

L'architecture dirigée par les modèles (MDA)



F.-Y. Villemin, CNAM

<http://deptinfo.cnam.fr/Enseignement/CycleSpecialisation/MAI/index.html>

Plan

- Les principes de l'approche MDA
- Les modèles CIM, PIM, PDM et PSM
- La transformation des modèles MDA
- Exemple

Les principes de l'approche MDA

L'initiative d'architecture dirigée par les modèles de l'OMG "Model Driven Architecture" (MDA) est motivée par le besoin de réduire les tâches de reconception des applications (nécessitées, en outre, par l'évolution constante des technologies informatiques)

Puisque les modèles sont plus pérennes que les codes, ils permettent de :

- conserver les exigences métiers (échanges entre analystes et donneurs d'ordre)
- réutiliser les choix d'architecture et de codage (échanges entre analystes et programmeurs)
- assurer l'intégrité et la cohérence entre les phases du projet (tests)

Les principes de l'approche MDA

Le principe de MDA est de séparer les spécifications fonctionnelles des spécifications de l'implantation sur une plate-forme donnée

=> interopérabilité des applications

L'idée centrale de MDA est d'élaborer des modèles, d'abord d'analyse puis de conception, jusqu'au code, par transformations, dérivations et enrichissements successifs

L'OMG propose le langage déclaratif (à base de règles) "QVT" (Query/View/Transformation) pour exprimer les transformations de ces modèles

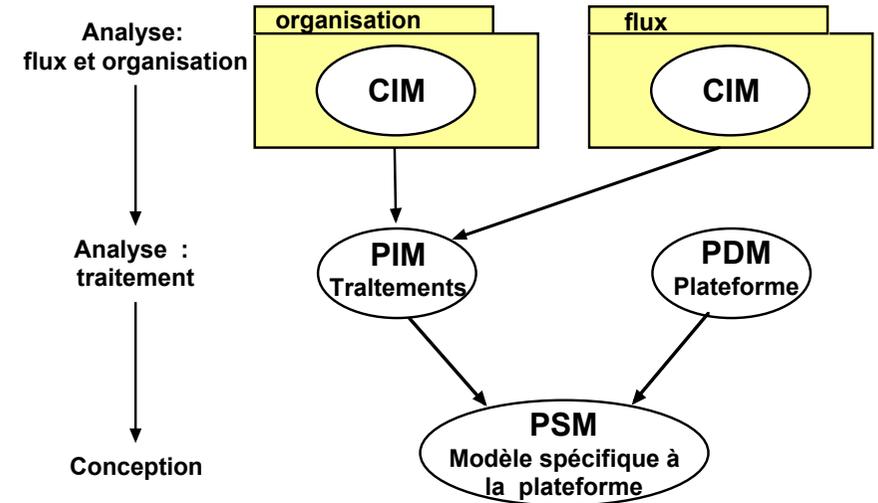
Les principes de l'approche MDA

MDA : Ensemble de techniques de modélisation et de transformation

Les principaux modèles sont :

- **CIM** (computation independant model) modèle indépendant de calcul :
décrit les flux et les actions sur le système
- **PIM** (platform independant model) modèle indépendant des plates-formes :
décrit les traitements orientés métier
- **PDM** (platform dependant model) modèle des plates-formes
décrit une architecture technique (plusieurs par projet)
- **PSM** (platform specific model) modèle dépendant des plates-formes :
décrit les détails techniques liés à l'implantation pour une plate-forme

Les principes de l'approche MDA



Le CIM (Computation Independent Model)

Le **CIM** :

- est le modèle d'analyse de base du métier ou du domaine d'application
- est indépendant de tout système informatique
- décrit les concepts de l'activité métier, le savoir faire les processus, la terminologie et les règles de gestion (de haut niveau)
- décrit la situation dans lequel le système est utilisé
- n'est modifié uniquement que si les connaissances ou les besoins métier changent (très longue durée de vie)

Les exigences modélisées dans le CIM seront prise en compte dans les constructions des PIM (Platform Independent Model) et des PSM (Platform Specific Model)

Le PIM (Platform Independent Model)

Le **PIM** :

- est un modèle de conception
- décrit le système indépendamment de toute plate-forme technique et de toute technologie utilisée pour déployer l'application
- représente la logique métier spécifique au système (fonctionnement des entités et des services)
- est pérenne dans le temps
- consiste en des diagrammes UML de classes (avec des contraintes en OCL)

Les différents niveaux de PIM précisent les choix de persistance, de gestion des transactions, de sécurité...

Le PDM (Plate-forme Description Model)

Un **PDM** :

- contient des informations pour la transformation des modèles vers une plateforme
- est spécifique à une plateforme
- est un modèle de transformation pour permettre le passage du PIM vers le PSM

Le PSM (Plate-forme Specific Model)

Un **PSM** :

- sert à la génération du code exécutable pour les plates-formes techniques particulières
- décrit comment le système utilisera la plate-forme
- est dépendant de la plate-forme

Niveaux de PSM :

Le premier niveau, issu de la transformation d'un PIM par l'adaptation des modèles UML aux spécificités la plate-forme

Les autres niveaux PSM sont obtenus par transformations successives en prenant en compte le langage (Java, C#, PHP...), les choix de conception...

