Introduction à l'Ingénierie Dirigée par les Modèles (IDM)

Cedric Dumoulin

Les challenges

- Concevoir une application en s'abstrayant des technologies cibles
- Assurer la pérennité des applications conçues
  - maintenance, adaptation aux changements
- Augmenter la productivité
- Cibler plusieurs plateformes d'exécutions à partir d'une seule conception
- Réutiliser l'existant
- Automatiser la génération du code
- Contrôler, simuler, tester à différents niveaux

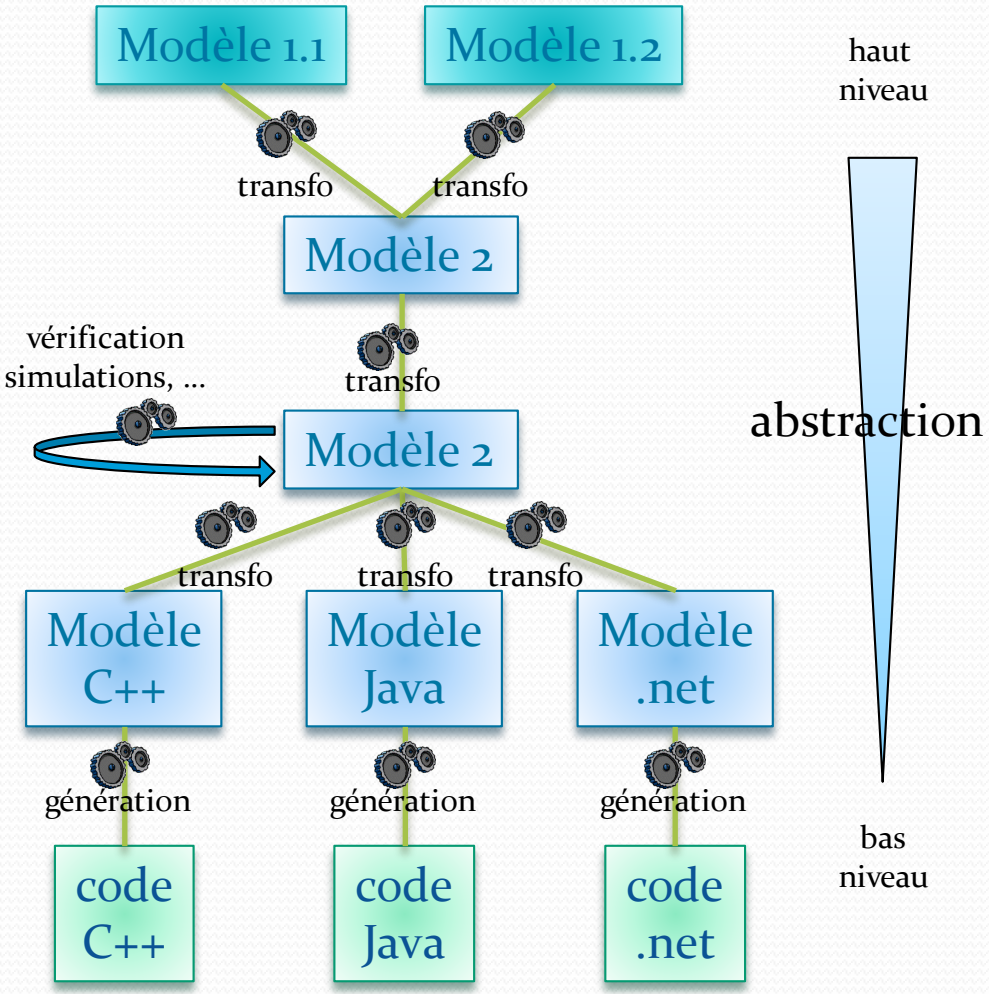
Que propose l'IDM ?

- **IDM : I**ngénierie **D**irigée par les **M**odèles
  ou MDE (**M**odel **D**riven **E**ngineering)

- Propose de modéliser les applications à un haut niveau d'abstraction
- Place le modèle au cœur du processus de conception
- Puis génère le code de l'application à partir des modèles

Plus en détails

- 1..n modèles de haut niveau
- des modèles intermédiaires
- 1..n technologies cibles
- transformation de modèles pour passer d'un niveau à l'autre
- génération de code à partir de modèles
- possibilité de contrôler, simuler et tester à différents niveaux



