Méta-modélisation

Attention : Version d'Eclipse à utiliser : `/mnt/n7fs/ens/tp_cregut/eclipse-gls/eclipse`

Important : Pour tous les TP IDM, il faudra utiliser les mêmes mémamodèles (les vôtres) et les mêmes projets Eclipse. Vous les ferez évoluer et les complèterez au fur et à mesure des TP...

1 Comprendre les outils de métamodélisation d'Eclipse EMF

L'objectif de ces premiers exercices est de prendre en main les outils proposés par Eclipse Modeling Project pour la métamodélisation. En fait, nous utiliserons surtout les outils développés dans le cadre d'Eclipse EMF (Eclipse Modeling Framework) et l'éditeur graphique [1] Ecore.

Nous nous appuyons sur un langage simplifié de modélisation de processus de développement, appelé SimplePDL. Son métamodèle Ecore est présenté à la figure 1.

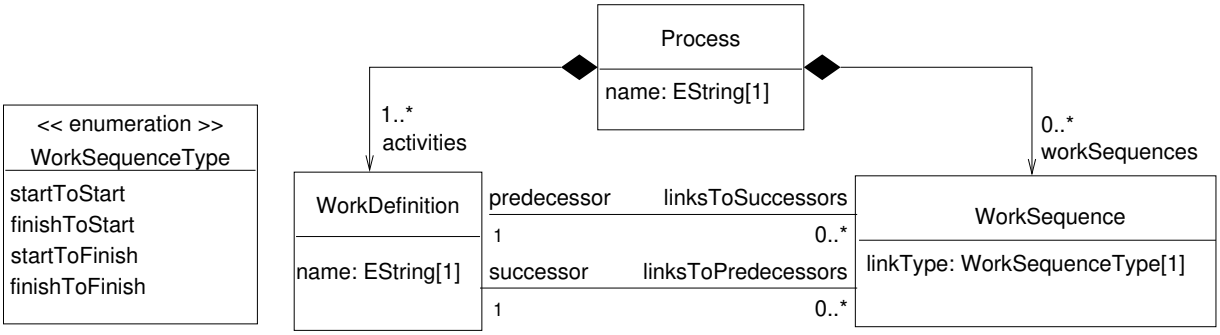

FIGURE 1 – Métamodèle initial de SimplePDL

Exercice 1 : Préparation

Nous allons commencer par créer un projet et y placer le métamodèle fourni.

1.1. *Créer un projet sous Eclipse.* Comme toujours sous Eclipse, il faut commencer par créer un projet. On peut faire *File / New / Project...*, puis sélectionner « General / Project ». Après avoir donné un nom au projet, par exemple *fr.n7.simplePDL*, faire *Finish*. Le projet est créé et apparaît dans le navigateur de projets (« Navigator »), généralement sur la gauche.

1.2. *Sélectionner la perspective Ecore.* Pour sélectionner la perspective Ecore, on peut faire *Window / Perspective / Open perspective / Other...*, puis cliquer « Ecore ». La liste des perspectives déjà ouvertes est affichée en haut à droite de l'atelier Eclipse sous la forme d'icônes, une par perspective. Cliquer sur l'icône active la perspective. Pour ouvrir une nouvelle perspective, on

1. Cet éditeur graphique a été développé avec l'outil Sirius d'Obeo et a été contribué au projet EMF. Sirius sera étudié lors de la définition de syntaxes concrètes graphiques.

peut cliquer sur l'icône la plus à gauche (*Open Perspective*) puis sélectionner dans la liste la perspective choisie (Ecore dans notre cas).

La perspective Ecore est composée de l'éditeur (ici l'éditeur Ecore) dans la partie centrale, avec en dessous une partie avec la vue « *Properties* » listant et permettant de modifier les propriétés de l'élément sélectionné sur l'éditeur. Sur la gauche, il y a le navigateur des projets en haut et l'*Outline* en bas. L'*Outline* présente, sous une forme arborescente, une vue complète du modèle dont l'un des diagrammes est en cours d'édition. Un diagramme est une vue du modèle.

1.3. *Récupérer le modèle fourni.* Mettre le modèle Ecore de SimplePDL fourni, `SimplePDL.ecore`, dans le projet créé à la question 1.1.