Le dernier niveau, ou PSM d'implantation, décrit, en autres, le code du programme, les schémas des tables, les bibliothèques utilisées, les descripteurs de déploiement...

La transformation des modèles MDA

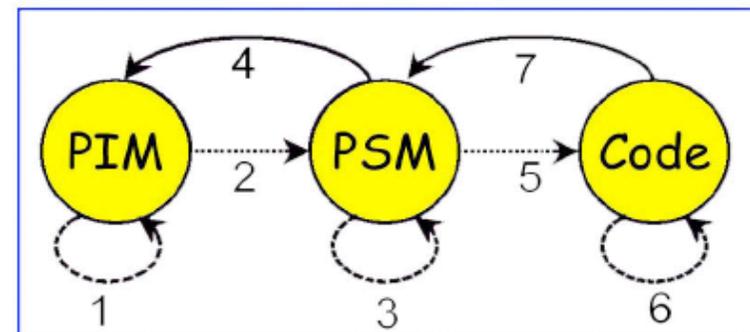
L'approche MDA précise quatre types de transformations pendant le cycle de développement, les modèles devenant de plus en plus concrets jusqu'à l'obtention du code

Par transformations successives, le PIM, modèle de niveau le plus abstrait, est transformé en un PSM exécutable (ou code exécutable)

Si la démarche MDA a été respectée, il est possible de générer un PSM, puis un PIM, à partir du code exécutable (rétro-ingénierie)

La transformation des modèles MDA

Les transformations :



La transformation des modèles MDA

Transformation **de PIM vers PIM**, ou raffinement, consiste à ajouter des informations (non liées à une plate-forme) sous forme d'annotations

Transformation **de PIM vers PSM** consiste à ajouter au PIM des informations propres à une plate-forme technique

Les plates-formes visées (J2EE, .NET...) sont décrites dans un PDM

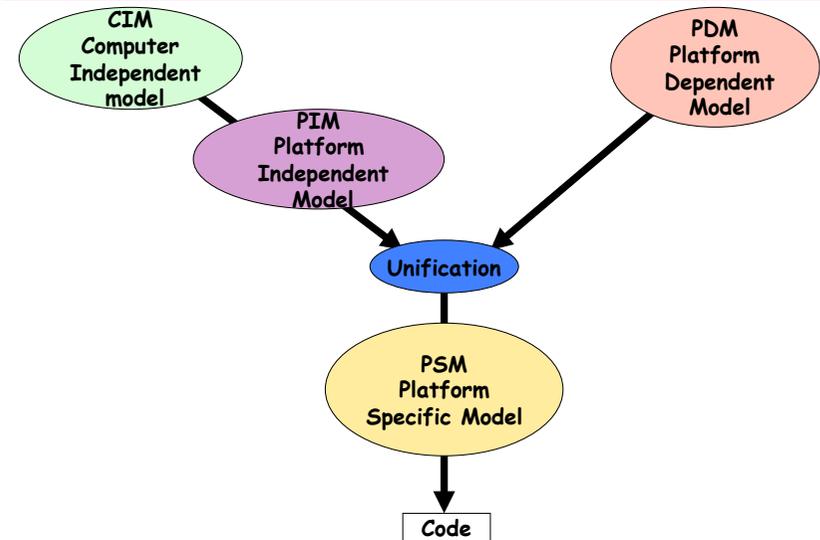
Les **règles de transformation** sont généralisées et capitalisées pour un réutilisation futur

Transformation **de PSM vers PSM** (raffinement), souvent nécessaire pour générer un code, se fait par l'utilisation de formalismes intermédiaires comme SDL

(Specification and Description Language)

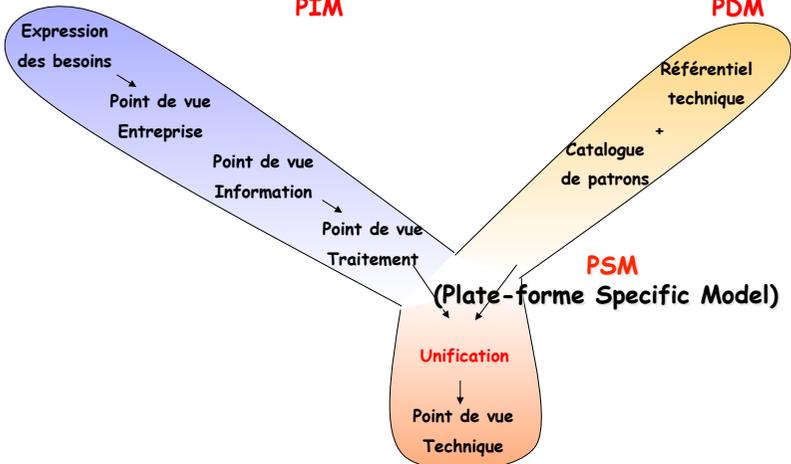
Transformation **de PSM (ou du code) vers PIM**, ou rétroingénierie (reverse engineering), indispensable pour permettre l'intégration d'applications existantes

