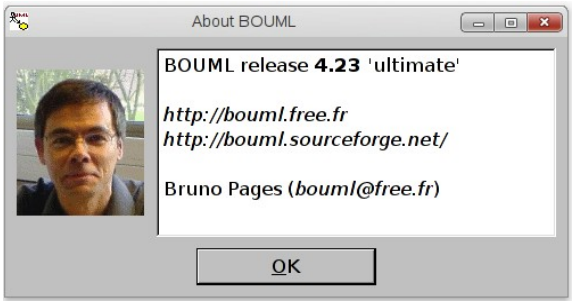
Table des matières

Introduction

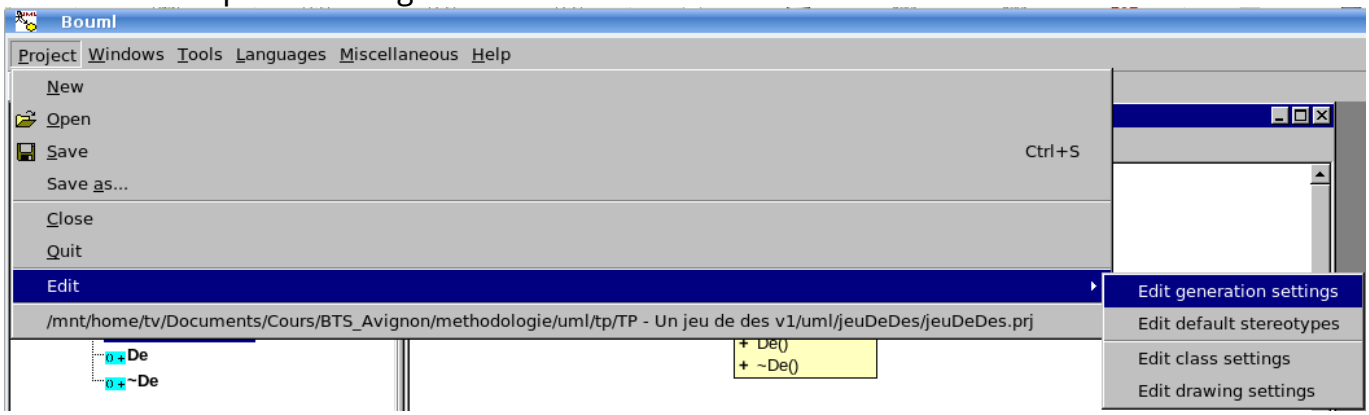
L'exemple présenté ici montre la génération de code automatique d'une classe avec l'atelier de génie logiciel **bouml**.

La version utilisée dans cet exemple est :

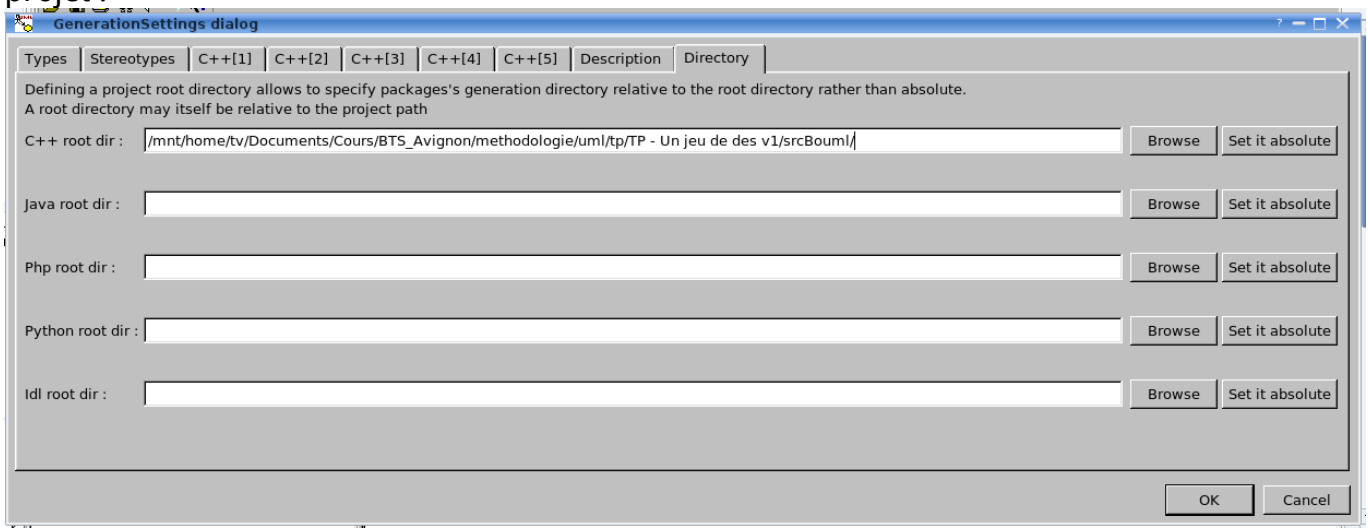


Éditer les paramètres du projet

Pour éditer les paramètres généraux :



Au minimum, on indiquera le chemin racine où seront générés 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, sera conseillé plus tard pour le formatage du code source.

