

Table des matières

| | |
|--|----------|
| Bibliographie..... | 1 |
| Objectifs..... | 1 |
| Travaux pratiques : un jeu de dés avec bouml..... | 2 |
| Introduction..... | 2 |
| étape n°1 : définir les paramètres généraux de la génération de code...2 | |
| étape n°2 : réaliser la conception du projet.....3 | |
| étape n°3 : établir le déploiement du projet en créant les artefacts associés.....4 | |
| étape n°4 : générer le code C++.....5 | |
| étape n°5 : produire les diagrammes de déploiement et de composants 6 | |
| Annexe : création d'un artefact de type « source » pour une classe.....7 | |

Bibliographie

Tutoriel bouml : http://bpages.developpez.com/tutoriels/bouml/classes-generation/?page=page_4#LIV

Guide utilisateur bouml : <http://bouml.free.fr/doc/index.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

Remarque : ce document décrit la méthode pour générer du code C++ avec l'outil bouml.

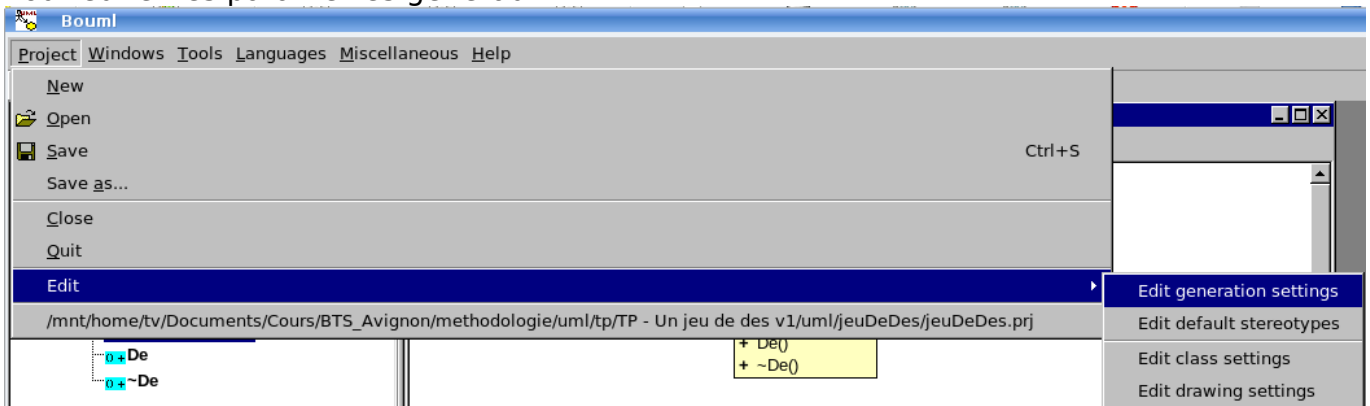
Travaux pratiques : un jeu de dés avec bouml

