



IDM

Ingénierie Dirigée par les Modèles

Olfa Dâassi



IDM?



« *One and three chairs* », Joseph Kosuth, 1965



Evolution des technologies

- Evolution permanente des technologies logicielles
- Exemple : systèmes distribués
 - Faire communiquer et interagir des éléments distants
- Evolution dans ce domaine
- C et sockets TCP/UDP
- C et RPC
- C++ et CORBA
- Java et RMI
- Java et EJB
- C# et Web Services
- A suivre ...



Evolution des technologies

- Idée afin de limiter le nombre de technologies
 - Normaliser un standard qui sera utilisé par tous

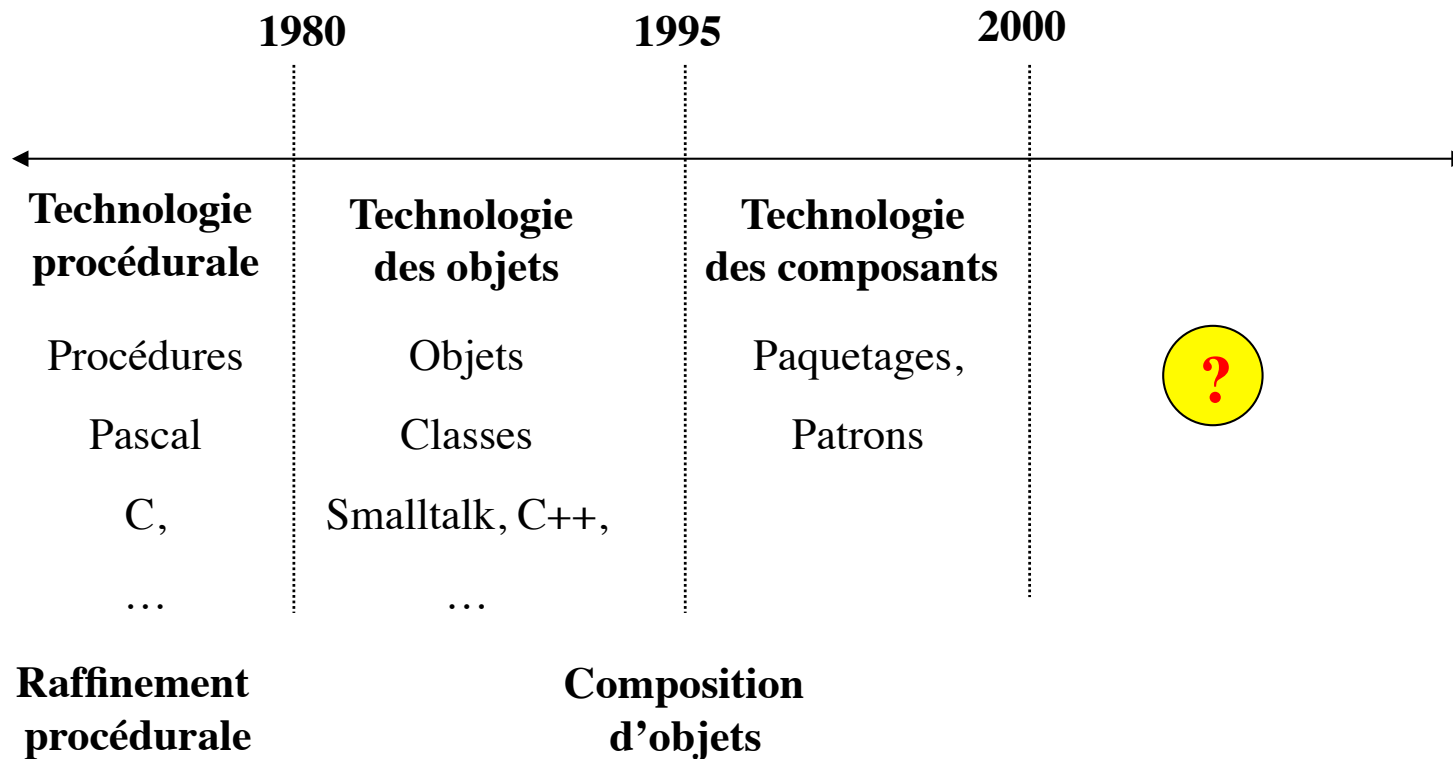


Evolution des technologies

- Evolutions apportent un gain réel
- Paradigmes de programmation
 - C : procédural
 - Java, C++, C# : objet
 - Encapsulation, réutilisation, héritage, spécialisation ...
 - EJB, CCM (Corba Component Model): composants
 - Meilleure encapsulation et réutilisation, déploiement ...



Evolution des technologies





Evolution des technologies

- Conclusion sur l'évolution des technologies
 - Nouveaux paradigmes, nouvelles techniques
 - Pour développement toujours plus rapide, plus efficace
 - Rend difficile la standardisation (désuétude rapide d'une technologie)
 - Et aussi car combats pour imposer sa technologie
- Principes de cette évolution
 - Evolution sans fin
 - La meilleure technologie est ... celle à venir



Evolution des technologies

- Quelles conséquences en pratique de cette évolution permanente ?
 - Si veut profiter des nouvelles technologies et de leurs avantages
 - Nécessite d'adapter une application à ces technologies
- Quel est le coût de cette adaptation ?
 - Généralement très élevé
 - Doit réécrire presque entièrement l'application
 - Car mélange et du code métier et du code technique
 - Aucune capitalisation de la logique et des règles métiers



Evolution des technologies

- Exemple : Application de calculs scientifiques distribués sur un réseau de machines
 - Passage de C/RPC à Java/EJB
 - Impossibilité de reprendre le code existant
 - Paradigme procédural à objet/composant
- Pourtant
 - Les algorithmes de distribution des calculs et de répartition des charges sur les machines sont indépendants de la technologie de mise en oeuvre
 - Logique métier indépendante de la technologie

