Table des matières

Bibliographie

Guide utilisateur bouml : http://bouml.free.fr/doc/index.html
GenCxxTest : http://bouml.free.fr/GenCxxTest/gencxxtest_readme.html
CxxTest : http://cxxtest.tigris.org/
Guide CxxTest : http://cxxtest.sourceforge.net/guide.html

Objectifs

Réaliser un exemple simple en suivant les étapes d'un processus de développement présentant une vue d'ensemble d'UML et de la modélisation graphique.

Activités du TP :

- utiliser un outil de génie logiciel = bouml
- produire des diagrammes UML (cas d'utilisation, diagrammes de séquence et de classes)
- générer du code à partir d'un outil de génie logiciel avec bouml
- **réaliser les tests unitaires et de validation d'une application simple avec bouml**

Remarque : ce document décrit la méthode pour générer des tests logiciels avec l'outil bouml.

Travaux pratiques : tests logiciels d'un jeu de dés avec bouml

Introduction

On décrit étape par étape la méthode pour générer des tests logiciels avec le plugout GenCxxTest avec l'outil bouml.

Étape n°1 : fabriquer et installer le plug-out GenCxxTest pour bouml

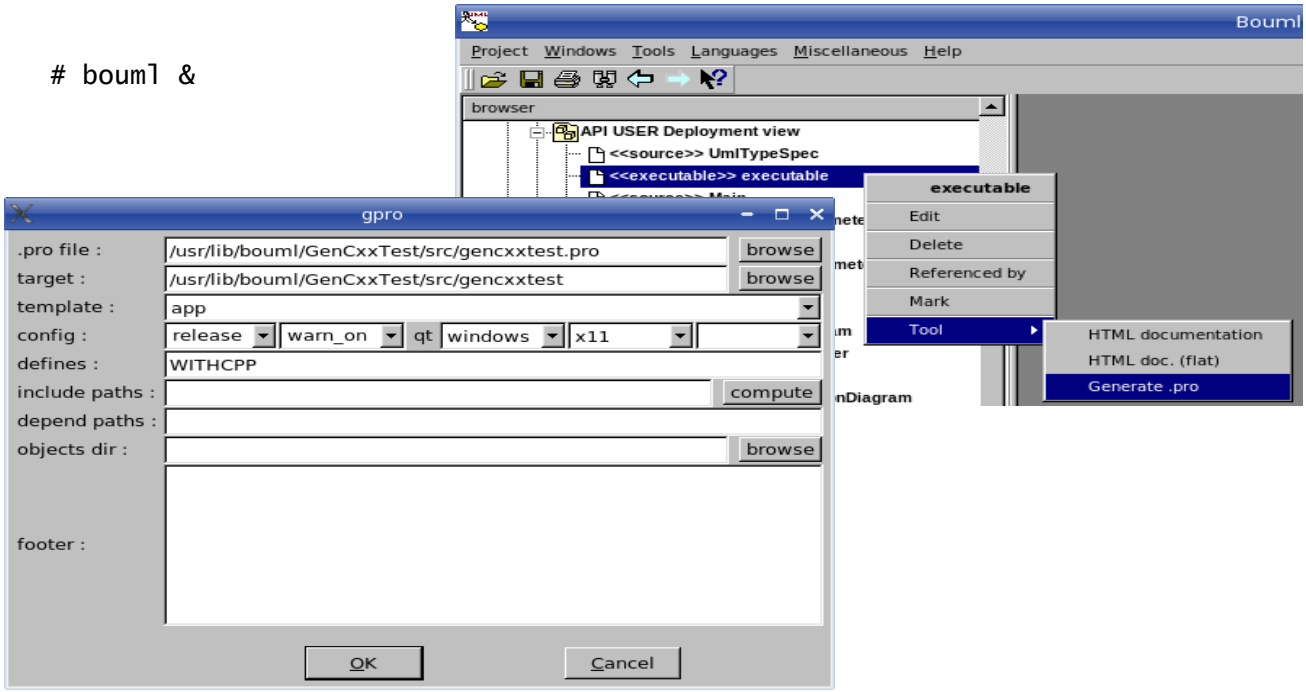
La procédure est décrite dans la section « Build et Install » du document http://bouml.free.fr/GenCxxTest/gencxxtest_readme.html :

