MM_1 .
- double [getM_percentComplete_i_s](#) ()
Return the percentage of completion.
- double [getM_percentProductive_i_s](#) ()
Return the percentage of productive states.
- int [getM_SR_i_s](#) ()
Return the number of matched state relations (SR) in an intention (i) as of the current state (s)
- int [getM_SR_Total](#) ()
Return the number of state relations (SR) (whether matched or not) in an intention (i)
- [Map](#)< [Integer](#), [Integer](#) > [getM_map_SRirs](#) ()
Return the HashMap that contains $SR_{i,r,s}$.
- int [getM_Si](#) ()
Return the number of states (S) that have occurred since (and including) the first matched state relation in an intention (i)
- int [getM_SR_i_r_s](#) ()
Return the value of $SR_{i,r,s}$ for the current intention.

- int [getM_found_detrimental_SR](#) ()
Return the number of detrimental state relations found for an intention.
- int [getM_intention_number](#) ()
Return the ID of an intention.
- double [getM_likelihood_observation](#) ()
Return the likelihood of observation under a kit.
- double [getM_probability_kit_observation](#) ()
Return the probability of a kit given observations.
- void [setExist](#) (Exist exist_)
Set the instance of the Java Class Exist to an intention.
- void [setIntentionName](#) (String intention_name_)
Set the name of the intention.
- void [setDetrimentalList](#) (ArrayList< Exist > detrimental_list_)
Set the list of detrimental state relations.
- void [setIndividual](#) (OWLIndividual individual_)
Set the intention as an OWL Individual.
- void [setNumberStateRelation](#) (int number_state_relation_)
Set the number of state relations to an intention.
- void [setOrderedList](#) (OrderedList orderedlist_)
Set the instance of Ordered List to the intention.
- void [setM_found_detrimental_SR](#) (int found_detrimental_SR_)
Set the number of detrimental state relations.
- void [setM_am1](#) (double am1_)
Set the value for AM_1 .
- void [setM_am2](#) (double am2_)
Set the value for AM_2 .
- void [setM_am3](#) (double am3_)
Set the value for AM_3 .
- void [setM_am4](#) (double am4_)
Set the value for AM_4 .
- void [setM_am5](#) (double am5_)
Set the value for AM_5 .
- void [setM_mm1](#) (double mm1_)
Set the value for MM_1 .
- void [setM_percentComplete_i_s](#) (double percentComplete_i_s_)
Set the percentage of completion.
- void [setM_percentProductive_i_s](#) (double percentProductive_i_s_)
Set the percentage of productive states.
- void [setM_SR_i_s](#) (int SR_i_s_)
Set the number of matched state relations (SR) in an intention (i) as of the current state (s)
- void [setM_SR_Total](#) (int SR_Total_)
Set the number of state relations (SR) (whether matched or not) in an intention (i)

- void [setM_Si](#) (int Si_)
Set the number of states (S) that have occurred since (and including) the first matched state relation in an intention (i)
- void [setM_map_SRirs](#) (Map< Integer, Integer > SRirs_)
Set a HashMap to m_SRirs .
- void [setM_SR_i_r_s](#) (int SR_i_r_s_)
Set the value of $SR_{i,r,s}$ for the current intention.
- void [setM_intention_number](#) (int intention_number_)
Set an ID to an intention.
- void [setM_likelihood_observation](#) (double likelihood_observation_)
Set the likelihood of observation under a kit.
- void [setM_probability_kit_observation](#) (double probability_kit_observation_)
Set the probability of a kit given observations.

Static Public Member Functions

- static String [getM_built_kit](#) ()
Return the kit chosen by the user in [gui.OptionFrame](#).
- static String [getM_selected_plan](#) ()
Get the plan chosen by the user in [gui.OptionFrame](#).
- static void [setM_built_kit](#) (String built_kit_)
Set the kit the user chose to build in [gui.OptionFrame](#).
- static void [setM_selected_plan](#) (String selected_plan_)
Set the value of $m_selected_plan$ from the plan selected by the user.

Public Attributes

- HashMap< String, String > [m_intention_orderingConstruct_list](#)

Private Attributes

- ArrayList< [Exist](#) > [m_detrimental_list](#)
List that contains detrimental states.
- [Exist](#) [m_exist](#)
Instance of the ordering construct [Exist](#).
- String [m_intention_name](#)
Name of the intention.
- [AnyOrder](#) [m_anyorder](#)
Instance of the ordering construct [AnyOrder](#).
- [OrderedList](#) [m_orderedlist](#)
Instance of the ordering construct [OrderedList](#).
- OWLIndividual [m_individual](#)

OWLIndividual (instance of the class [Intention](#))

- int [m_number_state_relation](#)

Number of state relations that consist an intention.

- int [m_SR_Total](#)

Number of state relations (SR) (whether matched or not) in an intention (i)

- int [m_Si](#)

Number of states (S) that have occurred since (and including) the first matched state relation in an intention (i)

- int [m_found_detrimental_SR](#)

Number of detrimental state relations (detrimentalSR) that have occurred in an intention (i) as of the current state (s).

- double [m_am1](#)

AM_1 : Number of observed state relations that are true in an intention (compared to other intentions).

- double [m_am2](#)

AM_2 : Percentage of an intention that is complete.

- double [m_am3](#)

AM_3 : Number of productive states that have occurred since the first productive state relation in an intention.

- double [m_am4](#)

AM_4 : Number of productive states that have occurred (recently) in the past r states.

- double [m_am5](#)

AM_5 : Probability of an intention (i) being recognized based on an observation (j).

- double [m_mm1](#)

MM_1 : The number of detrimental states.

- double [m_percentComplete_i_s](#)

- double [m_percentProductive_i_s](#)

- int [m_SR_i_s](#)

- int [m_SR_i_r_s](#)

- int [m_intention_number](#)

- double [m_likelihood_observation](#)

- double [m_probability_kit_observation](#)

- Map< Integer, Integer > [m_SRirs](#) = new HashMap<Integer, Integer>()

Static Private Attributes

- static String [m_built_kit](#)

- static String [m_selected_plan](#)

8.17.1 Detailed Description

Representation of intentions from their definition in the ontology.

This class consists of functions to represent intentions from their definition in the ontology. The ontology is read using OWL API tools and the description of each intention is stored in a list.

Author

zeidk

Date

2013/01/01

Contact: zeid.kootbally@nist.gov

8.17.2 Constructor & Destructor Documentation

8.17.2.1 `intention.Intention.Intention ()`

Class constructor.

Instantiate the list of ordering constructs for an intention

Instantiate a list of forbidden ordering constructs (list of detrimental states)

8.17.3 Member Function Documentation

8.17.3.1 `void intention.Intention.builDetrimentalList (Exist _exist)`

Add an ordering construct of type Exist to the list of detrimental state relations.

Parameters

<code>_exist</code>	An ordering construct of type Exist
---------------------	-------------------------------------

8.17.3.2 `ArrayList<Exist> intention.Intention.getDetrimentalList ()`

Return the list of detrimental state relations for an intention.

Returns

[Intention.m_detrimental_list](#)

8.17.3.3 `OWLIndividual intention.Intention.getIndividual ()`

Return an intention of type OWLIndividual.

Returns

[Intention.m_individual](#)

8.17.3.4 String intention.Intention.getIntentionName ()

Return the name of the intention.

Returns

[Intention.m_intention_name](#)

8.17.3.5 double intention.Intention.getM_am1 ()

Return the value of AM_1 .

Returns

[Intention.m_am1](#)

8.17.3.6 double intention.Intention.getM_am2 ()

Return the value of AM_2 .

Returns

[Intention.m_am2](#)

8.17.3.7 double intention.Intention.getM_am3 ()

Return the value of AM_3 .

Returns

[Intention.m_am3](#)

8.17.3.8 double intention.Intention.getM_am4 ()

Return the value of AM_4 .

Returns

[Intention.m_am4](#)

8.17.3.9 double intention.Intention.getM_am5 ()

Return the value of AM_5 .

Returns

[Intention.m_am5](#)

8.17.3.10 static String intention.Intention.getM_built_kit () [static]

Return the kit chosen by the user in [gui.OptionFrame](#).

Returns

[Intention.m_built_kit](#)

8.17.3.11 int intention.Intention.getM_found_detrimental_SR ()

Return the number of detrimental state relations found for an intention.

Returns

[Intention.m_found_detrimental_SR](#)

8.17.3.12 int intention.Intention.getM_intention_number ()

Return the ID of an intention.

Returns

[Intention.m_intention_number](#)

8.17.3.13 double intention.Intention.getM_likelihood_observation ()

Return the likelihood of observation under a kit.

Returns

[Intention.m_likelihood_observation](#)

8.17.3.14 Map<Integer, Integer> intention.Intention.getM_map_SRirs ()

Return the HashMap that contains $SR_{i,r,s}$.

Returns

[Intention.m_SRirs](#)

8.17.3.15 double intention.Intention.getM_mm1 ()

Return the value of MM_1 .

Returns

[Intention.m_mm1](#)

8.17.3.16 double intention.Intention.getM_percentComplete_i_s ()

Return the percentage of completion.

Returns

[Intention.m_percentComplete_i_s](#)

8.17.3.17 double intention.Intention.getM_percentProductive_i_s ()

Return the percentage of productive states.

Returns

[Intention.m_percentProductive_i_s](#)

8.17.3.18 double intention.Intention.getM_probability_kit_observation ()

Return the probability of a kit given observations.

Returns

[Intention.m_probability_kit_observation](#)

8.17.3.19 static String intention.Intention.getM_selected_plan () [static]

Get the plan chosen by the user in [gui.OptionFrame](#).

Returns

[Intention.m_selected_plan](#)

8.17.3.20 int intention.Intention.getM_Si ()

Return the number of states (S) that have occurred since (and including) the first matched state relation in an intention (i)

Returns

[Intention.m_Si](#)

8.17.3.21 `int intention.Intention.getM_SR_i_r_s ()`

Return the value of $SR_{i,r,s}$ for the current intention.

Returns

[Intention.m_SR_i_r_s](#)

8.17.3.22 `int intention.Intention.getM_SR_i_s ()`

Return the number of matched state relations (SR) in an intention (i) as of the current state (s)

Returns

[Intention.m_SR_i_s](#)

8.17.3.23 `int intention.Intention.getM_SR_Total ()`

Return the number of state relations (SR) (whether matched or not) in an intention (i)

Returns

[Intention.m_SR_Total](#)

8.17.3.24 `int intention.Intention.getNumberStateRelation ()`

Return the number of state relations that constitute an intention.

Returns

[Intention.m_number_state_relation](#)

8.17.3.25 `OrderedList intention.Intention.getOrderedList ()`

Return the ordering construct OrderedList for an intention.

Returns

[Intention.m_orderedlist](#)

8.17.3.26 `void intention.Intention.setDetrimentalList (ArrayList< Exist > detrimental_list_)`

Set the list of detrimental state relations.

Parameters

<i>detrimental_ list_</i>	Value to set to Intention.m_detrimental_list
-------------------------------	--

8.17.3.27 void intention.Intention.setExist (Exist *exist_*)

Set the instance of the Java Class Exist to an intention.

Parameters

<i>exist_</i>	Value to set to Intention.m_exist
---------------	---

8.17.3.28 void intention.Intention.setIndividual (OWLIndividual *individual_*)

Set the intention as an OWL Individual.

Parameters

<i>individual_</i>	Value to set to Intention.m_individual
--------------------	--

8.17.3.29 void intention.Intention.setIntentionName (String *intention_name_*)

Set the name of the intention.

Parameters

<i>intention_ name_</i>	Value to set to Intention.m_intention_name
-----------------------------	--

8.17.3.30 void intention.Intention.setM_am1 (double *am1_*)

Set the value for AM_1 .

Parameters

<i>am1_</i>	Value to set to Intention.m_am1
-------------	---

8.17.3.31 void intention.Intention.setM_am2 (double *am2_*)

Set the value for AM_2 .

Parameters

<i>am2_</i>	Value to set to Intention.m_am2
-------------	---

8.17.3.32 void intention.Intention.setM_am3 (double *am3_*)

Set the value for AM_3 .

Parameters

<i>am3_</i>	Value to set to AM_3
-------------	------------------------

8.17.3.33 void intention.Intention.setM_am4 (double *am4_*)

Set the value for AM_4 .

Parameters

<i>am4_</i>	Value to set to AM_4
-------------	------------------------

8.17.3.34 void intention.Intention.setM_am5 (double *am5_*)

Set the value for AM_5 .

Parameters

<i>am5_</i>	Value to set to AM_5
-------------	------------------------

8.17.3.35 static void intention.Intention.setM_built_kit (String *built_kit_*) [static]

Set the kit the user chose to build in [gui.OptionFrame](#).

Parameters

<i>built_kit_</i>	Value to set to Intention.m_built_kit
-------------------	---

8.17.3.36 void intention.Intention.setM_found_detrimental_SR (int *found_detrimental_SR_*)

Set the number of detrimental state relations.

Parameters

<i>found_ - detrimental_ - SR_</i>	Value to set to Intention.m_found_detrimental_SR
--	--

8.17.3.37 void intention.Intention.setM_intention_number (int *intention_number_*)

Set an ID to an intention.

Parameters

<i>intention_number_</i>	Value to set to Intention.m_intention_number
--------------------------	--

8.17.3.38 void intention.Intention.setM_likelihood_observation (double *likelihood_observation_*)

Set the likelihood of observation under a kit.

Parameters

<i>likelihood_observation_</i>	Value set to Intention.m_likelihood_observation
--------------------------------	---

8.17.3.39 void intention.Intention.setM_map_SRirs (Map< Integer, Integer > *SRirs_*)

Set a HashMap to m_SRirs.

Parameters

<i>SRirs_</i>	Value to set to Intention.m_SRirs
---------------	---

8.17.3.40 void intention.Intention.setM_mm1 (double *mm1_*)

Set the value for MM_1 .

Parameters

<i>mm1_</i>	Value to set to MM_1
-------------	------------------------

8.17.3.41 void intention.Intention.setM_percentComplete_i_s (double *percentComplete_i_s_*)

Set the percentage of completion.

Parameters

<i>percentComplete_i_s_</i>	Value to set to Intention.m_percentComplete_i_s
-----------------------------	---

8.17.3.42 `void intention.Intention.setM_percentProductive_i_s (double percentProductive_i_s)`

Set the percentage of productive states.

Parameters

<i>percentProductive_i_s</i>	Value to set to Intention.m_percentProductive_i_s
------------------------------	---

8.17.3.43 `void intention.Intention.setM_probability_kit_observation (double probability_kit_observation)`

Set the probability of a kit given observations.

Parameters

<i>probability_kit_observation</i>	Value to set to Intention.m_probability_kit_observation
------------------------------------	---

8.17.3.44 `static void intention.Intention.setM_selected_plan (String selected_plan)`
[static]

Set the value of *m_selected_plan* from the plan selected by the user.

Parameters

<i>selected_plan</i>	Value to set to Intention.m_selected_plan
----------------------	---

8.17.3.45 `void intention.Intention.setM_Si (int Si)`

Set the number of states (*S*) that have occurred since (and including) the first matched state relation in an intention (*i*)

Parameters

<i>Si</i>	The value to set to Intention.m_Si
-----------	--

8.17.3.46 `void intention.Intention.setM_SR_i_r_s (int SR_i_r_s)`

Set the value of $SR_{i,r,s}$ for the current intention.

Parameters

<code>SR_i_r_s_</code>	Value to set to Intention.m_SR_i_r_s
------------------------	--

8.17.3.47 `void intention.Intention.setM_SR_i_s (int SR_i_s_)`

Set the number of matched state relations (*SR*) in an intention (*i*) as of the current state (*s*)

Parameters

<code>SR_i_s_</code>	The value for Intention.m_SR_i_s
----------------------	--

8.17.3.48 `void intention.Intention.setM_SR_Total (int SR_Total_)`

Set the number of state relations (*SR*) (whether matched or not) in an intention (*i*)

Parameters

<code>SR_Total_</code>	The value to set to Intention.m_SR_Total
------------------------	--

8.17.3.49 `void intention.Intention.setNumberStateRelation (int number_state_relation_)`

Set the number of state relations to an intention.

Parameters

<code>number_ - state_ - relation_</code>	Value to set to Intention.m_number_state_relation
---	---

8.17.3.50 `void intention.Intention.setOrderedList (OrderedList orderedlist_)`

Set the instance of Ordered List to the intention.

Parameters

<code>orderedlist_</code>	Value to set to Intention.m_orderedlist
---------------------------	---

8.17.4 Member Data Documentation

8.17.4.1 `double intention.Intention.m_am1` [private]

AM_1 : Number of observed state relations that are true in an intention (compared to other intentions).

The formula for this additive metric for intention i in state s is:

$$AM_{1,i,s} = \frac{SR_{i,s}}{SR_{all,s}} = \frac{SR_{i,s}}{\sum_{i=1}^p SR_{i,s}}$$

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).
- $SR_{all,s}$: The number of matched state relations (SR) in all possible intentions as of the current state (s).

This formula represents the ratio of true states that are in intention i to the sum of all of the true states in all of intentions of interest.

The variable p represents the number of intentions of interest. It is evaluated for every intention of interest at every state.

8.17.4.2 double intention.Intention.m_am2 [private]

AM₂:Percentage of an intention that is complete.

The formula for the percentage complete for intention i in state s is:

$$PercentComplete_{i,s} = \frac{SR_{i,s}}{SR_{i,total}}$$

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).
- $SR_{i,total}$: The number of state relations (SR) (whether matched or not) in an intention (i).

We then normalize this for all intentions of interest to find the additive metric 2 for intention i in state s .

$$AM_{2,i,s} = \frac{PercentComplete_{i,s}}{\sum_{i=1}^p PercentComplete_{i,s}}$$

8.17.4.3 double intention.Intention.m_am3 [private]

AM₃:Number of productive states that have occurred since the first productive state relation in an intention.

The formula for the percentage complete for intention i in state s is:

$$PercentComplete_{i,s} = \frac{SR_{i,s}}{S_i}$$

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).

- S_i : The number of states (S) that have occurred since (and including) the first matched state relation in an intention (i).

We then normalize this for all intentions by determining additive metric 3 for intention i in state s .

$$AM_{3,i,s} = \frac{PercentProductive_{i,s}}{\sum_{i=1}^p PercentProductive_i}$$

8.17.4.4 double intention.Intention.m_am4 [private]

AM_4 : Number of productive states that have occurred (recently) in the past r states.

The formula for AM_4 is:

$$AM_{4,i,s} = \frac{SR_{i,r,s}}{\sum_{i=1}^p SR_{i,r,s}}$$

- $SR_{i,r,s}$: The number of matched state relations (SR) in the past r states in an intention (i) as of the current state (s).

In other words, in the most recent (r) states, how many true state relations for an intention exist?

8.17.4.5 double intention.Intention.m_am5 [private]

AM_5 : Probability of an intention (i) being recognized based on an observation (j).

- Suppose a kit is described by the number of parts it contains for each type.
That is, $kit_i = (n_{i_A}, n_{i_B}, \dots, n_{i_Q})$ has n_{i_A} parts of type “A”, n_{i_B} parts of type “B”, ..., n_{i_Q} parts of type “Q”.
- Suppose an observation is described by the number of parts seen for each type.
That is, $observation_j = (x_{j_A}, x_{j_B}, \dots, x_{j_Q})$ has seen x_{j_A} parts of type “A”, x_{j_B} parts of type “B”, ..., x_{j_Q} parts of type “Q”.
- The likelihood L of observation j under kit i is given by the multivariate hypergeometric distribution:

$$L(observation_j | kit_i) = \frac{\prod_{p=1}^q n_{i_p} choose x_{j_p}}{\sum_{p=1}^q n_{i_p} choose \sum_{p=1}^q x_{j_p}}$$

The additive metric AM_5 for kit_i is the probability of kit_i given $observation_j$:

$$Probability(kit_i | observation_j) = \frac{L(observation_j | kit_i)}{\sum_{n=1}^N L(observation_j | kit_n)}$$

where n is the total number of kits that are likely to be built.

8.17.4.6 AnyOrder intention.Intention.m_anyorder [private]

Instance of the ordering construct AnyOrder.

8.17.4.7 String intention.Intention.m_built_kit [static, private]

The kit (intention) that the user chose to build

8.17.4.8 ArrayList<Exist> intention.Intention.m_detrimental_list [private]

List that contains detrimental states.

8.17.4.9 Exist intention.Intention.m_exist [private]

Instance of the ordering construct Exist.

8.17.4.10 int intention.Intention.m_found_detrimental_SR [private]

Number of detrimental state relations (*detrimentalSR*) that have occurred in an intention (*i*) as of the current state (*s*).

Detrimental state relations are state relations that are explicitly prohibited in an intention.

8.17.4.11 OWLIndividual intention.Intention.m_individual [private]

OWLIndividual (instance of the class [Intention](#))

8.17.4.12 String intention.Intention.m_intention_name [private]

Name of the intention.

8.17.4.13 int intention.Intention.m_intention_number [private]

An ID given to an intention, for instance, $a_4b_4c_2$ is associated to the ID #2. IDs are given to intentions in [ontology.Ontology.sortIntentionList\(\)](#)

8.17.4.14 HashMap<String, String> intention.Intention.m_intention_orderingConstruct_list**8.17.4.15 double intention.Intention.m_likelihood_observation** [private]

[Likelihood](#) for computing AM_5

8.17.4.16 double intention.Intention.m_mm1 [private]

MM_1 : The number of detrimental states.

It was chosen to be a multiplicative metric because the presence of detrimental states should play a larger role in the overall likelihood of an intention as compared to the additive metrics above.

The formula for MM_1 is:

$$MM_{1,i,s} = \frac{SR_{i,s} - detrimentalSR_{i,s}}{SR_{i,s}}$$

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).
- $detrimentalSR_{i,s}$: The number of detrimental states relations that have occurred in intention i as of the current state s .

Percent complete for intention i in state s

8.17.4.17 int intention.Intention.m_number_state_relation [private]

Number of state relations that consist an intention.

8.17.4.18 OrderedList intention.Intention.m_orderedlist [private]

Instance of the ordering construct OrderedList.

