

Rendus des étudiants

MDA & MOF

Sommaire

1. ALBERT Frédéric albert.frederic@orange.fr MDA.....	2
2. ALI Loubna loubnaali@yahoo.fr Le Model Driven Architecture MDA & MOF Meta Object Facility	4
3. BENAHMED Mustapha mustapha.benahmed@laposte.net Synthèse d'informations tirées d'Internet sur les notions de MDA et MOF.....	6
4. CHERIF Fethi cfethi@yahoo.fr MDA-MOF	7
5. ELKOSANTINI Sabeur skossentini@yahoo.fr MDA et MOF.....	9
6. GRANDGIRARD Emilie egrandgir@9online.fr MDA : Model Driven Architecture	11
7. GRUAT-LA-FORME France-Anne France-anne.Gruat-la-forme@insa-lyon.fr MDA (Model Driven Architecture)	13
8. HOFFMANN Patrick hoffmannp@free.fr MDA et MOF.....	15
9. HUERTA Abraham Eneida <n_huerta70@hotmail.com> Architecture de Conduite Modèle (Model Driven Architecture) MDA	17
10. JABER Mayyad mayyad_j@hotmail.com Qu'est ce que le MDA ?.....	19
11. REYMONDON Francis freymondon@yahoo.fr MDA et MOF	20
12. SAEZ Hector <jaaipayofr@yahoo.fr> MDA (Model Driven Architecture)	22

ALBERT Frédéric
albert.frederic@orange.fr
MDA

MDA

<http://www.01net.com/article/148002.html>

Christophe Grosjean, Décision Micro, le 30/04/2001 à 00h00

Le but des standards UML,MOF,XMI est de permettre à des ateliers logiciels d'explorer et d'échanger les définitions des structures de données, leurs propriétés, les relations les unissant, ... Dans le but de former une seule base de donnée (vue par l'utilisateur).

Avant de construire une voiture, par exemple, il faut en créer les plans détaillés. Le moindre élément du véhicule, carrosserie, moteur, châssis, roues et jusqu'au plus petit boulon, doit être décrit, ainsi que les relations entre chaque élément : c'est le plan de montage. Des règles précises normalisent la façon dont de tels plans sont représentés sur papier. Si l'on souhaite créer, non plus une voiture, mais un logiciel ou une base de données, UML est le langage graphique normalisé permettant de décrire les plans d'un logiciel, c'est-à-dire les structures des données et les relations qui les lient. On parle de modèles UML. Lesquels sont rarement dessinés à la main, mais plutôt à l'aide d'outils informatiques (comme Rational Rose). En poursuivant l'analogie industrielle, MOF (Meta Object Facility) serait un ensemble de règles de réalisation des plans de la voiture : épaisseur des traits, façon de placer les légendes, etc. On doit la création de cette architecture de métaobjets MOF à l'OMG (Object Management Group), soutenu par des organismes comme IBM, Oracle, Rational ou Unisys.

<http://www.debrauwer.com/mda.html>

MDA (Model Driven Architecture) : "Architecture de pilotage de modèle" est une nouvelle proposition de l'OMG dont l'objectif est la conception de systèmes basée sur la modélisation du domaine. A partir de la conception des objets du domaine, MDA propose de concevoir par transformation les objets techniques capables de fonctionner dans une plateforme logicielle comme Java ou .NET.

Dans MDA, **le modèle des objets du domaine s'appelle PIM, c'est-à-dire "Platform Independent Model". Le PIM est constitué d'un schéma d'objets dont la conception doit se faire indépendamment de tout langage de programmation ou de technologie.** Ce modèle est ensuite transformé manuellement ou automatiquement en un modèle spécifique à une plateforme et à un langage de programmation. Un tel modèle spécifique s'appelle **PSM**, c'est-à-dire "Platform Specific Model". On trouvera dans les vues statiques d'un PSM des classes respectant les contraintes des langages de programmation comme l'héritage simple de Java ou la traduction des associations entre objets par des attributs à chaque extrémité. Le PSM introduit également les aspects techniques relatifs à la plateforme comme, par exemple, les EJB de la plateforme J2EE.

Un PSM peut ensuite être transformé, à l'aide de générateur, en code d'un langage. De tels générateurs existent déjà aujourd'hui dans la plupart des ateliers UML.

UML constitue naturellement un bon candidat comme langage de modélisation au niveau PIM. Cependant si le diagramme UML de classes muni du langage OCL permet, par son formalisme, une description précise des vues statiques, il en va autrement pour les aspects dynamiques. En effet, les différents diagrammes d'UML relatifs aux aspects dynamiques ne sont pas très formels et ne permettent pas une description complète et précise des traitements. Si l'on veut automatiser la traduction de l'aspect dynamique des objets, il sera nécessaire d'enrichir UML par un langage complet et formel de description des traitements.

Lexique:

UML (Unified Modeling Language) est un langage graphique utilisé pour la conception orientée objet (exemple de Language dans lequel un PSM peut être traduit.)

MOF (Meta Object Facility) utilise un sous-ensemble de UML pour décrire les objets manipulés par les outils de conception.

XMI (XML Metadata Interchange) Echange de métadonnées sur le Net, indique comment les modèles MOF peuvent être traduits en XML. XMI est le mécanisme utilisé pour communiquer entre registres par ebXML

MDA: Architecture de pilotage de Modèle (cadre pour modéliser des modèles PIM)

PIM: Modèle de Plateforme Indépendante (indépendant de tout langage ou de toute technologie)

PSM: Modèle créé à partir d'un PIM, spécifique à un langage et à une plateforme

1. Le Model Driven Architecture (MDA)

Le Model Driven Architecture (MDA) est une démarche de développement proposée par l'OMG . Elle permet de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate-forme donnée. A cette fin, le MDA définit une architecture de spécifications structurée en modèles indépendants des plates-formes (PIM) et en modèles spécifiques (PSM) .

