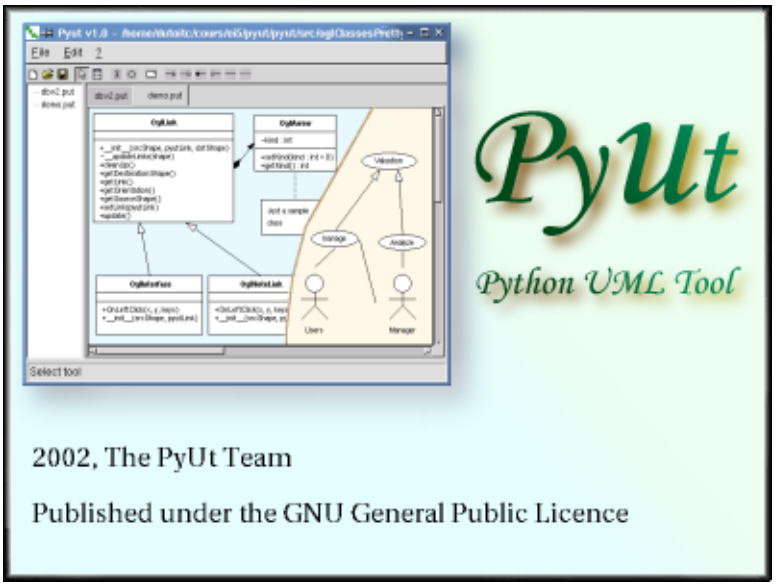
PyUT1.1 User documentation

Ecole d'Ingénieurs
du Canton de Vaud

July 17, 2002



Contents

Introduction

This document presents *PyUt* (which means Python UML tool) and is organized as follow : the first part will help you to install *PyUt*. Next section is a little introduction to UML 1.3 class and use cases diagrams. In the third section, you will be introduced to the user interface and it will be followed by a sample in the last section.

PyUt installation

Prerequisites

Windows

You have two alternatives on Windows, download the executable file which is an installer (PyutSetupXXX.exe) or work with sources (or CVS files). First option will require nothing to be preinstalled. For the second alternative, you will have to install *Python 2.2*[1] and *wxPython 2.3.2.1*[2].

Linux

For Linux (or Unix) platform, you will need to have *Python 2.2* and *wxPython 2.3.2.1* installed. For download location, see Windows prerequisites.

Installation

Windows

You will have the choice to either use the source code or the installer for the Windows platform. To use the source code directly, please read the paragraph below named "Generic install". The installer which you can download on *PyUt*'s web site is self-extracting : you'll simply need to execute the .exe, tell where to install *PyUt* and start working with it.

Linux

See "Generic install" below.

Generic install

For the generic install, download the *PyUt* source files (.tar.gz) and decompress them using : tar xvzf distrib.tar.gz. You can use WinZip (or Cygwin) for Windows platform.

CVS

You can download the current development sources with the two following steps :

- cvs -d :pserver:anonymous@cvs.pyut.sf.net:/cvsroot/pyut login
- cvs -z3 -d :pserver:anonymous@cvs.pyut.sf.net :/cvsroot/pyut co pyut

To run *PyUt*, go in the directory $PYUTPATH/src [3] and type python pyut.pyw in a console or double-click on it in Windows.

[1] http://www.python.org
[2] http://www.wxpython.org
[3] Where '$PYUTPATH' is the installation directory

UML reminder

UML stands for *Unified Modeling Language*. It's a standard notation to modelize problems and may be considered like the reference in object modelisation. Following diagrams have been found on the Rational Rose web site[4]. For a full UML reminder or to see the actual UML norm, see http://www.uml.org.

Class diagrams

A class diagram consists in a set of entities representing classes, notes and others things. More information is given in figure 1.

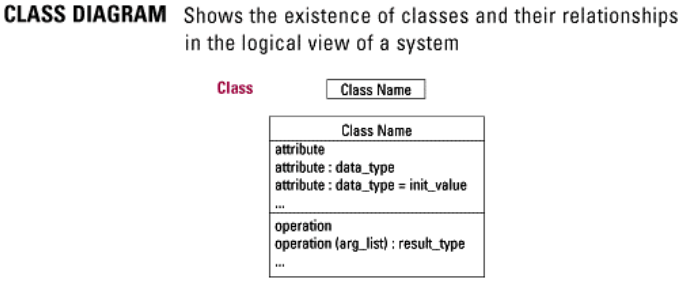


Figure 1: Class diagram reminder

A class entity consists of the class name, attributes and operations (methods). The different kinds of associations are given in figure 2 on the next page.