8.17.4.19 double intention.Intention.m_percentComplete_i_s [private]

Percent of productive states for intention i as of state s

8.17.4.20 double intention.Intention.m_percentProductive_i_s [private]

The number of matched state relations (SR) in an intention (i) as of the current state (s)

8.17.4.21 double intention.Intention.m_probability_kit_observation [private]

Probability for computing AM_5

8.17.4.22 String intention.Intention.m_selected_plan [static, private]

The plan the user chose for a given kit (intention)

8.17.4.23 `int intention.Intention.m_Si` [private]

Number of states (S) that have occurred since (and including) the first matched state relation in an intention (i)

8.17.4.24 `int intention.Intention.m_SR_i_r_s` [private]

8.17.4.25 `int intention.Intention.m_SR_i_s` [private]

Number of matched state relations (SR) in the past r states in an intention (i) as of the current state (s).

In other words, in the most recent (r) states, how many true state relations for an intention exist?

8.17.4.26 `int intention.Intention.m_SR_Total` [private]

Number of state relations (SR) (whether matched or not) in an intention (i)

8.17.4.27 `Map<Integer, Integer> intention.Intention.m_SRirs = new HashMap<Integer, Integer>()` [private]

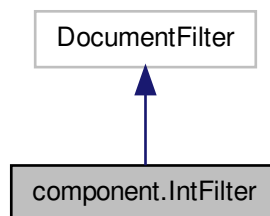
A HashMap where the key is the state and the value is the number of state relations found in the current state

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

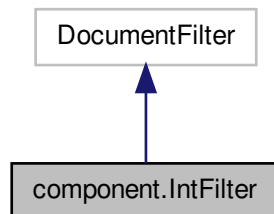
- `src/intention/Intention.java`

8.18 component.IntFilter Class Reference

Inheritance diagram for component.IntFilter:



Collaboration diagram for component.IntFilter:



Public Member Functions

- void [insertString](#) (FilterBypass fb, int offset, String string, AttributeSet attr) throws BadLocationException
- void [replace](#) (FilterBypass fb, int offset, int length, String text, AttributeSet attrs) throws BadLocationException
- void [remove](#) (FilterBypass fb, int offset, int length) throws BadLocationException

Private Member Functions

- boolean [test](#) (String text)

8.18.1 Member Function Documentation

8.18.1.1 void component.IntFilter.insertString (FilterBypass *fb*, int *offset*, String *string*, AttributeSet *attr*) throws BadLocationException

8.18.1.2 void component.IntFilter.remove (FilterBypass *fb*, int *offset*, int *length*) throws BadLocationException

8.18.1.3 void component.IntFilter.replace (FilterBypass *fb*, int *offset*, int *length*, String *text*, AttributeSet *attrs*) throws BadLocationException

8.18.1.4 boolean component.IntFilter.test (String *text*) [private]

Return true if the text entered contains an Integer, false otherwise

Parameters

<i>text</i>	The text that is checked
-------------	--------------------------

Returns

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

- [src/component/IntFilter.java](#)

8.19 main.Launcher Class Reference

Main class of the tool.

Public Member Functions

- [Launcher](#) ()
Constructor.

Static Public Member Functions

- static void [main](#) (String[] args) throws OWLException, InterruptedException, InvocationTargetException, ClassNotFoundException, InstantiationException, IllegalAccessException, IOException
Main of the project. The main file allows:
- static void [enabler](#) ()
Enable some gui components from the class gui.Mainframe.

8.19.1 Detailed Description

Main class of the tool.

Author

[Zeid Kootbally](#) zeid.kootbally@nist.gov

Precondition

Make sure the kits directory is present in the same directory as this tool
Make sure kittingClasses.owl, kittingInstances_ir.owl, and soap.owl are in the same directory as this tool

8.19.2 Constructor & Destructor Documentation

8.19.2.1 main.Launcher.Launcher ()

Constructor.

8.19.3 Member Function Documentation

8.19.3.1 static void main.Launcher.enabler () [static]

Enable some gui components from the class gui.Mainframe.

8.19.3.2 static void main.Launcher.main (String[] args) throws OWLException, InterruptedException, InvocationTargetException, ClassNotFoundException, InstantiationException, IllegalAccessException, IOException [static]

Main of the project. The main file allows:

- The creation of an object for the class [Ontology](#)
- Set the OWLAPI manager
- Initialize all the array lists that are used in the project
- Load the ontology
- Set the OWLAPI reasoner
- Set the OWLAPI data factory
- Parse the ontology to retrieve information on each intention

Parameters

<i>args</i>	
-------------	--

Exceptions

<i>OWLException</i>	
<i>InterruptedException</i>	
<i>InvocationTargetException</i>	
<i>UnsupportedLookAndFeelException</i>	
<i>IllegalAccessException</i>	
<i>InstantiationException</i>	
<i>ClassNotFoundException</i>	
<i>IOException</i>	

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

- src/main/[Launcher.java](#)

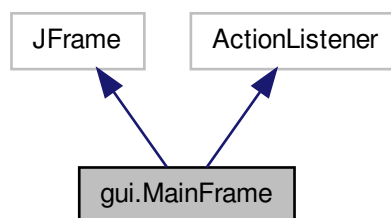
8.20 intention.Likelihood Class Reference

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

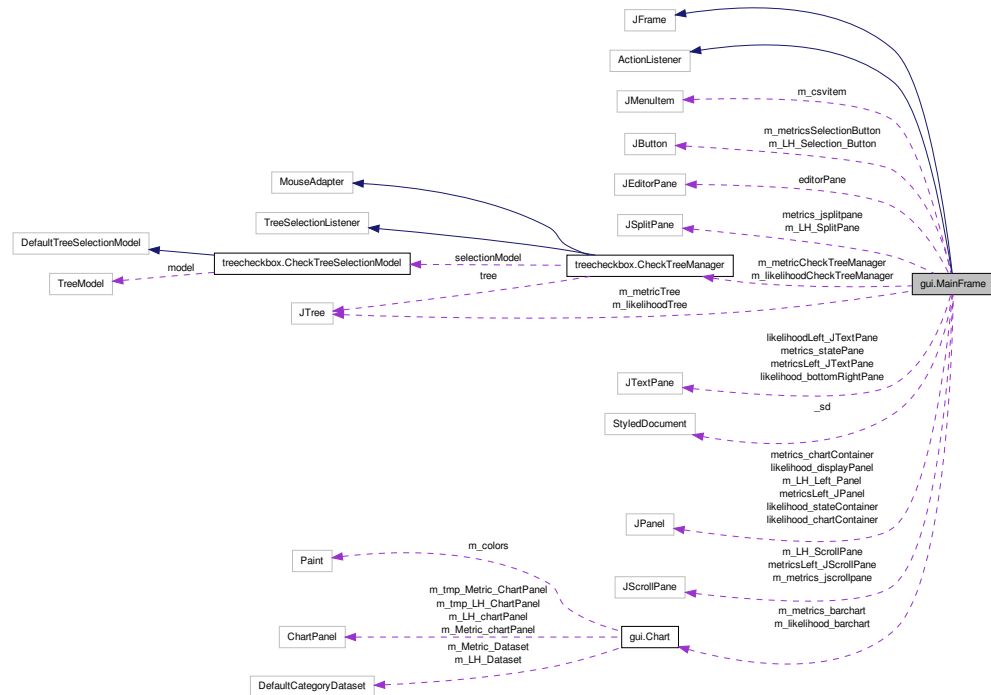
- `src/intention/Likelihood.java`

8.21 gui.MainFrame Class Reference

Inheritance diagram for gui.MainFrame:



Collaboration diagram for gui.MainFrame:



Classes

- class [DisplayMetrics](#)

Public Member Functions

- [MainFrame](#) (String s, JTree metricTree, JTree likelihoodTree)
- void [actionPerformed](#) (ActionEvent actionevent)
- void [saveToCSV](#) (String path)
Create the file file and write intention dataset.
- void [showErrorMessage](#) (String title, String message)

Static Public Member Functions

- static void [updateStateRelationPanel](#) (String s, Color c, JTextPane textpane)

Static Public Attributes

- static final String [EXIT_COMMAND](#) = "EXIT"
- static JTextPane [likelihood_bottomRightPane](#) = new JTextPane()
- static JTextPane [metrics_statePane](#) = new JTextPane()
- static JTextPane [likelihoodLeft_JTextPane](#) = new JTextPane()
- static JTextPane [metricsLeft_JTextPane](#) = new JTextPane()
- static JScrollPane [metricsLeft_JScrollPane](#) = new JScrollPane()
- static JPanel [metricsLeft_JPanel](#) = new JPanel()
- static JPanel [m_LH_Left_Panel](#) = new JPanel()
- static JPanel [metrics_chartContainer](#)
- static JButton [m_metricsSelectionButton](#)
- static JButton [m_LH_Selection_Button](#)
- static JTree [m_metricTree](#)
- static JTree [m_likelihoodTree](#)
- static JMenuItem [m_csvitem](#) = new JMenuItem("Export to CSV...", 112)

Package Attributes

- [CheckTreeManager m_metricCheckTreeManager](#)
- [CheckTreeManager m_likelihoodCheckTreeManager](#)

Private Member Functions

- JPanel [buildMetricPanel](#) (JTree metricTree)
- JPanel [buildLikelihoodPanel](#) (JTree likelihoodTree)
Build the main panel that contains components for the likelihoods.
- JComponent [createContent](#) (JTree metricTree, JTree likelihoodTree)
- void [lh_Button_ActionPerformed](#) (java.awt.event.ActionEvent evt)
- JMenuBar [createMenuBar](#) ()
- void [copyToClipboard](#) ()
- void [applyThemeToChart](#) ()
- void [exportToPDF](#) ()
- void [exportToCSV](#) ()
Export likelihoods result to a csv file.
- void [attemptExit](#) ()
- JPanel [createLikelihoodChartPanel](#) ()
Create the chart panel that will contain the likelihood chart.
- JPanel [createMetricsChartPanel](#) ()
Build the JPanel that will contain the chart displaying metrics values.

Private Attributes

- JPanel [likelihood_displayPanel](#)
- JPanel [likelihood_chartContainer](#)
- JPanel [likelihood_stateContainer](#)
- JEditorPane [editorPane](#)
- Chart [m_likelihood_barchart](#) = new [Chart\(\)](#)
- Chart [m_metrics_barchart](#) = new [Chart\(\)](#)

Static Private Attributes

- static final long [serialVersionUID](#) = 1384873058265918162L
- static StyledDocument [_sd](#)
- static JSplitPane [metrics_jsplitpane](#)
- static JSplitPane [m_LH_SplitPane](#)
- static JScrollPane [m_metrics_jscrollpane](#) = new JScrollPane()
- static JScrollPane [m_LH_ScrollPane](#) = new JScrollPane()

8.21.1 Constructor & Destructor Documentation

8.21.1.1 `gui.MainFrame.MainFrame (String s, JTree metricTree, JTree likelihoodTree)`

8.21.2 Member Function Documentation

8.21.2.1 `void gui.MainFrame.actionPerformed (ActionEvent actionevent)`

8.21.2.2 `void gui.MainFrame.applyThemeToChart () [private]`

8.21.2.3 `void gui.MainFrame.attemptExit () [private]`

8.21.2.4 `JPanel gui.MainFrame.buildLikelihoodPanel (JTree likelihoodTree) [private]`

Build the main panel that contains components for the likelihoods.

Parameters

<i>likelihoodTree</i>	The JTree that displays intentions
-----------------------	------------------------------------

Returns

8.21.2.5 `JPanel gui.MainFrame.buildMetricPanel (JTree metricTree) [private]`

8.21.2.6 `void gui.MainFrame.copyToClipboard () [private]`

8.21.2.7 `JComponent gui.MainFrame.createContent (JTree metricTree, JTree likelihoodTree)`
[private]

8.21.2.8 `JPanel gui.MainFrame.createLikelihoodChartPanel ()` [private]

Create the chart panel that will contain the likelihood chart.

Returns

The likelihood panel.

8.21.2.9 `JMenuBar gui.MainFrame.createMenuBar ()` [private]

8.21.2.10 `JPanel gui.MainFrame.createMetricsChartPanel ()` [private]

Build the JPanel that will contain the chart displaying metrics values.

Returns

The JPanel that will contain the chart displaying metrics values.

8.21.2.11 `void gui.MainFrame.exportToCSV ()` [private]

Export likelihoods result to a csv file.

8.21.2.12 `void gui.MainFrame.exportToPDF ()` [private]

8.21.2.13 `void gui.MainFrame.lh.Button_ActionPerformed (java.awt.event.ActionEvent evt)`
[private]

8.21.2.14 `void gui.MainFrame.saveToCSV (String path)`

Create the file *file* and write intention dataset.

Parameters

<i>file</i>	
-------------	--

8.21.2.15 `void gui.MainFrame.showErrorMessage (String title, String message)`

8.21.2.16 `static void gui.MainFrame.updateStateRelationPanel (String s, Color c, JTextPane textpane)` [static]

8.21.3 Member Data Documentation

- 8.21.3.1 `StyledDocument gui.MainFrame._sd` [static, private]
- 8.21.3.2 `JEditorPane gui.MainFrame.editorPane` [private]
- 8.21.3.3 `final String gui.MainFrame.EXIT_COMMAND = "EXIT"` [static]
- 8.21.3.4 `JTextPane gui.MainFrame.likelihood_bottomRightPane = new JTextPane()`
[static]
- 8.21.3.5 `JPanel gui.MainFrame.likelihood_chartContainer` [private]
- 8.21.3.6 `JPanel gui.MainFrame.likelihood_displayPanel` [private]
- 8.21.3.7 `JPanel gui.MainFrame.likelihood_stateContainer` [private]
- 8.21.3.8 `JTextPane gui.MainFrame.likelihoodLeft_JTextPane = new JTextPane()`
[static]
- 8.21.3.9 `JMenuItem gui.MainFrame.m_csvitem = new JMenuItem("Export to CSV...", 112)`
[static]
- 8.21.3.10 `JPanel gui.MainFrame.m_LH_Left_Panel = new JPanel()` [static]
- 8.21.3.11 `JScrollPane gui.MainFrame.m_LH_ScrollPane = new JScrollPane()`
[static, private]
- 8.21.3.12 `JButton gui.MainFrame.m_LH_Selection_Button` [static]
- 8.21.3.13 `JSplitPane gui.MainFrame.m_LH_SplitPane` [static, private]
- 8.21.3.14 `Chart gui.MainFrame.m_likelihood_barchart = new Chart()` [private]
- 8.21.3.15 `CheckTreeManager gui.MainFrame.m_likelihoodCheckTreeManager`
[package]
- 8.21.3.16 `JTree gui.MainFrame.m_likelihoodTree` [static]
- 8.21.3.17 `CheckTreeManager gui.MainFrame.m_metricCheckTreeManager`
[package]
- 8.21.3.18 `Chart gui.MainFrame.m_metrics_barchart = new Chart()` [private]
- 8.21.3.19 `JScrollPane gui.MainFrame.m_metrics_jscrollpane = new JScrollPane()`
[static, private]
- 8.21.3.20 `JButton gui.MainFrame.m_metricsSelectionButton` [static]

- 8.21.3.21 `JTree gui.MainFrame.m_metricTree` [static]
- 8.21.3.22 `JPanel gui.MainFrame.metrics_chartContainer` [static]
- 8.21.3.23 `JSplitPane gui.MainFrame.metrics_jsplitpane` [static, private]
- 8.21.3.24 `JTextPane gui.MainFrame.metrics_statePane = new JTextPane()` [static]
- 8.21.3.25 `JPanel gui.MainFrame.metricsLeft_JPanel = new JPanel()` [static]
- 8.21.3.26 `JScrollPane gui.MainFrame.metricsLeft_JScrollPane = new JScrollPane()`
[static]
- 8.21.3.27 `JTextPane gui.MainFrame.metricsLeft_JTextPane = new JTextPane()`
[static]
- 8.21.3.28 `final long gui.MainFrame.serialVersionUID = 1384873058265918162L`
[static, private]

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

- `src/gui/MainFrame.java`

8.22 intention.Metric Class Reference

Definition of additive and multiplicative metrics.

Static Public Member Functions

- static int `get_AM1_Weight ()`
- static int `get_AM2_Weight ()`
- static int `get_AM3_Weight ()`
- static int `get_AM4_Weight ()`
- static int `get_AM5_Weight ()`
- static void `set_AM1_Weight (int weight)`
- static void `set_AM2_Weight (int weight)`
- static void `set_AM3_Weight (int weight)`
- static void `set_AM4_Weight (int weight)`
- static void `set_AM5_Weight (int weight)`

Private Member Functions

- int `get_MM1_Weight ()`
- void `set_MM1_Weight (int weight)`

Static Private Attributes

- static int [m_AM1_weight](#)
- static int [m_AM2_weight](#)
- static int [m_AM3_weight](#)
- static int [m_AM4_weight](#)
- static int [m_AM5_weight](#)
- static int [m_MM1_weight](#)

8.22.1 Detailed Description

Definition of additive and multiplicative metrics.

This class consists of additive and multiplicative metrics definitions

Author

zeidk

Date

2013/01/01

Contact: zeid.kootbally@nist.gov

8.22.2 Member Function Documentation

8.22.2.1 static int intention.Metric.get_AM1_Weight () [static]

Get the weight for AM1.

Returns

The weight for AM1.

8.22.2.2 static int intention.Metric.get_AM2_Weight () [static]

Get the weight for AM2.

Returns

The weight for AM2.

8.22.2.3 static int intention.Metric.get_AM3_Weight () [static]

Get the weight for AM3.

Returns

The weight for AM3.

8.22.2.4 `static int intention.Metric.get_AM4_Weight () [static]`

Get the weight for AM4.

Returns

The weight for AM4.

8.22.2.5 `static int intention.Metric.get_AM5_Weight () [static]`

Get the weight for AM5.

Returns

The weight for AM5.

8.22.2.6 `int intention.Metric.get_MM1_Weight () [private]`

Get the weight for MM1.

Returns

The weight for MM1.

8.22.2.7 `static void intention.Metric.set_AM1_Weight (int weight) [static]`

Set the weight for AM1.

Parameters

<i>weight</i>	Value set to the weight for AM1.
---------------	----------------------------------

8.22.2.8 `static void intention.Metric.set_AM2_Weight (int weight) [static]`

Set the weight for AM2.

Parameters

<i>weight</i>	Value set to the weight for AM2.
---------------	----------------------------------

8.22.2.9 `static void intention.Metric.set_AM3_Weight (int weight) [static]`

Set the weight for AM3.

Parameters

<i>weight</i>	Value set to the weight for AM3.
---------------	----------------------------------

8.22.2.10 `static void intention.Metric.set_AM4_Weight (int weight) [static]`

Set the weight for AM4.

Parameters

<i>weight</i>	Value set to the weight for AM4.
---------------	----------------------------------

8.22.2.11 `static void intention.Metric.set_AM5_Weight (int weight) [static]`

Set the weight for AM5.

Parameters

<i>weight</i>	Value set to the weight for AM5.
---------------	----------------------------------

8.22.2.12 `void intention.Metric.set_MM1_Weight (int weight) [private]`

Set the weight for MM1.

Parameters

<i>weight</i>	Value set to the weight for MM1.
---------------	----------------------------------

8.22.3 Member Data Documentation

8.22.3.1 `int intention.Metric.m_AM1_weight [static, private]`

8.22.3.2 `int intention.Metric.m_AM2_weight [static, private]`

8.22.3.3 `int intention.Metric.m_AM3_weight [static, private]`

8.22.3.4 `int intention.Metric.m_AM4_weight [static, private]`

8.22.3.5 `int intention.Metric.m_AM5_weight [static, private]`

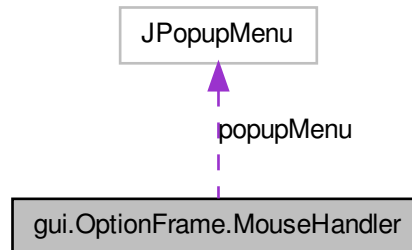
8.22.3.6 `int intention.Metric.m_MM1_weight [static, private]`

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

- [src/intention/Metric.java](#)

8.23 gui.OptionFrame.MouseHandler Class Reference

Collaboration diagram for gui.OptionFrame.MouseHandler:



Public Member Functions

- [MouseListener](#) (javax.swing.JPopupMenu popup)
- void [mousePressed](#) (java.awt.event.MouseEvent e)
- void [mouseReleased](#) (java.awt.event.MouseEvent e)

Private Attributes

- javax.swing.JPopupMenu [popupMenu](#)

8.23.1 Constructor & Destructor Documentation

8.23.1.1 `gui.OptionFrame.MouseHandler.MouseHandler (javax.swing.JPopupMenu popup)`

8.23.2 Member Function Documentation

8.23.2.1 `void gui.OptionFrame.MouseHandler.mousePressed (java.awt.event.MouseEvent e)`

8.23.2.2 `void gui.OptionFrame.MouseHandler.mouseReleased (java.awt.event.MouseEvent e)`

8.23.3 Member Data Documentation

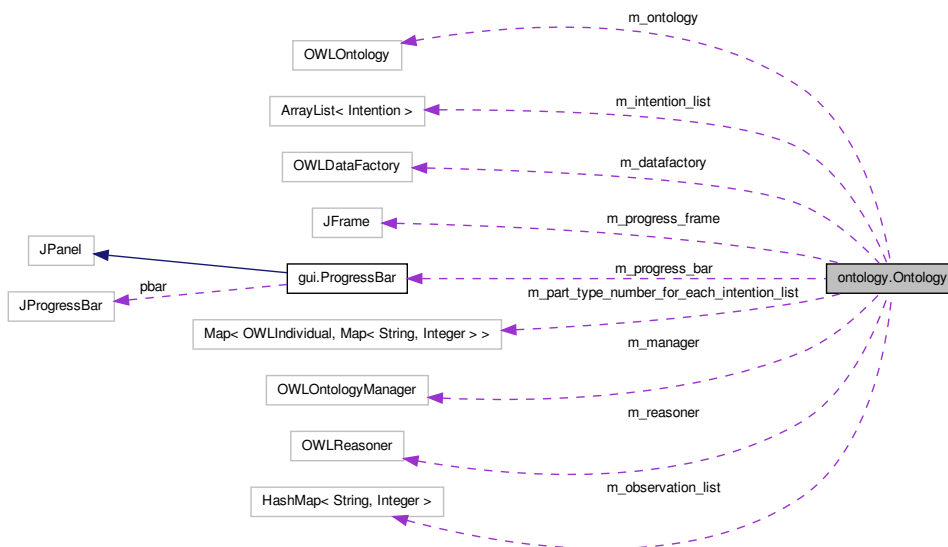
8.23.3.1 `javax.swing.JPopupMenu gui.OptionFrame.MouseHandler.popupMenu`
[private]

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

- [src/gui/OptionFrame.java](#)

8.24 ontology.Ontology Class Reference

Collaboration diagram for ontology.Ontology:



Public Member Functions

- [Ontology \(\)](#)
Constructor of the [Ontology](#) class.
- [String getOntologyPath \(\)](#)
Get the path of the ontology.
- [String getRootClass \(\)](#)
Get the class root from the ontology.
- [void setRootClass \(String rootClass_\)](#)
Set the class root from the ontology.
- [Set< OWLClassExpression > getIndividualClass \(OWLNamedIndividual individual\)](#)
- [OWLOntologyManager getManager \(\)](#)
Simple getter.
- [OWLOntology getOntology \(\)](#)
Simple getter.
- [String getPath \(\)](#)

Simple getter.