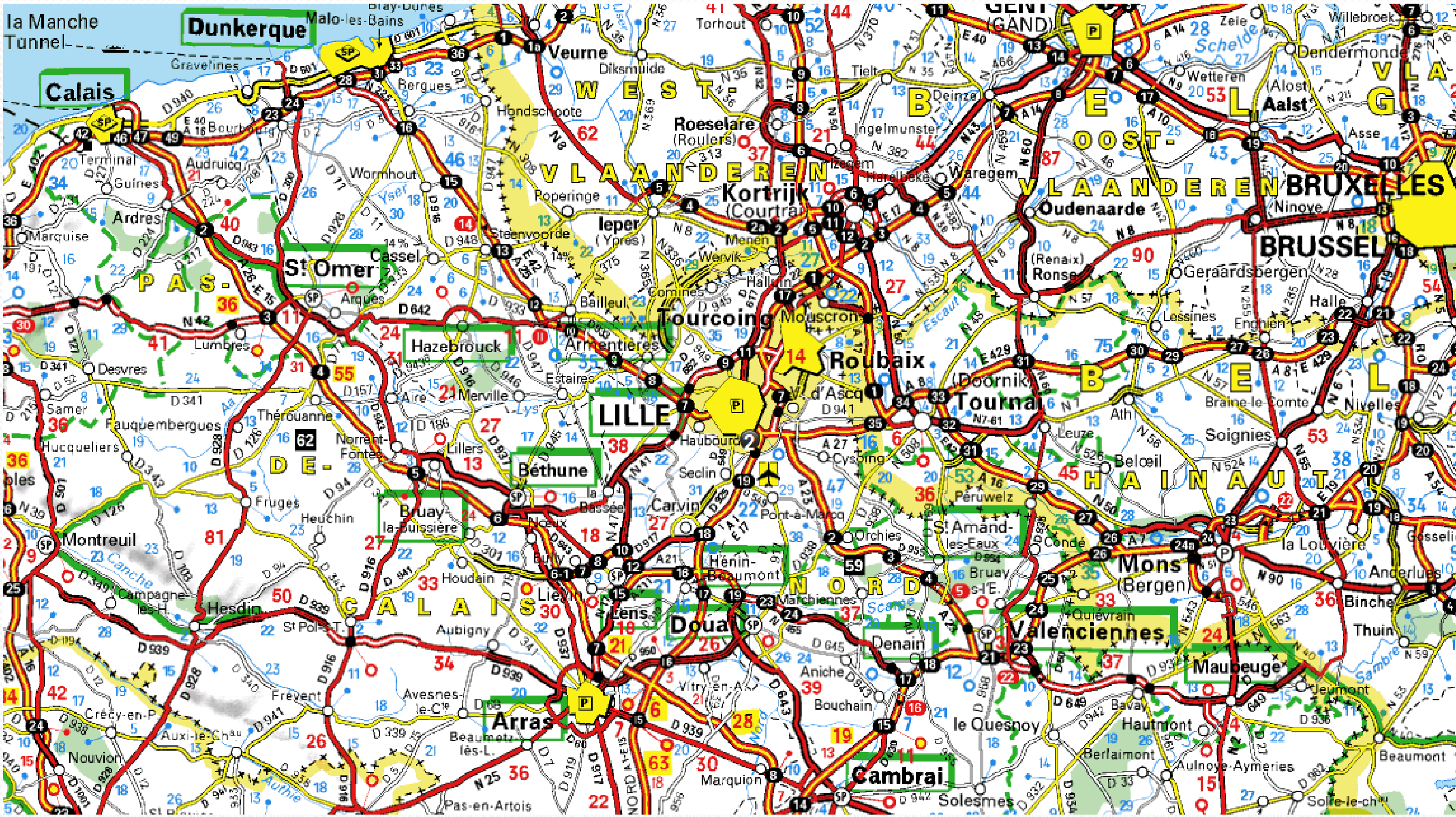
Qu'est ce qu'un modèle

- Définitions (Wikipedia by Google)
  - « *Un modèle mathématique est une traduction de la réalité pour pouvoir lui appliquer les outils, les techniques et les théories mathématiques* »
  - « [En économie] *Un modèle est une représentation de la réalité.* »
  - « *En informatique, un modèle a pour objectif de structurer les données, les traitements, et les flux d'informations entre entités.* »
  - C'est une abstraction d'un système

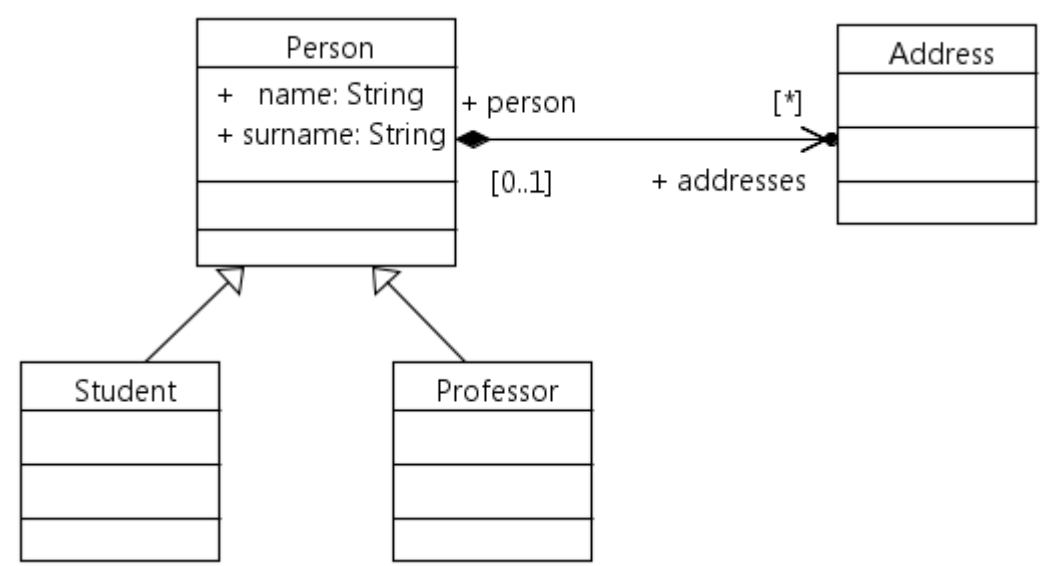
Le modèle doit pouvoir être utilisé pour répondre à des questions sur le système modélisé

« *This artwork is about different ways to show ideas. It presents one chair and three different ways of picturing this same chair.* »

Exemple de modèle



Exemple de modèle (2)



