

Compte rendu Conférence Models 2006

Introduction

La conférence models (anciennement UML) est un lieu à la fois technique (UML, EMF, MOF) et théorique (Sémantique, approche formelle, langages) mais est, surtout, un lieu pluricommunautaire. La conférence a été marquée par 2 initiatives qui vont dans le sens du regroupement des différentes communautés de recherche en informatique autour de l'Ingénierie Dirigée par les Modèles. Une réunion a été spécialement mise en place pour couvrir l'aspect « *respository of metamodel* ». Les divers ateliers présentés à la conférence montrent également l'hétérogénéité des participants à la conférence Models.

1 Ateliers

1.1 ATEM (MetaModels, Schemas, Grammars and Ontologies for Reverse Engineering). Favre, Winters, Gasevic, Lämmel (France, Canada, USA, Germany)

Cet Atelier a pour but de regrouper plusieurs communautés de recherche autour du reverse engineering ainsi que de montrer les équivalence entre les notions de metamodèles, schémas, grammaire et ontologies. Parmi les différents articles présentés dans cet atelier, il y avait notamment du reverse-engineering d'interface homme machine (IHM) en Java/Swing, l'utilisation d'ontologies pour analyser de la documentation logicielle. L'atelier s'est conclu sur une mise en commun des résultats de ces différentes communautés de recherche : vers une cartographie des modèles et transformations entre ces formes de metamodèles.

1.2 Model Driven Development of Advanced User Interfaces (MDDAUI).

Bödcher, Hussmann, Pleuss, Sauer, Van den Bergh (Germany, Netherlands)

La conception et le développement des interfaces homme machines (IHM) ayant une tradition forte en terme de modèles depuis les années 80, cet atelier s'inscrit dans la suite de ces travaux. Cependant, la prise en compte des concepts de l'IDM est importante pour une réflexion plus approfondie sur l'utilisation des modèles en particulier lors de la génération automatique de code. L'accent est surtout mis sur la génération automatique : a la conception ou alors lors de l'exécution (modulo une infrastructure logicielle adéquate). Un papier fait notamment le parallèle entre les méthodologies de conception d'IHM et l'IDM. Enfin, un papier s'intéresse à la mesure de la qualité logicielle spécifique aux IHM lors de la conception et la génération de ces interfaces.

1.3 Models at Run-time. Becomo, Blair, France (UK, USA)

La principale préoccupation de ces ateliers est de montrer l'utilisation possible des modèles pour réaliser les phases d'exécution d'un système : une question se pose alors « *Models at run-time or models for run-time* » ! Les présentations étaient accés sur les systèmes adaptatifs et adaptables pour lesquels la planification et l'exécution de l'adaptation est critique est complexe. L'idée étant que, passant par des modèles la complexité de la mise en oeuvre de l'adaptation serait réduite et plus facile à concevoir. Une présentation très intéressante fait cas de la mise en oeuvre « *l'autonomous computing* » sous les recommandations de l'OMG. Cette présentation conclue a une infrastructure a modèle pour couvrir les phases de planification, de mise en oeuvre et d'analyse de l'adaptation a fournir pour un système adaptable. D'autre papier présentent une approche spécifique à un domaine d'application pour lesquels ont établis des langages spécifiques au domaine dans le cadre de l'adaptation ; l'adaptation étant vue ici comme une transformation de modèle. Un problème se pose cependant quant au temps de réponses de

tels système : si la complexité de mise en œuvre est réduite pour le concepteur, elle ne l'est pas forcément pour la machine.

2 Keynote

Titre: Immune System Computation and the Immunological Homunculus

Irun R. Cohen

Cette présentation fait une mise en correspondance entre l'**IDM** et les réalités de la biologie. L'émergence de la vie étant vue comme un objet discret : la structure de l'organisme, des organes est donnée par un modèle (l'ADN). La création de la vie qui génère une nouvelle structure avec des caractéristiques différentes ainsi que l'évolution font un parallèle fort entre modèles et transformation et ce qui nous caractérise biologiquement. L'homme possède un modèle de son propre corps que l'on appelle l'homoncule : c'est le point de vue du cerveau sur l'organisme. Ce modèle, constitué d'un grand nombre de paramètres permet au système immunitaire de centrer les points importants du corps humain. C'est un modèle qui est hérité lors de la naissance et se construit au fur et à mesure des années. Le système immunitaire se sert ainsi de son modèle pour donner des réponses aux agressions, se faire passer des informations quant à ces agressions. Ceci dit les biologistes ne connaissent pas complètement le modèle du système immunitaires, ils pensent à termes pour le comprendre en établir un metamodèles en quelque sorte.

