



Université Ibn Zohr
Faculté des sciences – Département Informatique

IDM: Ingénierie Dirigée par les Modèles Ou MDE: Model-Driven Engineering

Ayoub SABRAOUI

Master SID BIG-DATA

Année universitaire 2019/2020

Plan

- ✓ **Principes et Définitions**
- ✓ **Promesses & Enjeux**
- ✓ **MDA et ses standards**

MDE: Définition

- **L'Ingénierie Dirigée par les Modèles (IDM)** est une méthodologie/vision de développement de logiciels qui met l'accent sur la création de **modèles** (ou **l'abstraction**) plus proche du concept du domaine concerné plutôt qu'orienté informatique ou algorithmique.
- Un paradigme de modélisation pour IDM est considéré comme efficace si ses modèles ont un sens par rapport au point de vue de l'utilisateur et peut servir de base pour l'implémentation de systèmes.

3

Les Principes de l'IDM

- **P1:** “ Models as first class entities “
- **P2:** Montée en abstraction
- **P3:** Séparation des préoccupations (separation of concerns)
- **P4:** Modèle productifs (Vs. Contemplatifs)

4

P1: « Models as first class entities »

■ Passage du « tout objet » au tout « modèle »

- c-à-d. Modéliser au lieu de coder
- Changement d'état d'esprit:
De ~~Write Once Run Anywhere~~ à Model Once, Generate Anywhere

■ Le modèle objet a atteint ses limites

- Les concepts objets sont plus proches de l'implémentation/technologie que du domaine métier
- « Objects have failed » by Richard. P. Gabriel, Opening OOPSLA' 02
- Difficile de communiquer, raisonner avec du code.
 - *Car les aspects fonctionnels/non-fonctionnels sont noyés dans les lignes de code*

- **Le défi** est donc de s'orienter vers une expertise métier et avoir un bon langage de modélisation.

5

Modèle: Définition

Définition: “A **Model** represents reality for the given purpose; the model is an **abstraction** of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality”

« The nature of modeling », 1989, Par Jeff Rothenberg

Traduction: *un modèle représente une réalité pour un sujet donné. Le modèle est une abstraction de la réalité dans le sens où il ne peut pas représenter tous les aspects de la réalité. Cela nous permet de traiter le problème de manière simplifiée en évitant la complexité, les dangers et l'irréversibilité de la réalité.*

6

Modélisation: Définition

Définition: “**Modeling**, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose“

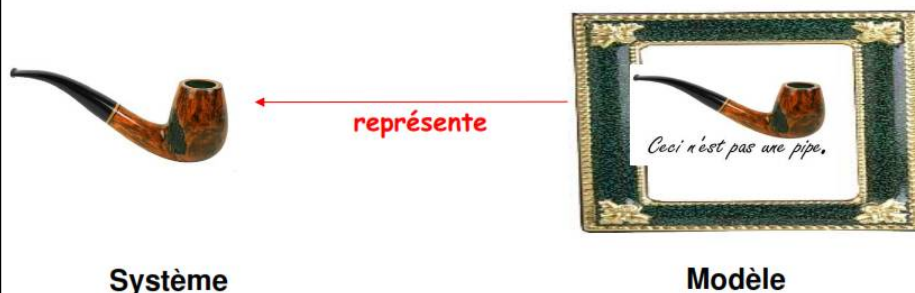
« The nature of modeling », 1989, Par Jeff Rothenberg

Traduction: *au sens large, c'est l'utilisation rentable de quelque chose à la place d'une autre dans un but cognitif. Elle nous permet d'utiliser quelque chose qui est plus simple, plus sûre ou moins chère que de la réalité.*

- Attention au débat: abstraction – simplification ?
 - La modélisation simplifie la compréhension et la communication autour du problème, elle ne simplifie pas le problème lui même
- Attention, un modèle n'est pas forcément graphique

7

Modèle: Exemple



8

Méta-Modèle: Définition

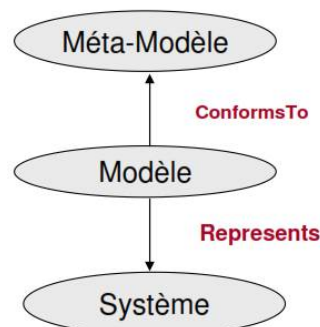
- Pour être productif, les modèles doivent être utilisés par des machines
 ➔ on a donc besoin d'un langage précis pour les définir = **Méta-Modèle**
- A « *metamodel is a model that defines the language for expressing a model* » [OMG].
- Pour Seidewitz, a « *metamodel makes statements about what can be expressed in the valid models of a certain modeling language* ». *Traduction : un méta-modèle fait des déclarations sur ce qui peut être exprimé en modèles valides de certains langages de modélisation.*