Use cases diagrams

A use cases diagram represents the system requirements. This explain what the system should be able to provide and how it will interact with the different actors. Description is given in figure 3 on the following page.

[4] http://www.rational.com

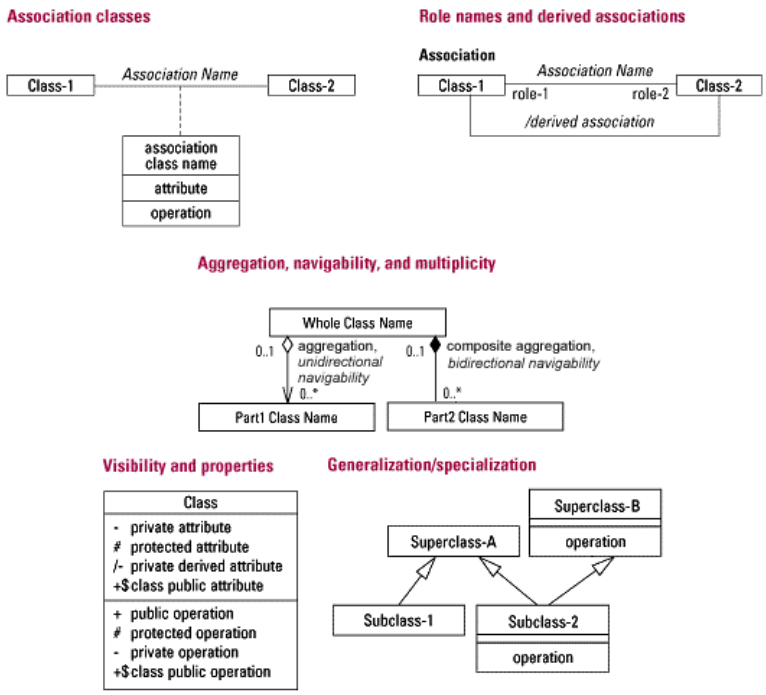


Figure 2: Class diagram reminder

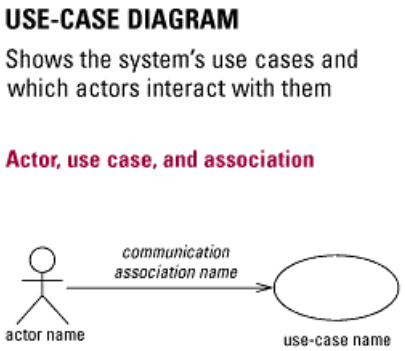


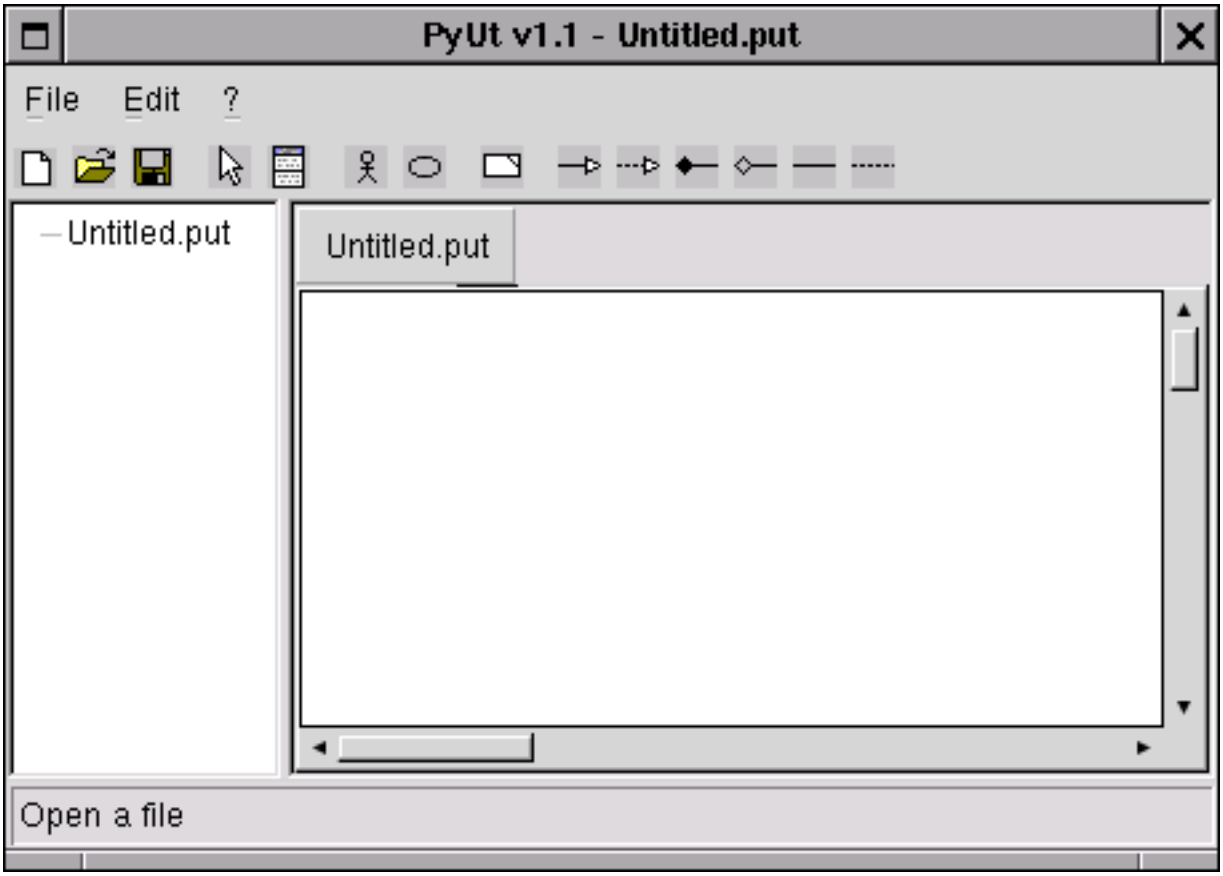
Figure 3: Use cases diagram reminder

Interface

This section describes the *PyUt* Interface and how to use it.

Graphical interface

When launched, *PyUt* will look like that :



On the left side, you have a ”project” tree to see which files are currently opened. The menu gives you access to the basic actions like import/export, cut/copy/paste, ...

Menus

File menu

Here are the elements of the file menu :

New Start a new class diagram

Open Open a file, load class diagram from a file

Save Save current diagram to a file; if a filename has already been specified, it will use it

Save as Save the current diagram to a file, and ask the filename to use

Import/Export Import/export datas, using plugins

Properties Diagram properties and preferences (language, ...)

Print setup Display the print setup dialog box