- Uncompress the downloaded archive into a folder :

```
# cd /usr/local/

# tar zxvf /partages/Cours-TD-
TP/tv/methodologie/tpTestsLogiciels/outils/GenCxxTest.tar.gz

# mv GenCxxTest /usr/lib/bouml/

# cd /usr/lib/bouml/

# chown -R root:root GenCxxTest

# cd GenCxxTest/src/
```

- Open the cxxtest.prj file using Bouml. Set the generation directory for C++ in the Bouml generation settings. Generate C++ source code and apply "Generate .pro" plug-out on the 'executable' artifact in the "API USER Deployment View" :

```
# bouml &
```



- Change to the folder where the C++ source (and .pro file) has been generated by Bouml :

```
# cd /usr/lib/bouml/GenCxxTest/src/
```

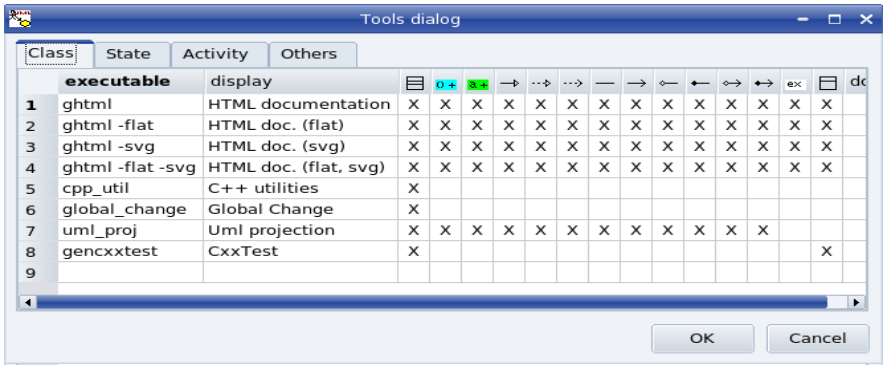
- Generate make file by using the "qmake gencxxtest.pro" command on linux/unix or use tmake on windows :

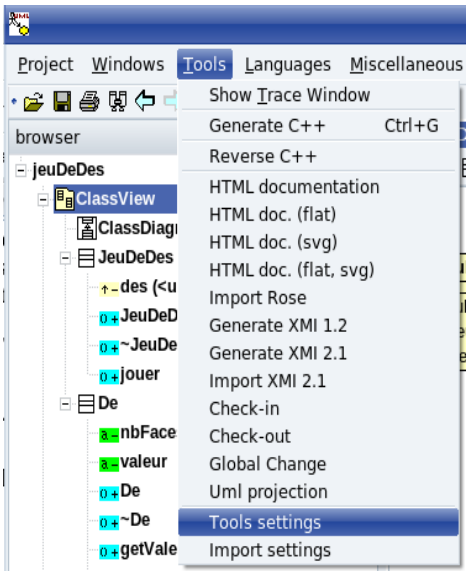
```
# qmake gencxxtest.pro

# make
```

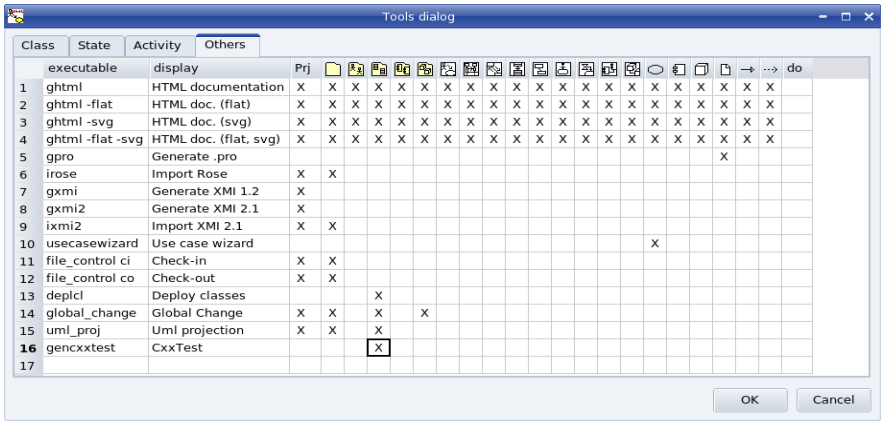
- Copy the executable 'gencxxtest' (or 'gencxxtest.exe' in windows) to the Bouml directory :

```
# cp gencxxtest /usr/lib/bouml/
```