Introduction

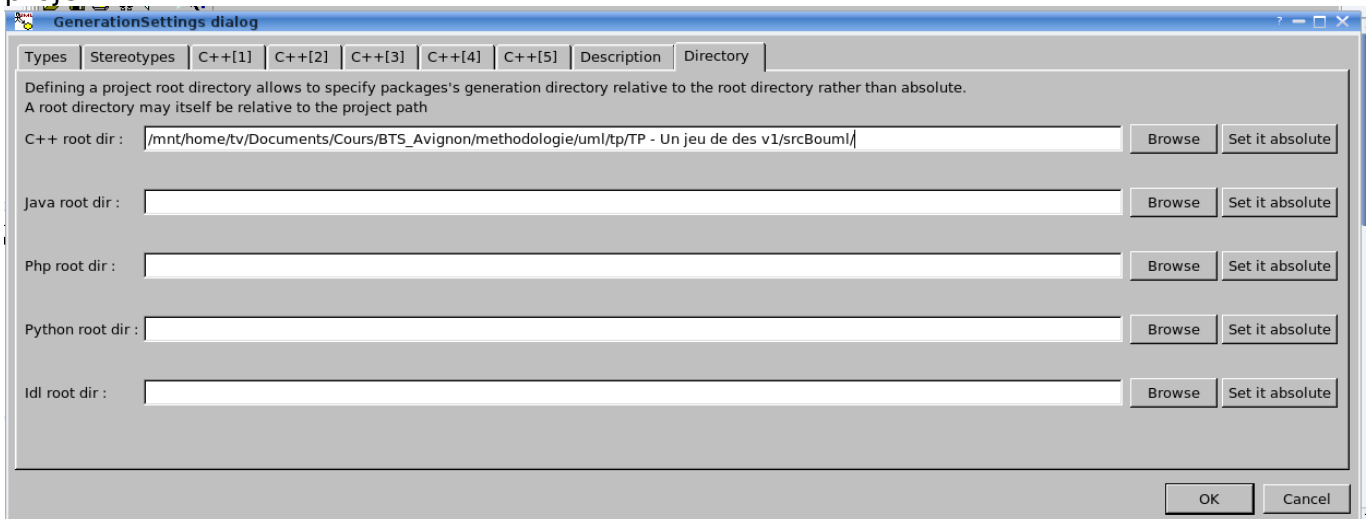
On décrit étape par étape la méthode pour générer du code C++ avec l'outil bouml.

Étape n°1 : définir les paramètres généraux de la génération de code

Pour éditer les paramètres généraux :



Au minimum, on indiquera le chemin racine où seront générés des les différents fichiers du projet :

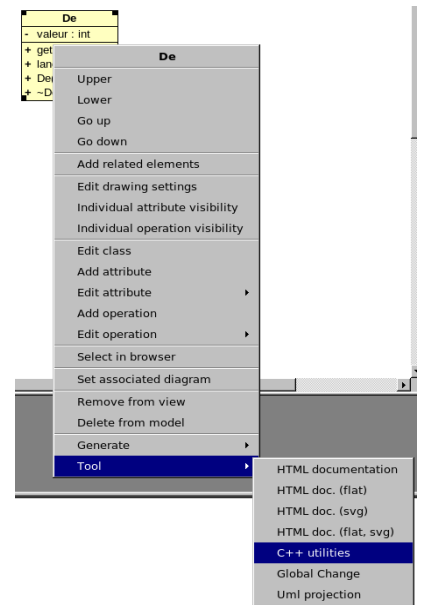
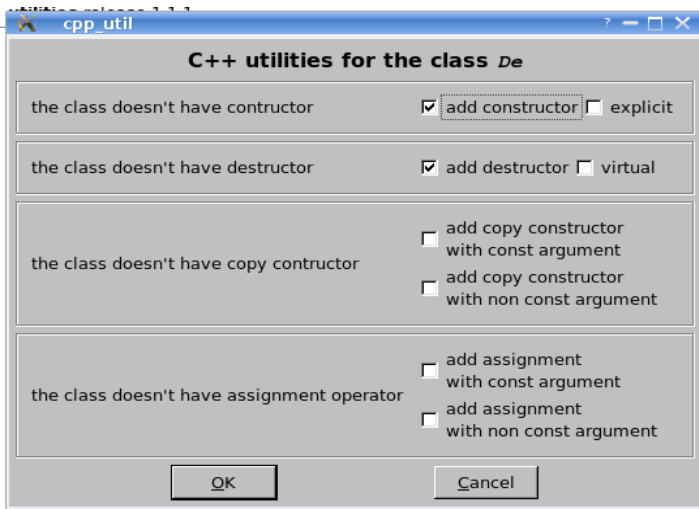
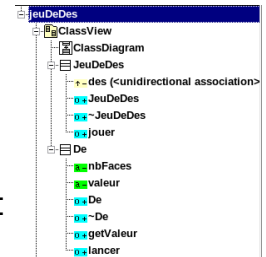


Remarque : On vous conseille, en utilisant les onglets C++[n], de personnaliser les règles de codage à respecter dans le projet. De toute manière, un outil, comme `bcpp`, est conseillé pour le formatage du code source.