La transformation des modèles MDA



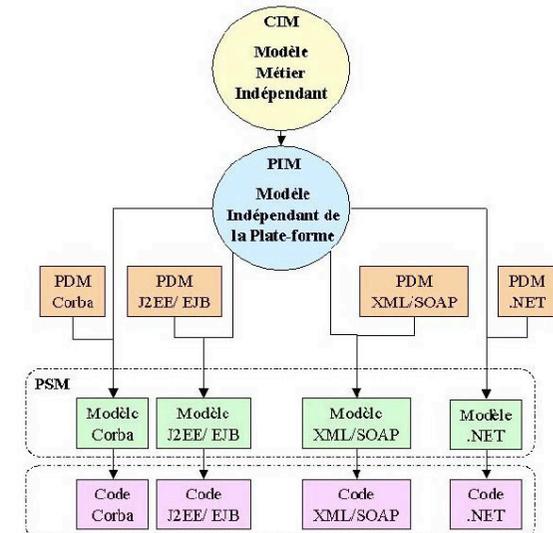
La transformation des modèles MDA

(Plate-forme Independent Model) **PIM** (Plate-forme Dependant Model) **PDM**



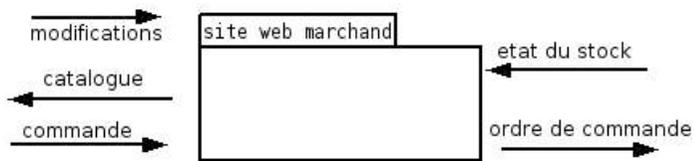
La transformation des modèles MDA

Exemple d'utilisation des modèles pour réaliser une application

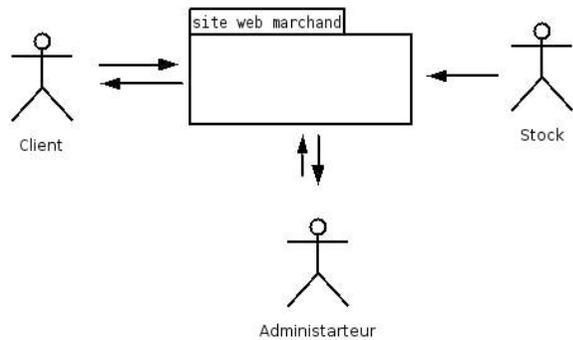


Exemple de CIM

Flux

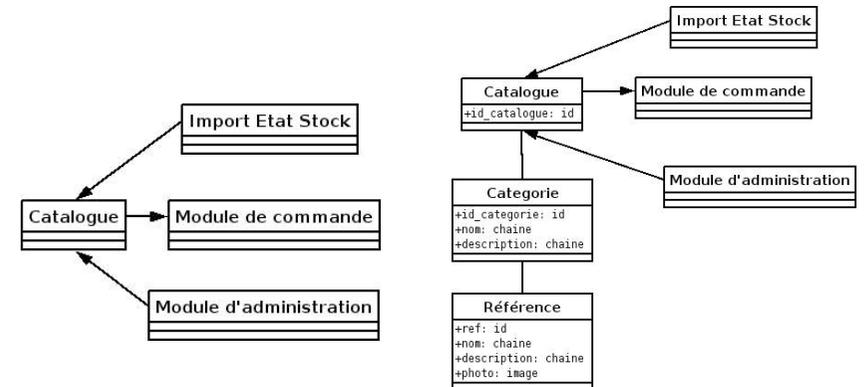


Organisation

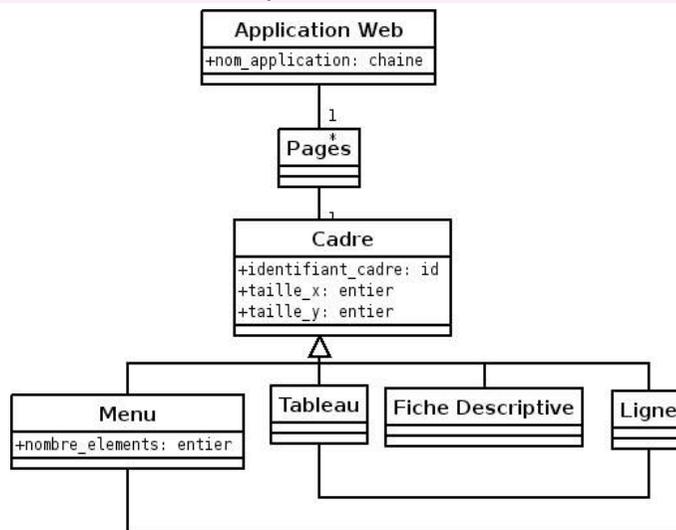


Exemple de PIM

Raffinements successifs des PIM indépendamment de tout plate-forme
Etape 1



Exemple de PDM



Exemple de PSM

