Class Editor

Creating and modifying packages with Class Editor

Introduction

Class Editor is used for creating visual languages for different problem domains. A CoCoViLa package consists of a package description in XML format (e.g., mypackage.xml), bitmaps for all visual classes of the package (to be shown on the tool bar in the Scheme Editor), and Java classes associated with the visual classes. Example packages distributed with CoCoViLa are situated in the *packages* subdirectory.

Files with type .java or bitmap files (with type .gif or .png) can be created in any convenient way, using some Java programming tool or text editor for Java files and a bitmap editor to create a bitmap.

Creating a package

1. Open the Class Editor.
2. Use File – Create Package to create a package. Enter the name and description of the package. You will be prompted with a dialog to choose where the newly created package will be saved. Typically, you should save it under a new directory under the packages directory.
3. Construct a visual class.
4. Export it into the package.
5. Repeat steps 3 and 4.

Constructing a visual class

1. Draw an image (a shape) of the class, using the drawing tools of the Class Editor.
2. Add ports to the image by clicking on the ***Add Port*** button and then on the image, and specify interface variable bound to the port and its type in the port pop-up window.
3. Add class fields (component fields that will be visible on the scheme when right-clicking on the object and selecting properties from menu).
4. Specify the bounding box that defines an area where the image will register mouse clicks in Scheme Editor. To do this, click on the bounding box button and then cover the image with the bounding box.
5. Specify class properties and export the visual class by using commands File - Export - To Package. Choose the package you have created for this class.

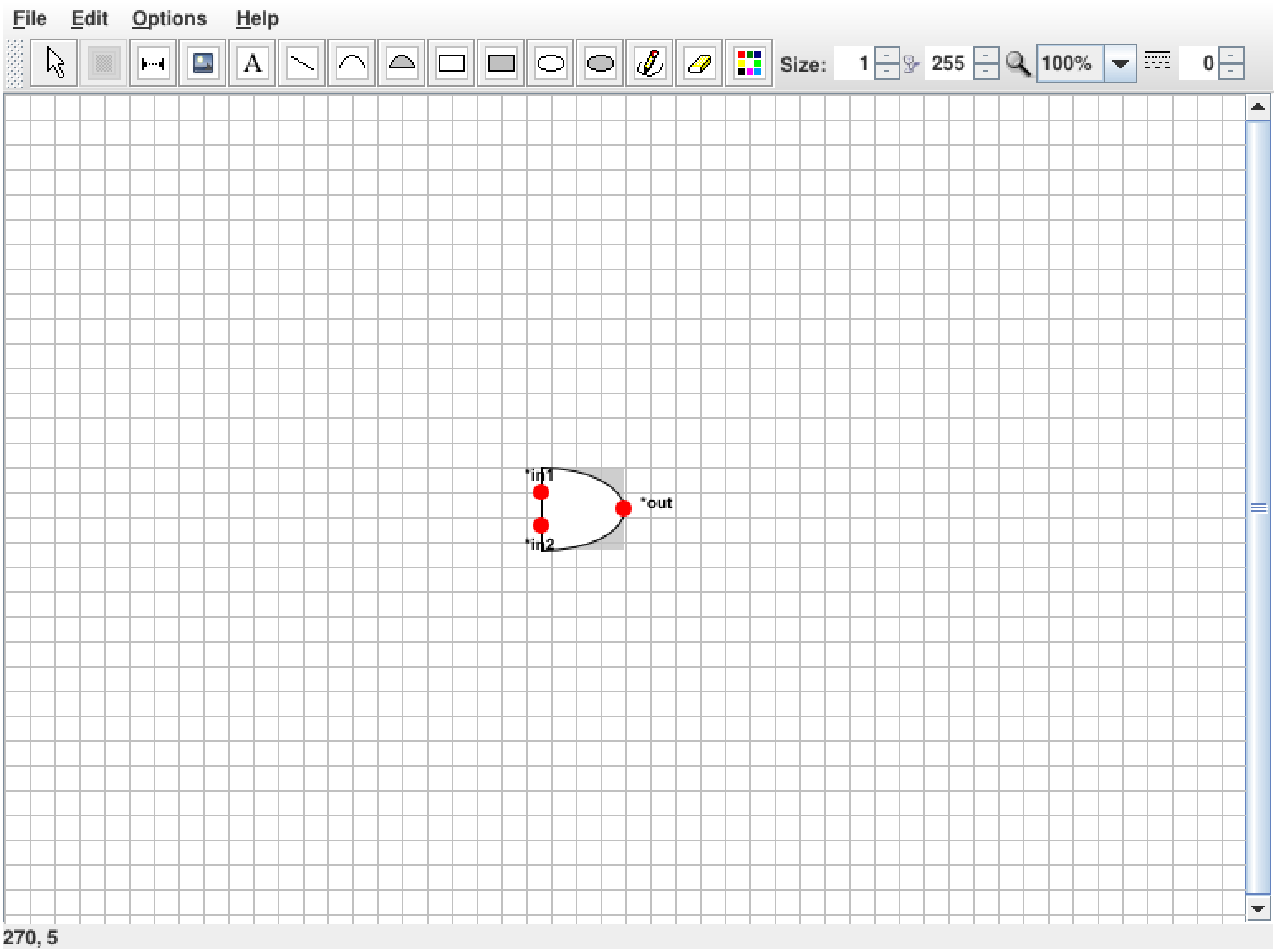
Note that right now, the Class Editor does not support drawing graphics for ports. Graphics for fields are supported partially. However the package format itself allows it, so graphics for open and closed ports and known and unknown fields can be given by directly editing the xml source of a package, but they will be lost when importing from package and exporting back to package.