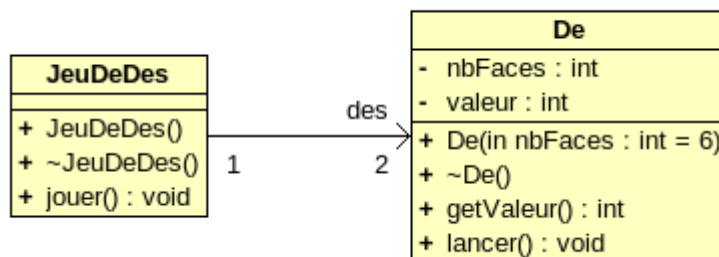
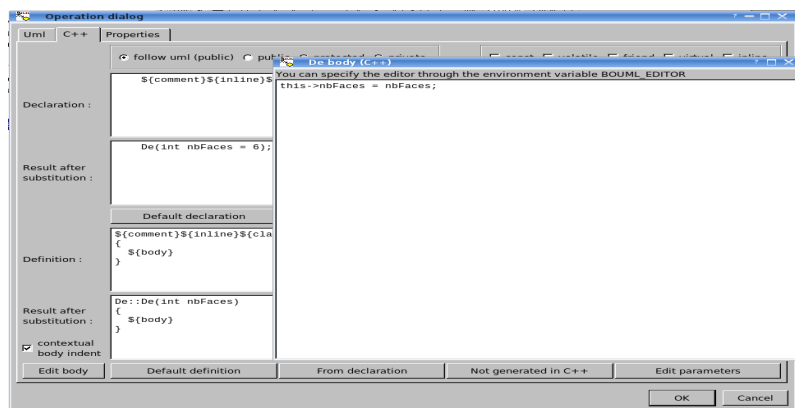
Étape n°2 : réaliser la conception du projet

Pour cela, il faut réaliser avec bouml les actions dans l'ordre suivant :

- créer une vue de classes (Class View)
- créer, dans cette nouvelle Class View, un diagramme de classes (Class Diagram)
- créer vos différentes classes dans ce diagramme de classes
- ajouter les attributs et les opérations à vos classes
- établir les relations entre classes ainsi que les multiplicités et le(s) nom(s)
- utiliser l'utilitaire c++ de bouml pour ajouter les constructeurs/destructeurs à vos classes qui en ont besoin :



- éditer vos opérations :



Étape n°3 : établir le déploiement du projet en créant les artefacts associés

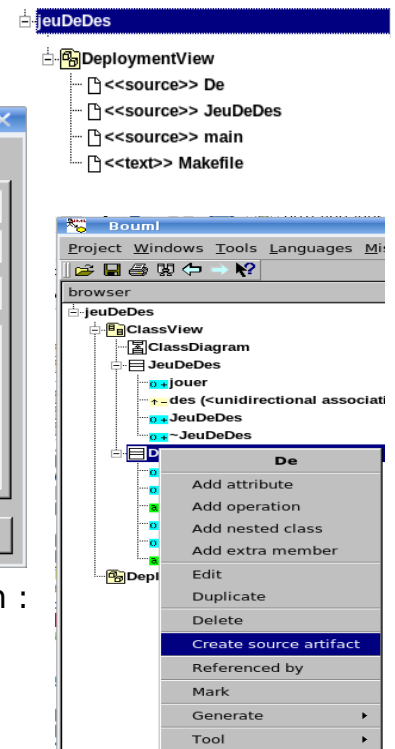
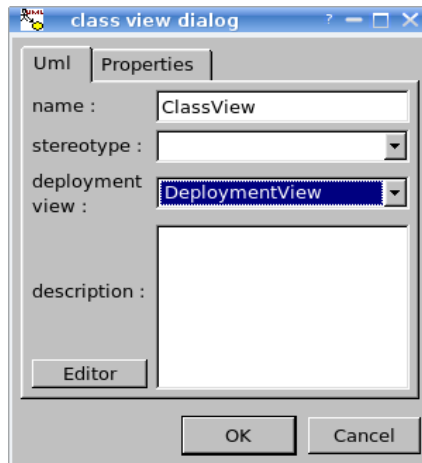
Avec bouml, pour générer du code, il faut tout d'abord créer les artefacts.