L'approche MDA permet de réaliser le même modèle sur plusieurs plates-formes grâce à des projections standardisées. Elle permet aux applications d'interopérer en reliant leurs modèles et supporte l'évolution des plates-formes et des techniques. La mise en œuvre du MDA est entièrement basée sur les modèles et leurs transformations.

L'OMG s'est engagé sur la voie du MDA, afin de résoudre les problèmes d'interopérabilité et de portabilité dès le niveau modélisation. Le MDA se veut donc indépendant de toute plate-forme et de tout système, il permet de concevoir des applications portables au niveau des langages de programmation, des systèmes d'exploitation mais aussi des middlewares.

Le logo du MDA (cf figure 1) représente les différentes couches de spécification. Au cœur se trouvent les techniques (UML, MOF, CWM), autour quelques-unes des plates-formes supportées, en surface les services systèmes et enfin à l'extérieur les domaines pour lesquels des composants métiers doivent être définis (Domain Facilities). Ces services (aussi bien systèmes que métiers) doivent être disponibles dès les premières phases de modélisation, c'est pourquoi ils doivent faire partie des spécifications du MDA.

2. Meta Object Facility (MOF)

Le langage MOF [MOF] fournit le standard de méta-modélisation et d'échange de constructions utilisées par MDA. Les autres modèles standards de l'OMG, comme UML et CWM, sont définis par des constructions MOF, ce qui permet de les relier entre elles. C'est également le mécanisme par lequel les modèles sont sérialisés en XML. Le MOF est un exemple de méta-méta-modèle, ou de modèle du méta-modèle. Il définit les éléments essentiels, la syntaxe et la structure des méta-modèles utilisés pour construire des modèles orientés objet. La spécification MOF fournit les points suivants :

- Un modèle abstrait d'objets MOF génériques et leurs associations.
- Un ensemble de règles pour exprimer un méta-modèle MOF à l'aide d'interfaces IDL. Une implantation de ces interfaces pour un méta-modèle donné peut être utilisée pour manipuler une instance de celui-ci (un modèle).
- Un ensemble de règles sur le cycle de vie, la composition et la fermeture sémantique des éléments d'un méta-modèle MOF.
- Une hiérarchie d'interfaces réflexives permettant de découvrir et manipuler des modèles basés sur des méta-modèles MOF dont on ne connaît pas les interfaces.

Un intérêt du MOF est qu'il permet de faire interopérer des méta-modèles différents. Une application MOF peut manipuler un modèle à l'aide d'opérations génériques sans connaissances du domaine.

MOF structure: MOF contient 4 différent couches:

- C3= meta-metamodel.
- C2= metamodels: UML metamodel , IDL metamodel .
- C1= Model: UML models, IDL interfaces.
- C0.

déférence:

1-Meta Object Facility (**MOF**): Revised Submission to OMG RFP, 2003-04-07.

2-MDAandMOFImplementations: Hnetyuka, 2003-12-02

3-MDA and MOF: Petr Hnetyuka 2003-Apr-23

4-La démarche MDA: Projet ACCORD (Assemblage de composants par contrats en environnement ouvert et réparti), Livrable 1.1-5, Mai 2002.

BENAHMED Mustapha
mustapha.benahmed@laposte.net

Synthèse d'informations tirées d'Internet sur les notions de MDA et MOF

Sites utilisés : www.omg.org et www.dstc.edu.au

MDA (Model Driven Architecture)

Introduction _ En Mars 2001, l' OMG (Object Management Group : consortium international à but non lucratif qui propose des standards informatiques) a adopté le modèle MDA en réponse au défi grandissant que représente l'interopérabilité entre entreprises.

Présentation : C'est un modèle qui offre une approche permettant de résoudre les problèmes de développement, de déploiement et d'intégration des systèmes distribués existants avec les nouvelles technologies qui apparaissent, et ce tout le long de leurs cycle de vie.

Les spécifications de MDA sont basées sur un modèle des fonctions d'une application d'entreprise d'une façon indépendante de toute plateforme. De cette façon, les entreprises peuvent intégrer leurs différentes applications en passant outre les barrières que représentaient les plateformes.

Avec MDA, OMG propose de faire passer la standardisation des applications portables du niveau de l'implémentation vers le niveau conceptuel.

(Source : <http://www.omg.org/news/releases/pr2001/2001-03-08a.htm><http://www.omg.org/news/releases/pr2001/2001-03-08a.htm>)

MOF (Meta-Object Facility)

Il s'agit d'un standard OMG qui définit une série de Services basés sur CORBA permettant de gérer les méta-informations. Celles sont destinées à décrire d'autres informations. Les modèles UML ou les schémas de base de données en sont des illustrations.

MOF est potentiellement destiné à tout système distribué où on a besoin de représenter des méta-informations. Il permet par exemple de représenter des schémas de base de données, d'entrepôts de données ou encore dans le développement logiciel pour représenter différents modèles.

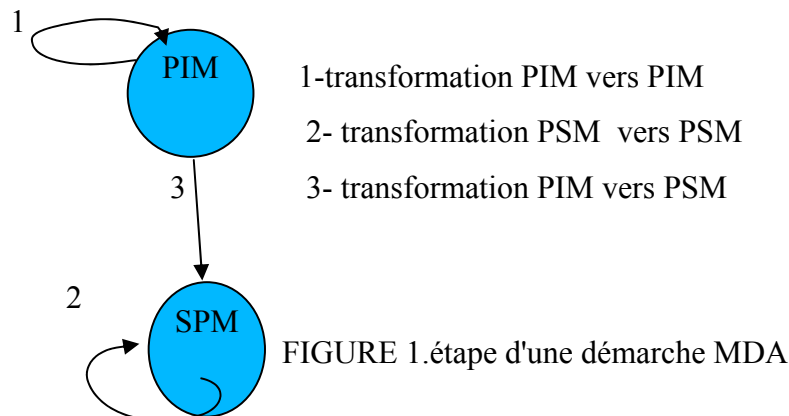
(Source : <http://www.dstc.edu.au/Research/Projects/MOF/MOFAQ.html#2>)

CHERIF Fethi
cfethi@yahoo.fr
MDA-MOF

MDA (Modèle Driven Architecture) une démarche de développement proposée par l'OMG(object Management Group),le MDA a pour objectif de définir une représentation abstraite et indépendante de toute architecture technique tout en lui associant une multitude de service métiers,et grâce a des projections la réalisation d'un modèle indépendant de toute plate forme appelée PIM (Platform Independent Model) le choix de cette plate forme et la génération du modèle spécifique correspond appelé PSM (Platform Specific Model), MDA permet de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate forme donnée.

