

PyUt
Python UML Tool
Cahier des charges

Laurent Burgbacher
EI5, eivd

12 décembre 2001

Table des matières

1	Informations générales	2
2	Cahier des charges	2
2.1	Possibilités	2
2.1.1	Général	2
2.1.2	Utilisation	2
2.2	Langage et environnement	3
2.2.1	Langage	3
2.2.2	Librairie graphique	3
3	Outils	3
3.1	CVS	3
3.2	Sourceforge	4
3.3	L ^A T _E X	4
4	Responsabilités	4
5	Groupes	4
5.1	Structure de données	4
5.2	wxOGL	4
5.3	Interface graphique	5

Résumé

PyUt est un petit éditeur UML graphique qui permet l'impression et l'export de diagrammes de classe sous différents formats. Ce document présente le cahier des charges du projet.

1 Informations générales

Nom du projet PyUt - Python UML Tool

Site de référence <http://pyut.sf.net>

Sources du projet Disponible par CVS (cf 3.1 page suivante)

Client J. Zuber <jzuber@aicos.com>

Membres du groupe

- Laurent Burgbacher <lb@alawa.ch>
- Nicolas Dubois <n_dub@altavista.com>
- Cédric Dutoit <dutoitc@vtxnet.ch>
- Nabil Hamadi <hamadi12@yahoo.fr>
- Dève Roux <droux@eivd.ch>
- Philippe Waelti <pwaelti@urbanet.ch>

2 Cahier des charges

2.1 Possibilités

2.1.1 Général

PyUt permet de dessiner des diagrammes de classe UML. Il présente une interface graphique de type programme de dessin. L'édition des méthodes et champs des classes se fait par l'intermédiaires de formulaires adaptés.

Les formats d'export supportés sont XML (format de données natif de PyUt) et postscript (avec possibilité d'ajouter plus tard d'autres formats).

2.1.2 Utilisation

La fenêtre principale propose une barre d'outils, contenant les icônes standard Fichier (nouveau, ouvrir, sauver), et probablement d'autres, selon les résultats des premiers tests pratiques qui seront faits. Il y a également une barre d'outils flottante, qui présente les outils propres à UML :

- Création de
 - Classe
 - Liens (héritage, composition, aggrégation, association...)

- Note
- Sélection d'objet

Une classe peut être redimensionnée comme un rectangle dans un programme de dessin vectoriel : 8 points (angles + milieu des côtés) peuvent être glissés. Une classe peut être déplacée par simple clické-glissé. Les liens restent toujours attachés aux classes, et se déplacent donc également.

Un double-click sur une classe ou un lien ouvre un formulaire permettant l'édition des attributs de l'objet.

2.2 Langage et environnement

2.2.1 Langage

PyUt est développé en Python 2.1[1]

Les caractéristiques principales de ce langage sont :

- Portable (linux, unix, win, mac)
- Interprété
- Orienté objet
- Dynamiquement typé
- Types de haut niveau natifs
 - Liste, pile
 - Dictionnaire (table de hachage)
- Support du paradigme fonctionnel
- Grande bibliothèque standard couvrant entre autres le réseau, XML, l'accès à des bases de données.
- Gratuit

2.2.2 Librairie graphique

Nous avons choisi *wxPython* 2.3.1[2]. Cet environnement est portable sous Windows, Unix et Linux. L'adaptation MacOS est en cours.

3 Outils

3.1 CVS

CVS est un outil de gestion de versions, libre et portable. Il est déjà utilisé par plusieurs membres du groupe. Il est intégré à la suite d'outils *Cygwin*[3] qui est libre.

L'accès aux sources se fait par les lignes de commandes suivantes :

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

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

3.2 Sourceforge

Sourceforge[4] est un site qui fournit gratuitement des outils de développement en ligne pour les concepteurs de logiciels libre. Ainsi, les sources du projet sont stockées sur leur serveur (cf 3.1 page précédente). Le site web du projet (cf 1 page 2) est également hébergé par Sourceforge. Tous les documents de conception, ainsi que des scans des documents papier sont disponible sur le site du projet.

3.3 \LaTeX

Toute la documentation sera produite en \LaTeX [5] (ce document également), excepté l'aide en ligne, dont le format reste à définir. Ainsi, la documentation sera disponible en HTML, pdf, postscript et texte.

4 Responsabilités

Chef de groupe Laurent Burgbacher
Chef de groupe remplaçant Philippe Wälti
Documentation Cédric Dutoit
Tests Nabil Hamadi
Qualité Dève Roux
Configuration Philippe Wälti
Composants Nicolas Dubois

5 Groupes

5.1 Structure de données

Il s'agit de la modélisation et de la conception de la structure des données que manipule le programme.

- Dève Roux
- Laurent Burgbacher

5.2 wxOGL

wxOGL est la partie dessins vectoriel de wxPython (cf 2.2.2 page précédente).

- Philippe Wälti
- Nabil Hamadi

5.3 Interface graphique

Design et implémentation de l'interface graphique utilisateur.

- Cédric Dutoit
- Nicolas Dubois

Références

- [1] <http://www.python.org>.
- [2] <http://www.wxpython.org>.
- [3] <http://www.cygwin.com>.
- [4] <http://www.sf.net>.
- [5] <http://www.latex-project.org>.