- NodeSet< OWLClass > [getSubclasses](#) (String myClassName)
- void [initializeList](#) ()
- void [loadFromFile](#) ()

Load the ontology from a file.

- void [loadOntologyFromPath](#) (String myPath) throws MalformedURLException, OWLException
- void [sortIntentionList](#) ()

Read m_intention_list and re-arrange the list using a number for each intention.

- void [parseIntention](#) (NodeSet< OWLClass > myClass) throws InterruptedException, InvocationTargetException, IOException
- void [showDialogBox](#) ()
- JTree [buildIntentionTree](#) ()
- JTree [buildIntentionMetricsTree](#) ()
- void [setDataFactory](#) ()
- void [setManager](#) ()
- void [setManager](#) (OWLOntologyManager manager)

Simple setter.

- void [setOntology](#) (OWLOntology ontology)

Simple setter.

- void [setPath](#) (String path)

Simple setter.

- void [setReasoner](#) (OWLOntology myOntology)

Static Public Member Functions

- static void [setInstanceFilePath](#) (String path_)

Set the path to the OWL instance file The path is retrieved from the field [OptionFrame.m_instance_txt_field](#).

- static String [cleanIRI](#) (Object entity)

Return the name of the entity without the IRI.

- static double [computeMetricAM1](#) (double _sr_i_s, double _sr_all)

Compute the additive metric AM_1 .

- static double [computeMetricAM2](#) (double _percentComplete_i_s, double _percent_complete_total)

Compute the additive metric AM_2 .

- static double [computeMetricAM3](#) (double percent_productive_i_s, double percent_productive_i)

Compute the additive metric AM_3 .

- static double [computeMetricAM4](#) (double sr_i_r_s, double sum_sr_i_r_s)

Compute the additive metric AM_4 .

- static double [computeMetricAM5](#) ([Intention](#) intention)

Compute the additive metric AM_5 .

- static double [computeMetricMM1](#) (int sr_i_s, int detrimental)

- static void [buildObservationList](#) ()
Build a list of the parts observed during kitting.
- static void [updateObservationList](#) (String part_type)
Update the observation list.
- static void [readObservationList](#) ()
Read the observation list m_observation_list.
- static void [chooseTest](#) ()
- static void [computeObservationLikelihood](#) (Intention intention_)
Compute the likelihood L of observation j under kit i.
- static void [computeProbabilityFromObservation](#) (Intention intention_)
- static int [compute_sum_observation](#) ()
- static int [compute_sum_part_type](#) (Intention intention_)
- static ArrayList< String > [removeDuplicates](#) (ArrayList< String > list_)
Remove duplicates in an ArrayList of String.
- static double [computePercentComplete](#) (double sr_i_s, double sr_i_total)
PercentComplete is the percentage complete for an intention in a state.
- static double [computePercentProductive](#) (double sr_i_s, double si)
The formula for the percent productive is used by AM₃ for an intention (i) as of state (s) is:
- static double [computeLikelihood](#) (Intention _intention, int _state)
The overall equation that is used to determine the likelihood of intentions is:

$$L_i = \left[\prod_{1 \leq j \leq m} MM_j \right] \times \left[\frac{\sum_{k=1}^n (AM_k \times W_{AM_k})}{\sum_{k=1}^n W_{AM_k}} \right] \times 100$$

Where:

- static void [computeMetricsInformation](#) (ArrayList< ArrayList< ArrayList< String >>> _states) throws InterruptedException, BadLocationException
Retrieve and compute the information required for metrics computation.
- static String [getIndividualClassString](#) (OWLNamedIndividual individual)
- static String [getIndividualClassString](#) (OWLIndividual individual)
- static char [getSeparator](#) ()
Simple getter.
- static void [matchDetrimentalStateRelationToIntention](#) (ArrayList _state_relation)
Check for each intention if _state_relation is a detrimental state relation.
- static void [updateMainFrame](#) (ArrayList _state_relation, int num)
Display the current state relation in MainFrame.
- static void [matchStateRelationToIntention](#) (ArrayList _state_relation, int num) throws BadLocationException
Check for each intention if the state relation _state_relation matches any of the intention state relations.
- static void [computeProceduresForAM5](#) ()

Public Attributes

- String `m_s_rootClass` = "Intention"

Static Public Attributes

- static ArrayList< `Intention` > `m_intention_list`
- static OWLOntology `m_ontology`
- static String `m_kitToBuild`
- static String `m_planToBuild`
- static String `m_s_subClass` = "Kitting"

Static Package Functions

- static int `compute_SR_i_r_s` (Map< Integer, Integer > `_map`, Integer `_r`, Integer `_id_current_state`)
Compute the number of true state relations (SR) in the past r states in intention i as of the current state s .
- static BigInteger `choose` (final int `N`, final int `K`)
Returns a double representation of the Binomial Coefficient, " N choose K ", the number of K -element subsets that can be selected from an N -element set.
- static double `roundTwoDecimals` (double `d`)

Private Member Functions

- void `readIntentionList` ()
Read the intentions previously stored.
- void `searchList` (String `individual`)

Static Private Member Functions

- static void `buildIntentionList` (NodeSet< OWLClass > `setOfSubclasses_`)
Parse the ontology and retrieve all the elements associated to each intention.
- static ArrayList< ArrayList< ArrayList< String > > > `buildStates` (String `plan_path`)
- static int `cleanDataPropertyInteger` (String `s`)
- static void `readForEachIntentionTheNumberOfPartsForEachType` ()
Read the map Map < OWLIndividual, Map< String, Integer> >
- static void `getForEachIntentionTheNumberOfPartsForEachType` ()
Retrieve the number of parts each intention contains for each type.
- static void `updateForEachIntentionTheNumberOfPartsForEachType` ()
Update `m_part_type_number_for_each_intention_list` with missing part types.
- static int `compute_SR_all_s` ()
- static String `getReferenceObjectClass` (OWLNamedIndividual `individual`)

- static String [getStateRelation](#) (OWLNamedIndividual individual)
- static String [getTargetObjectClass](#) (OWLNamedIndividual individual)
- static boolean [hasProperty](#) (OWLOntologyManager man, OWLReasoner reasoner, OWLClass cls, OWLObjectPropertyExpression prop)
- static ArrayList< String > [input](#) (String state_relation, String target_object, String target_class, String reference_object, String reference_class)
- static Color [getPartColor](#) (String _string)
- static String [getPartType](#) (String _string)
- static String [getStringHead](#) (String _string)
- static String [getStringTail](#) (String _string)
- static void [printProperties](#) (OWLOntologyManager man, OWLOntology ont, OWLReasoner reasoner, OWLClass cls)

Print the properties that an instance has to have.

Private Attributes

- OWLOntologyManager [m_manager](#)
- JFrame [m_progress_frame](#)
- [ProgressBar](#) [m_progress_bar](#)
- String [m_path](#)

Static Private Attributes

- static OWLDataFactory [m_datafactory](#)
- static OWLReasoner [m_reasoner](#)
- static String [m_hasIntention_OrderingConstruct](#) = "#hasIntention_OrderingConstruct"
- static String [m_hasOrderingConstruct_Predicate](#) = "#hasOrderingConstruct_Predicate"
- static String [m_hasOrderingConstruct_OrderingConstruct](#) = "#hasOrderingConstruct_OrderingConstruct"
- static String [m_hasOrderingConstruct_Position](#) = "#hasOrderingConstruct_Position"
- static String [m_hasPredicate_TargetObject](#) = "#hasPredicate_TargetObject"
- static String [m_hasIntention_Name](#) = "#hasIntention_Name"
- static String [m_hasPredicate_ReferenceObject](#) = "#hasPredicate_ReferenceObject"
- static String [m_hasCount_Occurrence](#) = "#hasCount_Occurrence"
- static HashMap< String, Integer > [m_observation_list](#) = new HashMap<String, Integer>()
- static Map< OWLIndividual, Map< String, Integer > > [m_part_type_number_for_each_intention_list](#) = new HashMap<OWLIndividual, Map<String, Integer>>()
- static final String [m_ontology_IRI](#) = "http://www.semanticweb.org/ontologies/2013/0/soap.owl"
- static String [m_s_ontopath](#)
- static final char [m_SEPARATOR](#) = '#'

8.24.1 Constructor & Destructor Documentation

8.24.1.1 ontology.Ontology.Ontology ()

Constructor of the [Ontology](#) class.

8.24.2 Member Function Documentation

8.24.2.1 static void ontology.Ontology.buildIntentionList (NodeSet< OWLClass > *setOfSubclasses_*) [static, private]

Parse the ontology and retrieve all the elements associated to each intention.

Each element found is stored in an attribute of their corresponding Java class. For instance, when the ordering construct "Exist" is found in the definition of an intention, an instance of the Java class [Exist](#) is created and information on [Exist](#) for the current intention is stored in the instance.

The steps to read an intention are as follows:

- Parse the set of subclasses *setOfSubclasses_* that consists of different domains (Kitting, Assembly, ...).
- In *setOfSubclasses_*, search for the *subclass* defined by the member variable [Ontology.m_s_subClass](#).
- Get each OWL individual of *subclass* with the OWL API function `getInstances(OWLClassExpression ce, boolean direct)`.
- For each individual:
 - Create a Java instance of the class `Intention`.
 - Set this instance's individual.
See also
[Intention.setIndividual\(OWLIndividual\);](#)
 - Set the percentage for this instance to 0.
See also
[Intention.setPercentage\(double\)](#)
 - Set the number of state relations for this instance to 0.
See also
[Intention.setNumberStateRelation\(int\)](#)

Some of the OWL API functions used are:

- `getFlattened()`: A convenience method that gets all of the entities contained in the Nodes in this `NodeSet`.
- `getInstances(OWLClassExpression ce, boolean direct)`: Gets the individuals which are instances of the specified class expression. The individuals are returned as a `NodeSet`.

- getObjectPropertyValues(OWLNamedIndividual ind, OWLObjectPropertyExpression pe): Gets the object property values for the specified individual and object property expression. The individuals are returned as a NodeSet.

Parameters

<i>setOfSubclasses</i>	A set of subclasses built from the root class.
------------------------	--

8.24.2.2 JTree ontology.Ontology.buildIntentionMetricsTree ()

8.24.2.3 JTree ontology.Ontology.buildIntentionTree ()

8.24.2.4 static void ontology.Ontology.buildObservationList () [static]

Build a list of the parts observed during kitting.

The list is an ArrayList that contains HashMaps. Each HashMap represents the type of part that exists for kitting.

Returns

8.24.2.5 static ArrayList<ArrayList<ArrayList<String>>> ontology.Ontology.buildStates (String *plan_path*) [static, private]

This function builds all the states for a given intention.

Parameters

<i>kit</i>	The kit to build
------------	------------------

Returns

An ArrayList that contains all the states for the given kit

8.24.2.6 static BigInteger ontology.Ontology.choose (final int *N*, final int *K*) [static, package]

Returns a double representation of the Binomial Coefficient, "N choose K", the number of K-element subsets that can be selected from an N-element set.

Parameters

<i>N</i>	N-element set
<i>K</i>	K-element subset

Returns

Double representation of the Binomial Coefficient

8.24.2.7 `static void ontology.Ontology.chooseTest () [static]`

8.24.2.8 `static int ontology.Ontology.cleanDataPropertyInteger (String s) [static, private]`

8.24.2.9 `static String ontology.Ontology.cleanIRI (Object entity) [static]`

Return the name of the entity without the IRI.

For example, if *entity* = [<http://www.semanticweb.org/ontologies/2013/0/soap.owl#Kitting>], this function returns *Kitting*. This function operates as follows: - Identify the index of the separator *SEPARATOR* - Keep only what is after the SEPARATOR - Remove characters that are not alphanumeric

Parameters

<i>entity</i>	Entity to be trimmed
---------------	----------------------

Returns

The name of the entity without the IRI

8.24.2.10 `static int ontology.Ontology.compute_SR_all.s () [static, private]`

8.24.2.11 `static int ontology.Ontology.compute_SR_i.r.s (Map< Integer, Integer > .map, Integer _r, Integer _id_current_state) [static, package]`

Compute the number of true state relations (SR) in the past *r* states in intention *i* as of the current state *s*.

Parameters

<i>_map</i>	Map<Integer, Integer> for an intention <i>i</i> where: <ul style="list-style-type: none"> • Key: The id of a state (0 for the first state, 1 for the second state, ...). • Value: The number of state relations found for the intention <i>i</i> in the Key state.
<i>_r</i>	The last <i>r</i> states
<i>_id_current_state</i>	The current state

Returns

The value of SR_i_r_s

8.24.2.12 `static int ontology.Ontology.compute_sum_observation () [static]`

8.24.2.13 `static int ontology.Ontology.compute_sum_part_type (Intention intention_) [static]`

8.24.2.14 `static double ontology.Ontology.computeLikelihood (Intention _intention, int _state) [static]`

The overall equation that is used to determine the likelihood of intentions is:

$$L_i = \left[\prod_{1 \leq j \leq m} MM_j \right] \times \left[\frac{\sum_{k=1}^n (AM_k \times W_{AM_k})}{\sum_{k=1}^n W_{AM_k}} \right] \times 100$$

Where:

- L_i is the likelihood of an intention i
- MM_j is the multiplication metric j
- AM_k is the additive metric k
- W_{AM_k} is the weight of the additive metric k
- m is the total number of multiplicative metrics
- n is the total number of additive metrics

All metrics (whether multiplicative or additive) must contain a value between 0 and 1, where 0 is the lowest value and 1 is the highest value.

Additive metrics (AM_k) along with their associated weights, are added together and then divided by the sum of all their weights.

Weights are associated with the additive metrics to show the relative importance of one metric over another. These weights can contain any value greater than 0.

Multiplicative metrics are significant enough in importance that their value is multiplied in the likelihood equation to carry a heavier effect on the overall likelihood.

Parameters

<code>_intention</code>	The intention for which the likelihood will be computed
<code>_state</code>	State

8.24.2.15 `static double ontology.Ontology.computeMetricAM1 (double _sr_i_s, double _sr_all) [static]`

Compute the additive metric AM_1 .

AM_1 : Number of observed state relations that are true in an intention (compared to other intentions).

The formula for this additive metric for intention i in state s is:

$$AM_{1,i,s} = \frac{SR_{i,s}}{SR_{all,s}} = \frac{SR_{i,s}}{\sum_{i=1}^p SR_{i,s}}$$

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).
- $SR_{all,s}$: The number of matched state relations (SR) in all possible intentions as of the current state (s).

This formula represents the ratio of true states that are in intention i to the sum of all of the true states in all of intentions of interest.

The variable p represents the number of intentions of interest. It is evaluated for every intention of interest at every state.

Parameters

<code>_sr_i_s</code>	The number of matched state relations (SR) in an intention (i) as of the current state (s).
<code>_sr_all</code>	The number of matched state relations (SR) in all possible intentions as of the current state (s).

8.24.2.16 `static double ontology.Ontology.computeMetricAM2 (double _percentComplete_i_s, double _percent_complete_total) [static]`

Compute the additive metric AM_2 .

AM_2 : Percentage of an intention that is complete.

The formula for the percentage complete for intention i in state s is:

$$PercentComplete_{i,s} = \frac{SR_{i,s}}{SR_{i,total}}$$

We then normalize this for all intentions of interest to find the additive metric 2 for intention i in state s .

$$AM_{2,i,s} = \frac{PercentComplete_{i,s}}{\sum_{i=1}^p PercentComplete_{i,s}}$$

Parameters

<code>_percentComplete_i_s</code>	Percent complete for intention (i) in state (s).
<code>_percent_complete_total</code>	Sum of percent complete for each intention in state (s).

8.24.2.17 `static double ontology.Ontology.computeMetricAM3 (double percent_productive_i_s,
double percent_productive_i) [static]`

Compute the additive metric AM_3 .

AM_3 : Number of productive states that have occurred since the first productive state relation in an intention.

The formula for the percentage complete for intention i in state s is:

$$PercentComplete_{i,s} = \frac{SR_{i,s}}{S_i}$$

We then normalize this for all intentions by determining additive metric 3 for intention i in state s .

$$AM_{3,i,s} = \frac{PercentProductive_{i,s}}{\sum_{i=1}^p PercentProductive_i}$$

8.24.2.18 `static double ontology.Ontology.computeMetricAM4 (double sr_i_r_s, double
sum_sr_i_r_s) [static]`

Compute the additive metric AM_4 .

AM_4 : Number of productive states that have occurred (recently) in the past r states.

The formula for AM_4 is:

$$AM_{4,i,s} = \frac{SR_{i,r,s}}{\sum_{i=1}^p SR_{i,r,s}}$$

Parameters

<i>sr_i_r_s</i>	The number of matched state relations (SR) in the past r states in an intention (i) as of the current state (s).
<i>sum_sr_i_r_s</i>	The sum of all <i>sr_i_r_s</i> for each intention.

Returns

8.24.2.19 `static double ontology.Ontology.computeMetricAM5 (Intention intention)
[static]`

Compute the additive metric AM_5 .

- Suppose a kit is described by the number of parts it contains for each type.
That is, $kit_i = (n_{i_A}, n_{i_B}, \dots, n_{i_Q})$ has n_{i_A} parts of type “A”, n_{i_B} parts of type “B”, ..., n_{i_Q} parts of type “Q”.
- Suppose an observation is described by the number of parts seen for each type.

That is, $observation_j = (x_{jA}, x_{jB}, \dots, x_{jQ})$ has seen x_{jA} parts of type “A”, x_{jB} parts of type “B”, ..., x_{jQ} parts of type “Q”.

- The likelihood L of observation j under kit i is given by the multivariate hypergeometric distribution:

$$L(observation_j|kit_i) = \frac{\prod_{p=1}^q n_{ip} choose x_{jp}}{\sum_{p=1}^q n_{ip} choose \sum_{p=1}^q x_{jp}}$$

The additive metric AM_5 for kit_i is the probability of kit_i given $observation_j$:

$$Probability(kit_i|observation_j) = \frac{L(observation_j|kit_i)}{\sum_{n=1}^N L(observation_j|kit_n)}$$

where n is the total number of kits that are likely to be built.

8.24.2.20 `static double ontology.Ontology.computeMetricMM1 (int sr_i_s, int detrimental)`
`[static]`

8.24.2.21 `static void ontology.Ontology.computeMetricsInformation (ArrayList< ArrayList< ArrayList< String >>> _states) throws InterruptedException, BadLocationException` `[static]`

Retrieve and compute the information required for metrics computation.

The following pieces of information are used to determine individual metrics:

- $SR_{i,s}$: The number of matched state relations (SR) in an intention (i) as of the current state (s).
- $SR_{i,total}$: The number of state relations (SR), whether matched or not, in an intention (i).
- $SR_{all,s}$: The number of matched state relations (SR) in all possible intentions as of the current state (s).
- S_{total} : The number of states (S) that have occurred since observation began.
- S_i : The number of states (S) that have occurred since (and including) the first matched state relation in an intention (i).
- $detrimentalSR_{i,s}$: The number of detrimental state relations ($detrimentalSR$) that have occurred in an intention (i) as of the current state (s).
- $SR_{i,r,s}$: The number of matched state relations (SR) in the past r states in an intention (i) as of the current state (s). In other words, in the most recent n states, how many true state relations for an intention exist?

Parameters

<code>_states</code>	A list of states.
----------------------	-------------------

Exceptions

<code>InterruptedException</code>	
<code>BadLocationException</code>	

8.24.2.22 `static void ontology.Ontology.computeObservationLikelihood (Intention intention_)`
`[static]`

Compute the likelihood L of observation j under kit i .

The likelihood L of observation j under kit i is given by the multivariate hypergeometric distribution:

$$L(\text{observation}_j | \text{kit}_i) = \frac{\prod_{p=1}^q n_{i_p} \text{choose } x_{j_p}}{\sum_{p=1}^q n_{i_p} \text{choose } \sum_{p=1}^q x_{j_p}}$$

Returns

8.24.2.23 `static double ontology.Ontology.computePercentComplete (double sr_i_s, double sr_i_total)`
`[static]`

PercentComplete is the percentage complete for an intention in a state.

This formula is used by AM_2 and is computed as follows:

$$\text{PercentComplete}_{i,s} = \frac{SR_{i,s}}{SR_{i,total}}$$

Parameters

<code>sr_i_s</code>	The Number of matched state relations (SR) in an intention (i) as of the current state (s).
<code>sr_i_total</code>	The number of states (S) that have occurred since observation began.

Returns

8.24.2.24 `static double ontology.Ontology.computePercentProductive (double sr_i_s, double si) [static]`

The formula for the percent productive is used by AM_3 for an intention (*i*) as of state (*s*) is:

$$PercentProductive_{i,s} = \frac{SR_{i,s}}{S_i}$$

Parameters

<i>sr_i_s</i>	The Number of matched state relations (<i>SR</i>) in an intention (<i>i</i>) as of the current state (<i>s</i>).
<i>si</i>	The number of states (<i>S</i>) that have occurred since (and including) the first matched state relation in an intention (<i>i</i>).

