



Ingénierie Dirigée par les Modèles

Mini-Projet IDM

Groupe L34-02

Élèves :

THEVENET Louis

SABLAYROLLES Guillaume

Octobre 2024

Contents

1. Méta-Modèles	3
1.1. SimplePDL	3
1.2. PetriNet	3
2. Transformation Modèle à Modèle	5
2.1. SimplePDL vers PetriNet	5
3. Transformation Modèle à Texte	5
4. Transformation Texte à Modèle	5

1. Méta-Modèles

1.1. SimplePDL

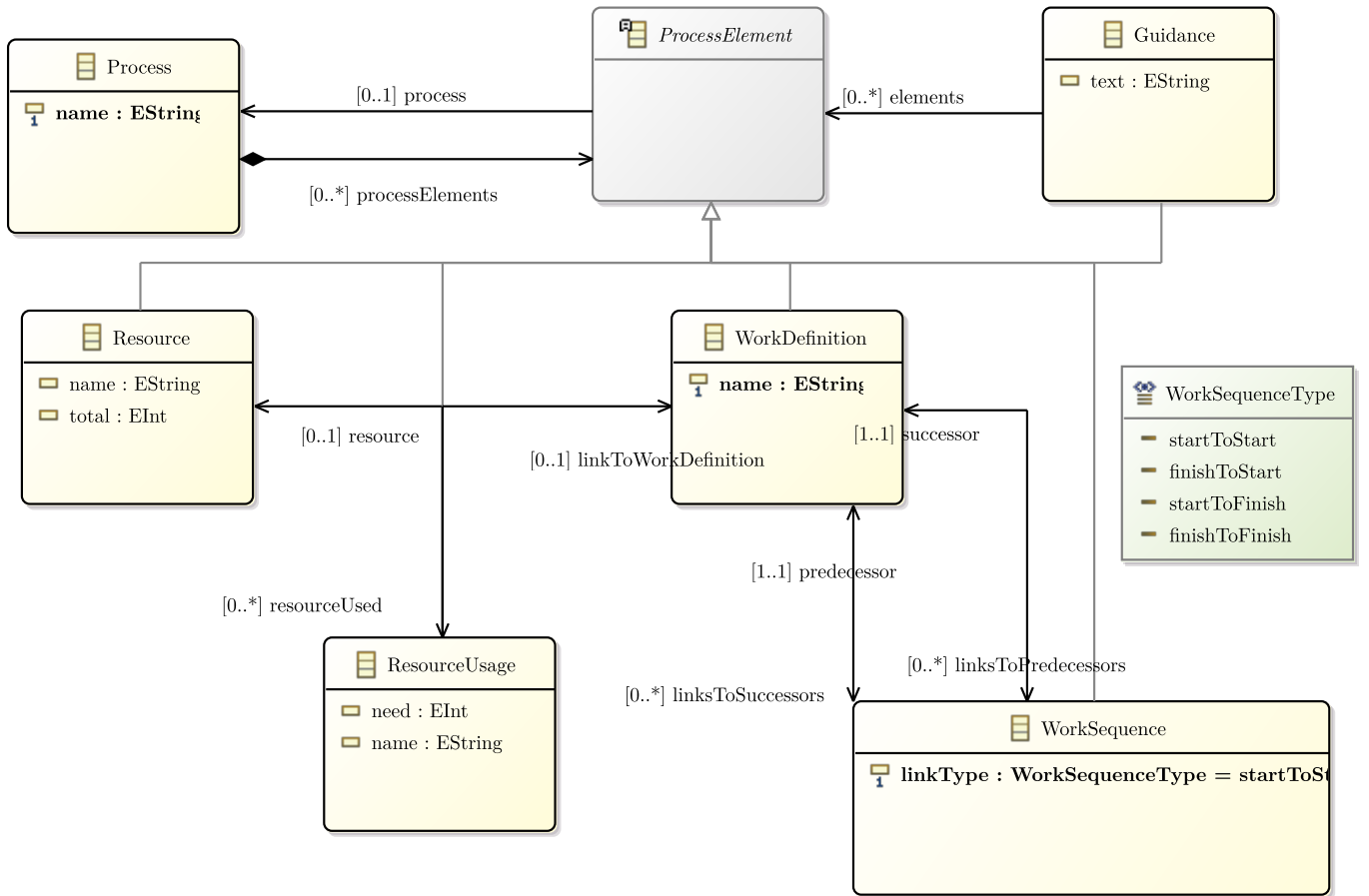


Figure 1: Méta-Modèle de SimplePDL

A partir du méta-modèle fourni, nous avons rajouté l'utilisation des ressources. Une ressource est défini comme une EClass Resource ayant un nom (**name**) et un nombre total d'éléments disponible pour cette ressource (**total**).

Ensuite, Resource peut être manipulé par les WorkDefinition grâce aux ResourceUsage : les ResourceUsage définissent le nombre d'une Resource demandées (**need**) par la WorkDefinition ainsi que le nom de la Resource demandée (**name**).

L'utilisation de ResourceUsage nous permet de partager l'utilisation d'une même Resource entre différente WorkDefinition.