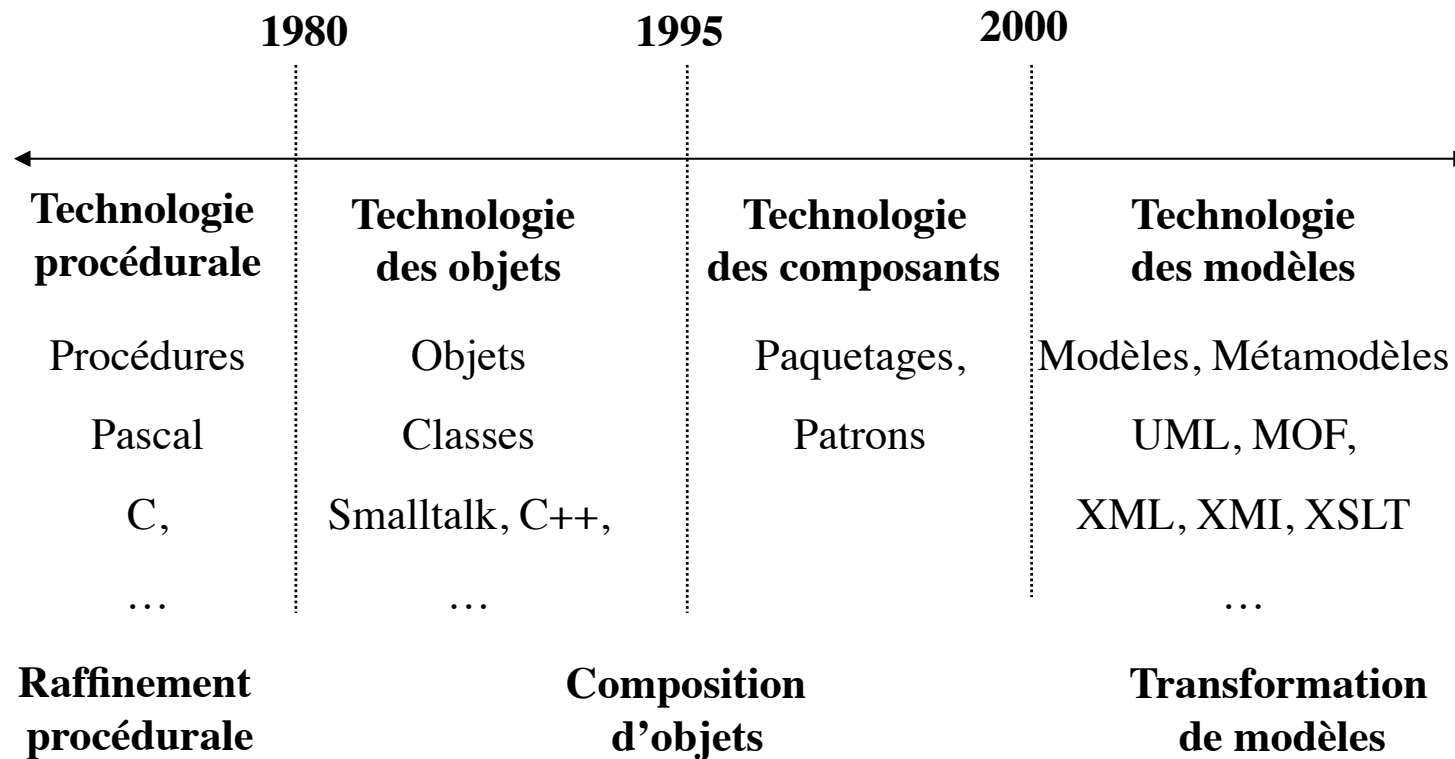


IDM

- « *Modéliser est le futur, et je pense que les sociétés qui travaillent dans ce domaine ont raison* » B. Gates
- « *Obtenir du code à partir d'un modèle stable est une capacité qui s'inscrit dans la durée* » R. Soley
- « A quoi sert bon modéliser puisque *in fine* il faudra toujours écrire du code? »
- « Un bon schéma vaut mieux qu'un long discours ... sauf qu'à un schéma (UML) correspond plus d'un long discours ! »



