

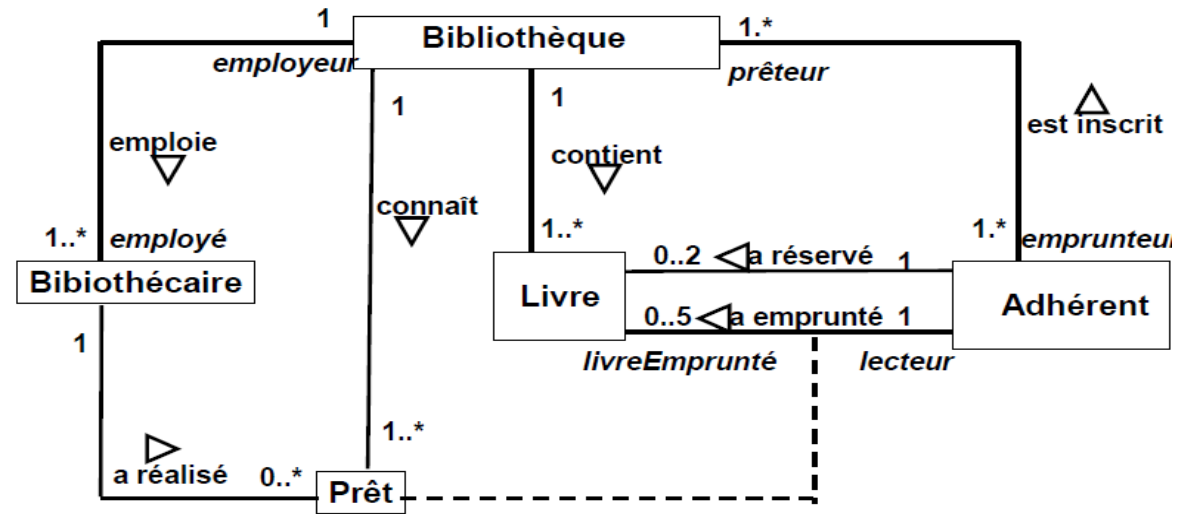
Architectures dirigées par les modèles

Model Driven Architecture

Quelques définitions: modèle, métamodèle

- **Un modèle** M est une représentation simplifiée du monde, en fait, d'une partie du monde S qu'on appelle système

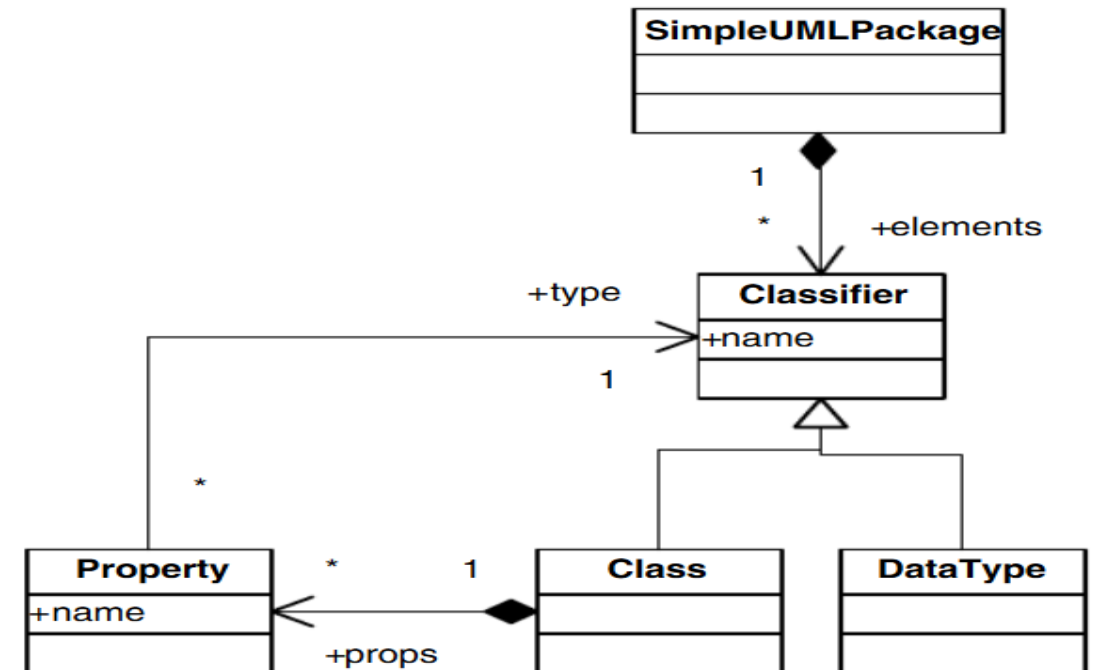
Exemple: modèle représentant les Entités impliquées dans la gestion d'une bibliothèque et leurs relations