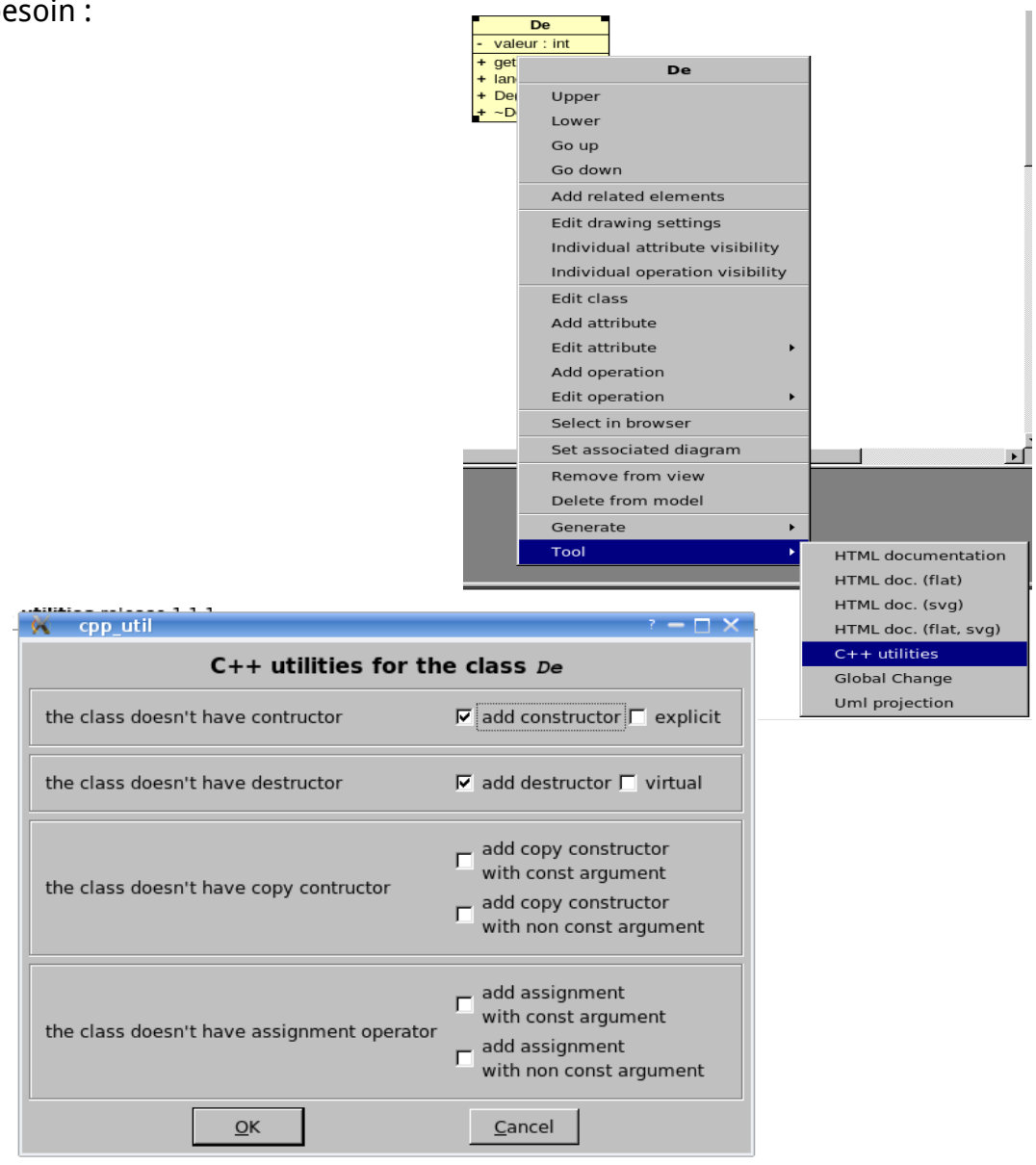
Pré-requis

Il vous faut disposer d'un projet bouml dans lequel les actions dans l'ordre suivant ont été réalisées :

- 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

Ajouter un constructeur (et/ou un destructeur) à vos classes

Vous pouvez utiliser l'utilitaire c++ de bouml pour ajouter les constructeurs/destructeurs à vos classes qui en ont besoin :



