

# Ingénierie des Modèles

(1)

## Généralité sur GL

GL : - Automatisation de processus

- Méthodes, techniques et outils pour la production et la maintenance des composants logiciels de qualité.

Bases Théoriques + Méthodes et outils validés par la pratique




**Fabriquer des SI complexes**

obj Optimiser le coût de développement et de maintenance des logiciels

## Principes d'IDM

- **Intégration** faire communiquer 2 objets ou plus afin de réaliser un objectif spécifique
- **Interopérabilité** • Capacité d'échanger les services et données indépendamment de l'environnement
  - faire communiquer 2 objets de l'espace : CORBA

2 formes d'intégration  de le temps

principes d'IDM :

① Capitalisation

- Approche objet / Composant la réutilisation des éléments logiciels / code
- Approche MDE Model Driven Engineering réutilisation du logiciel métier

② ~~logique~~ Abstraction

- permet d'adapter une logique métier à un contexte
- Modélisation d'une manière indep de la technologie



## ⑤ Modelisation

l'usage de modèles qui changent

MDE : djs

↳ Passage d'une vision **Contemplative** de modèles  
(documentation, spécification, comon)

↳ d'une vision **Productive**  
(Générer le code final du logiciel pour une technologie donnée)

Ce passage a besoin de :

- Les modèles doivent être bien définis  
(Langage et MetaModèles)
- Savoir manipuler et interpréter les modèles via des outils
- Séparation des préoccupations  
(2 préoccupations : Matière et plateforme de mise en œuvre)
- Projection et fusion de modèles  
Conception orientée Aspect

## Modèles et Meta modèles

Modèle : • Description, spécification formelle d'une fonction ou une structure du comportement d'un système

• Entité Representation abstraite d'une entité du monde réel, afin de la décrire

but : • Faciliter la compréhension d'un système  
• Simuler le fonctionnement

• Un système donné peut avoir plusieurs modèles  
• chaque modèle représente un aspect donné du système

• Spécification  
⇓  
d'un système à construire

VS description  
↓  
d'un système existant



un modèle est écrit ds un langage

- ↳ Non ou peu formel : Langue naturelle, dessin, texte...
- ↳ formel et bien définis : suit une syntaxe, grammaire sémantique  
⇒ Metamodelle

**Modèle est conforme à sa metamodelle**

- Metamodelle est une entité de 1<sup>er</sup> classe
  - Metamodelle est un modèle qui définit un langage pour définir un modèle
  - Metamodelle :  $\Sigma$  règles et concepts d'un modèle
- MetaModelle  $\neq$  Langage.

### Synthèse

- Modèle représentation d'un système
- modèle est écrit ds un langage unique conforme à sa propre metamodelle
- metamodelle s'écrit ds le langage unique de son unique meta metamodelle
- Modèle pt avoir une représentation graphique

Espaces Technologiques ET

- ↳ correspond à un Contexte du travail
- ↳  $\Sigma$  concepts, méthodes, outils

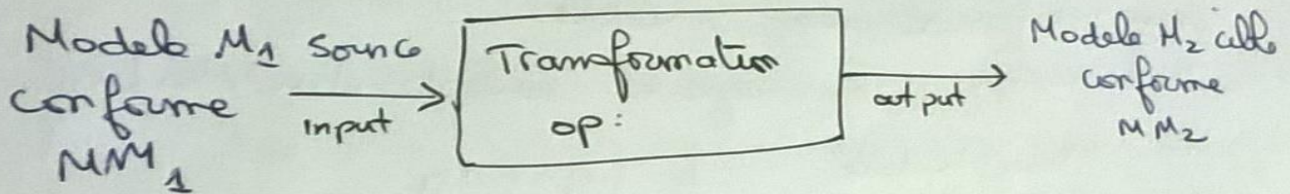
ET associé à une certaine communauté possédant une expertise commune, des pratiques communes et des problématiques partagées

- les technologies sont généralement structurées par des sous-familles
- MOF : Meta Object Facility : Modèle qui permet de réaliser des MM.