MDA couvre toutes les étapes de développement (tout le cycle de vie) et dans une démarche MDA tout est considéré comme modèle.

Revenons maintenant au PIM, c'est le premier modèle que MDA va réaliser dans son processus et elle exprime en UM, ce PIM représente les capacités fonctionnelles métiers et le comportement du système, après sa réalisation, il sera projeté vers un modèle spécifique PSM voir figure1,et ce la demande de choisir la ou les plate formes d'exécution (plusieurs plates-formes peuvent être utilisées).



Le MDA représente les différentes couches de spécification, des techniques utilisées UML, MOF (Méta Objet Facility), CWM (Common Warehouse Métamodèle), ces derniers sont des standards important de l'OMG.

Prenons le Meta-model Object Facility (MOF),c'est un langage qui fournit le standards de méta-modélisation et d'échange de constructions utilisé par MDA, et les autre langages XML,UML sont défini par de construction MOF.

MOF est un mécanisme par le quel les modèles sont sérialisé en XML.

MOF est un exemple de méta modèle ou de modèle du méta modèle, permet de faire interopérer des méta-modèles différents, une application MOF peut manipuler un modèle a l'aide d'opération générique son connaissances du domaine.

Référence bibliographique:

<http://www.debrauwer.com/mda.html>

http://www.adae.pm.gouv.fr/upload/documents/desfray_xml_journeeadae.pdf

<http://deasi.lip6.fr/DEA-SI/THESES/Prop-Theses09-02%2F03.html>

http://gizmoeti.free.fr/mda/toutmda_html/node10.html

<http://www-valoria.univ->

<ubs.fr/Jacques.Malenfant/ALP.OCM/Journee2003/Nano.pdf>

<http://www-src.lip6.fr/homepages/Lionel.Seinturier/middleware/cs.pdf>

<http://www.e-nef.com/ai/orchid/rapports/rapport/node18.html>

MDA : Introduction et contexte

Le Model Driven Architecture (MDA) est une démarche de développement centré sur une suite de modèles intégrant toutes les dimensions du projet et permettant aux entreprises de créer des modèles qui sont essentiellement des modèles d'application indépendants de tout matériel, système d'exploitation, plate-forme ou middleware. Elle comble une lacune importante qui existait entre les modèles de gestion et le développement de logiciels en veillant à ce que les modèles pilotent le développement d'applications, et non l'inverse.

A cette fin, le MDA définit une architecture de spécifications structurée en modèles indépendants des plates-formes appelés PIM (Platform Independent model) et en modèles spécifiques appelé PSM (Platform specific model).

Démarche MDA

L'approche MDA permet de réaliser le même modèle sur plusieurs plates-formes grâce à des projections standardisées. Elle permet aux applications d'interopérer en reliant leurs modèles et supporte l'évolution des plates-formes et des techniques. La mise en oeuvre du MDA est entièrement basée sur les modèles et leurs transformations.

La démarche MDA supporte toutes les étapes de développement et standardise les passages de l'une à l'autre. Elle peut se découper en quatre points, les points 2 et 4 peuvent être répétés un nombre indéterminé de fois :

La réalisation d'un modèle indépendant de toute plate-forme appelé PIM pour « Platform Independent Model ».

L'enrichissement de ce modèle par étapes successives.

Le choix d'une plate-forme de mise en oeuvre et la génération du modèle spécifique correspondant appelé PSM pour « Platform Specific Model ».

Le raffinement de celui-ci jusqu'à obtention d'une implantation exécutable.

Dans une démarche MDA, tout est considéré comme modèle, aussi bien les schémas que le code source ou le code binaire. Les deux types de modèles identifiés sont donc les PIM et les PSM.

MOF (Meta Object Facility)

Au cœur du MDA, se trouvent plusieurs standards importants de l'OMG (Object Management Group) définissant l'infrastructure du MDA. Ils devront être complétés par des règles de transformation de modèles qui seront standardisées par l'OMG. Parmi ces standards on trouve UML (Unified Modeling Language), MOF (Meta Object Facility),...

Le langage MOF (Meta Object Facility) fournit le standard de méta-modélisation et d'échange de constructions utilisées par MDA. Un méta-modèle permet de spécifier la syntaxe et la sémantique des modèles d'un domaine particulier. Il correspond à la définition d'un ensemble de concepts, propriétés, opérations et relations entre concepts. Les concepts ont pour objectif de définir toutes les entités nécessaires lors de la modélisation d'un système spécifique.

Il définit les éléments essentiels, la syntaxe et la structure des méta-modèles utilisés pour construire des modèles orientés objet. Le MOF dispose d'un mécanisme d'extension qui permet d'ajouter de nouveaux concepts à un méta-modèle existant.

Références

La démarche MDA : <http://www.lifl.fr/~mullera/publi/demarchemda.pdf>

La Transformation Automatique est la clef pour le passage de la vision MDA à la réalité : <http://www.compuware.fr/insight/insightnews/cover.htm>

Transformation entre un profil UML et un méta-modèle MOF:

http://www.sciences.univ-nantes.fr/info/lrsg/Pages_perso/MP/pdf/lmo_2002.pdf

GRANDGIRARD Emilie
egrandgir@9online.fr
MDA : Model Driven Architecture

La prolifération des middleware incompatibles entre eux entraîne le besoin croissant d'un niveau d'abstraction supérieur afin de capitaliser la définition des applications et les rendre plus portables dans le cadre de différentes technologies. Depuis quelques années, l'OMG (Object Management Group, consortium d'entreprises) s'est donc engagé dans une nouvelle voie de développement pour résoudre les problèmes d'interopérabilité et de portabilité : l'approche MDA.

