Stockage et manipulation des modèles

Zineb BOUGROUN

Plan

- XMI : Xml Metadata Interchage
- EMF : Eclipse Modeling Framework

MDA : représentation

- Les modèles sont des entités abstraites :
  - comment les utiliser ?
- MDA définit deux représentations
  - Forme Textuelle
    - XMI (XML Metadata Interchange)
  - Forme d'objets de programmation
    - JMI (Java Metadata Interface)
    - EMF (Eclipse Modeling Framework)

XMI : Xml Metadata Interchage

Définition

- Le **XML Metadata Interchange** (**XMI**) est un standard de l'OMG pour échanger les métadonnées *via* le *Extensible Markup Language (XML)*.

- Il peut être utilisé pour toutes métadonnées dont le métamodèle peut être exprimé en *Meta-Object Facility (MOF)*.

➔ vise à stocker et échanger des modèles

➔ en décrivant les modèles sous format de fichier XML

XMI : Xml Metadata Interchage

Définition

- permettre l'interopérabilité des méta-données
  - entre outils de modélisation
  - entre répertoires de méta-modélisation

- Est basé sur 3 standards : XML, UML, MOF

- Les modèles sont sérialisés avec la DTD générée à partir de leur méta-modèle.

➔ **N'importe quel modèle peut être ainsi représenté par un document XMI**

XMI : Xml Metadata Interchage

Quelque régles

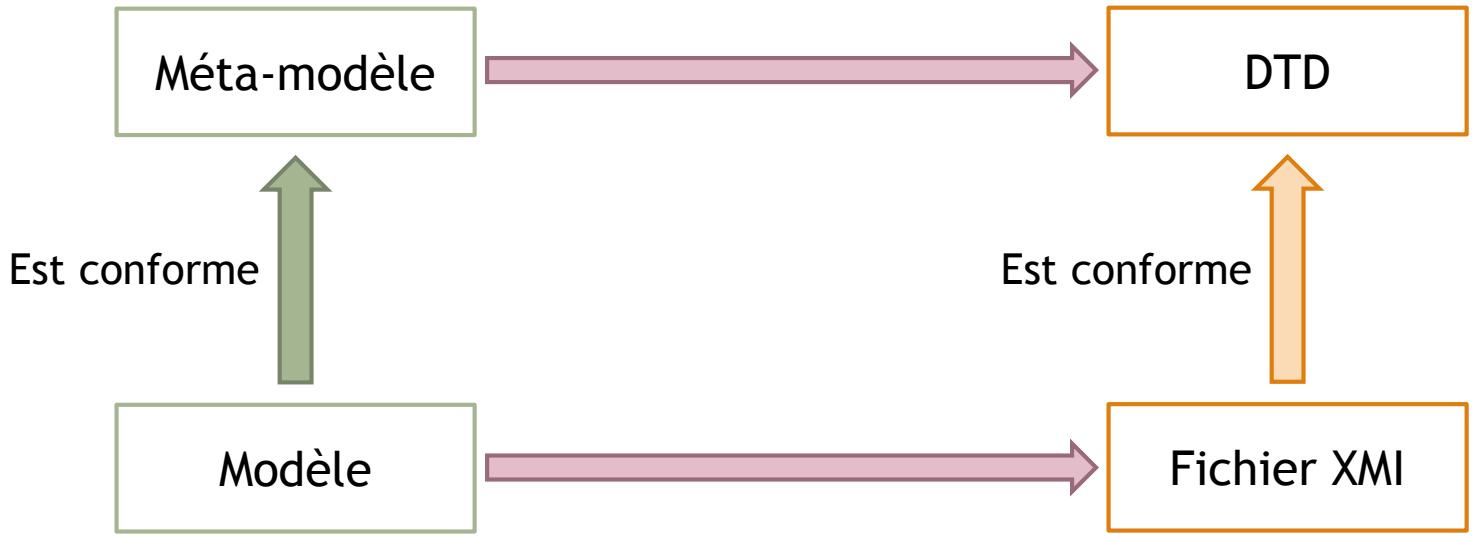
1. Toute métaclasse fournit la définition d'une balise ayant comme nom le nom de la méta-classe. Cette balise doit posséder un attribut XML nommé `id` permettant d'identifier l'instance de la métaclasse afin de la référencer au besoin.

2. Tout méta-attribut d'une métaclasse fournit la définition d'une balise ayant comme nom le nom du méta-attribut. Le contenu de la balise représentera la valeur du méta-attribut. Cette balise doit être contenue dans la balise correspondant à la métaclasse contenant l'attribut.

3. Toute métaréférence d'une métaclasse fournit la définition d'une balise ayant comme nom le nom de la métaréférence. Le contenu de la balise représentera un `id` identifiant l'instance référencée. Cette balise doit être contenue dans la balise correspondant à la métaclasse contenant la référence.