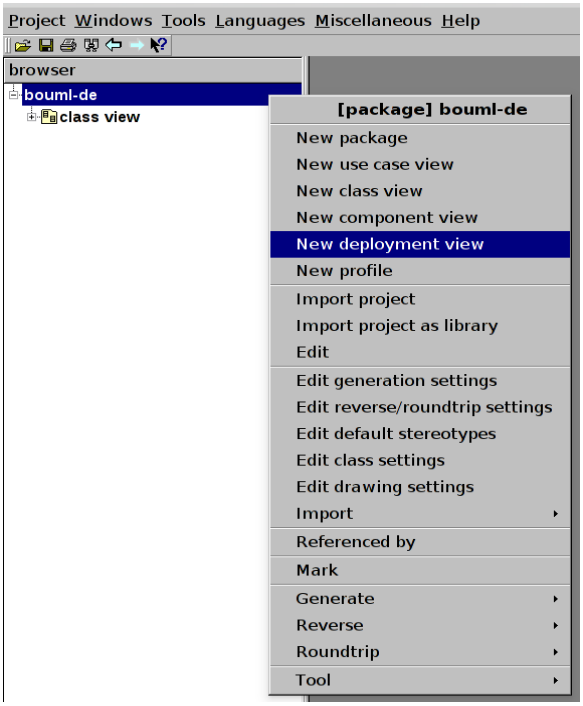
Générer le code source automatiquement

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

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

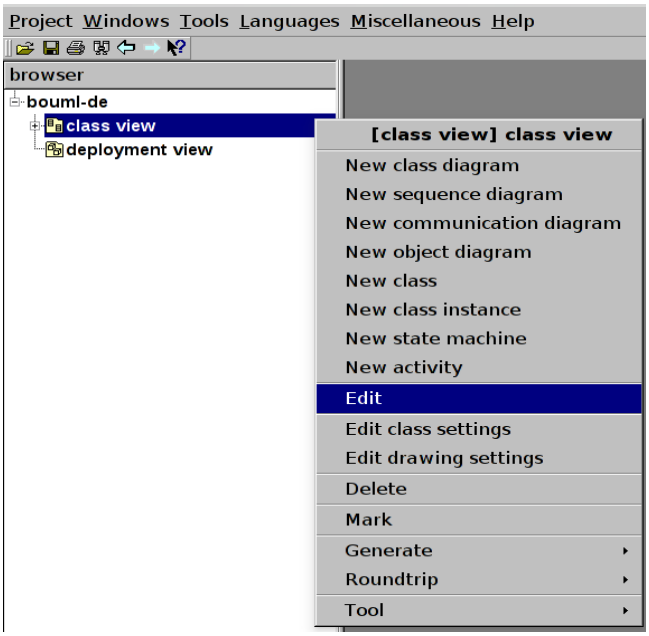
Dans bouml, les artefacts sont associés à un diagramme de déploiement.

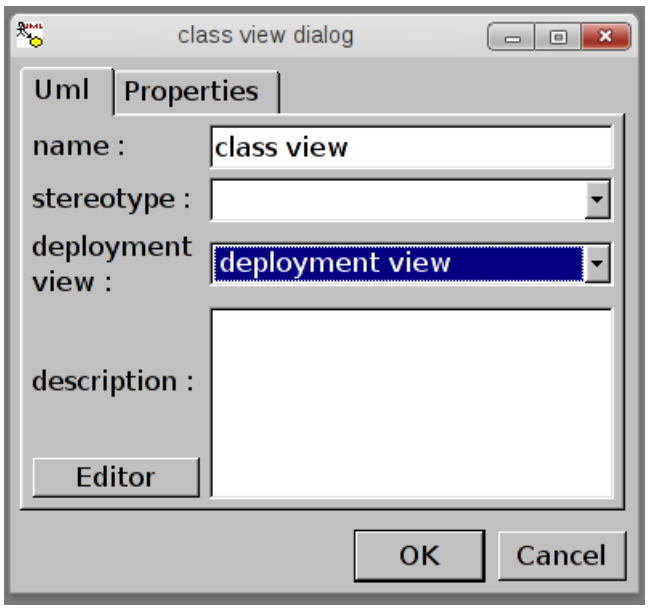
Pour cela, il faut réaliser avec bouml les actions dans l'ordre suivant :