Deleting and importing classes

1. Delete a class from package: File - Delete (This command requires a package name and a class name.)
2. Import a class from a package: File – Import
3. To save the imported class use File - Export - To Package in the same way as for a new visual class.

Detailed description of a Class Editor menus and tools

Class Editor allows to create visual representations of the visual classes and specify class properties like its name, description, fields and ports. In addition to editing classes some package modification can be done. For example, creating a package and exportig a visual class into packege. Existing visual class can be modified after importing it from package, and exporting it back to package.



Main window of the Class Editor consists of menu bar, tool bar and drawing area. Right click on the object brings out object menu.

Menu bar

File menu

Exporting visual class into a separate XML file.

File – Export – As Class

This function is useful when you want to save the class your working with, but do not want to export it into any package yet. It allows to create XML file with a single class in it. It can be accessed by importing it from package (you have to choose the file you saved your class in as a package description file). This will only create XML, but the Java class has to be created separately.

Exporting class into a package

File – Export – To Package

This is a file menu function that is most commonly used. When your class is ready you can export it into a package. If you did not choose a package using File – Select Package before you will be asked to select one now.

Exporting window graphics

File – Export – Window Graphics

Allows to export window graphics (see **Exporting window graphics in Scheme Editor**).

Importing visual class from package

File – Import – From Package

In order to modify a class in your package your working on or reuse a class from some other already existing package, it has to be imported into the Class Editor and exported into your package once the package is ready (see **Exporting class into package**).

Deleting class from package

File – Delete From Package

Deleting class from package means deleting any class from any package (not necessarily the one your working with). To do that, first, a package and second, a class have to be chosen. You can choose a class by selecting it from *Delete Class* dialog. You can also delete associated Java class, by choosing a *Delete Java class* option.

Creating a package

File – Create Package

Opens a popup window asking for Package name and description. After that a directory for the new package needs to be chosen.

Selecting a package

File – Select Package

Opens a dialog that lets you choose the directory containing package files and select package description (.xml) file. Selected file becomes an active package description. Active package description and its location will be shown on the title bar.

Printing an image of a visual class

File – Print

Opens a print dialog.

Exit Class Editor

File – Exit

Closes Class Editor window.

Edit menu

Select all objects in the drawing area

Edit – Select All

Keybord shortcut Ctrl-a

Delete objects

Edit – Clear All

Deletes objects from the drawing area without removing class properties.

Switch grid on and off

Edit – Grid

Switches grid of the drawing area on and off. Default settings for grid (both for Class Editor and Scheme Editor) can be defined in CoCoViLa settings dialog (Options - Settings)

Edit class properties

Edit – Class Properties

Opens class properties window that allows to define the following properties for the class:

- *Class Name* – name of the class
- *Class Description* – description of the class that will be shown as a tool tip in the Scheme Editor.
- *Class Icon* - file name of the icon that will be shown in the tool bar in the Scheme Editor. You can choose an icon by clicking on the button next to the icon name area, and selecting an icon created before (See **tips on creating a class icon**). Default value for this property is *default.gif* that will show a default icon in the tool bar.
- *Class is Relation* – a check box, when checked determines that the class image can be used as a connection between two ports.
- *Class Fields* – table containing fields defined for the class. You can add a new field with ***Add New Field*** button and remove a field with ***Delete Selected Field*** button. When adding a field its name and type have to be defined. As an option, a default value for the field can be

defined.

