

## TP 1: Introduction à EMF (Eclipse Modeling Framework)

**Eclipse Modeling Framework (EMF)** est un Framework de modélisation, une infrastructure de génération de code et des applications basées sur des modèles de données structurées. Partant d'une spécification décrite généralement sous la forme d'un modèle en XMI, EMF fournit des outils permettant de produire des classes Java représentant le modèle avec un ensemble de classes pour adapter les éléments du modèle afin de pouvoir les visualiser, les éditer avec un système de commandes et les manipuler dans un éditeur.

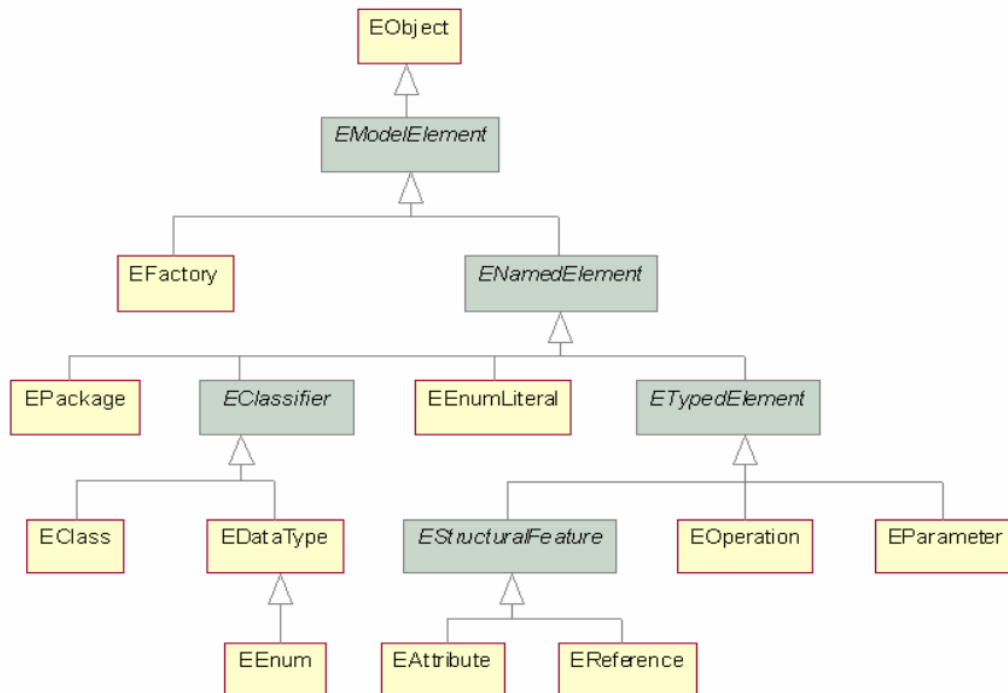
EMF vous permettra de :

- Générer du code Java
- Manipuler des modèles dynamiques (pas besoin de codes générés)
- Interroger, créer et mettre à jour des instances de modèles
- Sérialiser et désérialiser des instances
- Valider des instances
- Écouter les changements des instances

EMF s'appuie sur le métamodèle Ecore qui respecte les principes définis par le « eMOF » (Extended Meta Object Facility) qui est un standard OMG :

- EClass : désigne les classes des modèles, identifiées par un nom, peuvent contenir des StructuralFeatures (attributs ou références). Supporte l'héritage multiple, peut être abstrait (pas d'instance possible) ou une interface (pas d'implémentation générée)
- EAttribute : identifié par un nom et un type. Bornes mini et maxi sont utilisés pour la cardinalité

- EReference : association entre deux classes, identifiée par un nom et un type (une classe). Relation inverse possible (opposite). Bornes mini et maxi sont utilisées pour la cardinalité. Association de type composition autorisé (containment).