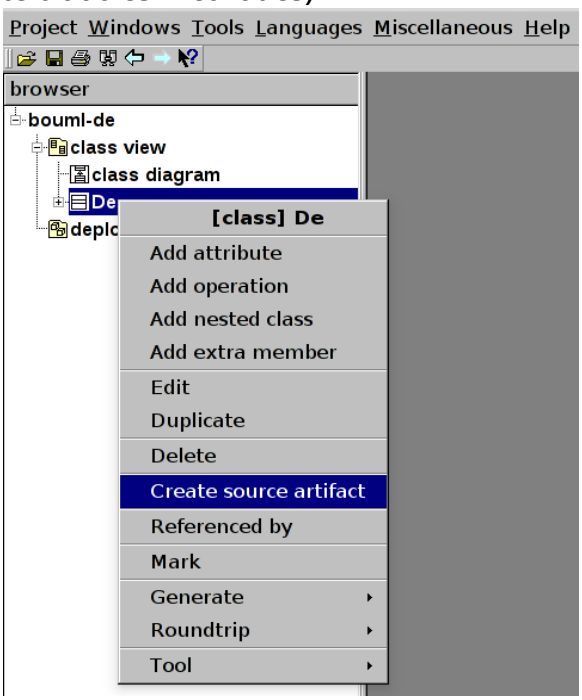
- créer une vue de déploiement (Deployment View)

- associer cette nouvelle vue de déploiement à la vue de classes de conception (ici « class view ») :

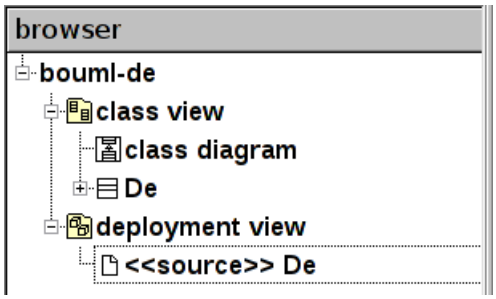




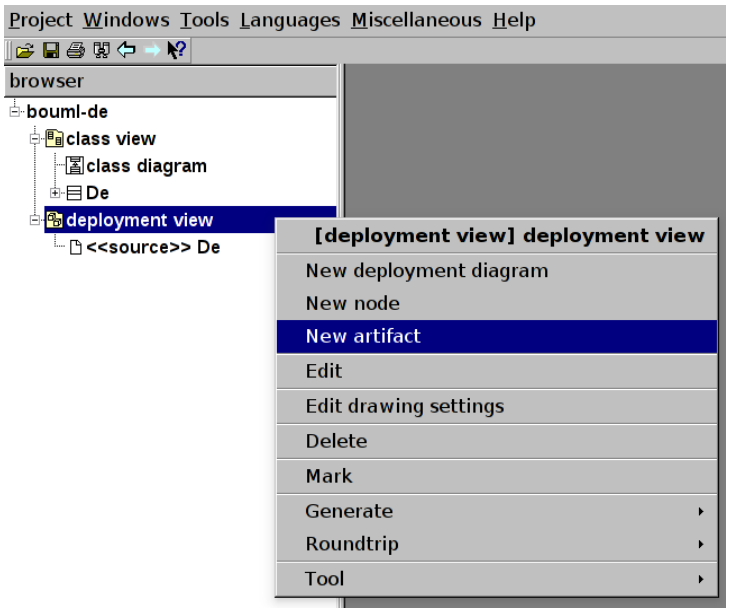
- créer les artefacts associés aux classes de conception : (c'est la technique la plus efficace, sinon il existe d'autres méthodes)

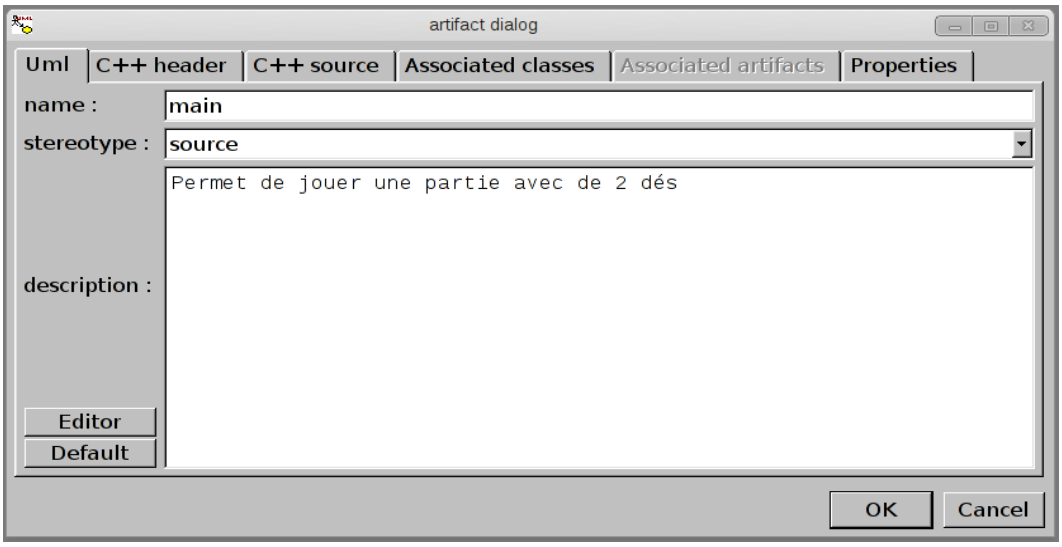


On obtient :