3 Evaluer UML

An Experimental Investigation of UML Modeling Conventions Lange, Du Bois, Chaudron, Demeyer

C'est une étude qui a pour objectif d'évaluer l'efficacité des conventions de modélisation, que ce soit syntaxique, sémantique ou pragmatique. Cette étude expérimentale porte sur 3 groupes : le premier modélise avec **UML** sans aucune convention (témoins), le deuxième utilise seulement une convention et le troisième groupe profite en plus des conventions d'outils (vérification validation). Les résultats montrent que la qualité de la modélisation est meilleure avec les outils et les conventions mais cela consomme beaucoup de temps. Le groupe avec les outils a perdu en outre beaucoup de temps dans l'apprentissage de l'outil. La remarque finale est que le plaisir de développer est moindre lors de l'utilisation de convention. Cette étude mériterait un affinement des mesures quantitatives et un échantillon d'individu beaucoup plus large, pour distinguer plus clairement les différentes tendances.

Improving the Definition of UML

O'Keefe Australia

La sémantique d'**UML** n'est pas quelque chose de définit. C'est un vaste sujet de débat mais aucun accord n'est trouvé. L'auteur de cet article cherche à donner un coter plus formelle à **UML** pour en améliorer sa compréhensibilité et permettre une validation automatique sa sémantique. Partant du principe qu'il est difficile parfois de se mettre d'accord sur le sens que véhicule un diagramme de classe, l'auteur espère que l'expression d'une sémantique pour **UML** apportera un plus. L'auteur identifie 2 voies à suivre : la définition formelle en mathématique et la psychologie cognitive nécessaire à l'appréhension par l'humain des diagrammes. L'auteur propose d'utiliser la logique dynamique pour

vérifier la satisfiabilité de diagrammes de séquence. Des discussions suivent sur l'utilité de la sémantique pour **UML** soulignant l'intérêt du côté informelle d'**UML** et de sa flexibilité pour s'accorder avec les différents métiers de l'informatique.

4 Panel: Is Standardization of Model Driven Technologies Harming or Helping the Field

Ce panel traite du rôle de **MDE** en tant que standard pour aider à « mettre ensemble les communautés de l'informatique ». Seulement d'un point de vue de la recherche la standardisation est la pire des solutions. Les participants de ce panel ont montré une rapide évolution des langages standard du point de vue de l'objet puis des modèles : C / C++,SDL , **UML SVG**, **QVT** avec **MTL** , ... Cependant si les standards sont parfois mal perçus ils aident à : avoir un environnement stable, à avoir une base solide pour l'enseignement. D'autant plus que certains standards servent aussi à déclencher des réactions et faire jaillir de nouvelles technologies : **MOF** vers **EMF**, etc. Ces standards peuvent provenir de plusieurs sources : soit d'un outil duquel émerge le standard (bonne utilisabilité mais ambiguïtés possibles), les standards artificiels souvent développés par des petits groupes d'utilisateur sans véritable implémentations. Enfin les standards commerciaux qui sont basés sur une implémentation (mais ne couvre souvent que les préoccupations de la compagnie qui l'émet). Le **MDE** ne dérogera pas à la règle de standard aidant ainsi à mieux comprendre et à accéder aux bénéfices qu'il apporte par le biais de standards commerciaux (**EMF**) mais aussi des standards un peu plus consensuels (**MOF**). Il est en tout cas nécessaire pour rassembler les communautés de l'informatique de promouvoir des points d'appuis qui sont ces standards tout en gardant un œil ouvert aux nouvelles préoccupations requises par ces métiers.

5 Applying UML to Interaction and Coordination

Composition MDA Van Gool, Punter, Hamilton, Van Engelen

Ce papier traite de la coordination des aspects lors de la génération de code pour un scanner particulier (système embarqué) dont la partie logicielle représentée environ 30 millions de lignes de code. Ce papier présente donc la coordination d'un ensemble de transformations de modèle mise en composition. L'exemple portant sur ce scanner, montre l'utilisation de divers modèles en entrée (certains provenant de rétro-conception) mais on note en particulier l'utilisation d'un modèle de l'activité édité par les ingénieurs (ici la seule partie variable de la conception). Les ingénieurs définissent ensuite un ensemble de contraintes pour coordonner la génération des modèles spécifiques qui donneront par une seconde transformation le code. Un point est à creuser cependant : la validation et le test du diagramme d'activité.

