Colored Petrinets Simulator Gui Documentation

Release alpha 2

Christoph Kuhr

May 06, 2016

Contents:

CHAPTER

ONE

GUI

class gui.MainWindow.**MainWindow**(*workingDir*)

Bases: PyQt4.QtGui.QMainWindow

The MainWindow is loaded by the application main loop.

Member workingDir Working directory.

Member editors List of all editors: *gui.DiagramEditor*.

Member editorwidgets List of all editor widgets.

Member logWidget Log widget.

Member tabWidget Widget containing *editors[0]*, and the *logWidget*.

Member simulator *model.CPNSimulator* instance.

Member timer Simulation timer.

addSubnetEditor(*subnet*)

Create a subnet editor.

@param subnet: The string name of the substitution transition, name of the subnet

closeEvent(*event*)

Close window event.

Parameters **event** – *QtGui.closeEvent*.

createItemsAssignToEditor(*editor*, *subnet*, *tmpSubnets*, *importSubnet=False*)

Assign places, transition and connections to subnet editors

Parameters

- **editor** – *gui.DiagramEditor* subnet.
- **subnet** – Subnet string name.
- **tmpSubnets** – List of subnet names and corresponding items [”subnet”,[items]].
- **importSubnet** – Flag determining, whethera new net is loaded or a subnet is imported.

Return subnetConnections List of connection in subnet for later inter subnet processing.

createSubnetItemLists(*tmpSubnets*, *serPlaces*, *serTransitions*, *serConnections*)

Assign subnet to items.

Parameters

- **tmpSubnets** – List of subnet names
- **serPlaces** – List with parameters for the Place generation: [[uniqueName, name, portClone, port, [pos], [initMarking]],...].
- **serTransitions** – List with parameters for the Transition generation: [[uniqueName, name, [pos], guardExpression, subnet],...].

- **serConnections** – List with parameters for the Connection generation: [[uniqueNameSRC, uniqueNameDST, name, sourceConnector, destinationConnector],...].

Return newTmpSubnets List of subnet names and corresponding items [”subnet”,[items]].

export_Step_as_SVG()
Export Petrinet and subnets to seperate SVG files

importSubnet(*transition*)
Import subnet into editor.

Parameters **transition** – Substitution transition *model.TransitionItem*.

initNetFromFile(*editor*, *subnetList*)
Create all items for a subnet.

Parameters

- **editor** – *gui.DiagramEditor* subnet.
- **subnetList** – “subnet”,[items].

Return places, transitions, connectionsInSubnet List of visual places, List of visual Transitions, list of visualk connections.

loadNet(*editor=None*, *filename=None*, *subnet=None*, *importSubnet=False*)
Load a net with subnets or import subnets from file.

Parameters

- **editor** – *gui.DiagramEditor* to fill with the subnet. *editors[0]* if editor=none
- **filename** – The filename that is used when a new net is loaded.
- **subnet** – Subnet string name.
- **importSubnet** – Flag determining, whethera new net is loaded or a subnet is imported.

lookupSubstitutionConnectors(*subnetConnections*)
Lookup and substitute connectors.

Parameters **subnetConnections** – List of connection in subnet for inter subnet processing.

newNet(*init=False*)
Create a new Petrinet.

This method resets the main editor widget. It deletes the previously loaded Petrinet, as well as any hierarchical subnet.

@param init: Load on startup?

openSubnet(*transition*)
Open subnet editor.

Parameters **transition** – Substitution transition *model.TransitionItem*.

saveNet()
Save Petrinet with respect to subnets to XML file.

setArc(*editor*, *srcConnector*, *dstConnector*, *itemName*)
Create arcs based on new uniqueNames.

Parameters

- **editor** – *gui.DiagramEditor* subnet.
- **srcConnector** – Source *model.AbstractItem.Connector*.
- **dstConnector** – Destination *model.AbstractItem.Connector*.
- **itemName** – Annotation text.

Return newArc Newly created *model.ArcItem*.

gui.MainWindow.**bin_**(*QTextStream*) → QTextStream

gui.MainWindow.**hex_**(*QTextStream*) → QTextStream

gui.MainWindow.**oct_**(*QTextStream*) → QTextStream

class gui.DiagramEditor.**DiagramEditor**(*mainWindow=None*, *parent=None*, *workingDir=''*, *subnet=None*)

Bases: PyQt4.QtGui.QWidget

Editor widget, containing the *libraryModelView*, the *colorSetListView* and the *gui.DiagramScene*.

