

TP1 : Métamodélisation avec Eclipse/EMF

Professeur : M. EL HAMLAOUI

Contexte :

L'objectif de ce TP est d'illustrer les techniques de l'Ingénierie Dirigée par les modèles. La préoccupation poursuivie consiste à permettre la manipulation de modèles de bases de données relationnelles, dans des espaces technologiques différents. Pour commencer, et manipuler des modèles de SGBDR, un DSL correspondant à un SGBDR doit être construit. Il doit permettre la création de modèles SQL sans passer par des outils tels que MySQL, Oracle ou autres. Dans un deuxième temps, pour les utilisateurs qui travaillent avec les bases de données NoSQL (système de gestion de base de données qui n'est pas fondé sur l'architecture classique relationnelle), une transformation du modèle SQL vers le modèle XML est obligatoire afin de pouvoir exploiter des outils tel que : BaseX¹. Cette transformation utilise un modèle pivot conforme au métamodèle KM3 (proposé par l'INRIA²). Ainsi, nous réalisons deux transformations, d'abord une transformation modèle vers modèle (M2M), SQL vers KM3, puis une transformation du modèle KM3 vers un modèle XML. Enfin, pour les utilisateurs travaillant avec des bases de connaissances, une transformation SQL vers OWL est nécessaire. L'ontologie résultante (modèle OWL) peut être utilisée par des outils de développement d'ontologie tels que Protégé³.

L'objectif de ces premiers exercices est de prendre en main les outils proposés par Eclipse Modelling Tools pour la méta-modélisation. Nous allons principalement utiliser les outils développés dans le cadre d'Eclipse EMF (Eclipse Modeling Framework).

Exercice 1: Préparation

Nous allons commencer par créer un projet EMF. La création d'un projet sous Eclipse peut se faire de la manière suivante : File / New / Other / Eclipse Modeling Framework / Empty EMF Project. Après avoir donné un nom au projet, par exemple *IDM*, faire Finish. Le projet est créé et apparaît dans le navigateur de projets (« Navigator »), généralement sur la gauche.

Exercice 2 : Création du métamodèle SQL

La première chose à faire maintenant est de créer un fichier Ecore. La création de ce fichier se fait en sélectionnant le dossier modèle au sein du projet EMF *IDM* puis par clic droit, de choisir New / Other / Eclipse Modeling Framework / Ecore Model. Après avoir donné un nom au métamodèle, par exemple *SQL*, faire Finish. Il faut ensuite sélectionner la perspective Ecore (en haut à droite de l'éditeur). Sélectionner le fichier Ecore *SQL* ; l'éditeur s'ouvre dans la partie

¹ <http://basex.org/home/>

² <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.95.4399&rep=rep1&type=pdf>

³ <http://protege.stanford.edu>

centrale, avec en dessous une partie correspondant à la vue « Properties » listant et permettant de modifier les propriétés de l'élément sélectionné dans l'éditeur.

La deuxième chose à faire est d'alimenter le fichier Ecore avec les concepts du métamodèle SQL : de deux façons possibles :

- Avec l'éditeur arborescent Ecore d'EMF. Nous pouvons aussi le visualiser sous forme arborescente en utilisant le « Sample Ecore Model Editor » (par défaut ou obtenu après sélection du modèle par « Open With »).
- Avec l'éditeur graphique Ecore de EMF. Il suffit pour cela de sélectionner le métamodèle Ecore *SQL.ecore* puis New / Ecore Diagram, vérifier que l'option « Create from an existing model » est sélectionnée ainsi que le modèle *SQL.ecore*. Après avoir vérifié que l'option « Initialize the diagram with existing model objects » est activée, cliquer sur Finish. L'éditeur graphique Ecore apparaît par la suite.

