

MDA

Cours de 2^e année ingénieur
Spécialisation « Génie Informatique »

Fabien.Romeo@fromeo.fr
<http://www.fromeo.fr>

Plan général¹

- Introduction (Séance 1)
 - Principes généraux
 - Modèles et méta-modèles
- Manipulation de modèles (Séances 2&3)
 - JMI
 - EMF
- Transformation de modèles (Séance 4)
 - Principes généraux
 - Règles de transformation par programmation
- Exécution de modèles (Séance 5)
 - Executable MDA
 - PauWare

¹ prévisionnel !

Principes généraux

Model-Driven Architecture

- Nouvelle méthode de génie logiciel (2000), initié par l'Object Management Group (ceux du standard UML)
- Méthode = Langage
+ Démarche
+ Outils
- A donné naissance au domaine plus général du Model-Driven Engineering (Ingénierie Dirigée par les Modèles)

Motivation 1

- Constante évolution des technologies
- Pour une entreprise, il est crucial de posséder la dernière technologie
 - Problème : pour la même application, il faut tout recommencer (\$\$\$)
- Exemple :
 - Procédural (C, Cobol, ...) → Objet
 - Objet (Java, C++, ...) → Composant
 - Composant (Java EE, .Net, ...) → Services
 - Services (WS-*, REST, OSGi, ...) → Futur paradigme

Motivation 2

- A quoi sert la modélisation pour le logiciel, puisqu'il faudra toujours produire du code ?
 - A quoi servent les diagrammes UML ?
 - Pour être à la mode, il faut en faire
 - Pour remplir un rapport et montrer qu'on travaille sur des choses complexes
 - A rien, il n'y a pas besoin de diagrammes UML quand on fait de l'open source, la documentation c'est le code source

Motivation 3

- La miniaturisation et les avancées technologiques ont permis d'embarquer du logiciel dans des systèmes tout aussi nombreux que différents
 - Ex : ordinateurs, pda, téléphones, set-top boxes, automobile, domotique, ...
- Comment développer une application logicielle pour qu'elle soit exécutable sur tous ces appareils ?

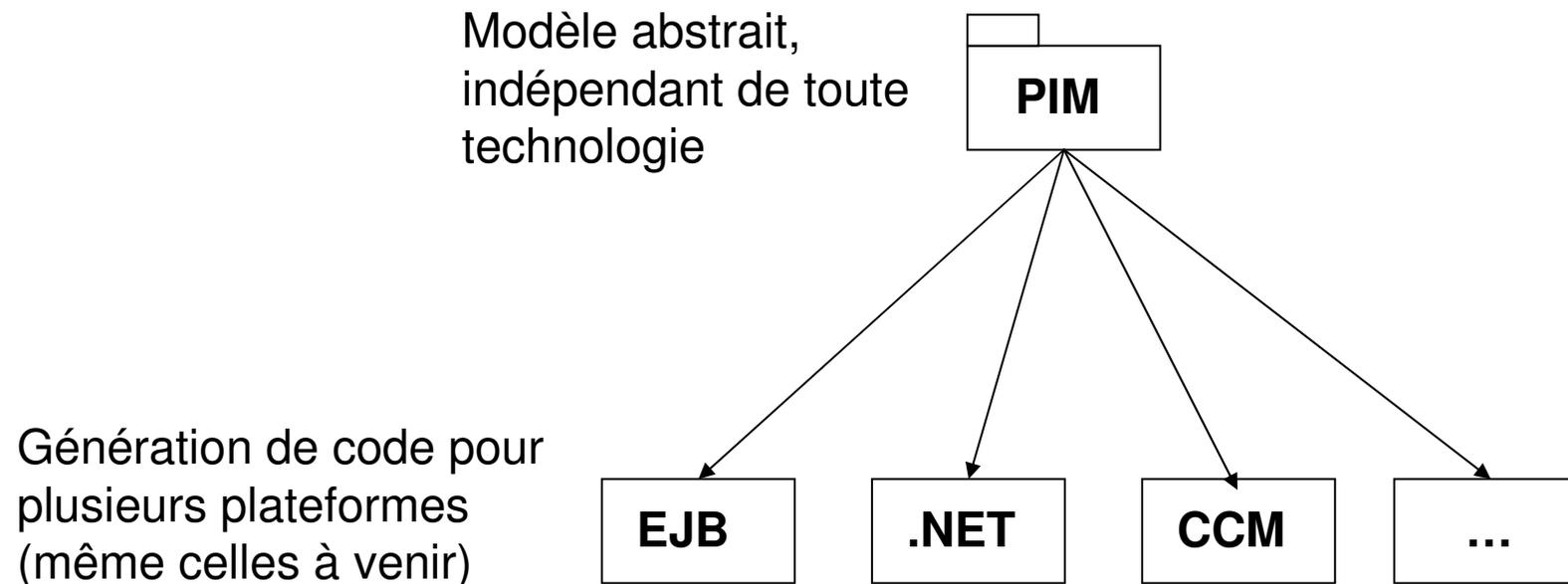
Intention du MDA

- Pérennité des savoir-faire
 - La durée de vie des modèles doit être plus longue que celle du code (modèles abstraits)
- Gains de productivité
 - L'utilisation des modèles ne doit plus être seulement contemplative mais productive (automatisation des opérations sur les modèles et génération de code)
- Prise en compte des plateformes d'exécution
 - La prise en compte des plateformes doit être explicitée par des modèles (modélisation des plateformes et lien avec les modèles des applications)