Print preview Preview before printing

Print Print the current diagram

1-5 A list of the five last files opened

Exit Exit the program

Edit menu

Cut Cut the selected classes

Copy Copy the selected classes

Paste Paste classes from the clipboard

Add Pyut Hierarchy Import an introspection of *PyUt* data structures.

Add Ogl Hierarchy Import an introspection of graphic structures.

Help menu

Index Display the help index

Check for newer version Check for newer version on Internet

Web site Launch *PyUt*'s web site (`http://pyut.sf.net`)

About Pyut Display the *PyUt* about box which give some usual informations.

Toolbar

PyUt's toolbar is on the top of the main window. Here is the signification of the buttons :

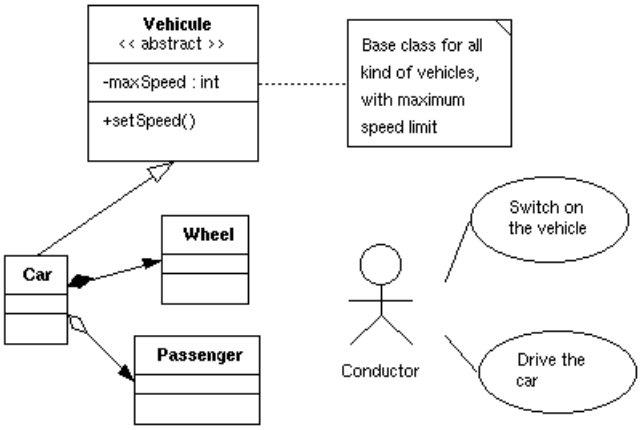
1. Begin a new diagram
2. Open a file

3. Save the current diagram
4. Selection tool
5. Insert a class
6. Insert an actor
7. Insert a use case
8. Insert a note
9. Insert an inheritance link
10. Insert a realisation link
11. Insert a composition link
12. Insert an agregation link
13. Insert an association link
14. Insert a note link

If you stay one second on a button with the mouse pointer, a little help appears. If you just move the mouse pointer on a button, a tip is given in the statusbar (bottom of the screen).