Returns

8.24.2.25 `static void ontology.Ontology.computeProbabilityFromObservation (Intention intention_) [static]`

8.24.2.26 `static void ontology.Ontology.computeProceduresForAM5 () [static]`

8.24.2.27 `static void ontology.Ontology.getForEachIntentionTheNumberOfPartsForEachType () [static, private]`

Retrieve the number of parts each intention contains for each type.

Returns

Map <OWLIndividual, Map<String, Integer> > A HashMap that has the intention as the key and a HashMap as the value. The nested HashMap has the Target object (type of part) of the predicate as the key and the number of parts of the given type as value.

8.24.2.28 `Set<OWLClassExpression> ontology.Ontology.getIndividualClass (OWLNamedIndividual individual)`

8.24.2.29 `static String ontology.Ontology.getIndividualClassString (OWLNamedIndividual individual) [static]`

8.24.2.30 `static String ontology.Ontology.getIndividualClassString (OWLIndividual individual) [static]`

8.24.2.31 `OWLOntologyManager ontology.Ontology.getManager ()`

Simple getter.

Returns

manager

8.24.2.32 `OWL.Ontology ontology.Ontology.getOntology ()`

Simple getter.

Returns

ontology

8.24.2.33 `String ontology.Ontology.getOntologyPath ()`

Get the path of the ontology.

Returns**8.24.2.34** `static Color ontology.Ontology.getPartColor (String _string) [static, private]`**8.24.2.35** `static String ontology.Ontology.getPartType (String _string) [static, private]`**8.24.2.36** `String ontology.Ontology.getPath ()`

Simple getter.

Returns

path

8.24.2.37 `static String ontology.Ontology.getReferenceObjectClass (OWLNamedIndividual individual) [static, private]`

Return The class of the reference object for a given instance of predicate

Parameters

<i>individual</i>	An instance of predicate
-------------------	--------------------------

Returns

The class of the reference object

8.24.2.38 String ontology.Ontology.getRootClass ()

Get the class root from the ontology.

Returns

The class root

8.24.2.39 static char ontology.Ontology.getSeparator () [static]

Simple getter.

Returns

SEPARATOR

8.24.2.40 static String ontology.Ontology.getStateRelation (OWLNamedIndividual *individual*) [static, private]

Return The state relation for a given instance of predicate

Parameters

<i>individual</i>	An instance of predicate
-------------------	--------------------------

Returns

The state relation

8.24.2.41 static String ontology.Ontology.getStringHead (String *_string*) [static, private]

8.24.2.42 static String ontology.Ontology.getStringTail (String *_string*) [static, private]

8.24.2.43 NodeSet<OWLClass> ontology.Ontology.getSubclasses (String *myClassName*)

8.24.2.44 static String ontology.Ontology.getTargetObjectClass (OWLNamedIndividual *individual*) [static, private]

Return The class of the target object for a given instance of predicate

Parameters

<i>individual</i>	An instance of predicate
-------------------	--------------------------

Returns

The class of the target object

8.24.2.45 static boolean ontology.Ontology.hasProperty (OWLOntologyManager *man*,
OWLReasoner *reasoner*, OWLClass *cls*, OWLObjectPropertyExpression *prop*)
[static, private]

8.24.2.46 void ontology.Ontology.initializeList ()

8.24.2.47 static ArrayList<String> ontology.Ontology.input (String *state_relation*, String
target_object, String *target_class*, String *reference_object*, String *reference_class*)
[static, private]

8.24.2.48 void ontology.Ontology.loadFromFile ()

Load the ontology from a file.

8.24.2.49 void ontology.Ontology.loadOntologyFromPath (String *myPath*) throws
MalformedURLException, OWLException

8.24.2.50 static void ontology.Ontology.matchDetrimentalStateRelationToIntention (ArrayList
_state_relation) [static]

Check for each intention if *_state_relation* is a detrimental state relation.

Parameters

<i>_state_relation</i>	state relation of the form "State Relation Name","Target Object" ,"OWL CClass of the target object" ,"Reference object","OWL CClass of the reference object"
------------------------	--

8.24.2.51 static void ontology.Ontology.matchStateRelationToIntention (ArrayList
_state_relation, int *num*) throws BadLocationException [static]

Check for each intention if the state relation *_state_relation* matches any of the intention state relations.

Parameters

<i>_state_relation</i>	A state relation of the form "State Relation Name","Target Object" ,"OWL CClass of the target object" ,"Reference object","OWL CClass of the reference object"
------------------------	--

Exceptions

<i>BadLocationException</i>	
-----------------------------	--

8.24.2.52 `void ontology.Ontology.parseIntention (NodeSet< OWLClass > myClass)` throws `InterruptedException`, `InvocationTargetException`, `IOException`

Parse the class *myClass* and store data in arrays (HashMap and ArrayList).

Parameters

<i>myClass</i>	Class to be parsed
----------------	--------------------

Exceptions

<i>InterruptedException</i>	
<i>InvocationTargetException</i>	
<i>IOException</i>	

8.24.2.53 `static void ontology.Ontology.printProperties (OWLOntologyManager man, OWLOntology ont, OWLReasoner reasoner, OWLClass cls)` [`static`, `private`]

Print the properties that an instance has to have.

Parameters

<i>man</i>	The manager
<i>ont</i>	The ontology
<i>reasoner</i>	The reasoner
<i>cls</i>	The class for which we need to check the properties

8.24.2.54 `static void ontology.Ontology.readForEachIntentionTheNumberOfPartsForEachType ()` [`static`, `private`]

Read the map Map <OWLIndividual,Map<String, Integer> >

8.24.2.55 `void ontology.Ontology.readIntentionList ()` [`private`]

Read the intentions previously stored.

Display intention individuals.

8.24.2.56 `static void ontology.Ontology.readObservationList ()` [`static`]

Read the observation list `m_observation_list`.

8.24.2.57 `static ArrayList<String> ontology.Ontology.removeDuplicates (ArrayList<String> list_) [static]`

Remove duplicates in an ArrayList of String.

The easiest way to remove duplicates is to add the contents to a Set (which will not allow duplicates) and then add the Set back to the ArrayList.

Parameters

<i>list_</i>	Original list that contains duplicates
--------------	--

Returns

list_ with duplicates removed

8.24.2.58 `static double ontology.Ontology.roundTwoDecimals (double d) [static, package]`

8.24.2.59 `void ontology.Ontology.searchList (String individual) [private]`

8.24.2.60 `void ontology.Ontology.setDataFactory ()`

8.24.2.61 `static void ontology.Ontology.setInstanceFilePath (String path_) [static]`

Set the path to the OWL instance file The path is retrieved from the field [OptionFrame.m_instance_txt_field](#).

8.24.2.62 `void ontology.Ontology.setManager ()`

8.24.2.63 `void ontology.Ontology.setManager (OWLOntologyManager manager)`

Simple setter.

Parameters

<i>manager</i>	
----------------	--

8.24.2.64 `void ontology.Ontology.setOntology (OWLOntology ontology)`

Simple setter.

Parameters

<i>ontology</i>	
-----------------	--

8.24.2.65 void ontology.Ontology.setPath (String *path*)

Simple setter.

Parameters

<i>path</i>	
-------------	--

8.24.2.66 void ontology.Ontology.setReasoner (OWLOntology *myOntology*)

8.24.2.67 void ontology.Ontology.setRootClass (String *rootClass*)

Set the class root from the ontology.

Parameters

<i>rootClass</i>	
------------------	--

8.24.2.68 void ontology.Ontology.showDialogBox ()

8.24.2.69 void ontology.Ontology.sortIntentionList ()

Read *m_intention_list* and re-arrange the list using a number for each intention.

The bubble sort algorithm is used to sort *m_intention_list*.

Among the 5 intentions in the ontology, numbers are associated to each intention in the following way:

- *a4b3c3* = 1
- *a4b4c2* = 2
- *a2b3c5* = 3
- *a4b2c2d1* = 4
- *a2b3c3d1e1* = 5

8.24.2.70 static void ontology.Ontology.updateForEachIntentionTheNumberOfPartsForEachType () [static, private]

Update *m_part_type_number_for_each_intention_list* with missing part types.

m_part_type_number_for_each_intention_list contains the number of parts of each part type that each intention contains. For instance, the kit *kit_{a4b4c2}* has 4 for part A, 4 for part B, and 2 for part C. However, Parts D and E also need to be included in *m_part_type_number_for_each_intention_list* with 0 for part D and 0 for part E. The missing part types are retrieved from *m_observation_list*.

8.24.2.71 `static void ontology.Ontology.updateMainFrame (ArrayList _state_relation, int num)`
`[static]`

Display the current state relation in MainFrame.

Parameters

<i>_state_relation</i>	The state relation to display
------------------------	-------------------------------

8.24.2.72 `static void ontology.Ontology.updateObservationList (String part_type)`
`[static]`

Update the observation list.

This function searches for *part_type* in *m_observation_list* and updates its value (occurrence for this part type).

Parameters

<i>part_type</i>	The part type to search in <i>m_observation_list</i>
------------------	--

8.24.3 Member Data Documentation

8.24.3.1 `OWLDataFactory ontology.Ontology.m_datafactory` `[static, private]`

8.24.3.2 `String ontology.Ontology.m_hasCount_Occurrence = "#hasCount.Occurrence"` `[static, private]`

8.24.3.3 `String ontology.Ontology.m_hasIntention_Name = "#hasIntention_Name"`
`[static, private]`

8.24.3.4 `String ontology.Ontology.m_hasIntention_OrderingConstruct = "#hasIntention_OrderingConstruct"` `[static, private]`

8.24.3.5 `String ontology.Ontology.m_hasOrderingConstruct_OrderingConstruct = "#hasOrderingConstruct_OrderingConstruct"` `[static, private]`

8.24.3.6 `String ontology.Ontology.m_hasOrderingConstruct_Position = "#hasOrderingConstruct.Position"` `[static, private]`

8.24.3.7 `String ontology.Ontology.m_hasOrderingConstruct_Predicate = "#hasOrderingConstruct.Predicate"` `[static, private]`

8.24.3.8 `String ontology.Ontology.m_hasPredicate_ReferenceObject = "#hasPredicate.ReferenceObject"` `[static, private]`

- 8.24.3.9 `String ontology.Ontology.m_hasPredicate_TargetObject =
"#hasPredicate_TargetObject" [static, private]`
- 8.24.3.10 `ArrayList<Intention> ontology.Ontology.m_intention_list [static]`
- 8.24.3.11 `String ontology.Ontology.m_kitToBuild [static]`
- 8.24.3.12 `OWLOntologyManager ontology.Ontology.m_manager [private]`
- 8.24.3.13 `HashMap<String,Integer> ontology.Ontology.m_observation_list = new
HashMap<String, Integer>() [static, private]`
- 8.24.3.14 `OWLOntology ontology.Ontology.m_ontology [static]`
- 8.24.3.15 `final String ontology.Ontology.m_ontology_IRI =
"http://www.semanticweb.org/ontologies/2013/0/soap.owl" [static,
private]`
- 8.24.3.16 `Map<OWLIndividual,Map<String, Integer> > ontology.Ontology.m_
part_type_number_for_each_intention_list = new
HashMap<OWLIndividual,Map<String, Integer> >() [static, private]`
- 8.24.3.17 `String ontology.Ontology.m_path [private]`
- 8.24.3.18 `String ontology.Ontology.m_planToBuild [static]`
- 8.24.3.19 `ProgressBar ontology.Ontology.m_progress_bar [private]`
- 8.24.3.20 `JFrame ontology.Ontology.m_progress_frame [private]`
- 8.24.3.21 `OWLReasoner ontology.Ontology.m_reasoner [static, private]`
- 8.24.3.22 `String ontology.Ontology.m_s_ontopath [static, private]`
- 8.24.3.23 `String ontology.Ontology.m_s_rootClass = "Intention"`
- 8.24.3.24 `String ontology.Ontology.m_s_subClass = "Kitting" [static]`
- 8.24.3.25 `final char ontology.Ontology.m_SEPARATOR = '#' [static,
private]`

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

- [src/ontology/Ontology.java](#)

8.25 Ontology Class Reference

Class for the ontology.

8.25.1 Detailed Description

Class for the ontology.

This class is used to manipulate the ontology and extract data from it.

Author

zeidk

Date

2013/01/01

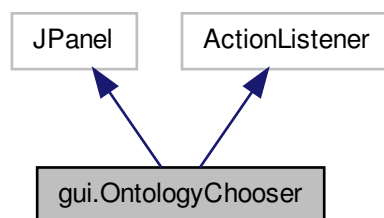
Contact: zeid.kootbally@nist.gov

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

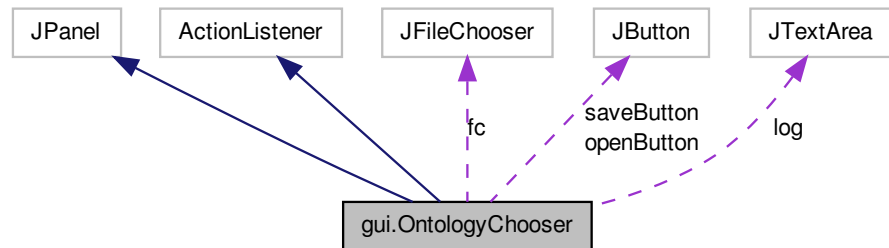
- [src/ontology/Ontology.java](#)

8.26 gui.OntologyChooser Class Reference

Inheritance diagram for gui.OntologyChooser:



Collaboration diagram for gui.OntologyChooser:



Public Member Functions

- [OntologyChooser](#) ()
- void [actionPerformed](#) (ActionEvent e)

Static Public Member Functions

- static void [createAndShowGUI](#) ()

Static Protected Member Functions

- static ImageIcon [createImageIcon](#) (String path)

Package Attributes

- JButton [openButton](#)
- JButton [saveButton](#)
- JTextArea [log](#)
- JFileChooser [fc](#)

Static Private Attributes

- static final String [newline](#) = "\n"

8.26.1 Detailed Description

Author

zeid This class consists of methods that allow the user to select the kittingInstances.owl file

8.26.2 Constructor & Destructor Documentation

8.26.2.1 `gui.OntologyChooser.OntologyChooser ()`

8.26.3 Member Function Documentation

8.26.3.1 `void gui.OntologyChooser.actionPerformed (ActionEvent e)`

8.26.3.2 `static void gui.OntologyChooser.createAndShowGUI () [static]`

Create the GUI and show it. For thread safety, this method should be invoked from the event dispatch thread.

8.26.3.3 `static ImageIcon gui.OntologyChooser.createImageIcon (String path) [static, protected]`

Returns an ImageIcon, or null if the path was invalid.

8.26.4 Member Data Documentation

8.26.4.1 `JFileChooser gui.OntologyChooser.fc [package]`

8.26.4.2 `JTextArea gui.OntologyChooser.log [package]`

8.26.4.3 `final String gui.OntologyChooser.newline = "\n" [static, private]`

8.26.4.4 `JButton gui.OntologyChooser.openButton [package]`

8.26.4.5 `JButton gui.OntologyChooser.saveButton [package]`

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

- [src/gui/OntologyChooser.java](#)

8.27 ontology.OnTopWithContactStateRelation Class Reference

Public Member Functions

- [OnTopWithContactStateRelation \(\)](#)

- void [setName](#) (final String name)

Private Attributes

- String [sr_name](#)

8.27.1 Constructor & Destructor Documentation

8.27.1.1 `ontology.OnTopWithContactStateRelation.OnTopWithContactStateRelation ()`

8.27.2 Member Function Documentation

8.27.2.1 `void ontology.OnTopWithContactStateRelation.setName (final String name)`

8.27.3 Member Data Documentation

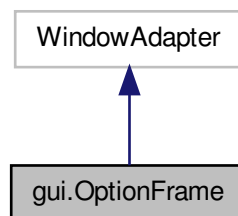
8.27.3.1 `String ontology.OnTopWithContactStateRelation.sr_name` [private]

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

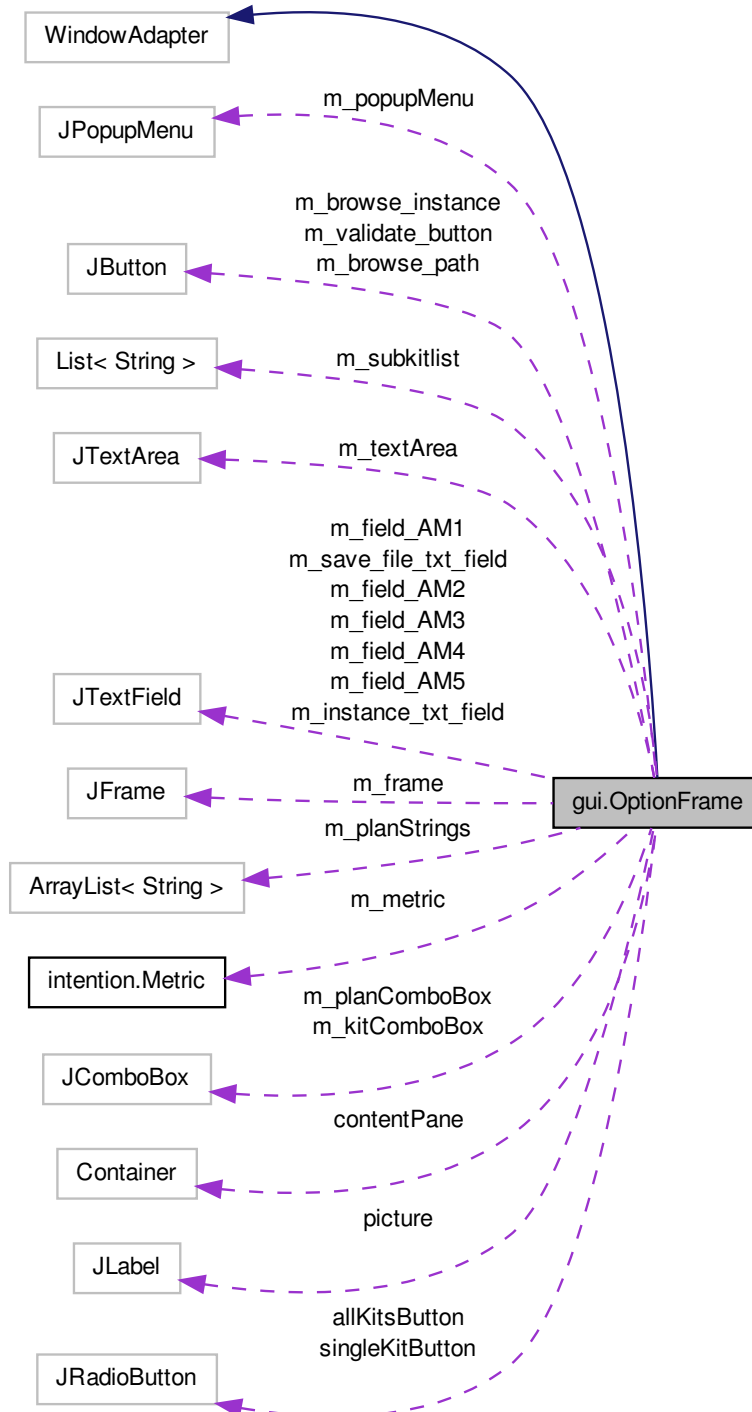
- `src/ontology/OnTopWithContactStateRelation.java`

8.28 gui.OptionFrame Class Reference

Inheritance diagram for gui.OptionFrame:



Collaboration diagram for gui.OptionFrame:



Classes

- class [MouseHandler](#)
- class [PlanComboBoxListener](#)

Public Member Functions

- [OptionFrame](#) ()
- void [addComponentsToPane](#) (Container pane, final [Ontology](#) onto)
- void [updateConfigFile](#) (String text) throws IOException
- List< String > [readConfigFile](#) (String configFile)
- void [createAndUpdateConfigFile](#) (String instance) throws FileNotFoundException, UnsupportedEncodingException
- void [createPopupMenu](#) ()
Create a popup menu when right-click on a JTextField.
- void [createAndShowGUI](#) ([Ontology](#) onto)
- void [setMetric](#) ([Metric](#) metric)
- [Metric](#) [getMetric](#) ()

Static Public Member Functions

- static ArrayList< String > [getKitList](#) ()
- static JButton [createButtonFromTemplate](#) (JButton _button)
- static void [findFilesinDirectory](#) (String plan_path)

Static Public Attributes

- static JTextField [m_instance_txt_field](#)
- static JButton [m_browse_instance](#)
- static JFrame [m_frame](#)
- static Boolean [m_validate](#)
- static JTextArea [m_textArea](#)
- static Container [contentPane](#)
- static List< String > [m_subkitlist](#)
- static JLabel [picture](#)
- static JRadioButton [singleKitButton](#)

Static Protected Member Functions

- static void [updateLabel](#) (String name)

Static Protected Attributes

- static final String [NO_DECORATIONS](#) = "no_dec"
- static final String [LF_DECORATIONS](#) = "laf_dec"
- static final String [WS_DECORATIONS](#) = "ws_dec"
- static final String [CREATE_WINDOW](#) = "new_win"
- static final String [DEFAULT_ICON](#) = "def_icon"
- static final String [FILE_ICON](#) = "file_icon"
- static final String [PAINT_ICON](#) = "paint_icon"

Package Attributes

- static JTextField [m_save_file_txt_field](#)
- static JTextField [m_field_AM1](#)
- static JTextField [m_field_AM2](#)
- static JTextField [m_field_AM3](#)
- static JTextField [m_field_AM4](#)
- static JTextField [m_field_AM5](#)
- static JButton [m_validate_button](#)
- static JButton [m_browse_path](#)
- static JComboBox [m_planComboBox](#)
- static JRadioButton [allKitsButton](#)

Static Package Attributes

- static final boolean [shouldFill](#) = false
- static final boolean [shouldWeightX](#) = true
- static final boolean [RIGHT_TO_LEFT](#) = false
- static boolean [m_bool_allKits](#) = false

Static Private Member Functions

- static void [add](#) (Component c, GridBagLayout gbl, GridBagConstraints gbc, int x, int y, int w, int h)

Private Attributes

- JPopupMenu [m_popupMenu](#)

Static Private Attributes

- static [Metric m_metric](#)
- static JComboBox [m_kitComboBox](#)
- static ArrayList< String > [m_planStrings](#) = new ArrayList<String>()

8.28.1 Constructor & Destructor Documentation

8.28.1.1 `gui.OptionFrame.OptionFrame ()`

8.28.2 Member Function Documentation

8.28.2.1 `static void gui.OptionFrame.add (Component c, GridBagLayout gbl, GridBagConstraints gbc, int x, int y, int w, int h)` [static, private]

8.28.2.2 `void gui.OptionFrame.addComponentToPane (Container pane, final Ontology onto)`

8.28.2.3 `void gui.OptionFrame.createAndShowGUI (Ontology onto)`

Create the GUI and show it. For thread safety, this method should be invoked from the event-dispatching thread.

