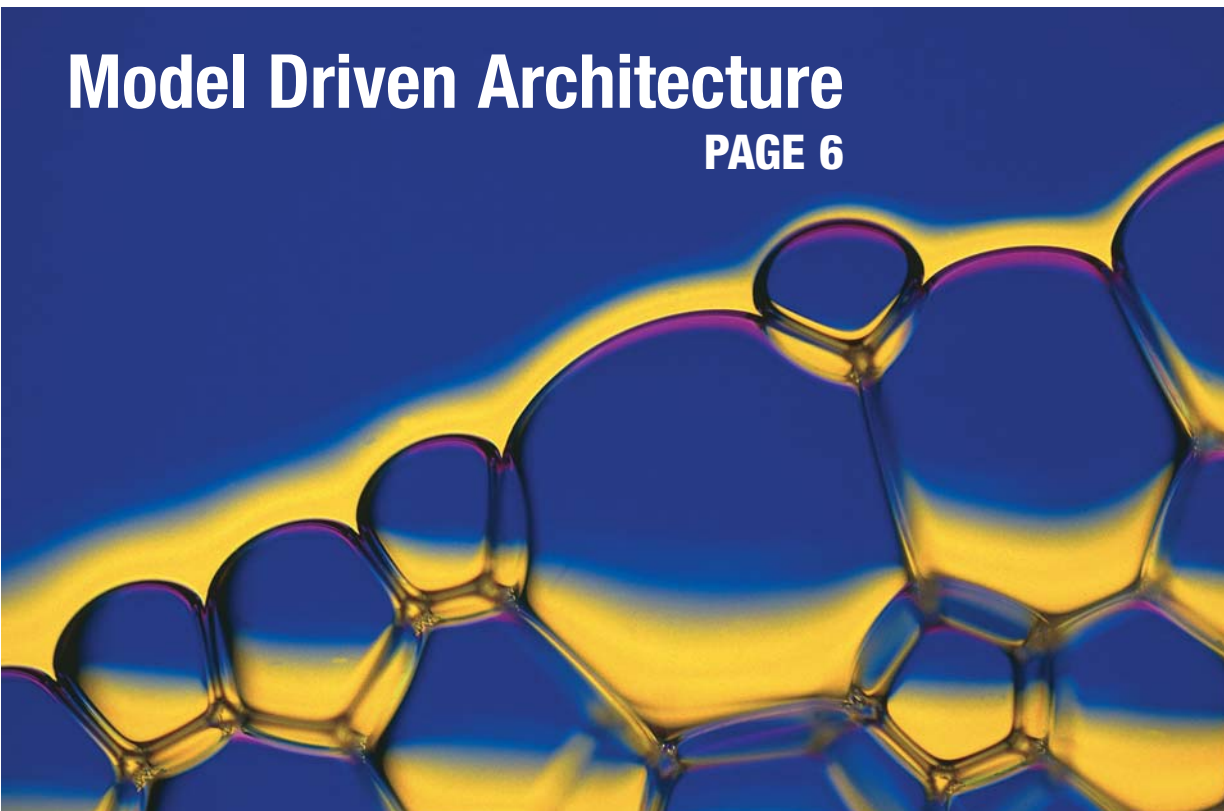


Model Driven Architecture

PAGE 6



CMMI

PAGE 38

**Travail collaboratif et open source :
des opportunités ?**

PAGE 19



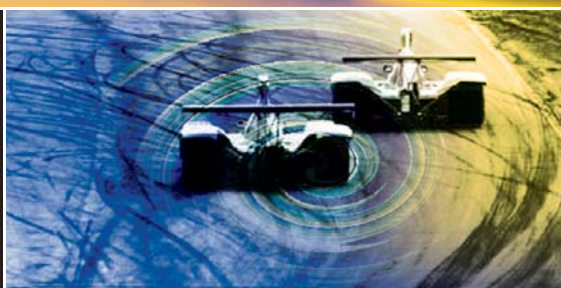
Nomadisme & sécurité

PAGE 53



n°51

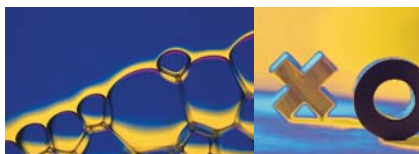
Bimestriel
Septembre/Octobre
2004
9€



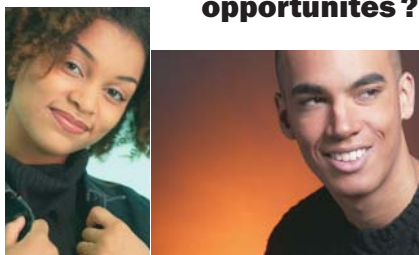
**Les évolutions récentes
en ingénierie des modèles**
PAGE 33

Sommaire

6 MDA: Model Driven Architecture



19 Travail collaboratif et open source: des opportunités ?



33 Les évolutions récentes en ingénierie des modèles



38 CMMI

Capability Maturity Model Integration



6 Dossier

MDA: Model Driven Architecture ou architecture guidée par le modèle

L'architecture guidée par les modèles constitue-t-elle l'avenir du développement logiciel ou n'est-elle que le énième avatar d'une longue lignée débutée avec les outils CASE ? Notre dossier propose un premier aperçu des mythes et réalités de ce que l'OMG et de nombreux éditeurs considèrent comme une voie d'avenir.

19 Technique

Travail collaboratif et open source: des opportunités ?

Les outils de travail collaboratif sont un puissant moyen d'améliorer la productivité. Malheureusement de tels outils sont coûteux. L'open source est donc une voie à considérer, tant pour les PME aux budgets serrés que pour les entreprises de taille plus conséquente qui y verront le moyen de mûrir leur projet.

28 Actualités Internationales

Actualité internationale et annonces produits chez les éditeurs...

33 Quoi de neuf Docteur ?

Les évolutions récentes en ingénierie des modèles

Le MDA qui fait l'objet de notre dossier connaît de nombreuses et fréquentes avancées. Cet article fait le point sur les développements les plus récents en la matière.

38 Comment ça marche ?

CMMI - Capability Maturity Model Integration