Définition

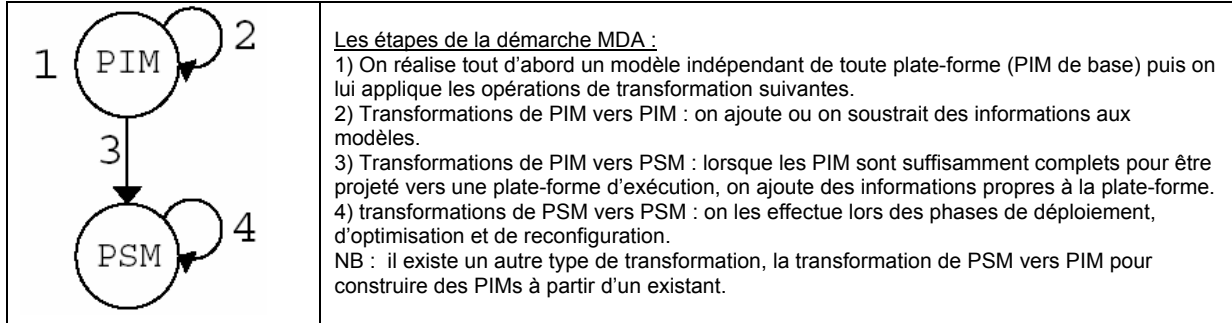
MDA est une démarche de développement qui permet de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate-forme donnée. Elle permet ainsi aux applications d'interopérer en reliant leurs modèles et supporte l'évolution des plates-formes et des techniques, car les modèles sont indépendants de ces considérations techniques.

Fonctionnement

Le principe de base de l'approche MDA consiste en l'élaboration de modèles indépendants des plates-formes : les PIM (Platform Independent Models). Ces modèles sont ensuite convertis par des opérations de transformation en modèles dépendants des plates-formes : les PSM (Platform Specific Models).

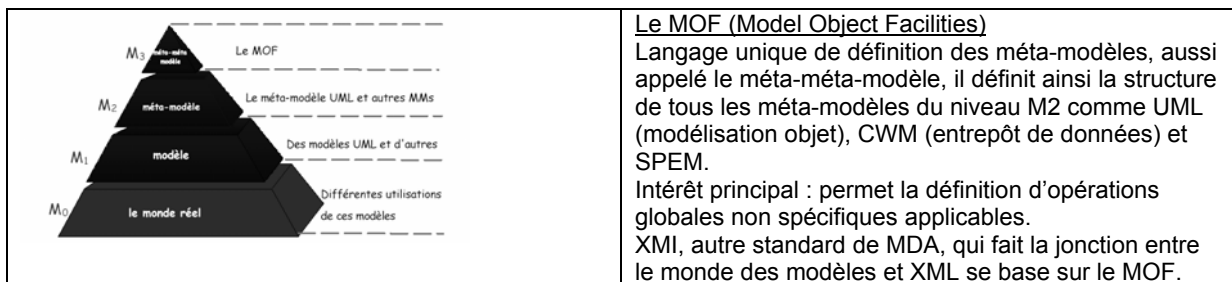
Les PIM : à la base, on a une simple représentation des différentes entités fonctionnelles d'un système avec leurs interactions, mais ce modèle peut être enrichi avec l'intégration des aspects technologiques et architecturaux (informations sur la persistance, les transactions, la sécurité,...).

Les PSM : ils servent essentiellement de base à la génération de code exécutable vers les plates-formes techniques. Le PSM de base est obtenu directement à partir du modèle PIM mais il peut être raffiné par transformations successives jusqu'à l'obtention d'une implantation exécutable.



L'architecture à quatre niveaux

Etant donné que les modèles manipulés dans le MDA sont de natures très diverses (modèles d'objets métiers, modèles de processus, modèles de règles, modèles de service,...), l'OMG a défini une structure appelée « architecture à quatre niveaux » afin d'organiser tous ces modèles.



Avantages, inconvénients

Avantages : capitalisation des acquis logiciels, possibilité d'interopérer avec toutes les plates-formes.

Inconvénients : on ne sait pas actuellement si le MDA va vraiment permettre de résoudre tous les problèmes actuels.

Conclusion : avenir

Le MDA est un projet ambitieux : c'est un énorme chantier qui débute et qui va certainement durer plusieurs années !

Références :

- La démarche MDA
<http://www.lifl.fr/~mullera/publi/demarchemda.pdf>
- MDA : Vers un important changement de paradigme en génie logiciel
<http://www.sciences.univ-nantes.fr/info/perso/permanents/vailly/Enseignement/Documents/MDA.Partie1.JBXB.Last.prn.pdf>
- Vers des patrons de méta-modélisation
<http://www-valoria.univ-ubs.fr/Jacques.Malenfant/ALP.OCM/Journee2003/Marvie.pdf>
- MDA : Model Driven Architecture
http://gizmoeti.free.fr/mda/toutmda_ppt.pdf

GRUAT-LA-FORME France-Anne
France-anne.Gruat-la-forme@insa-lyon.fr
MDA (Model Driven Architecture)

Le thème des recherches de cette semaine est le MDA. Ne connaissant pas du tout ce modèle, j'ai basé mon apprentissage sur les informations fournies par Internet, la notion étant trop récente pour figurer dans un quelconque livre de doc-insa. Les références de mes recherches figurent en fin d'exposé.

Ma présentation se déroulera en trois points successifs. Dans un premier temps, je mettrai en évidence les besoins qui ont favorisé la naissance du projet MDA par l'OGM. J'expliquerai ensuite le principe de fonctionnement de cette méthode pour finir sur les perspectives et défis que doit remplir le MDA pour vraiment atteindre l'objectif fixé à l'origine.

Pourquoi MDA ?