Rappel : un **artefact** est simplement le terme général qui désigne tout chose produite ou consommée par une étape du processus de développement. Cela peut désigner du texte, du code , des documents ... etc .

Dans bouml, les artefacts sont associés à un diagramme de déploiement.

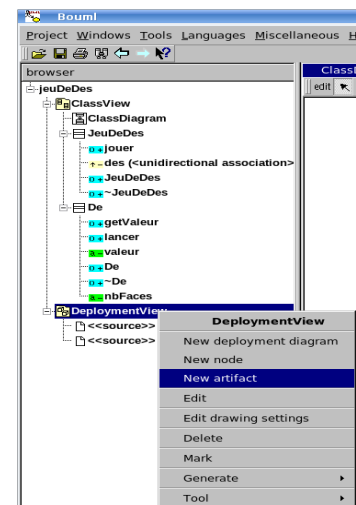
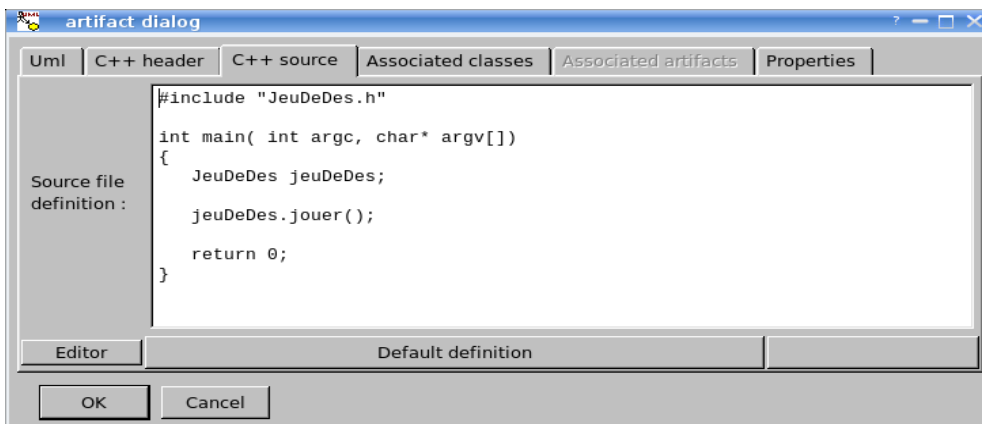
Pour cela, il faut réaliser avec bouml les actions dans l'ordre suivant :

- créer une vue de déploiement (Deployment View)
- associer cette nouvelle vue de déploiement à la vue de classes de conception :



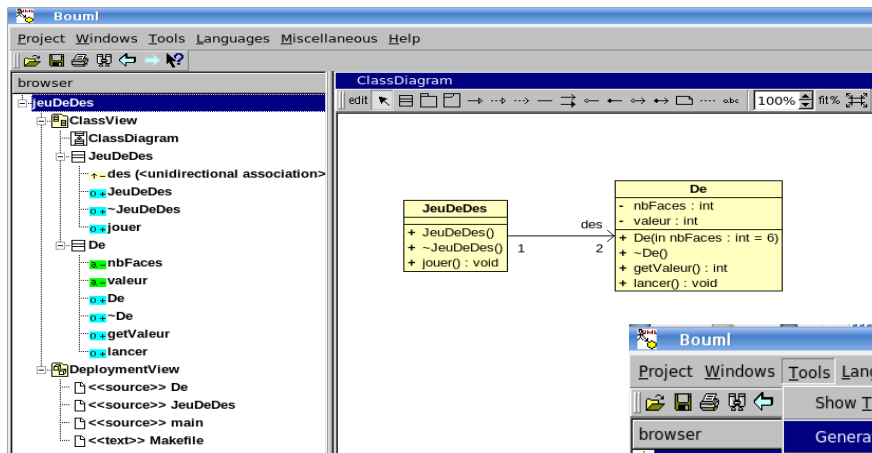
- créer les artefacts associés aux classes de conception : (c'est la technique la plus efficace, sinon une autre méthode est décrite en Annexe)