Le modèle de maturation des capacités est une méthode évolutive de gestion du processus de développement logiciel, souvent positionné comme une alternative pragmatique des modèles ISO. Au cours des années, plusieurs versions ont vu le jour. Le CMMI, constitue une tentative - a priori réussie - d'unification de ces différentes évolutions.

46 Fenêtre sur cour

Interview THALES

Serge Salicki, responsable du programme pilote Mirror au sein de Thales Research & Technology nous explique les raisons qui ont poussé l'un des grands acteurs mondiaux de l'électronique à mettre en place une démarche d'ingénierie guidée par les modèles (MDE) et les enseignements qu'il en tire.

54 Rubrique à bras

Nomadisme & sécurité

La possibilité d'accéder en tout lieu et en tout temps au système d'information de l'entreprise est un projet prioritaire pour deux tiers des entreprises françaises. Cette facilité s'accompagne de fortes préoccupations sécuritaires appelant des solutions spécifiques aux technologies mises en œuvre. Cet article fait le point sur ce sujet.



Les évolutions récentes en ingénierie des modèles



■ Le MDA™ et l'ingénierie des modèles

En novembre 2000 l'OMG a proposé une approche nommée Model Driven Architecture (MDA™) pour le développement et la maintenance des systèmes à prépondérance logicielle. L'idée initiale consistait à s'appuyer sur le standard UML pour décrire séparément les parties des systèmes indépendantes des plates-formes spécifiques (PIM ou Platform Independent Models) et les parties liées à ces plates-formes (PSM ou Platform Specific Models). Il s'agissait de parier sur la stabilité possible de ce standard pour que l'on puisse reprendre dans 5, 10, 15 ans ou plus des descriptions abstraites en UML (modèles métiers) et générer à partir de celles-ci des systèmes exécutables sur les nouvelles plates-formes technologiques du futur. Dans les années qui ont suivi, ce projet est devenu plus ambitieux, il a mûri, il s'est élargi et il a évolué de façon interne et de façon externe. De façon interne d'abord, l'OMG met en avant