- Configure Bouml to use the plugout for class and class view items as discribed above :







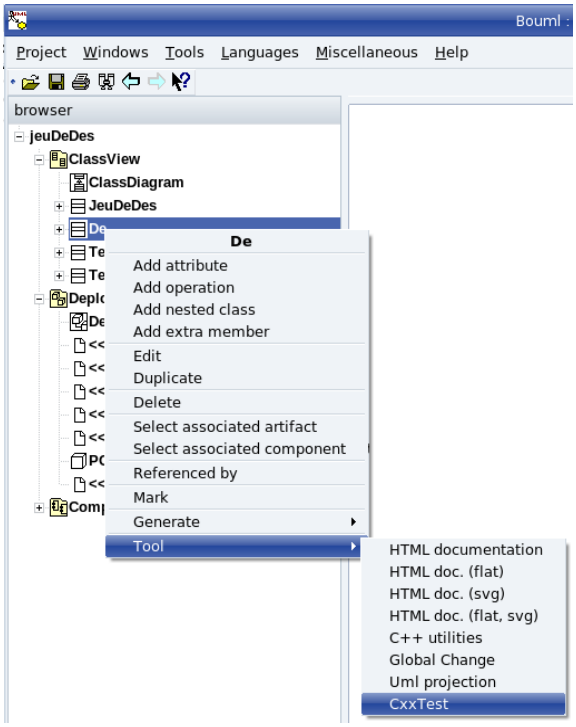
Étape n°2 : installer cxxtest

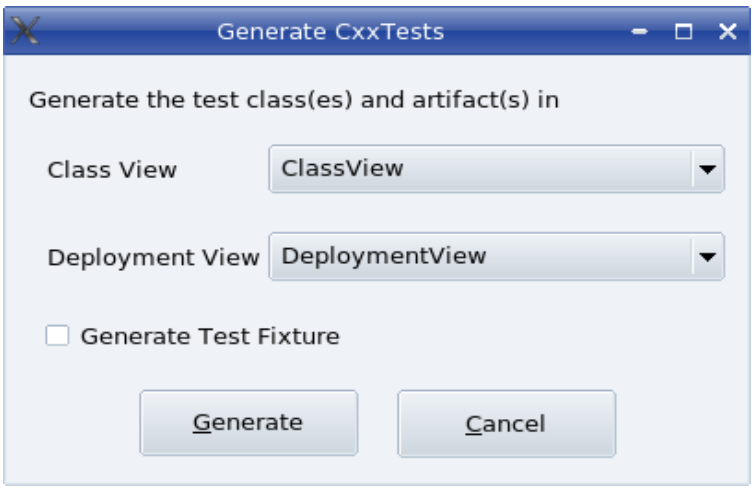
CxxTest is a JUnit/CppUnit/xUnit-like framework for C/C++. Pour installer CxxTest, il suffit de décompresser l'archive :

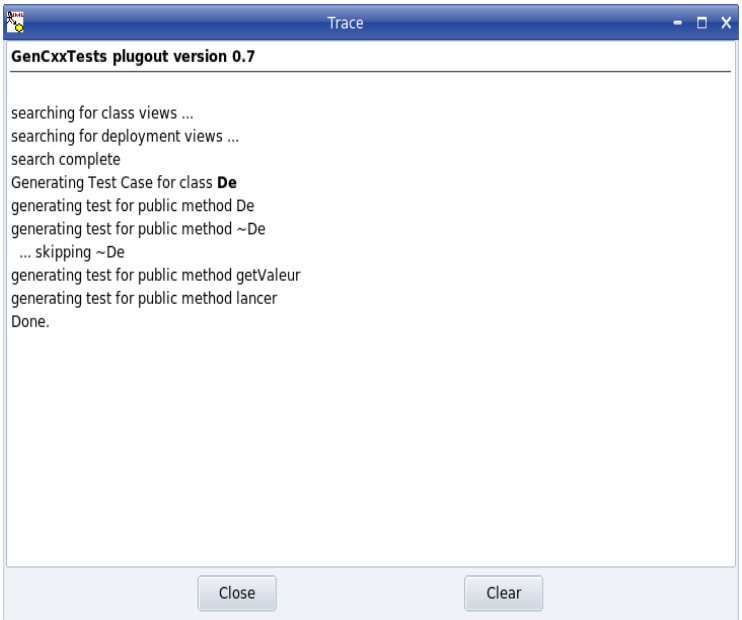
```
# cd /usr/local/

# unzip /partages/Cours-TD-TP/tv/methodologie/tpTestsLogiciels/outils/cxxtest-3.10.1.zip
```