Member mainWindow *gui.MainWindow*. Main application window.

Member parent Parent widget.

Member workingDir Working directory.

Member subnet Name of visualized subnet, also window title.

Member colorListItems Defined colors.

Member libItems Visual library items, CPN elements.

Member portConnections List of connections between portClone port places and substitution transitions.

Member visualPlaces List of all *model.PlaceItem's shown in this 'gui.DiagramScene*.

Member visualTransitions List of all *model.TransitionItem's shown in this 'gui.DiagramScene*.

Member visualConnectionList List of all *model.ArcItem's shown in this 'gui.DiagramScene*.

Member libraryModel Library model for visual CPN elements.

Member mouseScreenPos Recorded mouse position.

Member diagramScene Contained *gui.DiagramScene*.

Member diagramView *gui.DiagramView* controlling *gui.DiagramScene*.

Member startedArc Status, determining whether the creation of a connection has started.

Member showCanvasInfos Status variable for activation of the tooltip *model.AbstractItem.DescriptionCanvas* s.

createIcon(*nodeType*)

Create Icons for *libraryModel*.

Parameters **nodeType** – Type of icon to create.

defineNewToken()

Define a new color set.

deleteArc(*editor*, *connection*)

Delete arc.

Parameters

- **editor** – *gui.DiagramEditor* containing the *connection*.
- **connection** – *model.ArcItem* for deletion.

deleteItems(*editor*, *items*)

Delete items.

Parameters

- **editor** – *gui.DiagramEditor* containing the *items*.
- **items** – List with items (selection) to delete.

deletePlace(*editor*, *place*, *portClone=False*)
Delete places, port places and port clone places.

Parameters

- **editor** – *gui.DiagramEditor* containing the *place*.
- **place** – *model.PlaceItem* for deletion.
- **portClone** – Flag determining whether a port clone place shall be deleted.

deleteSubnet(*editor*)
Recursivly delete subnet editor and their content.

Parameters **editor** – Root *gui.DiagramEditor* for deletion.

deleteToken(*editor*, *token*)
Delete a token.

Parameters

- **editor** – *gui.DiagramEditor* containing the *place*.
- **token** – *model.TokenItem* for deletion.

deleteTransition(*editor*, *transition*)
Delete transitions, substitution transitions and their subnets.

Parameters

- **editor** – *gui.DiagramEditor* containing the *transition*.
- **transition** – *model.TransitionItem* for deletion.

keyPressEvent(*event*)
Callback method, when a key is pressed on the keyboard.

Parameters **event** – *QtGui.keyPressEvent*.

newToken(*item*)
Callback method, when *add Token* is clicked in *colorListView*.

Parameters **item** – *colorListView* item.

sceneMouseMoveEvent(*event*)
Catch sceneMouseMoveEvent during arc creation.

Parameters **event** – *QtGui.sceneMouseMoveEvent*.

sceneMouseReleaseEvent(*event*)
Catch *QtGui.sceneMouseReleaseEvent* on arc completion.

Parameters **event** – *QtGui.sceneMouseReleaseEvent*.

setPortConnection(*portPlaceClone*, *transition*)
Set a connection between a port place clone and a substitution transition.

Parameters

- **portPlaceClone** – Port place clone *model.PlaceItem*.
- **transition** – Substitutaion transition *model.TransitionItem*.

setTokenForPlace(*place*, *actualMarking*)
Show the tokens contained in place.

Parameters

- **place** – *model.PlaceItem* to show the *model.TokenItem* for.
- **actualMarking** – SNAKES Marking to determine the tokens for *place*.

setTokens(*actualMarking*)
Set tokens for *actualMarking*.

Parameters **actualMarking** – SNAKES Marking to determine the tokens for *place*.

shortcutCreateNode()
Create CPN elements with keyboard shortcuts.

startArc(*nodeItem*)
Start an arc conneciton.

Parameters **nodeItem** – *model.AbstractItem* source element.

validConnection()
Callback method, when a valid arc connection was created.

wheelEvent(*event*)
Callback method, when the mouse wheel is used.

Parameters **event** – *QtGui.wheelEvent*.

class gui.DiagramEditor.**EditorGraphicsView**(*parent=None*)
Bases: PyQt4.QtGui.QGraphicsView

Viewport for the *gui.DiagramScene* s.