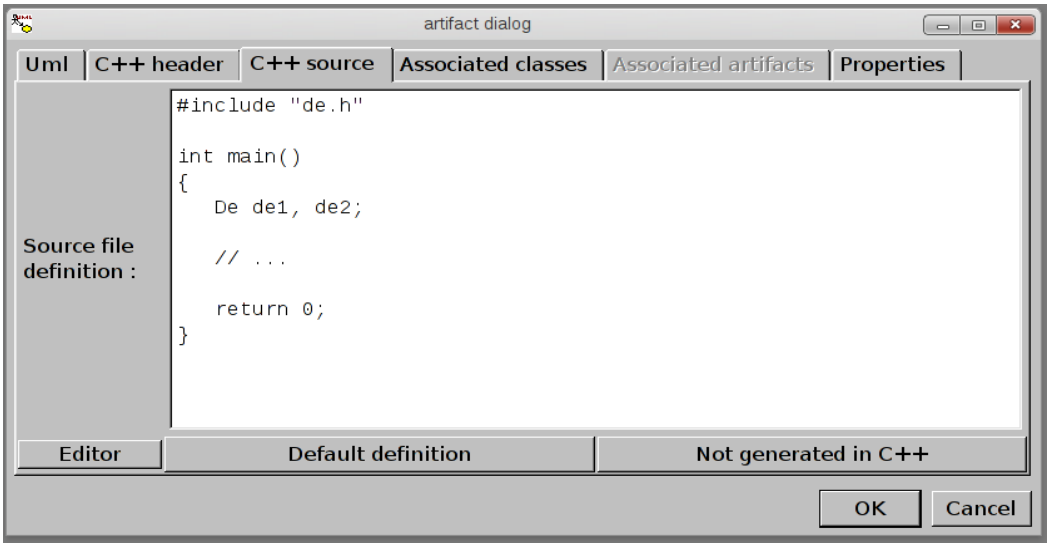


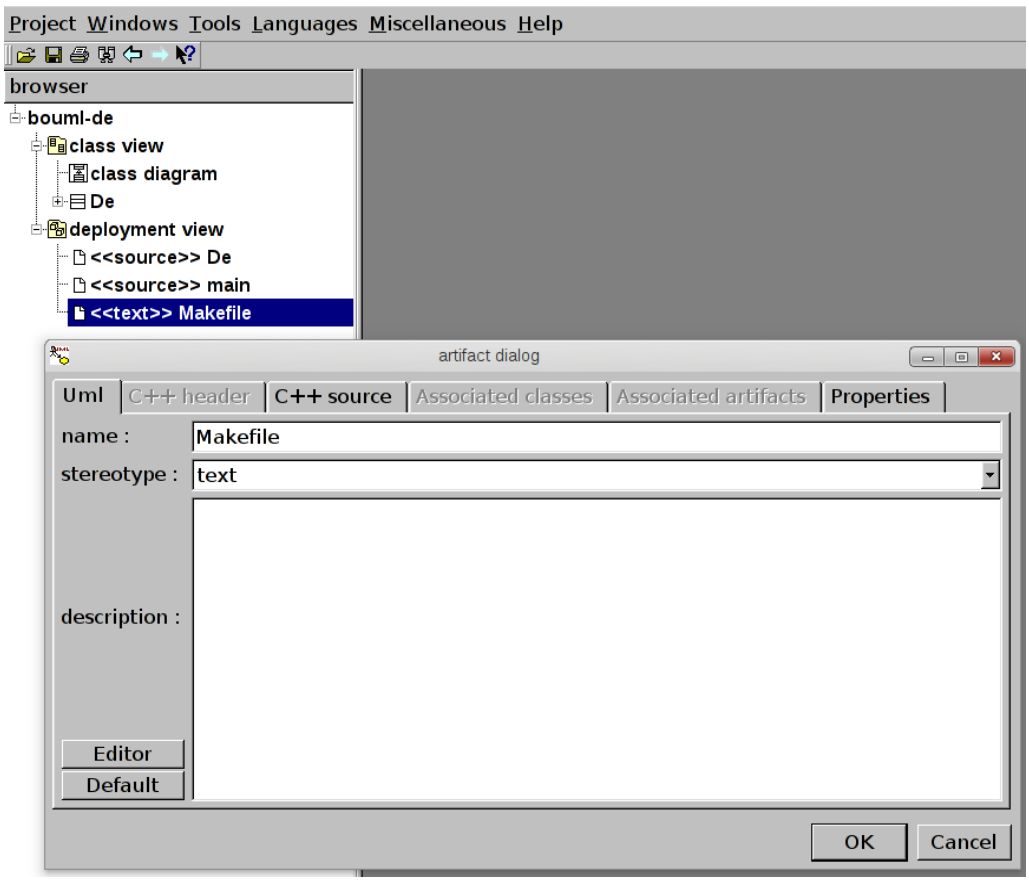
- créer les artefacts suivants : main (stéréotype <<source>>) et Makefile (stéréotype <<text>>) et éditer leur contenu :