actuellement une architecture basée sur le MOF (Meta-Object Facility), sur une collection de métamodèles, dont UML 2.0 n'est qu'un élément parmi d'autres, mais surtout sur une technologie émergente de transformation de modèles nommée MOF/QVT qui est en cours de standardisation ^[5]. De façon externe ensuite, l'approche MDA™ devient une variante spécifique de ce que l'on nomme aujourd'hui plus généralement l'ingénierie des modèles logiciels (MDE ou Model-Driven Engineering) qui se développe à la fois dans les laboratoires de recherche et chez les industriels impliqués dans les grands projets de développement logiciels. Deux de ces industriels (IBM et Microsoft) ont récemment défini leur stratégie MDE de façon assez convergente. Des initiatives internationales importantes, principalement en Europe et en Amérique du Nord, sont par ailleurs en cours de développement pour mettre à disposition des supports d'interopérabilité en logiciel libre permettant aux outils d'ingénierie des modèles de coopérer.



L'un des exemples typiques est le projet Européen Modelware mis en place à l'été 2004 et visant à construire, notamment autour des standards actuellement reconnus (MOF, XMI, OCL, UML, CWM, etc.), un bus d'interopérabilité sur lequel vont s'échanger non plus du code comme dans le bus intergiciel CORBA, mais des modèles, par exemple sérialisés dans le format XMI. Ce bus d'échange de modèles définira les protocoles et les normes d'interconnexion d'outils pour que ceux-ci puissent interopérer à l'aide d'adaptateurs spécifiques.

En France l'Observatoire Français des Techniques Avancées a très rapidement créé un groupe de travail sur l'ingénierie des modèles logiciels qui a récemment rendu ses conclusions en mai 2004, dans un rapport disponible à l'OFTA ^[4] et qui fait le point sur les avancées de ces technologies.

■ Le MDE et UML

Dans un article récent ^[5], Microsoft ne cache pas que son utilisation d'UML restera probablement essentiellement « contemplative » ou encore documentaire, c'est-à-dire utilisable essentiellement pour produire et communiquer des esquisses entre opérateurs humains. La déclinaison plus solide du MDE chez Microsoft, celle qui s'intègre progressivement dans l'outillage Visual Studio, s'appelle « Software Factories » et est basée essentiellement sur des langages de domaines (DSL ou Domain Specific Languages) de petite taille, facilement manipulables, transformables, combinables, etc. En un mot ces DSL sont la base de l'automatisation du MDE chez Microsoft.

IBM ne dit pas autre chose dans son manifeste MDATM ^[2]. Les trois axes de l'ingénierie des modèles sont, d'après IBM, les standards ouverts, l'automatisation et la représentation directe. Parmi les standards ouverts UML peut avoir sa place mais aux côtés de XML ou d'autres standards. Par contre, ce qui est essentiel, c'est la possibilité de traitement automatique (par exemple tissage, vérification, transformation, etc.) de modèles s'appuyant sur des standards précis, limités en taille et spécialisés. On retrouve encore ici cette idée des DSL. Les DSL peuvent être des petits langages spécifiques de domaines adaptés à des corporations particulières ou à des besoins particuliers. C'est ce que le manifeste IBM appelle la représentation directe, c'est-à-dire la mise à disposition de métiers particuliers ou de tâches spécifiques de langages précis et outillés (éditeurs, générateurs, vérificateurs, etc.).

Il faut séparer clairement les approches MDE du formalisme UML. La portée du MDE est beaucoup plus large que celle d'UML qui reste, dans ses versions 1.5 et 2.0, un standard assez monolithique obtenu par consensus à maxima, dont on doit réduire et préciser la portée à l'aide de mécanismes comme les profils. La pérennité de la version UML 2.0 et des profils associés n'est pas suffisamment établie pour garantir la stabilité des investissements logiciels sur le long terme. La tendance actuelle consiste plus à se baser sur des DSL, concrétisés par des métamodèles spécialisés de petite taille, pour prendre en compte de façon séparée les différents aspects des systèmes logiciels. En procédant ainsi, on pourra

mieux contrôler l'évolution de ces différents aspects, notamment ceux qui sont très liés à l'évolution technologique.