- EDataType : type primitif ou type objet défini par Java
- EPackage : désigne les packages des modèles qui sont des conteneurs de classifieurs (classes et types). Défini par un nom de package (unique) et une URI pour l'identification lors de la sérialisation
- EOperation : désigne les opérations d'une classe pouvant être invoquées. Identifiée par un nom, un type de retour et des paramètres. Autorise les exceptions
- EEnum : désigne le types énumérés parmi un ensemble de EEnumLiteral

Les éléments de type StructuralFeatures (attributs et références) contiennent des paramètres pour contrôler le code généré

- Unsettable (true, false) : précise qu'une valeur d'un attribut n'a pas encore été déterminée (exemple : booléen true/false/undetermined)
- Containment (true, false) : l'association est une composition
- Unique (true, false) : pour les cardinalités multiples, précise qu'il ne peut y avoir la même valeur d'objet
- Changeable (true, false) : valeur ne peut changer (pratique pour les relations inverses)

- Volatile (true, false) : ne génère pas l'attribut pour stocker l'état, le corps de la méthode est également laissé à vide
- Transient (true, false) : non persisté
- Derived (true, false) : calculé à partir d'autres StructuralFeatures (attribut généralement marqué Volatile et Transient).

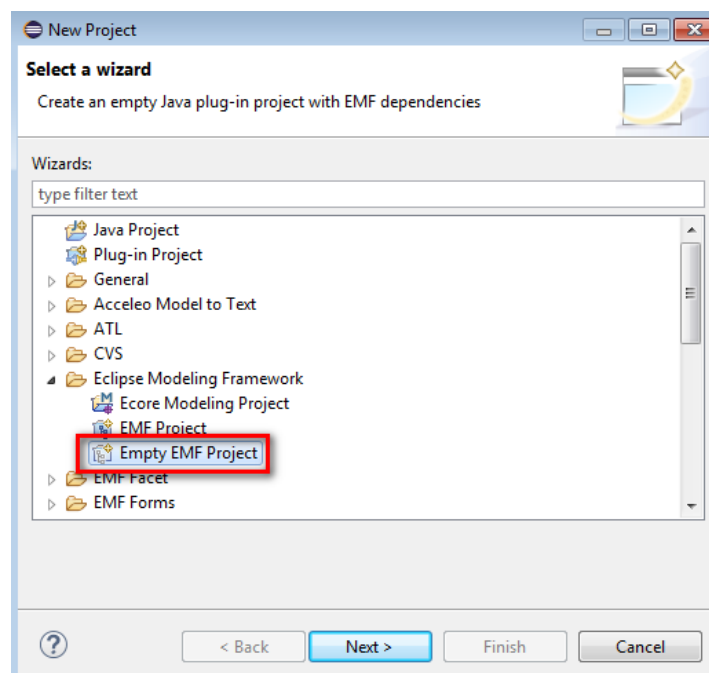
Pour construire un modèle EMF plusieurs formats disponibles

- Modèle Ecore (voir la suite)
- Classes Java annotées
- Modèle de classes Rose
- Modèle UML
- XML Schema : les instances XMI seront conformes à l'XML Schema de départ