Depuis longtemps, un des objectifs majeurs de l'évolution informatique est de faciliter l'interopérabilité et la portabilité d'applications différentes. Une multitude de middlewares ont ainsi vu le jour. Cette recrudescence est à l'origine même du problème actuel, car ces middlewares ne peuvent pas naturellement communiquer entre eux. Une solution à court terme a été de concevoir de nombreux ponts entre middlewares mais ceci n'est évidemment pas viable éternellement.

L'OMG a donc décidé de lancer un projet visant à standardiser la description des architectures et ensuite favoriser les transformations vers les plates-formes spécifiques.

Principe général de MDA

MDA est une architecture structurée en modèles indépendants des plates-formes d'utilisation. D'un point de vue macroscopique, le but est de capitaliser le travail et ce dès le début des opérations (ie : la modélisation). La finalité est, d'une part de pouvoir préserver les modèles lorsqu'un nouveau middleware arrive sur le marché et d'autre part, pouvoir suivre les évolutions des plates-formes et des techniques sans avoir à tout détruire pour reconstruire.

Pour cela, le principe de MDA se fonde entièrement sur les modèles suffisamment génériques, qui vont constituer une base commune et qui vont être amenés à se transformer, ensuite, selon les besoins spécifiques

Il sera ensuite possible d'inter opérer avec toutes les plates-formes possibles (ex Java ou XML), à partir d'une même souche, afin de bénéficier des avantages de chacune.

Précisions techniques

Le MDA se compose de 2 modèles successifs : le PIM et le PSM ; le PIM va définir les caractéristiques communes à toutes les plates-formes. Le PSM consiste à implémenter les caractéristiques spécifiques à une plate-forme donnée.

Les PIM ne représentent que les capacités fonctionnelles d'un métier et le comportement du système mais restent toujours indépendant des plates-formes. Le PIM n'a pas sa version final d'un coup. Un PIM de base est défini et par étapes successives, il faut enrichir et spécialiser les PIM précédents. Lorsque nous disposons d'un PIM suffisamment raffiné, nous le projetons vers une plate-forme d'exécution. C'est à cette étape qu'intervient le deuxième type de modèle : le PMS. Ce qui a été conçu de manière générique est alors converti pour tenir compte des spécificités de la plate-forme. La aussi, nous assistons à des transformations successives de PSM pour arriver au PSM final exécutable.

Nous avons vu un peu plus tôt que le souhait de MDA était aussi de pouvoir faire communiquer des applications de plates formes différentes. Ceci est possible et se matérialise dans la conception des PIM où des relations sont prédéfinies entre les futures plates-formes.

Evolutions et perspectives des MDA

MDA est encore sous forme d'un vaste chantier et doit encore séduire les opinions en relevant de nombreux défis.

1. Défis d'ordre technique : Créer des modèles prend du temps. Il faudra donc capitaliser au maximum ce travail en maîtrisant la manipulation (séparation, fusion) et les transformations de ces modèles.
2. Défis d'ordre méthodologiques : il faudra définir les méthodes qui sont applicables au MDA.
3. Défis d'ordre stratégique : L'objectif est de faire préférer le MDA aux entreprises qui accepteront alors de se détacher des autres middlewares déjà en place. Pour faciliter l'adoption du MDA il faudra bien définir les métiers ayant intérêt à l'utiliser, les compétences requises ainsi que le retour sur investissement potentiel.

Bibliographie :

WWW.lifl.fr

www.sciences.univ-nantes.fr

www-sop.inria.fr/oasis/Didier.Parigot/MDA/resumer.doc

www.gizmoeti.free.fr

HOFFMANN Patrick
hoffmannp@free.fr
MDA et MOF

Contexte

L'OMG, Object Management Group, cherche à définir des standards et à développer des modèles permettant de réaliser les principes objets fondamentaux [www.design-up.com]. Ainsi, la démarche actuelle est de s'affranchir des diverses plate-formes et de la technicité : ce concept est appelé MDA pour Model Driven Application (Application dirigée d'après un modèle)

Pourquoi s'affranchir des plate-formes ? Parce qu'il y en a plusieurs, et que cela pose des problèmes de portabilité (une fois qu'un système est développé pour une plate-forme, tel CORBA, EJB, DotNet, ... il faut tout reprendre si on veut l'adapter à une autre utilisation!) et d'inter-opérabilité (il n'est pas évident de garder une manière identique de communiquer entre systèmes lorsque les plate-formes sont différentes!).

Qu'est-ce que le MDA ?

C'est en fait une succession de modèles, du plus haut niveau (le plus abstrait, conceptuel, ... mais aussi le plus orienté métier, donc le plus compréhensible par les clients et aussi le plus pérenne) au plus bas niveau : lorsque le modèle est implanté pour une plate-forme, pour un (ensemble de)langage(s) de programmation donné.

On passe d'un modèle à un autre par transformations (raffinements) successives. Il s'agit tout d'abord d'enrichir, filtrer ou spécialiser le modèle initial (par exemple des informations sur la persistance, les transactions, la sécurité, etc.). Lorsque le modèle est suffisamment détaillé, il est projeté (le but de l'OMG est que cette étape puisse être rendue automatique) vers un modèle spécifique à une plate-forme donnée, dans le but de l'implémenter. Jusqu'alors, il s'agissait d'un PIM (Platform Independent Model), désormais il sera reconnu sous le nom de PSM (Platform Specific Model).

Ensuite, on fait évoluer le modèle de manière successive, de manière la plus automatique possible, jusqu'à l'obtention du système exécutable (exemple de transformations : génération de code, compilation, mise en paquetages, initialisation, configuration, ...).

Comment sont décrits les modèles préconisés par la démarche MDA ?

Les modèles PIM sont tous décrits en UMLa (UML version 2.0); la plupart des modèles PSM également mais pour être précis, ils nécessitent les extensions d'UMLa, qui sont les stéréotypes, les valeurs marquées, les contraintes. Mais pour être plus précis, c'est le langage MOF qui est la brique de base : UML est décrit à l'aide de ce langage, ainsi que d'autres modèles standards de l'OMG (tel CWM, Common Warehouse Metamodel), ce qui permet une communication entre eux.

