

Réalisation d'un éditeur et d'une transformation pour le langage Divine à l'aide de Xtext

Yann Thierry-Mieg, Yann.Thierry-Mieg@lip6.fr

Projet pour deux étudiants. Il est recommandé de postuler en binôme avec un étudiant avec qui vous vous entendez bien.

Introduction

L'objectif du stage est de réaliser un plugin Eclipse s'appuyant sur XText offrant des fonctionnalités d'édition avancées (auto-complétion, quick-fix...) pour un langage dédié à la modélisation de processus concurrents : Divine.

Outre l'éditeur Xtext, le métamodèle développé servira de base à une transformation vers le langage GAL, permettant l'analyse. Les règles de transformation à appliquer ont déjà été définies dans un prototype donc il s'agit de les porter sur les technologies EMF.

Objectifs :

1. Construction de l'éditeur Divine avec XText
2. Intégration dans le serveur de build de l'équipe, permettant la mise en ligne d'un update-site pour installer l'éditeur
3. Implémentations de la transformation vers GAL
4. Tests sur un benchmark
5. Améliorations diverses (quick-fix...)
6. Documentation de l'outil

Exemple Divine:

```
channel Sin, Sout, toK, fromK, toL, fromL, timeout, Rout;  
channel shake, shakePC;
```

```
byte prod_n;
```

```
process Producer {  
  byte result,n;  
  state ready, start_send, wait_result, check;  
  init ready;  
  trans  
    ready -> start_send { effect prod_n = 1; },  
    ready -> start_send { effect prod_n = 2; },  
  
  start_send -> wait_result {sync Sin!prod_n;},
```

```
wait_result -> check {sync Sout?result;},
check -> ready {guard result == 1; sync shakePC!;},
check -> start_send { guard result == 2 || result == 3;};
}

process Consumer {
// etc.
}
```

Pré-requis et apports du projet :

La connaissance de Java est nécessaire, une connaissance du point de vue utilisateur d'Eclipse est souhaitable.

Le stage permettra de découvrir comment définir un plugin Eclipse et comment utiliser XText pour définir un langage spécifique à un domaine (DSL). C'est aussi une bonne introduction à l'analyse formelle du comportement des systèmes concurrents.

L'outil sera développé en utilisant des outils classiques du développement collaboratif : gestion de versions (svn), serveur d'intégration continue (teamcity), conventions de codage (nommage, commentaires...), métriques de qualité de code (Sonar) et permettra donc de se familiariser avec ces outils. Le benchmark de tests est conséquent et fixera l'objectif au long du stage.

Si la réalisation est de bonne qualité, le projet sera intégré dans la plateforme développée au sein de l'équipe Move, et distribué publiquement. Le stage sera encadré par une réunion hebdomadaire.

Références :

XText, un outil pour construire des DSL (domain specific languages) : www.xtext.org

Sonar, outil d'analyse de qualité du code : <http://www.sonarsource.org/>

Teamcity, serveur d'intégration continue : <http://www.jetbrains.com/teamcity/>

ITS et GAL : <http://ddd.lip6.fr/>