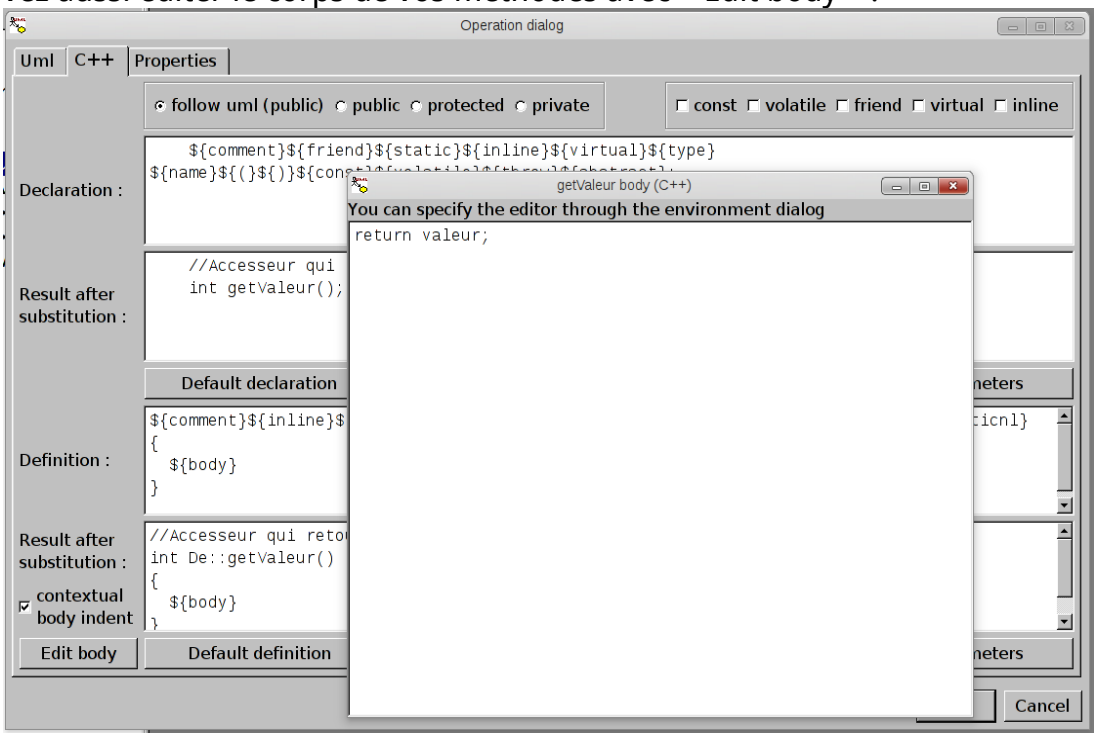
Il est même possible d'éditer le code qui sera généré :



