

Rapport

Hamid Oukhnini

Contenu du rendu:

- PetriNet.ocl: présentant les différentes contraintes du réseau petriNet
- SimplePDL2PetriNet.java: le code de la transformation en Java permettant de traduire un modèle de processus en un réseau de Petri.
- rapport
- des exemples de modèles PetriNet sans erreur (net*) ou avec erreurs (ko-net*).

Explication des contraintes principales ajoutées au réseau de Petri :

— Toute Place doit avoir un nombre entier naturel de Jetons, en fait, ceci découle de la définition du réseau de Petri.

context *Place*

inv marquageInitial : **self**.tokens >= 0

— Le nombre de Jetons que transporte un arc doit être un entier naturel.

context *arc*

inv : **self**.weight >= 1

— Un arc ne peut pas lier deux Places, en fait, il lie une Place avec une Transition:

context *arc*

inv arcValid : **self**.source.ocIsTypeOf(Place) <> **self**.cible.ocIsTypeOf(Place)

and self.source.ocIsTypeOf(Transition) <> **self**.cible.ocIsTypeOf(Transition)

Principe suivi pour l'implantation de SimplePDL2PetriNet.java:

pour implanter SimplePDL2PetriNet.java nous avons suivi les trois règles suivantes:

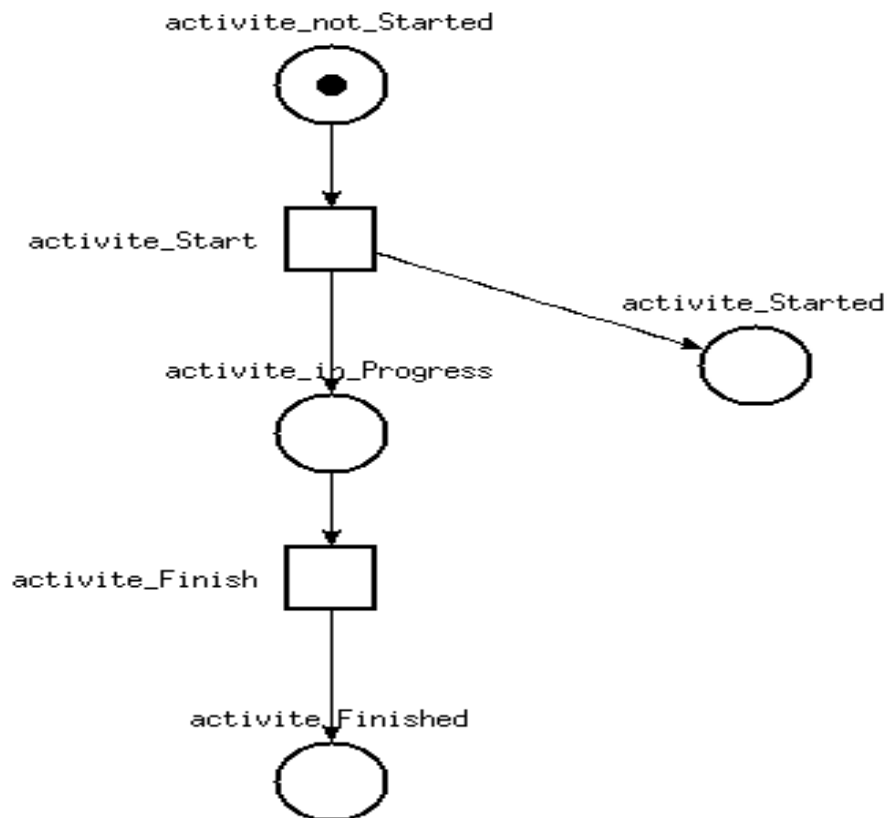
La première traduit chaque activité (WorkDefinition) en quatre places caractérisant son état (NotStarted, Started, InProgress et Finished) et deux transitions (Start et Finish).

L'état Started permet de mémoriser qu'une activité a démarré. Ces places et transitions sont utilisées dans la deuxième règle pour traduire chaque WorkSequence en un read-arc entre une place de l'activité source et une transition de l'activité cible, la place et la transition dépendant de la valeur de l'attribut kind.

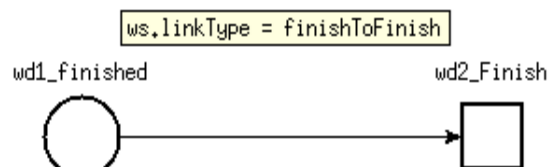
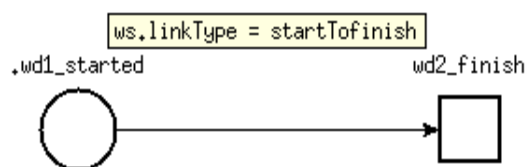
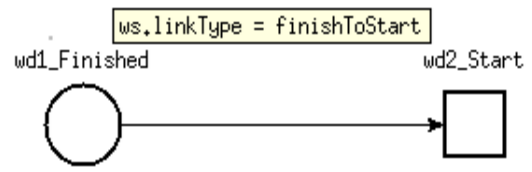
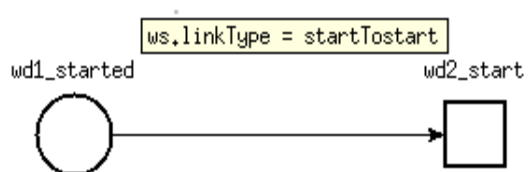
La dernière règle traduit un Process en un PetriNet regroupant les nœuds et arcs construits par les autres règles.

La traduction d'une WorkDefinition activité **<<WorkDefinition>>** en réseau de Petri est la suivante:

activité



La traduction d'une WorkSequenece ws en réseau de petri nous avons se basé sur le principe illustré dans la figure suivante:



- * Les ressources de leurs part sont transformés en places avec comme marquage initial la quantité disponible des ressources et les Allocates qui permettent l'accès à ces ressources sont transformés en arcs.
- * L'implémentation de la transformation passe par les étapes suivantes :
 - Chargement des packages SimplePDL et PetriNet afin de l'enregistrer dans le registre d'Eclipse.
 - On récupère le premier élément du modèle process(élément à la racine), puis instancier la fabrique.
 - On commence par construire le PetriNet, en convertissant les workDefinitions et Resources en Places, puis workSequences en Arcs.