Fields can have complex graphics in Scheme Editor, but it is not yet completely supported by Class Editor. When importing a class with complex field graphics most of the graphics will be lost.

Options menu

Change settings

Options – Settings

Opens a CoCoViLa settings menu that allows to change values, such as whether to show grid or debug information. These settings will be applied for both – Class Editor and Scheme Editor.

Change look and feel of the Class Editor

Options – L'n'F

This option lets you change the look and feel for the Class Editor.

Help menu

Help – Docs

In theory should get you to a CoCoViLa documentation.

Help – License

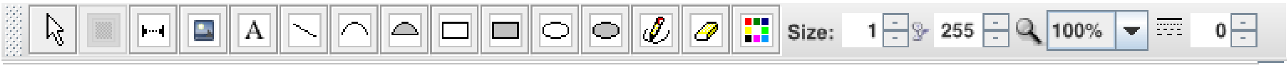
Shows licensing information.

Help – About

Shows an outdated information about CoCoViLa.

Tool bar

Tool bar includes tools for drawing visual representation for the class.



Main tools

↖	Select	When this tool is selected you can select objects by clicking on them or dragging your mouce over the object. You can then delete, define additional options or move the selected object.
▣	Bounding Box	Button for defining a bounding box – the area in which the mouce clicks will be detected in Scheme Editor. A class cannot be exported into a package without defining a bounding box first. Each class can only have one bounding box. To use this button again, existing bounding box has to be deleted.

├┤	Add Port	Allows to add ports to a class. Clicking on the image when this button is selected opens *Define Port Properties* window. In order to add a new port – its name (*Port Name*) and its type (*Port Type*), has to be defined. In addtion port can be: *Area Connected – see ...* *Strict Port – see ...* *Multi Port – see ...* After the port is added a red dot will be placed in the drawing area. You can move ports around by clicking S*elect* button and selecting a port you want to move. To change properties for selected port, you have to right click on it – this will open *Define Port Properties* window. Ports can have complex graphics in Scheme Editor, but it is not yet completly supported by Class Editor. When importing a class with complex port graphics, most of the graphics will be lost
🖼	Insert Image	This button allows you to use existing image as part of the visual class. As the image you select will be copied into your package directory, you have to choose a package before using this tool.
A	Insert Text	Inserts a text object into a selected location. After clicking on the drawing area with this button selected *Text Dialog* is opened. *Text Dialog* allows to set properties for text (e.g., font, size and color). You have to insert some text *Text to be displayed area*, otherwise text object will be created but it will not be visible. You can assotiate text objects with class fields, this will show a field's value in Scheme Editor. In order do to that, insert text object in a proper place in the image by using an ***Insert Text*** button and write ******fieldname*** (replace ***fieldname*** with your actual field name, e.g., *diameter) into the *Text to be displayed* area.
╲	Draw Line	Tool for drawing a line.
◠	Draw Arc	Tool for drawing a non-filled arc.
◓	Draw Filled Arc	Tool for drawing a filled arc.
□	Draw Rectangle	Tool for drawgin a non-filled rectange.
■	Draw Filled Rectangle	Tool for drawing a filled rectange.

⬭	Draw Oval	Tool for drawing a non-filled oval.
⬬	Draw Filled Oval	Tool for drawing a filled oval.
✎	Freehand drawing	Tool for a freehand drawing.
▱	Rubber	Deletes a complete object in case of, for example, text, line, arc, rectange or part of it, in case of a freehand drawing.
▦	Choose Color	Tool for choosing a color for objects.

Additional Options

These options have to be used together with main tool (exept zoom tool, which is applied to whole drawing area). It is also possible to change already selected objects by clicking on the object or choosing objects with mouse after selecting a select tool.

Size option

Sets the line witdh or point size of the selected object. This option can be used together Line tool, Freehand drawing and unfilled objects (Arc, Rectangle, Oval). Minimal value is 1 and maximal value is 10. Default size is 1.

Object transparency

Sets the object transparency. Minimal value is 1 (almost transparent) and maximal value is 255 (non-transparent).

Zoom percentage

This option is applied to the whole drawing area. Minimal value is 10% and maximal value is 400%. Default zoom of the drawing area is 100%.

Dash style

Sets the dash style.