- créer les artefacts suivants : main (stéréotype <<source>>) et Makefile (stéréotype <<text>>) et éditer leur contenu :



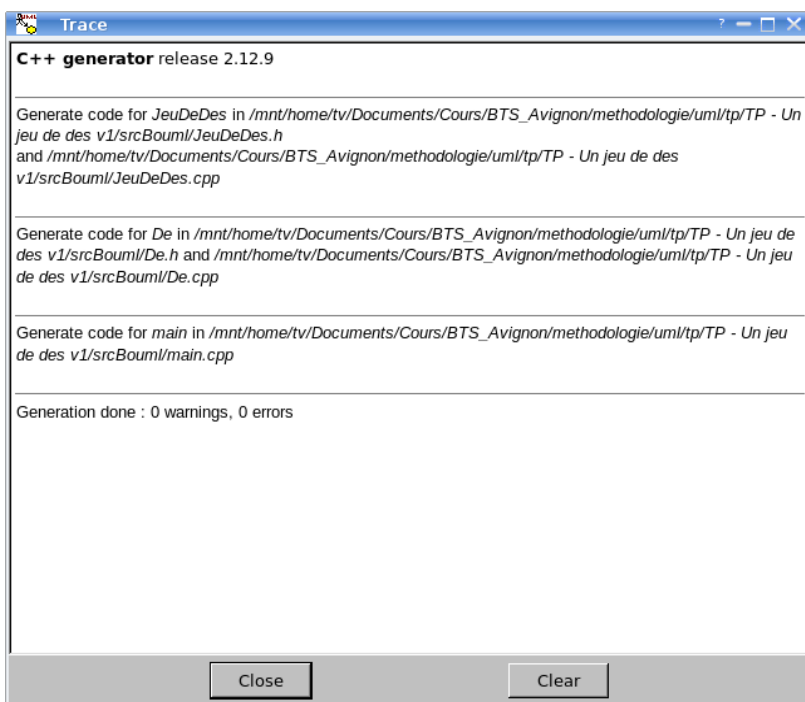
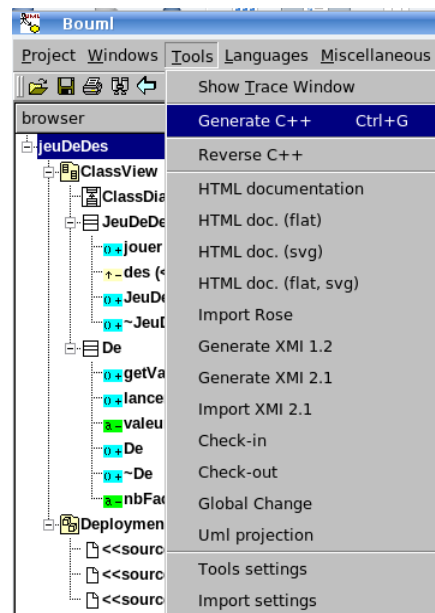
Étape n°4 : générer le code C++

Avant de lancer la génération de code automatique, il est nécessaire d'organiser les attributs et les opérations des classes concernées afin d'obtenir un code plus lisible :