Avantages d'un modèle

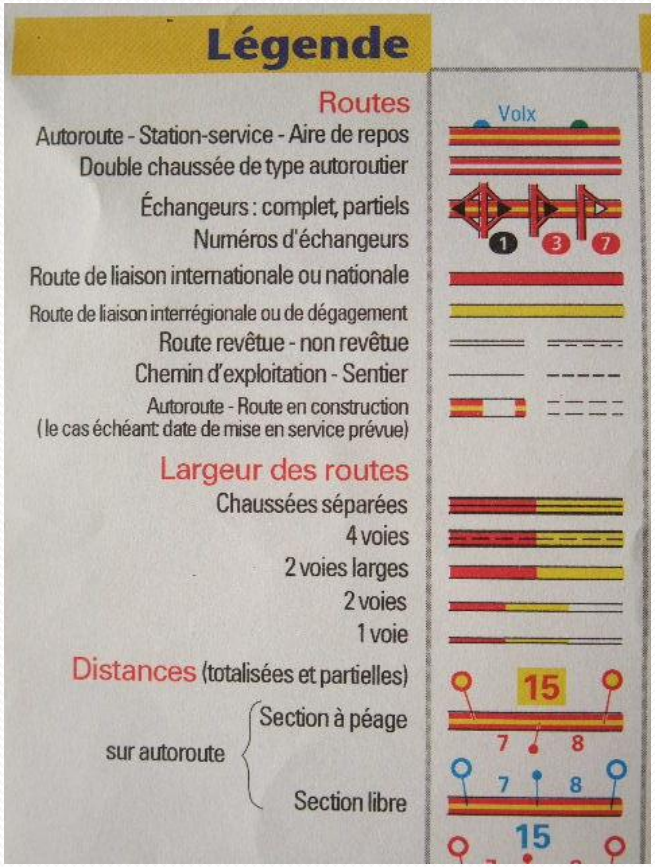
- Abstrait
  - Il fait ressortir les points importants tout en enlevant les détails non nécessaires
- Compréhensible
  - Il permet d'exprimer une chose complexe dans une forme plus facilement compréhensible par l'observateur
- Précis
  - Il représente fidèlement le système modélisé
- Prédictif
  - Il permet de faire des prévisions correcte sur le système modélisé
- Peu coûteux
  - Il est bien moins coûteux à construire et étudier que le système lui même

Pourquoi comprend t-on un modèle ?

- Parce qu'il est simple ?
- Parce qu'il représente la réalité ?
- Parce qu'il existe une légende !

Pourquoi comprend t-on un modèle ?

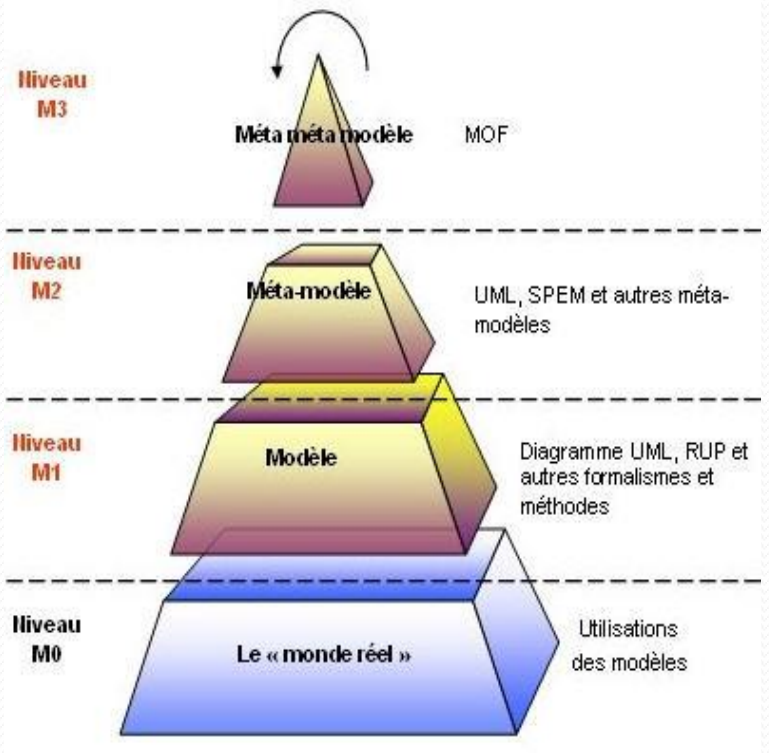
- La légende c'est
  - l'explication des concepts /dessins du modèle
  - la grammaire du modèle
- La légende est elle même un modèle !
  - On l'appelle *un métamodèle*
- Existe-t-il un métamodèle décrivant le modèle-légende ?
  - `nom, nom, … = signe, signe, …`
  - c'est le méta-métamodèle
- Existe-t-il un métamodèle décrivant le modèle décrivant le modèle-légende ?
- Et si oui, Existe-t-il …



