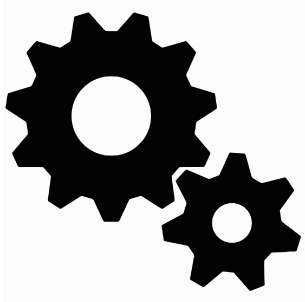
ENSEEIHT : Génie du Logiciel et des Systèmes (GLS)

Rapport de projet

GLS : Rapport de projet



Matthieu Pizenberg

20 décembre 2013

Table des matières

Chapitre 1

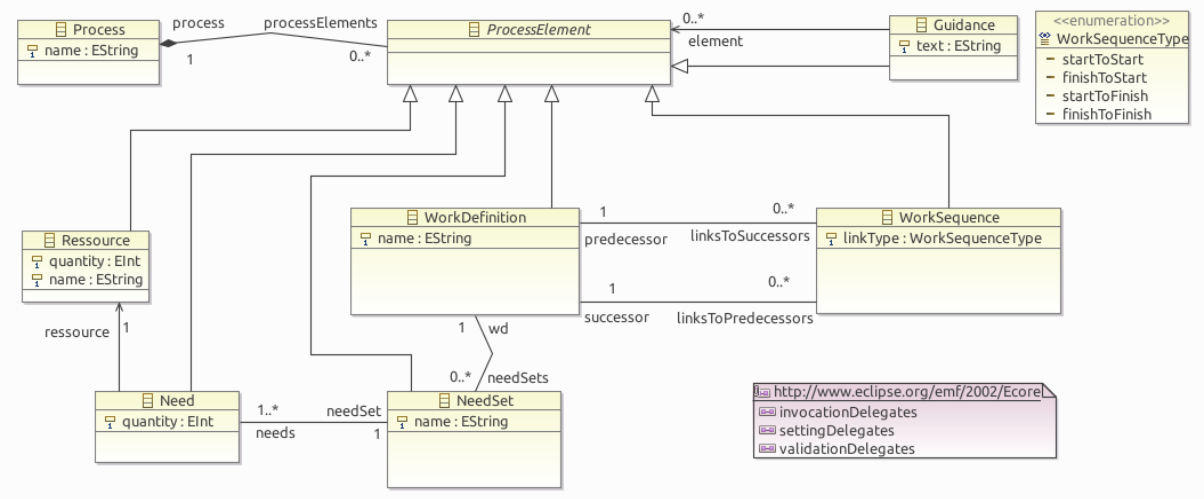
Les métamodèles SimplePDL et PetriNet

Pour l'ensemble de la chaine de transformation on aura besoin des métamodèles SimplePDL et PetriNet, on définit donc ces deux métamodèles ainsi que des exemples de modèles qui serviront à faire les tests.

1.1 SimplePDL

On a ajouté au modèle SimplePDL la gestion des ressources. Pour celà, on a introduit :
- Ressource : une ressource ayant un nom et étant présente en une certaine quantité
- Need : un besoin représentant un nombre de ressources d'un certain type
- NeedSet : un ensemble de besoins nécessaires à la réalisation d'une WD

Voilà un diagramme représentant le métamodèle SimplePdl que j'ai utilisé :

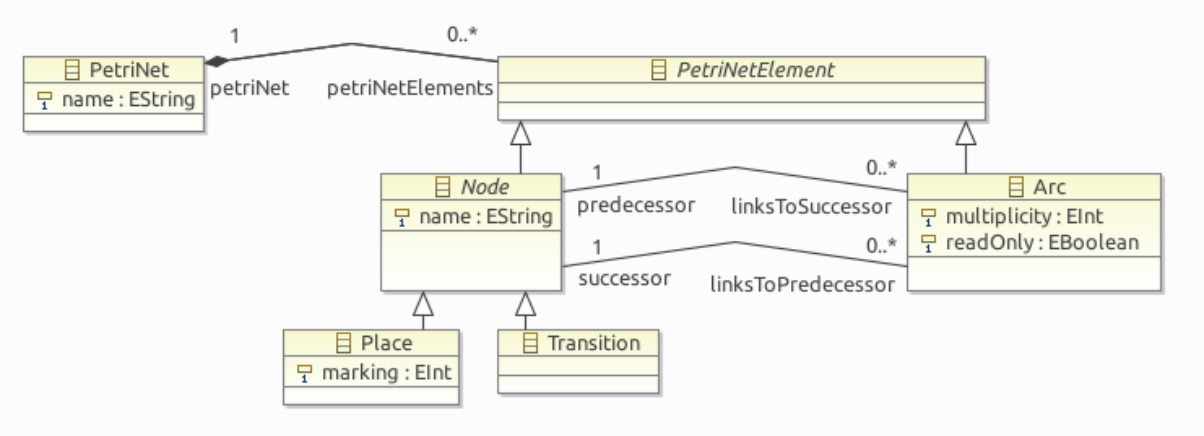


Ce modèle n'est pas suffisant pour décrire entièrement l'ensemble des contraintes. Pour celà on a donc ajouté des contraintes OCL avec OCLinEcore.

On a ajouté les règles suivantes :
- class Process
  - sameWDName : les WD doivent avoir des noms différents
  - sameRessourcesName : les ressources doivent avoir des noms différents
  - nameForbidden : un processus ne peut pas s'appeler : "Process"
  - sameNeedSetsName :
- class WorkDefinition
  - voidName : nom vide interdit
- class WorkSequence
  - previousWDinSameProcess : la WD précédente doit être dans le même processus
  - reflexivity : le prédécesseur et le successeur oivent être différents
  - nextWDinSameProcess : la WD suivante doit être dans le même processus
- class Ressource
  - voidName : nom vide interdit
  - positiveQuantity : la quantité de ressources doit être $>= 0$
- class Need
  - positiveQuantity : la quantité de ressources d'un besoin doit être > 0
- class NeedSet
  - voidName : nom vide interdit

1.2 PetriNet

Le modèle PetriNet que j'ai utilisé est représenté par le diagramme suivant :



Ce modèle n'est pas suffisant pour décrire entièrement l'ensemble des contraintes. Pour celà on a donc ajouté des contraintes OCL avec OCLinEcore.

On a ajouté les règles suivantes :
- class PetriNet
  - voidPetriName : nom vide interdit
  - sameNodeName : les noms des noeuds doivent être différents
- abstract class Node
  - voidNodeName : nom vide interdit
- class Arc
  - positiveMultiplicity : la multiplicité doit être > 0
  - sameTypeOfPredecessorAndSuccessor : Prédecesseur et successeur doivent être différents
  - nextNodeNotInSamePetriNet : les noeuds reliés sont dans le même réseau de pétri
  - previousNodeNotInSamePetriNet :les noeuds reliés sont dans le même réseau de pétri
- class Place
  - positiveMarking : le marquage d'une place est $>= 0$

1.3 Exemples de modèles

Voici quelques exemples de modèles SimplePDL (sans ressources) :

Chapitre 2

Transformations modèle à modèle (M2M)

2.1 L'outil ATL

2.2 SimplePDL

2.3 PetriNet

Chapitre 3

Transformations modèles à textes (M2T)

3.1 L'outil Acceleo

3.2 SimplePDL

3.3 PetriNet

Chapitre 4

Transformations textes à modèles (T2M)

4.1 L'outil Xtext

4.2 SimplePDL

4.3 PetriNet