- JavaXML : Prog Java écrit sous XML
- ↳ classes et méthodes sont des éléments XML
  - ↳ structure est représentée par imbrication des éléments
- XMI : XML Model Interchange
- tissage : séparation : utilisation de l'information dont on a besoin

## Transformation des Modèles :



### → Transformation Endogène

- Dans un  $\hat{m}$  ET.  
 $M_1$  et  $M_2$  sont conforme à une seule MM. (une seule MM en jeu)

### → Transformation Exogène

- Entre 2 ET différents  
Des MM différents

- Op Monadiques  
les opérations portent sur un seul modèle  
"affichage, sérialisation, stockage de modèle"
- Op dyadiques  
les opérations portent sur plusieurs modèles  
"comparaison, transformation, fusion ..."



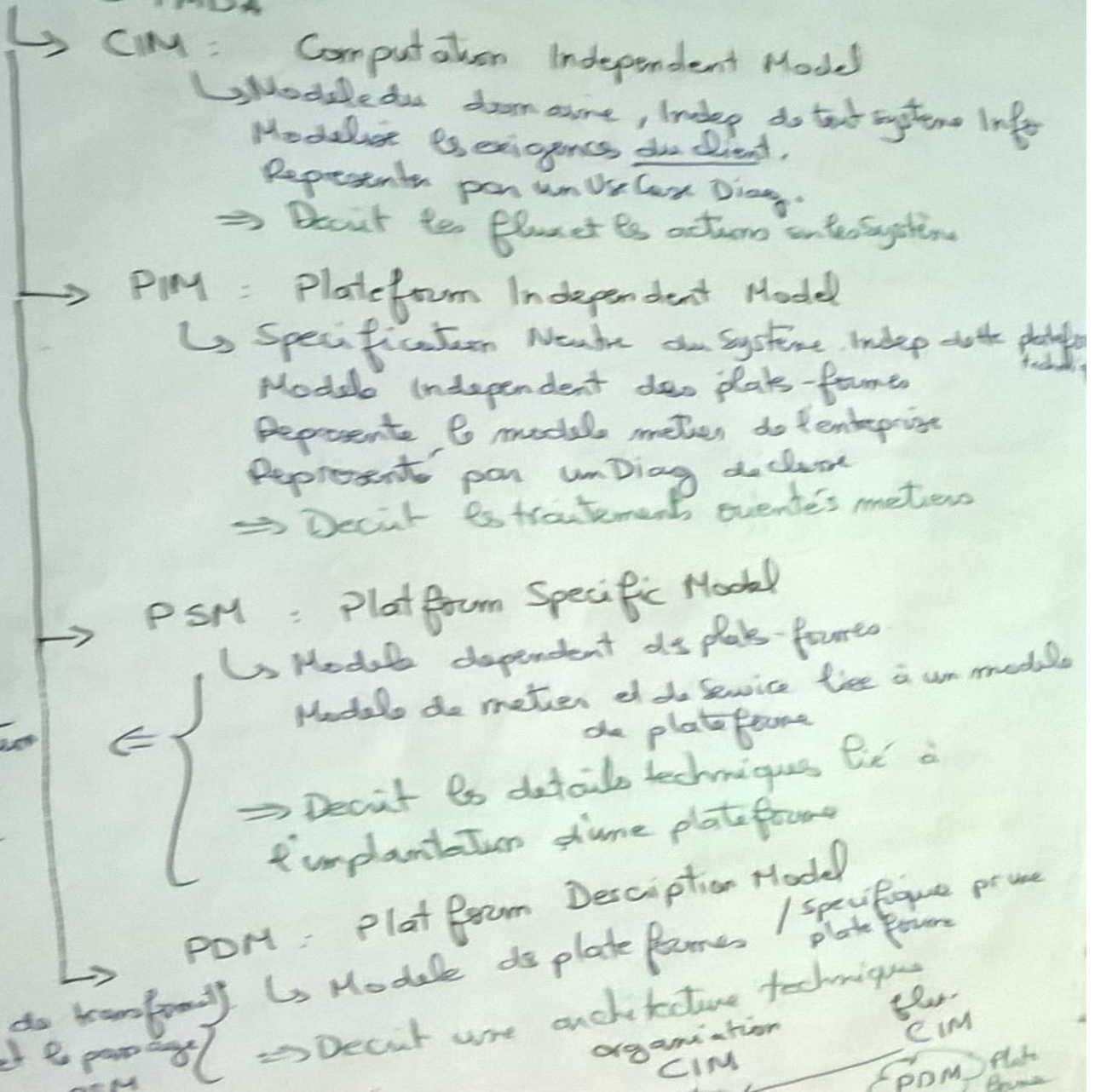
# Approche MDA

## Modelo Driven Architecture

Obj :

- Abstraire les parties metiers de leur mise en œuvre
- Basé sur des technologies et des standards de l'OMG (UML, OCL...)

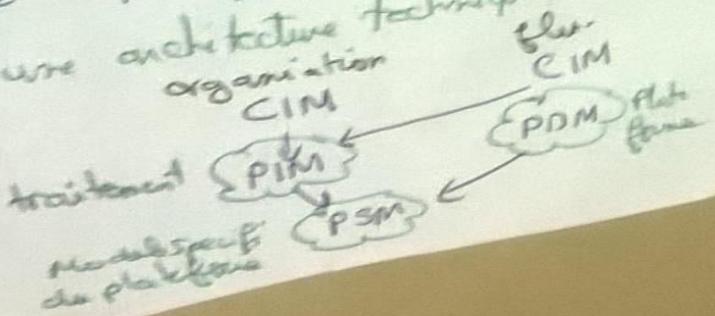
### Modelos de iMDA



Generation du Code

modelos de transform. qui permet le passage PIM -> PSM

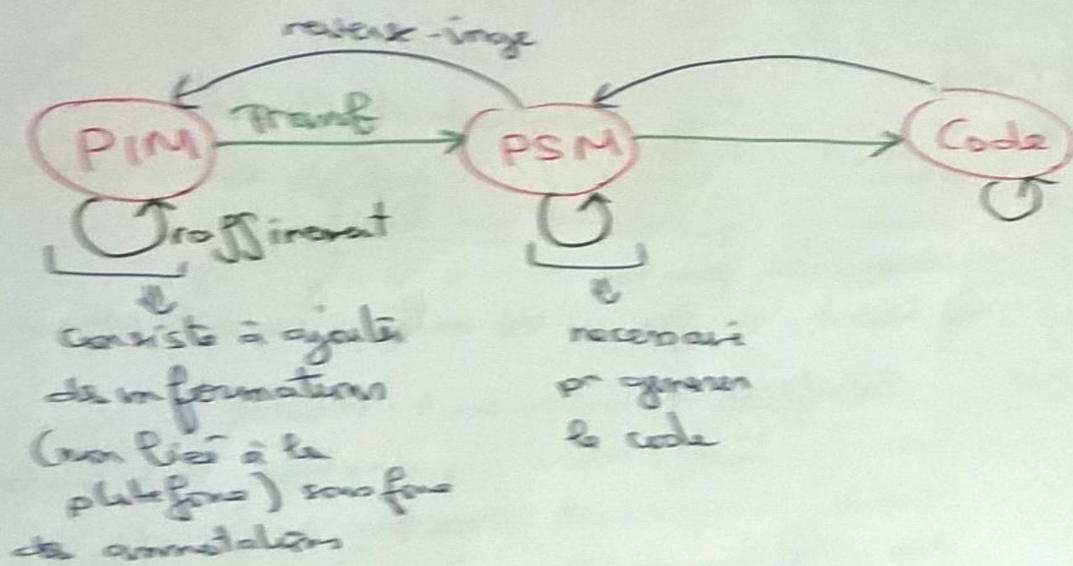
Analyse : flux et organisation  
 Analyse : traitement  
 Conception