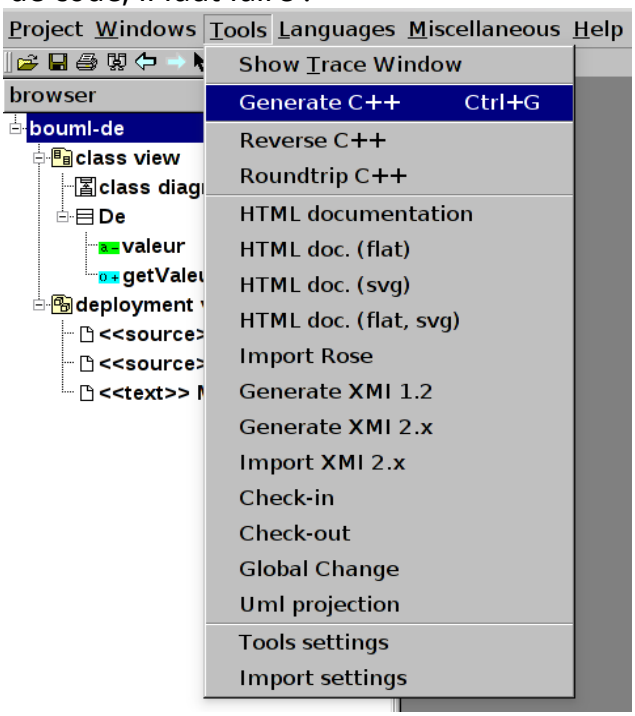


Conseil : Avant de lancer la génération de code automatique, il est nécessaire d'organiser les attributs et les opérations des classes concernées afin d'obtenir un code plus lisible.

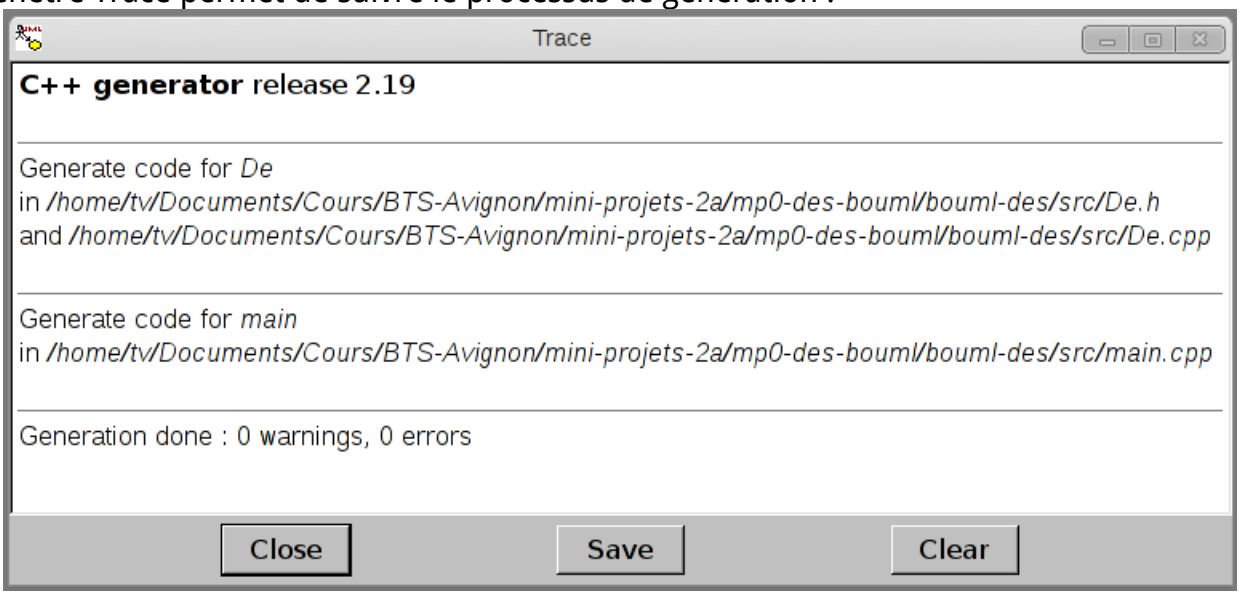
Vous pouvez aussi éditer le corps de vos méthodes avec « Edit body » :



Pour lancer la génération de code, il faut faire :



La fenêtre Trace permet de suivre le processus de génération :



Ici, trois fichiers sources ont été générés avec succès.

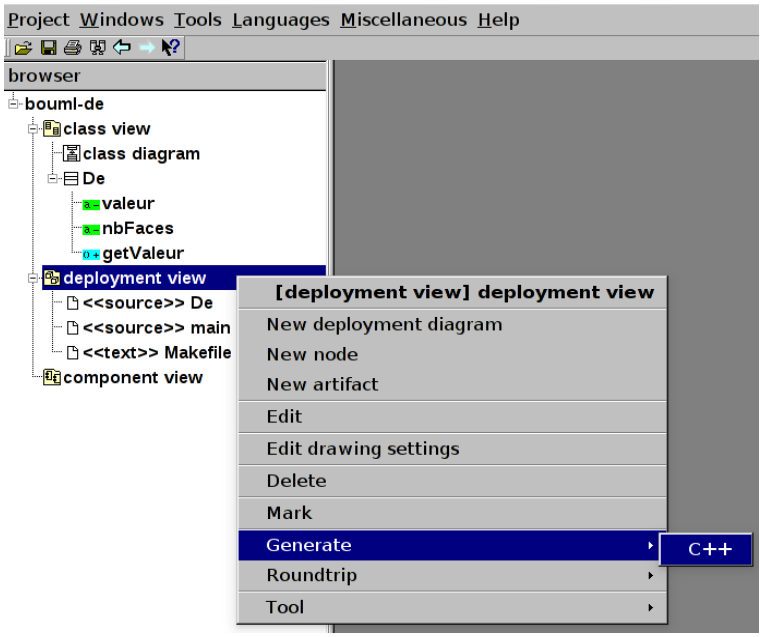
Remarque : le code source du fichier Makefile n'a pas été généré car il était vide !