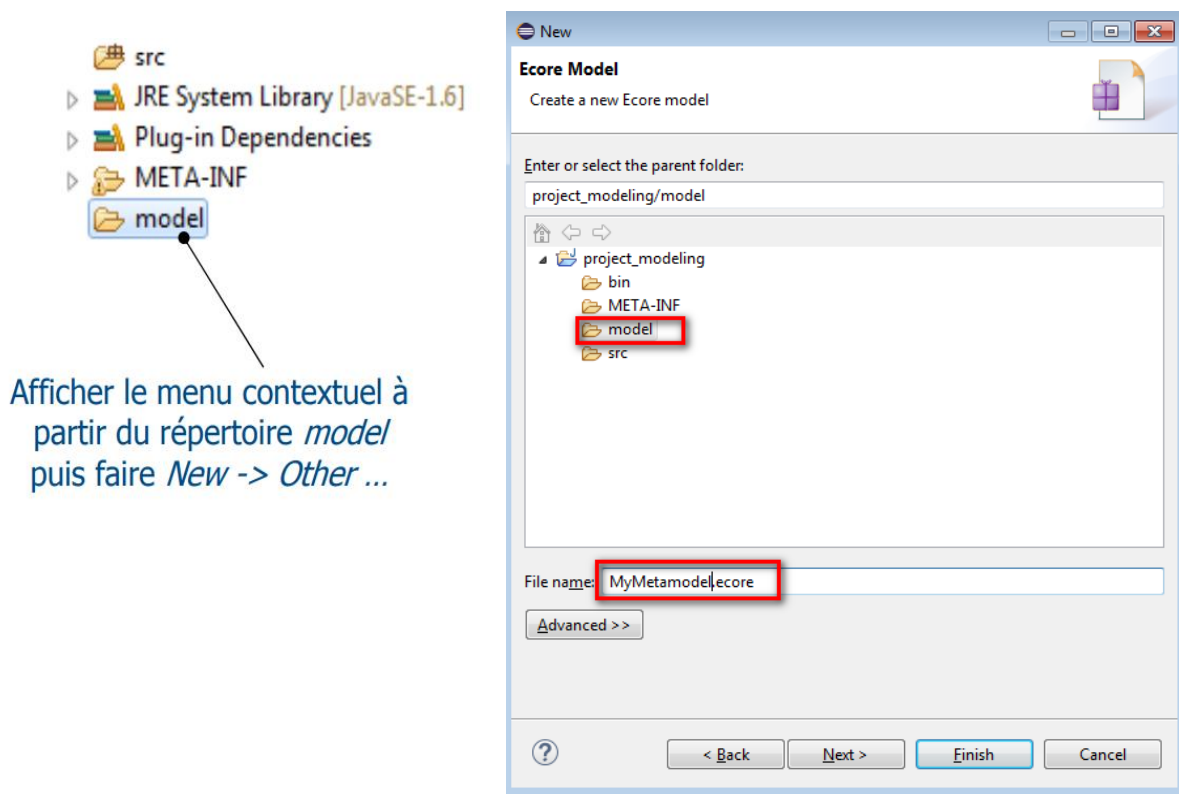
Nous utiliserons par la suite un modèle Ecore puisque l'outillage fourni par EMF facilite la construction

### Création du modèle EMF (extension .ecore)

1. **Ajouter les outils de modélisation** à votre eclipse ou utiliser « eclipse modeling tools ».
2. **Créer un nouveau projet EMF** vide appelé « project\_modeling »