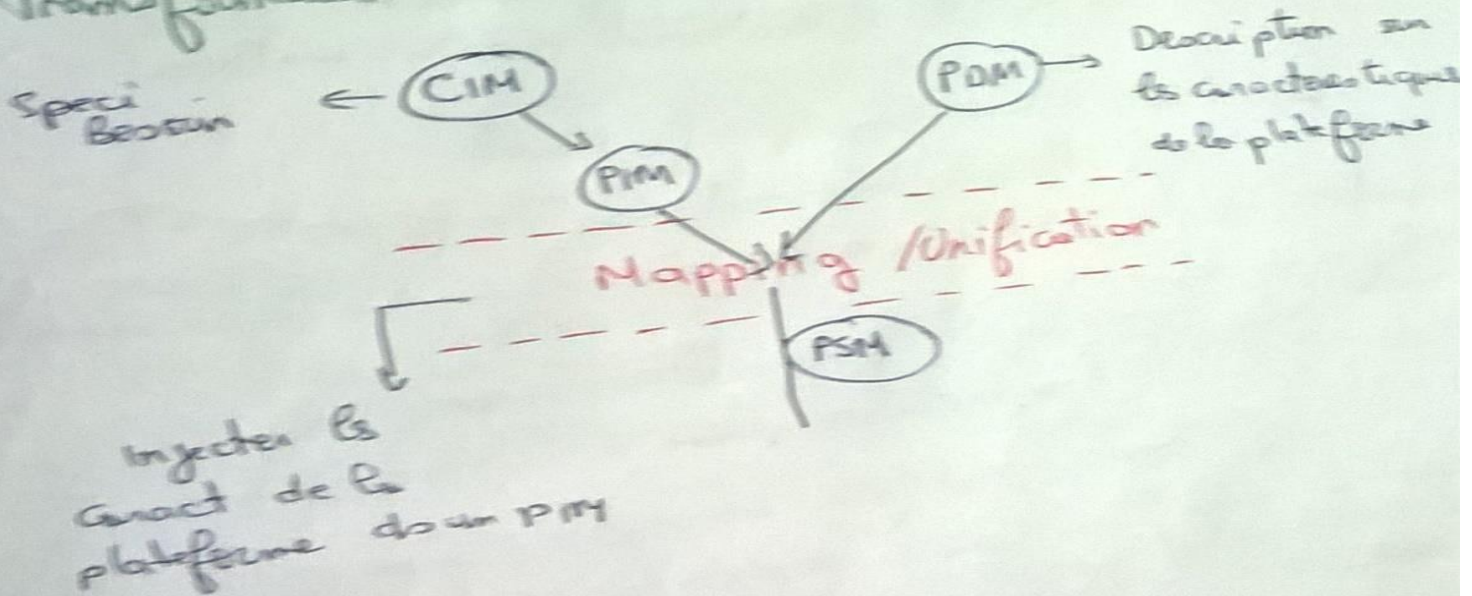


o les exigences Modelisées dans CIM sont prise en compte de la construction du PIM.