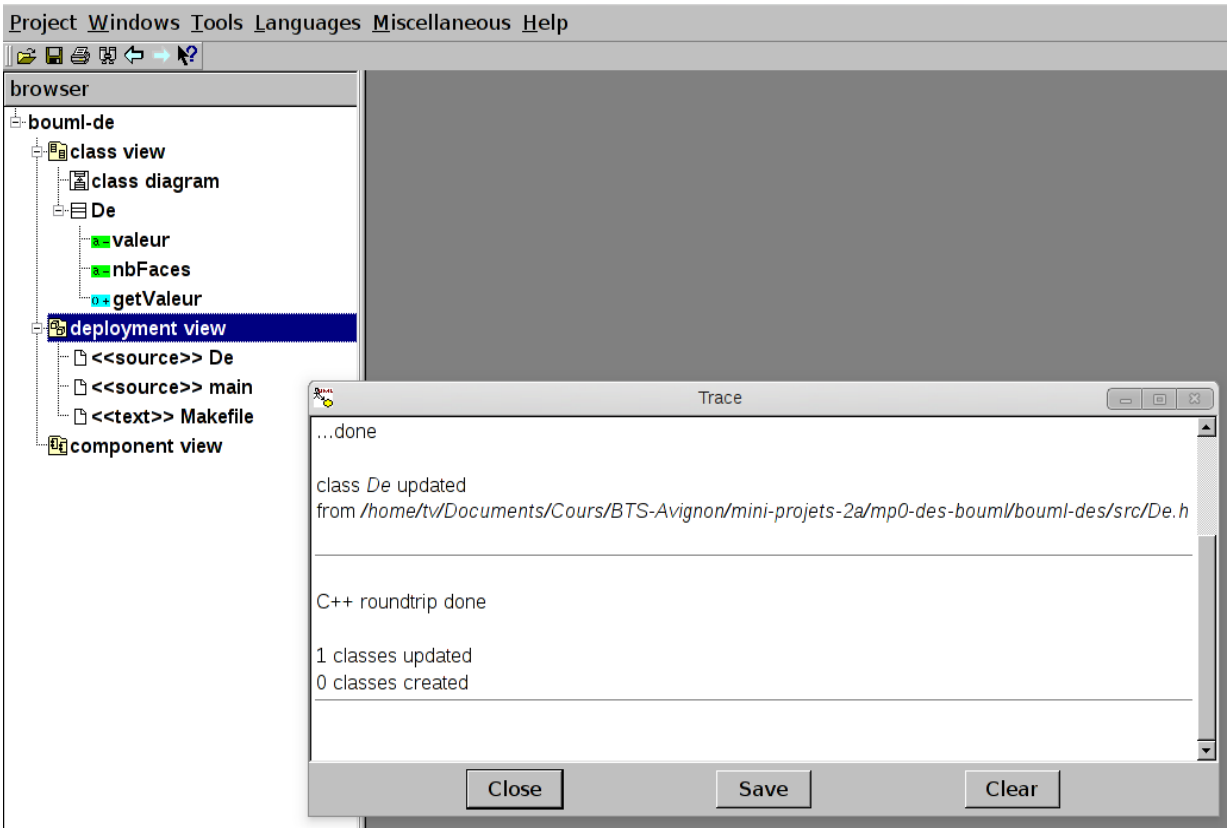
Attention : à chaque nouvelle génération de code, les fichiers générés précédemment seront écrasés. Dans ce cas, il vous faudra faire des « roundtrip ».

Aller-Retour ou « Roundtrip »

Ce mécanisme nommé « Roundtrip » dans bouml vous permet d'éditer les fichiers sources générés en dehors de bouml.

Par exemple, si vous ajoutez un attribut à la classe De directement dans le fichier De.h, vous aurez alors un diagramme de classe incohérent (non à jour) dans bouml. Pour mettre à jour vos diagrammes dans bouml, il vous faut faire un « Roundtrip » :





Reverse engeneering

On a vu dans cet exemple la génération de code, on peut aussi souligner que les outils de génie logiciel comme bouml sont aussi capables de faire du ***reverse engeneering***, c'est-à-dire de générer des diagrammes à partir du code.