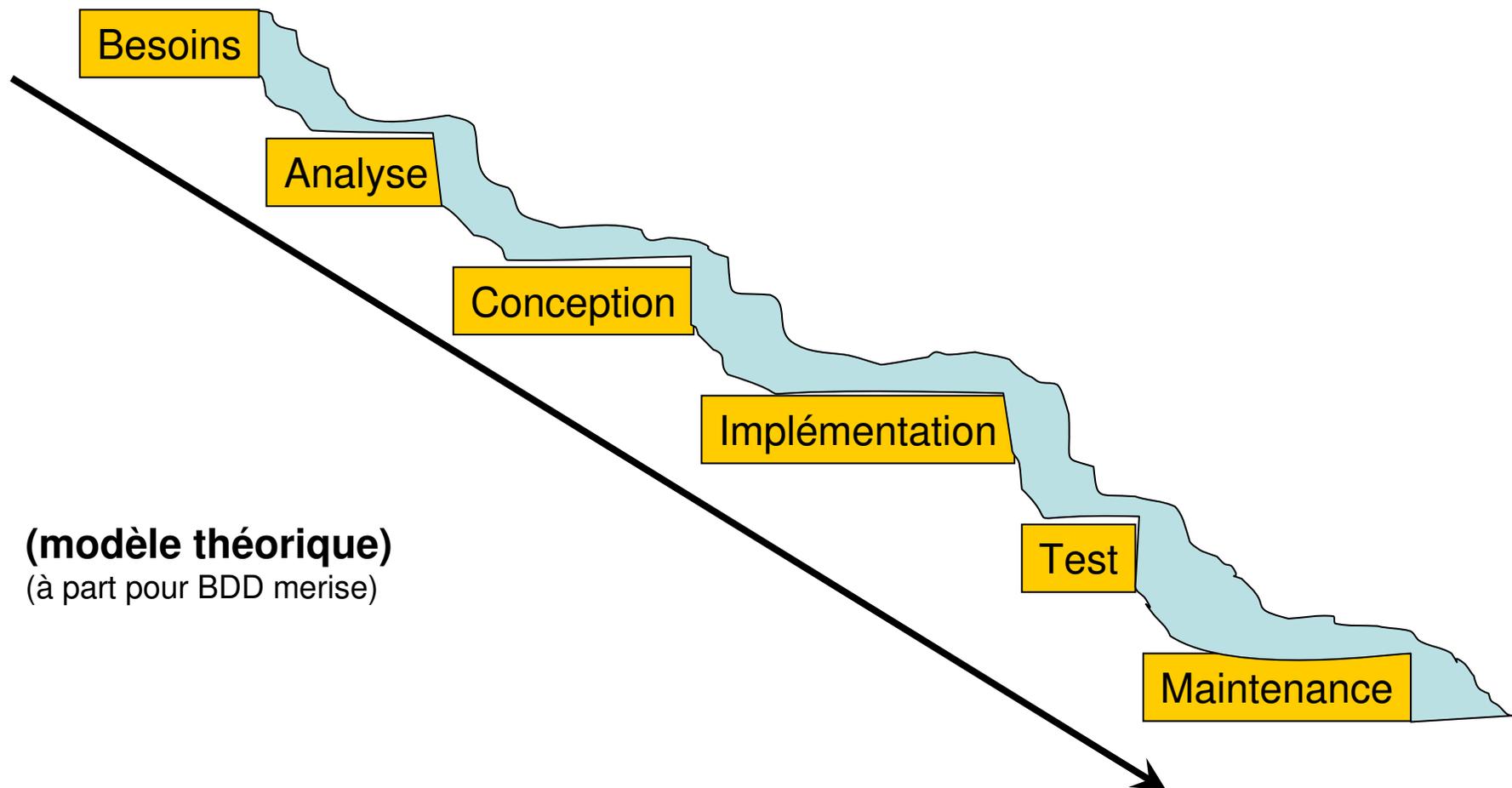
Modèles du MDA

- Le MDA définit 2 principaux niveaux de modèles d'application
 - PIM (Platform Independent Model)
 - Modèle spécifiant une application indépendamment de la technologie mise en œuvre
 - Modélise seulement la partie métier
 - PSM (Platform Specific Model)
 - Modèle spécifiant une application après projection sur une plateforme technologique donnée (définie par un PDM)
- Et un niveau pour les plateformes
 - PDM (Platform Description Model)
 - Modèle décrivant une plateforme de déploiement

Model Once, Generate Everywhere

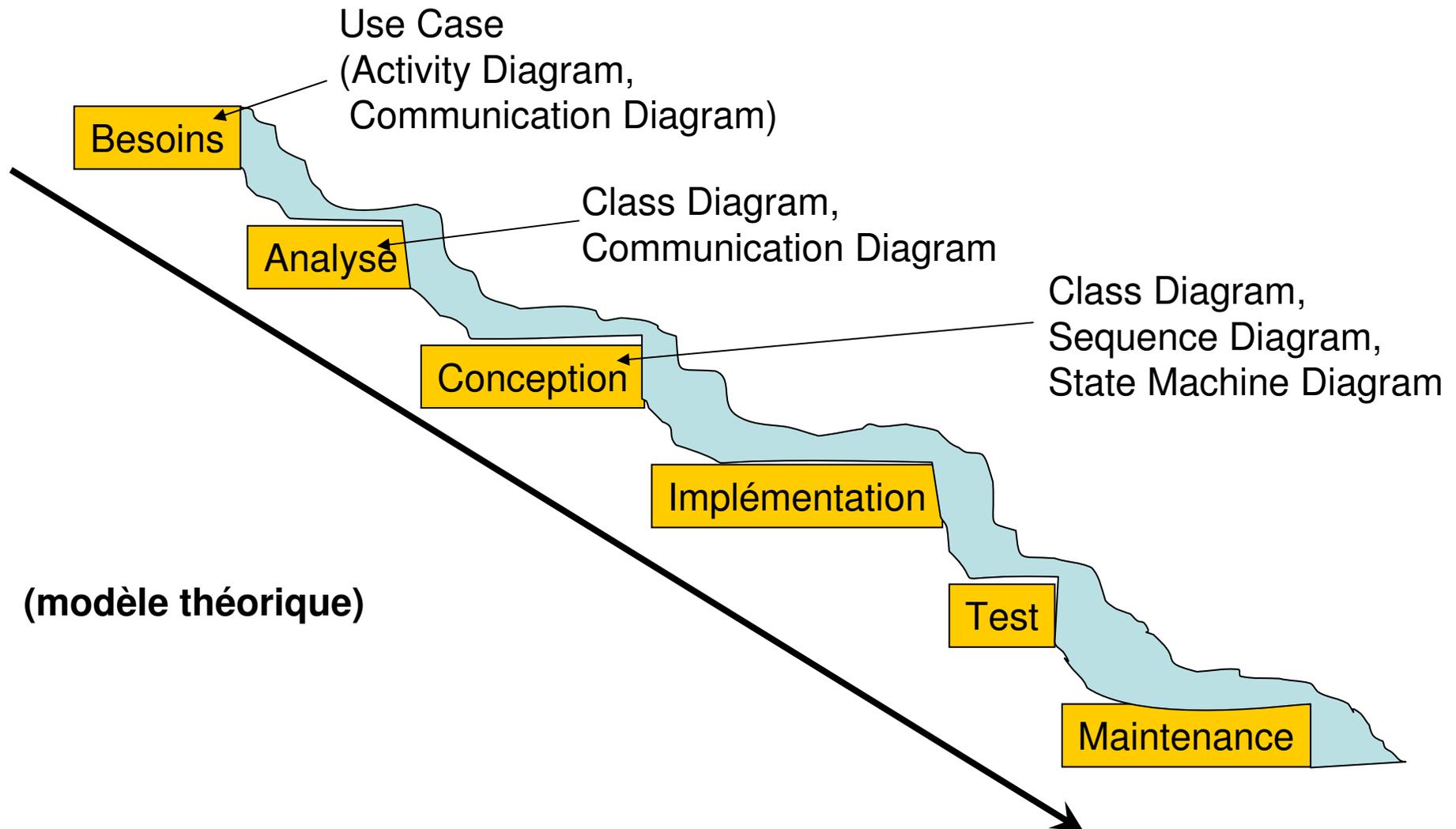


Cycle de développement en cascade (rappel)