- Le modèle est décrit à l'aide d'un langage ou **formalisme de modélisation**

Exemple : UML

- Le **métamodèle** ou **métaformalisme** est le langage d'expression de modèles de formalismes

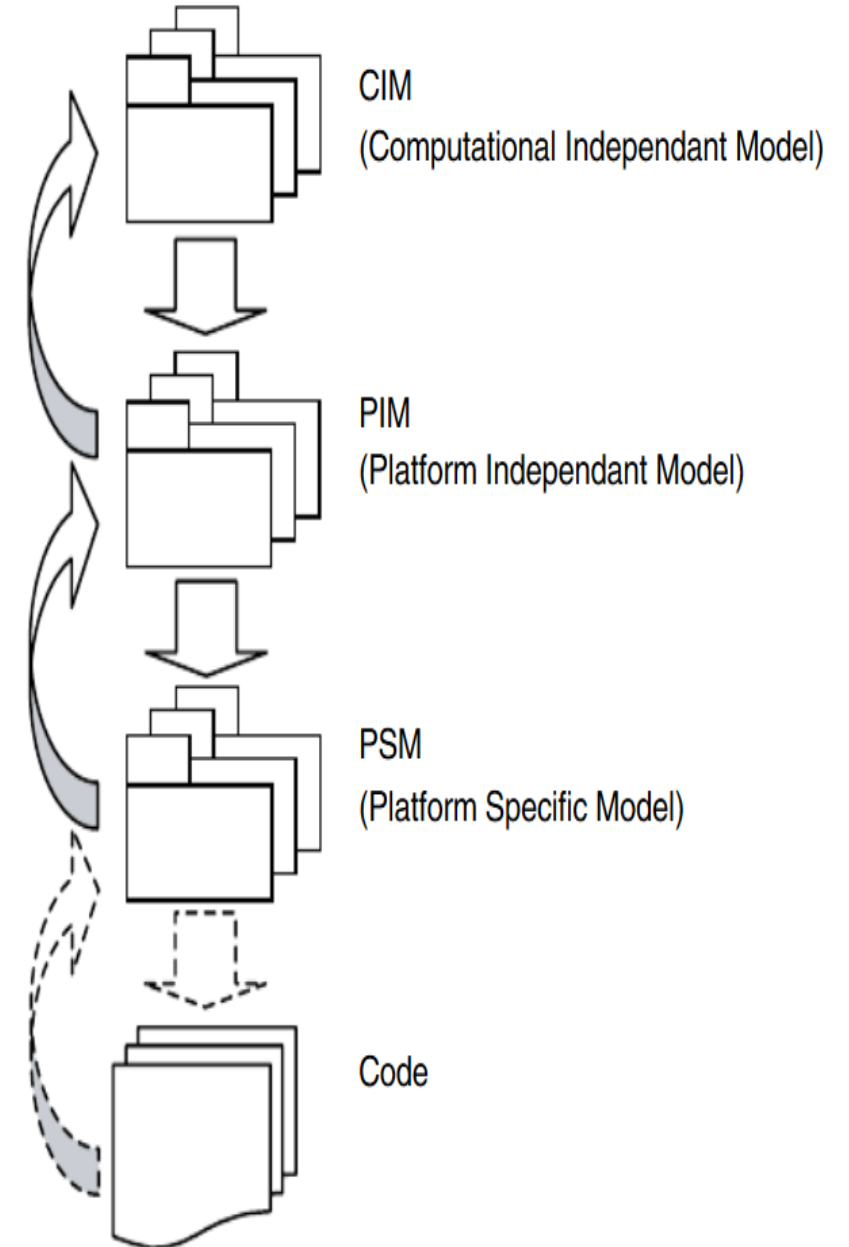


MDA: principe

- Utilisation de modèles aux différentes phases du cycle de développement d'une application
- Elaboration de modèles pérennes, indépendants des détails techniques des plates-formes d'exécution (J2EE, .Net, PHP ou autres)
- Génération automatique de la totalité du code des applications pour permettre un gain significatif de la productivité: transformation de modèles