Étape n°3 : générer les classes de tests avec GenCxxTest

Avec bouml, cliquer avec le bouton droit sur une classe :

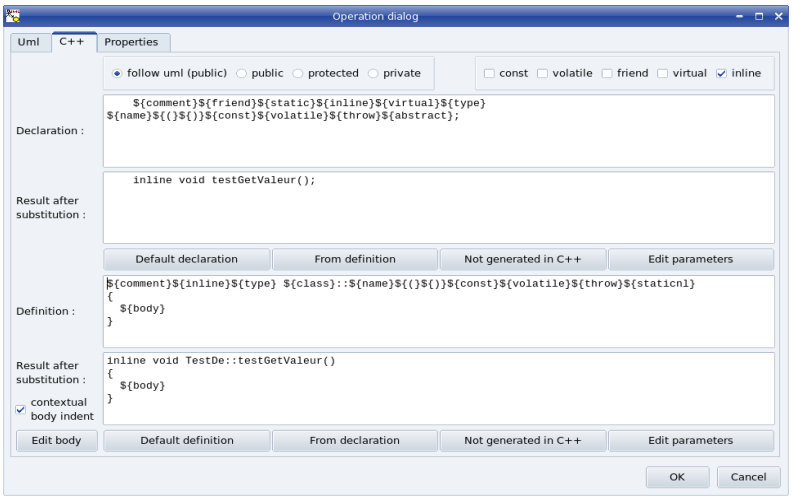






Étape n°4 : générer le code C++

Il est conseiller maintenant d'écrire ses tests unitaire en éditant (« Edit body ») les méthodes de la classe de test :



Aidez-vous de la documentation sur le framework CxxTest : http://cxxtest.sourceforge.net/guide.html

Puis, dans le menu « Tools », choisir « Generate C++ ».

Étape n°5 : fabriquer un programme de test

Le plugout GenCxxTest a générer un fichier AllTests.h qu'il faut maintenant transformer en programme de test :

```
$ /usr/local/cxxtest/cxxtestgen.pl --error-printer -o runner.cpp AllTests.h

$ g++ runner.cpp -I/usr/local/cxxtest -c

$ g++ runner.o De.o JeuDeDes.o -o  runner

$ ./ runner
```

ou plus simplement en utilisant le Makefile fourni :

```
$ make -f Makefile.tests

$ ./ runner

Running 5 tests
In TestDe::testDe:
AllTests.h:24: Warning: Test testLancer() not implemented
.
In TestDe::testGetValeur:
AllTests.h:37: Trace: Test testGetValeur() : la valeur du de lance doit etre
comprise entre 1 et 6
AllTests.h:53: Trace: Test testGetValeur() : chaque valeur doit sortir plus
d'une fois (equi-probabilite)

1 -> 16563 > 15000
2 -> 16630 > 15000
3 -> 16632 > 15000
4 -> 16742 > 15000
5 -> 16732 > 15000
6 -> 16701 > 15000
..
In TestJeuDeDes::testJeuDeDes:
AllTests.h:97: Warning: Test testJeuDeDes() not implemented
.
In TestJeuDeDes::testJouer:
AllTests.h:111: Trace: Test testJouer() : les valeurs des deux des lances ne
doivent pas etre toujours egales

1540 < 5000 = 15.4%
.OK!
```