Guided sample

This section describes how to make a little mixed diagram with classes, use cases, actors and notes. Here is the diagram that we're going to make :

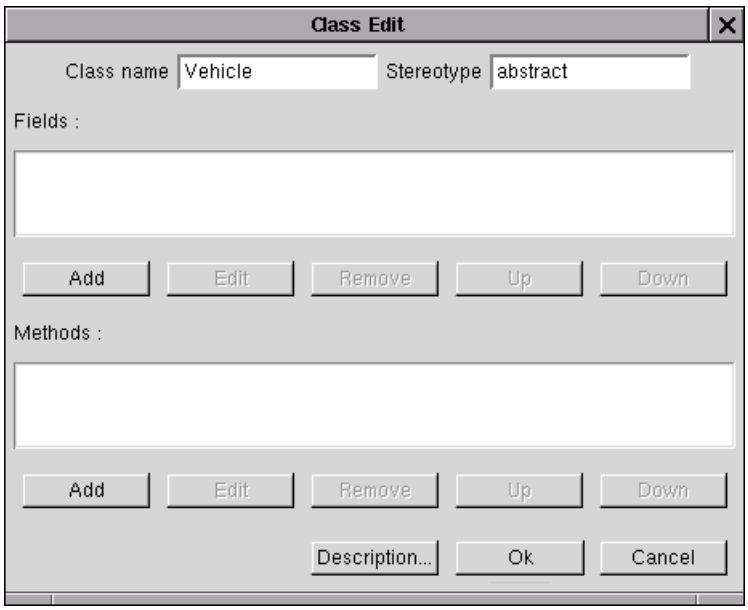


On the beginning, there was... nothing

Start with a blank diagram (File → New or launch *PyUt*)

Class insertion

First step will be to add the four classes to the diagram. Click on the class icon on the toolbar and place the Vehicle class in the top left corner. Now you should see the class editor dialog, like this :



Set the class name Vehicle and the stereotype abstract. Now we are going to set the class fields. To do this, click on the first Add button and enter your field's name, type and default value (for the example, maxSpeed - int). Just click Ok to come back to the class editor. Now add the method, just like we do it for fields. You may also add a description to your class with the button Description.... Once you have entered all the parameters, click Ok to leave the class editor.
Now you should see your first class shining on your diagram. Just repeat the same steps to place the three other classes.

Note insertion

Ok, now we are ready to insert a note to describe the Vehicle class. Click on the note tool and place the note on the right of our considered class. Enter the note text and press Ok. You will see that you can't read the whole text in the note if the text is too long. Just resize the note to see all text.

Actor and use case insertion

We are now ready to place the actor and use cases to describe the interactions between users and the system. Just click the actor tool, place one and give him a sweet little name. Add two use cases on the right of the actor, with a little description of what the actor may do.

Links

In this section, we are going to make the links between our different components :
Class Car inherit from class Vehicle :

- Click on the inheritance link button in the toolbar
- Click on the class Car, then click on the class Vehicle

Now we have our first relationship.
By the same way, insert the next links :

- Composition link from Car to Wheel.
- Agregation link from Car to Passenger class.
- Note link between Vehicle and the note
- Association link between Conductor and the two use cases.

We can redefine or remove a link parameters by clicking the link.
Now we have a basic UML diagram. We can save it, print it, export it, or modify the different component by double-clicking on them.

Saving...

You can now save your diagram using the default *PyUt* file format .put or use a plugin to export your diagram. Supported plugins are :

- Import/Export XML
- Import XMI
- Import/Export Python (Reverse/Generation)
- Import/Export Java (Reverse/Generation)
- Export C++ (Generation)
- Export BMP, JPG

The end

You did it ! We saw that creating an UML class diagram with *PyUt* is easy !