8.28.2.4 `void gui.OptionFrame.createAndUpdateConfigFile (String instance)` throws `FileNotFoundException`, `UnsupportedEncodingException`

8.28.2.5 `static JButton gui.OptionFrame.createButtonFromTemplate (JButton _button)` [static]

8.28.2.6 `void gui.OptionFrame.createPopupMenu ()`

Create a popup menu when right-click on a JTextField.

8.28.2.7 `static void gui.OptionFrame.findFilesInDirectory (String plan_path)` [static]

8.28.2.8 `static ArrayList<String> gui.OptionFrame.getKitList ()` [static]

8.28.2.9 `Metric gui.OptionFrame.getMetric ()`

8.28.2.10 `List<String> gui.OptionFrame.readConfigFile (String configFile)`

8.28.2.11 `void gui.OptionFrame.setMetric (Metric metric)`

8.28.2.12 `void gui.OptionFrame.updateConfigFile (String text)` throws `IOException`

8.28.2.13 `static void gui.OptionFrame.updateLabel (String name)` [static, protected]

8.28.3 Member Data Documentation

8.28.3.1 `JRadioButton gui.OptionFrame.allKitsButton` [package]

- 8.28.3.2 Container `gui.OptionFrame.contentPane` [static]
- 8.28.3.3 final String `gui.OptionFrame.CREATE_WINDOW = "new_win"` [static, protected]
- 8.28.3.4 final String `gui.OptionFrame.DEFAULT_ICON = "def_icon"` [static, protected]
- 8.28.3.5 final String `gui.OptionFrame.FILE_ICON = "file_icon"` [static, protected]
- 8.28.3.6 final String `gui.OptionFrame.LF_DECORATIONS = "laf_dec"` [static, protected]
- 8.28.3.7 boolean `gui.OptionFrame.m_bool_allKits = false` [static, package]
- 8.28.3.8 JButton `gui.OptionFrame.m_browse_instance` [static]
- 8.28.3.9 JButton `gui.OptionFrame.m_browse_path` [package]
- 8.28.3.10 JTextField `gui.OptionFrame.m_field_AM1` [package]
- 8.28.3.11 JTextField `gui.OptionFrame.m_field_AM2` [package]
- 8.28.3.12 JTextField `gui.OptionFrame.m_field_AM3` [package]
- 8.28.3.13 JTextField `gui.OptionFrame.m_field_AM4` [package]
- 8.28.3.14 JTextField `gui.OptionFrame.m_field_AM5` [package]
- 8.28.3.15 JFrame `gui.OptionFrame.m_frame` [static]
- 8.28.3.16 JTextField `gui.OptionFrame.m_instance_txt_field` [static]
- 8.28.3.17 JComboBox `gui.OptionFrame.m_kitComboBox` [static, private]
- 8.28.3.18 Metric `gui.OptionFrame.m_metric` [static, private]
- 8.28.3.19 JComboBox `gui.OptionFrame.m_planComboBox` [package]
- 8.28.3.20 ArrayList<String> `gui.OptionFrame.m_planStrings = new ArrayList<String>()` [static, private]
- 8.28.3.21 JPopupMenu `gui.OptionFrame.m_popupMenu` [private]
- 8.28.3.22 JTextField `gui.OptionFrame.m_save_file_txt_field` [package]

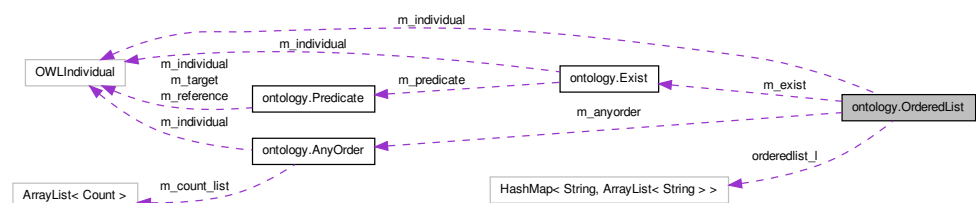
- 8.28.3.23 `List<String> gui.OptionFrame.m_subkitlist` [static]
- 8.28.3.24 `JTextArea gui.OptionFrame.m_textArea` [static]
- 8.28.3.25 `Boolean gui.OptionFrame.m_validate` [static]
- 8.28.3.26 `JButton gui.OptionFrame.m_validate_button` [package]
- 8.28.3.27 `final String gui.OptionFrame.NO_DECORATIONS = "no_dec"` [static, protected]
- 8.28.3.28 `final String gui.OptionFrame.PAINT_ICON = "paint_icon"` [static, protected]
- 8.28.3.29 `JLabel gui.OptionFrame.picture` [static]
- 8.28.3.30 `final boolean gui.OptionFrame.RIGHT_TO_LEFT = false` [static, package]
- 8.28.3.31 `final boolean gui.OptionFrame.shouldFill = false` [static, package]
- 8.28.3.32 `final boolean gui.OptionFrame.shouldWeightX = true` [static, package]
- 8.28.3.33 `JRadioButton gui.OptionFrame.singleKitButton` [static]
- 8.28.3.34 `final String gui.OptionFrame.WS_DECORATIONS = "ws_dec"` [static, protected]

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

- [src/gui/OptionFrame.java](#)

8.29 ontology.OrderedList Class Reference

Collaboration diagram for ontology.OrderedList:



Public Member Functions

- [OrderedList](#) ()
- [Exist](#) [getExist](#) ()
- void [setExist](#) ([Exist](#) e)
- [AnyOrder](#) [getAnyOrder](#) ()
- void [setAnyOrder](#) ([AnyOrder](#) ao)
- [OWLIndividual](#) [getIndividual](#) ()
- void [setIndividual](#) ([OWLIndividual](#) i)

Public Attributes

- [HashMap](#)< [String](#), [ArrayList](#)< [String](#) > > [orderedlist_l](#)

Private Attributes

- [String](#) [orderedlist_name](#)
- [Exist](#) [m_exist](#)
- [AnyOrder](#) [m_anyorder](#)
- [OWLIndividual](#) [m_individual](#)

8.29.1 Constructor & Destructor Documentation

8.29.1.1 [ontology.OrderedList.OrderedList](#) ()

8.29.2 Member Function Documentation

8.29.2.1 [AnyOrder](#) [ontology.OrderedList.getAnyOrder](#) ()

8.29.2.2 [Exist](#) [ontology.OrderedList.getExist](#) ()

8.29.2.3 [OWLIndividual](#) [ontology.OrderedList.getIndividual](#) ()

8.29.2.4 void [ontology.OrderedList.setAnyOrder](#) ([AnyOrder](#) ao)

8.29.2.5 void [ontology.OrderedList.setExist](#) ([Exist](#) e)

8.29.2.6 void [ontology.OrderedList.setIndividual](#) ([OWLIndividual](#) i)

8.29.3 Member Data Documentation

8.29.3.1 [AnyOrder](#) [ontology.OrderedList.m_anyorder](#) [private]

8.29.3.2 [Exist](#) [ontology.OrderedList.m_exist](#) [private]

8.29.3.3 OWLIndividual `ontology.OrderedList.m_individual` [private]

8.29.3.4 `HashMap<String, ArrayList<String> >` `ontology.OrderedList.orderedlist_I`

8.29.3.5 `String` `ontology.OrderedList.orderedlist_name` [private]

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

- `src/ontology/OrderedList.java`

8.30 `ontology.OrderingConstruct` Class Reference

Public Member Functions

- `OrderingConstruct()`

8.30.1 Constructor & Destructor Documentation

8.30.1.1 `ontology.OrderingConstruct.OrderingConstruct()`

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

- `src/ontology/OrderingConstruct.java`

8.31 `ontology.PartiallyInStateRelation` Class Reference

Public Member Functions

- `PartiallyInStateRelation()`
- `void setName(final String name)`

Private Attributes

- `String sr_name`

8.31.1 Constructor & Destructor Documentation

8.31.1.1 `ontology.PartiallyInStateRelation.PartiallyInStateRelation()`

8.31.2 Member Function Documentation

8.31.2.1 `void ontology.PartiallyInStateRelation.setName(final String name)`

8.31.3 Member Data Documentation

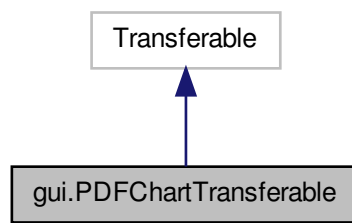
8.31.3.1 String `ontology.PartiallyInStateRelation.sr_name` [private]

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

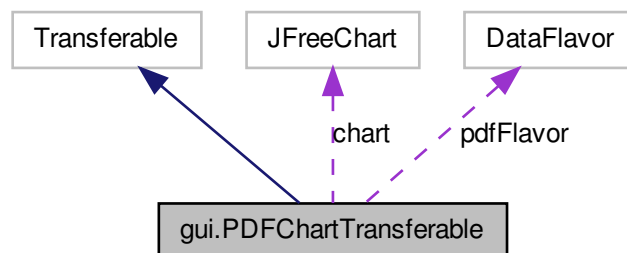
- `src/ontology/PartiallyInStateRelation.java`

8.32 gui.PDFChartTransferable Class Reference

Inheritance diagram for gui.PDFChartTransferable:



Collaboration diagram for gui.PDFChartTransferable:



Public Member Functions

- [PDFChartTransferable](#) (JFreeChart *jfreechart*, int *i*, int *j*)
- [PDFChartTransferable](#) (JFreeChart *jfreechart*, int *i*, int *j*, boolean *flag*)
- DataFlavor[] [getTransferDataFlavors](#) ()
- boolean [isDataFlavorSupported](#) (DataFlavor *dataflavor*)
- Object [getTransferData](#) (DataFlavor *dataflavor*) throws UnsupportedOperationException, IOException

Static Public Member Functions

- static void [writeChartAsPDF](#) (ByteArrayOutputStream *bytearrayoutputstream*, JFreeChart *jfreechart*, int *i*, int *j*, FontMapper *fontmapper*) throws IOException

Package Attributes

- final DataFlavor [pdfFlavor](#)

Private Attributes

- JFreeChart [chart](#)
- int [width](#)
- int [height](#)

8.32.1 Constructor & Destructor Documentation

8.32.1.1 `gui.PDFChartTransferable.PDFChartTransferable (JFreeChart jfreechart, int i, int j)`

8.32.1.2 `gui.PDFChartTransferable.PDFChartTransferable (JFreeChart jfreechart, int i, int j, boolean flag)`

8.32.2 Member Function Documentation

8.32.2.1 Object `gui.PDFChartTransferable.getTransferData (DataFlavor dataflavor)` throws UnsupportedOperationException, IOException

8.32.2.2 DataFlavor [] `gui.PDFChartTransferable.getTransferDataFlavors ()`

8.32.2.3 boolean `gui.PDFChartTransferable.isDataFlavorSupported (DataFlavor dataflavor)`

8.32.2.4 static void `gui.PDFChartTransferable.writeChartAsPDF (ByteArrayOutputStream bytearrayoutputstream, JFreeChart jfreechart, int i, int j, FontMapper fontmapper)`
throws IOException `[static]`

8.32.3 Member Data Documentation

8.32.3.1 JFreeChart gui.PDFChartTransferable.chart [private]

8.32.3.2 int gui.PDFChartTransferable.height [private]

8.32.3.3 final DataFlavor gui.PDFChartTransferable.pdfFlavor [package]

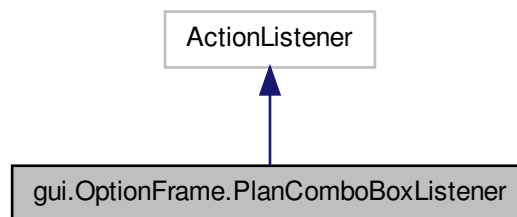
8.32.3.4 int gui.PDFChartTransferable.width [private]

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

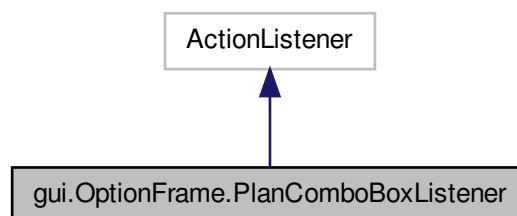
- src/gui/[PDFChartTransferable.java](#)

8.33 gui.OptionFrame.PlanComboBoxListener Class Reference

Inheritance diagram for gui.OptionFrame.PlanComboBoxListener:



Collaboration diagram for gui.OptionFrame.PlanComboBoxListener:



Public Member Functions

- void [actionPerformed](#) (ActionEvent e)

8.33.1 Member Function Documentation

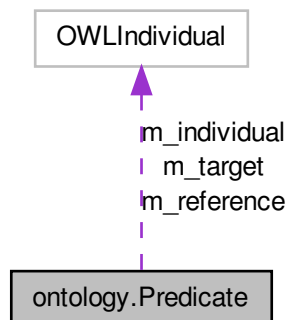
8.33.1.1 void gui.OptionFrame.PlanComboBoxListener.actionPerformed (ActionEvent e)

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

- src/gui/[OptionFrame.java](#)

8.34 ontology.Predicate Class Reference

Collaboration diagram for ontology.Predicate:



Public Member Functions

- [Predicate](#) ()
- OWLIndividual [getIndividual](#) ()
- OWLIndividual [getReference](#) ()
- String [getReferenceObjectClass](#) ()
- String [getStateRelation](#) ()
- OWLIndividual [getTarget](#) ()
- String [getTargetObjectClass](#) ()
- void [setIndividual](#) (OWLIndividual i)
- void [setReference](#) (OWLIndividual i)

- void [setReferenceObjectClass](#) (String s)
- void [setStateRelation](#) (String s)
- void [setTarget](#) (OWLIndividual i)
- void [setTargetObjectClass](#) (String s)

Private Attributes

- OWLIndividual [m_individual](#)
- OWLIndividual [m_reference](#)
- String [m_reference_class](#)
- String [m_staterelation](#)
- OWLIndividual [m_target](#)
- String [m_target_class](#)

8.34.1 Constructor & Destructor Documentation

8.34.1.1 `ontology.Predicate.Predicate ()`

8.34.2 Member Function Documentation

8.34.2.1 `OWLIndividual ontology.Predicate.getIndividual ()`

8.34.2.2 `OWLIndividual ontology.Predicate.getReference ()`

8.34.2.3 `String ontology.Predicate.getReferenceObjectClass ()`

8.34.2.4 `String ontology.Predicate.getStateRelation ()`

8.34.2.5 `OWLIndividual ontology.Predicate.getTarget ()`

8.34.2.6 `String ontology.Predicate.getTargetObjectClass ()`

8.34.2.7 `void ontology.Predicate.setIndividual (OWLIndividual i)`

8.34.2.8 `void ontology.Predicate.setReference (OWLIndividual i)`

8.34.2.9 `void ontology.Predicate.setReferenceObjectClass (String s)`

8.34.2.10 `void ontology.Predicate.setStateRelation (String s)`

8.34.2.11 `void ontology.Predicate.setTarget (OWLIndividual i)`

8.34.2.12 `void ontology.Predicate.setTargetObjectClass (String s)`

8.34.3 Member Data Documentation

8.34.3.1 `OWLIndividual ontology.Predicate.m_individual` `[private]`

8.34.3.2 `OWLIndividual ontology.Predicate.m_reference` `[private]`

8.34.3.3 `String ontology.Predicate.m_reference_class` `[private]`

8.34.3.4 `String ontology.Predicate.m_staterelation` `[private]`

8.34.3.5 `OWLIndividual ontology.Predicate.m_target` `[private]`

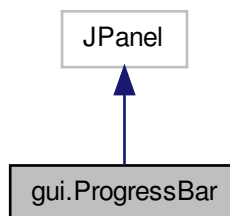
8.34.3.6 `String ontology.Predicate.m_target_class` `[private]`

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

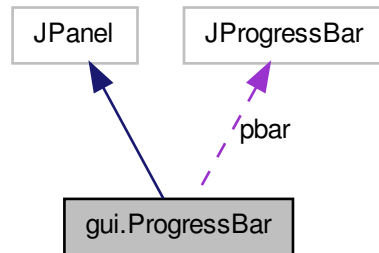
- `src/ontology/Predicate.java`

8.35 `gui.ProgressBar` Class Reference

Inheritance diagram for `gui.ProgressBar`:



Collaboration diagram for gui.ProgressBar:



Public Member Functions

- [ProgressBar](#) ()
- void [updateBar](#) (int newValue)

Package Attributes

- JProgressBar [pbar](#)

Static Package Attributes

- static final int [MY_MINIMUM](#) = 0
- static final int [MY_MAXIMUM](#) = 100

8.35.1 Constructor & Destructor Documentation

8.35.1.1 `gui.ProgressBar.ProgressBar ()`

8.35.2 Member Function Documentation

8.35.2.1 `void gui.ProgressBar.updateBar (int newValue)`

8.35.3 Member Data Documentation

8.35.3.1 `final int gui.ProgressBar.MY_MAXIMUM = 100` [static, package]

8.35.3.2 `final int gui.ProgressBar.MY_MINIMUM = 0` [static, package]

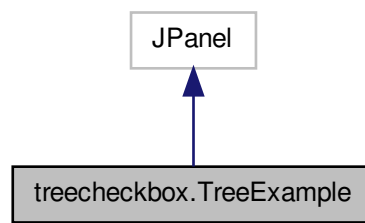
8.35.3.3 `JProgressBar gui.ProgressBar.pbar` [package]

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

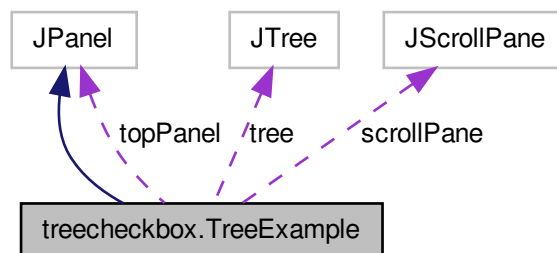
- [src/gui/ProgressBar.java](#)

8.36 treecheckbox.TreeExample Class Reference

Inheritance diagram for treecheckbox.TreeExample:



Collaboration diagram for treecheckbox.TreeExample:



Public Member Functions

- [TreeExample](#) ()

Private Attributes

- JPanel [topPanel](#)
- JTree [tree](#)
- JScrollPane [scrollPane](#)

Static Private Attributes

- static final long [serialVersionUID](#) = 1L

8.36.1 Constructor & Destructor Documentation

8.36.1.1 `treecheckbox.TreeExample.TreeExample ()`

8.36.2 Member Data Documentation

8.36.2.1 `JScrollPane treecheckbox.TreeExample.scrollPane` [private]

8.36.2.2 `final long treecheckbox.TreeExample.serialVersionUID = 1L` [static, private]

8.36.2.3 `JPanel treecheckbox.TreeExample.topPanel` [private]

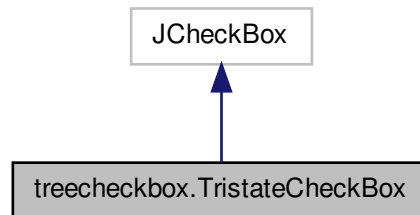
8.36.2.4 `JTree treecheckbox.TreeExample.tree` [private]

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

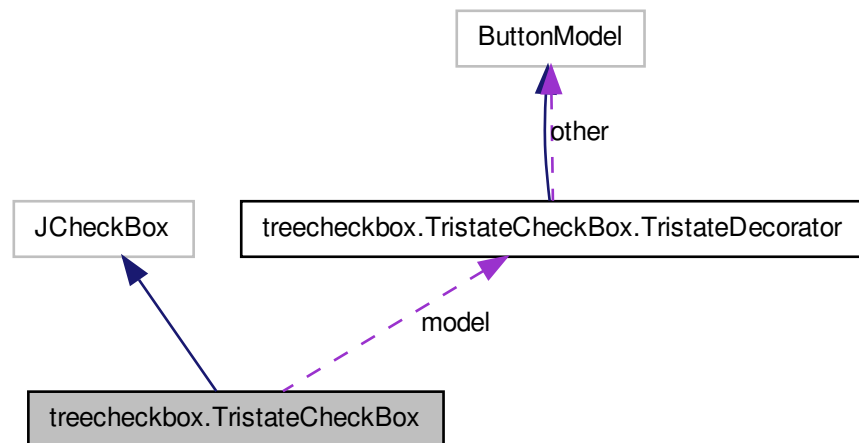
- `src/treecheckbox/TreeExample.java`

8.37 treecheckbox.TristateCheckBox Class Reference

Inheritance diagram for treecheckbox.TristateCheckBox:



Collaboration diagram for treecheckbox.TristateCheckBox:



Classes

- class [TristateDecorator](#)

Public Member Functions

- [TristateCheckBox](#) (String text, Icon icon, Boolean initial)
- [TristateCheckBox](#) (String text, Boolean initial)
- [TristateCheckBox](#) (String text)
- [TristateCheckBox](#) ()
- void [addMouseListener](#) (MouseListener l)
- void [setState](#) (Boolean state)
- Boolean [getState](#) ()

Private Attributes

- final [TristateDecorator model](#)

8.37.1 Detailed Description

Maintenance tip - There were some tricks to getting this code working:

1. You have to overwrite [addMouseListener\(\)](#) to do nothing 2. You have to add a mouse event on mousePressed by calling super.addMouseListener() 3. You have to replace the UIActionMap for the keyboard event "pressed" with your own one. 4. You have to remove the UIActionMap for the keyboard event "released". 5. You have to grab focus when the next state is entered, otherwise clicking on the component won't get the focus. 6. You have to make a [TristateDecorator](#) as a button model that wraps the original button model and does state management.