IDM





IDM: Principes

- Modèle
- Méta-modèle
- Transformation de modèles

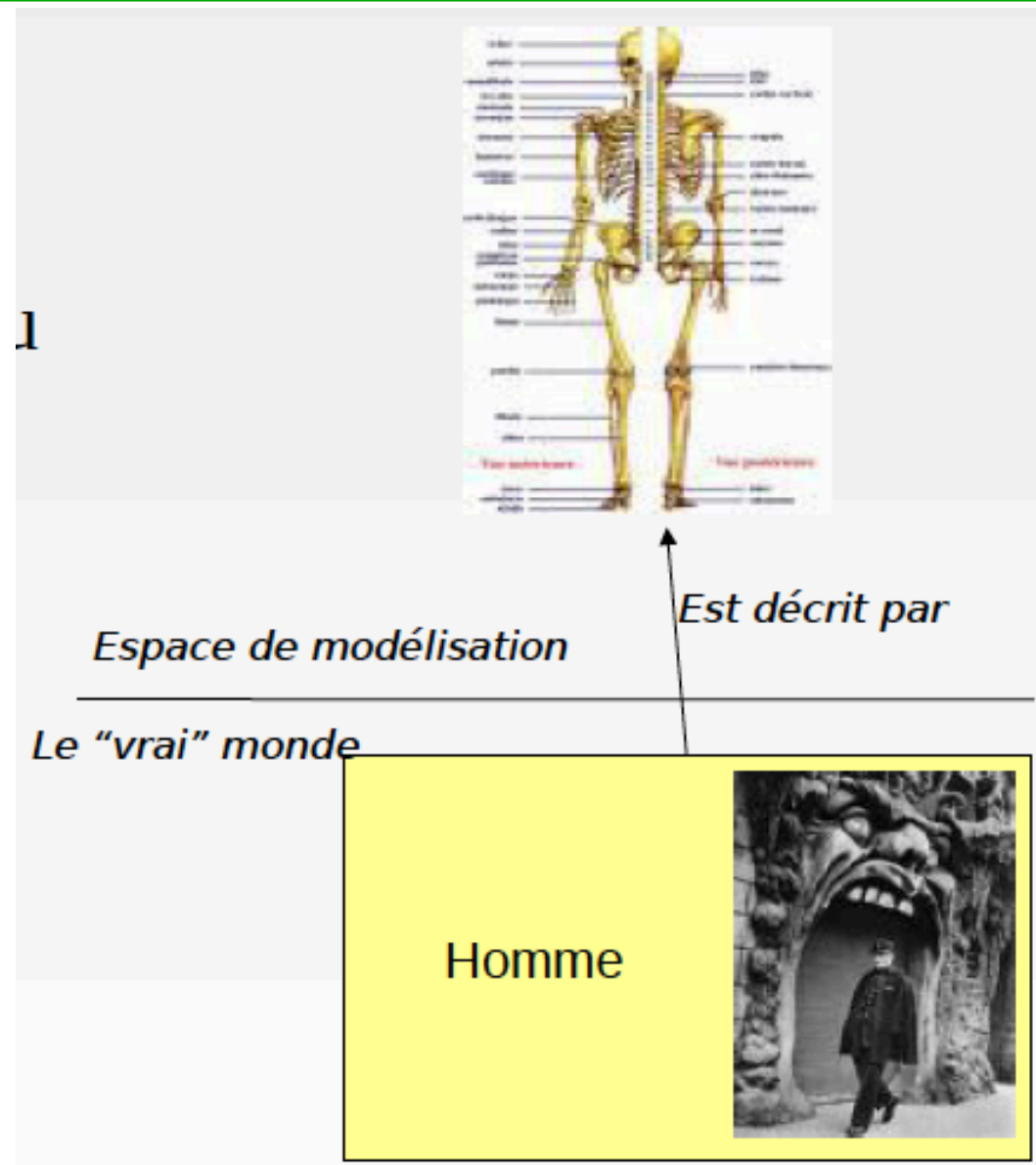


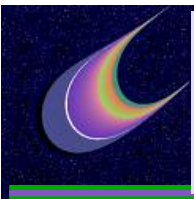
Modèle

- Un modèle est une description, une spécification partielle d'un système
 - Abstraction de ce qui est intéressant pour un contexte et dans un but donné
 - Vue subjective et simplifiée d'un système
- But d'un modèle
 - Faciliter la compréhension d'un système
 - Simuler le fonctionnement d'un système
- Exemples
 - Modèle économique,
 - Modèle démographique ...