It controls the zooming feature and drag and drop operations from the item library.

Member parent Parent editor widget.

Member scaleFactor Zooming factor.

dragEnterEvent(*event*)
Callback method, when an icon is dragged from the *libraryModelView* to *gui.diagramScene*.

Parameters **event** – *QtGui.dragEnterEvent*.

dragMoveEvent(*event*)
Callback method, when an icon is moved over the *gui.diagramScene*.

Parameters **event** – *QtGui.dragMoveEvent*.

dropEvent(*event*)
Callback method, when an icon is dropped on the *gui.diagramScene*.

Parameters **event** – *QtGui.dropEvent*.

validDrop()
Callback method, when the drop on *gui.DiagramScene* was valid.

wheelEvent(*event*)
Callback method, when the mouse wheel is used.

Parameters **event** – *QtGui.wheelEvent*.

gui.DiagramEditor.**bin_**(*QTextStream*) → QTextStream

gui.DiagramEditor.**hex_**(*QTextStream*) → QTextStream

gui.DiagramEditor.**oct_**(*QTextStream*) → QTextStream

class gui.DiagramScene.**DiagramScene**(*parent=None*)
Bases: PyQt4.QtGui.QGraphicsScene

Drawing Area.

Member editor Parent *gui.DiagramEditor*.

Member hovering Flag determining, whether hovering is happening. (Workaround, since hovering is not forwarded).

mouseMoveEvent(*event*)
Forward mouseMoveEvent during arc creation and reimplement hovering of *model.AbstractItem.Connector*.

Parameters **event** – *QtGui.mouseMoveEvent*.

mouseReleaseEvent(*event*)
Forward mouseReleaseEvent during arc creation.

Parameters **event** – *QtGui.mouseReleaseEvent*.

gui.DiagramScene.**bin_**(*QTextStream*) → QTextStream

gui.DiagramScene.**hex_**(*QTextStream*) → QTextStream

gui.DiagramScene.**oct_**(*QTextStream*) → QTextStream

class gui.LibraryModel.**LibraryModel**(*parent=None*)
Bases: PyQt4.QtGui.QStandardItemModel

Abstract class for drag and drop support

mimeData(*idxs*)

mimeTypes()

gui.LibraryModel.**bin_**(*QTextStream*) → QTextStream

gui.LibraryModel.**hex_**(*QTextStream*) → QTextStream

gui.LibraryModel.**oct_**(*QTextStream*) → QTextStream

class gui.NameDialog.**NameDialog**(*parent=None*, *item=None*, *title='Unnamed'*, *default='a'*)
Bases: PyQt4.QtGui.QDialog

cancel()

getItem()

Return self.item

getName()

Return self.lineEdit.text()

ok()

gui.NameDialog.**bin_**(*QTextStream*) → QTextStream

gui.NameDialog.**hex_**(*QTextStream*) → QTextStream

gui.NameDialog.**oct_**(*QTextStream*) → QTextStream

class gui.ParameterDialog.**ParameterDialog**(*node*, *parent=None*)
Bases: PyQt4.QtGui.QDialog

OK()

gui.ParameterDialog.**bin_**(*QTextStream*) → QTextStream

gui.ParameterDialog.**hex_**(*QTextStream*) → QTextStream

gui.ParameterDialog.**oct_**(*QTextStream*) → QTextStream

class gui.SubnetDialog.**SubnetDialog**(*mainWindow*, *parent=None*, *superNet=None*, *subnet=None*)
Bases: PyQt4.QtGui.QDialog

Dialog for subnet creation

cancel()

closeEvent(*event*)

Parameters **event** –

gui.SubnetDialog.**bin_**(*QTextStream*) → QTextStream

gui.SubnetDialog.**hex_**(*QTextStream*) → QTextStream

gui.SubnetDialog.**oct_**(*QTextStream*) → QTextStream

class gui.TokenDialog.**TokenDialog**(*parent=None*, *title='Choose Colour and Amount of the Token'*)

Bases: PyQt4.QtGui.QDialog

Dialog to choose a color for new token

cancel()

getCountToken()

Return self.countToken

getInitMarking()

Return self.initMarking

getListEntry()

Return self.listEntry

ok()

setCountToken(*value*)

Set token count.

Parameters value –

setInitMarking(*value*)

Set initial Marking.

