

TP2 : Créer votre propre DSML avec EMF

NB. : Ce TP est à commencer en salle et à finir chez vous au cas où vous manqueriez de temps. Il est important que vous fassiez l'effort de répondre à toutes les questions. Cela est primordial pour la suite des TPs. Ne comptez pas sur une solution de notre part, vous serez déçus ;-)

Méta-modélisation avec Eclipse Modeling Framework (EMF)

Pour débiter avec la méta-modélisation, on vous propose de créer le méta-modèle i.e. le langage du fameux Logo, connu aussi sous le nom de Turtle Graphics. (Page wiki : http://en.wikipedia.org/wiki/Logo_%28programming_language%29). Pour cela, il faudra lire attentivement le cahier des charges (description) et utiliser EMF pour créer votre méta-modèle. L'objectif final étant de fournir le langage (M2) qui permettra de représenter le comportement d'une tortue sous forme d'un modèle (M1) puis de simuler l'exécution de ce dernier. Vous pouvez vous inspirer du résultat d'une implémentation web de Turtle Graphics sur ce lien : <http://www.sonic.net/~nbs/webturtle/webturtle.cgi>

Description (en Anglais) :

- A Turtle has a name, and can be used as target of several Actions:
 - Forward (with given integer length)
 - Rotate (with given float angle)
 - SetPen (can be one of UP or DOWN)
- A Choreography is defined as a sequence of Actions executed by the Turtle.
- A collection of Turtles live inside a named Farm
- Turtles play Choreography(ies) on a Stage:
 - With given dimension (height and width)
 - It can holds several Choreography(ies).
 - Farms are shared over different stages
 - A Stage can be displayed on screen by executing a display operation.
- A Color is defined as a combination of red, green and blue channels ([0-255]), and an alpha channel.
 - A Stage define a background Color
 - An evolved species of Turtle (called ColouredTurtle) brings a Color.

1) Réaliser le méta-modèle de Logo (i.e. Turtle Graphics)**Aide Eclipse :**

- Créer un nouveau projet EMF Eclipse (File->New->Project...>Eclipse Modeling Project->Empty EMF Project. Nommez le TurtleGraphics.
- Click droit sur le sous répertoire « model », puis New->Other...>Ecore Tools>Ecore Diagram. Nommer votre diagramme turtle.ecore.
- Eclipse vous générera deux diagrammes. Le premier (l'ancêtre) **turtle.ecore** qui représente une vue arborescente de votre méta-modèle et le deuxième, rajouté récemment par Eclipse, **turtle.ecoredia** qui lui vous donne plutôt une vue « diagramme de classes ». Les deux diagrammes représentent la même chose et sont (normalement) synchronisés !

- Commencez à remplir le diagramme avec les concepts de votre langage dans le .ecoredia. Si vous avez du mal (et seulement si !!!) avec l'outil, vous pouvez consulter le tuto-video (Merci à S.Mosser) qui se trouve dans votre package de TP. Faites surtout attention aux Types (i.e. EString, Exxx), aux associations bidirectionnelles (notion d'opposite) et aux compositions (notion de containment).
- Rajoutez une opération display() : EString à la classe Stage
- Une fois terminé, validez votre méta-modèle en faisant un click droit sur la racine de votre méta-modèle dans l'éditeur arborescent puis -> Validate
- N'hésitez pas à valider votre solution avec votre enseignant

Création dynamique d'instances de modèles

2) Votre méta-modèle prêt, il s'agit maintenant de créer des instances (M1) de ce dernier (M2). Voici le modèle que nous voulons créer (le moment aussi de vérifier si votre méta-modèle le permet) :

Un Stage de 200Hx300L contient une Choreography composée de cette suite d'actions:

- Set Pen DOWN
- Forward 100
- Set Pen UP

Pour créer l'instance, placez-vous au niveau de l'éditeur arborescent de votre ecore. Bouton droit sur la méta-classe Stage-> Create Dynamique instance (voir figure ci-dessous)

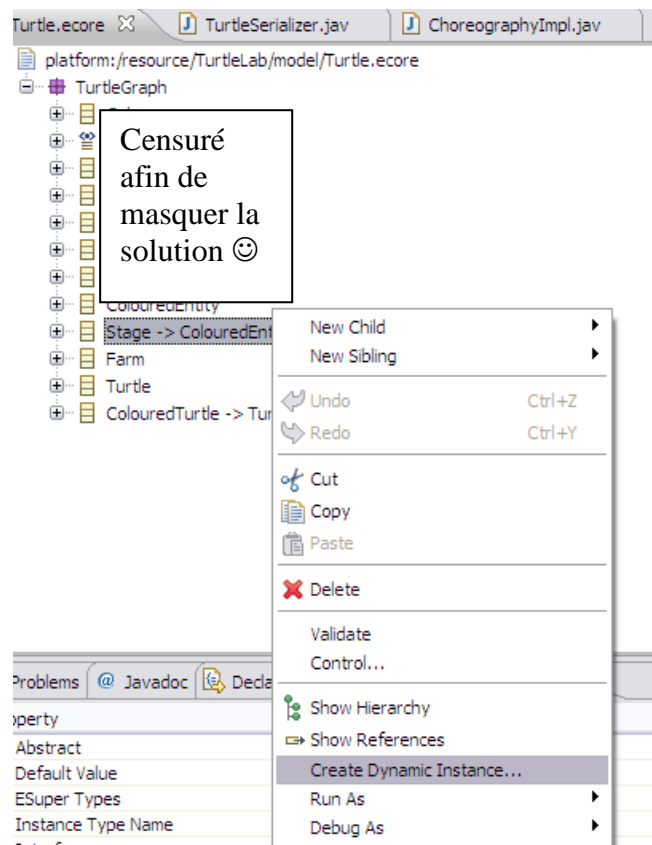


Fig. 1 création d'une instance dynamique de modèle

Le modèle devrait ressembler à cela :