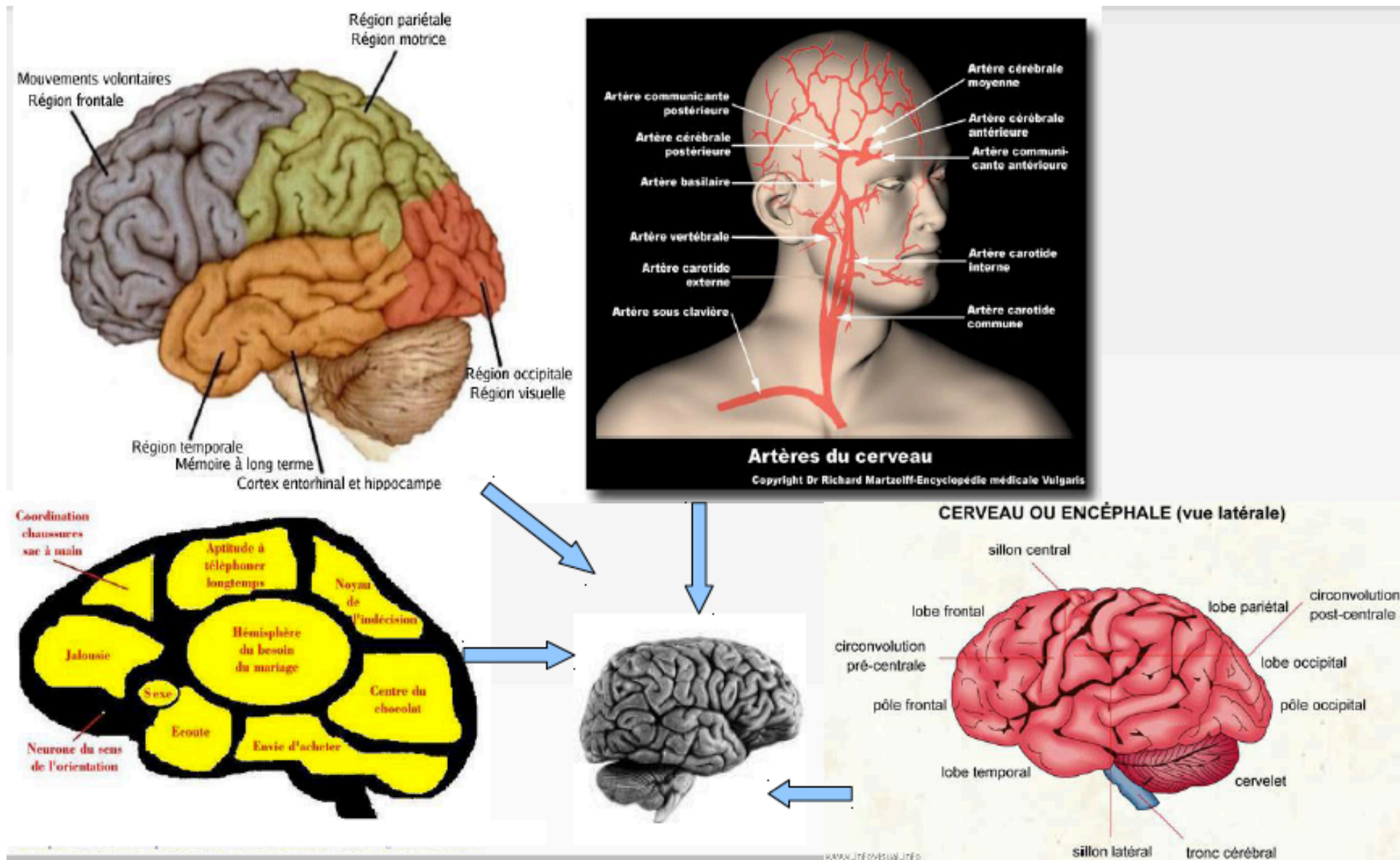


Modèle





Modèle : C'est un point de vue



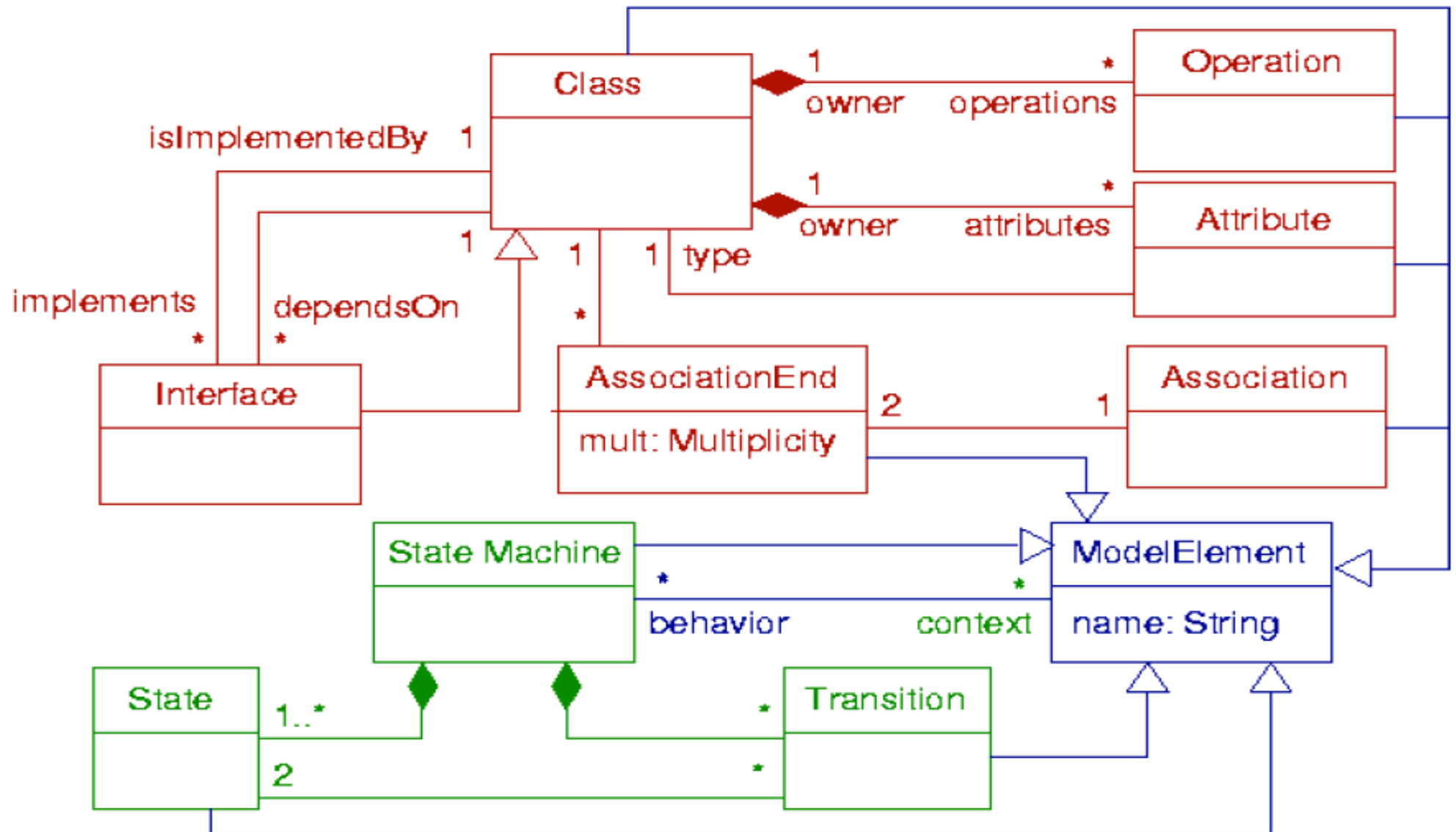


Méta-modèle

- Est une description des éléments du modèle
- La relation entre modèle et méta-modèle est une relation de conformité
- Chaque élément du modèle
 - Est une « instance » d'un élément du méta-modèle (d'un métaélément)
 - En respectant les contraintes définies dans le méta-modèle
 - Un texte écrit est conforme à une orthographe et une grammaire
 - Un programme Java est conforme à la syntaxe et la grammaire du langage Java
 - Un fichier XML est conforme à sa DTD
 - Un modèle UML est conforme au méta-modèle UML