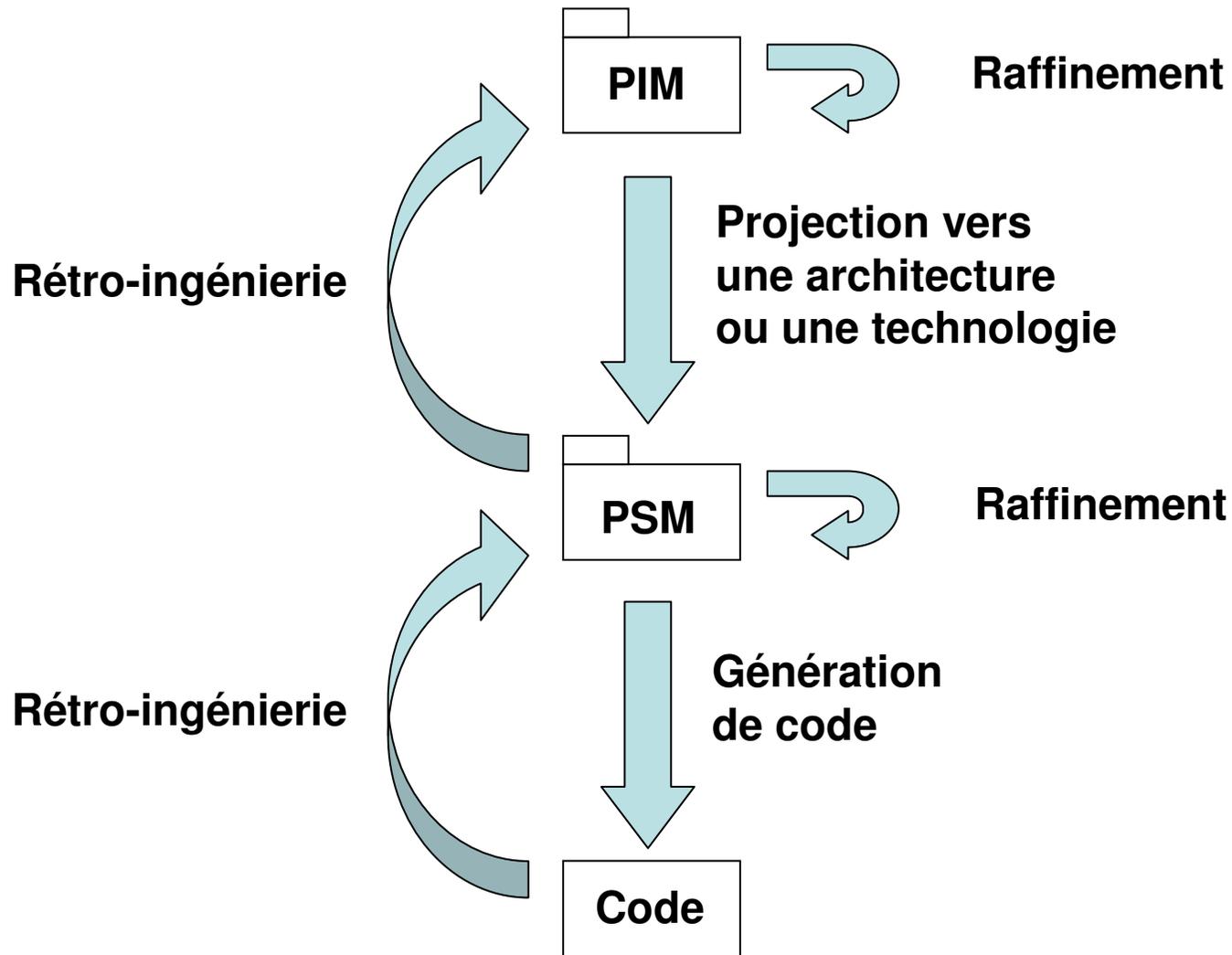


(modèle théorique)
(à part pour BDD merise)

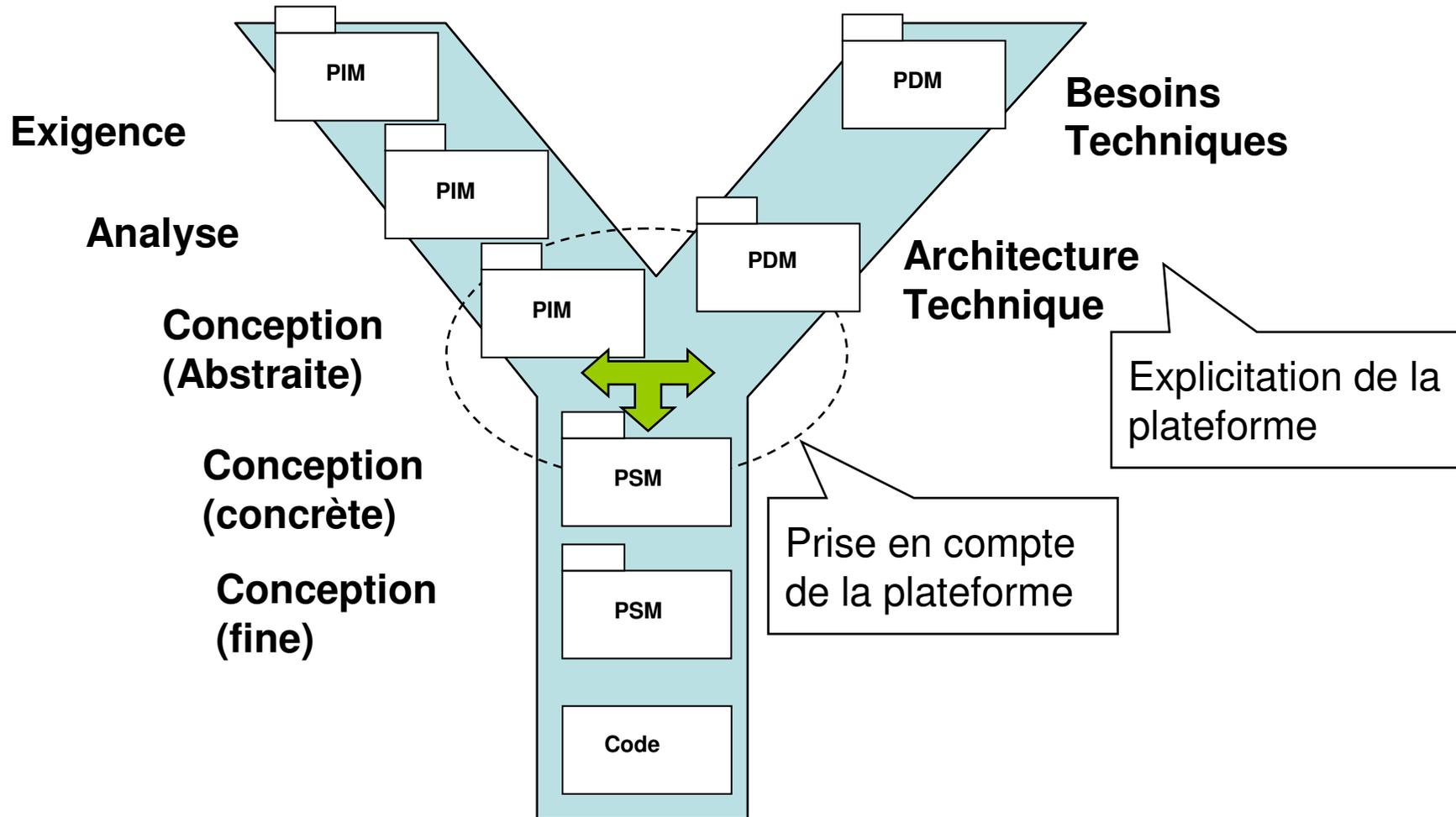
Cycle de développement en cascade avec UML (rappel)