MDA: principe

- Elaboration d'un ou de plusieurs modèles d'exigences (**CIM**).
- Elaboration des modèles d'analyse et de conception abstraite de l'application (**PIM**). Ceux-ci doivent en théorie être partiellement générés à partir des CIM afin que des liens de traçabilité soient établis. : aucune information sur les plates-formes d'exécution.
- Transformation des PIM en y ajoutant les informations techniques relatives aux plates-formes d'exécution: PSM



modèle d'exigences CIM (Computation Independent Model)

- Spécifier les exigences du client: quels sont les services offerts par l'application et quelles sont les autres entités avec lesquelles elle interagit: conceptualisation des besoins du système
- En UML: un modèle d'exigences peut se résumer à un diagramme de cas d'utilisation. Ces derniers contiennent en effet les fonctionnalités fournies par l'application ainsi que les différentes entités qui interagissent avec elle (acteurs) sans apporter d'information sur le fonctionnement de l'application

Le modèle d'analyse et de conception abstraite PIM (Platform Independent Model)

- Les modèles d'analyse et de conception doivent être indépendants de toute plate-forme d'implémentation, qu'elle soit J2EE, .Net, PHP, etc
- Faire le lien entre le modèle d'exigences et le code de l'application
- Etre suffisamment précis et contenir suffisamment d'information pour qu'une génération automatique de code soit envisageable

Le modèle de code ou de conception concrète PSM (Platform Specific Model)

- Servent essentiellement à faciliter la génération de code à partir d'un modèle d'analyse et de conception
- Contiennent toutes les informations nécessaires à l'exploitation d'une plate-forme d'exécution, comme les informations permettant de manipuler les systèmes de fichiers ou les systèmes d'authentification.

De l'importance des formalismes de modélisation

- Langage qui permet d'exprimer des modèles, par exemple UML

Un formalisme définit les concepts ainsi que les relations entre concepts nécessaires à l'expression de modèles

Par exemple, pour UML: classe, objet, association

- Pour définir des transformations, Il faut exprimer des liens entre les concepts des différents formalismes

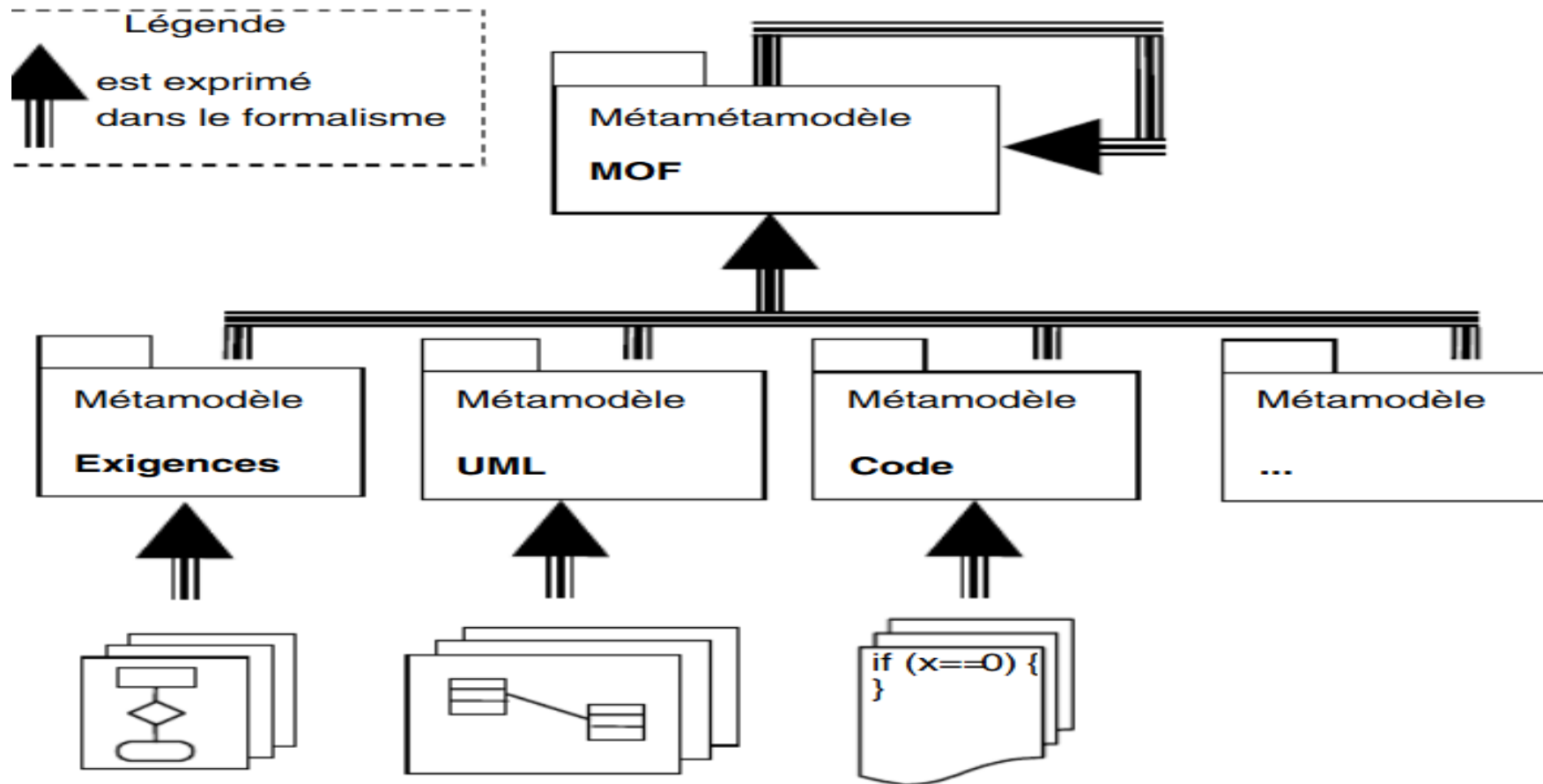
Par exemple, lien entre le concept de classe d'UML et le concept de classe en Java