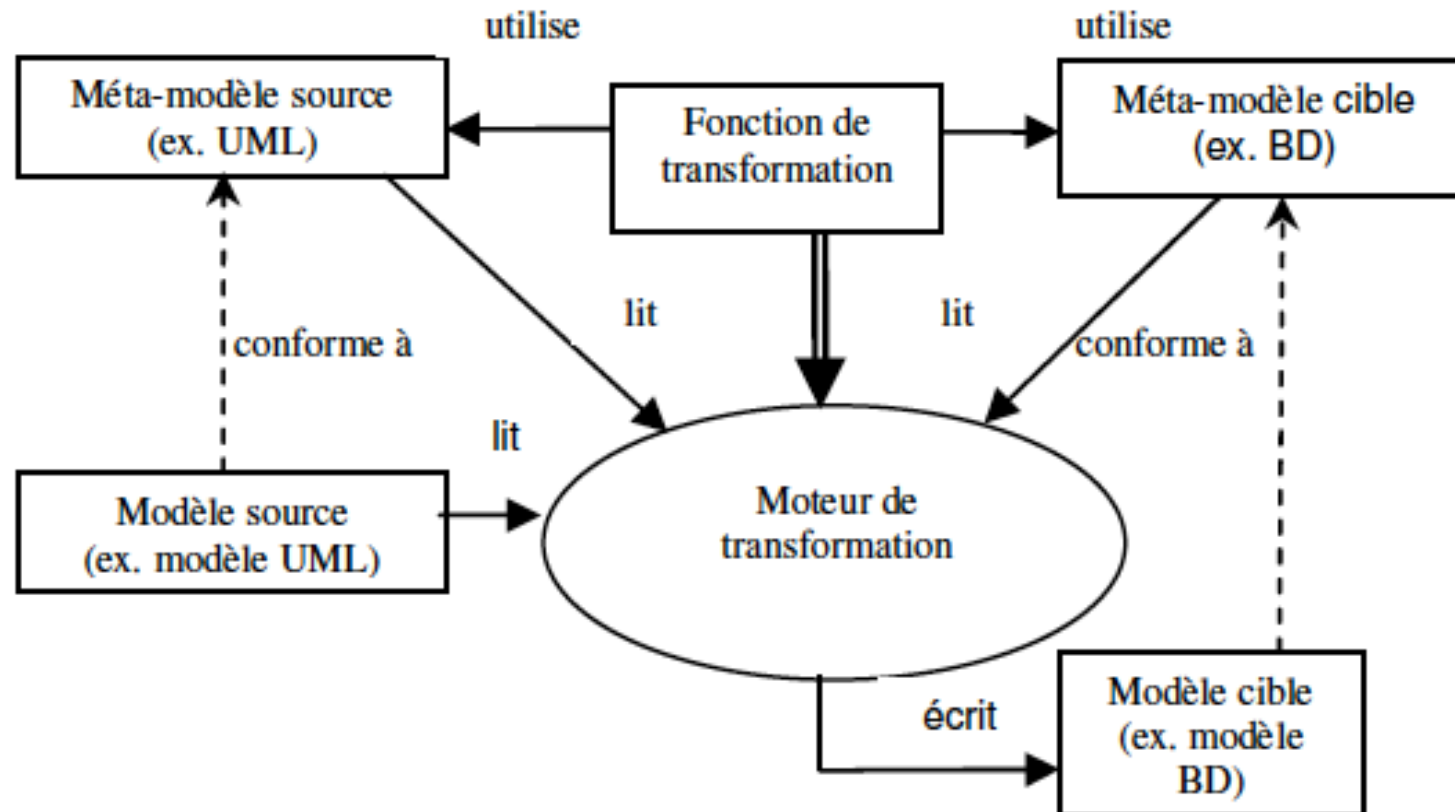
Méta-modèle : en rouge celui des diagrammes de classes, en vert celui des





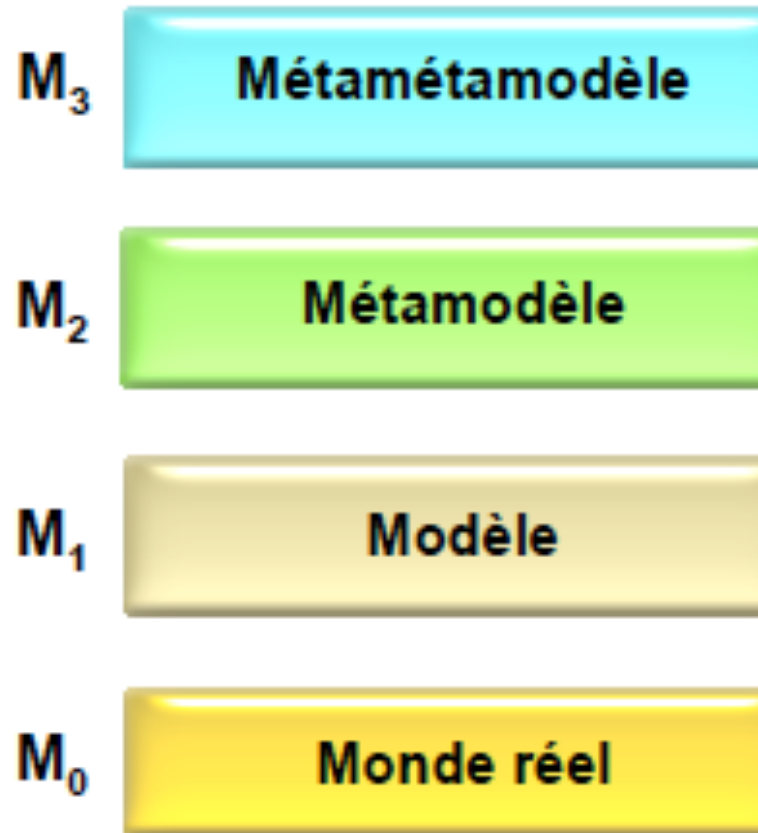
Transformation de modèles

- Une transformation est la génération automatique d'un modèle cible à partir d'un modèle source conformément à une définition de transformation



