

# Prise en main de l'environnement EMF/Ecore

Objectifs :

- Découverte de l'environnement EMF/Ecore
- Édition de métamodèles Ecore
- Édition de modèles conformes à un métamodèle

## 1 Installation

Nous allons travailler sur une version d'Eclipse différente de celle disponible en salles machines. Vous allez donc installer cette version sur votre compte (ou sur votre machine personnelle bien sûr). Pour cela, il faut récupérer une archive Eclipse Modeling Tools version juno :

<http://www.eclipse.org/downloads/packages/release/juno/sr2>

Désarchivez cette archive à l'endroit de votre choix sur votre compte, rentrez dans le répertoire eclipse ainsi obtenu et lancez **eclipse** (bien sûr, ne lancez pas eclipse à partir du menu principal du système, cela lancerait la version installée sur les machines). Nous allons maintenant installer le plug-in Papyrus qui nous servira par la suite. Pour cela, allez dans le menu Help, puis Install New Software. Créez un nouveau site de nom Papyrus, et d'*update site* : <http://download.eclipse.org/modeling/mdt/papyrus/updates/releases/juno> Poursuivez ensuite la procédure d'installation. Fermer **eclipse**, et relancez-le (attention, toujours sans passer par le menu système!)

## 2 Saisie d'un méta-modèle

Créez un projet (projet simple ou projet Papyrus). Dans ce projet, créez un diagramme de métamodèle ecore : **new** → **other** → **ecore tools** → **ecore diagram**.

Vérifiez que vous avez bien la vue **Properties** qui s'affiche (dans un onglet en bas normalement, de même type que l'onglet **TODO** ou **Problems** auxquels vous devez être habitués). Si ce n'est pas le cas : **Window** → **Show views** → **other** → **General** → **properties**.

Saisissez le petit métamodèle de machines à états vu en TD.

Vous noterez que vous obtenez deux fichiers : le fichier qui contient le métamodèle (de suffixe.ecore) et celui qui contient la partie graphique de votre métamodèle. Nous retrouverons souvent ce genre de paires de fichiers.

Ouvrez le fichier ecore (en double-cliquant dessus sous eclipse). Vous voyez votre métamodèle sous forme d'instance d'Ecore.

Ouvrez-le maintenant avec un éditeur de texte (clic droit, ouvrir avec, éditeur de texte). Vous voyez que ce fichier est en fait stocké au format XMI (dialecte XML).

## 3 L'éditeur (réflexif) de modèles

Quand on dispose d'un métamodèle ecore, l'environnement EMF fournit un éditeur de modèles. L'édition revient à construire l'arborescence du modèle vu comme instance du métamodèle, en partant d'un élément racine.

Ouvrez de nouveau votre métamodèle ecore avec l'éditeur par défaut (en double-cliquant sur le fichier de suffixe ecore).

Sélectionnez l'élément racine de votre méta-modèle, c'est-à-dire celui qui "contient" (transitivement via les liens de composition) tous les autres, ensuite : clic droit, Create dynamic instance. Choisissez un nom de fichier pour votre modèle. L'éditeur s'ouvre. Si c'est l'éditeur de texte qui s'ouvre (et que vous visualisez du XMI), fermez-le et réouvrez le fichier de modèle en double cliquant dessus.

Vous voyez une instance de l'élément racine choisi, dont vous pouvez changer les propriétés dans l'onglet **Properties** (si votre élément définit des propriétés bien sûr). Construire un modèle se fera en rajoutant des éléments à la racine puis aux éléments ainsi ajoutés. Par exemple, après un clic droit sur l'élément racine, la liste des éléments que vous pouvez ajouter apparaît. Cette liste dépend de votre métamodèle. Vous pouvez ainsi ajouter des éléments et créer l'arbre couvrant de votre modèle en suivant l'arbre couvrant défini par votre métamodèle (avec les compositions). Pour affiner cet arbre, on peut renseigner les propriétés de chaque élément, et y ajouter des liens.

Saisissez le petit exemple vu en TD.

Vous remarquez que cet éditeur est pratique pour saisir un modèle en syntaxe concrète. Toutefois, il est en général plus pratique de saisir les modèles avec une syntaxe concrète et donc un éditeur adapté à cette syntaxe concrète. L'environnement EMF/Ecore ne peut pas vous fournir ce type d'éditeurs graphiques par défaut, puisqu'il ne connaît pas la syntaxe concrète associée à votre métamodèle. Toutefois EMF fournit des outils pour construire un éditeur de syntaxe concrète, nous en reparlerons plus tard.

## 4 Syntaxe abstraite et concrète : l'exemple d'UML

Papyrus est un plug-in permettant d'éditer des modèles UML. Nous allons saisir un modèle UML simple.

Sur votre projet, clic droit → **new** → **other** → **papyrus** → **papyrus model** → etc.

Choisissez de créer un diagramme de classe. Créez une classe avec au moins une méthode, puis sur cette classe, créez une machine à états.

Créez un état initial, un autre état, un état final, et deux transitions liant ces trois états. Pour les transitions, ne vous énervez pas à ajouter un trigger (qui est l'événement déclencheur de la transition, l'éditeur ne le permet pas) lié à une méthode.

Vous noterez que pour votre modèle, vous avez à nouveau deux fichiers : un fichier suffixé .uml (qui est en fait un fichier XMI, qui contient votre modèle, et l'autre suffixé .di (qui contient la représentation graphique de votre modèle, di abrégeant diagram).

Fermez le fichier .di, et ouvrez le .uml. Regardez votre modèle. Vous constatez que le métamodèle UML est différent du vôtre (ce qui est très normal). On va rajouter un trigger manquant sur la transition de l'état à l'état final. Pour cela, commencez par créer un événement d'appel de méthode (**CallEvent**) dans votre model. Vous associez ce **CallEvent** à l'appel de la méthode que vous avez définie dans votre classe de base. Puis, retrouvez dans le modèle la transition à compléter, créez un trigger, associez-y le **CallEvent**. Sauvez, et fermez le .uml. Ouvrez de nouveau le .di. La transition est maintenant bien renseignée.