- Les métamodèles fournissent la définition des entités d'un modèle, ainsi que les propriétés de leurs connexions et de leurs règles de cohérence.

MOF les représente sous forme de diagrammes de classes.

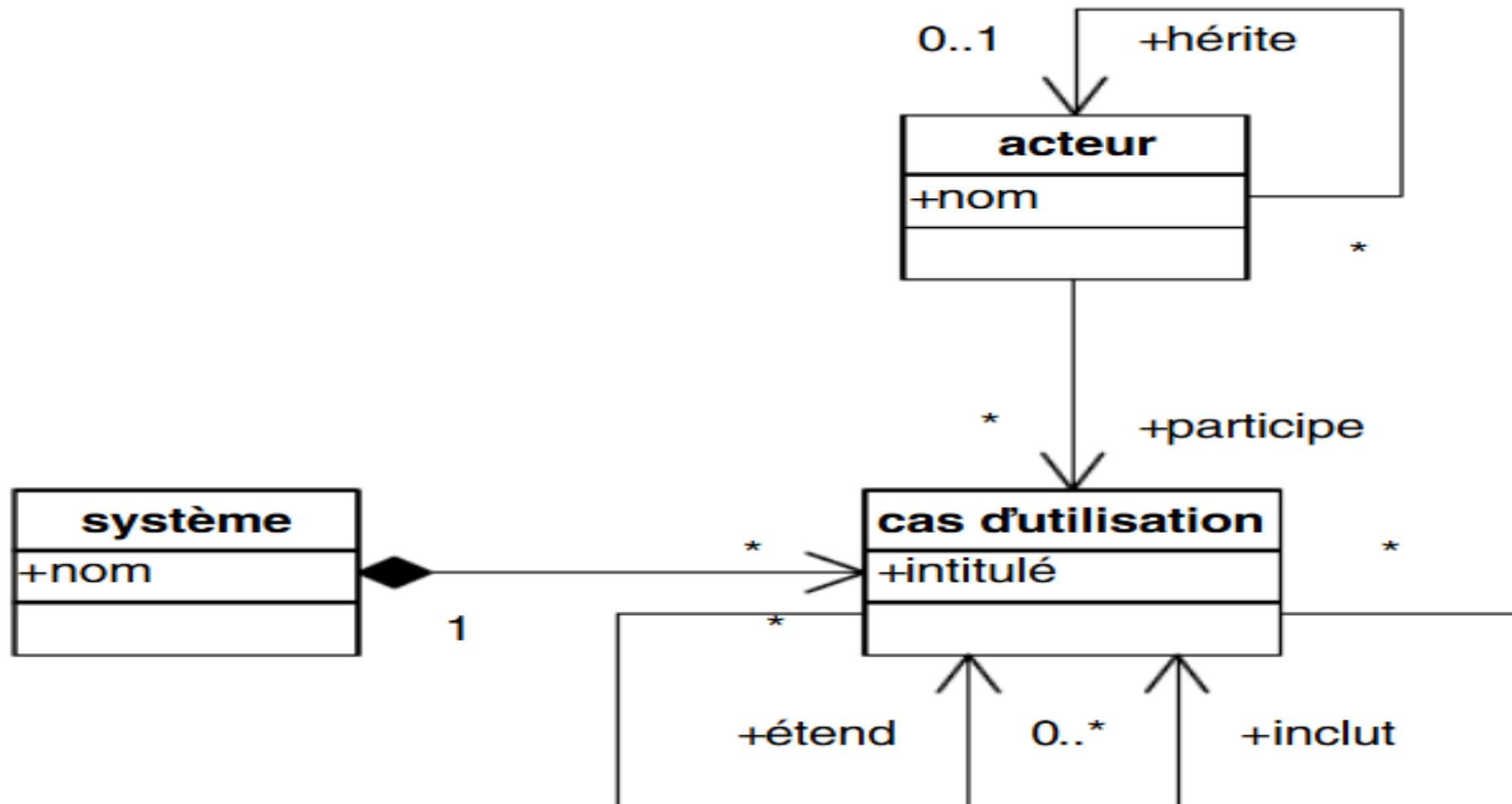
Les trois niveaux de MDA

- Seuls trois niveaux suffisent: le modèle, le métamodèle ou formalisme de modélisation et le métaformalisme. Ce dernier est son propre formalisme: **MOF (Meta Object Facility)**