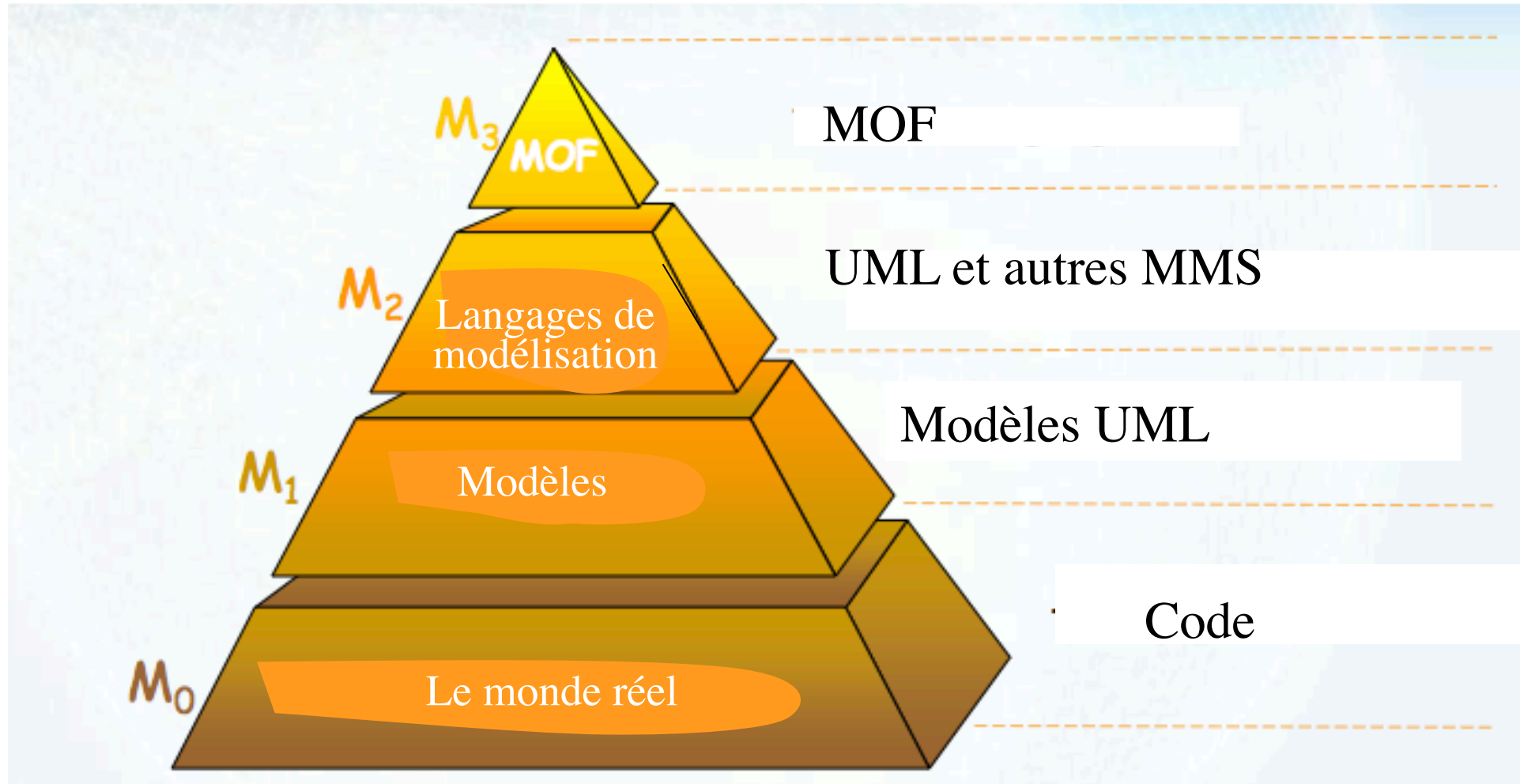


IDM : Les 4 niveaux d'abstraction





MDA: Architecture à quatre niveaux





Niveaux d'abstraction de l'IDM

M3 : Méta-méta-modèle
des méta-modèles de
M2... et de M3

M3 (MOF)

M2 : Méta-modèles des
modèles de M1

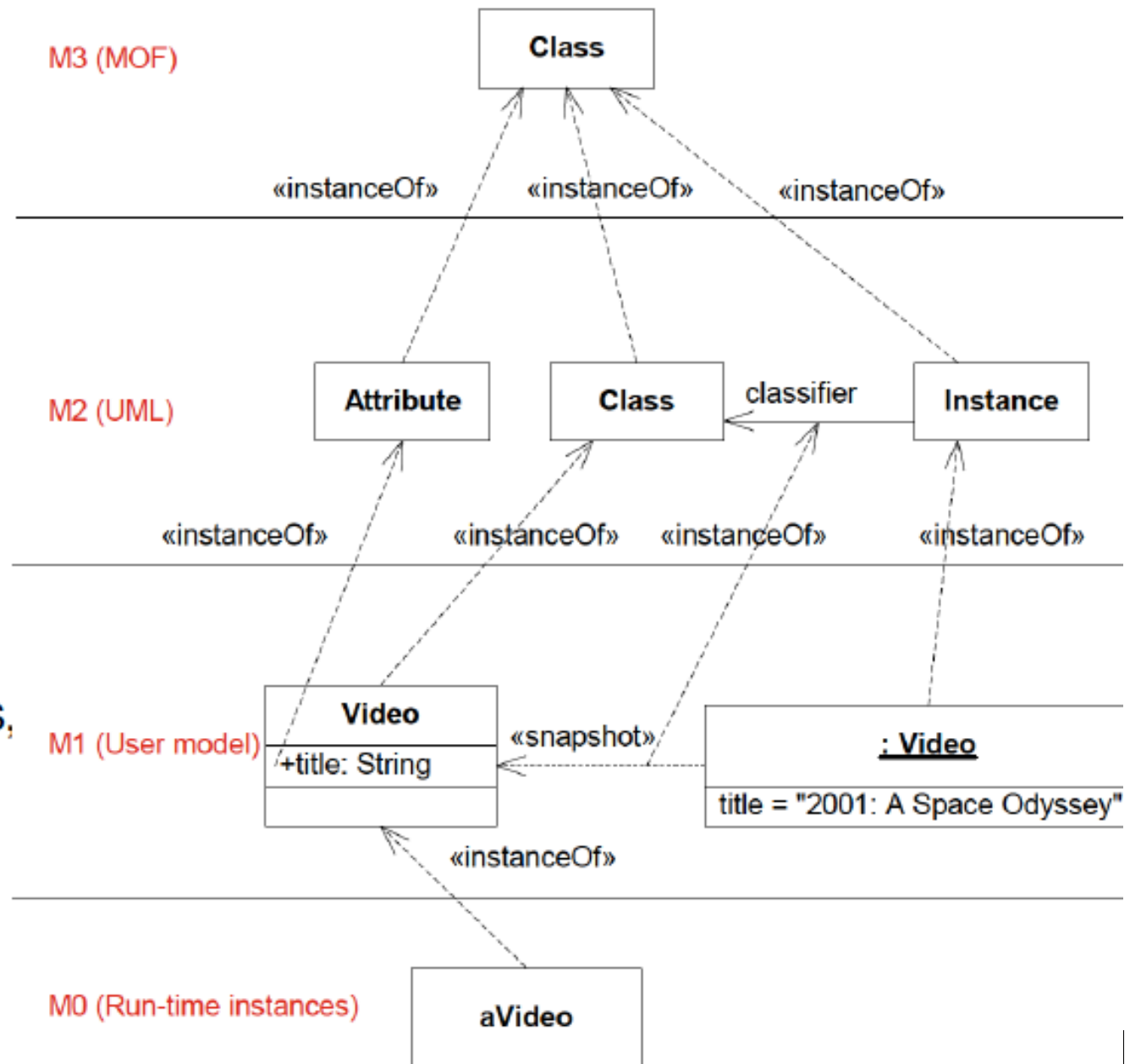
M2 (UML)