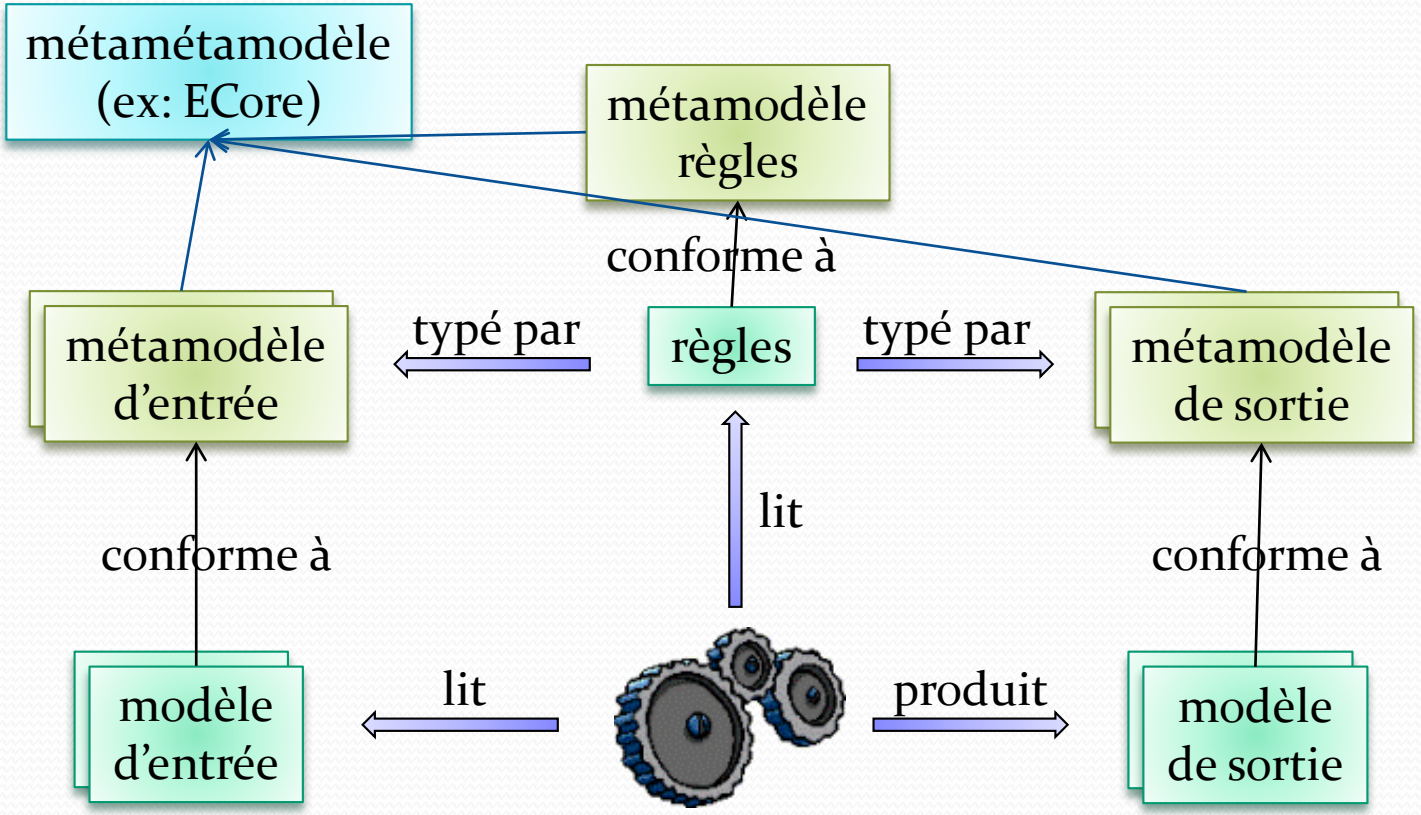
Récapitulons !
Modèle et Métamodèle

- Standardisé par l'OMG

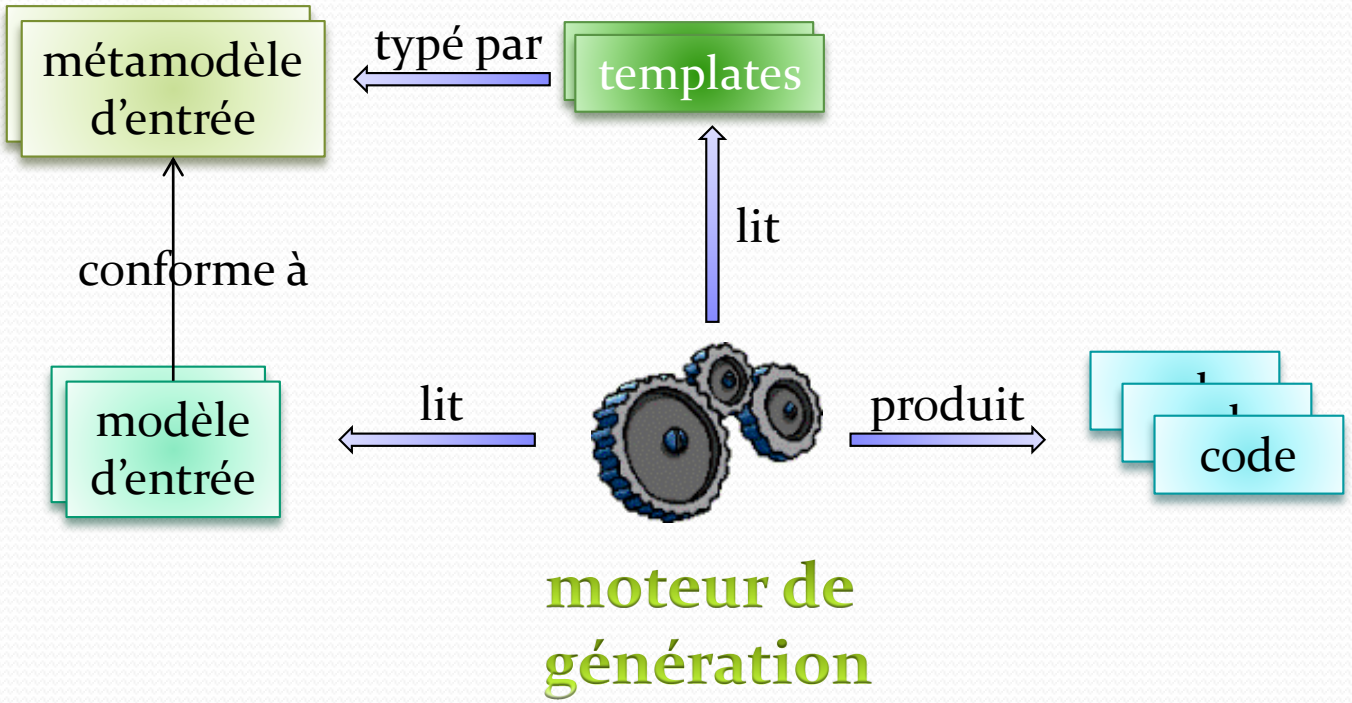
- méta-métamodèle
  - langage pour décrire des langages
  - ce décrit lui-même !
- métamodèle
  - langage pour décrire des modèles
- modèle
  - abstraction de la réalité ☺
- Le monde réel