Parameters value –

setListEntry()

Add new color to list.

gui.TokenDialog.**bin_**(*QTextStream*) → QTextStream

gui.TokenDialog.**hex_**(*QTextStream*) → QTextStream

gui.TokenDialog.**oct_**(*QTextStream*) → QTextStream

CHAPTER

TWO

MODEL

class model.CPNSimulator.**CPNSimulator**(*mainWindow*)

Bases: object

CPN Simulator.

Member mainWindow *gui.MainWindow*. Main application window.

Member net SNAKES Colored Petrinet.

Member markingHistory SNAKES Marking stored for every ‘ simulationStep‘.

Member simulationStep Steps calculated for the Petrinet *net*

Member displayStep Step displayed in editors.

Member simulatorSpeed The speed with which the simulator progresses.

Member enabledTransitions Number of enabled transitions.

Member uniqueNameBase Unique integer for the creation of new CPN elements.

Member initialMarking SNAKES Marking at step 0.

Member colourSets Set of defined string colors.

Member connectionList List of all *model.ArcItem* s, except port substitution transition connections.

Member transitions All *model.TransitionItem* except substitution transitions.

Member places All *model.PlaceItem* except port places.

Member subnets List of all subnets.

__init__(*mainWindow*)

Create CPN Simulator.

Parameters **mainWindow** – *gui.MainWindow*. Main application window.

back2beginning()

Return to step 0.

backStep()

Go one step back in history.

checkTranistionActivation(*transition*, *mode*, *currentStep*)

Check whether a SNAKES transition is activated.

Parameters

- **transition** – SNAKES transition to check.
- **mode** – SNAKES mode to check for activation.
- **currentStep** – Step for which to calculate the activation

Return activated Activated True or False.

checkTransitionEnabled(*transition*, *mode*, *currentStep*, *activated*, *transitions2Fire*)
Check whether a SNAKES transition is enabled.

Parameters

- **transition** – SNAKES transition to check.
- **mode** – SNAKES mode to check for activation.
- **currentStep** – Step for which to calculate the enabling.
- **activated** – Flag that determines, whether *transition* is activated.
- **transitions2Fire** – List of SNAKES transitions enabled to fire.

Return enabled Enabled True or False.

defineNewColour(*colName*)
Define a new Color.

Parameters **colName** – String color.

fireEnabledTransitions(*currentStep*)
Fire transitions in list *transitions2Fire*.

Parameters **currentStep** – Step for which to calculate the firing.

forward2lastStep()
Go forward to last step in history, given at ‘ simulationStep‘.

forwardStep()
Go step forward in history or claculate new step.

getActualMarking(*actualMarking*)
Process status of visual net, depending on *actualMarking*.

Parameters **actualMarking** – SNAKES Marking.

resetSimulator(*init=False*)
Reset simulator history to step 0.

Parameters **init** – Initialization on startup/load

setNetName()
Set the name of the *net*.

startSim()
Start simulator with the speed chosen with the radio edit.

stopSim()
Stop simulator.

tokenAdded()
Called when a *model.TokenItem* was added to the *net*.

model.CPNSimulator.**bin_**(*QTextStream*) → QTextStream

model.CPNSimulator.**hex_**(*QTextStream*) → QTextStream

model.CPNSimulator.**oct_**(*QTextStream*) → QTextStream

class model.AbstractItem.**AbstractItem**(*parent=None*)
Bases: PyQt4.QtGui.QGraphicsItem

Base class of the CPN elements transition and place.

Member parent *gui.DiagramEditor*. Editor to show in.

Member tokens List of *model.TokenItem* s, if inheriting class is *model.PlaceItem* else None

Member name Name of the CPN element.

Member posCallbacks List of callback functions, to calculate position changes.

Member connectorList List of 20 *Connector* s

Member planeMap Dictionary->Set(), mapping the relative orientation of an any other *model.AbstractItem* in the containing editor.

Member connectorMap Dictionary->List(), mapping the relative orientation of the 20 *model.AbstractItem.Connector* s.

Member nodeType Determining the type of the item at creation time.

Member superNet The super net is used to determine the editor for port place clones.

Member label *QtGui.QGraphicsTextItem*, visual representation of the *model.AbstractItem* s name.

Member descCanvas *gui.AbstractItem.DescriptionCanvas*, showing detailed information.

__init__(*parent=None*)
Create abstract item.

Parameters **parent** – *gui.DiagramEditor*. Editor to show in.

checkItem(*item*, *orientation*)
Check *item* is of propper type.