Transformation des Modèles de MDA

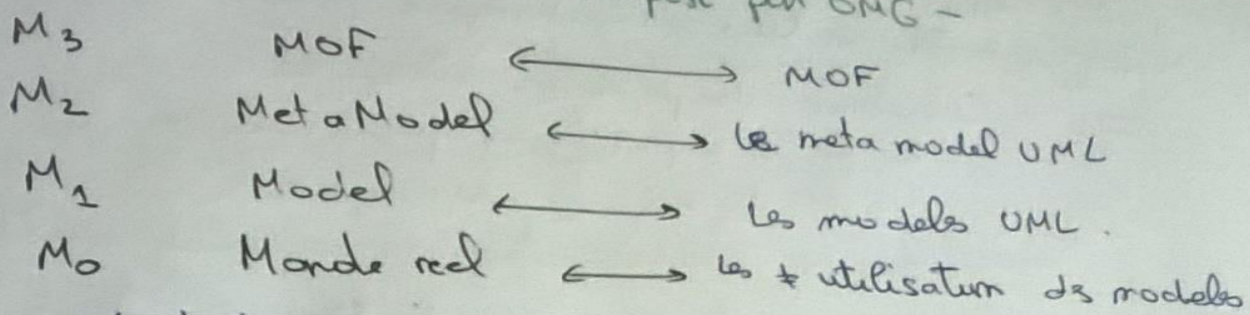


Transformation PIM → PSM Cycle en Y

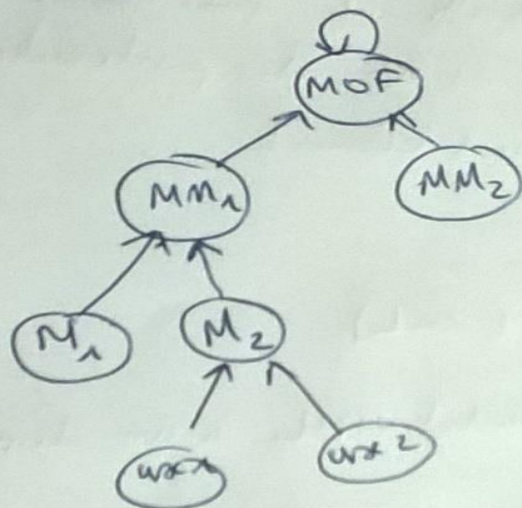




Architecture à 4 niveaux - Proposée par OMG -



Architecture proposée par OMG mais valable pour les autres langages.



Meta Models ⇒ Packages  
 Meta Entities ⇒ classes

Spécification de meta modèle:

Définir un type de modèle avec tous ces types d'éléments et leurs contraintes:

- 3 cas possibles
- ① Définir un méta modèle nouveau à partir de rien
  - ② Modifier un méta modèle existant
  - ③ Spécifier un méta modèle existant (ajout des éléments et des contraintes sans enlever) ⇒ Profil UML.

Profil:

Spécification d'un MM UML  
 - Ajout de nouveaux types d'éléments, contraintes sur éléments ou relations, aucune suppression -



Profil : mécanisme d'extension d'UML pour l'adapter à un contexte métier ou technique particulier.  
⇒ Création de profils adaptés  
↳ pr une plateforme particulière  
↳ pr des aspects de temps réel  
↳ pr modélisation de processus métier.

Stereotype : extension, spécification (spécialisation d'un élément du méta modèle « Interface »)

Tagged Value : Marquage des attributs d'une classe pour préciser une contrainte ou un rôle particulier {unique}

Profil UML est composé de 3 éléments  
↳ Stereotype  
↳ Tagged Value  
↳ Contraintes (OCL)

Profil ⇒ Adapter un Model UML à un domaine

## Transformation des modèles :

→ Model - to - Text : M2T  
◦ transformation du modèle en texte ⇒ Code.  
◦ Mise en œuvre : Analyse syntaxique du modèle  
Génération de fichiers.  
- M2M

→ Model - to Model  
◦ Abstraction  
◦ fusion des modèles  
◦ Refactoring → Améliorer la structure du modèle.  
◦ Raffinement → enrichir et préciser le modèle.  
◦ Mise en œuvre : Analyse syntaxique, sémantique du modèle  
Génération d'un nouveau modèle défini par un MM.  
column correspond au de de un