D'un modèle à l'autre
La transformation de modèles



Du modèle au code
La génération de code

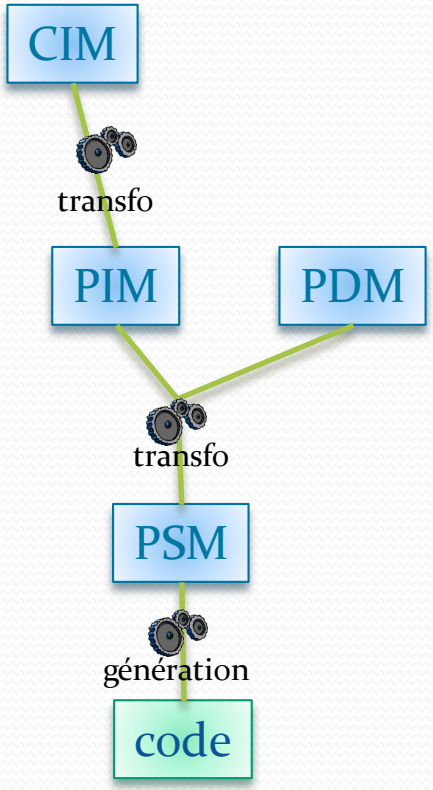


IDM vs MDA

- Model Driven Architecture
  - Architecture Dirigé par les Modèles
  - Modèle proposé par l'OMG (le nom est déposé !)
  - s'appuie sur UML

- Part d'un CIM (*Computation Independent Model*)
- Transforme en PIM (*Platform Independent Model*)
- Puis en PSM (*Platform Specific Model*)
- Et génère le code …

- Variante particulière de l'IDM
  - Plus restrictive



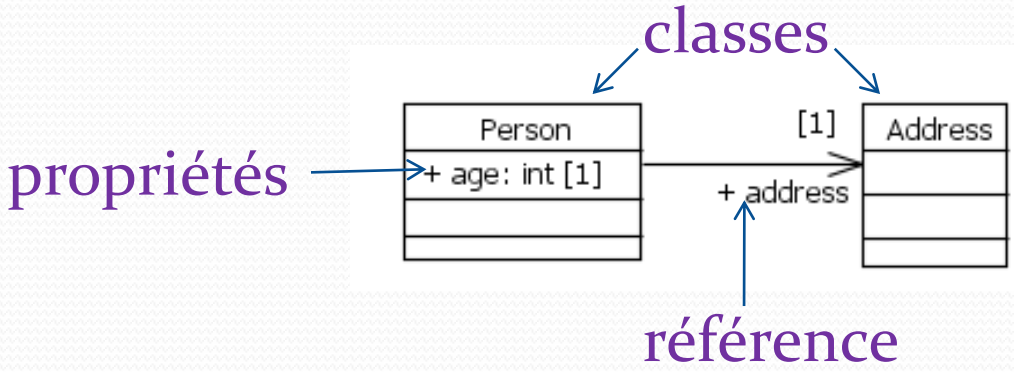
Construire un modèle

- 2 approches possibles

- Développer son propre langage (DSML)
  - Domain Specific Modeling Langage (ou DSL)
  - Prend la forme d'un métamodèle
  - approche ++ IDM
- Utiliser UML + un profil
  - approche ++ MDA

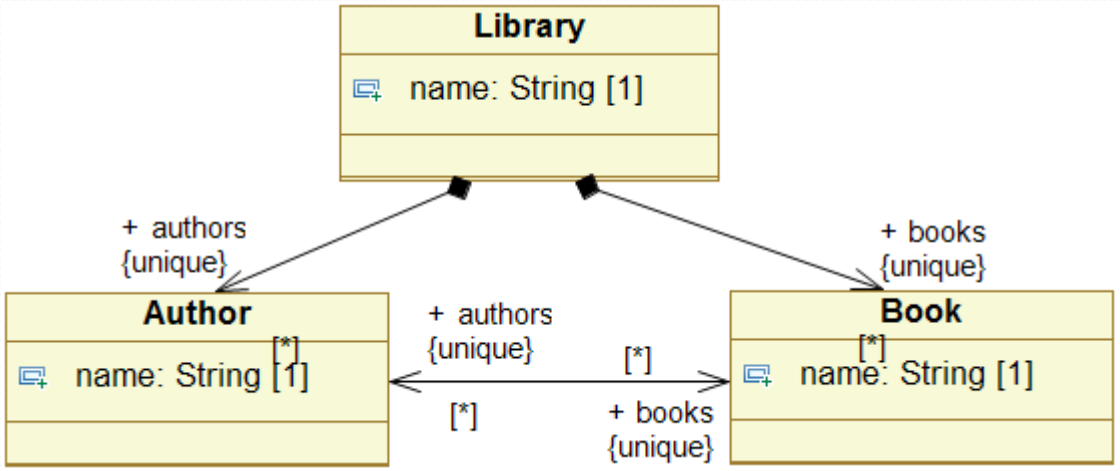
Concevoir un métamodèle

- Il existe plusieurs langages (i.e. méta métamodèles)
  - MOF, ECore, MetaGME, KM3, Kermeta, …
- Tous reposent sur les mêmes bases:
  - concept de classe
  - une classe et composé de propriétés
  - une propriété est appelé référence quand elle est typé par une autre classe