Qu'est-ce donc que le MOF ?

Le MOF (méta-object facilities) est un langage graphique, qui est décrit de manière textuelle par XMI (eXtensible Markup Interchange), basé lui sur le format de description de données XML. Il s'agit d'un langage permettant de décrire la signification des méta-modèles que sont UML, CWM, CCM (et autres : voir le catalogue des méta-modèles [OMG]) . Il permet donc une interopérabilité entre méta-modèles.

La définition de l'OMG : le MOF est conçu pour permettre la communication dans une optique conduite par le modèle ; il est prévu pour définir, manipuler et intégrer des méta-données et des données sans dépendance aucune avec une plate-forme. Les standards basés sur MOF (UML, CWM, ...) peuvent ainsi permettre la communication entre outils, applications et données.

La version actuelle (standard) est la 1.5, mais l'OMG travaille actuellement sur la version 2.0

Que faire des systèmes déjà réalisés ?

L'OMG préconise des transformations inverses, permettant, en partant d'un système existant, de remonter d'un niveau et d'arriver finalement à s'affranchir des langages utilisés et de la plate-forme, afin d'arriver jusqu'au modèle représentant les concepts métiers ... mais bien sûr, il s'agit là d'un doux rêve.

Bibliographie :

[www.design-up.com] <http://www.design-up.com/design/principesoo/index.html>

[OMG] http://www.omg.org/technology/documents/modeling_spec_catalog.htm

Pages francaises :

Conférence Univ Nantes, mars 2003

Mémoire de DESS Informatique 2001-2002 Etienne Juliot

http://gizmoeti.free.fr/mda/toutmda_html/node7.html ou <http://www.toutmda.fr.st/>

un exemple : XML et MOF

<http://www.01net.com/article/148002.html>

Pages anglaises

www.omg.org/mda/

http://modeldrivenarchitecture.esi.es/mda_publicDocuments.html

<http://www.dstc.edu.au/Research/Projects/MOF/>

Article bien référencié :

K. Baclawski, M. Kokar, J. Smith, "Metamodeling Facilities, Work in progress for UML 2.0 Math Framework and MOF 2.0" Transformation Proposal for OMG

The Role of Metamodeling in MDA

www.metamodel.com/wisme-2002/papers/atkinson.pdf

Books

MDA Explained: The Model Driven Architecture: Practice and Promise

HUERTA Abraham Eneida
<n_huerta70@hotmail.com>
Architecture de Conduite Modèle
(Model Driven Architecture) MDA

Le MDA est l'outil qui permet à une industrie de décrire ses fonctions indépendamment des implémentations. Cette approche contribue à l'émergence de normes en offrant une lisibilité que peuvent partager une masse critique d'entreprises.

Les avantages que retirent les entreprises sur le MDA sont principalement le fait qu'une architecture basée sur MDA est prête pour les évolutions technologiques. la plus grande facilité d'intégration des applications et des systèmes autour d'une architecture partagée. une interopérabilité plus large permettant de ne pas être lié à une plateforme.

En terme de produits, MDA est implémenté dans un outil qui intègre la modélisation et le développement. Il gère des classes servant les objets métiers, leur présentation et leur persistance.

Adresses de sites Web visités

http://www.omg.org/mda/mda_files/3302_AccelDev_PP.pdf

<http://www.developpez.com/hcesbronlavau/MDA.htm> (de aqui lo obtuve)

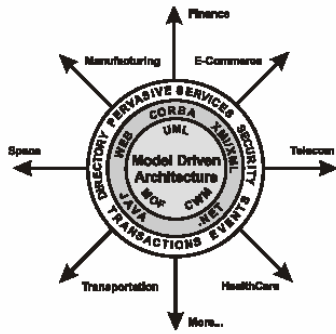
http://www.service-architecture.com/web-services/articles/model_driven_architecture_mda.html

<http://www.cwmforum.org/Model-Driven%20Architecture.pdf>

Service de Méta Objet
(Meta Object Facility) MOF

Le service d'Méta-Objet (MOF) est un service commun de CORBA pour la gestion de méta-information. Le MOF est prévu pour l'usage dans une grande variété de scénarios - de la gestion de type au développement de logiciel, de la gestion de l'information et des données entreposant - le MOF peut être employé comme dépôt de le méta-information dans les systèmes répartis par CORBA.

MOF, le service d'Méta-Objet, est une norme de groupe de gestion d'objet pour représenter le metamodèles, et les manoeuvrer (par CORBA). Le modèle (mais pas la représentation) de MOF a quelques similitudes à UML, mais il a été conçu principalement pour l'usage avec CORBA, et par conséquent exclut quelques équipements il est difficile employer que dans CORBA et inclut d'autres fonctions de CORBA-specific.



Adresses de sites Web visités

<http://www.dstc.edu.au/Research/Projects/MOF/>

<http://www.adtmag.com/article.asp?id=3965>

http://www.tutorig.com/encyclopedia/getdefn.jsp?keywords=Meta-Object_Facility

JABER Mayyad
mayyad_j@hotmail.com
Qu'est ce que le MDA ?

Le MDA, ou Model Driven Architecture, est une démarche de développement proposée par l'OMG.(Object Management Group) Cette démarche permet la séparer deux principales phases de développement d'un logiciel que sont les spécifications et l'implémentation.

Le MDA définit ainsi deux types modèles principaux:

- les PIM (Platform Independent Model) qui comment leur nom l'indiquent font abstraction de toute implémentation. On trouve dans cette catégorie (du plus général au plus pointu) le modèle UML, MOF ou encore CWM.
- les PSM (Platform Specific Model) qui au contraire concernent la partie "implémentation". On retrouve ici des modeles comme Java, .Net, etc

L'objectif de MDA est de créer une représentation UML de la logique métier et de lui associer des caractéristiques MDA. Ensuite, il suffit de demander une génération automatique des composants en fonction de l'architecture à composants choisie. Du travail complémentaire est encore bien sûr nécessaire afin de raffiner le modèle obtenue en fonction du contexte choisi (mais le travail est largement allégé).