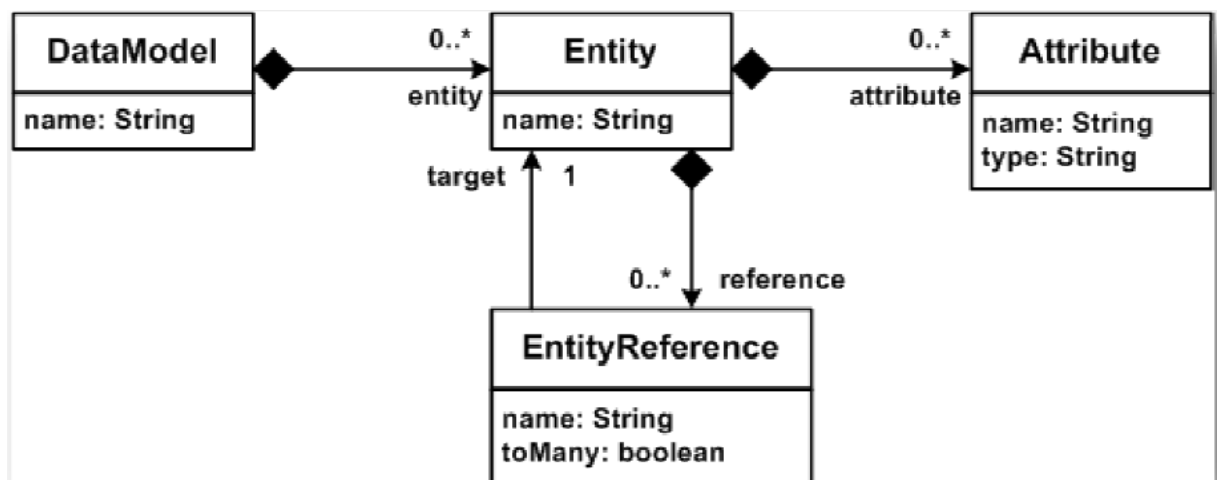


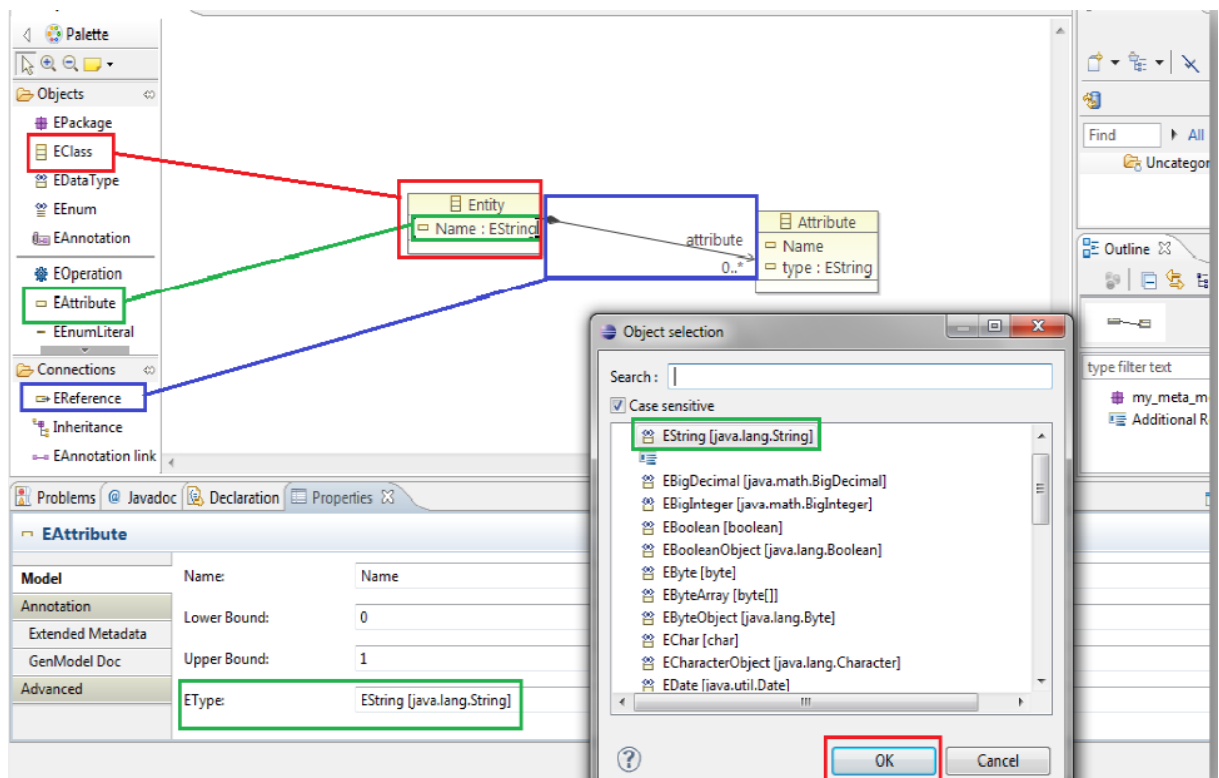
### 3. Création d'un méta modèle :



### 4. Créer un ecore diagramme : Sélectionner le fichier métamodèle, right click et sélectionner Initialize Ecore Diagram