Concevoir un métamodèle

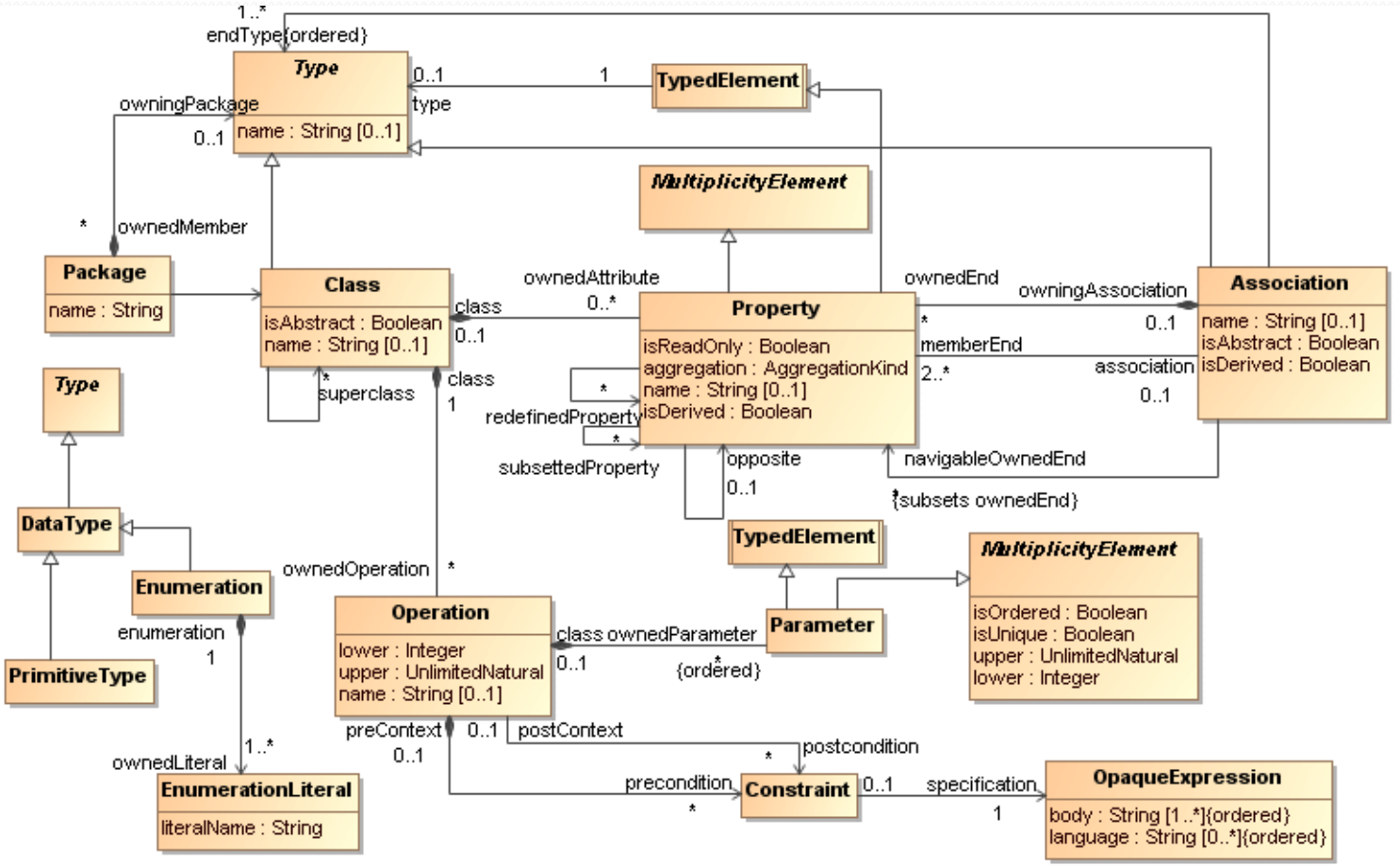
- 1 concept → 1 (méta)classe
- 1 relation entre concepts → 1 référence entre classes
- Peut se faire dans un diagramme de classes UML
- Exemple : bibliothèque de livres



UML 2

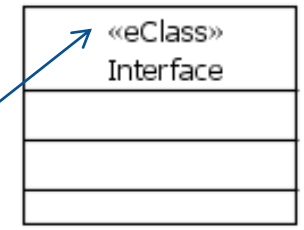
- Unified Modeling Language
  - standard OMG
- Langage de modélisation généraliste
- Permet de construire de nombreuses sortes de modèles
  - ne se limite pas à l'informatique
- Propose 13 types de diagrammes
  - structurels : classe, cas d'utilisation …
  - comportementaux : activités, états …

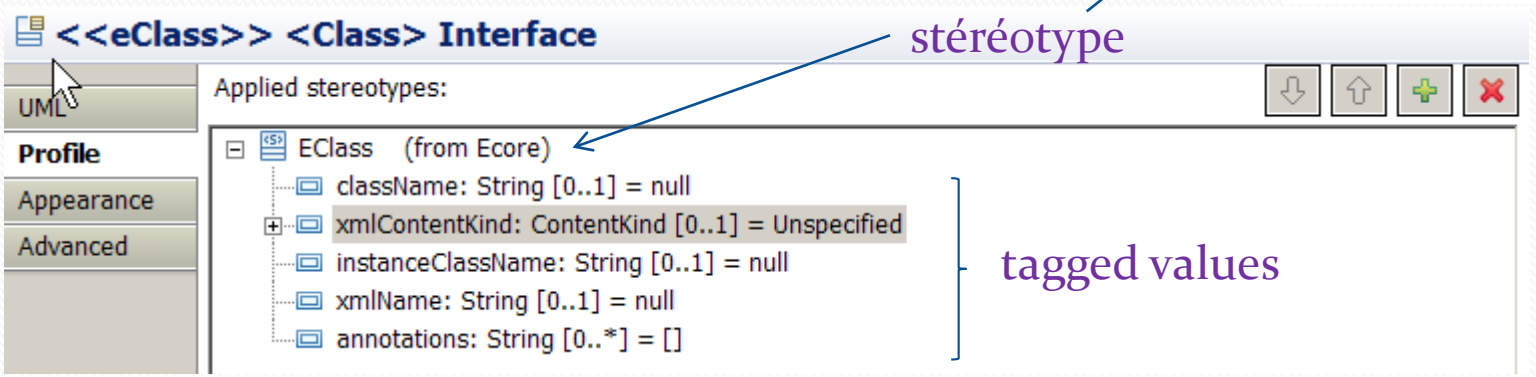
Métamodèle UML 2



Etendre UML : Les Profils

- Un profil permet d'étendre ou de contraindre UML
- Un profil contient un ensemble de <<stéréotypes>>
- Un stéréotype permet d'ajouter de l'information à un élément UML
- Un stéréotype peut contenir des 'tagged values' (valeurs marquées)
  - paires `{nom, valeur typé}`