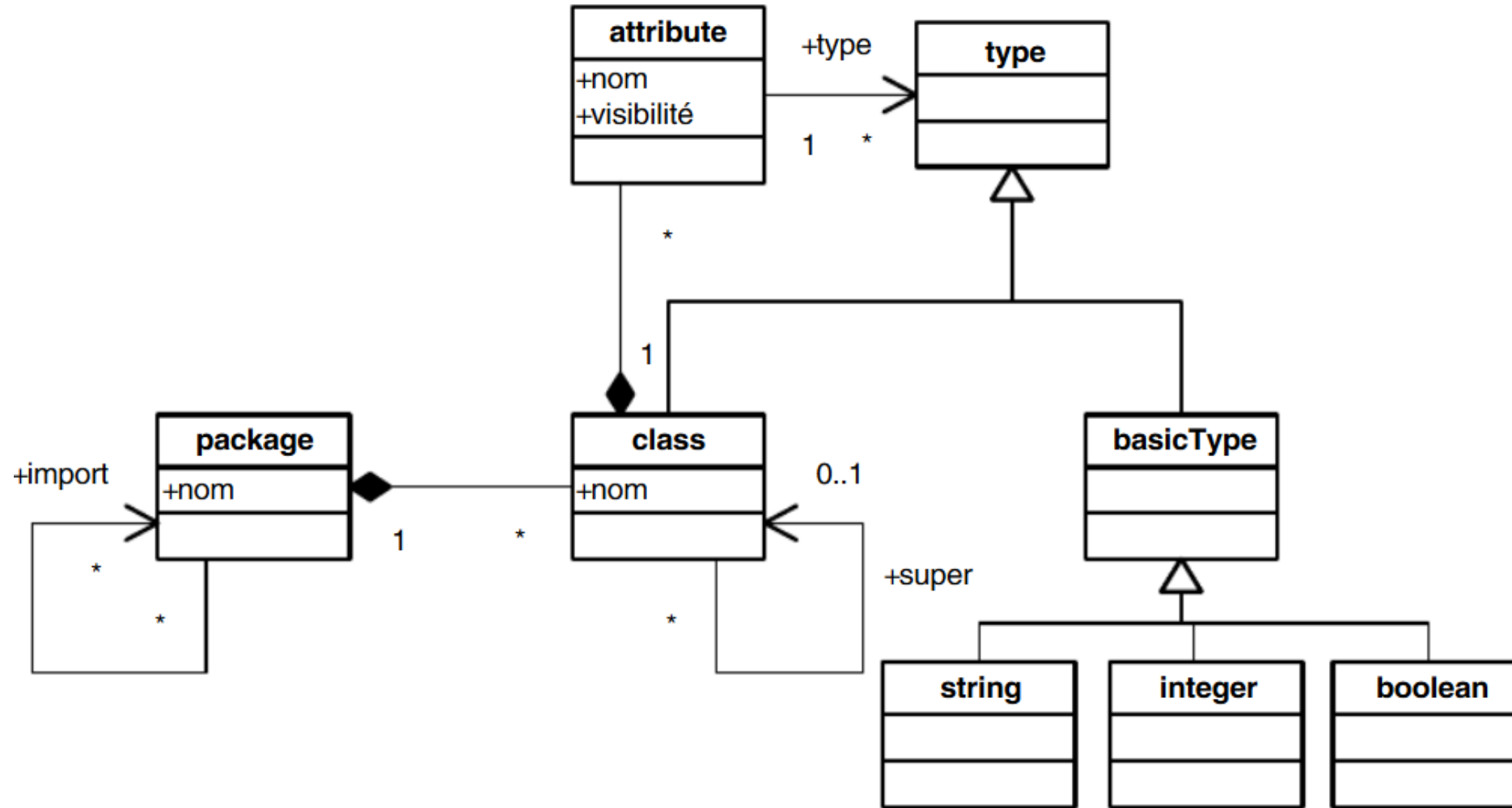
Exemples de métamodèles

- Métamodèle de diagramme de cas d'utilisation



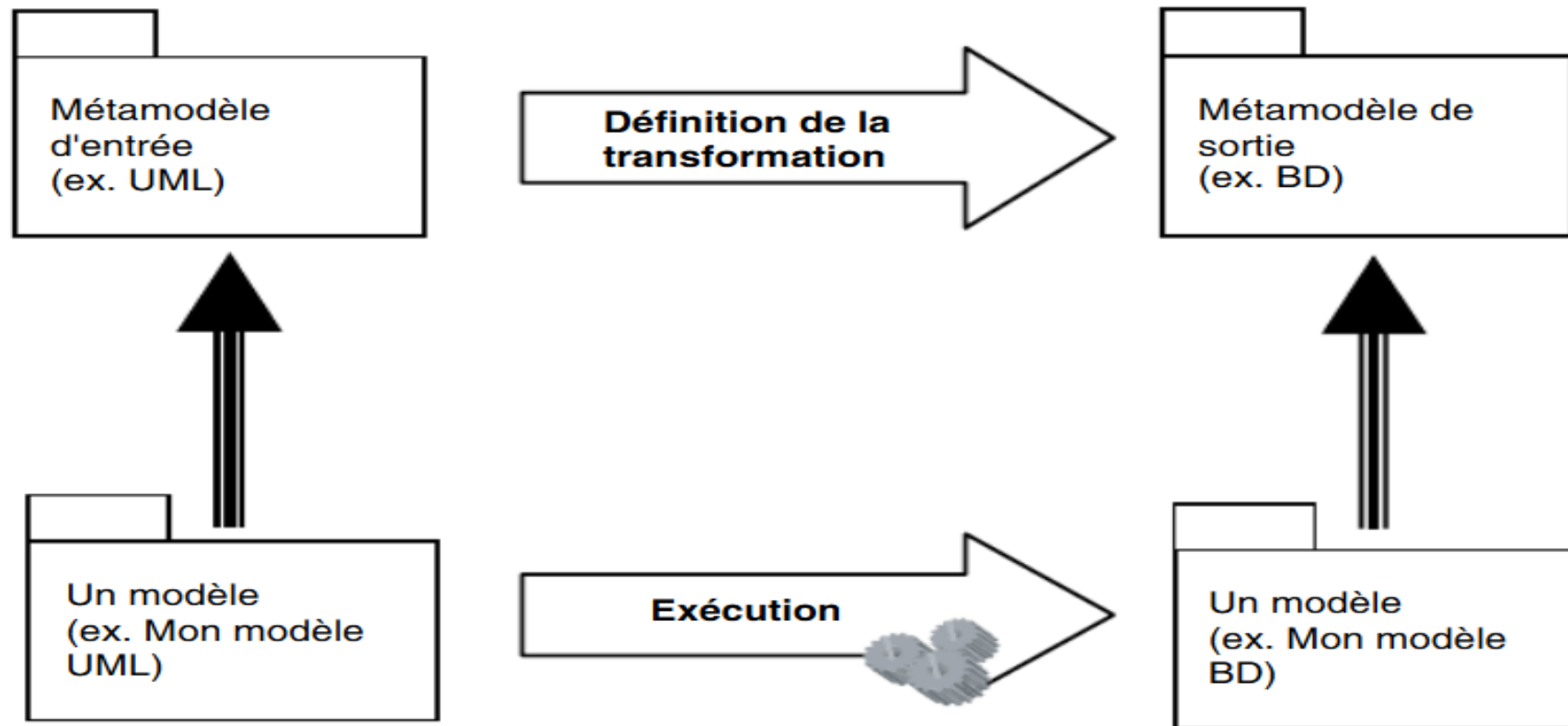
Exemples de métamodèles