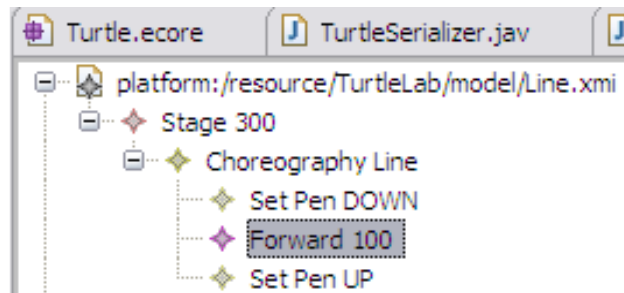
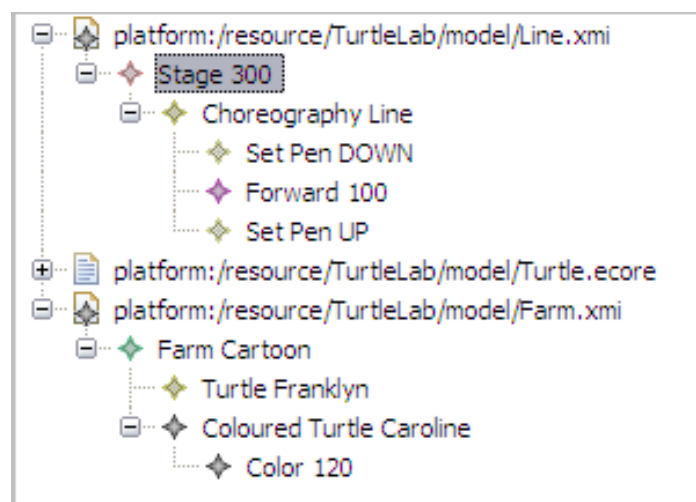


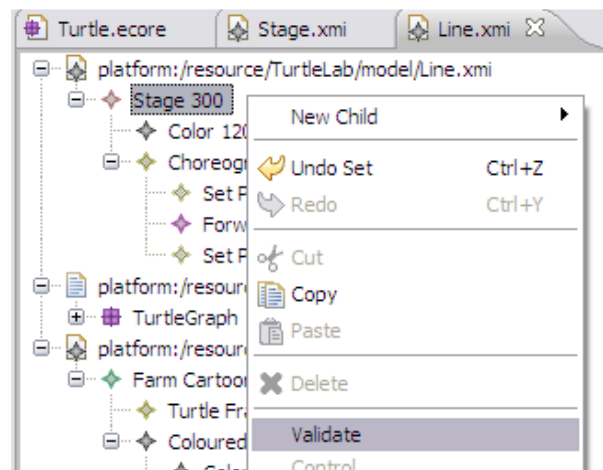
Fig. 1 un modèle instance de votre langage

Maintenant, dans la même instance de modèle on aimerait spécifier que cette chorégraphie (les actions) est exécutée par la Turtle Alice qui vit dans la Farm wonderland. Selon votre réponse à la question 1) :

- Soit au niveau du modèle vous avez la possibilité d'exprimer cela et donc de créer une instance de Farm (wonderland) et de Turtle (Alice) et de les lier à Stage et Action (respectivement). Action a normalement un attribut target qui pointe vers une instance de type Turtle responsable de réaliser l'action
- Soit il vous faudra créer une autre dynamique instance de modèle pour Farm wonderland qui contient la turtle Alice et de la lier à l'instance de Stage (et Action). **Comment lier les deux modèles ?**
 - **Aide :** pour que deux modèles puissent se référencer il faut qu'ils se connaissent et surtout qu'ils puissent être accessibles l'un à l'autre (comme l'import en Java). Dans votre 1^{er} modèle de Stage, placez-vous au niveau de Stage (dans l'éditeur arborescent). Click droit-> Load Resource -> Browse Workspace puis pointez vers le modèle contenant la Farm wonderland. Vous aurez alors l'équivalent de la figure ci-dessous :
 - Allez ensuite dans Stage et faites en sorte qu'il pointe vers Farm Wonderland. Même chose pour les actions de stage, elles doivent toutes avoir comme target pour l'exécution, la turtle Alice !
 - N'oubliez pas de sauvegarder



Si vous avez fini votre modèle, validez-le en faisant un click droit sur la racine i.e. Stage dans l'éditeur arborescent puis -> Validate (voir figure ci-dessous)



La suite au prochain TP :

- Génération de l'éditeur arborescent
- Génération de Code Java
- Manipulation de modèles avec le code Java généré
- Un peu d'OCL

Pré-requis

- Eclipse avec le projet Modeling Tools à cette URL :
- <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools-includes-incubating-components/galileo2>

Vous avez le choix (en haut à droite entre Linux, Mac OS et Windows en 32/64 bits)

- Fichier-> New...->Project, sous General, choisir Project
- Nommer le Projet, « TP2 par exp. »

Solution