Niveaux de modélisation

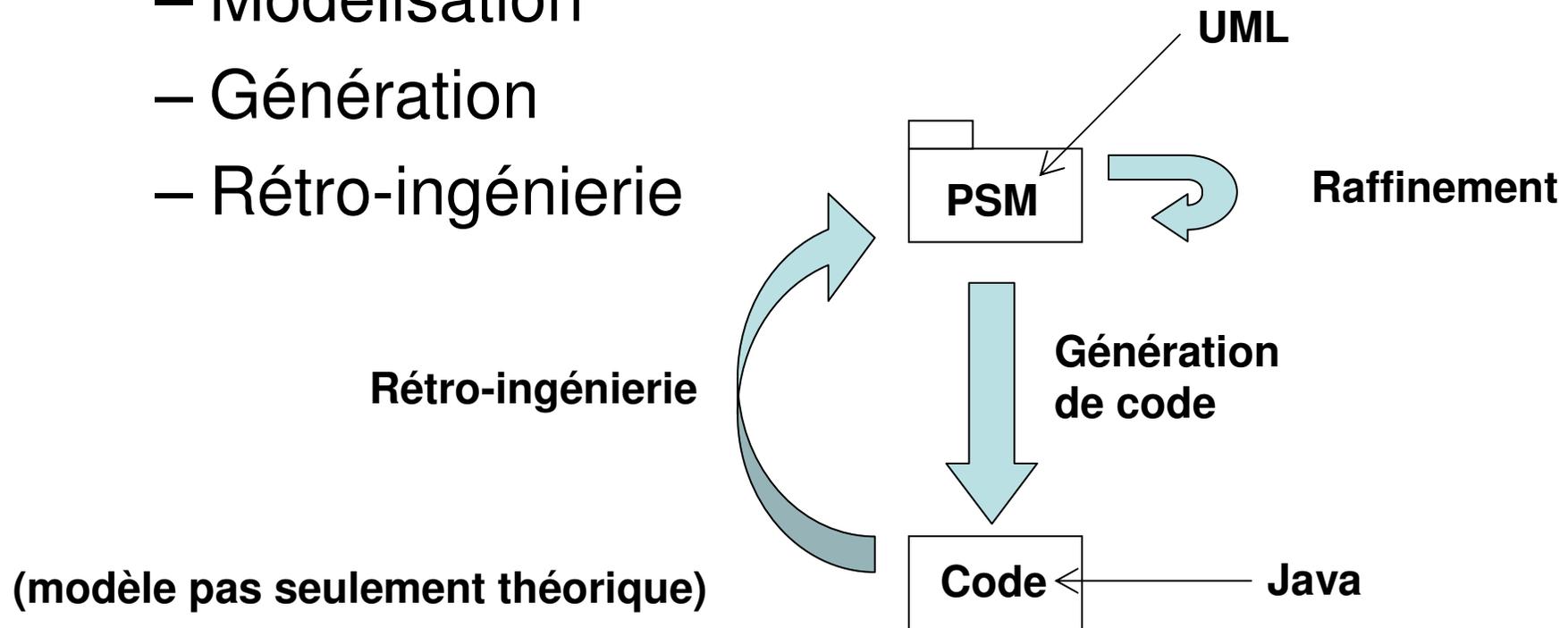


Cycle de développement et plateforme



Travaux pratiques (n°1)

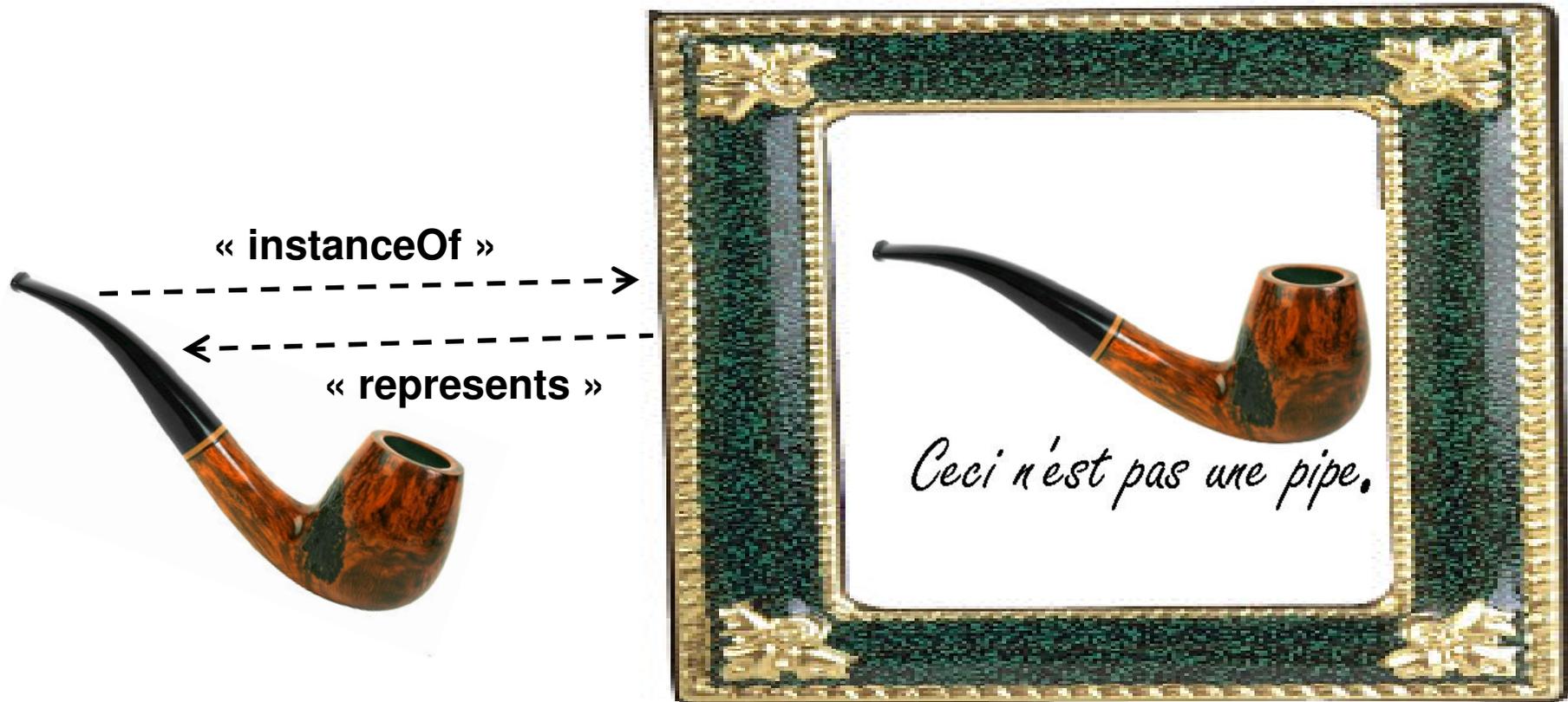
- Utilisation de NetBeans pour pratiquer le cycle de développement du MDA
 - Modélisation
 - Génération
 - Rétro-ingénierie