Parameters

- **item** – *model.AbstractItem* to lookup.
- **orientation** – Orientation to assign, if type check is passed.

createItem(*editor*, *name='Untitled'*, *nodeType='undefined'*)
Create the typed item.

Parameters

- **editor** – *gui.DiagramEditor*. Editor to show in.
- **name** – Name of the CPN element.
- **nodeType** – Determining the type of the item at creation time.

Return w, h Width and height of CPN element, depending on the label length.

deleteItemLocal()
Capture delete event and call editor delete function.

findItemsInPlanes()
Finds the orientation of any other item in the *gui.DiagramScene*.

mouseDoubleClickEvent(*event*)
Edit visual name on *QtGui.mouseDoubleClickEvent*.

Parameters **event** – *QtGui.mouseDoubleClickEvent*.

renameElement()
Capture rename event and call virtual function of *model.PlaceItem* or *model.TrasnitionItem*.

class model.AbstractItem.**Connector**(*parent*, *idx*)
Bases: PyQt4.QtGui.QGraphicsRectItem

Connector for visual arcs.

A connector is a socket, to or from which a visual arc may be connected.

Member parent Parent port place.

Member orientation Orientation relative to its parent: "N","NE","E","SE","S","SW","W","NW".

Member idx Index assinged from parent.

Member connectionArc Reference to the arc connected to this Connector.

Member posCallbacks List of callback functions, to calculate position changes.

Member position Position relative to parent.

__init__(*parent*, *idx*)
Create a connector.

Parameters

- **parent** – Parent AbstractItem.
- **idx** – Absolut number in the parent list of connectors.

hoverEnterEvent(*event*)
Make connector visible on hoverEnterEvent.

Parameters **event** – hoverEnterEvent

hoverLeaveEvent(*event*)
Make connector invisible on hoverLeaveEvent.

Parameters **event** – hoverLeaveEvent

itemChange(*change*, *value*)
Item position has changed, calculate new position.

Parameters

- **change** – Change event.
- **value** – QtCore.QPointF().

Return value QtCore.QPointF(x, y) or super(Connector, self).itemChange(change, value).

mousePressEvent(*event*)
Capture QtGui.mousePressEvent with **Shift-Key** modifier.

Parameters **event** – QtGui.mousePressEvent

class model.AbstractItem.**DescriptionCanvas**(*parent=None*)
Bases: PyQt4.QtGui.QGraphicsRectItem

Description area for tokens, exceptions and modes. Toggle with **Ctrl**+**M**.

In this canvas label, detailed information about the CPN element is shown. Firstly the unique name used by the simulator is shown, followed by the visual name, that is assigned by the user. The latter does not have to be unique.

Member parent Parent CPN element.

Member label Visual representation of the information about the CPN element.

Member text Unique name of parent CPN element.

Member visibility Switch for the visibility of the description canvas.

__init__(*parent=None*)
Create description canvas.

Parameters **parent** – Abstract CPN element.

setCanvasString(*infoString*)
Set info string, that appended to the first line.

Parameters **infoString** – Info string, may contain line breaks.

setVisibility(*visible*)
Set visibility of DescriptionCanvas.

Parameters **visible** – True if visible, False if invisible.

model.AbstractItem.**bin_**(*QTextStream*) → QTextStream

model.AbstractItem.**hex_**(*QTextStream*) → QTextStream

model.AbstractItem.**oct_**(*QTextStream*) → QTextStream

class model.PlaceItem.**PlaceItem**(*editor*, *name*, *position*, *initMarking=[]*, *uniqueName='p0'*, *port=None*, *loadFromFile=False*, *portDirection=None*, *portClone=None*, *superNet=None*, *subnet=None*)

Bases: PyQt4.QtGui.QGraphicsEllipseItem, *model.AbstractItem.AbstractItem*

CPN Transition element.

Member editor *gui.DiagramEditor*. Editor to show in.

Member name Name of the CPN element assigned by the user.

Member subnet Name of the subnet this transition is contained in.

Member position Position of this item in the *gui.DiagramScene*.

Member port *model.PortItem* if this is a port place, else None.

Member portClone A port place in the super net or None.

Member portDirection Direction of the port, if this is a port place.

Member uniqueName Unique name of this transition used by the simulator.

Member place SNAKES place node with id *uniqueName*, except portClones.

Member initMarking List of initial tokens present in this place.

Member tokens List (deque) of *model.TokenItem* s in this place.