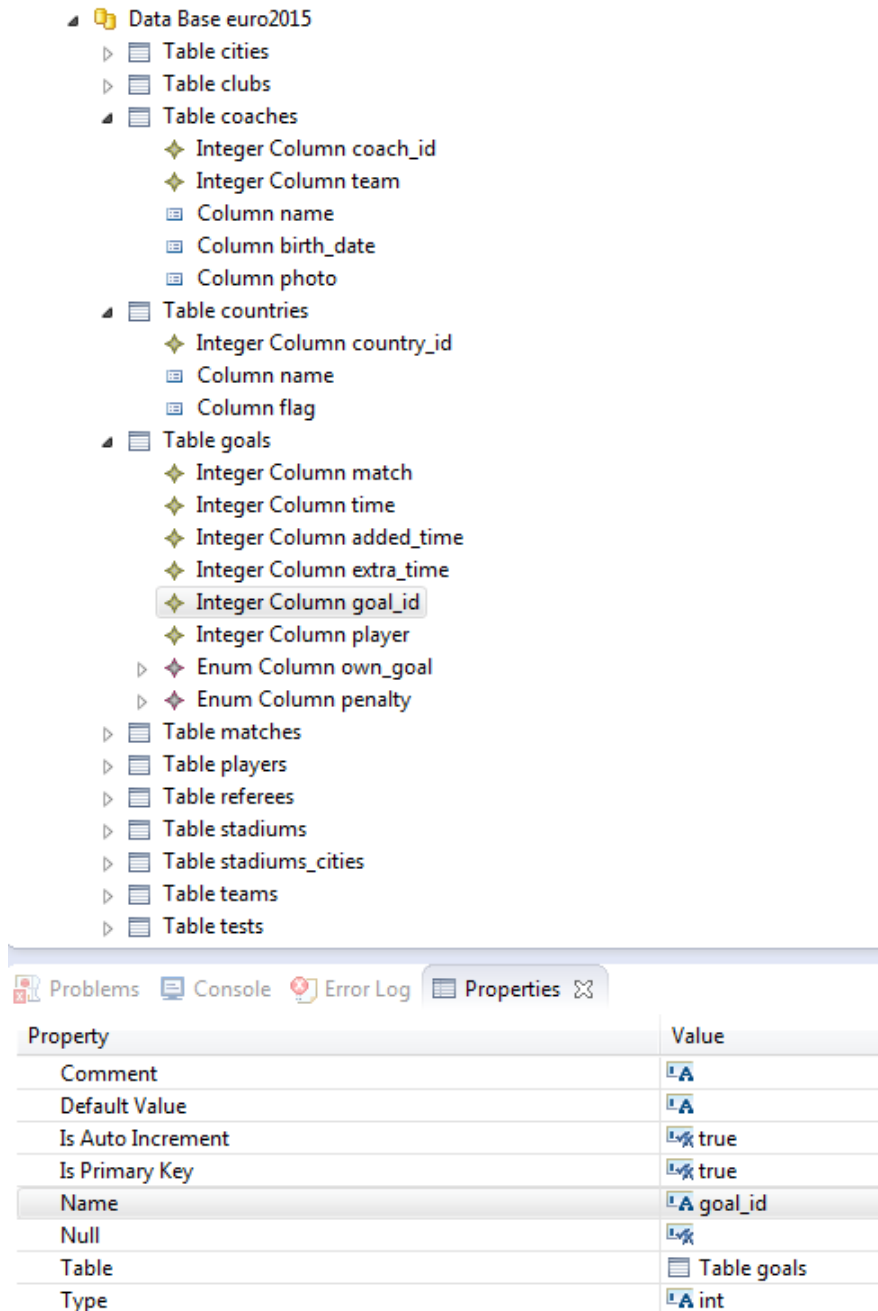
Exercice 3 : Valider un modèle Ecore

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.

- Depuis l'éditeur graphique. Il suffit de cliquer sur le symbole vert (Validate), à la fin de la barre des boutons.
- Depuis l'éditeur arborescent. Après avoir ouvert le modèle Ecore dans l'éditeur arborescent (voir question précédente), il suffit de se placer sur l'élément racine, sélectionner Validate après avoir cliqué à droite. La vérification est lancée.