8.37.2 Constructor & Destructor Documentation

8.37.2.1 `treecheckbox.TristateCheckBox.TristateCheckBox (String text, Icon icon, Boolean initial)`

8.37.2.2 `treecheckbox.TristateCheckBox.TristateCheckBox (String text, Boolean initial)`

8.37.2.3 `treecheckbox.TristateCheckBox.TristateCheckBox (String text)`

8.37.2.4 `treecheckbox.TristateCheckBox.TristateCheckBox ()`

8.37.3 Member Function Documentation

8.37.3.1 `void treecheckbox.TristateCheckBox.addMouseListener (MouseListener l)`

No one may add mouse listeners, not even Swing!

8.37.3.2 `Boolean treecheckbox.TristateCheckBox.getState ()`

Return the current state, which is determined by the selection status of the model.

8.37.3.3 void treecheckbox.TristateCheckBox.setState (Boolean state)

Set the new state to either SELECTED, NOT_SELECTED or DONT_CARE. If state == null, it is treated as DONT_CARE.

8.37.4 Member Data Documentation

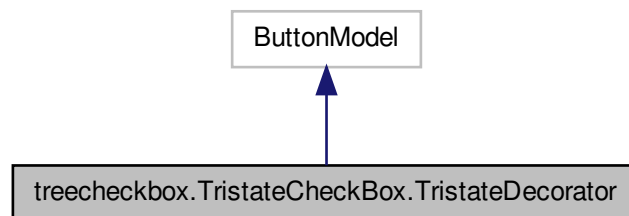
8.37.4.1 final TristateDecorator treecheckbox.TristateCheckBox.model [private]

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

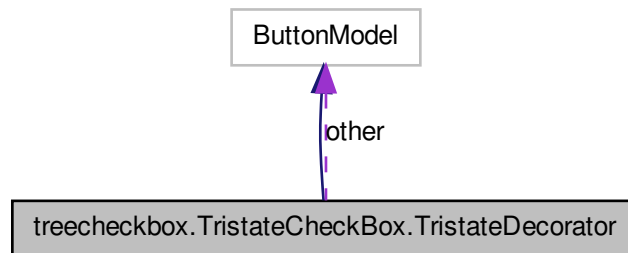
- src/treecheckbox/[TristateCheckBox.java](#)

8.38 treecheckbox.TristateCheckBox.TristateDecorator Class Reference

Inheritance diagram for treecheckbox.TristateCheckBox.TristateDecorator:



Collaboration diagram for treecheckbox.TristateCheckBox.TristateDecorator:



Public Member Functions

- void [setArmed](#) (boolean b)
- boolean [isFocusTraversable](#) ()
- void [setEnabled](#) (boolean b)
- boolean [isArmed](#) ()
- boolean [isSelected](#) ()
- boolean [isEnabled](#) ()
- boolean [isPressed](#) ()
- boolean [isRollover](#) ()
- void [setSelected](#) (boolean b)
- void [setPressed](#) (boolean b)
- void [setRollover](#) (boolean b)
- void [setMnemonic](#) (int key)
- int [getMnemonic](#) ()
- void [setActionCommand](#) (String s)
- String [getActionCommand](#) ()
- void [setGroup](#) (ButtonGroup group)
- void [addActionListener](#) (ActionListener l)
- void [removeActionListener](#) (ActionListener l)
- void [addItemListener](#) (ItemListener l)
- void [removeItemListener](#) (ItemListener l)
- void [addChangeListener](#) (ChangeListener l)
- void [removeChangeListener](#) (ChangeListener l)
- Object[] [getSelectedObjects](#) ()

Private Member Functions

- [TristateDecorator](#) (ButtonModel [other](#))
- void [setState](#) (Boolean state)
- Boolean [getState](#) ()
- void [nextState](#) ()

Private Attributes

- final ButtonModel [other](#)

8.38.1 Detailed Description

Exactly which Design Pattern is this? Is it an Adapter, a Proxy or a Decorator? In this case, my vote lies with the Decorator, because we are extending functionality and "decorating" the original model with a more powerful model.

8.38.2 Constructor & Destructor Documentation

8.38.2.1 `treecheckbox.TristateCheckBox.TristateDecorator.TristateDecorator (ButtonModel other) [private]`

8.38.3 Member Function Documentation

8.38.3.1 `void treecheckbox.TristateCheckBox.TristateDecorator.addActionListener (ActionListener I)`

8.38.3.2 `void treecheckbox.TristateCheckBox.TristateDecorator.addChangeListener (ChangeListener I)`

8.38.3.3 `void treecheckbox.TristateCheckBox.TristateDecorator.addItemListener (ItemListener I)`

8.38.3.4 `String treecheckbox.TristateCheckBox.TristateDecorator.getActionCommand ()`

8.38.3.5 `int treecheckbox.TristateCheckBox.TristateDecorator.getMnemonic ()`

8.38.3.6 `Object [] treecheckbox.TristateCheckBox.TristateDecorator.getSelectedObjects ()`

8.38.3.7 `Boolean treecheckbox.TristateCheckBox.TristateDecorator.getState () [private]`

The current state is embedded in the selection / armed state of the model.

We return the SELECTED state when the checkbox is selected but not armed, DONT_CARE state when the checkbox is selected and armed (grey) and NOT_SELECTED when the checkbox is deselected.

8.38.3.8 boolean treecheckbox.TristateCheckBox.TristateDecorator.isArmed ()

All these methods simply delegate to the "other" model that is being decorated.

8.38.3.9 boolean treecheckbox.TristateCheckBox.TristateDecorator.isEnabled ()

8.38.3.10 boolean treecheckbox.TristateCheckBox.TristateDecorator.isFocusTraversable ()

8.38.3.11 boolean treecheckbox.TristateCheckBox.TristateDecorator.isPressed ()

8.38.3.12 boolean treecheckbox.TristateCheckBox.TristateDecorator.isRollover ()

8.38.3.13 boolean treecheckbox.TristateCheckBox.TristateDecorator.isSelected ()

8.38.3.14 void treecheckbox.TristateCheckBox.TristateDecorator.nextState () [private]

We rotate between NOT_SELECTED, SELECTED and DONT_CARE.

8.38.3.15 void treecheckbox.TristateCheckBox.TristateDecorator.removeActionListener (
 ActionListener l)

8.38.3.16 void treecheckbox.TristateCheckBox.TristateDecorator.removeChangeListener (
 ChangeListener l)

8.38.3.17 void treecheckbox.TristateCheckBox.TristateDecorator.removeItemListener (
 ItemListener l)

8.38.3.18 void treecheckbox.TristateCheckBox.TristateDecorator.setActionCommand (String s)

8.38.3.19 void treecheckbox.TristateCheckBox.TristateDecorator.setArmed (boolean b)

Filter: No one may change the armed status except us.

8.38.3.20 void treecheckbox.TristateCheckBox.TristateDecorator.setEnabled (boolean b)

We disable focusing on the component when it is not enabled.

8.38.3.21 void treecheckbox.TristateCheckBox.TristateDecorator.setGroup (ButtonGroup group
)

8.38.3.22 void treecheckbox.TristateCheckBox.TristateDecorator.setMnemonic (int key)

8.38.3.23 void treecheckbox.TristateCheckBox.TristateDecorator.setPressed (boolean b)

8.38.3.24 void treecheckbox.TristateCheckBox.TristateDecorator.setRollover (boolean b)

8.38.3.25 void treecheckbox.TristateCheckBox.TristateDecorator.setSelected (boolean *b*)

8.38.3.26 void treecheckbox.TristateCheckBox.TristateDecorator.setState (Boolean *state*)
[private]

8.38.4 Member Data Documentation

8.38.4.1 final ButtonModel treecheckbox.TristateCheckBox.TristateDecorator.other
[private]

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

- src/treecheckbox/[TristateCheckBox.java](#)

8.39 ontology.UnderWithContactStateRelation Class Reference

Public Member Functions

- [UnderWithContactStateRelation](#) ()
- void [setName](#) (final String name)

Private Attributes

- String [sr_name](#)

8.39.1 Constructor & Destructor Documentation

8.39.1.1 ontology.UnderWithContactStateRelation.UnderWithContactStateRelation ()

8.39.2 Member Function Documentation

8.39.2.1 void ontology.UnderWithContactStateRelation.setName (final String *name*)

8.39.3 Member Data Documentation

8.39.3.1 String ontology.UnderWithContactStateRelation.[sr_name](#) [private]

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

- src/ontology/[UnderWithContactStateRelation.java](#)

Chapter 9

File Documentation

9.1 src/component/IntFilter.java File Reference

Classes

- class [component.IntFilter](#)

Packages

- package [component](#)

9.2 src/gui/Chart.java File Reference

Classes

- class [gui.Chart](#)
Chart display for metrics and likelihoods.
- class [gui.Chart.CustomRenderer](#)
- class [gui.Chart.CustomRendererLine](#)

Packages

- package [gui](#)

9.3 src/gui/CommonGUIComponents.java File Reference

Classes

- class [gui.CommonGUIComponents](#)

Common GUI components used across different files of the project.

Packages

- package [gui](#)

9.4 src/gui/DemoPanel.java File Reference

Classes

- class [gui.DemoPanel](#)

Packages

- package [gui](#)

9.5 src/gui/DrawStringPanel.java File Reference

Classes

- class [gui.DrawStringPanel](#)

Packages

- package [gui](#)

9.6 src/gui/MainFrame.java File Reference

Classes

- class [gui.MainFrame](#)
- class [gui.MainFrame.DisplayMetrics](#)

Packages

- package [gui](#)

9.7 src/gui/OntologyChooser.java File Reference

Classes

- class [gui.OntologyChooser](#)

Packages

- package [gui](#)

9.8 src/gui/OptionFrame.java File Reference

Classes

- class [gui.OptionFrame](#)
- class [gui.OptionFrame.PlanComboBoxListener](#)
- class [gui.OptionFrame.MouseHandler](#)

Packages

- package [gui](#)

9.9 src/gui/PDFChartTransferable.java File Reference

Classes

- class [gui.PDFChartTransferable](#)

Packages

- package [gui](#)

9.10 src/gui/ProgressBar.java File Reference

Classes

- class [gui.ProgressBar](#)

Packages

- package [gui](#)

9.11 src/intention/Intention.java File Reference

Classes

- class [intention.Intention](#)
Representation of intentions from their definition in the ontology.

Packages

- package [intention](#)

9.12 src/intention/Likelihood.java File Reference

Classes

- class [intention.Likelihood](#)

Packages

- package [intention](#)

9.13 src/intention/Metric.java File Reference

Classes

- class [intention.Metric](#)
Definition of additive and multiplicative metrics.

Packages

- package [intention](#)

9.14 src/main/Launcher.java File Reference

Contains the main of the program.

Classes

- class [main.Launcher](#)
Main class of the tool.

Packages

- package [main](#)

9.14.1 Detailed Description

Contains the main of the program.

Author

[Zeid Kootbally](#) zeid.kootbally@nist.gov

Version

1.0

Date

September 2013

Precondition

Make sure the kits directory is present in the same directory as this tool
Make sure kittingClasses.owl, kittingInstances_ir.owl, and soap.owl are in the same directory as this tool

9.15 src/ontology/AnyOrder.java File Reference

Classes

- class [ontology.AnyOrder](#)

Packages

- package [ontology](#)

9.16 src/ontology/Count.java File Reference

Classes

- class [ontology.Count](#)

Packages

- package [ontology](#)

9.17 src/ontology/Exist.java File Reference

Classes

- class [ontology.Exist](#)

Packages

- package [ontology](#)

9.18 src/ontology/Ontology.java File Reference

Classes

- class [ontology.Ontology](#)

Packages

- package [ontology](#)

9.19 src/ontology/OnTopWithContactStateRelation.java File Reference

Classes

- class [ontology.OnTopWithContactStateRelation](#)

Packages

- package [ontology](#)

9.20 src/ontology/OrderedList.java File Reference

Classes

- class [ontology.OrderedList](#)

Packages

- package [ontology](#)

9.21 src/ontology/OrderingConstruct.java File Reference

Classes

- class [ontology.OrderingConstruct](#)

Packages

- package [ontology](#)

9.22 src/ontology/PartiallyInStateRelation.java File Reference

Classes

- class [ontology.PartiallyInStateRelation](#)

Packages

- package [ontology](#)

9.23 src/ontology/Predicate.java File Reference

Classes

- class [ontology.Predicate](#)

Packages

- package [ontology](#)

9.24 src/ontology/UnderWithContactStateRelation.java File Reference

Classes

- class [ontology.UnderWithContactStateRelation](#)

Packages

- package [ontology](#)

9.25 src/textfiles/FileOperator.java File Reference

Classes

- class [textfiles.FileOperator](#)

Packages

- package [textfiles](#)

9.26 src/tools/Configuration.java File Reference

Classes

- class [tools.Configuration](#)

Packages

- package [tools](#)

9.27 src/treecheckbox/CheckTreeCellRenderer.java File Reference

Classes

- class [treecheckbox.CheckTreeCellRenderer](#)

Packages

- package [treecheckbox](#)

9.28 src/treecheckbox/CheckTreeManager.java File Reference

Classes

- class [treecheckbox.CheckTreeManager](#)

Packages

- package [treecheckbox](#)

9.29 src/treecheckbox/CheckTreeSelectionModel.java File Reference

Classes

- class [treecheckbox.CheckTreeSelectionModel](#)

Packages

- package [treecheckbox](#)

9.30 src/treecheckbox/TreeExample.java File Reference

Classes

- class [treecheckbox.TreeExample](#)

Packages

- package [treecheckbox](#)

9.31 src/treecheckbox/TristateCheckBox.java File Reference

Classes

- class [treecheckbox.TristateCheckBox](#)
- class [treecheckbox.TristateCheckBox.TristateDecorator](#)

Packages

- package [treecheckbox](#)

Index

`_sd`
 gui::MainFrame, 74

`actionPerformed`
 gui::MainFrame, 73
 gui::MainFrame::DisplayMetrics, 41
 gui::OntologyChooser, 105
 gui::OptionFrame::PlanComboBoxListener,
 118

`add`
 gui::OptionFrame, 110

`addActionListener`
 treecheckbox::TristateCheckBox::TristateDecorator,
 128

`addChangeListener`
 treecheckbox::TristateCheckBox::TristateDecorator,
 128

`addChart`
 gui::DemoPanel, 40

`addChildPaths`
 treecheckbox::CheckTreeManager, 26

`addComponentsToPane`
 gui::OptionFrame, 110

`addCountToList`
 ontology::AnyOrder, 18

`addItemListener`
 treecheckbox::TristateCheckBox::TristateDecorator,
 128

`addMouseListener`
 treecheckbox::TristateCheckBox, 125

`addSelectionPaths`
 treecheckbox::CheckTreeSelectionModel,
 28

`allKitsButton`
 gui::OptionFrame, 110

`anchor`
 gui::DrawStringPanel, 43

`angle`
 gui::DrawStringPanel, 44

`AnyOrder`
 ontology::AnyOrder, 18

`applyThemeToChart`
 gui::MainFrame, 73

`areSiblingsSelected`
 treecheckbox::CheckTreeSelectionModel,
 28

`attemptExit`
 gui::MainFrame, 73

`buildDetrimentalList`
 intention::Intention, 52

`buildIntentionList`
 ontology::Ontology, 86

`buildIntentionMetricsTree`
 ontology::Ontology, 87

`buildIntentionTree`
 ontology::Ontology, 87

`buildLikelihoodPanel`
 gui::MainFrame, 73

`buildMetricPanel`
 gui::MainFrame, 73

`buildObservationList`
 ontology::Ontology, 87

`buildStates`
 ontology::Ontology, 87

`Chart`
 gui::PDFChartTransferable, 116

`chart`
 chartLeftPaneColor
 gui::CommonGUIComponents, 31
 chartPanelColor
 gui::CommonGUIComponents, 31

`charts`
 gui::DemoPanel, 40

`checkBox`
 treecheckbox::CheckTreeCellRenderer,
 25

`CheckTreeCellRenderer`
 treecheckbox::CheckTreeCellRenderer,
 25

- CheckTreeManager
 - treecheckbox::CheckTreeManager, [26](#)
- CheckTreeSelectionModel
 - treecheckbox::CheckTreeSelectionModel, [28](#)
- choose
 - ontology::Ontology, [87](#)
- chooseTest
 - ontology::Ontology, [88](#)
- cleanDataPropertyInteger
 - ontology::Ontology, [88](#)
- cleanIRI
 - ontology::Ontology, [88](#)
- colors
 - gui::Chart::CustomRenderer, [37](#)
 - gui::Chart::CustomRendererLine, [38](#)
- component, [13](#)
- component::IntFilter, [66](#)
 - insertString, [67](#)
 - remove, [67](#)
 - replace, [67](#)
 - test, [67](#)
- compute_SR_all_s
 - ontology::Ontology, [88](#)
- compute_SR_i_r_s
 - ontology::Ontology, [88](#)
- compute_sum_observation
 - ontology::Ontology, [88](#)
- compute_sum_part_type
 - ontology::Ontology, [89](#)
- computeLikelihood
 - ontology::Ontology, [89](#)
- computeMetricAM1
 - ontology::Ontology, [89](#)
- computeMetricAM2
 - ontology::Ontology, [90](#)
- computeMetricAM3
 - ontology::Ontology, [90](#)
- computeMetricAM4
 - ontology::Ontology, [91](#)
- computeMetricAM5
 - ontology::Ontology, [91](#)
- computeMetricMM1
 - ontology::Ontology, [92](#)
- computeMetricsInformation
 - ontology::Ontology, [92](#)
- computeObservationLikelihood
 - ontology::Ontology, [93](#)
- computePercentComplete
 - ontology::Ontology, [93](#)
- computePercentProductive
 - ontology::Ontology, [93](#)
- computeProbabilityFromObservation
 - ontology::Ontology, [94](#)
- computeProceduresForAM5
 - ontology::Ontology, [94](#)
- contentPane
 - gui::OptionFrame, [110](#)
- copyToClipboard
 - gui::MainFrame, [73](#)
- Count
 - ontology::Count, [35](#)
- CREATE_WINDOW
 - gui::OptionFrame, [111](#)
- createAndShowGUI
 - gui::OntologyChooser, [105](#)
 - gui::OptionFrame, [110](#)
- createAndUpdateConfigFile
 - gui::OptionFrame, [110](#)
- createButtonFromTemplate
 - gui::OptionFrame, [110](#)
- createChart_likelihood
 - gui::Chart, [21](#)
- createChart_metrics
 - gui::Chart, [21](#)
- createContent
 - gui::MainFrame, [73](#)
- createImagelcon
 - gui::OntologyChooser, [105](#)
- createLikelihoodChartPanel
 - gui::MainFrame, [74](#)
- createLikelihoodDataset
 - gui::Chart, [21](#)
- createMenuBar
 - gui::MainFrame, [74](#)
- createMetricsChartPanel
 - gui::MainFrame, [74](#)
- createMetricsDataset
 - gui::Chart, [21](#)
- createPopupMenu
 - gui::OptionFrame, [110](#)
- CustomRenderer
 - gui::Chart::CustomRenderer, [37](#)
- CustomRendererLine
 - gui::Chart::CustomRendererLine, [38](#)
- DEFAULT_ICON
 - gui::OptionFrame, [111](#)
- delegate

- treecheckbox::CheckTreeCellRenderer, 25
- DemoPanel
 - gui::DemoPanel, 40
- DocumentFilter, 41
- DrawStringPanel
 - gui::DrawStringPanel, 43
- editorPane
 - gui::MainFrame, 75
- enabler
 - main::Launcher, 69
- Exist
 - ontology::Exist, 45
- EXIT_COMMAND
 - gui::MainFrame, 75
- exportToCSV
 - gui::MainFrame, 74
- exportToPDF
 - gui::MainFrame, 74
- fc
 - gui::OntologyChooser, 105
- FILE_ICON
 - gui::OptionFrame, 111
- FileOperator
 - textfiles::FileOperator, 47
- findFilesinDirectory
 - gui::OptionFrame, 110
- font
 - gui::DrawStringPanel, 44
- get_AM1_Weight
 - intention::Metric, 77
- get_AM2_Weight
 - intention::Metric, 77
- get_AM3_Weight
 - intention::Metric, 77
- get_AM4_Weight
 - intention::Metric, 77
- get_AM5_Weight
 - intention::Metric, 78
- get_MM1_Weight
 - intention::Metric, 78
- getActionCommand
 - treecheckbox::TristateCheckBox::TristateCheckBox, 128
- getAllCheckedPaths
 - treecheckbox::CheckTreeManager, 27
- getAnyOrder
 - ontology::OrderedList, 113
- getCharts
 - gui::DemoPanel, 40
- getChartTitle
 - gui::Chart, 21
- getCountList
 - ontology::AnyOrder, 18
- getDescendants
 - treecheckbox::CheckTreeManager, 27
- getDetrimentalList
 - intention::Intention, 52
- getExist
 - ontology::OrderedList, 113
- getFont
 - gui::DrawStringPanel, 43
- getForEachIntentionTheNumberOfPartsForEachType
 - ontology::Ontology, 94
- getIndividual
 - intention::Intention, 52
 - ontology::AnyOrder, 18
 - ontology::Count, 35
 - ontology::Exist, 45
 - ontology::OrderedList, 113
 - ontology::Predicate, 119
- getIndividualClass
 - ontology::Ontology, 94
- getIndividualClassString
 - ontology::Ontology, 94
- getIntentionName
 - intention::Intention, 53
- getItemPaint
 - gui::Chart::CustomRenderer, 37
 - gui::Chart::CustomRendererLine, 38
- getKitList
 - gui::OptionFrame, 110
- getLegendItems
 - gui::Chart, 21
- getM_am1
 - intention::Intention, 53
- getM_am2
 - intention::Intention, 53
- getM_am3
 - intention::Intention, 53
- getM_am4
 - intention::Intention, 53
- getM_am5
 - intention::Intention, 53
- getM_built_kit
 - intention::Intention, 54
- getM_found_detrimental_SR