Quel que soit son avenir, le formalisme UML aura été le déclencheur d'une évolution très importante dans la façon de construire et de maintenir les systèmes informatiques. Il a montré que l'on pouvait monter en abstraction en passant de la production et la maintenance manuelle de code à la génération automatique à partir de modèles de fort niveau d'abstraction. Cependant le noyau d'UML (infrastructure) qui est à la base de la définition standardisée de métamodèles restera probablement très utilisé dans les différentes variantes du MDE.

■ La prise en compte des aspects par les modèles

Dans des approches comme la programmation par aspects on essaye de réaliser la synthèse des préoccupations logicielles sur la base du code, par exemple avec des extensions du langage Java comme AspectJ ou HyperJ. Dans les approches MDATM strictes on essaye de réaliser la séparation des aspects liés ou indépendants de la plate-forme par des modèles différents (PIM et PSM). Dans les approches plus générales MDE chaque aspect ou point de vue particulier d'un projet de développement ou de maintenance est pris en compte par un modèle particulier, conforme à un métamodèle précis.

La séparation et le tissage des aspects sont donc au centre des approches MDE. Pour l'OMG ce qui est prioritaire dans le MDATM c'est la séparation des aspects liés ou indépendants de la plate-forme mais pour le MDE de façon plus générale les aspects à prendre en compte sont plus généraux et plus diversifiés et couvrent toutes les préoccupations qui peuvent apparaître lors de la réalisation ou de la maintenance de systèmes à prépondérance logicielle (aspects fonctionnels mais aussi aspects non-fonctionnels comme la qualité de service, la fiabilité, la sécurité, etc.).

■ La transformation de modèles

Une partie de l'activité de l'OMG se concentre actuellement sur la définition d'un langage neutre (c'est-à-dire indépendant des métamodèles) pour réaliser les transformations de modèles ^[6]. La recommandation MOF/QVT spécifiant ce langage devrait être disponible au premier semestre 2005. Mais ceci ne constitue qu'un point de départ et il faudra ensuite démontrer le réalisme de cette solution. Il existe actuellement de nombreuses propositions industrielles de systèmes de transformation de modèles se réclamant du MDATM et utilisant des langages propriétaires. L'intérêt d'une norme est d'espérer voir se développer un marché des composants de transformation où un composant acheté à un fournisseur pourrait s'exécuter sur le moteur de transformation d'un autre fournisseur. L'avenir des approches MDATM semble actuellement plus lié au succès des techniques de transformation automatique de modèles qu'à la diffusion du langage UML.



A titre illustratif citons le langage ATL ^[4] défini par l'équipe INRIA Atlas de Nantes en collaboration avec la société TNI/Valiosys de Brest et avec le soutien des sociétés Microsoft Research et IBM. Il existe plusieurs autres langages de même nature avec des propriétés voisines qui visent à s'aligner sur la norme QVT lorsque celle-ci sera officiellement disponible. Beaucoup de ces langages, et en particulier ATL, seront supportés par des moteurs de transformation disponibles en logiciel libre. La question importante et difficile consiste alors à savoir si, à partir de la disponibilité de ces moteurs, des bibliothèques de composants de transformation vont se développer et être réellement utilisées.

La place centrale de la transformation de modèles se trouve encore renforcée par les initiatives récentes de l'OMG visant à étendre le périmètre de ses activités à la modernisation du logiciel (re/reverse-engineering). Dans l'approche ADM [6], l'objectif n'est plus de générer à partir de PIM vers les plates-formes du futur, mais de pouvoir récupérer des PIMs à partir des plates-formes du passé, basées par exemple sur Cobol, GAP, PL/1, ADA, Pascal, etc. Le groupe de travail qui s'est formé récemment sur ce thème est très actif, essentiellement issu de sociétés qui ont acquis de l'expérience dans des opérations de modernisation comme celle de l'an 2000 et qui veulent aujourd'hui monter en abstraction et faire interopérer leurs différentes solutions propriétaires. La reconnaissance que le filtrage des parties significatives (candidates à la pérennité) d'un existant (legacy) peut se faire à partir de DSL et de transformations de modèles exprimées dans des langages de type MOF/QVT comme ATL a permis une progression très rapide de cette approche ADM.