M1 : Modèles
(Diagrammes de classes,
de séquence, ...)

M1 (User model)

M0 : Instances des
modèles à l'exécution

M0 (Run-time instances)





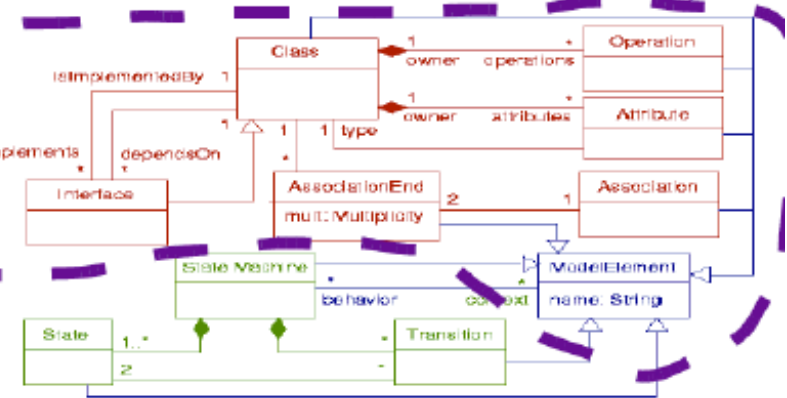
Niveau M3

conforme à



Niveau M2

conforme à



Niveau M1

conforme à

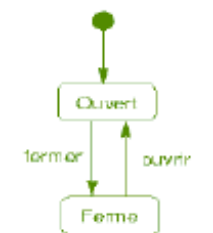
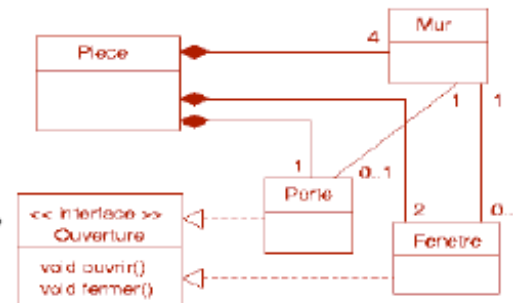
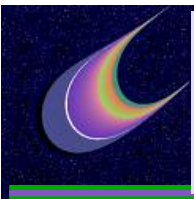


Diagramme d'état associé à l'interface Ouverture

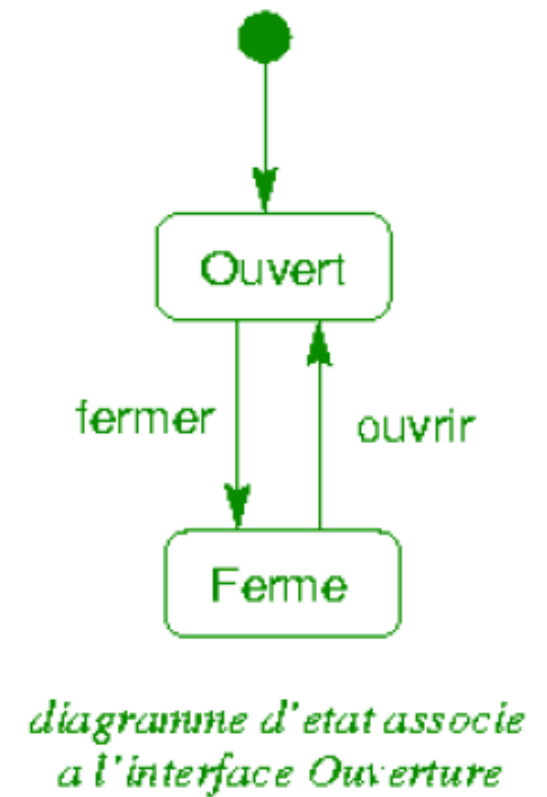
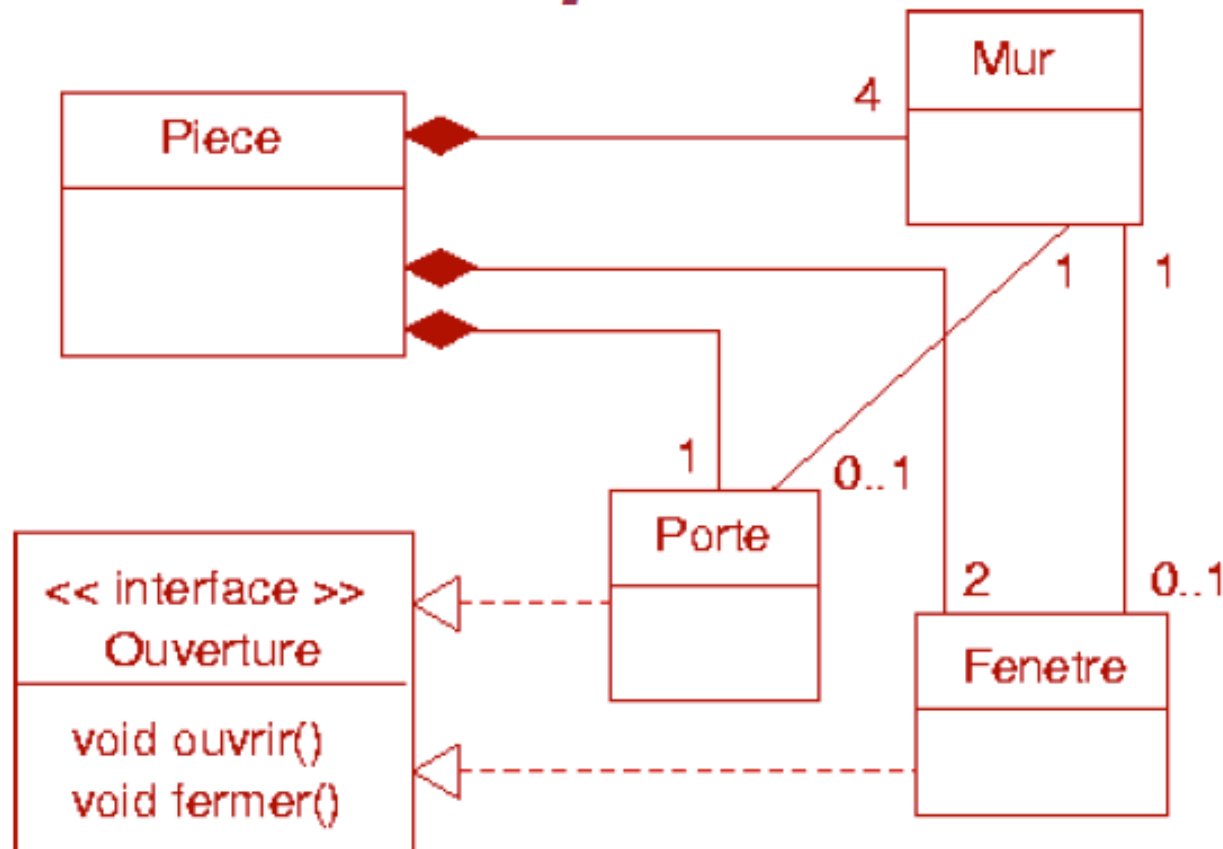
Niveau M0