UML

Langage généraliste

- standard– facilite la compréhension en dehors du domaine
- beaucoup de concepts

Doit être étendu

- Pas toujours adéquat

Syntaxe concrète

- Des éditeurs graphique existe

DSL / métamodèle

Langage dédié à un domaine

- Concepts propre au métier
- Peu de concepts

Plus simple

- à transformer
- à comprendre
- à manipuler

Pas de syntaxe concrète

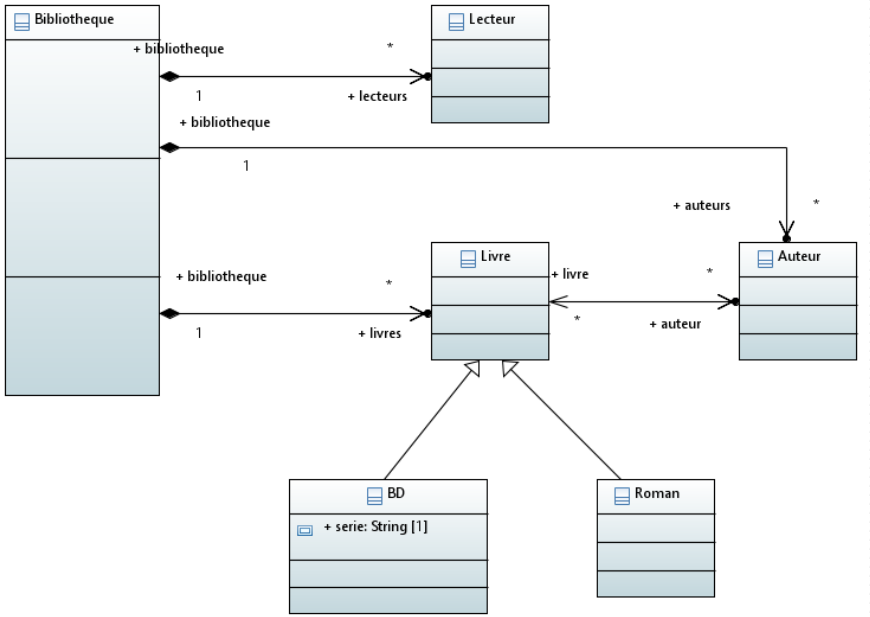
- il faut développer ses propres éditeurs graphique

- Une transformation permet de passer de UML à une DSL

Syntaxe abstraite
Syntaxe concrète

- Un métamodèle représente une syntaxe abstraite
  - Définition des concepts
- Pour construire un modèle, il faut une syntaxe concrète
  - Définition graphique ou texte
- Un même métamodèle peut avoir plusieurs syntaxes concrètes !
  - souvent une graphique et une textuelle

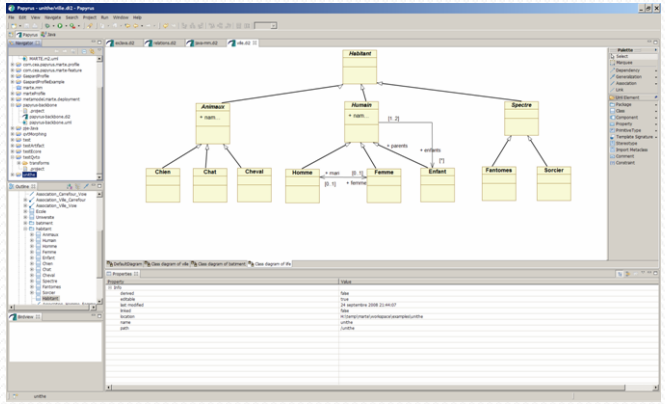
Plusieurs syntaxe concrètes




```
<packagedElement xmi:type="uml:Class" xmi:id="_8hhTcKKq" name="Lecteur"/>
 <packagedElement xmi:type="uml:Class" xmi:id="_96cOMKKq" name="Livre">
  <ownedAttribute xmi:type="uml:Property" xmi:id="_CMl_4KKr" name="auteur"
                                          type="__B-eAKKqEeel" association="_CMcO4KKr">
   <lowerValue xmi:type="uml:LiteralInteger" xmi:id="_EOwE4KKr"/>
   <upperValue xmi:type="uml:LiteralUnlimitedNatural" xmi:id="_EOo9YKKr" value="*"/>
  </ownedAttribute>
 </packagedElement>
 <packagedElement xmi:type="uml:Class" xmi:id="__B-eAKKq" name="Auteur"/>
 <packagedElement xmi:type="uml:Class" xmi:id="_AQ5IMKKr" name="Emprunt"/>
```