Member toolTipString Information string containing the tokens present in this place.

__init__(*editor*, *name*, *position*, *initMarking=[]*, *uniqueName='p0'*, *port=None*, *loadFromFile=False*, *portDirection=None*, *portClone=None*, *superNet=None*, *subnet=None*)

Create and initialize place item.

Parameters

- **editor** – *gui.DiagramEditor*. Editor to show in.
- **name** – Name of the CPN element assigned by the user.
- **position** – Position of this item in the *gui.DiagramScene*.
- **initMarking** – List of initial tokens present in this place.
- **uniqueName** – Unique name of this transition used by the simulator.
- **port** – *model.PortItem* if this is a port place, else None.
- **loadFromFile** – Flag determining, whether the place is loaded from file.
- **portDirection** – Direction of the port, if this is a port place.
- **portClone** – A port place in the super net or None.
- **superNe** – The super net is used to determine the editor for port place clones.
- **subnet** – Name of the subnet this transition is contained in.

addToken()

Method to add a *model.TokenItem* to this place

contextMenuEvent(*event*)

Generate Context menu on context menu event.

Parameters **event** – QContextMenuEvent.

deleteItemLocal()

Capture delete event and call editor delete function.

mouseMoveEvent(*event*)

Prevents the movement of this item, when connections are drawn.

Parameters **event** – *QtGui.mouseMoveEvent*

mousePressEvent(*event*)
Prevents the movement of this item, when connections are drawn.

Parameters **event** – *QtGui.mousePressEvent*

mouseReleaseEvent(*event*)
Forward *QtGui.mouseReleaseEvent*.

Parameters **event** – *QtGui.mouseReleaseEvent*

newTokenValue()
Accept new token.

renameModifications(*name*)
Make neccessary modification and renaming.

Parameters **name** – Name of the CPN element assigned by the user.

setPort(*loadFromFile=False*)
Create a new *model.PortItem* or edit the existing port.

Parameters **loadFromFile** – Flag determining, whether the place is loaded from file.

setPortDiag()
Callback function to modify the port.

stackTokens()
Method to order the visual stacking of different tokens.

model.PlaceItem.**bin_**(*QTextStream*) → QTextStream

model.PlaceItem.**hex_**(*QTextStream*) → QTextStream

model.PlaceItem.**oct_**(*QTextStream*) → QTextStream

class model.TransitionItem.**TransitionItem**(*editor*, *name*, *position*, *guardExpression=None*, *uniqueName='t0'*, *loadFromFile=False*, *substitutionTransition=False*, *subnet=None*)
Bases: PyQt4.QtGui.QGraphicsRectItem, *model.AbstractItem.AbstractItem*

CPN Transition element.

Member editor *gui.DiagramEditor*. Editor to show in.

Member name Name of the CPN element assigned by the user.

Member substitutionTransition Flag indication a substitutaion transition.

Member uniqueName Unique name of this transition used by the simulator.

Member transition SNAKES transition node with id *uniqueName*, only if *substitutionTransition* is *False* else None.

Member subnet Name of the subnet this transition is contained in.

Member position Position of this item in the *gui.DiagramScene*.

Member enabled Flag determining if this transition is enabled.

Member guardExpression The guard expression for this transition.

Member subnetBorder Second border indicating a substitution transition.

Member exceptionString Exception string containing simulator errors.

Member modeString Mode string containing the enabled modes for this transition

__init__(*editor*, *name*, *position*, *guardExpression=None*, *uniqueName='t0'*, *loadFromFile=False*, *substitutionTransition=False*, *subnet=None*)
Create transition item.

Parameters

- **editor** – *gui.DiagramEditor*. Editor to show in.

- **name** – Name of the CPN element assigned by the user.
- **position** – Position of this item in the *gui.DiagramScene*.
- **guardExpression** – The guard expression for this transition.
- **uniqueName** – Unique name of this transition used by the simulator.
- **loadFromFile** – Flag determining, whether the transition is loaded from file.
- **substitutionTransition** – Flag indication a substitutaion transition.
- **subnet** – Name of the subnet this transition is contained in.

acceptEditGuard()
Apply modified guard expression, after *gui.NameDialog*.

contextMenuEvent(*event*)
Generate Context menu on context menu event.