9

Méta-Modèle: Définition

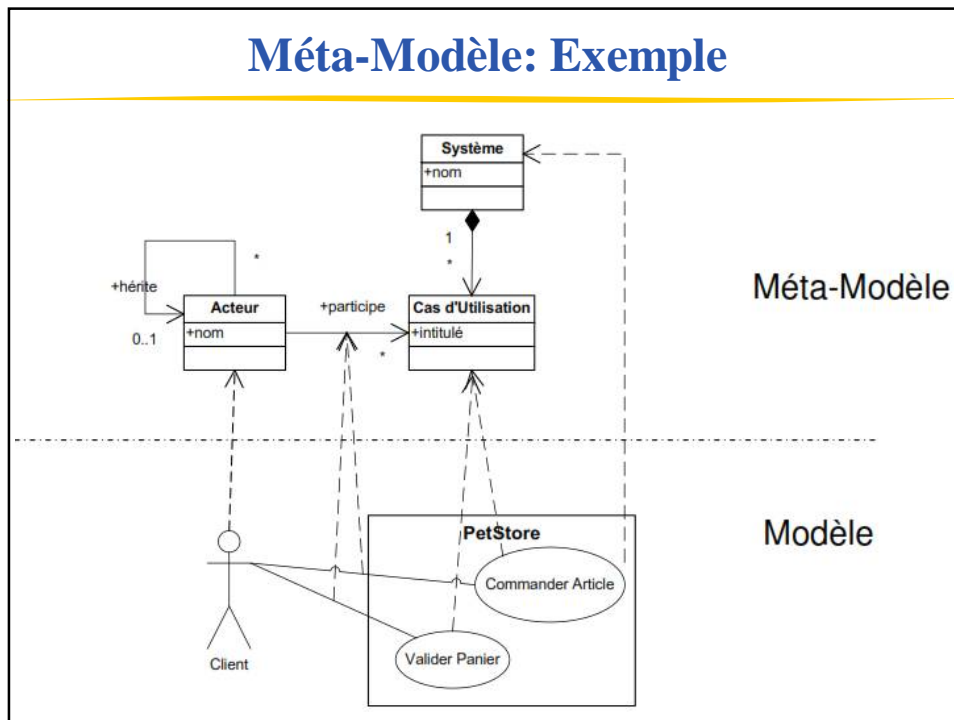
La relation entre un modèle et son méta-modèle est une relation "**conforme à**".

Un modèle est conforme à son méta-modèle si les éléments et les relations entre ces éléments sont définis dans le méta modèle



10

Méta-Modèle: Exemple

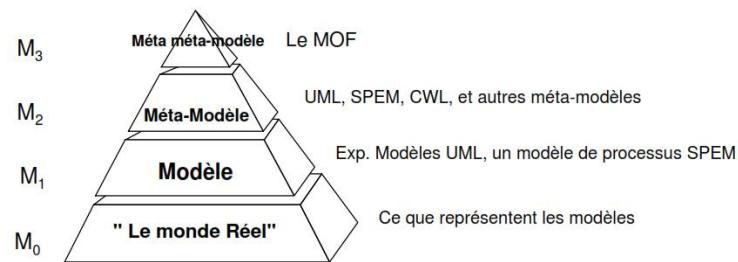


Méta Méta-Modèle: Définition

- A « *metametamodel* is a model that defines the language for building metamodels » [OMG].
- À l'OMG, MOF (Meta-Object Facility) représente le langage pour la spécification de nouveaux méta-modèles (langages). Exp. UML, SPEM, OCL..
- Le MOF est conforme à lui-même, auto descriptif.

Architecture à 4 niveaux: MOF

- La spécification de MOF (Meta-Object Facilities) adopte une architecture de méta-modélisations à quatre couches :