Modèles et méta-modèles

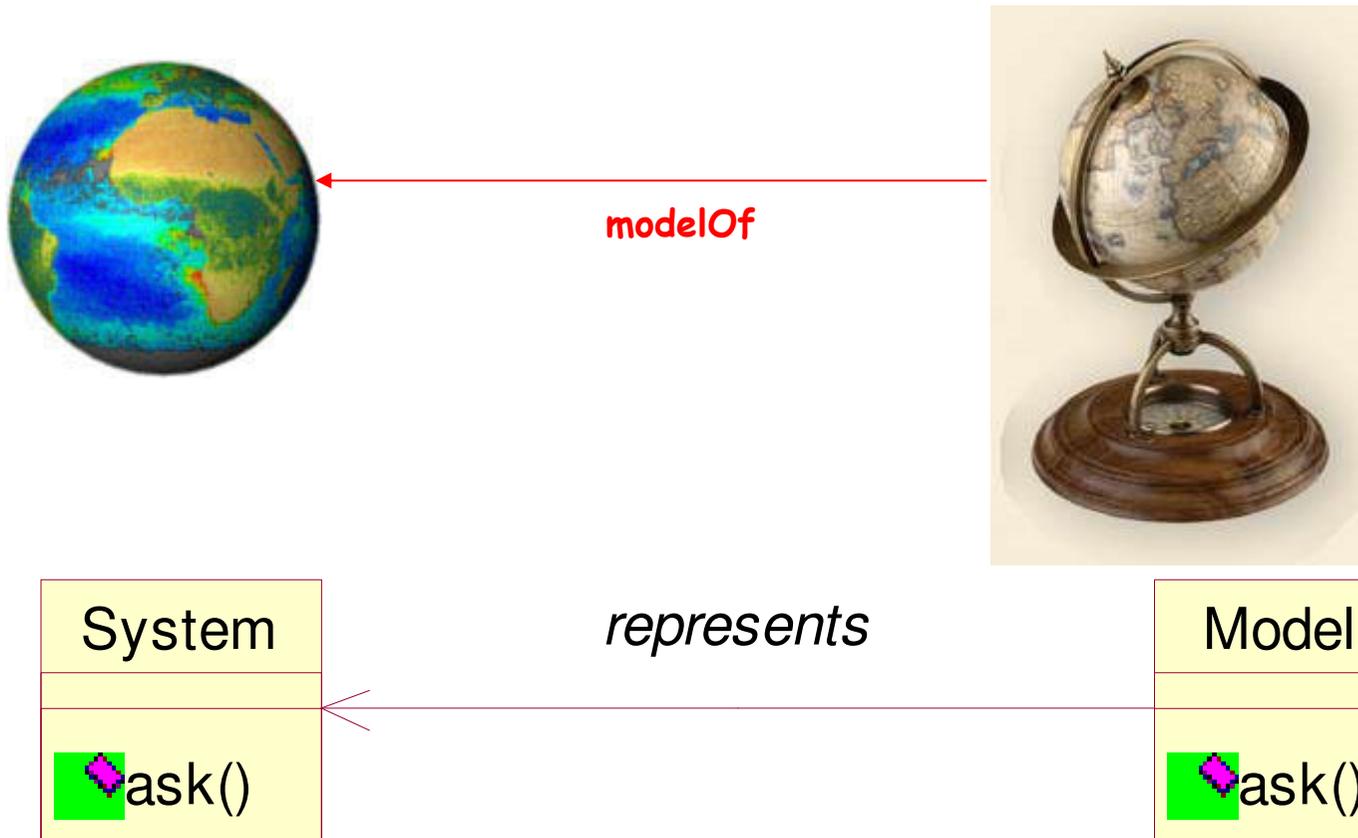
Systeme et modele

- Un modele est une abstraction d'un systeme physique, construit dans une intension particuliere.



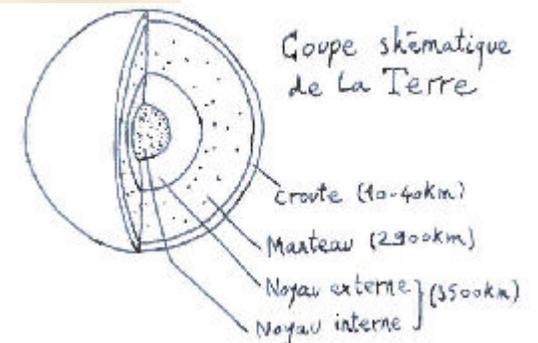
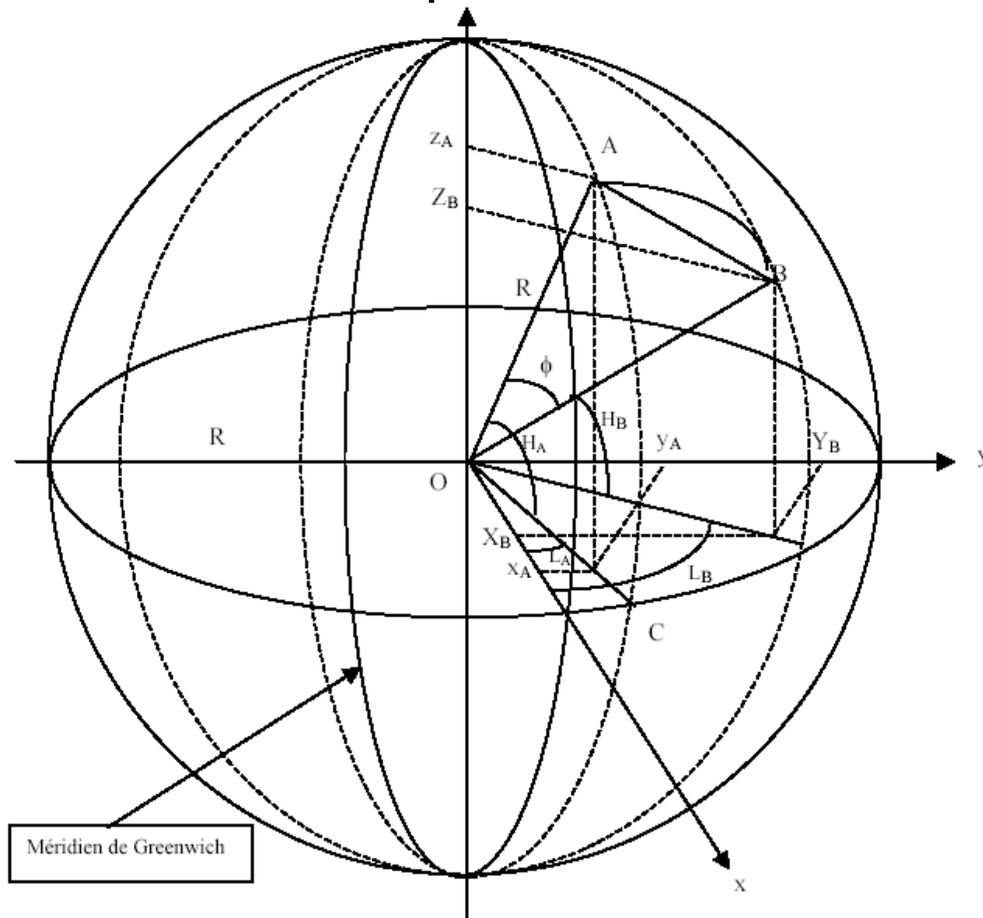
Autre exemple