Il suffit par exemple de sélectionner le projet « fr.n7.simplePDL », faire un clic droit pour sélectionner « *import* », puis *General / File System*. Attention, il faut d'abord sélectionner le dossier depuis lequel l'import sera fait (on peut utiliser *Browse*) puis, une fois le dossier sélectionné, cocher les fichiers à importer (partie droite de la boîte de dialogue).

Exercice 2 : Visualiser le métamodèle

Le métamodèle de SimplePDL est fourni sous la forme d'un fichier Ecore. Voyons comment en consulter le contenu.

2.1. *Avec un simple éditeur texte.* Le fichier Ecore n'est pas très... lisible ! Pour s'en convaincre, il suffit de l'ouvrir dans un éditeur texte, par exemple en choisissant « Text Editor » pour « open with » dans le menu contextuel.

2.2. *Avec l'éditeur arborescent Ecore d'EMF.* Nous pouvons aussi le visualiser sous forme arborescente en utilisant le « Sample Ecore Model Editor ». Utilisable mais pas forcément des plus pratique...

2.3. *Avec l'éditeur graphique Ecore de EMF.* Sélectionner le fichier Ecore SimplePDL.ecore, faire un clic droit dessus, choisir « Initialize Ecore Diagram... », accepter le nom par défaut en cliquant sur *Next*, choisir la représentation graphique classique du diagramme de classe en cliquant sur « Entities in a Class diagram » puis sur *Next*, et enfin cliquer sur *Finish*. Ceci crée un « Representation file » (`*.aird`).

Une fenêtre apparaît avec une consigne indiquant d'effectuer un double-clic pour importer le contenu du paquetage à la racine du métamodèle. On peut ensuite déplacer les éléments à la souris pour rendre le diagramme plus lisible (en particulier les relations bi-directionnelles liant *WorkDefinition* et *WorkSequence*.

Modifier la position des éléments pour retrouver un métamodèle SimplePDL proche de celui donné à la figure 1.

Exercice 3 : Valider un modèle Ecore

À tout moment, il est possible de vérifier la validité du métamodèle par rapport aux règles Ecore. En fait, on vérifie que le fichier Ecore contient bien un modèle conforme au métamodèle Ecore. Vérifions maintenant la correction du métamodèle que l'on vient d'écrire. Commencer par sauver le métamodèle.

3.1. *Depuis l'éditeur graphique.* La vérification est faite automatiquement et les éléments incorrects sont affichés en rouge. On peut aussi faire un clic droit et choisir « Validate diagram ». Attention, rien ne s'affiche si le modèle est valide.

3.2. *Depuis l'éditeur arborescent.* Après avoir ouvert le modèle Ecore dans l'éditeur arborescent (voir question 2.2), il suffit de se placer sur l'élément racine (le paquetage *simplepdl*), sélectionner *Validate* après avoir fait un clic droit. La vérification est lancée. *Validate* est disponible dans le menu contextuel des différents éléments du modèle Ecore mais il ne vérifie que cet élément (et ses sous-éléments).

Exercice 4 : Créer un modèle avec l'éditeur réflexif
Pour saisir un modèle conforme à un métamodèle, on peut procéder comme suit :

1. ouvrir le métamodèle (avec l'éditeur arborescent),
2. déplier les éléments du métamodèle,
3. faire un clic droit sur l'élément qui sera la racine du modèle (Process dans notre cas),
4. sélectionner « *Create dynamic instance...* » dans le menu contextuel,
5. indiquer le nom du fichier (extension .xmi) et faire « *Finish* ».

Quand on déplie la première ressource, on voit apparaître une instance de l'élément racine (Process). On peut alors utiliser la vue *propriétés* pour renseigner les attributs et les références de l'élément sélectionné. En faisant un clic droit sur un élément, on peut ajouter un fils ou un frère (voir les relations de composition sur le métamodèle).
4.1. Saisir un modèle de procédé à l'aide de cet éditeur. Valider le modèle de processus vis à vis de son métamodèle en utilisant le menu contextuel pour choisir *Validate*.