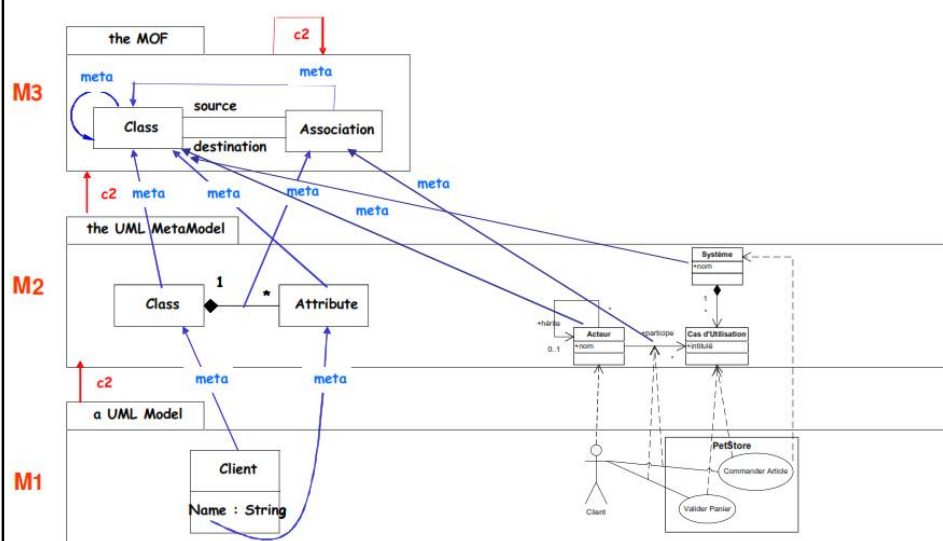
M3 : Un niveau qui comporte le méta-méta-modèle décrivant le langage de méta-modélisation.

M2 : Dans ce niveau on trouve le méta-modèle décrivant le langage de modélisation.

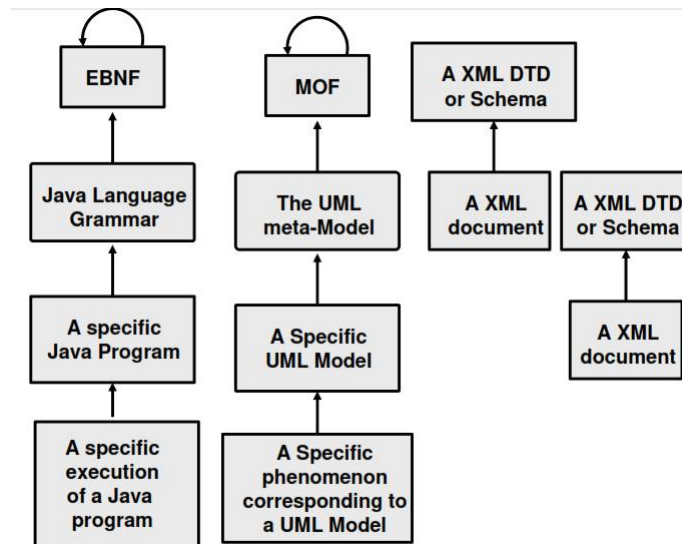
M1 : Ce niveau est composé du modèle représentant le système réel à modéliser.

M0 : Ce niveau représente le système réel à modéliser.

Méta Méta-Modèle: Exemple



Méta Méta-Modèle: autres systèmes



15

P2: Abstraction

Abstraction c'est:

- Ignorer les détails insignifiants
- Ressortir des détails les plus importants
 - Important: c'est décider de ce qui est signifiant et de ce qui ne l'est pas, dépend directement de l'utilisation visée du modèle

Abstraction: « Opération intellectuelle qui consiste à isoler par la pensée l'un des caractères de quelque chose et à le considérer indépendamment des autres caractères de l'objet »

[Source Larousse en ligne](#)

Les modèles permettent de s'abstraire de certains détails qui ne sont pas indispensables pour comprendre le système selon un point de vue donné.

16

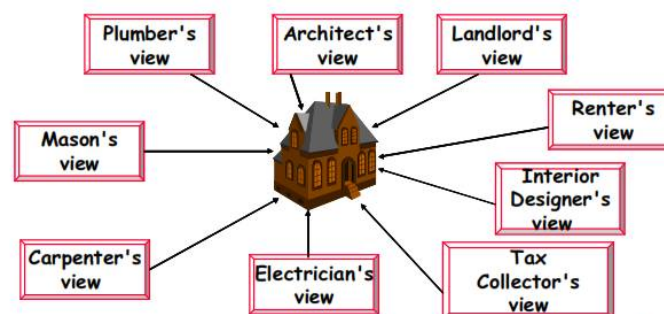
Abstraction

- **Histoire du GL:** l'amélioration de la productivité s'est toujours faite en augmentant le niveau d'abstraction
- Selon Caper's Jones Software Productivity research (SPR 2006), les langages de 3^{ème} G ont amélioré la productivité du développeur par **450%** (C, Fortran, COBOL, etc.)
 - L'introduction des langages OO n'ont pas fait autant
 - Java **20%** plus productif que le Basic
- Les langages de prog. arrivent aujourd'hui à leur limite en terme d'abstraction
- **Défis:**
 - **Préoccupation Majeure: Assurer la transformation/transition vers le niveau le plus bas**
 - Choisir le bon niveau d'abstraction