- Métamodèle de diagramme de classes



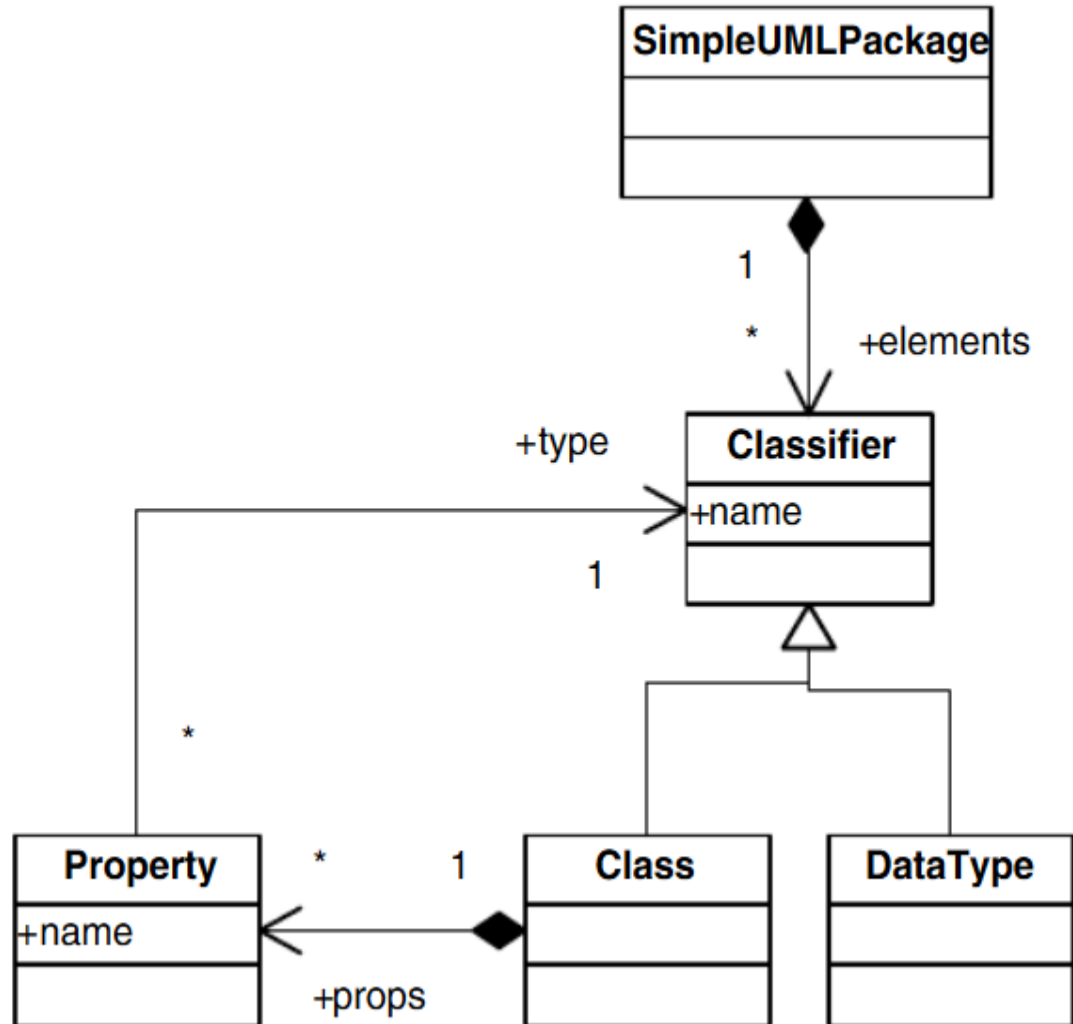
Transformation de modèles: principe

- La clé de la mise en production de MDA
- Une transformation :
 - une fonction qui prend en entrée un ensemble de modèles et qui fournit en sortie un ensemble de modèles. Les modèles sont structurés par des métamodèles
 - Se décrit au niveau des métamodèles source et cible