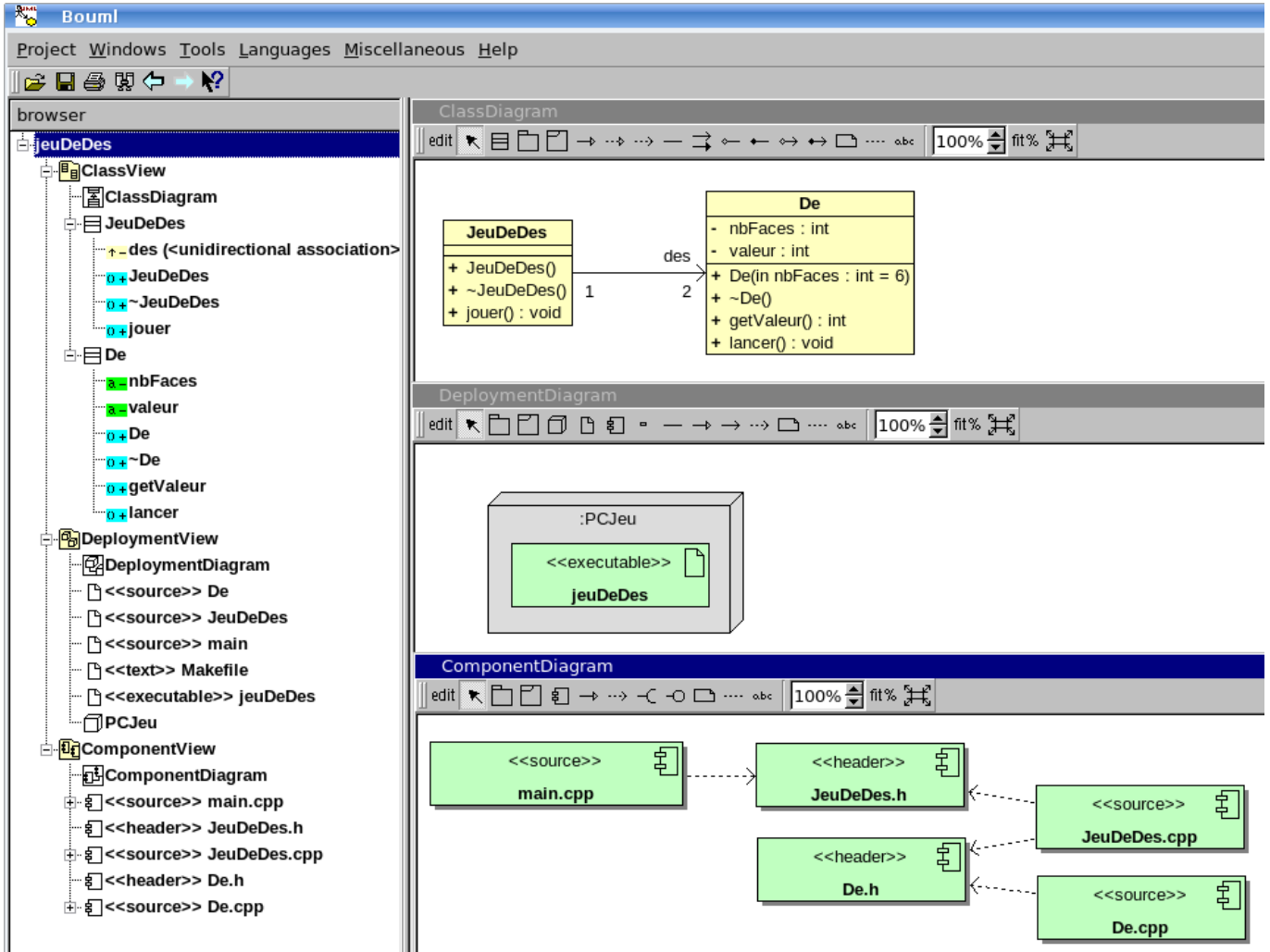
Pour lancer la génération de code, il faut faire :

La fenêtre Trace permet de suivre le processus de génération :