■ En résumé

La technologie des modèles s'appuie sur quelques constats :

- Le coût de migration des systèmes informatiques d'une plate-forme à une nouvelle plate-forme technologique (middleware) devient rédhibitoire, surtout si la fréquence d'apparition de ces nouvelles plates-formes s'accroît (COM/DCOM, CORBA, EJB, HTML/HTTP, XML/SOAP, C#/DotNet, etc.). Il devient de plus en plus difficile de justifier ces coûts importants auprès des utilisateurs finaux par le simple constat de la mise sur le marché d'une nouvelle technologie un peu plus performante, sans réel retour sur investissement pour les décideurs de l'entreprise.
- La seule solution possible consiste à découpler encore plus les parties "métier" et les parties "technologiques" dans les systèmes. Ceci revient à monter en abstraction et à isoler le plus possible les modèles métiers de l'obsolescence technologique. La façon naturelle de travailler consistera à utiliser au maximum la génération paramétrique semi-automatique à partir de modèles de fort niveau d'abstraction vers des plates-formes différenciées. Les fournisseurs de nouvelles plates-formes (Microsoft, Sun, etc.) seront incités à prévoir également les outils d'assistance à la génération vers leur propre cible technologique. Un marché de

L'OMG donne une définition précise et concise du MDA

A sa dernière réunion de Montréal, Canada, l'OMG a voté le 26 août 2004 une définition officielle du MDA : « MDA is an OMG initiative that proposes to define a set of non-proprietary standards that will specify interoperable technologies with which to realize model-driven development with automated transformations. Not all of these technologies will directly concern the transformations involved in MDA.

MDA does not necessarily rely on the UML, but, as a specialized kind of MDD (Model Driven Development), MDA necessarily involves the use of model(s) in development, which entails that at least one modeling language must be used. Any modeling language used in MDA must be described in terms of the MOF language, to enable the metadata to be understood in a standard manner, which is a precondition for any ability to perform automated transformations. »

L'outillage de transformation de modèles devrait se développer dans les prochaines années, favorisé par des organismes comme l'OMG.

- Le code exécutable ne représente plus le référentiel unique d'informations pour le cycle de développement. De nombreux modèles (de métier, de test, d'architecture, de déploiement, etc.) existent, sont développés, documentés et maintenus en dehors du code. Code et modèles sont synchronisés à partir de techniques diverses, essentiellement basées sur des opérations de transformation.
- Chaque modèle prend en compte de façon séparée les aspects différents du développement logiciel. La séparation des aspects est nécessaire et beaucoup mieux supportée par un système à modèles séparés que par une projection canonique sur le code comme dans la programmation par aspects.
- Le statut des modèles évolue d'une phase contemplative à une phase productive. Jusqu'à présent en effet, un modèle était souvent construit pour être observé ("contemplé") par un opérateur humain qui en dérivait ("synthétisait") une implémentation exécutable. Aujourd'hui les modèles peuvent être traités automatiquement par des ordinateurs et on peut leur appliquer des transformations formelles. Des systèmes homogènes de représentation de modèles sont apparus comme XMI (XML Model Interchange).
- Le nombre de modèles devient important et leur niveau d'abstraction augmente. On voit apparaître des modèles prenant en compte des aspects du développement logiciel que l'on ne savait pas très bien gérer automatiquement comme les aspects non fonctionnels, la QoS, etc.