Quelques outils actuels

- Papyrus UML
  - modeleur UML 2 open source
  - projet Eclipse

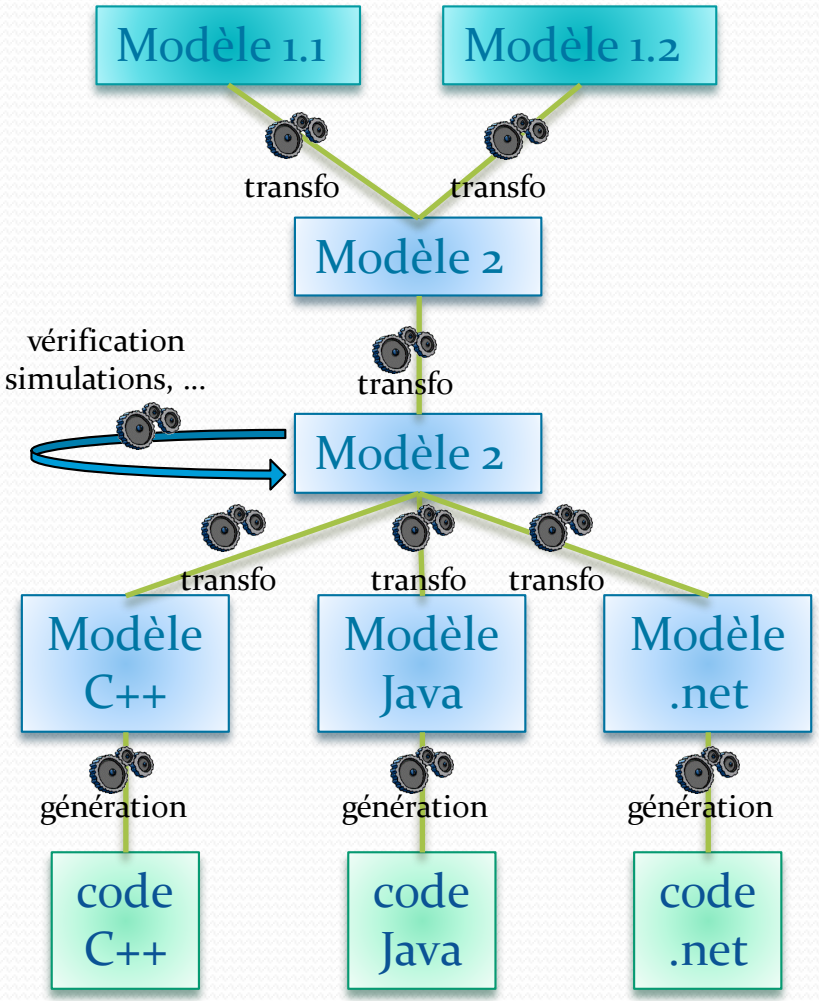


- Éclipse - EMF
  - framework pour manipuler modèle et métamodèle

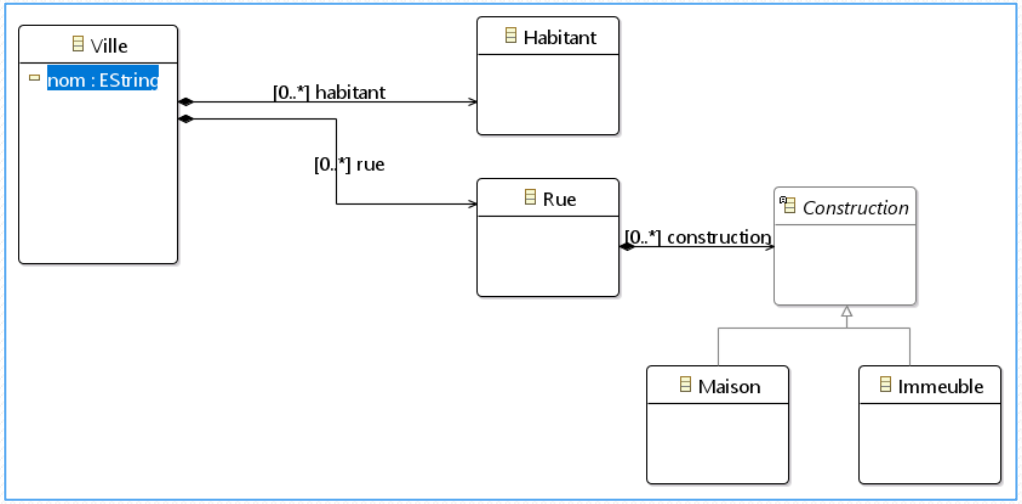
- QVT (QVTO)
  - standard pour transformer des modèles
- M2T (Acceleo)
  - standard pour la génération de texte (code)

L'IDM répond t-elle aux challenges ?

- s'abstraire des technologies cibles
- Assurer la pérennité
- Augmenter la productivité
- Cibler plusieurs plateformes d'exécutions
- Réutiliser l'existant
- Automatiser la génération du code
- Contrôler, simuler, tester à différents niveaux



Demo !

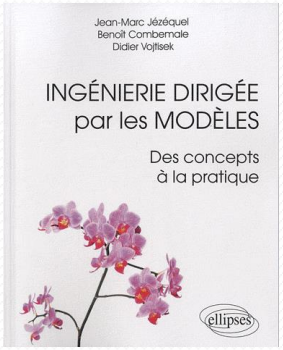


- Modéliser une ville
  - les rues
  - les bâtiments
  - les habitants
- Générer les programmes Java
  - pour manipuler des modèles de villes
- Construire des modèles !!

- Mieux que les syms ☺ !!

En savoir plus

- Wikipedia
  - http://fr.wikipedia.org/wiki/Ingénierie_dirigée_par_les_modèles
- OMG – www.omg.org
  - UML, QVT, M2T, …
- Etat de l'art
  - http://hal.archives-ouvertes.fr/docs/00/37/15/65/PDF/mde-stateoftheart.pdf
- Ingénierie dirigée par les modèles - Des concepts à la pratique
  - Jean-Marc Jézéquel, Benoît Combemale, Didier Vojtisek



En savoir plus

- Eclipse Modeling
  - http://www.eclipse.org/downloads/
- Eclipse EMF
  - http://www.eclipse.org/modeling/emf/
  - Tutorial : Help > Help Contents > EMF Developer Guide > Tutorials > Generating an EMF Model
- Papyrus – modeleur UML Open Source
  - Papyrus (projet Eclipse)
    - projet Eclipse – CEA – LIFL – Airbus – Atos
    - www.eclipse.org/papyrus
- QVTo
  - http://www.eclipse.org/m2m/
- Acceleo
  - http://www.eclipse.org/modeling/m2t/?project=acceleo

That's All Folks!