Parameters **event** – QContextMenuEvent.

deleteItemLocal()
Capture delete event and call editor delete function.

editGuard()
Callback function to modify the guard expression.

importSubnet()
Forward menu action *Import Subnet* to *gui.DiagramEditor*.

initTransition(*loadFromFile*)
Initialize transition.

Parameters **loadFromFile** – Flag determining, whether the transition is loaded from file.

mouseMoveEvent(*event*)
Prevents the movement of this item, when connections are drawn.

Parameters **event** – *QtGui.mouseMoveEvent*

mousePressEvent(*event*)
Prevents the movement of this item, when connections are drawn.

Parameters **event** – *QtGui.mousePressEvent*

mouseReleaseEvent(*event*)
Forward *QtGui.mouseReleaseEvent*.

Parameters **event** – *QtGui.mouseReleaseEvent*

openSubnet()
Forward menu action *Open Subnet* to *gui.DiagramEditor*.

renameModifications(*name*)
Make neccessary modification and renaming.

Parameters **name** – Name of the CPN element assigned by the user.

setInfoString(*stringInfo*)
Set the string for the *gui.AbstractItem.DescritionCanvas*

Parameters **stringInfo** – String appended to the first line of its *gui.AbstractItem.DescritionCanvas*.

model.TransitionItem.**bin_**(*QTextStream*) → QTextStream

model.TransitionItem.**hex_**(*QTextStream*) → QTextStream

model.TransitionItem.**oct_**(*QTextStream*) → QTextStream

class model.ArcItem.**ArcItem**(*editor*, *srcConnector*, *dstConnector*, *name='undefined'*, *isPortConnection=False*)

Bases: PyQt4.QtGui.QGraphicsItem

CPN arc connection element.

Member editor *gui.DiagramEditor*. Editor to show in.

Member srcConnector Source *model.AbstractItem.Connector*.

Member dstConnector Destination *model.AbstractItem.Connector*.

Member isPortConnection Flag that determines, whether the arc represents the connection between a substitution transition and a port place..

Member name Expression of the CPN arc assigned by the user.

Member arcDefined Flag that determines, whether an arc creation was successful (internal).

Member variable SNAKES Variable.

Member expression SNAKES Expression.

Member pos1 Point of origin of the arc.

Member pos2 Point of destination of the arc.

Member arrowPolygon *QtGui.QPolygonF* showing the direction of the arc.

__init__(*editor*, *srcConnector*, *dstConnector*, *name='undefined'*, *isPortConnection=False*)

Parameters

- **editor** – *gui.DiagramEditor*. Editor to show in.
- **srcConnector** – Source *model.AbstractItem.Connector*.
- **dstConnector** – Destination *model.AbstractItem.Connector*.
- **name** – Expression of the CPN arc assigned by the user.
- **isPortConnection** – Flag that determines, whether the arc represents the connection between a substitution transition and a port place..

checkForExpression(*text*)

Check whether the annotation is intended to be an expression or a variable.

Parameters **text** – Annotation text.

Return ret True/False.

deleteItemLocal()

Capture delete event and call editor delete function.

getName()

Return the annotation of this arc.

multiInput(*multiArcAnnotations*)

Register a multi input arc in the simulator.

Parameters **multiArcAnnotations** – SNAKES MultiArc annotation.

rename()

Rename arc and apply changes to simulator and visual representation.

setArcAnnotation(*annotationText=None*)

Finalize arc creation and set annotation.

Parameters **annotationText** – String arc annotation.

Return self.arcDefined Flag that determines, whether an arc creation was successful (internal).

setBeginPos(*pos1*)
Callback method to keep *pos1* up to date.

Parameters **pos1** – *QtCore.QPointF()*.

setDestination(*dstConnector*)
Sets the destination *model.AbstractItem.Connector*.

Parameters **dstConnector** – Destination *model.AbstractItem.Connector*.

setEndPos(*endpos*)
Callback method to keep *pos2* up to date.

Parameters **endpos** – *QtCore.QPointF()*.

setName(*name*)
Set the annotation of this arc.

Parameters **name** – New annotation.

setPolygon()
Calculate position and rotation of the arc arrow head.

singleInput(*variableExpression*)
Register a single input arc in the simulator.

Parameters **variableExpression** – SNAKES Variable or Expression.

singleOutput(*variableExpression*)
Register a multi input arc in the simulator.

Parameters **variableExpression** – SNAKES Variable or Expression.

class model.ArcItem.**LineItem**(*parent*)
Bases: PyQt4.QtGui.QGraphicsLineItem

Visual representation of the line itself.

