

## Prise en main de l'environnement EMF/Ecore

Objectifs :

- Édition de métamodèles Ecore
- Édition de modèles conformes à un métamodèle
- Premiers pas EMF

### 1 Installation

Il va nous falloir installer un nouveau composant pour eCore. Pour cela :

- Help->install new Software.
- Dans work with, choisir : <https://download.eclipse.org/releases/2022-03/>
- Puis sous Modeling, choisir, Ecore Diagram Editor (SDK). Terminez l'installation.

### 2 Résumé du travail à réaliser

Au cours de cette séance nous allons travailler sur des modèles de cartes (ou de plans).

La première étape est de définir un métamodèle de carte, permettant de définir les concepts qui seront utilisés pour modéliser des cartes. Ce métamodèle vous est fourni, et est simpliste, vous le saisissez en utilisant eCore comme méta-métamodèle.

La deuxième étape est de pouvoir saisir des modèles conformes à ce métamodèle. Comme aucune syntaxe graphique n'est définie, on ne peut le faire qu'en syntaxe abstraite, comme instance du métamodèle. Nous utiliserons l'éditeur que nous fournit EMF.

La troisième étape est de générer le code Java permettant de manipuler les modèles de cartes.

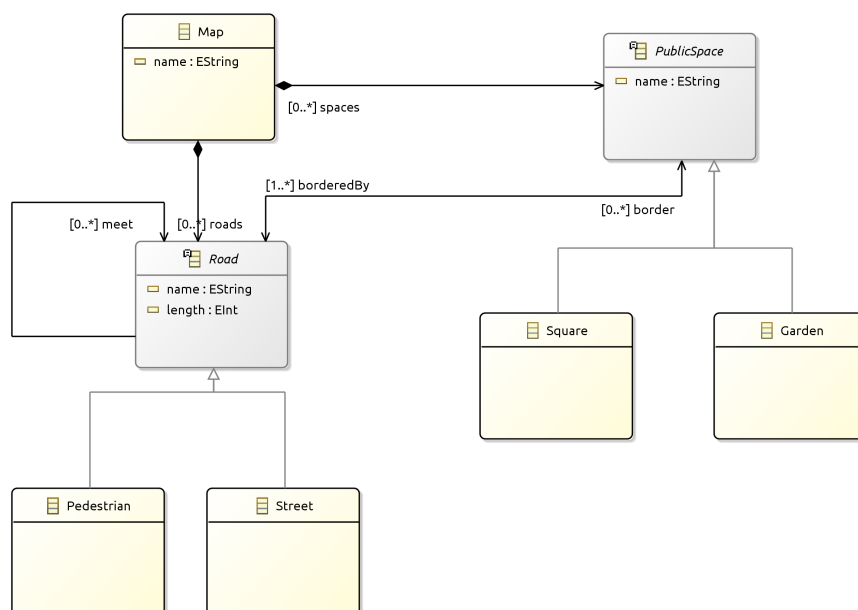
La quatrième étape est d'utiliser le code généré et un code qui vous est fourni pour charger un modèle de carte, et en afficher le nom.

La cinquième étape est, toujours en utilisant le code généré, d'écrire des méthodes de parcours de modèles.

La sixième étape consiste à amorcer de premières transformations de modèles, en ajoutant des éléments aux modèles, par programme.

### 3 Un métamodèle simplissime de cartes

Ce métamodèle est un exemple jouet créé par Adel Ferdjouxh.



## 4 Travail à réaliser

### 4.1 Saisie d'un méta-modèle

- Créez un projet (projet EMF vide, empty EMF project).
- Dans ce projet, créez un modèle ecore dans le répertoire model : new → other → Ecore Model.
- Un éditeur s'ouvre pour votre fichier eCore. Sélectionnez l'icône sous la racine du fichier, puis regardez la fenêtre properties (si elle n'est pas visible : window-> show view -> properties).
- Remplissez les champs name, nsURI et nsPrefix avec les noms de votre choix (par exemple maps).
- Notre métamodèle est prêt à être saisi. Pour en faciliter la saisie, nous allons utiliser un éditeur graphique. Nous allons pour cela créer un diagramme Ecore.
- Sur le fichier, choisissez initialize Ecore Diagram (que vous nommerez) → Entities in a class Diagram. Laissez inchangés les champs proposés, sélectionnez Finish et donnez un nom à votre diagramme
- Vous obtenez un éditeur proche d'un modèleur UML, vous permettant de saisir un métamodèle. Saisissez le métamodèle simplissime de cartes.

Ouvrez le fichier ecore (en double-cliquant dessus sous eclipse). Vous voyez votre métamodèle sous une autre forme, et en étudiant les properties, vous voyez que c'est une 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).

### 4.2 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 (à placer dans le répertoire model). L'éditeur s'ouvre.

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 un exemple de modèle.

### 4.3 Génération de code

Pour générer le code du métamodèle de cartes, suivez les étapes suivantes :

- Créer un fichier maps.genmodel.
  - Menu contextuel sur maps.ecore → New → Other,
  - EMF Generator Model → Nommer puis Next → Choisir Ecore Model puis Next,
  - Appuyer sur Load (pas d'erreurs) puis Next → Finish.
- Le fichier créé s'ouvre, puis menu contextuel sur le package → Generate Model Code.
- Regarder le dossier src du projet, 3 nouveaux packages ont été créés. Ils contiennent des classes pour instancier le métamodèle et manipuler ses modèles conformes.

### 4.4 Charger une instance

- Créer un nouveau package java dans le dossier src et le nommer utilisationMaps.
- Copier le fichier ManipulationsMaps.java dans le package src/utilisationMaps.
- Il y aura peut être des erreurs au niveau des premiers imports (import maps.MapsPackage; et import maps.Map;) du fait de différences de nommage, adaptez ces imports à votre projet.
- Il y aura certainement des erreurs au niveau des imports de type : import org.eclipse.emf.ecore.xmi; pour les résoudre, cliquez sur la croix rouge, fix project setup puis suivez les instructions.
- Exécuter le programme ManipulationsMaps.java pour voir ce qu'il fait.

## 4.5 Parcours de modèle

Écrire les méthodes permettant de :

- Récupérer les noms de toutes les rues (Street) d'une map donnée.
- Récupérer toutes les rues piétonnes (Pedestrian) dont la longueur dépasse 1000m.
- Pour un nom de rue donné, trouver tous les noms des rues ou chemins adjacents.

## 4.6 Modification de modèle

Écrire une méthode permettant de créer un nouveau jardin. Cette méthode prendra en paramètre le nom du jardin et les "roads" la bordant. Pour créer un jardin, vous utiliserez la méthode fabriquer de la classe MapsFactory. Pour enregistrer le modèle après modification, vous utiliserez la méthode sauvegarderModele. Attention, le nouveau modèle doit être stocké dans un fichier impérativement suffixé par xmi.

# 5 Premiers pas avec Papyrus, modèle en syntaxe concrète et abstraite

Pour vous préparer aux séances suivantes, créez un modèle papyrus avec un diagramme de vos classes. Ajoutez-y quelques éléments (classes, attributs, méthodes, ...). Vous verrez que vous obtenez plusieurs fichiers. Celui suffixé par uml contient le modèle en syntaxe abstraite sérialisé en xmi. Si vous l'ouvrez avec l'éditeur de texte, vous retrouvez la syntaxe abstraite. Dans le fichier postfixé par notation, vous avez la partie graphique (regardez mais on ne s'en servira pas).