17

P3: Séparation des préoccupations

- La séparation des préoccupations est indispensable dans le cas d'applications complexes.
 - Plusieurs aspects/points de vue signifient que nous avons plusieurs métiers
Exp. : sécurité, communication, GUI, QoS,...
 - Un point de vue → un modèle



Source: Bézivin

Séparation des préoccupations: Une Vue, un Langage

- Chaque vue peut être exprimée en utilisant un langage de modélisation différent
- Il y a du coup plusieurs intervenants dans la boucle
- **Le défis de l'IDM est :**
 - Intégration des vues
 - Assurer la cohérence entre ces vues

19

Abstraction et séparation des préoccupations Exemple: Google Maps

- Différents niveaux d'abstractions
- Différents points de vues



P4: Modèles Productifs

- Modéliser reste un investissement qu'il faudra rentabiliser
- Vers de plus en plus de code généré automatiquement à partir des modèles
 - 100% dans certains domaines: web, scripts de config., réseaux
 - Plus difficile dans le cas d'applications complexes
- Beaucoup d'expertise dans les générateurs de code
 - Moins d'erreurs dans le code généré
 - Gains en temps et en qualité

21

Modèles Productifs

Les défis des modèles productifs consistent en:

- Des langages de modélisation expressifs (proche du métier) et précis
- Des générateurs de code fiables
- Intégration du code généré à partir de plusieurs vues vers un seul système!
- Génération du comportement?
- Définition de la chaîne de transformation (raffinement) modèles vers code
 - Complicé dans le cas où plusieurs langages de modélisation sont utilisés en même temps pour modéliser le même système

22

Plan

- ✓ **Principes et Définitions**
- ✓ **Promesses & Enjeux**
- ✓ **MDA et ses standards**

23

MDE: Promesses

Gérer la complexité

- Les applications sont de plus en plus énormes
- Exemple:
 - **Windows 2000**: ~40 millions de lignes de code
 - **Google**: 500 000 Serveurs, une capacité de calcul gigantesque
 - Répondre à plus d'1 milliard de requêtes par jour
 - Chacune interrogeant 8 milliards de pages web en moins d' 1/5 S.

➔ Impossible à gérer par un seul programmeur

24

MDE: Promesses

Meilleure communication

- Le code n'est pas toujours compréhensible par les développeurs qui ne l'ont pas écrit
 - On peut difficilement communiquer avec du code, **pas assez abstrait!**
- Dans de gros projets souvent:
 - Des centaines de personnes
 - Un peu partout dans le monde
- L'apparition de nouvelles façons de travailler: (L'outsourcing, sous-traitance, etc.)
 - ➔ La communication y apparaît comme très importante.

25

MDE: Promesses

Pérenniser un savoir-faire

- Certains projets peuvent durer des années
 - Pas toujours les mêmes personnes qui travaillent sur le projet
 - Besoin de capitaliser un savoir faire indépendant du code et des technologies
 - Capturer le métier sans se soucier des détails techniques
- Exemples de projets:
 - Contrôleur aérien (Thalès): Projet ~8 ans, durée de vie 40 ans
 - La construction d'un avion (Airbus): Projet ~10 ans, durée de vie 50 ans

26

MDE: Promesses

Augmenter la productivité

- Génération de code à partir des modèles
- Maîtrise de la variabilité
 - **La vision ligne de produit**
 - Un modèle générique pour un produit avec plusieurs variantes
- Exemple: Nokia
 - Qui a vendu 1,1 milliard de téléphones portables
 - Avec des milliers de versions de logiciels
 - Alors que le délai de mise sur le marché n'est que de 3 mois.

27

Mais quel langage utiliser pour modéliser, métamodéliser (créer de nouveaux langages)?

Plusieurs approches existent

- **L'approche MDA, utilisation d'un ensemble de standards fournis par l'OMG i.e. MOF, UML**
 - UML pour la modélisation des applications OO
 - Définir un nouveau langage si UML ne suffit pas
 - Soit en partant de rien en utilisant MOF
 - Soit par extension d'un langage existant exp. UML, SPEM
 - Soit à l'aide du mécanisme de profile UML
- **L'approche DSML (Domain-Specific Modeling Languages)**
 - Des solutions propriétaires, exp. EMF, KM3, Kermata, MetaEdit+
 - Propres à des domaines, au métier de l'entreprise
 - **Objectif principal:** se rapprocher du domaine métier, générer 100% du code