

TP4 : Code utilisateur et sous-processus

Ce TP permet de mettre en pratique l'implémentation de code utilisateur dans les processus, de sous-processus et d'une exécution parallèle.

Présentation des données fournies

Importer les sources fournies dans un projets jBPM nommé TP4.

Les classes de tests unitaires sont fournies. L'objectif du TP est de modéliser un processus qui permet la bonne exécution des cas de test.

Le processus de départ est celui du TP précédent.

Processus

1. Activité custom et traitement des Exceptions

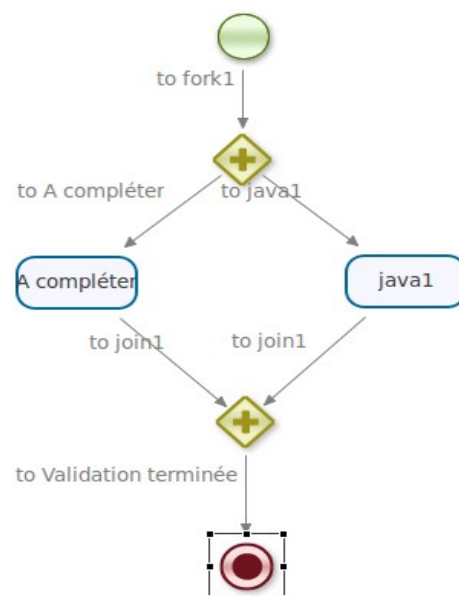
Dans le processus principal, lorsqu'un jeton atteint l'état final « Accordé », nous voulons pouvoir exécuter une action simulant l'appel à un service externe de type Webservice.

L'URL du service web est un attribut de l'action à implémenter.

Cette activité gère les Exceptions ; dans le sens ou le webservice n'est pas accessible, elle route l'exécution vers un état de fin en erreur sinon elle route l'exécution vers l'état de fin « Accordé ».

2. Sous-processus

Dans le cas d'un gros débit, nous voulons déclencher le sous-processus suivant :



Ce sous-processus est composé de deux branches s'exécutant en parallèle :

- La première branche contient un état « A compléter ». Cet état permet de renseigner une variable d'instance nommé *avis* qui sera remonté dans le processus parent. Cette variable prend une valeur entière
- La seconde branche exécute automatiquement une Activité java qui simule le calcul d'une variable d'instance nommé *incidents* (entier également). Cette variable est également remontée dans le processus parent

Ainsi, à la fin du sous-processus, deux variables d'instance sont positionnées. La suite du processus parent dépend de ces deux variables.

- Si *incidents* > *avis*, l'opération est refusée
- Sinon, l'opération est accordée

3. EventListener

Pour tous les noeuds du processus parent, nous voulons tracer sur la console certaines informations à l'entrée et à la sortie du noeud.

Les informations à afficher sont :

- L'id de l'activité
- Les valeurs des variables du processus

Modéliser le processus avec jBPM Designer afin que les tests JUnit fournis passent.