La mappemonde est un modèle de la terre

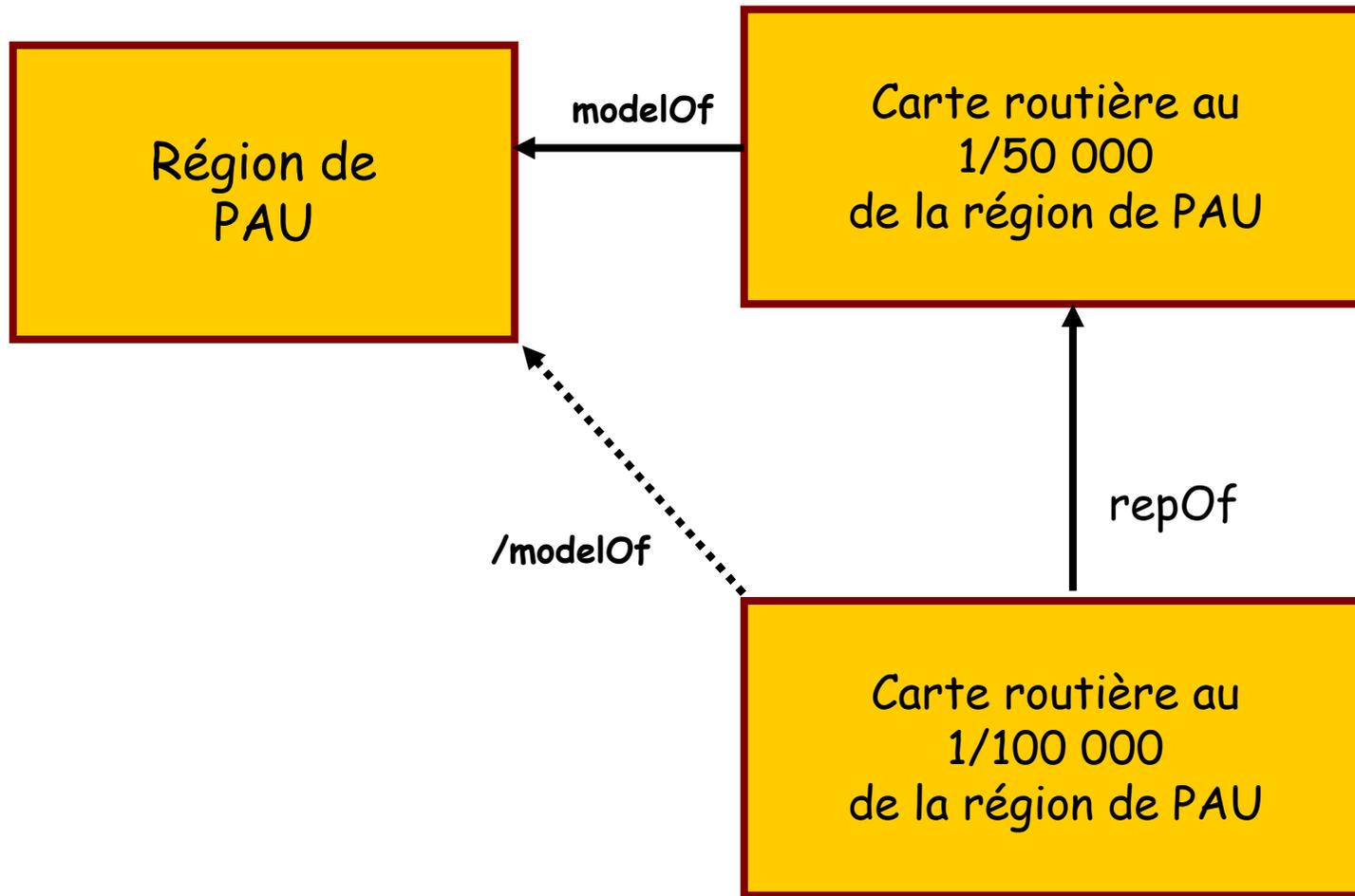


Autre exemple (suite)

- Permettant de poser certaines questions ...
- mais pas d'autres.



