



MDD est plutôt un concept

Questions à Grégory Weinbach (Responsable pôle MDA, Objet Direct)

Programmez ! : On entend beaucoup parler de MD comme le MDT, MDD ou encore MDA. Quel est aujourd'hui le MD le plus en vogue et pourquoi ?

Grégory Weinbach : MDA, MDD, MDT, MDE... il est difficile d'établir une hiérarchie de "popularité". MDD est plutôt un concept, finalement assez vague : on produit des modèles plutôt que du code, le code étant généré à partir de ces modèles. Leur contenu n'est pas précisé.

En ce sens, beaucoup de gens pratiquent ou ont pratiqué ! le MDD : générer des scripts SQL à partir d'un MCD, utiliser un AGL Cobol à la Pacbase ou faire de la génération de code à partir de schémas XML... c'est du MDD ! Et finalement, MDA c'est du MDD "normalisé". Il suppose un enchaînement de transformations types qui vont d'un modèle, souvent exprimé en UML, jusqu'au code. Les modèles intermédiaires sont identifiés, PIM et PSM, même si leur contenu précis n'est pas défini par la norme. En particulier, le PIM est une notion mal comprise et donc, sujette à interprétation : il est très rarement, vraiment "platform independent" et joue mal son rôle de

pivot. Les outils permettant de faire de la transformation de modèle se disent "MDA compliant" mais ils ne proposent jamais sur étagère un ensemble de méta-modèles et de transformateurs/générateurs permettant effectivement de mettre en place un processus MDA. Le MDE est la discipline de l'ingénierie logicielle qui regroupe tout ce qui a trait au MD. Dès qu'on parle de Model Driven on est dans le MDE. Dès qu'on a affaire à une transformation d'un modèle en code, c'est du MDD, si le processus et/ou l'outillage se réfère(nt) à la norme on est dans le MDA.

P ! : L'un des soucis est de préserver la synchronisation bidirectionnelle entre le modèle et ce qui est généré. En MDA, le round trip sur des projets complexes est au mieux médiocre, au pire, quasi impossible à maintenir dans les cas extrêmes. La situation va-t-elle changer dans l'avenir ?

G.W. : Une des premières (et légitimes) questions posées par les gens à qui on présente un outillage MDA est : que se passe-t-il si je modifie le code généré ? Si certains outils MDA excellent dans le round-trip PSM

<-> code, c'est souvent au prix d'un code illisible : des commentaires "balises" parsement le code et ne doivent en aucun cas être modifiés sous peine de perdre la belle mécanique, "génération, modification manuelle, régénération...".

Dans tous les cas, dans une approche MDA, le source véritable étant le PIM modélisé en UML, ce round-trip n'a d'intérêt que s'il permet aussi la mise à jour du PIM à partir des modifications du PSM (PIM <-> PSM). Ce n'est presque jamais le cas, même quand l'outil le permet, ce qui est rare, peu de gens prennent la peine d'écrire les transformations inverses qui sont souvent très complexes. Je pense qu'il faut privilégier une approche pragmatique sans round-trip.

P ! : Et la notion de qualité dans tout cela ?

G.W. : C'est un problème difficile. Intrinsèquement, la qualité du code produit dans une approche MD est meilleure : conformité aux normes, à l'architecture, aux bonnes pratiques, homogénéité. C'est d'abord la plate-forme de génération qu'il faut tester de manière fine. Il reste les problèmes de conformité aux exigences fonctionnelles. Les systèmes d'automatisation des tests fonctionnels apportent de bonnes réponses.

Pour aller plus loin : <http://mdblog.fr/>

AVIS D'EXPERT



Si la démarche MDA est respectée, il n'y a aucune raison valable de désynchronisation

Nous avons posé quelques questions à Guy Cartigny

(Responsable technique développement, Compuware)

Programmez : la synchronisation reste un problème majeur dans le développement MD. Qu'en pensez-vous ?

Guy Cartigny : Vrai et faux : Si la démarche MDA est respectée, il n'y a aucune raison valable de désynchronisation. Ce n'est que la conséquence de solutions incomplètes et de développeurs qui ne sont pas prêts à abandonner leurs lignes de code.

Il existe aujourd'hui des solutions qui garantissent la synchronisation bidirectionnelle.

P ! : La profusion de Model Driven ne nuit-elle pas à sa compréhension ? Comment sensibiliser le développeur, l'architecte à cela ?

G.C. : Catégoriquement : Oui, elle y nuit ! Et il sera toujours très difficile de sensibiliser les développeurs. L'approche par les modèles, et quel que soit le nom utilisé : MDA, MDD, MDE, DSL, est synonyme de la fin du développeur. A terme, celui-ci devra synthétiser l'expression des besoins utilisateurs sous forme de modèle, puis géné-

rer l'application, effectuer les tests de performance et en assurer la maintenance fonctionnelle ! Ce qui, reconnaissons le, n'est pas forcément dans ses aspirations. L'architecte a comme intérêt, par rapport à l'ingénierie des modèles, le fait de n'avoir qu'à concevoir, valider la ou les architectures optimales en fonction de la nature des applications qui seront développées. Aujourd'hui, une bonne partie de son temps est consacrée à expliquer l'utilisation de l'architecture retenue, à en vérifier le respect, voire à aider à la bonne fin du projet, et là encore, ce n'est pas forcément dans ses aspirations !

AVIS D'EXPERT