2 Compléter le métamodèle de SimplePDL

Notre but est maintenant de compléter le métamodèle de SimplePDL. Nous proposons un nouveau métamodèle (figure 2) qui a deux modifications principales :

1. la notion de *ProcessElement* a été ajoutée comme généralisation de *WorkDefinition* et *WorkSequence*. Un processus est donc un ensemble d'éléments de processus qui sont soit des activités, soit des dépendances.
2. la notion de *Guidance* a été également ajoutée. C'est l'équivalent d'une annotation UML : elle permet d'associer un texte à plusieurs éléments d'un processus.

Remarque : La notion de composition est décrite par la propriété *Containment*. L'héritage est matérialisé dans la sous-classe via la propriété *ESuperType* qui liste les super-classes.

Exercice 5 Discuter les avantages/inconvénients du nouveau métamodèle.

Exercice 6 Modifier [2] le métamodèle fourni pour qu'il corresponde à celui de la figure 2. Utiliser l'option validate pour vérifier que le métamodèle de SimplePDL ne contient pas d'erreurs.
Attention : On peut supprimer un élément du diagramme sans le supprimer du modèle. Le diagramme n'est qu'une vue sur le modèle ! L'éditeur graphique propose deux icônes : la grosse

2. Attention à la différence entre *Delete from Diagram* et *Delete from Model*.

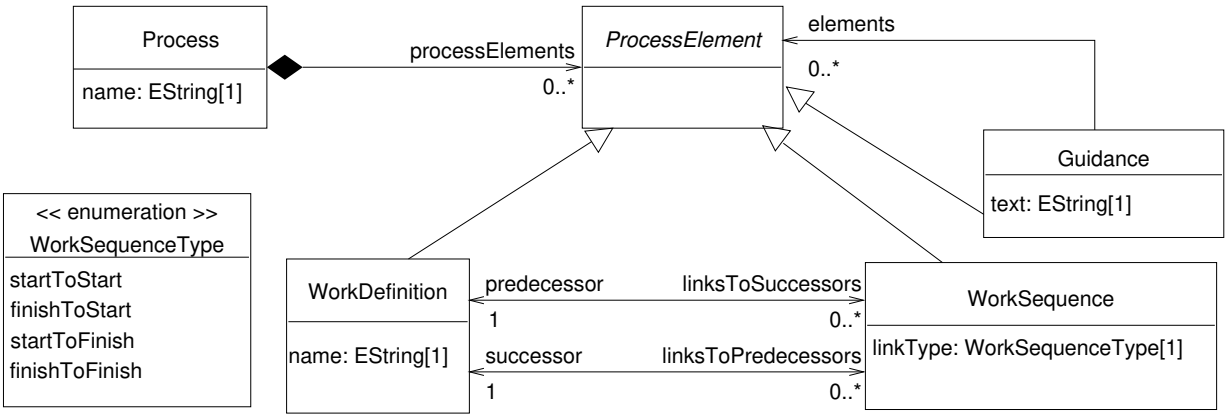

FIGURE 2 – Nouveau métamodèle de SimplePDL

croix rouge supprime uniquement du diagramme (*Delete from Diagram*) et une double croix rouge supprime l'élément du modèle (*Delete from Model*) et donc aussi de du diagramme.

Exercice 7 Ouvrir le modèle précédemment saisi. Que constate-t-on ? Pourquoi ?

Exercice 8 : (Mini-projet)
Compléter le métamodèle pour ajouter les ressources et leur utilisation par les activités.

3 Définir un métamodèle des réseaux de Petri

Définissons un métamodèle pour les réseaux de Petri. Il ne sera pas nécessaire de modéliser les aspects temporels qui ne seront pas utilisés dans la suite des TP et du mini-projet.

Exercice 9 Proposer un métamodèle pour représenter les réseaux de Petri et le définir avec Ecore. On créera un projet `fr.n7.petriNet`. On appelera le métamodèle `PetriNet.ecore`. Bien sûr, il faudra définir plusieurs modèles de réseau de Petri pour valider le métamodèle proposé.

Exercice 10 Est-ce que tous les modèles conformes au métamodèle *PetriNet* correspondent effectivement à des réseaux de Petri. Donner quelques contre-exemples. Est-il facile/possible de modifier le métamodèle pour exclure ces modèles indésirables ?