Un modèle peut représenter un modèle

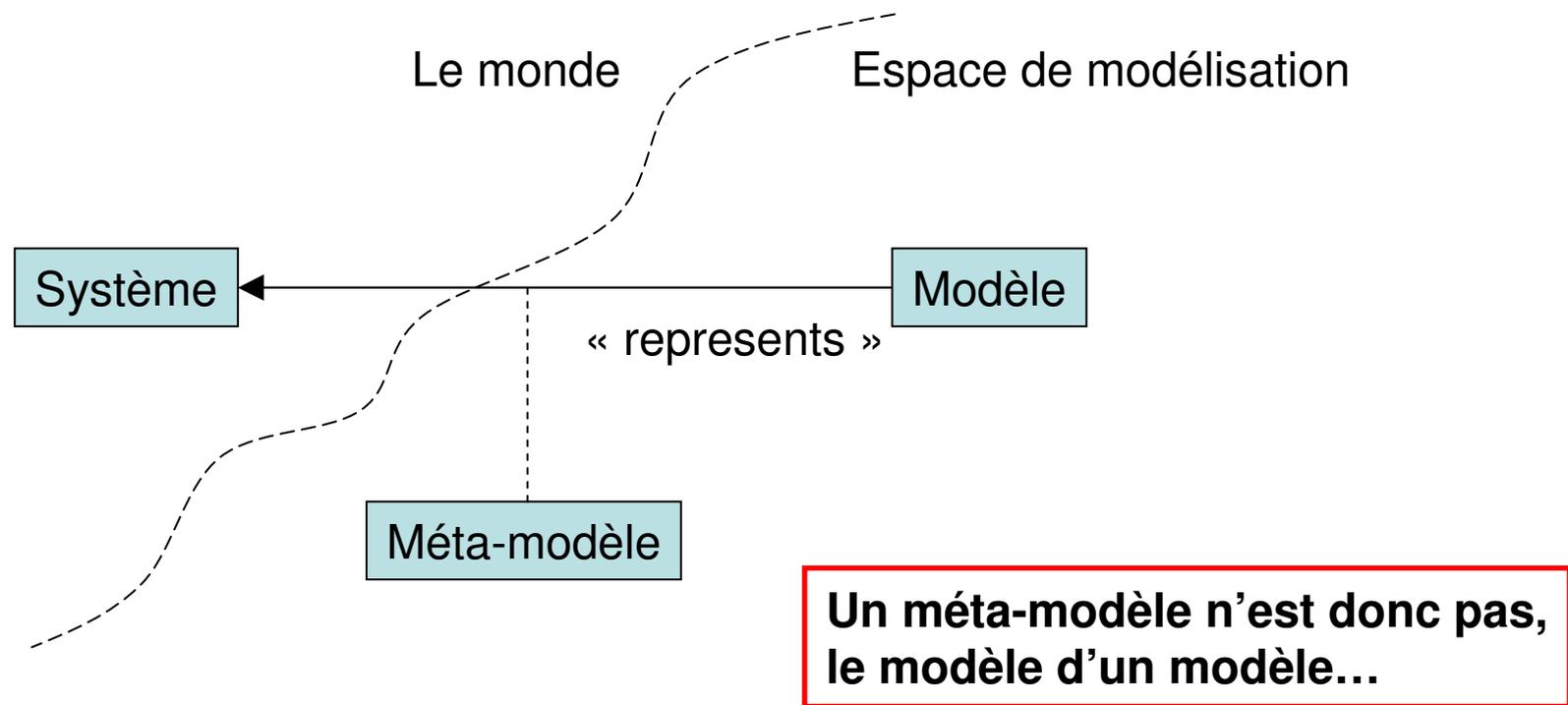


Méta-modèle

- L'approche MDA est basée sur les notions de modèle et de méta-modèle
- Le préfixe « méta » exprime l'auto-référence
- Méta-*
 - Méta-physique : la physique de la physique
 - Méta-classe : la classe des classes
 - `java.lang.Class` en Java
 - Méta-table : la table des tables
 - `user_tables` pour Oracle
 - Méta-modèle : le modèle d'un modèle ?

Modèle vs Méta-modèle

- Un méta-modèle définit la relation entre un modèle et un système

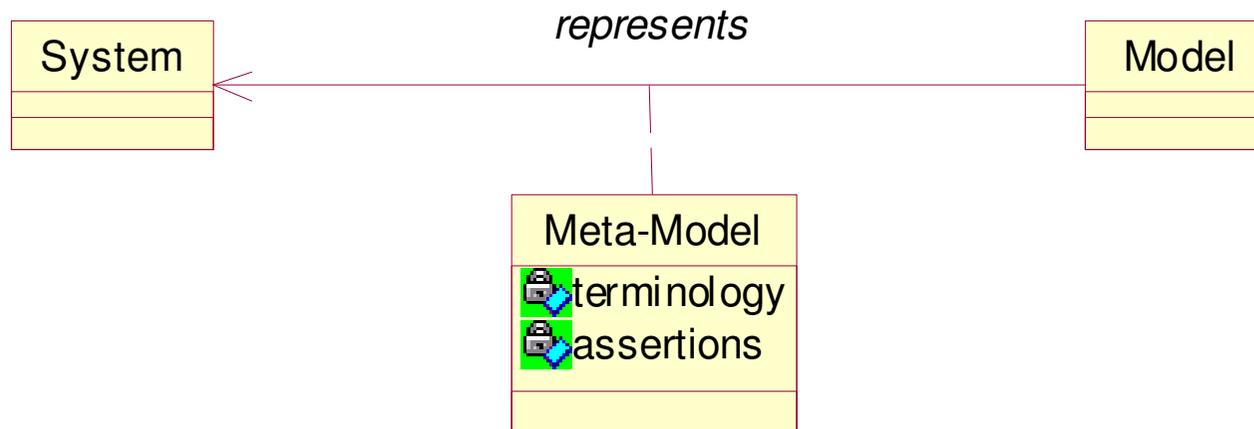
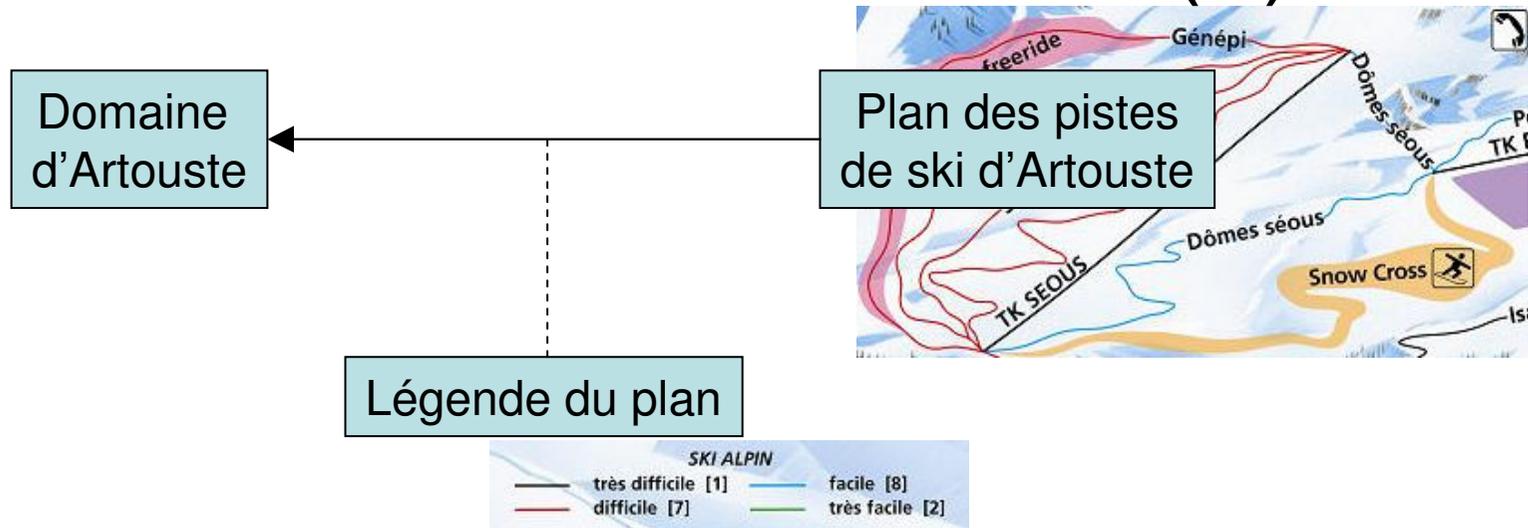


Modèle vs méta-modèle (2)