Exercice 4 : Construire le métamodèle Ecore

Dans l'éditeur graphique, compléter le modèle *SQL.ecore* fourni, afin que les concepts présentés dans *modelSQL* soient supportés par le métamodèle.



Property	Value
Comment	
Default Value	
Is Auto Increment	<input checked="" type="checkbox"/> true
Is Primary Key	<input checked="" type="checkbox"/> true
Name	goal_id
Null	<input checked="" type="checkbox"/>
Table	Table goals
Type	int

Figure 1: modèle SQL

Exercice 5 : Engendrer le code Java et l'éditeur arborescent

Une fois le métamodèle construit, nous pouvons engendrer le code Java correspondant ainsi qu'un éditeur arborescent permettant de saisir un modèle conforme au métamodèle SQL.

- Sélectionner SQL.Ecore puis New / Other / Eclipse Modeling Framework / EMF Generator Model et faire Next. Sélectionner Model puis Next (vérifier que le Filename soit bien *SQL.genmodel*). Sélectionner EcoreModel puis Next. Choisir SQL.ecore, en faisant Browse

workspace..., puis ouvrir Workspace, cliquer sur le projet initial et cocher SQL.ecore avant de cliquer Load. Faire enfin Next et Finish. Le fichier SQL.genmodel est créé.

- Configurer la génération. Le fichier SQL.genmodel contient les propriétés qui permettent de configurer la génération du code. Par exemple, on peut décider si la valeur d'un attribut sera éditable ou non. Nous nous limitons ici à une configuration minimale, donc pas de changements.
- Engendrer le code Java et l'éditeur arborescent. En sélectionnant la racine (du .genmodel), on peut maintenant sélectionner Generate All. Les projets *.edit et *.editor apparaissent.

Exercice 6 : Utiliser l'éditeur arborescent

- Lancer un nouvel Eclipse. Sélectionner le projet créé, en faisant un clic droit, sélectionner Run As / Run.... Sélectionner Eclipse Application, puis New_configuration. Si New_configuration n'apparaît pas, sélectionner New, en faisant un clic droit sur Eclipse Application. On peut ensuite cliquer sur Run.
- Créer un projet. Comme toujours, nous commençons par créer un projet (File / New / Project, puis General / Project) que l'on peut appeler ExempleSQL.
- Lancer l'éditeur arborescent. Dans le projet que l'on vient de créer, on peut lancer l'éditeur arborescent en faisant New / Other... puis dans Example EMF Model Creation Wizards, on sélectionne SQL Model. C'est bien le notre ! On peut ensuite conserver le nom par défaut proposé pour le modèle (My.SQL). Sur l'écran suivant, il faut choisir le Model Object, l'élément racine de notre modèle. On peut enfin faire Finish.
- Saisir le modèle SQL en utilisant New Child du menu contextuel. Pour avoir accès aux propriétés, il est conseillé de repasser dans la perspective Ecore (vue « Properties »).
- On peut utiliser Validate sur chacun des éléments du modèle. Ceci vérifie cet élément et ses sous-éléments.
- Quitter le deuxième Eclipse. Maintenant que notre modèle SQL est saisi, nous pouvons le sauver et quitter Eclipse pour revenir au premier Eclipse que nous avons lancé.

Exercice 7 : Création du métamodèle XML

Maintenant que nous avons manipulé les techniques liées à la métamodélisation sous Eclipse/EMF pour SQL, nous allons de la même façon, définir un métamodèle pour XML et créer un modèle XML conforme à ce métamodèle.

Exercice 8 : Création de l'éditeur textuel SQL

- Créer un projet Xtext de la manière suivante : File / New / Other / Xtext Project From Existing Ecore Models / Next / puis ajouter SQL.genmodel et sélectionner la racine du métamodèle XML comme Entry rule puis Next et finir en définissant l'extension du langage et le nom du projet.
- Modifier par la suite la grammaire en définissant les éléments de la syntaxe concrète.
- En sélectionnant la grammaire, faire clic droit, puis Run As Generate Xtext Artefacts.
- Reprendre les étapes décrites dans l'exercice 6 pour lancer l'éditeur de création de modèle.