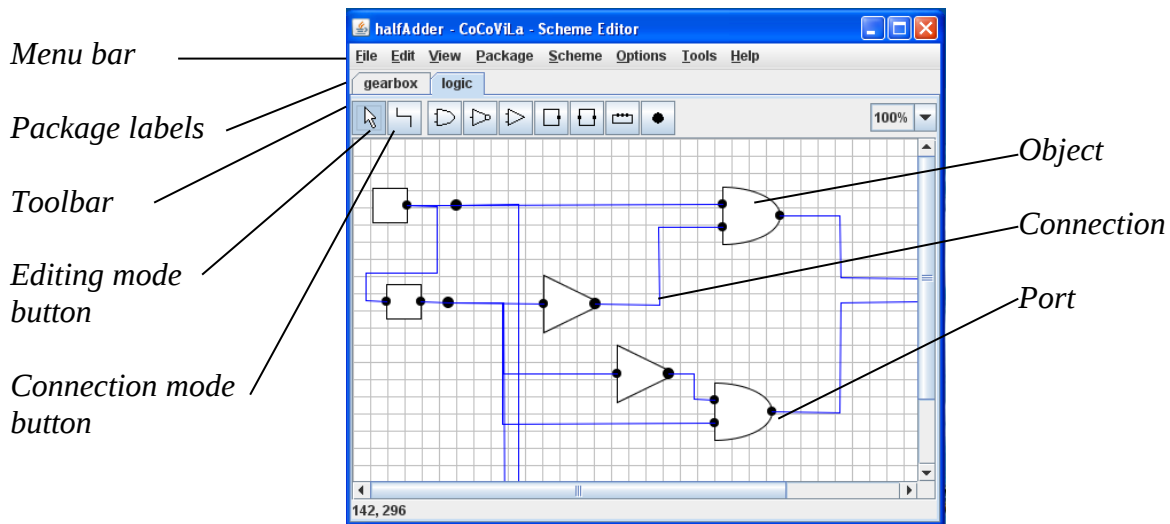# Using Scheme Editor

The Scheme Editor is used as a visual environment for specifying models by drawing schemes and solving problems on the models. To solve a problem, first a package must be loaded and, thereafter a scheme that specifies a model has to be built or opened from a package folder. A package may contain ready-made schemes for solving problems. The simplest way to solve problems in CoCoViLa is to use available schemes.

Menu bar

Package labels

Toolbar

Editing mode button

Connection mode button

Object

Connection

Port

The figure shows Scheme Editor window with a `logic` package loaded and selected and an opened `halfAdder` scheme.

Solving problems on available schemes

A scheme may contain complete information for solving a problem, but usually some input values have to be added or changed. To solve a problem on a prepared scheme one has to perform the following steps:

- Find and open a package and a scheme
- Give input values
- Select a suitable mode and fulfill the computations.
- Store a scheme with new values, if needed.

Remark: If the checkbox *Propagate* in the *Scheme* menu has been checked, then the computed values are stored in the scheme, and nothing will be computed again, even if the `run` command is given again. Therefore, to repeat the computations, one has to reload the scheme.

Opening a package

Use
Package -> Load
browse, select the package description file (with suffix `.xml`) and click *Open*.

Selecting a scheme

Use
File -> Load Scheme
Browse, select the scheme file (with suffix `.syn`) and click *Open*.

Giving input values

One can give values to parameters of components of the scheme, change and delete values of the components.

Set a scheme in the editing mode by clicking on (the leftmost button on the toolbar).
Right-click on a component image and select *Properties*.
A pop-up window will open with fields for the object name and for parameters of the object.
Add/change values as needed, complying to the given types of parameters, shown at the fields.
Click *Apply*.
Close the pop-up window, by clicking *Close*.
OK button applies changes and closes the window.

Solving a problem on a scheme

Use
Scheme -> Run.

If the computed values should be propagated to the specification and shown in the scheme, the checkbox *Propagate* in the *Scheme* menu must be checked.

Using the *Specification* window

There is a way for solving problems by opening the specification window and using its menu. This provides several options for computing.

Use
Scheme -> Specification
to open the specification window.

For simple computations use
Specification -> Compute all.
A program window opens and the synthesized program will be visible in it. This window contains a *Propagate* checkbox for propagating computed values to the scheme. If you propagate the values, see the remark in the beginning of *Solving problems on available schemes*.

Use
Compile & Run in the *Program* window to perform the computations.

For computing with a goal, first, specify a goal (see Using the Properties window)
and use
Specification -> Compute goal

Clicking *Compile & Run* in the *Program* tab will perform the computations.

Saving a scheme

Use
File -> Save Scheme As ...
Give/select the name of the scheme in the pop-up window and click *Save*.