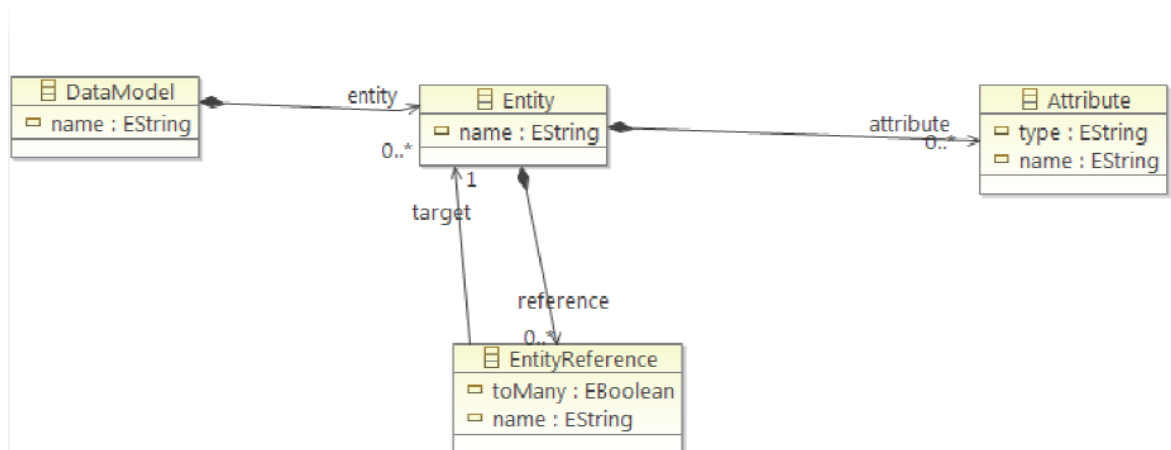
Un éditeur graphique avec une palette vous parait : La prochaine étape est de créer les éléments du métamodèle





PS : Pour les associations on utilise EReferences . S'assurer upper bound est à "\*" et que "Is Containment" est coché (la composition)

Le méta modèle sous EMF est le suivant :

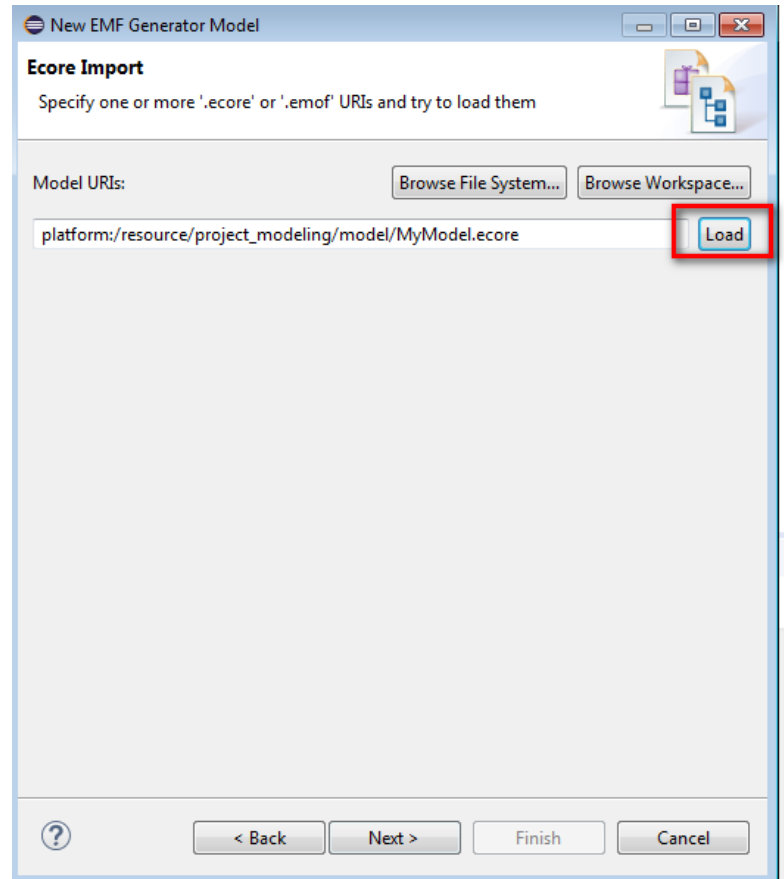
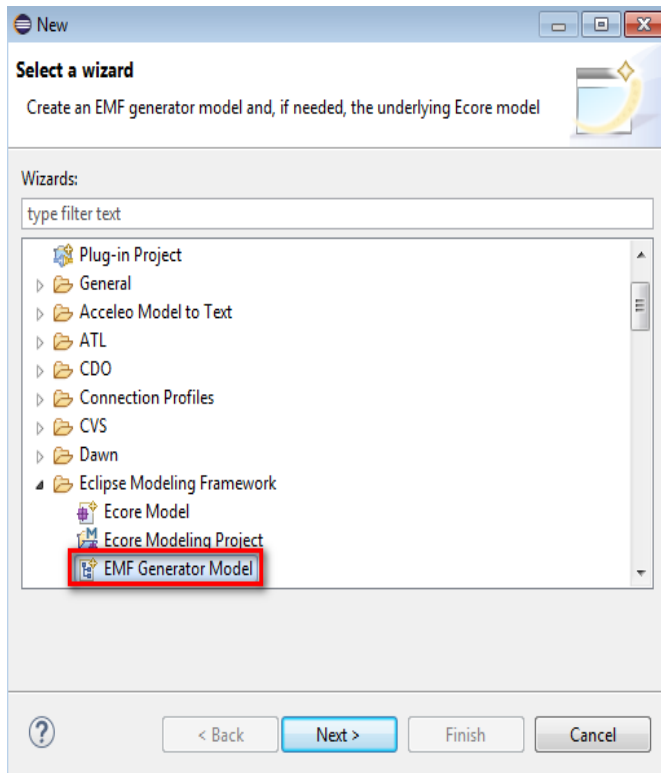