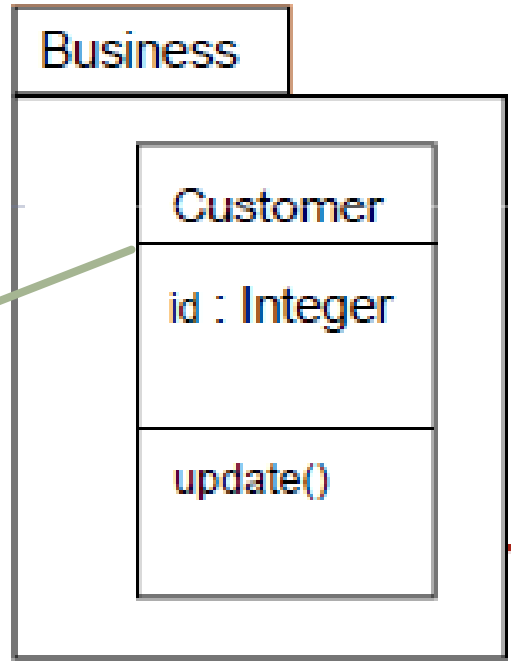
4. Toute méta-association entre deux métaclasses fournit la définition d'une balise ayant comme nom le nom de la méta-association. Le contenu de la balise contiendra les instances reliées ou les `id` les identifiant. Si la méta-association est navigable, la balise de la méta-association doit être contenue dans la balise correspondant à la métaclasse source de l'association.

XMI : Xml Metadata Interchage



XMI : Xml Metadata Interchage

```
<UML:Model xmi.id="a1">
      <name> Business </name>
      <ownedElement>
            <Class xmi:type="UML:Class” xmi.id="id2">
                  <name> Customer </name>
                  <feature>
                        <Attribute name=“id” type=“type1”/>
                        <Operation name=“update” />
                  </feature>
            </Class>
      </ownedElement>
      <ownedElement xmi:type="UML:Datatype" name="Integer"
xmi:id="type1"/>
</UML:Model>
```



XMI : Xml Metadata Interchage

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3c.org/2001/XMLSchema " xmlns:xmi="http://www.omg.org/XMI"
xmlns:cds="http://www.example.org/CDs">
    <xsd:import namespace="http://schema.omg.org/spec/XMI/2.1" schemaLocation="XMI.xsd"/>
     <xsd:complexType name="CD">
          <xsd:choice minOccurs="0" maxOccurs="unbounded">
               <xsd:element name="title" type="xsd:string"/>
               <xsd:element name="artist" type="xsd:string"/>
               <xsd:element name="num_tracs" type="xsd:integer"/>
               <xsd:element ref="xmi:Extension"/>
          </xsd:choice>
          <xsd:attribute ref="xmi:id"/>
          <xsd:attributeGroup ref="xmi:ObjectAttribs"/>
          <xsd:attribute name="title" type="xsd:string"/>
          <xsd:attribute name="artist" type="xsd:string"/>
          <xsd:attribute name="num_tracs" type="xsd:integer"/>
          <xsd:attribute name="single" type="xsd:string"/>
     </xsd:complexType>
```

XMI : Xml Metadata Interchage

- Donner un modèle respectant le schéma:

```
<?xml version="1.0" encoding="UTF-8"?>
    <cds:CD xmlns:cds="http://www.example.org/CDs"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xmi="http://schema.omg.org/spec/XMI/2.1"
   xsi:schemaLocation="http://www.example.org/CDs"
        artist = " Adele"
        title = "25«
        num_tracs = "11"
        single =" Hello"
        xmi:id = "3">
    </cds:CD>
```

EMF : Eclipse Modeling Framewwork

Définition

- Est un framework Java qui permet la génération de code pour la construction d'outils et d'applications basés sur les modèles structurés.

- Il permet donc :
  - La modélisation et l'intégration de données
  - La génération de code

EMF : Eclipse Modeling Framewwork

Définition

- Un environnement intégré, dédié à la modélisation
  - Définition de méta-modèles
  - Création de modèles satisfaisants les méta-modèles définis
  - Génération automatique de code (squelettes de classes) à partir de modèles
  - C’est un format de représentation des modèles permettant leur manipulation dans les langages de programmation

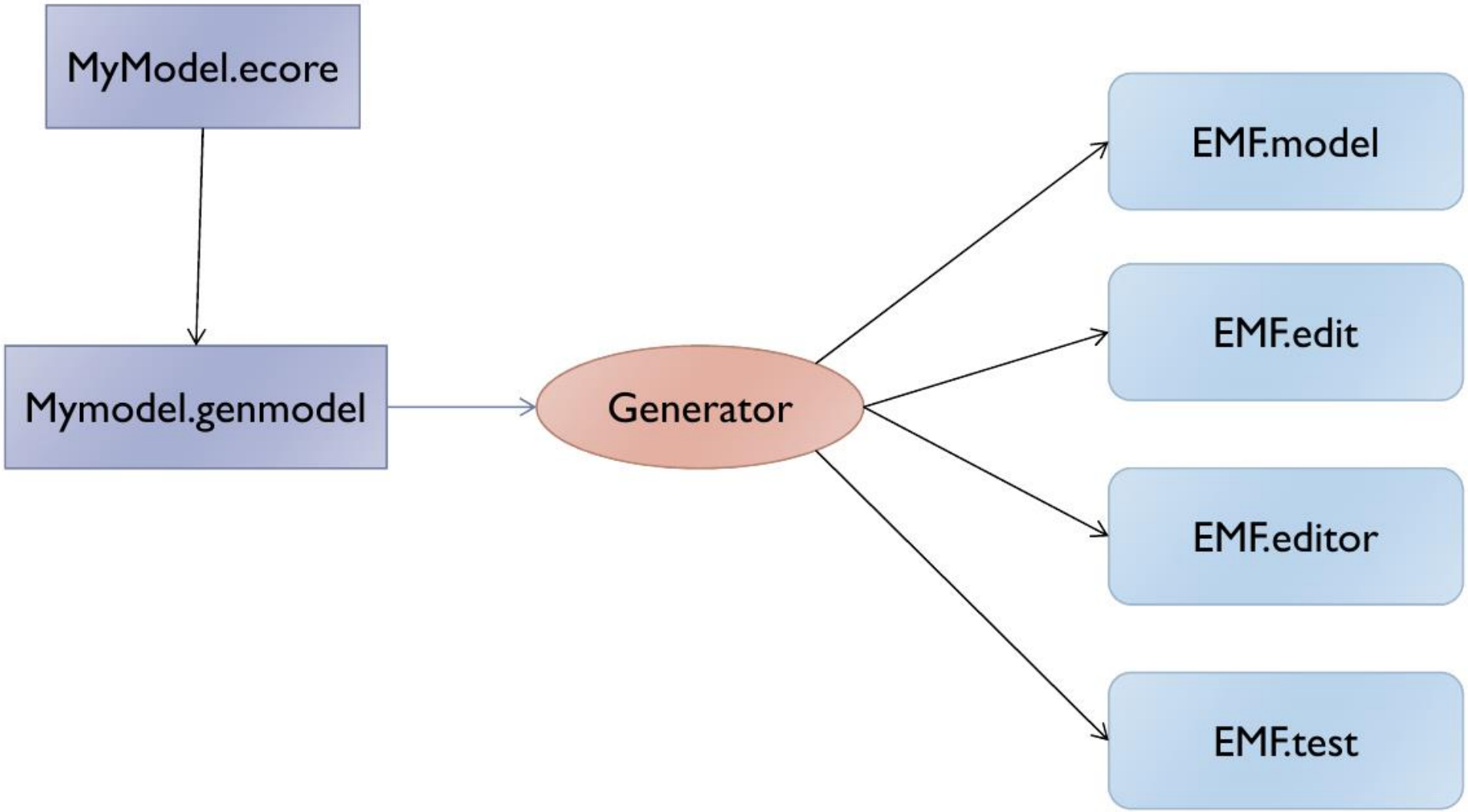
EMF : Eclipse Modeling Framewwork

- Un langage pour définir des modèles
- Ressemble à UML (diagramme de classe) mais en plus simple
- Comment?
  - EMF editor
  - Schéma XML
  - Annotation java

- En utilisant EMF pour mon modèle, je peux :
  - générer le code java équivalent à mon modèle
  - Créer un éditeur pour instancier les éléments de mon modèle
  - Sérialiser ces modèle avec XMI

EMF : Eclipse Modeling Framewwork

Processus EMF



EMF : Eclipse Modeling Framewwork

Fichiers EMF

- .Ecore
  - Est un méta-modèle possédant plusieurs concepts (Eclass, Eattribute, Edatatype…)
- .Genmodel:
  - Procure des détails à la plateforme
  - Généré à partir du fichier .ecore du modèle
  - Indispensable pour générer le code du modèle
- Modèle:
  - Le modèle contient toutes les entités, les forfaits et les usines pour créer des instances du modèle.
- Edit:
  - Contient les providers pour afficher un modèle dans une interface utilisateur.
- Editeur:
  - Est un éditeur exemple généré pour créer et modifier les instances d'un modèle.
- Test:
  - Contient des modèles pour écrire des tests pour un modèle.