PIC DU MIDI D'OSSAU
2885



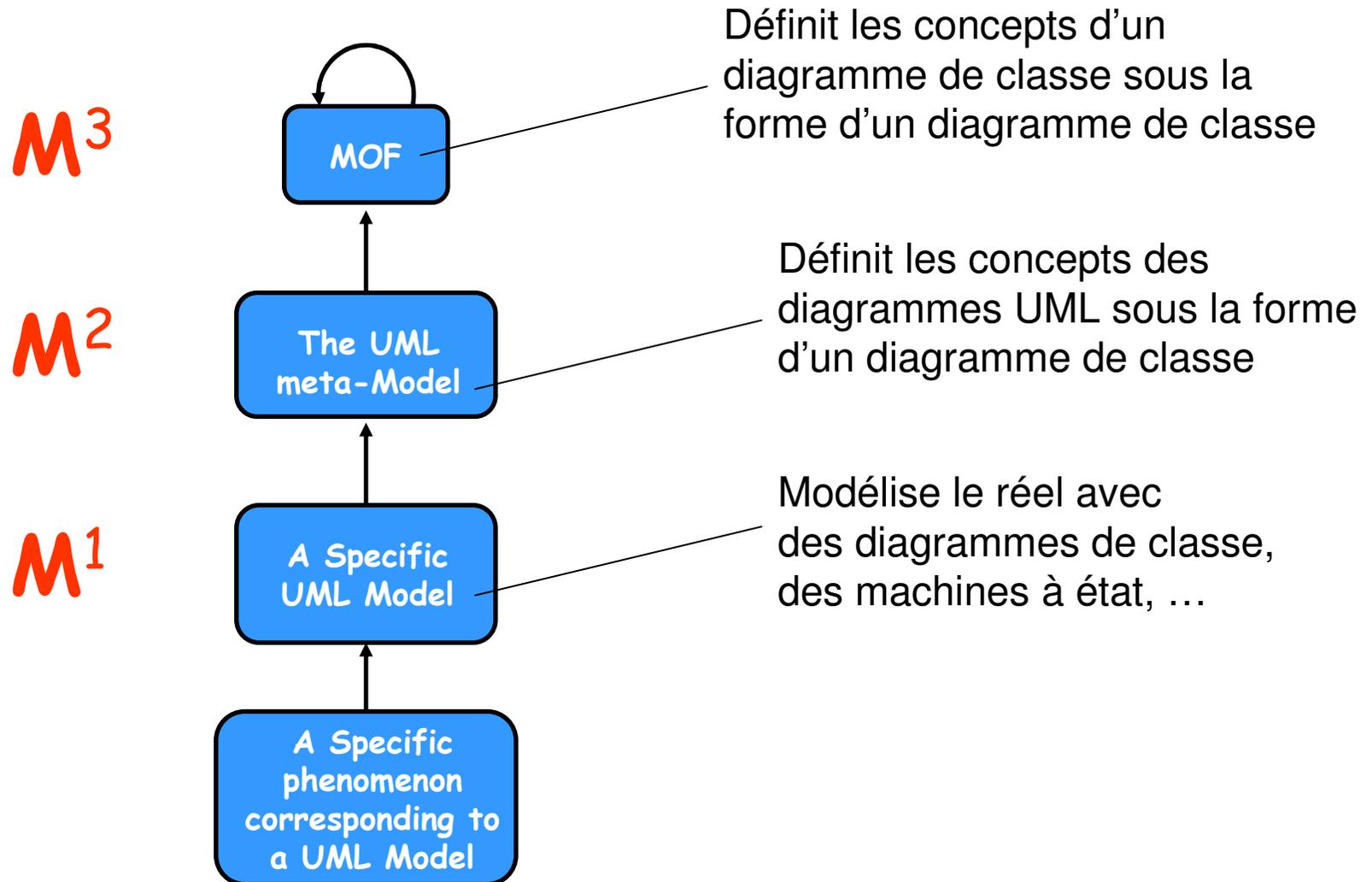
Modèle vs méta-modèle (3)



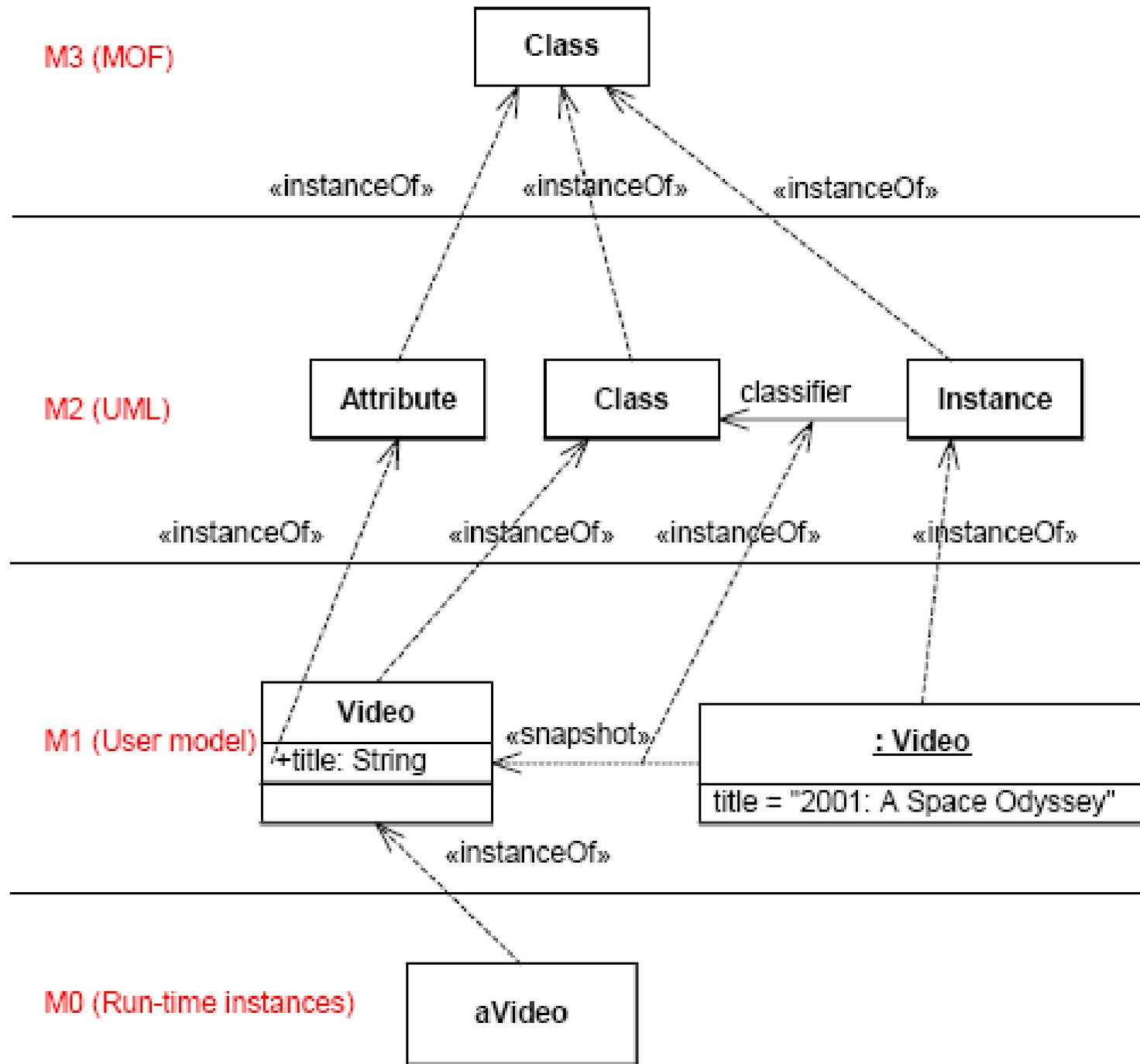
Méta-méta-méta-...

- Comment peut-on définir un méta-modèle ?
 - À l'aide d'un méta-méta-modèle !
- Et comment définir ce méta-méta-modèle ?
 - À l'aide d'un méta-méta-méta-modèle !
- Et comment définir ce méta-méta-méta-... ?
 - À l'aide d'un méta-méta-méta-...
- Le problème paraît sans fin ...

Niveaux de modélisation dans UML



Les 4 niveaux de modélisation dans UML (exemple)



Niveaux de modèles : comparaison