- La multiplicité des modèles, basés sur des DSL précis et bien délimités, amène à développer des ateliers où il sera possible de travailler conjointement avec plusieurs modèles. Les outils correspondants pourront être générés de façon semi-automatique. L'expérience des années 90 sur les outils CASE ou meta-CASE est réutilisée ici, mais avec de nombreux plus, notamment sur les caractéristiques normatives (XML et MOF essentiellement).
- Chaque modèle voit sa « sémantique » spécifiée par un métamodèle (correspondant à une grammaire formelle ou une ontologie au sens moderne). Chaque vue spécialisée est donc définie par un métamodèle précisant l'ensemble des aspects pris en charge. Un métamodèle permet de caractériser un DSL.
- Il existe par ailleurs un langage d'écriture de métamodèles (un métamétamodèle). Le MOF remplit cette fonction à l'intérieur du périmètre des activités de l'OMG, c'est-à-dire du MDA™.

ment assez radical dans les pratiques du génie logiciel et un défi pour les vingt années qui viennent.

La popularité de l'approche MDA™ de l'OMG a démontré la nécessité d'un changement profond de ces pratiques de développement de logiciel si l'on veut pouvoir répondre dans ces vingt prochaines années à une forte augmentation de la demande en volume et en qualité, avec un nombre global d'informaticiens en relative stabilité. Cependant l'idée qu'il suffirait d'utiliser UML pour pouvoir exprimer des modèles métier neutres (PIM) et ensuite les lier à différentes plates-formes d'exécution (PSM) est aujourd'hui considérée comme un peu simpliste. La complexité du formalisme UML, son manque de définition rigoureuse pour plusieurs concepts et les fortes limitations de son formalisme d'extension par profils ont conduit plusieurs équipes de recherche et de nombreux industriels à s'appuyer sur des DSL bien délimités pour prendre en compte les différents aspects du développement logiciel. Bien évidemment la définition du formalisme UML par l'OMG aura été le point de départ de ces approches d'ingénierie des modèles qui vont modifier profondément les pratiques du développement logiciel. ■

■ Conclusions

La recherche de plates-formes middleware pour l'intégration de systèmes distribués a constitué un objectif important dans les années 1980. Pour répondre aux nouvelles problématiques de complexité, d'évolutivité et d'hétérogénéité des systèmes il semble nécessaire aujourd'hui de recourir à des méthodes assez nouvelles basées sur des techniques de nature générative organisées autour de nombreux DSL s'appuyant sur des métamodèles précis. Le développement des approches MDA™/MDE depuis les années 2000 constitue un change-

Jean BEZIVIN
Equipe ATLAS (INRIA & LINA)
Université de Nantes



Références Bibliographiques

- [1] ATL, ATLAS Transformation Language Reference
<http://www.sciences.univ-nantes.fr/lina/atl/>
- [2] Booch G., Brown A., Iyengar S., Rumbaugh J., Selic B. The IBM MDA Manifesto The MDA Journal, May 2004,
[http://www.bptrends.com/publicationfiles/05-04 % 20COL % 20IBM % 20Manifesto % 20- % 20Frankel % 20-3.pdf](http://www.bptrends.com/publicationfiles/05-04%20COL%20IBM%20Manifesto%20-%20Frankel%20-3.pdf)
- [3] Cook, S. Domain Specific Modeling and Model Driven Architecture The MDA Journal, January 2004,
[http://www.bptrends.com/publicationfiles/01 % 2D04 % 20COL % 20Dom % 20Spec % 20Modeling % 20Frankel % 2DCook % 2Epdf](http://www.bptrends.com/publicationfiles/01%2004%20COL%20Dom%20Spec%20Modeling%20Frankel%20Cook%20.pdf)
- [4] OFTA Rapport du groupe de travail Ingénierie des modèles : logiciels et systèmes Rapport Arago #30 ISBN 2-906028-16-9, Disponible à l'Observatoire Français des Techniques Avancées, OFTA - 5, rue Descartes - 75005 PARIS
<http://perso.wanadoo.fr/ofta/>
- [5] OMG The MOF/QVT Queries, Views, Transformations request for proposal
<http://www.omg.org>
- [6] Ulrich, W. A status on OMG Architecture-Driven Modernization Task Force.
[http://adm.omg.org/MELS_EDOC2004_Ulrich_Extended_Submission\(pdf\).pdf](http://adm.omg.org/MELS_EDOC2004_Ulrich_Extended_Submission(pdf).pdf)