M2T

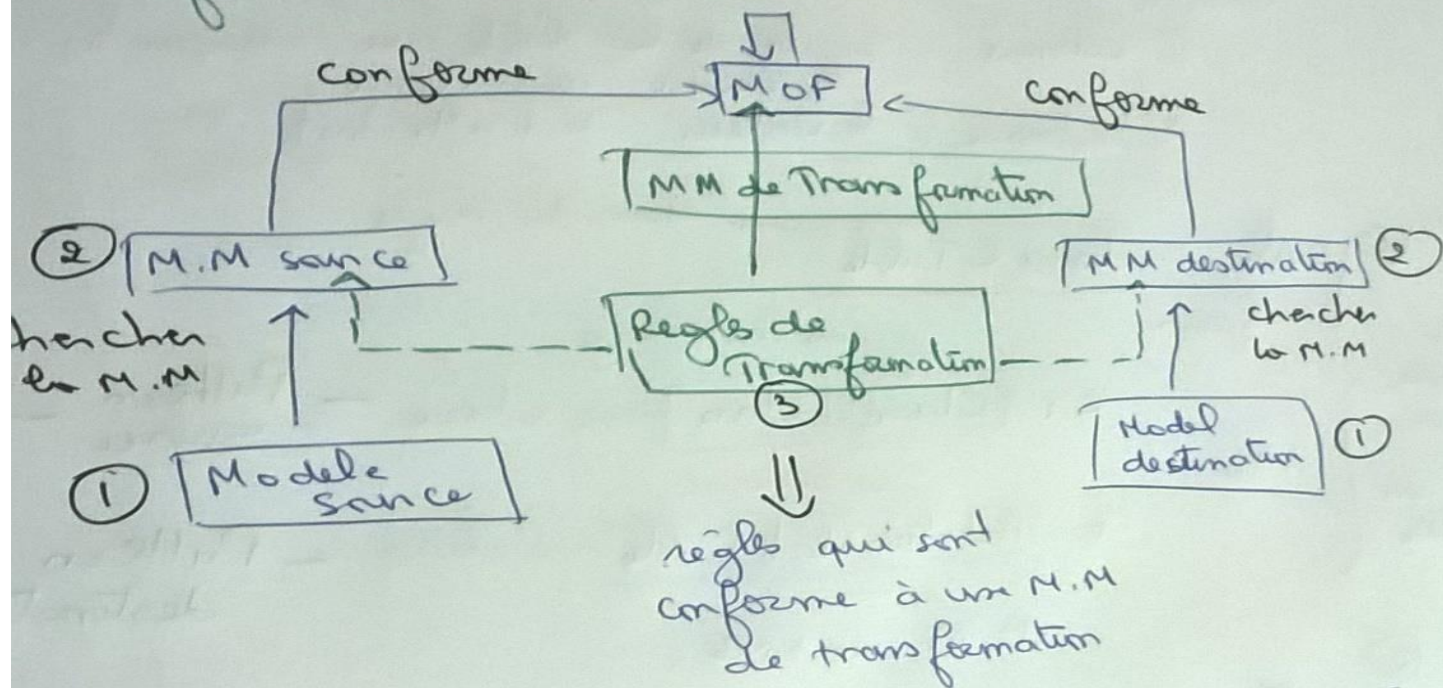
base sur des concepts et des langages de programmation

(5)

M2M

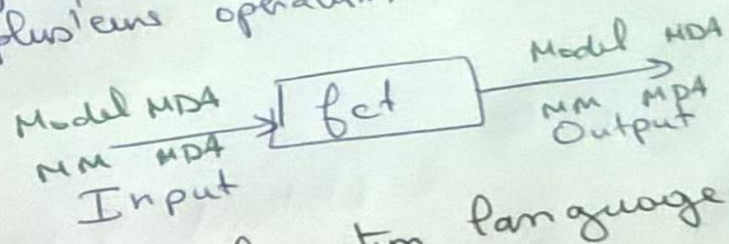
base sur des langages de Transformation (ATL)

transformation des modeles



→ Composant MDA : Produit / outil généré ou consommé par un outil MDA.

→ Outil MDA : Système opérationnel implémentant une ou plusieurs opérations MDA

→ Operation MDA:  Model MDA / M.M MDA Input → fct → Model MDA / M.M MDA Output

ATL : ATLAS Transformation language

Regles : class & Table;

- une Table est créée à partir de chaque classe
- les Columns de la Table correspond au valeurs des attributs de la classe (single-valued)
- Une column correspond au clé de la table créée.



Single Valued Attribute 2 Column

One cellule est créé par chaque attribut

Multi Valued Attribute 2 Column

Table à 2 colonnes et créé par chaque valeur attribut multiple

1<sup>re</sup> colonne : la clé de la table créé à partir de la classe génératrice  
2<sup>de</sup> colonne : la valeur de l'attribut

