

# Travaux Pratiques

## Ingénierie Dirigée par les Modèles

### Tâche 5:

## Transformation de modèles

**Frédéric Fondement**

[frederic.fondement@uha.fr](mailto:frederic.fondement@uha.fr)

### 1. Description du problème

La tâche précédente a consisté à générer du code C en utilisant un générateur de code à base de patrons. Une alternative à cette méthode pour une utilisation pratique des modèles est la transformation de modèles. L'idée générale est de convertir (plus ou moins simplement) les informations contenues dans un modèle soit vers un modèle qui exprime un résultat (dans un domaine sémantique donné), soit vers un modèle qui peut être directement utilisé (par exemple un modèle exécutable). Notez que lors de cette transformation, des informations peuvent être perdues, rendant impossible de revenir du modèle qu'on qualifia d'exécutable, à celui qu'on qualifia d'abstrait. Notez aussi que dans certaines applications, il est peut être nécessaire de traduire un même modèle abstrait en plusieurs modèles exécutables. Un exemple est la modélisation d'applications internet qui nécessitent la génération de modèles de comportement (comme JSP) et d'un modèle de stockage de données (comme MySQL).

Bien sûr, cette approche peut être itérative et un modèle exécutable peut à son tour jouer le rôle de modèle abstrait et être transformé à nouveau. Une autre force de l'IDM est de pouvoir s'adapter très rapidement aux changements de plates-formes. Lors de la présentation finale, vous pourriez expliquer par exemple comment vous porteriez le processus de développement que vous avez conçu lors de ces deux séries de travaux pratiques sur la plate-forme J2EE.

Le but de cette tâche est le même que celui de la tâche précédente, cette fois-ci en utilisant une transformation de modèles. Un métamodèle pour le langage C vous est fourni, et est rendu exécutable par génération de code C.

### 2. Travail demandé

Vous utiliserez le langage de transformation de modèles ATL pour traduire vos modèles IACA en modèles C pour la plate-forme PIC 18F458. Vous vous référerez au descriptif de la tâche 4 pour la description de la génération de code.

Vous aurez à disposition en salle de TP une maquette à base de PIC 18F458 pour tester votre génération de code sur une cible matérielle. Bien sûr, vous vérifierez que vous êtes capables de générer du code valide, ainsi que d'autres modèles de votre choix.

### 3. Ressources et astuces

Le projet Eclipse contenant le métamodèle C et l'outillage pour génération de code C à partir de modèles C est disponible sur le site du cours. Ce métamodèle, dont une représentation est donnée par des diagrammes TopCASED, est supposé auto-explicite. Vous constaterez que plusieurs diagrammes sont disponibles dans le fichier TopCASED. Un fichier exemple est inclu dans le projet `c.mm` (`model/gen.cm`). Vous trouverez dans le répertoire `library` les déclarations sous forme de modèle C de la bibliothèque IACA (comme les prototypes des structures `c_bit`, `c_TOR`, etc., ainsi que des opérations `c_bit_init`, `c_bit_run`, `c_TOR_init`, etc.).

Le guide de l'utilisateur ATL est disponible sur le site du cours. Vous créez un nouveau projet de type ATL. Comme toujours, un exemple vous est donné pour les diagrammes d'états. Une fois de plus, ATL ne supporte pas bien les dépendances de modèles, vous installerez donc le plug-in correctif dans le répertoire `plugins` d'Eclipse, et veillerez bien à avoir la dernière version du plug-in chargeur de dépendances. D'autres exemples et ressources pour le langage ATL sont disponibles sur le site de documentation collaborative.

Vous voudrez bien vous référer au document de la tâche 4 pour les directives de génération de code, ainsi que pour la manipulation de la maquette.

### 4. Délivrables

- Votre métamodèle (`.ecore`) et le modèle correspondant à la Figure 1(tâche 4) au format XMI y conformant, accompagné du modèle standard IACA qui en dépend.
- La spécification ATL de la transformation d'un modèle IACA vers un modèle C dont le métamodèle vous est fourni.
- Des modèles de test au format XMI, déjà fournis comme livrables pour la tâche 4.

Le tout sera empaqueté dans un projet Eclipse au format archive ZIP et accompagné d'un fichier explicatif (par exemple `lisez-moi.pdf`).