- intention::Intention, [54](#)
- getM_intention_number
 - intention::Intention, [54](#)
- getM_likelihood_observation
 - intention::Intention, [54](#)
- getM_map_SRirs
 - intention::Intention, [54](#)
- getM_mm1
 - intention::Intention, [54](#)
- getM_percentComplete_i_s
 - intention::Intention, [55](#)
- getM_percentProductive_i_s
 - intention::Intention, [55](#)
- getM_probability_kit_observation
 - intention::Intention, [55](#)
- getM_selected_plan
 - intention::Intention, [55](#)
- getM_Si
 - intention::Intention, [55](#)
- getM_SR_i_r_s
 - intention::Intention, [55](#)
- getM_SR_i_s
 - intention::Intention, [56](#)
- getM_SR_Total
 - intention::Intention, [56](#)
- getManager
 - ontology::Ontology, [94](#)
- getMetric
 - gui::OptionFrame, [110](#)
- getMetricsChartTitle
 - gui::Chart, [22](#)
- getMnemonic
 - treecheckbox::TristateCheckBox::TristateDecorator, [128](#)
- getNumberStateRelation
 - intention::Intention, [56](#)
- getOccurence
 - ontology::Count, [35](#)
- getOccurrence
 - ontology::Exist, [45](#)
- getOntology
 - ontology::Ontology, [95](#)
- getOntologyPath
 - ontology::Ontology, [95](#)
- getOrderedList
 - intention::Intention, [56](#)
- getPartColor
 - ontology::Ontology, [95](#)
- getPartType
 - ontology::Ontology, [95](#)
- getPath
 - ontology::Ontology, [95](#)
- getPosition
 - ontology::AnyOrder, [18](#)
 - ontology::Exist, [45](#)
- getPredicate
 - ontology::Count, [35](#)
 - ontology::Exist, [45](#)
- getPreferredSize
 - gui::DrawStringPanel, [43](#)
- getReference
 - ontology::Predicate, [119](#)
- getReferenceObjectClass
 - ontology::Ontology, [95](#)
 - ontology::Predicate, [119](#)
- getRootClass
 - ontology::Ontology, [95](#)
- getSelectedObjects
 - treecheckbox::TristateCheckBox::TristateDecorator, [128](#)
- getSelectionModel
 - treecheckbox::CheckTreeManager, [27](#)
- getSeparator
 - ontology::Ontology, [96](#)
- getState
 - treecheckbox::TristateCheckBox, [125](#)
 - treecheckbox::TristateCheckBox::TristateDecorator, [128](#)
- getStateRelation
 - ontology::Ontology, [96](#)
 - ontology::Predicate, [119](#)
- getStringHead
 - ontology::Ontology, [96](#)
- getStringTail
 - ontology::Ontology, [96](#)
- getSubclasses
 - ontology::Ontology, [96](#)
- getTarget
 - ontology::Predicate, [119](#)
- getTargetObjectClass
 - ontology::Ontology, [96](#)
 - ontology::Predicate, [119](#)
- getTotalNumber
 - ontology::AnyOrder, [18](#)
 - ontology::Exist, [45](#)
- getTransferData
 - gui::PDFChartTransferable, [116](#)
- getTransferDataFlavors
 - gui::PDFChartTransferable, [116](#)
- getTreeCellRendererComponent

- treecheckbox::CheckTreeCellRenderer, 25
- Graphical User Interface, 11
- gui, 13
- gui::Chart, 18
 - Chart, 21
 - createChart_likelihood, 21
 - createChart_metrics, 21
 - createLikelihoodDataset, 21
 - createMetricsDataset, 21
 - getChartTitle, 21
 - getLegendItems, 21
 - getMetricsChartTitle, 22
 - m_colors, 22
 - m_LH_Chart_Title, 23
 - m_LH_chartPanel, 23
 - m_LH_Dataset, 23
 - m_Metric_Chart_Title, 23
 - m_Metric_chartPanel, 23
 - m_Metric_Dataset, 23
 - m_tmp_LH_ChartPanel, 23
 - m_tmp_Metric_ChartPanel, 23
 - setChartTitle, 22
 - setMetricsChartTitle, 22
 - updateChart_likelihood, 22
 - updateChart_metrics, 22
- gui::Chart::CustomRenderer, 36
 - colors, 37
 - CustomRenderer, 37
 - getItemPaint, 37
 - serialVersionUID, 37
- gui::Chart::CustomRendererLine, 37
 - colors, 38
 - CustomRendererLine, 38
 - getItemPaint, 38
 - serialVersionUID, 38
- gui::CommonGUIComponents, 29
 - chartLeftPaneColor, 31
 - chartPanelColor, 31
 - intentionColor1, 31
 - intentionColor10, 31
 - intentionColor2, 31
 - intentionColor3, 31
 - intentionColor4, 31
 - intentionColor5, 31
 - intentionColor6, 32
 - intentionColor7, 32
 - intentionColor8, 32
 - intentionColor9, 32
 - JFreeChartBackground, 32
 - JTabbedPaneColor, 32
 - m_legendFont, 32
 - m_part_green, 32
 - m_part_orange, 32
 - m_part_yellow, 32
 - m_state, 32
 - m_stateRelationFont, 32
 - m_titleFont, 32
 - menuBarColor, 32
 - state_color, 32
 - tabColor, 32
- gui::DemoPanel, 39
 - addChart, 40
 - charts, 40
 - DemoPanel, 40
 - getCharts, 40
- gui::DrawStringPanel, 42
 - anchor, 43
 - angle, 44
 - DrawStringPanel, 43
 - font, 44
 - getFont, 43
 - getPreferredSize, 43
 - paintComponent, 43
 - PREFERRED_SIZE, 44
 - rotate, 44
 - rotationAnchor, 44
 - setAnchor, 43
 - setAngle, 43
 - setFont, 43
 - setRotationAnchor, 43
 - text, 44
- gui::MainFrame, 70
 - _sd, 74
 - actionPerformed, 73
 - applyThemeToChart, 73
 - attemptExit, 73
 - buildLikelihoodPanel, 73
 - buildMetricPanel, 73
 - copyToClipboard, 73
 - createContent, 73
 - createLikelihoodChartPanel, 74
 - createMenuBar, 74
 - createMetricsChartPanel, 74
 - editorPane, 75
 - EXIT_COMMAND, 75
 - exportToCSV, 74
 - exportToPDF, 74
 - lh_Button_ActionPerformed, 74
 - likelihood_bottomRightPane, 75

- likelihood_chartContainer, 75
- likelihood_displayPanel, 75
- likelihood_stateContainer, 75
- likelihoodLeft_JTextPane, 75
- m_csvitem, 75
- m_LH_Left_Panel, 75
- m_LH_ScrollPane, 75
- m_LH_Selection_Button, 75
- m_LH_SplitPane, 75
- m_likelihood_barchart, 75
- m_likelihoodCheckTreeManager, 75
- m_likelihoodTree, 75
- m_metricCheckTreeManager, 75
- m_metrics_barchart, 75
- m_metrics_jscrollpane, 75
- m_metricsSelectionButton, 75
- m_metricTree, 75
- MainFrame, 73
- metrics_chartContainer, 76
- metrics_jsplitpane, 76
- metrics_statePane, 76
- metricsLeft_JPanel, 76
- metricsLeft_JScrollPane, 76
- metricsLeft_JTextPane, 76
- saveToCSV, 74
- serialVersionUID, 76
- showErrorMessage, 74
- updateStateRelationPanel, 74
- gui::MainFrame::DisplayMetrics, 40
 - actionPerformed, 41
- gui::OntologyChooser, 103
 - actionPerformed, 105
 - createAndShowGUI, 105
 - createImageIcon, 105
 - fc, 105
 - log, 105
 - newline, 105
 - OntologyChooser, 105
 - openButton, 105
 - saveButton, 105
- gui::OptionFrame, 106
 - add, 110
 - addComponentsToPane, 110
 - allKitsButton, 110
 - contentPane, 110
 - CREATE_WINDOW, 111
 - createAndShowGUI, 110
 - createAndUpdateConfigFile, 110
 - createButtonFromTemplate, 110
 - createPopupMenu, 110
- DEFAULT_ICON, 111
- FILE_ICON, 111
- findFilesInDirectory, 110
- getKitList, 110
- getMetric, 110
- LF_DECORATIONS, 111
- m_bool_allKits, 111
- m_browse_instance, 111
- m_browse_path, 111
- m_field_AM1, 111
- m_field_AM2, 111
- m_field_AM3, 111
- m_field_AM4, 111
- m_field_AM5, 111
- m_frame, 111
- m_instance_txt_field, 111
- m_kitComboBox, 111
- m_metric, 111
- m_planComboBox, 111
- m_planStrings, 111
- m_popupMenu, 111
- m_save_file_txt_field, 111
- m_subkitlist, 111
- m_textArea, 112
- m_validate, 112
- m_validate_button, 112
- NO_DECORATIONS, 112
- OptionFrame, 110
- PAINT_ICON, 112
- picture, 112
- readConfigFile, 110
- RIGHT_TO_LEFT, 112
- setMetric, 110
- shouldFill, 112
- shouldWeightX, 112
- singleKitButton, 112
- updateConfigFile, 110
- updateLabel, 110
- WS_DECORATIONS, 112
- gui::OptionFrame::MouseHandler, 80
 - MouseHandler, 80
 - mousePressed, 80
 - mouseReleased, 80
 - popupMenu, 80
- gui::OptionFrame::PlanComboBoxListener, 117
 - actionPerformed, 118
- gui::PDFChartTransferable, 115
 - chart, 116
 - getTransferData, 116

- getTransferDataFlavors, 116
- height, 117
- isDataFlavorSupported, 116
- PDFChartTransferable, 116
- pdfFlavor, 117
- width, 117
- writeChartAsPDF, 116
- gui::ProgressBar, 120
 - MY_MAXIMUM, 121
 - MY_MINIMUM, 121
 - pbar, 122
 - ProgressBar, 121
 - updateBar, 121
- hasProperty
 - ontology::Ontology, 97
- height
 - gui::PDFChartTransferable, 117
- hotspot
 - treecheckbox::CheckTreeManager, 27
- initializeList
 - ontology::Ontology, 97
- input
 - ontology::Ontology, 97
- insertString
 - component::IntFilter, 67
- Intention
 - intention::Intention, 52
- intention, 13
- Intention Structure, 11
- intention::Intention, 47
 - buildDetrimentalList, 52
 - getDetrimentalList, 52
 - getIndividual, 52
 - getIntentionName, 53
 - getM_am1, 53
 - getM_am2, 53
 - getM_am3, 53
 - getM_am4, 53
 - getM_am5, 53
 - getM_built_kit, 54
 - getM_found_detrimental_SR, 54
 - getM_intention_number, 54
 - getM_likelihood_observation, 54
 - getM_map_SRirs, 54
 - getM_mm1, 54
 - getM_percentComplete_i_s, 55
 - getM_percentProductive_i_s, 55
 - getM_probability_kit_observation, 55
 - getM_selected_plan, 55
 - getM_Si, 55
 - getM_SR_i_r_s, 55
 - getM_SR_i_s, 56
 - getM_SR_Total, 56
 - getNumberStateRelation, 56
 - getOrderedList, 56
 - Intention, 52
 - m_am1, 61
 - m_am2, 62
 - m_am3, 62
 - m_am4, 63
 - m_am5, 63
 - m_anyorder, 63
 - m_built_kit, 64
 - m_detrimental_list, 64
 - m_exist, 64
 - m_found_detrimental_SR, 64
 - m_individual, 64
 - m_intention_name, 64
 - m_intention_number, 64
 - m_intention_orderingConstruct_list, 64
 - m_likelihood_observation, 64
 - m_mm1, 64
 - m_number_state_relation, 65
 - m_orderedlist, 65
 - m_percentComplete_i_s, 65
 - m_percentProductive_i_s, 65
 - m_probability_kit_observation, 65
 - m_selected_plan, 65
 - m_Si, 65
 - m_SR_i_r_s, 66
 - m_SR_i_s, 66
 - m_SR_Total, 66
 - m_SRirs, 66
 - setDetrimentalList, 56
 - setExist, 57
 - setIndividual, 57
 - setIntentionName, 57
 - setM_am1, 57
 - setM_am2, 57
 - setM_am3, 58
 - setM_am4, 58
 - setM_am5, 58
 - setM_built_kit, 58
 - setM_found_detrimental_SR, 58
 - setM_intention_number, 58
 - setM_likelihood_observation, 59
 - setM_map_SRirs, 59
 - setM_mm1, 59

- setM_percentComplete_i_s, [59](#)
- setM_percentProductive_i_s, [59](#)
- setM_probability_kit_observation, [60](#)
- setM_selected_plan, [60](#)
- setM_Si, [60](#)
- setM_SR_i_r_s, [60](#)
- setM_SR_i_s, [61](#)
- setM_SR_Total, [61](#)
- setNumberStateRelation, [61](#)
- setOrderedList, [61](#)
- intention::Likelihood, [70](#)
- intention::Metric, [76](#)
 - get_AM1_Weight, [77](#)
 - get_AM2_Weight, [77](#)
 - get_AM3_Weight, [77](#)
 - get_AM4_Weight, [77](#)
 - get_AM5_Weight, [78](#)
 - get_MM1_Weight, [78](#)
 - m_AM1_weight, [79](#)
 - m_AM2_weight, [79](#)
 - m_AM3_weight, [79](#)
 - m_AM4_weight, [79](#)
 - m_AM5_weight, [79](#)
 - m_MM1_weight, [79](#)
 - set_AM1_Weight, [78](#)
 - set_AM2_Weight, [78](#)
 - set_AM3_Weight, [78](#)
 - set_AM4_Weight, [79](#)
 - set_AM5_Weight, [79](#)
 - set_MM1_Weight, [79](#)
- intentionColor1
 - gui::CommonGUIComponents, [31](#)
- intentionColor10
 - gui::CommonGUIComponents, [31](#)
- intentionColor2
 - gui::CommonGUIComponents, [31](#)
- intentionColor3
 - gui::CommonGUIComponents, [31](#)
- intentionColor4
 - gui::CommonGUIComponents, [31](#)
- intentionColor5
 - gui::CommonGUIComponents, [31](#)
- intentionColor6
 - gui::CommonGUIComponents, [32](#)
- intentionColor7
 - gui::CommonGUIComponents, [32](#)
- intentionColor8
 - gui::CommonGUIComponents, [32](#)
- intentionColor9
 - gui::CommonGUIComponents, [32](#)
- isArmed
 - treecheckbox::TristateCheckBox::TristateDecorator, [128](#)
- isDataFlavorSupported
 - gui::PDFChartTransferable, [116](#)
- isDescendant
 - treecheckbox::CheckTreeSelectionModel, [29](#)
- isEnabled
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- isFocusTraversable
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- isMac
 - tools::Configuration, [33](#)
- isPartiallySelected
 - treecheckbox::CheckTreeSelectionModel, [29](#)
- isPathSelected
 - treecheckbox::CheckTreeSelectionModel, [29](#)
- isPressed
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- isRollover
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- isSelected
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- isUnix
 - tools::Configuration, [33](#)
- isWindows
 - tools::Configuration, [33](#)
- JFreeChartBackground
 - gui::CommonGUIComponents, [32](#)
- JTabbedPaneColor
 - gui::CommonGUIComponents, [32](#)
- Launcher
 - main::Launcher, [68](#)
- LF_DECORATIONS
 - gui::OptionFrame, [111](#)
- lh_Button_ActionPerformed
 - gui::MainFrame, [74](#)
- likelihood_bottomRightPane
 - gui::MainFrame, [75](#)
- likelihood_chartContainer

- gui::MainFrame, 75
- likelihood_displayPanel
 - gui::MainFrame, 75
- likelihood_stateContainer
 - gui::MainFrame, 75
- likelihoodLeft_JTextPane
 - gui::MainFrame, 75
- loadFromFile
 - ontology::Ontology, 97
- loadOntologyFromPath
 - ontology::Ontology, 97
- log
 - gui::OntologyChooser, 105
- m_am1
 - intention::Intention, 61
- m_AM1_weight
 - intention::Metric, 79
- m_am2
 - intention::Intention, 62
- m_AM2_weight
 - intention::Metric, 79
- m_am3
 - intention::Intention, 62
- m_AM3_weight
 - intention::Metric, 79
- m_am4
 - intention::Intention, 63
- m_AM4_weight
 - intention::Metric, 79
- m_am5
 - intention::Intention, 63
- m_AM5_weight
 - intention::Metric, 79
- m_anyorder
 - intention::Intention, 63
 - ontology::OrderedList, 113
- m_bool_allKits
 - gui::OptionFrame, 111
- m_browse_instance
 - gui::OptionFrame, 111
- m_browse_path
 - gui::OptionFrame, 111
- m_built_kit
 - intention::Intention, 64
- m_colors
 - gui::Chart, 22
- m_count_list
 - ontology::AnyOrder, 18
- m_csvitem
 - gui::MainFrame, 75
- m_datafactory
 - ontology::Ontology, 101
- m_detrimental_list
 - intention::Intention, 64
- m_exist
 - intention::Intention, 64
 - ontology::OrderedList, 113
- m_field_AM1
 - gui::OptionFrame, 111
- m_field_AM2
 - gui::OptionFrame, 111
- m_field_AM3
 - gui::OptionFrame, 111
- m_field_AM4
 - gui::OptionFrame, 111
- m_field_AM5
 - gui::OptionFrame, 111
- m_found_detrimental_SR
 - intention::Intention, 64
- m_frame
 - gui::OptionFrame, 111
- m_hasCount_Occurrence
 - ontology::Ontology, 101
- m_hasIntention_Name
 - ontology::Ontology, 101
- m_hasIntention_OrderingConstruct
 - ontology::Ontology, 101
- m_hasOrderingConstruct_OrderingConstruct
 - ontology::Ontology, 101
- m_hasOrderingConstruct_Position
 - ontology::Ontology, 101
- m_hasOrderingConstruct_Predicate
 - ontology::Ontology, 101
- m_hasPredicate_ReferenceObject
 - ontology::Ontology, 101
- m_hasPredicate_TargetObject
 - ontology::Ontology, 101
- m_individual
 - intention::Intention, 64
 - ontology::AnyOrder, 18
 - ontology::Count, 35
 - ontology::Exist, 46
 - ontology::OrderedList, 113
 - ontology::Predicate, 119
- m_instance_txt_field
 - gui::OptionFrame, 111
- m_intention_list
 - ontology::Ontology, 102
- m_intention_name

- intention::Intention, 64
- m_intention_number
 - intention::Intention, 64
- m_intention_orderingConstruct_list
 - intention::Intention, 64
- m_kitComboBox
 - gui::OptionFrame, 111
- m_kitToBuild
 - ontology::Ontology, 102
- m_legendFont
 - gui::CommonGUIComponents, 32
- m_LH_Chart_Title
 - gui::Chart, 23
- m_LH_chartPanel
 - gui::Chart, 23
- m_LH_Dataset
 - gui::Chart, 23
- m_LH_Left_Panel
 - gui::MainFrame, 75
- m_LH_ScrollPane
 - gui::MainFrame, 75
- m_LH_Selection_Button
 - gui::MainFrame, 75
- m_LH_SplitPane
 - gui::MainFrame, 75
- m_likelihood_barchart
 - gui::MainFrame, 75
- m_likelihood_observation
 - intention::Intention, 64
- m_likelihoodCheckTreeManager
 - gui::MainFrame, 75
- m_likelihoodTree
 - gui::MainFrame, 75
- m_manager
 - ontology::Ontology, 102
- m_metric
 - gui::OptionFrame, 111
- m_Metric_Chart_Title
 - gui::Chart, 23
- m_Metric_chartPanel
 - gui::Chart, 23
- m_Metric_Dataset
 - gui::Chart, 23
- m_metricCheckTreeManager
 - gui::MainFrame, 75
- m_metrics_barchart
 - gui::MainFrame, 75
- m_metrics_jscrollpane
 - gui::MainFrame, 75
- m_metricsSelectionButton
 - gui::MainFrame, 75
- m_metricTree
 - gui::MainFrame, 75
- m_mm1
 - intention::Intention, 64
- m_MM1_weight
 - intention::Metric, 79
- m_number_state_relation
 - intention::Intention, 65
- m_observation_list
 - ontology::Ontology, 102
- m_occurrence
 - ontology::Count, 35
- m_ontology
 - ontology::Ontology, 102
- m_ontology_IRI
 - ontology::Ontology, 102
- m_orderedlist
 - intention::Intention, 65
- m_OS
 - tools::Configuration, 33
- m_part_green
 - gui::CommonGUIComponents, 32
- m_part_orange
 - gui::CommonGUIComponents, 32
- m_part_type_number_for_each_intention_ - list
 - ontology::Ontology, 102
- m_part_yellow
 - gui::CommonGUIComponents, 32
- m_path
 - ontology::Ontology, 102
- m_percentComplete_i_s
 - intention::Intention, 65
- m_percentProductive_i_s
 - intention::Intention, 65
- m_planComboBox
 - gui::OptionFrame, 111
- m_planStrings
 - gui::OptionFrame, 111
- m_planToBuild
 - ontology::Ontology, 102
- m_popupMenu
 - gui::OptionFrame, 111
- m_position
 - ontology::AnyOrder, 18
 - ontology::Exist, 46
- m_predicate
 - ontology::Count, 35
 - ontology::Exist, 46

