

# Travaux Pratiques

## Ingénierie Dirigée par les Modèles

### Tâche 2:

## Cohérence des Modèles IACA

Frédéric Fondement

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

### 1. Description du problème

Lors de la tâche précédente, vous avez créé un métamodèle pour le langage IACA. Ce métamodèle vous a permis de construire des modèles comme la bibliothèque standard ICACA ou un processus utilisant des composants (voir les Figure 1 et Tableau 1 du document “Travaux Pratiques Ingénierie Dirigée par les Modèles Tâche 1: Spécification des concepts IACA”). Un problème est que ce seul métamodèle permet la construction de modèles incohérents. Par exemple, il peut être nécessaire de vérifier si un port d’entrée est bien connecté à un et un seul port de sortie, au contraire du processus monté dans la Figure 1. Un autre type exemple d’incohérence est la liaison d’un port d’événement avec un port de données. Habituellement, ces règles de cohérences sont exprimées dans un langage de contraintes comme OCL.

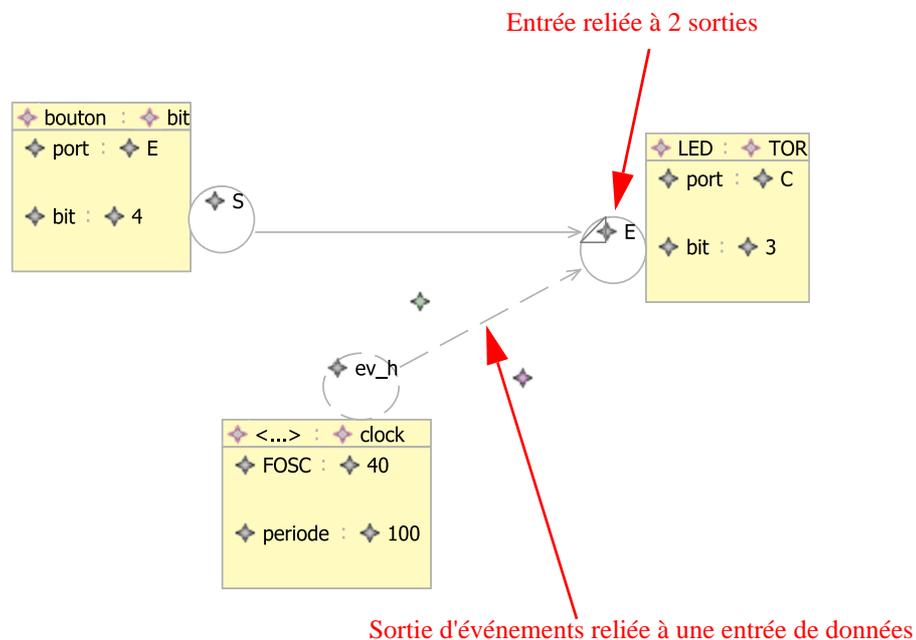


FIGURE 1. Exemple de Modèle de Processus Incohérent

## 2. Travail demandé

Je vous demande de décrire autant de règles de cohérence que vous pouvez l'imaginer pour le métamodèle que vous avez réalisé à la tâche 1. Le but est de prévenir au mieux la définition de modèles faux. Pour se faire, vous utiliserez le langage Check d'openArchitectureWare, langage proche d'OCL. Vous n'hésitez pas à définir des fonctions d'aide en utilisant le langage Xtend.

Vous produirez également une série de modèles, instances de votre métamodèle produit en tâche 1, qui seront le sujet de vérifications de cohérence. Ces modèles devront être mal formés et démontrer l'efficacité de vos règles de cohérence.

## 3. Ressources et astuces

Vous pourrez continuer le tutoriel <http://moogli/Fondement/2007-8/IMM/openArchitectureWare-42-reference.pdf>, sections I.1.9 à I.1.10. La référence des langages Check et Xtend se trouvent dans le même document en section II.6.

Un exemple de fichier de configuration openArchitectureWare se trouve sur <http://moogli/Fondement/2007-8/IMM/genC.txt>. Des exemples de spécifications Xtend et Check pour le langage des diagrammes d'états sont disponibles sur le site du cours. La clause import correspond au nom du métamodèle des modèles à vérifier (soit le nom de son paquetage principal. Bien que vous y trouverez la plupart des opérations d'itération (comme select, reject, collect), Check n'est pas OCL. Sa syntaxe ressemble à celle de Java. Par contre, vous ne trouverez pas l'opération allInstances. Vous aurez à naviguer à travers votre modèle pour parcourir ses instances. Pour vous y aider, sachez qu'un objet peut produire son conteneur par l'opération eContainer.

Pour pouvoir utiliser Check ou Xtend, vous aurez peut-être besoin de référencer le plug-in org.openarchitectureware.dependencies (dans la propriété Dependencies de META-INF/MANIFEST.MF).

Un problème qui apparaît souvent avec openArchitectureWare, comme pour toute utilisation d'EMF hors contexte, est de retrouver le métamodèle d'un modèle qu'on souhaite charger. Une solution (un peu radicale) est de modifier dans le XMI du modèle l'URI du métamodèle. Par exemple, dans la balise racine d'un fichier modèle, si l'URI que vous avez définie pour votre métamodèle est `http://dummy/iaca`, vous trouverez:

```
<iaca ... xmlns:iaca="http://dummy/iaca">...</iaca>
```

Il suffit de remplacer ladite URI par l'emplacement réel du métamodèle. Pour cet exemple, on trouvera donc plutôt:

```
<iaca ... xmlns:iaca="file:/D:\\Users\\toto\\TP-IMM\\eclipseWS\\iaca.mm\\model\\iaca.ecore">...</iaca>
```

De même, il peut arriver que vos contraintes ne parviennent pas correctement interroger un fichier référencé. Typiquement, le fichier de la librairie standard (que

nous appellerons `iaca.iaca`) peut ne pas être accessible par le moteur de vérification de contraintes pour un modèle de processus (que nous appellerons `process.iaca`). Il est possible de corriger ce problème en modifiant les références relative en références absolues, dans notre cas dans le fichier `process.iaca`. Il faudra donc remplacer toute occurrence de

```
iaca.iaca
```

par

```
file:/D:\\Users\\toto\\TP-  
IMM\\eclipseWS\\iaca.mm\\model\\iaca.iaca
```

Un workflow openArchitectureWare définit un processus de traitement de modèle, à l'instar d'un makefile ou d'un fichier ant. Vous avez un exemple d'un tel fichier sur <http://moogli/Fondement/2007-8/IMM/genC.txt>.

## 4. Délivrables

- Votre métamodèle sous forme `ecore`.
- Les spécifications Check et Xtend.
- Un script de lancement `oaw`.
- Votre librairie standard IACA.
- Divers modèles de processus exposant les erreurs que vous vérifiez.

Le tout sera empaqueté dans un projet Eclipse au format archive ZIP et accompagné d'un fichier explicatif (par exemple `lisez-moi.pdf`). Il doit être suffisant de lancer le fichier `oaw` pour lancer les tests.