Meta Object Facility (MOF)

Le langage MOF [MOF] fournit le standard de méta-modélisation et d'échange de constructions utilisées par MDA. . C'est également le mécanisme par lequel les modèles sont sérialisés en XML. Le MOF est un exemple de méta-méta-modèle, ou de modèle du méta-modèle. Il définit les éléments essentiels, la syntaxe et la structure des méta-modèles utilisés pour construire des modèles orientés objet. La spécification MOF fournit les points suivants :

- Un modèle abstrait d'objets MOF génériques et leurs associations.
- Un ensemble de règles pour exprimer un méta-modèle MOF à l'aide d'interfaces IDL. Une implantation de ces interfaces pour un méta-modèle donné peut être utilisée pour manipuler une instance de celui-ci (un modèle).
- Un ensemble de règles sur le cycle de vie, la composition et la fermeture sémantique des éléments d'un méta-modèle MOF.
- Une hiérarchie d'interfaces réflexives permettant de découvrir et manipuler des modèles basés sur des méta-modèles MOF dont on ne connaît pas les interfaces.

Un intérêt du MOF est qu'il permet de faire interopérer des méta-modèles différents. Une application MOF peut manipuler un modèle à l'aide d'opérations génériques sans connaissances du domaine.

<http://www.omg.org/mda/>

http://gizmoeti.free.fr/mda/toutmda_html/node7.html

<http://formation.t->

systems.fr/cours_M5_De_Merise_vers_la_mod%C3%A9lisation_avec_UML.html

<http://www.lifl.fr/~mullera/publi/demarchemda.pdf>

REYMONDON Francis
freymondon@yahoo.fr
MDA et MOF

MDA : Model Driven Architecture (Ingénierie des modèles VS la théorie des objets)
MOF : Meta Object Facility

MDA est une démarche de développement proposée par l'OMG. Elle permet de séparer les spécifications fonctionnelles d'un système des spécifications de son implémentation sur une plate-forme donnée.

L'approche MDA permet de réaliser le même modèle sur plusieurs plates-formes grâce à des projections standardisées. Elle permet aux applications d'interopérer en reliant leurs modèles et supporte l'évolution des plates-formes et des techniques. La mise en oeuvre du MDA est entièrement basée sur les modèles et leurs transformations.

Le MDA introduit la notion de Modelware, et propose une alternative ou représente l'avenir des Middlewares

Le MDA veut résoudre les problèmes d'interopérabilité et de portabilité dès le niveau modélisation. Le MDA se veut donc indépendant de toute plate-forme et de tout système, il permet de concevoir des applications portables au niveau des langages de programmation, des systèmes d'exploitation mais aussi des middlewares. Cette indépendance totale doit permettre de changer d'infrastructure sans perdre ce qui a déjà été conçu. Cette approche permet ainsi de capitaliser le travail effectué pendant les phases d'analyse et de conception.

La démarche MDA : 4 étapes

1 - Réalisation du modèle indépendant de toute plate-forme appelé PIM : Platform Independent Model

La première étape du processus MDA consiste à réaliser un modèle indépendant de toutes plates-formes

(PIM), exprimé en UML. Le PIM de base représente uniquement les capacités fonctionnelles métiers et le comportement du système, sans "dégradations" par des considérations technologiques.

Le bénéfice de l'indépendance technique de ce modèle est qu'il garde tout son intérêt au cours du temps et doit être modifié uniquement si les connaissances ou besoins métiers changent.

2. L'enrichissement de ce modèle par étapes successives.

3. le choix d'une plate-forme de mise en oeuvre et la génération du modèle spécifique PSM : Platform Specific Model .

Une fois le PIM suffisamment détaillé, il est projeté vers un modèle spécifique (PSM). Pour obtenir un modèle spécifique, qui tient compte des spécificités de la plate-forme.

4. le raffinement de celui-ci jusqu'à obtention d'une implantation exécutable.

Les bases techniques

Au coeur du MDA, se trouvent plusieurs standards importants de l'OMG : Unified Modeling Language (UML), le **Meta Object Facility (MOF)**, le XML Metadata Interchange (XMI) et le Common Warehouse Metamodel (CWM).

Meta Object Facility (MOF)

Le langage MOF fournit le standard de méta-modélisation et d'échange de constructions utilisées par MDA. Les autres modèles standards de l'OMG, comme

UML et CWM, sont définis par des constructions MOF, ce qui permet de les relier entre elles. C'est également le mécanisme par lequel les modèles sont sérialisés en XML. **Le MOF est un exemple de méta-méta-modèle**, ou de modèle du méta-modèle. Il définit les éléments essentiels, la syntaxe et la structure des méta-modèles utilisés pour construire des modèles orientés objet.

Sources bibliographiques principales :

<http://www.lifl.fr/jfiadsma2002/talks/jfiadsma2002-Bezivin.pdf>

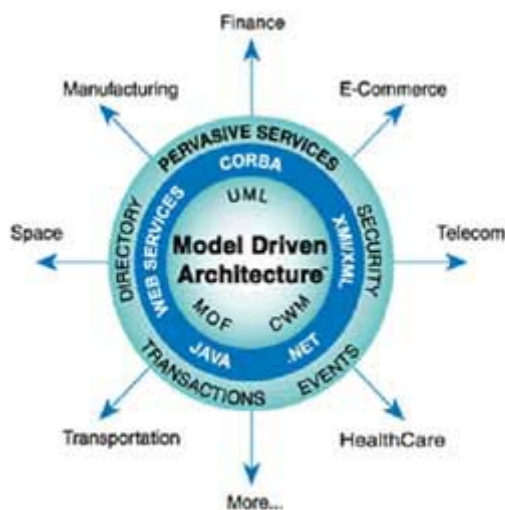
<http://www-valoria.univ-ubs.fr/Jacques.Malenfant/ALP.OCM/Journee2003/Marvie.pdf>