- m_probability_kit_observation
 - intention::Intention, [65](#)
- m_progress_bar
 - ontology::Ontology, [102](#)
- m_progress_frame
 - ontology::Ontology, [102](#)
- m_reasoner
 - ontology::Ontology, [102](#)
- m_reference
 - ontology::Predicate, [120](#)
- m_reference_class
 - ontology::Predicate, [120](#)
- m_s_ontopath
 - ontology::Ontology, [102](#)
- m_s_rootClass
 - ontology::Ontology, [102](#)
- m_s_subClass
 - ontology::Ontology, [102](#)
- m_save_file_txt_field
 - gui::OptionFrame, [111](#)
- m_selected_plan
 - intention::Intention, [65](#)
- m_SEPARATOR
 - ontology::Ontology, [102](#)
- m_Si
 - intention::Intention, [65](#)
- m_SR_i_r_s
 - intention::Intention, [66](#)
- m_SR_i_s
 - intention::Intention, [66](#)
- m_SR_Total
 - intention::Intention, [66](#)
- m_SRirs
 - intention::Intention, [66](#)
- m_state
 - gui::CommonGUIComponents, [32](#)
- m_staterelation
 - ontology::Predicate, [120](#)
- m_stateRelationFont
 - gui::CommonGUIComponents, [32](#)
- m_subkitlist
 - gui::OptionFrame, [111](#)
- m_target
 - ontology::Predicate, [120](#)
- m_target_class
 - ontology::Predicate, [120](#)
- m_textArea
 - gui::OptionFrame, [112](#)
- m_titleFont
 - gui::CommonGUIComponents, [32](#)
- m_tmp_LH_ChartPanel
 - gui::Chart, [23](#)
- m_tmp_Metric_ChartPanel
 - gui::Chart, [23](#)
- m_total_number
 - ontology::AnyOrder, [18](#)
 - ontology::Exist, [46](#)
- m_validate
 - gui::OptionFrame, [112](#)
- m_validate_button
 - gui::OptionFrame, [112](#)
- main, [14](#)
 - main::Launcher, [69](#)
 - main::Launcher, [68](#)
 - enabler, [69](#)
 - Launcher, [68](#)
 - main, [69](#)
- MainFrame
 - gui::MainFrame, [73](#)
- matchDetrimentalStateRelationToIntention
 - ontology::Ontology, [97](#)
- matchStateRelationToIntention
 - ontology::Ontology, [97](#)
- menuBarColor
 - gui::CommonGUIComponents, [32](#)
- metrics_chartContainer
 - gui::MainFrame, [76](#)
- metrics_jsplitpane
 - gui::MainFrame, [76](#)
- metrics_statePane
 - gui::MainFrame, [76](#)
- metricsLeft_JPanel
 - gui::MainFrame, [76](#)
- metricsLeft_JScrollPane
 - gui::MainFrame, [76](#)
- metricsLeft_JTextPane
 - gui::MainFrame, [76](#)
- model
 - treecheckbox::CheckTreeSelectionModel, [29](#)
 - treecheckbox::TristateCheckBox, [126](#)
- mouseClicked
 - treecheckbox::CheckTreeManager, [27](#)
- MouseHandler
 - gui::OptionFrame::MouseHandler, [80](#)
- mousePressed
 - gui::OptionFrame::MouseHandler, [80](#)
- mouseReleased
 - gui::OptionFrame::MouseHandler, [80](#)
- MY_MAXIMUM

- gui::ProgressBar, 121
- MY_MINIMUM
 - gui::ProgressBar, 121
- newline
 - gui::OntologyChooser, 105
- nextState
 - treecheckbox::TristateCheckBox::TristateDeco, 129
- NO_DECORATIONS
 - gui::OptionFrame, 112
- occurrence
 - ontology::Exist, 46
- Ontology, 103
 - ontology::Ontology, 86
- ontology, 14
- ontology::AnyOrder, 17
 - addCountToList, 18
 - AnyOrder, 18
 - getCountList, 18
 - getIndividual, 18
 - getPosition, 18
 - getTotalNumber, 18
 - m_count_list, 18
 - m_individual, 18
 - m_position, 18
 - m_total_number, 18
 - setIndividual, 18
 - setPosition, 18
 - setTotalNumber, 18
- ontology::Count, 34
 - Count, 35
 - getIndividual, 35
 - getOccurrence, 35
 - getPredicate, 35
 - m_individual, 35
 - m_occurrence, 35
 - m_predicate, 35
 - setIndividual, 35
 - setOccurrence, 35
 - setPredicate, 35
- ontology::Exist, 44
 - Exist, 45
 - getIndividual, 45
 - getOccurrence, 45
 - getPosition, 45
 - getPredicate, 45
 - getTotalNumber, 45
 - m_individual, 46
 - m_position, 46
 - m_predicate, 46
 - m_total_number, 46
 - occurrence, 46
 - setIndividual, 46
 - setOccurrence, 46
 - setPosition, 46
 - setPredicate, 46
 - setTotalNumber, 46
- ontology::Ontology, 81
 - buildIntentionList, 86
 - buildIntentionMetricsTree, 87
 - buildIntentionTree, 87
 - buildObservationList, 87
 - buildStates, 87
 - choose, 87
 - chooseTest, 88
 - cleanDataPropertyInteger, 88
 - cleanIRI, 88
 - compute_SR_all_s, 88
 - compute_SR_i_r_s, 88
 - compute_sum_observation, 88
 - compute_sum_part_type, 89
 - computeLikelihood, 89
 - computeMetricAM1, 89
 - computeMetricAM2, 90
 - computeMetricAM3, 90
 - computeMetricAM4, 91
 - computeMetricAM5, 91
 - computeMetricMM1, 92
 - computeMetricsInformation, 92
 - computeObservationLikelihood, 93
 - computePercentComplete, 93
 - computePercentProductive, 93
 - computeProbabilityFromObservation, 94
 - computeProceduresForAM5, 94
 - getForEachIntentionTheNumberOfParts-ForEachType, 94
 - getIndividualClass, 94
 - getIndividualClassString, 94
 - getManager, 94
 - getOntology, 95
 - getOntologyPath, 95
 - getPartColor, 95
 - getPartType, 95
 - getPath, 95
 - getReferenceObjectClass, 95
 - getRootClass, 95
 - getSeparator, 96

- getStateRelation, 96
- getStringHead, 96
- getStringTail, 96
- getSubclasses, 96
- getTargetObjectClass, 96
- hasProperty, 97
- initializeList, 97
- input, 97
- loadFromFile, 97
- loadOntologyFromPath, 97
- m_datafactory, 101
- m_hasCount_Occurrence, 101
- m_hasIntention_Name, 101
- m_hasIntention_OrderingConstruct, 101
- m_hasOrderingConstruct_OrderingConstruct, 101
- m_hasOrderingConstruct_Position, 101
- m_hasOrderingConstruct_Predicate, 101
- m_hasPredicate_ReferenceObject, 101
- m_hasPredicate_TargetObject, 101
- m_intention_list, 102
- m_kitToBuild, 102
- m_manager, 102
- m_observation_list, 102
- m_ontology, 102
- m_ontology_IRI, 102
- m_part_type_number_for_each_intention_list, 102
- m_path, 102
- m_planToBuild, 102
- m_progress_bar, 102
- m_progress_frame, 102
- m_reasoner, 102
- m_s_ontopath, 102
- m_s_rootClass, 102
- m_s_subClass, 102
- m_SEPARATOR, 102
- matchDetrimentalStateRelationToIntention, 97
- matchStateRelationToIntention, 97
- Ontology, 86
- parseIntention, 98
- printProperties, 98
- readForEachIntentionTheNumberOfParts-ForEachType, 98
- readIntentionList, 98
- readObservationList, 98
- removeDuplicates, 98
- roundTwoDecimals, 99
- searchList, 99
- setDataFactory, 99
- setInstanceFilePath, 99
- setManager, 99
- setOntology, 99
- setPath, 99
- setReasoner, 100
- setRootClass, 100
- showDialogBox, 100
- sortIntentionList, 100
- updateForEachIntentionTheNumberOfPartsForEachType, 100
- updateMainFrame, 100
- updateObservationList, 101
- ontology::OnTopWithContactStateRelation, 105
- OnTopWithContactStateRelation, 106
- setName, 106
- sr_name, 106
- ontology::OrderedList, 112
- getAnyOrder, 113
- getExist, 113
- getIndividual, 113
- m_anyorder, 113
- m_exist, 113
- m_individual, 113
- OrderedList, 113
- orderedlist_l, 114
- orderedlist_name, 114
- setAnyOrder, 113
- setExist, 113
- setIndividual, 113
- ontology::OrderingConstruct, 114
- OrderingConstruct, 114
- ontology::PartiallyInStateRelation, 114
- PartiallyInStateRelation, 114
- setName, 114
- sr_name, 115
- ontology::Predicate, 118
- getIndividual, 119
- getReference, 119
- getReferenceObjectClass, 119
- getStateRelation, 119
- getTarget, 119
- getTargetObjectClass, 119
- m_individual, 119
- m_reference, 120
- m_reference_class, 120
- m_staterelation, 120
- m_target, 120
- m_target_class, 120

- Predicate, 119
- setIndividual, 119
- setReference, 119
- setReferenceObjectClass, 119
- setStateRelation, 119
- setTarget, 119
- setTargetObjectClass, 119
- ontology::UnderWithContactStateRelation, 130
- setName, 130
- sr_name, 130
- UnderWithContactStateRelation, 130
- OntologyChooser
 - gui::OntologyChooser, 105
- OnTopWithContactStateRelation
 - ontology::OnTopWithContactStateRelation, 106
- openButton
 - gui::OntologyChooser, 105
- openFile
 - textfiles::FileOperator, 47
- OptionFrame
 - gui::OptionFrame, 110
- OrderedList
 - ontology::OrderedList, 113
- orderedlist_l
 - ontology::OrderedList, 114
- orderedlist_name
 - ontology::OrderedList, 114
- OrderingConstruct
 - ontology::OrderingConstruct, 114
- other
 - treecheckbox::TristateCheckBox::TristateDecorator, 130
- PAINT_ICON
 - gui::OptionFrame, 112
- paintComponent
 - gui::DrawStringPanel, 43
- parseIntention
 - ontology::Ontology, 98
- PartiallyInStateRelation
 - ontology::PartiallyInStateRelation, 114
- pbar
 - gui::ProgressBar, 122
- PDFChartTransferable
 - gui::PDFChartTransferable, 116
- pdfFlavor
 - gui::PDFChartTransferable, 117
- picture
 - gui::OptionFrame, 112
- popupMenu
 - gui::OptionFrame::MouseHandler, 80
- Predicate
 - ontology::Predicate, 119
- PREFERRED_SIZE
 - gui::DrawStringPanel, 44
- printProperties
 - ontology::Ontology, 98
- ProgressBar
 - gui::ProgressBar, 121
- readConfigFile
 - gui::OptionFrame, 110
- readForEachIntentionTheNumberOfPartsForEachType
 - ontology::Ontology, 98
- readIntentionList
 - ontology::Ontology, 98
- readLines
 - textfiles::FileOperator, 47
- readObservationList
 - ontology::Ontology, 98
- remove
 - component::IntFilter, 67
- removeActionListener
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- removeChangeListener
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- removeDuplicates
 - ontology::Ontology, 98
- removeIntentionListener
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- removeSelectionPaths
 - treecheckbox::CheckTreeSelectionModel, 29
- replace
 - component::IntFilter, 67
- RIGHT_TO_LEFT
 - gui::OptionFrame, 112
- rotate
 - gui::DrawStringPanel, 44
- rotationAnchor
 - gui::DrawStringPanel, 44
- roundTwoDecimals
 - ontology::Ontology, 99
- saveAllKitsData

- textfiles::FileOperator, 47
- saveButton
 - gui::OntologyChooser, 105
- saveToCSV
 - gui::MainFrame, 74
- scrollPane
 - treecheckbox::TreeExample, 123
- searchList
 - ontology::Ontology, 99
- selectionModel
 - treecheckbox::CheckTreeCellRenderer, 25
 - treecheckbox::CheckTreeManager, 27
- serialVersionUID
 - gui::Chart::CustomRenderer, 37
 - gui::Chart::CustomRendererLine, 38
 - gui::MainFrame, 76
 - treecheckbox::TreeExample, 123
- set_AM1_Weight
 - intention::Metric, 78
- set_AM2_Weight
 - intention::Metric, 78
- set_AM3_Weight
 - intention::Metric, 78
- set_AM4_Weight
 - intention::Metric, 79
- set_AM5_Weight
 - intention::Metric, 79
- set_MM1_Weight
 - intention::Metric, 79
- setActionCommand
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- setAnchor
 - gui::DrawStringPanel, 43
- setAngle
 - gui::DrawStringPanel, 43
- setAnyOrder
 - ontology::OrderedList, 113
- setArmed
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- setChartTitle
 - gui::Chart, 22
- setDataFactory
 - ontology::Ontology, 99
- setDetrimentalList
 - intention::Intention, 56
- setEnabled
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- setExist
 - intention::Intention, 57
 - ontology::OrderedList, 113
- setFont
 - gui::DrawStringPanel, 43
- setGroup
 - treecheckbox::TristateCheckBox::TristateDecorator, 129
- setIndividual
 - intention::Intention, 57
 - ontology::AnyOrder, 18
 - ontology::Count, 35
 - ontology::Exist, 46
 - ontology::OrderedList, 113
 - ontology::Predicate, 119
- setInstanceFilePath
 - ontology::Ontology, 99
- setIntentionName
 - intention::Intention, 57
- setM_am1
 - intention::Intention, 57
- setM_am2
 - intention::Intention, 57
- setM_am3
 - intention::Intention, 58
- setM_am4
 - intention::Intention, 58
- setM_am5
 - intention::Intention, 58
- setM_built_kit
 - intention::Intention, 58
- setM_found_detrimental_SR
 - intention::Intention, 58
- setM_intention_number
 - intention::Intention, 58
- setM_likelihood_observation
 - intention::Intention, 59
- setM_map_SRirs
 - intention::Intention, 59
- setM_map_SRmmr
 - intention::Intention, 59
- setM_percentComplete_i_s
 - intention::Intention, 59
- setM_percentProductive_i_s
 - intention::Intention, 59
- setM_probability_kit_observation
 - intention::Intention, 60
- setM_selected_plan

- intention::Intention, [60](#)
- setM_Si
 - intention::Intention, [60](#)
- setM_SR_i_r_s
 - intention::Intention, [60](#)
- setM_SR_i_s
 - intention::Intention, [61](#)
- setM_SR_Total
 - intention::Intention, [61](#)
- setManager
 - ontology::Ontology, [99](#)
- setMetric
 - gui::OptionFrame, [110](#)
- setMetricsChartTitle
 - gui::Chart, [22](#)
- setMnemonic
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- setName
 - ontology::OnTopWithContactStateRelation, [106](#)
 - ontology::PartiallyInStateRelation, [114](#)
 - ontology::UnderWithContactStateRelation, [130](#)
- setNumberStateRelation
 - intention::Intention, [61](#)
- setOccurrence
 - ontology::Count, [35](#)
 - ontology::Exist, [46](#)
- setOntology
 - ontology::Ontology, [99](#)
- setOrderedList
 - intention::Intention, [61](#)
- setPath
 - ontology::Ontology, [99](#)
- setPosition
 - ontology::AnyOrder, [18](#)
 - ontology::Exist, [46](#)
- setPredicate
 - ontology::Count, [35](#)
 - ontology::Exist, [46](#)
- setPressed
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- setReasoner
 - ontology::Ontology, [100](#)
- setReference
 - ontology::Predicate, [119](#)
- setReferenceObjectClass
 - ontology::Predicate, [119](#)
- setRollover
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- setRootClass
 - ontology::Ontology, [100](#)
- setRotationAnchor
 - gui::DrawStringPanel, [43](#)
- setSelected
 - treecheckbox::TristateCheckBox::TristateDecorator, [129](#)
- setSelectionPaths
 - treecheckbox::CheckTreeSelectionModel, [29](#)
- setState
 - treecheckbox::TristateCheckBox, [125](#)
 - treecheckbox::TristateCheckBox::TristateDecorator, [130](#)
- setStateRelation
 - ontology::Predicate, [119](#)
- setTarget
 - ontology::Predicate, [119](#)
- setTargetObjectClass
 - ontology::Predicate, [119](#)
- setTotalNumber
 - ontology::AnyOrder, [18](#)
 - ontology::Exist, [46](#)
- shouldFill
 - gui::OptionFrame, [112](#)
- shouldWeightX
 - gui::OptionFrame, [112](#)
- showDialogBox
 - ontology::Ontology, [100](#)
- showErrorMessage
 - gui::MainFrame, [74](#)
- singleKitButton
 - gui::OptionFrame, [112](#)
- sortIntentionList
 - ontology::Ontology, [100](#)
- sr_name
 - ontology::OnTopWithContactStateRelation, [106](#)
 - ontology::PartiallyInStateRelation, [115](#)
 - ontology::UnderWithContactStateRelation, [130](#)
- src/component/IntFilter.java, [131](#)
- src/gui/Chart.java, [131](#)
- src/gui/CommonGUIComponents.java, [131](#)
- src/gui/DemoPanel.java, [132](#)
- src/gui/DrawStringPanel.java, [132](#)
- src/gui/MainFrame.java, [132](#)

- src/gui/OntologyChooser.java, 133
- src/gui/OptionFrame.java, 133
- src/gui/PDFChartTransferable.java, 133
- src/gui/ProgressBar.java, 133
- src/intention/Intention.java, 134
- src/intention/Likelihood.java, 134
- src/intention/Metric.java, 134
- src/main/Launcher.java, 134
- src/ontology/AnyOrder.java, 135
- src/ontology/Count.java, 135
- src/ontology/Exist.java, 136
- src/ontology/Ontology.java, 136
- src/ontology/OnTopWithContactStateRelation.java, 136
- src/ontology/OrderedList.java, 136
- src/ontology/OrderingConstruct.java, 137
- src/ontology/PartiallyInStateRelation.java, 137
- src/ontology/Predicate.java, 137
- src/ontology/UnderWithContactStateRelation.java, 137
- src/textfiles/FileOperator.java, 138
- src/tools/Configuration.java, 138
- src/treecheckbox/CheckTreeCellRenderer.java, 138
- src/treecheckbox/CheckTreeManager.java, 138
- src/treecheckbox/CheckTreeSelectionModel.java, 139
- src/treecheckbox/TreeExample.java, 139
- src/treecheckbox/TristateCheckBox.java, 139
- state_color
 - gui::CommonGUIComponents, 32
- tabColor
 - gui::CommonGUIComponents, 32
- test
 - component::IntFilter, 67
- text
 - gui::DrawStringPanel, 44
- textfiles, 14
- textfiles::FileOperator, 46
 - FileOperator, 47
 - openFile, 47
 - readLines, 47
 - saveAllKitsData, 47
 - translatePlanToStateRelation, 47
- toggleRemoveSelection
 - treecheckbox::CheckTreeSelectionModel, 29
- tools, 14
- tools::Configuration, 33
 - isMac, 33
 - isUnix, 33
 - isWindows, 33
 - m_OS, 33
- topPanel
 - treecheckbox::TreeExample, 123
- translatePlanToStateRelation
 - textfiles::FileOperator, 47
- tree
 - treecheckbox::CheckTreeManager, 27
 - treecheckbox::TreeExample, 123
- treecheckbox, 15
 - treecheckbox::CheckTreeCellRenderer, 24
 - checkBox, 25
 - CheckTreeCellRenderer, 25
 - delegate, 25
 - getTreeCellRendererComponent, 25
 - selectionModel, 25
 - treecheckbox::CheckTreeManager, 25
 - addChildPaths, 26
 - CheckTreeManager, 26
 - getAllCheckedPaths, 27
 - getDescendants, 27
 - getSelectionModel, 27
 - hotspot, 27
 - mouseClicked, 27
 - selectionModel, 27
 - tree, 27
 - valueChanged, 27
- treecheckbox::CheckTreeSelectionModel, 27
 - addSelectionPaths, 28
 - areSiblingsSelected, 28
 - CheckTreeSelectionModel, 28
 - isDescendant, 29
 - isPartiallySelected, 29
 - isPathSelected, 29
 - model, 29
 - removeSelectionPaths, 29
 - setSelectionPaths, 29
 - toggleRemoveSelection, 29
- treecheckbox::TreeExample, 122
 - scrollPane, 123
 - serialVersionUID, 123
 - topPanel, 123
 - tree, 123
 - TreeExample, 123
- treecheckbox::TristateCheckBox, 124
 - addMouseListener, 125
 - getState, 125

- model, 126
- setState, 125
- TriStateCheckBox, 125
- treecheckbox::TriStateCheckBox::TriStateDecorator, 126
- addActionListener, 128
- addChangeListener, 128
- addItemListener, 128
- getActionCommand, 128
- getMnemonic, 128
- getSelectedObjects, 128
- getState, 128
- isArmed, 128
- isEnabled, 129
- isFocusTraversable, 129
- isPressed, 129
- isRollover, 129
- isSelected, 129
- nextState, 129
- other, 130
- removeActionListener, 129
- removeChangeListener, 129
- removeItemListener, 129
- setActionCommand, 129
- setArmed, 129
- setEnabled, 129
- setGroup, 129
- setMnemonic, 129
- setPressed, 129
- setRollover, 129
- setSelected, 129
- setState, 130
- TriStateDecorator, 128
- TreeExample
 - treecheckbox::TreeExample, 123
- TriStateCheckBox
 - treecheckbox::TriStateCheckBox, 125
- TriStateDecorator
 - treecheckbox::TriStateCheckBox::TriStateDecorator, 128
- UnderWithContactStateRelation
 - ontology::UnderWithContactStateRelation, 130
- updateBar
 - gui::ProgressBar, 121
- updateChart_likelihood
 - gui::Chart, 22
- updateChart_metrics
 - gui::Chart, 22
- updateConfigFile
 - gui::OptionFrame, 110
- updateForEachIntentionTheNumberOfPartsForEachType
 - ontology::Ontology, 100
- updateLabel
 - gui::OptionFrame, 110
- updateMainFrame
 - ontology::Ontology, 100
- updateObservationList
 - ontology::Ontology, 101
- updateStateRelationPanel
 - gui::MainFrame, 74
- valueChanged
 - treecheckbox::CheckTreeManager, 27
- width
 - gui::PDFChartTransferable, 117
- writeChartAsPDF
 - gui::PDFChartTransferable, 116
- WS_DECORATIONS
 - gui::OptionFrame, 112