Une fois le méta modèle finalisé, ouvrir le fichier « .ecore » crée automatiquement. C'est le fichier décrivant sous d'arborescence le méta modèle.

## 5. Créer le générateur de modèle EMF

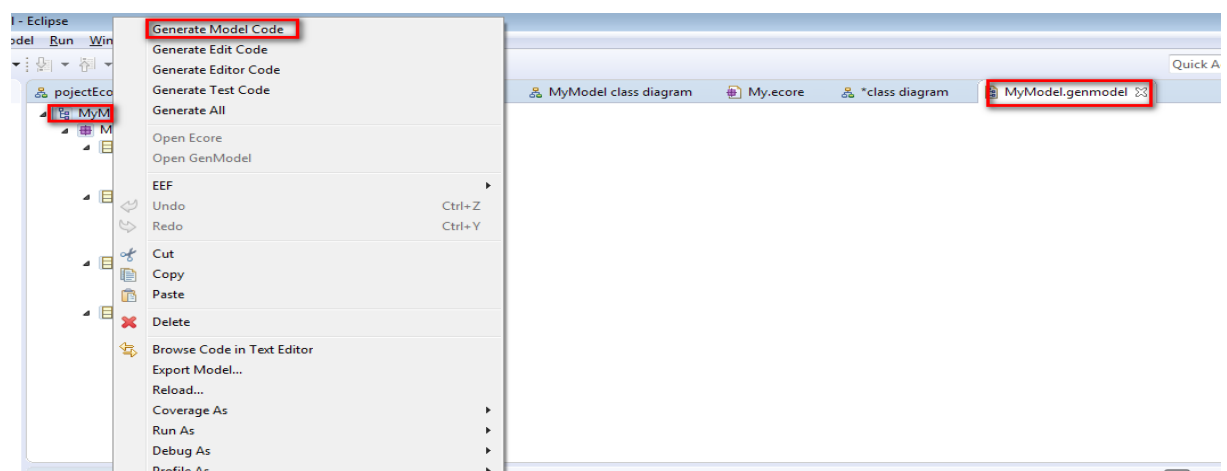
Clic droit sur le fichier « .ecore » et sélectionner File → Other → EMF generator model

Choisissez ensuite votre fichier .ecore et charger le via load. A ce niveau se fera une vérification et validation du méta modèle. Des erreurs peuvent être levées (les cardinalités, les types d'attributs, les relations multiples etc...)



## 6. Génération du code

Cette étape vous permet de créer l'implémentation java du modèle EMF dans le projet courant.



Une fois la génération finalisée, vous aurez 3 nouveaux projets associés et des sources ajoutées à votre projet courant.

Analyser et essayer de comprendre le code généré.