conforme à



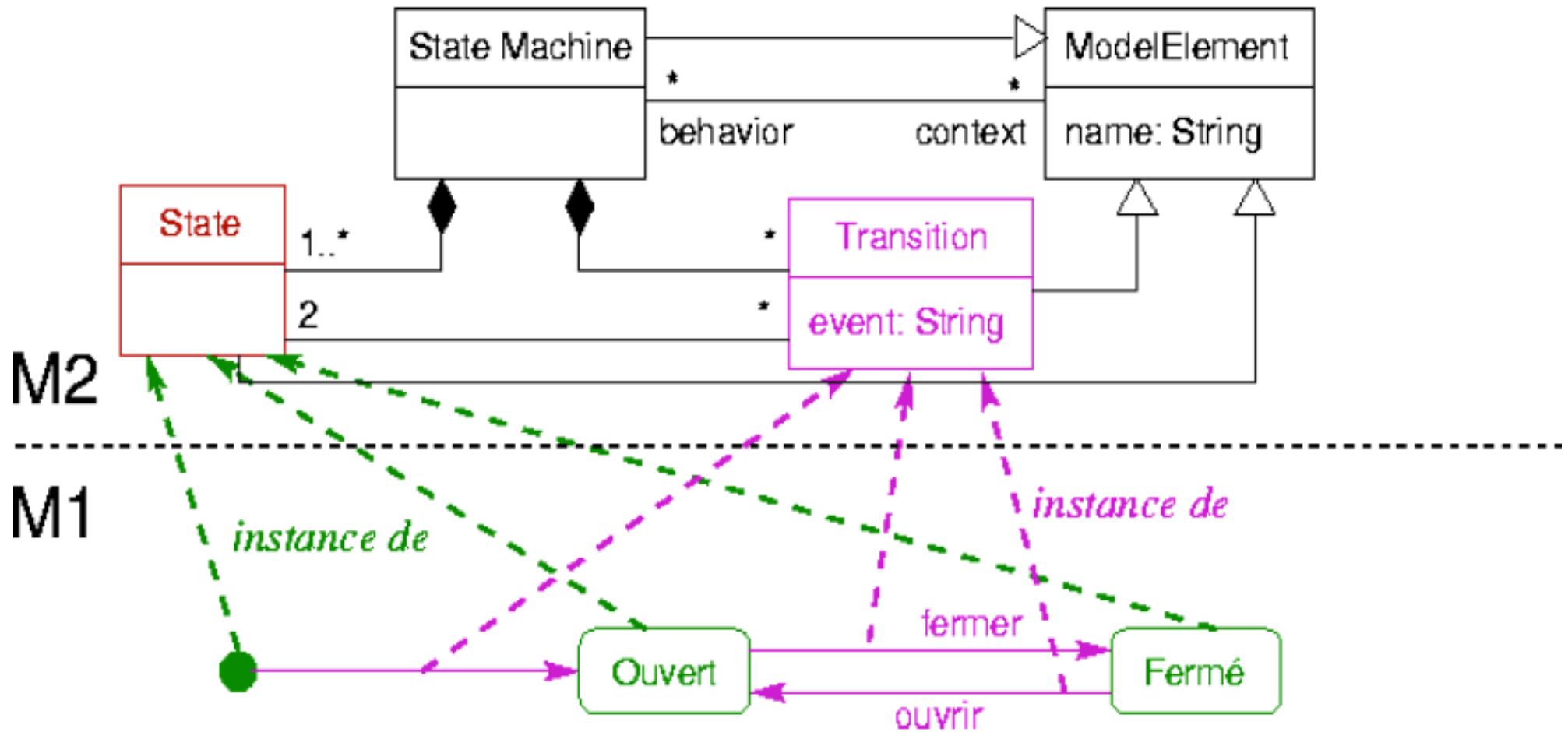


Exemple : M1





Conformité





Méta-modèle des BD relationnelles

