

## TP3 : Modélisation

Ce TP permet de se familiariser avec différents types de noeuds : Script, Timer, Subprocess, Rule, diverge, converge.

### Étapes

#### 1. Création de projet

Utiliser l'assistant pour créer un projet jBPM nommé TP3

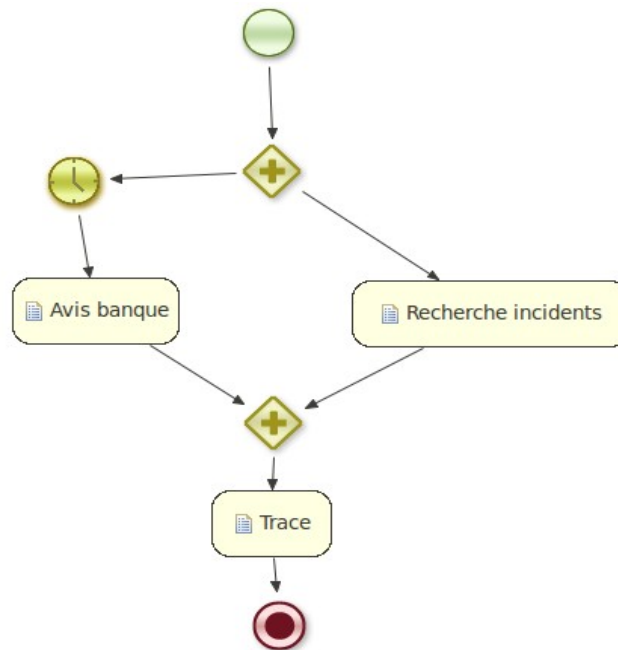
#### 2. Branches parallèles

Créer un fichier processus dans le répertoire ressources nommé *tp3/subprocess.bpmn*

Ce processus modélisera un demande de crédit à une banque.

Le processus est composé de 2 branches parallèles :

- La première branche consistera, à terme, en une tâche utilisateur (un banquier qui étudie la demande de crédit). Pour l'instant, nous modéliserons cette branche avec juste un nœud timer et un nœud script.
- La seconde branche modélisera une action automatique censé vérifier dans l'historique client les éventuels incidents bancaires. Pour l'instant, nous modéliserons cette branche par un simple nœud script.



Exécuter la classe Java fournie : *org.formation.tp3.test.TP3\_1test.java* et s'assurer du bon déroulement du processus

## 2. Sous-processus et traitement des erreurs

Créer un fichier processus dans le répertoire ressources nommé *tp3/Main.bpmn*

Ce processus démarre le sous-processus précédemment modélisé

A la sortie du sous-processus, nous allons systématiquement déclencher un événement erreur.

Cet événement erreur devra déclencher un gestionnaire d'erreur.

Tester le bon déroulement du processus en exécutant la classe Java fournie :

*org.formation.tp3.test.TP3\_2test.java*

## 3. Nœud règle métier

Dans cette dernière partie de ce TP, nous allons utiliser un nœud règle métier qui permettra de démarrer le processus principal dès que la règle sera satisfaite.

Le processus sera donc composé de trois branches :

1. Une branche boucle qui périodiquement teste la présence d'un objet *Demande* dans la base de connaissance. Si une demande existe, un signal est émis déclenchant la branche principale
2. La branche principale est la branche faisant appel au sous-processus
3. La branche de traitement d'erreur modélisé précédemment.

Récupérer la classe *Demande.java* et le fichier de règles *rules.drl* fournis.

Modifier la branche principale afin qu'elle soit démarrée par un signal particulier

Modéliser la branche boucle et faire correspondre le nœud Rule au *ruleflow-group* du fichier de règle

Enlever les états terminaux afin que la branche boucle soit infinie.

Exécuter la classe *org.formation.tp3.TP3\_3test* fournie pour valider votre modélisation.