Member parent Parent *model.ArcItem*.

__init__(*parent*)
Create line.

Parameters **parent** – Parent *model.ArcItem*.

mouseDoubleClickEvent(*event*)
Edit arc annotation on *QtGui.mouseDoubleClickEvent*.

Parameters **event** – *QtGui.mouseDoubleClickEvent*.

model.ArcItem.**bin_**(*QTextStream*) → QTextStream

model.ArcItem.**hex_**(*QTextStream*) → QTextStream

model.ArcItem.**oct_**(*QTextStream*) → QTextStream

class model.TokenItem.**TokenItem**(*editor*, *token*, *count*, *qpos*, *parent=None*)
Bases: PyQt4.QtGui.QGraphicsEllipseItem

A String Token.

Member editor Referenced editor.

Member countToken Number of tokens.

Member countTokenLabel Visible representation of the number of tokens.

Member token String value of token, shown in tooltip

__init__(*editor*, *token*, *count*, *qpos*, *parent=None*)
Create a token.

Parameters

- **editor** – DiagramEditor. Editor to show in.
- **token** – Token value.
- **count** – Number of Tokens to create.
- **qpos** – Parent top right position.
- **parent=None** – Parent Place Element

contextMenuEvent(*event*)
Generate Context menu on context menu event.

Parameters **event** – QContextMenuEvent.

deleteItemLocal()
Capture delete event and call editor delete function.

setCountToken(*count*)
Token count, shown in green circle.

Parameters **count** – Number to show.

model.TokenItem.**bin_**(*QTextStream*) → QTextStream

model.TokenItem.**hex_**(*QTextStream*) → QTextStream

model.TokenItem.**oct_**(*QTextStream*) → QTextStream

class model.PortItem.**PortItem**(*direction*, *parent=None*)
Bases: PyQt4.QtGui.QGraphicsEllipseItem

Port place indicator.

The label indicates the port direction of the port place.

Member direction Port direction: (i)nput, (o)utput, (io) bidirectional.

Member parent Parent port place.

Member label Visual representation of direction.

__init__(*direction*, *parent=None*)
Create a port

Parameters

- **direction** – Port direction: (i)nput, (o)utput, (io) bidirectional.
- **parent** – Parent port place.

contextMenuEvent(*event*)
Generate Context menu on context menu event.

Parameters **event** – QContextMenuEvent.

editPort()
Edit port direction.

getDirection()
Return direction of the port: “i”, “o”, “io”.

itemChange(*change*, *value*)
Item position has changed, calculate new position.

Parameters

- **change** – Change value.
- **value** – QtCore.QPointF().

Return value QtCore.QPointF(x, y) or super(PortItem, self).itemChange(change, value).

setDirection(*direction*)
Set direction of the port.

Parameters **direction** – Direction of port: “i”, “o”, “io”.

setPort()
Is this really neccessary?

model.PortItem.**bin_**(*QTextStream*) → QTextStream

model.PortItem.**hex_**(*QTextStream*) → QTextStream

model.PortItem.**oct_**(*QTextStream*) → QTextStream

CHAPTER

THREE

INOUT

class inout.XMLIO.**XMLIO**(*simulator*, *rootElementName=''*)

Bases: object

XML Input and Output

loadNet(*filename*)

Parse XML file and prepare data for object creation.

Parameters **filename** – Filepath to XML file which shall be loaded.

Return [netName, subnets, serConnections, serPlaces, serTransitions] A list containing lists for object creation.

netToXML(*subnet*, *placesS*, *transitionsS*, *connectionsS*)

Save Colored Petrinet **Subnet** to XML tree.

Parameters

- **subnet** – Subnet name.
- **placesS** – Place contained in *subnet*.
- **transitionsS** – Transitions contained in *subnet*.
- **connectionsS** – Connections contained in *subnet*.

saveLog(*logList*)

Save Log Entries to XML tree.

Parameters **logList** – List of log entries from the log widget.

saveNet(*filename*)

Save XML tree to file.

Parameters **filename** – Filepath where the XML tree is saved.

inout.XMLIO.**bin_**(*QTextStream*) → QTextStream

inout.XMLIO.**hex_**(*QTextStream*) → QTextStream

inout.XMLIO.**oct_**(*QTextStream*) → QTextStream

CHAPTER

FOUR

INDICES AND TABLES

- genindex
- modindex
- search

g

i

m