1.2. PetriNet

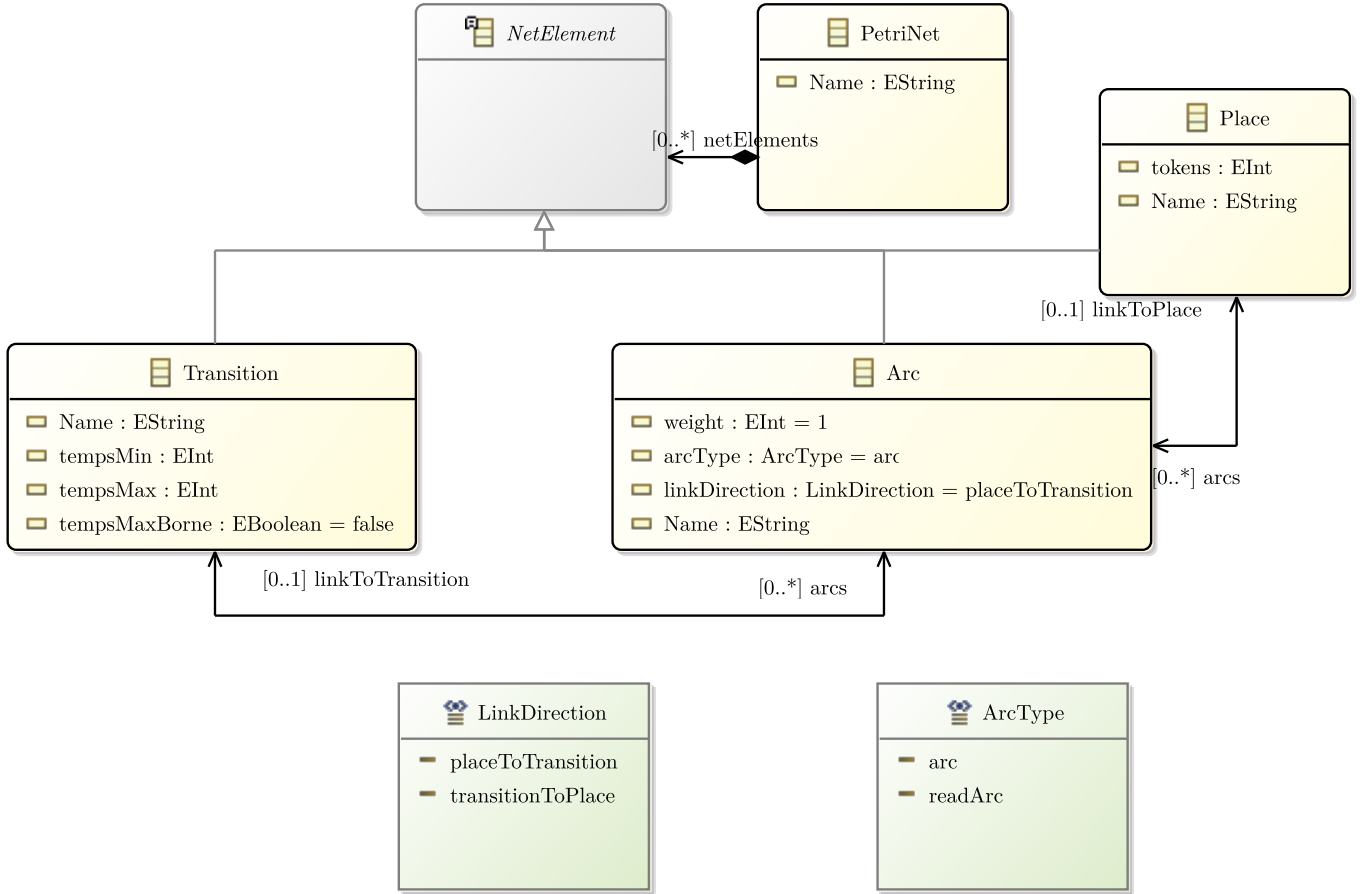


Figure 2: Méta-Modèle de PetriNet

Pour définir un PetriNet, nous avons repris le principe de SimplePDL : un réseau **PetriNet** est un ensemble de **NetElements** où les **NetElements** représentent les éléments d'un réseau de Petri : **Place**, **Transition** et **Arc**.

Une **Place** est définie par un nom (**name**) et le nombre de jetons présents (**tokens**).

Pour définir une **Transition** il faut qu'elle est un nom (**name**) et l'intervalle de temps sur laquelle elle est franchissable (entre **tempsMin** et **tempsMax**, l'option **tempsMaxBorne** indique si cette transition est franchissable pendant une infinité de temps après le début possible).

Enfin, pour relier une place et une transition (ou transition et place), il faut utiliser des **Arc**. Un **Arc** est paramétré par son nom (**name**), un poids (**weight**, initialisé à 1), par son type (**arcType**) permettant de le définir comme un arc ou un read arc (défini dans l'enum **ArcType**) et, le plus important, sa direction (**linkDirection**) pour différencier **Place** → **Transition** et **Transition** → **Place** (défini dans l'enum **LinkDirection**).

Différencier les arcs, que ce soit pour les read arcs ou les directions, est un point clé de ce méta-modèle pour les modèles conformes, permettant de les distinguer et d'effectuer des traitements plus rapidement sur eux.

2. Transformation Modèle à Modèle

2.1. SimplePDL vers PetriNet

2.1.1. Java/EMF

Java et EMF nous permettent d'obtenir des structures arborescentes de nos modèles.

Pour réaliser notre transformation SimplePDL vers PetriNet nous nous sommes basés sur l'algorithme donné :

- Une `WorkDefinition` devient 4 Places et 2 Transitions
- Une `WorkSequence` devient un `ReadArc` entre les places \rightarrow transition `Started` \xrightarrow{r} `Start` et `Finished` \xrightarrow{r} `Finish`

pour pouvoir déterminer comment implémenter la transformation d'une `Resource` en PetriNet.

D'abord une `Resource` possède un certains nombre d'éléments (que les `WorkDefinition` utilisent pour fonctionner) donc nous avons eu l'idée de transformer `Resource` en une `Place(name: Resource.name, tokens: Resource.toal)` ie une place avec le nombre de jetons égal au nombre de ressources disponibles.

2.1.2. ATL

3. Transformation Modèle à Texte

4. Transformation Texte à Modèle