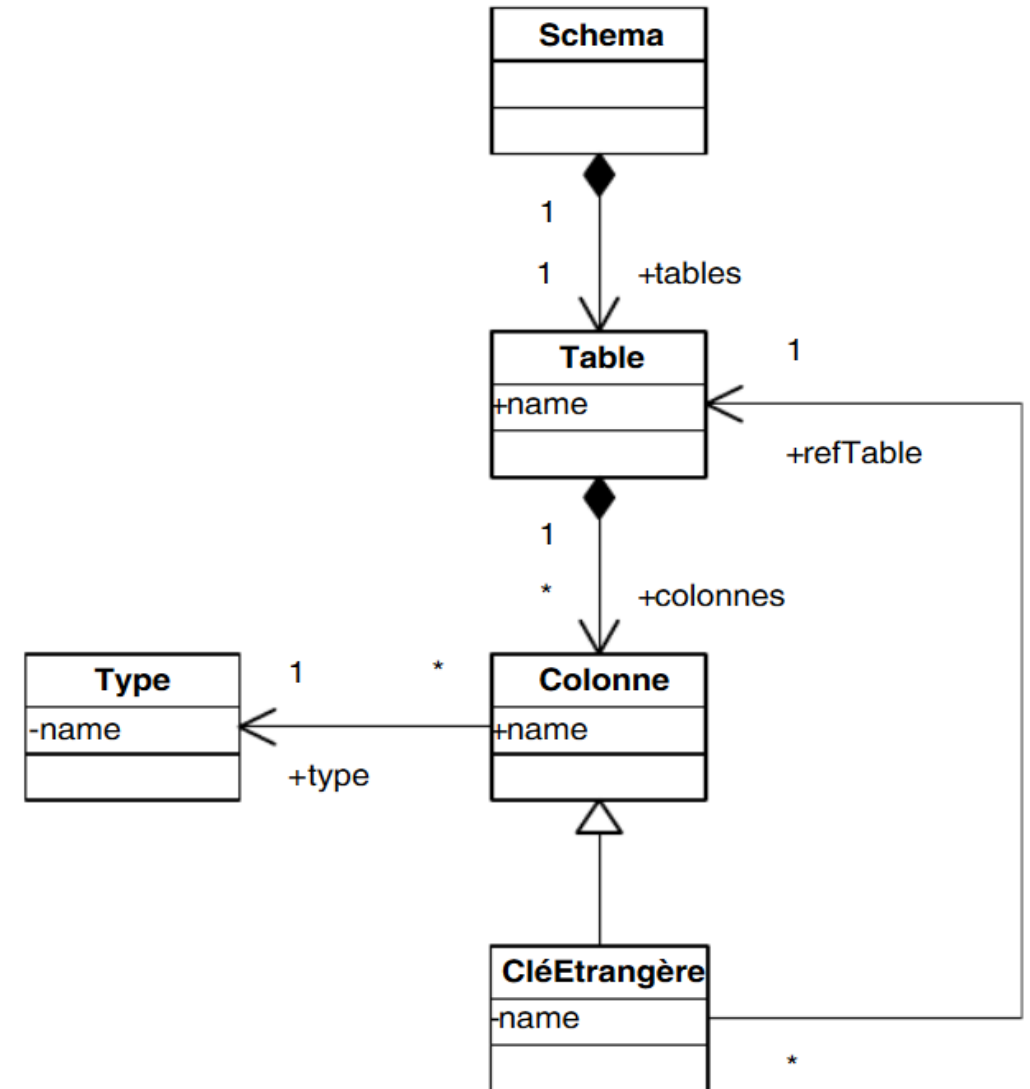


Transformation de modèles: Exemple

Métamodèle UML



Métamodèle d'un schéma relationnel



Transformation d'un modèle UML vers un schéma de bases de données (BD)

Règles de transformation:

1. À tout package UML correspond un schéma.
 2. À toute classe UML du package correspond une table dans le schéma.
 3. À toute propriété UML d'une classe correspond une colonne dans la table :
 - Si le type de la propriété est une classe UML, la colonne est une clé étrangère.
 - Si le type de la propriété est un type de donnée UML, la colonne est du type correspondant.
 4. À tout type de donnée UML d'un package correspond un type de donnée dans le schéma.
-
- Programmer les transformations de modèles en utilisant les langages de programmation orientée objet
 - Exemple d'EMF (Eclipse Modeling Framework): il est possible de définir un métamodèle et de générer les interfaces tailored dédiées à ce métamodèle afin de pouvoir manipuler les instances du métamodèle dans Eclipse.
 - D'autres outils: IBM Rational Software Modeler et Softeam MDA Modeler

- [Blanc, Xavier](#), [Salvatori, Olivier](#). MDA en action : Ingénierie logicielle guidée par les modèles.
[Eyrolles](#)