<http://www.sciences.univ->

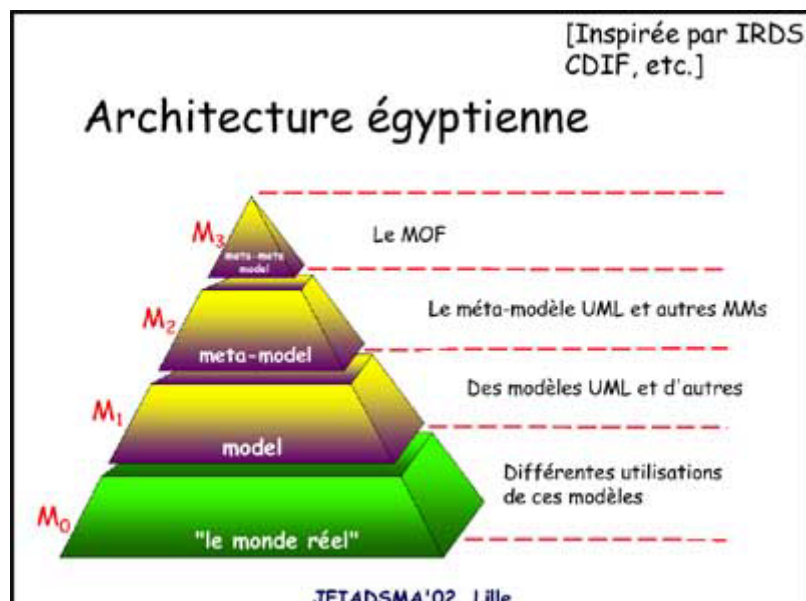
[nantes.fr/info/perso/permanents/vailly/Enseignement/Documents/MDA.Partie2.JBXB.Last.prn.pdf](http://www.sciences.univ-nantes.fr/info/perso/permanents/vailly/Enseignement/Documents/MDA.Partie2.JBXB.Last.prn.pdf)

<http://www.lifl.fr/~mullera/publi/demarchemda.pdf>

MDA



MOF



SAEZ Hector
<jaaipayofr@yahoo.fr>
MDA (Model Driven Architecture)

Le Object Management Group (OMG) a changé de l'Object Management Architecture vision (OMA) au Model-Driven Architecture (MDA). Il s'agit de **penser l'application au niveau du modèle** et de laisser le soin de l'implémentation aux outils. Ce paradigme est omniprésent en informatique. Le MDA parle de **l'interopérabilité au niveau des modèles**. Il s'agit d'avoir la possibilité d'écrire et de faire évoluer le modèle en fonction de l'organisation à la vitesse requise par l'organisation et non plus par les plate-formes.

Au niveau de l'organisation on parle de **PIM (Platform Independent Model)** ; au niveau des plate-formes, de **PSM (Platform Specific Model)**. Une application complète de MDA se composera donc d'un PIM et d'un ou de plusieurs PSM. On retrouve à un niveau d'abstraction supérieur la démarche qui avec le même langage de programmation permet d'obtenir un exécutable qui pourra faire appel aux API Windows ou à Linux, avec le même source.

La langage de choix dans lequel écrire le MDA est UML. C'est en effet un standard depuis son adoption par l'OMG en 1997. Les PSM peuvent communiquer entre eux en faisant intervenir plusieurs plate-formes qui ont à échanger des données : le développement de CORBA continuera d'être favorisé par l'OMG dans ce sens.

Cette approche globale de haut niveau présente beaucoup d'avantages par rapport aux approches spécifiques. En effet, le partage d'une logique commune au sein d'une industrie réduit d'emblée les nombreux problèmes d'interconnexion de systèmes qu'ont fait surgir au fil du temps les particularismes développés sans concertations. **Le MDA est l'outil qui permet à une industrie de décrire ses fonctions indépendamment des implémentations**. Cette approche contribue à l'émergence de normes en offrant une lisibilité que peuvent partager une masse critique d'entreprises.

Les avantages que retirent les entreprises sur le MDA sont principalement:

- le fait qu'une architecture basée sur MDA est **prête pour les évolutions technologiques**.
- la plus grande facilité d'intégration des applications et des systèmes autour d'une **architecture partagée**.
- une **interopérabilité** plus large permettant de ne pas être lié à une plate-forme.

En terme de produits, MDA est implémenté dans un outil qui intègre la modélisation et le développement. Il gère des classes servant les objets **métiers**, leur **présentation** et leur **persistance**.

Links:

www.developpez.com/hcesbronlavau/MDA.htm

www.omg.org/mda/ Web Site officiel (Anglais)

Livres et documents:

www.amazon.com/exec/obidos/tg/detail/-/0471319201?v=glance

<http://www.omg.org/mda/presentations.htm> (Anglais)

MOF (Meta-Object facility)

Le MDA est muni surtout par quelques normes d'OMG telles que: le Unified Modeling Language (UML), le **Meta-Object Facility (MOF)**, le XML Metadata Interchange (XMI), et le Common Warehouse Metamodel (CWM).

Le MOF permet spécifications dérivées pour travailler ensemble d'une manière naturel. Le MOF aussi définit un dépôt standard pour meta-modèles et donc, pour les modèles (parceque un meta-modèle c'est un cas spécial de modèle).

Le MOF sert comme un modèle común entre les modèles CWM et UML.

Le MOF fourni :

- un modèle abstrait d'objets MOF génériques et ses associations.
- une ensemble de règles pour transforme un modèle basé sur MOF dans une interface indépendant du langage (défini en IDL de CORBA). Une implémentation de cette interface pour un méta-modèle concret peut servir pour accéder et modifier quelque modèle basé sur un méta-modèle.

Links:

<http://www.omg.org/mda/specs.htm#MOF> (Anglais)

<http://www.omg.org/technology/documents/formal/mof.htm> MOF version 1.3 (Anglais)

Documents:

www.cwmforum.org/Model-Driven%20Architecture.pdf (Anglais)

<http://www.omg.org/docs/formal/02-04-03.pdf> (Anglais)