If the scheme is saved when its name is not changed, then use
 File -> Save Scheme
and click *Save*.

Reloading a scheme

Use
File -> Reload Scheme.

Developing a scheme

A scheme is developed by selecting and loading a package, creating a new scheme, filling the scheme with objects and connecting the objects through their ports. The objects of the scheme have properties that can be edited through pop-up windows.

Loading a package

Use
Package -> Load
browse, select the package description file (with suffix `.xml`) and click *Open*.

Creating a new scheme

A development of a new scheme starts from empty Scheme Editor window that you get using
Edit -> Clear All

Saving, loading and reloading a scheme

To save a scheme, use
File -> Save Scheme As ...
Give/select the name of the scheme in the pop-up window and click *Save*.

If the scheme is saved when its name is not changed, then use
 File -> Save Scheme
and click *Save*.

To open a scheme, use
File -> Load Scheme
Browse, select the scheme file (with suffix `.syn`) and click *Open*.

To reload a scheme, use
File -> Reload Scheme.

Selecting a mode

The scheme editor can be in editing mode or in connecting mode.

To select editing mode, click ⌖.

To select connecting mode, click ⌐.

Adding an object

In the editing mode, select a component from the toolbar by clicking on the button of the component. Then move the cursor in the place where the object must be and click.

Connecting objects

In the connecting mode, click on a port to be connected, move the cursor to another port to be connected and click.

If you wish to connect ports by a broken line, click at each breaking point, when moving the cursor.

Using object's Properties window

To open the Properties window
double-click or right-click on the object and click *Properties* in the pop-up menu.

To edit values of a parameters
edit the values in the parameters' fields and click *Apply*.

To specify a *goal*
check the corresponding fields as inputs and outputs of a goal and click *Apply*.

Reshaping a scheme

A scheme can be reshaped in the editing mode:

Objects can be moved by dragging.
Connection lines can be reshaped by dragging their breaking points.
Objects can be reshaped by dragging their corners when an object has been selected by clicking on it.
Part of a scheme can be selected by constraining it by means of a rectangle.

Selected part of a scheme can be reshaped as one object.

Using menus

The Scheme Editor supports a number of useful functions accessible through its menus. The functionality of often used menus is described in the present section. Some of the functions have been used/described already in other parts of this document.

File

Saving a scheme:
File -> Save Scheme As ...
Give/select the name of the scheme in the pop-up window and click *Save*.

Saving a scheme with existing name:
File -> Save Scheme
and click *Save*.

Loading a scheme:
File -> Load Scheme
Browse, select the scheme file (with suffix `.syn`) and click *Open*.

Reloading a scheme:
File -> Reload Scheme.

Deleting a scheme:
File -> Delete Scheme.

View

This menu enables one to customize the view of a scheme by checking the following checkboxes:

To turn the grid on or off, use:
View->Grid

To show or hide ports of an object, use:
View->Ports

To display names of objects, use:
View->Object names.

Package

Loading a package
Package -> load
browse, select the package description file (with suffix `.xml`) and click *Open*.

Reloading a package
Package-> Reload

Getting information about a loaded package
Package -> Info

Closing the activated package
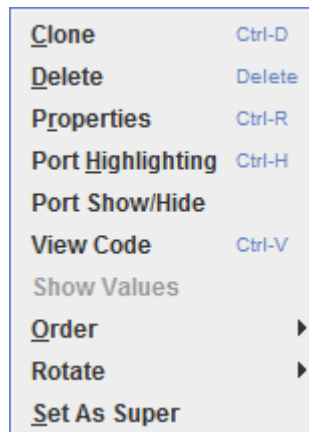*Package -> Close*

Closing all packages
Package -> Close All

Opening from a list of recently used packages
Package -> Recent
click on the name of required package.

Object pop-up menu

Object pop-up menu is called by right-clicking on an object in a scheme.

Clone
Duplicates an object, copying all values of given variables and setting a new name for the new object.

Delete
Deletes an object from a scheme. All connections to other objects are also removed.

Properties
Opens Properties window (see Using object's Properties window).

Port Highlighting
Highlights ports of an object.

Port Show/Hide
Shows or hides ports of an object.

View Code

Opens an editor for an underlying Java class of an object. Custom editor can be set-up in the Tools->Settings->Editor option.

Show Values

After running a program, displays a dialog with values of computed variables.

Order

Objects can overlap in a scheme. This option enables user to send an object to the front or to the back.

Rotate

Enables user to rotate an object for a predefined or an arbitrary angle.

Set As Super

Sets a superclass of a scheme's Java class and specification. Having a superclass, scheme's metaclass inherits all its functionality. There can only be one superclass in a scheme.