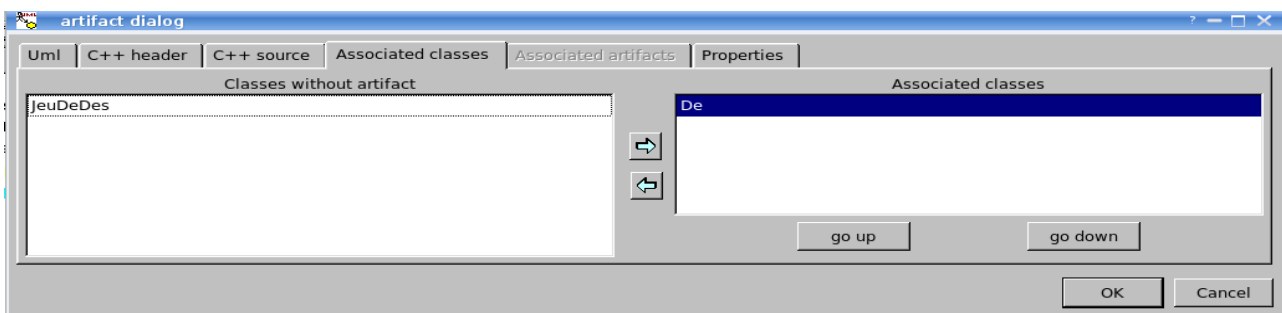
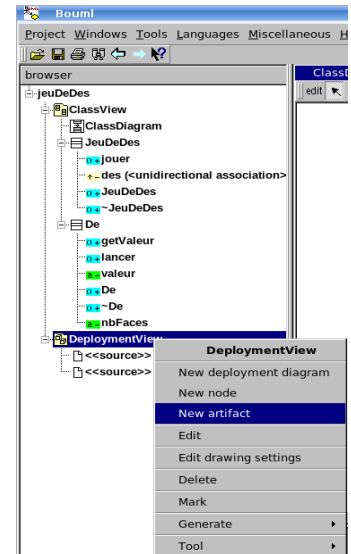
On peut maintenant fabriquer l'application en tapant simplement : `$ make`

Étape n°5 : produire les diagrammes de déploiement et de composants

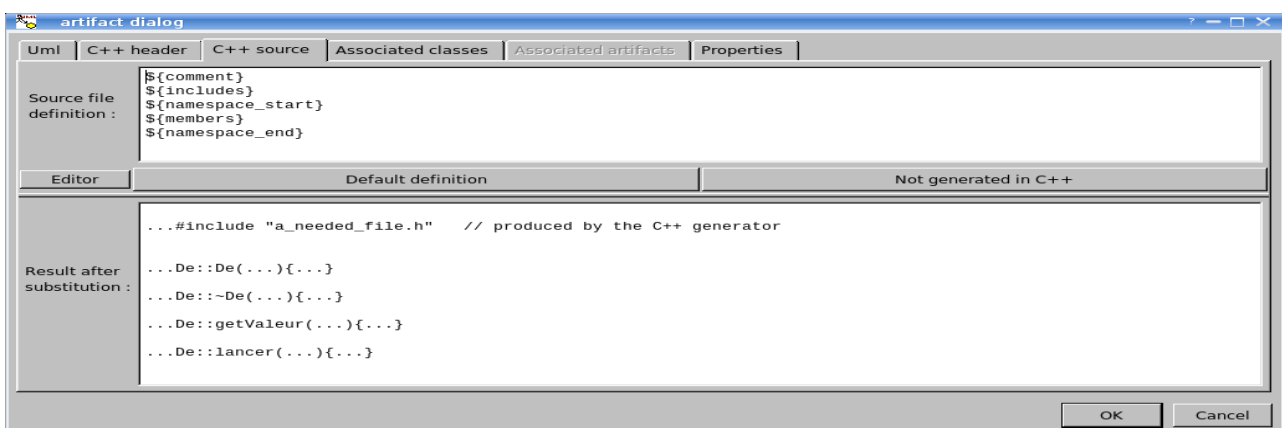


Annexe : création d'un artefact de type « source » pour une classe

- Sélectionner la vue de déploiement et avec le bouton droit de la souris, créer un nouvel artefact :
- Donner le nom de la classe désirée
- Editer le nouvel artefact et sélectionner le stéréotype « source »
- Puis, dans l'onglet « Associated classes », choisir la classe désirée :



- Dans l'onglet « C++ source », cliquer sur « Default definition » :



- Reproduire la même action pour l'onglet « C++ header »

Maintenant, vous pouvez générer le code C++ pour cette classe.