CUP2.0: High-Level Modeling of Context-Sensitive Interactive application Van den Bergh, Coninx

Cette présentation traite principalement du problème en *interaction personne système* dans le cadre de l'informatique ambiante. Il s'agit de décomposer l'interface en un ensemble de modèles ainsi que de modéliser le contexte de l'interaction. Ce contexte est capté par des « context collector » qui captent à l'exécution par exemple: la position de l'utilisateur, la luminosité, le dispositif (téléphone, pc,...) sur

lequel s'exécute l'IHM (interface homme machine). L'IHM est déployé dynamiquement pour un contexte donnée selon un modèle de l'activité utilisateur : seules les tâches courantes de l'utilisateur sont transformés et exécutés sur la plateforme cible.

6 Model Integration

Detecting and Resolving Model Inconsistencies Using Transformation Dependency

Analysis Mens, Van Der Straeten, D'Hondt Belgium, France

Les auteurs ont implémenté une solution qui permet de détecter des incohérences dans des diagrammes UML et de résoudre ces incohérences par des règles de transformation sur des graphes. L'analyse des dépendances entre les graphes leur permet d'identifier les points d'incohérences.

Merging Models with the Epsilon Merging Language (EML) Kolovos, Paige, Polack UK

Ce papier traite d'un DSL pour assembler, fusionner des modèles (EMF) basé sur Epsilon : une plateforme extensible pour spécifier l'intégration et la gestion des modèles. D'abord les modèles sont comparés puis vérifiés en terme de conformité et enfin les modèles sont restructurés pour leur fusion. Il est à noter que cette préoccupation est sensiblement proche des approches de fusion d'ontologies et soulève par conséquent le problème de comparaison entre les modèles à fusionner.

7 Analysing dynamic models

Analysis and Visualization of Behavioral Dependencies among Distributed Objects based on UML Models Garousi, Briand, Labiche Canada, Norway

Ce papier montre au travers de graphes hiérarchique (ou nestés) les relations de dépendances entre les objets. Ces relations sont tracées par le nombre de messages synchrones envoyés d'un objet à l'autre. Ce graphe prend aussi en compte les problèmes de dépendances transitives entre objets ainsi que l'importance des messages (criticité, message = passage d'objets). Ce graphe permet une visualisation de la charge que porte chaque objet en terme de communication et de penser une re-conception en visant à éliminer les redondances et décharger certains objets. Cette vision orientée modèle s'appuie sur le paradigme objet et reste donc assez asymétrique ; il faudrait par conséquent revoir cette approche dans le cadre des applications orientées service qui prennent en considérations les services fournis et requis par les objets qui les composent.

Dynamic and Generic Manipulation of Models: from Introspection to Scripting

Tombelle, Vanwormhoudt France

Les auteurs mettent en avant la manipulation de modèle par langages de script pour permettre en particulier une flexibilité de mise à jour lors de l'exécution. Ils illustrent une manipulation direct d'EMF au travers de Java-Script : chaque éléments d'EMF est vu comme un objet Java-Script. Cette

manipulation par script permet en outre l'intégration des événements mais surtout la manipulation à chaque niveau de modélisation : metamodelle, modèle, ...

8 Panels: Integrating Model-Driven Technology into the Computer Science Curricula

Ce panel traite de l'intégration dans le l'enseignement de l'informatique en général par l'**IDM**. En effet, comme pour beaucoup de science on se sert de modèle pour expliquer la réalité des choses : un modèle mathématique pour un phénomène physique. L'apport de l'**IDM** par exemple pour l'apprentissage des bases de données paraît assez direct à partir du moment où l'on a montré la correspondance entre diagramme de classe et une BD relationnelle. Mais l'apprentissage par l'**IDM** va outre cette simple mise en correspondance, elle peut amener les étudiants à extraire les connaissances des matières qu'ils apprennent par le biais de la meta-modélisation et ainsi de poser des points comparatifs entre les technologies étudiées.

9 Risk, Trust and Dependability

A Graphical Approach to Risk Identification, Motivated by Empirical Investigations

Hogganvik, Stølen Norway

Les auteurs présentent une évaluation des risques basée sur des modèles et généralisent la modélisation de systèmes « standard » en incluant les comportements indésirables. La plateforme Coras (disponible sur sourceforge) se base sur **UML** pour décrire les systèmes, en incluant les risques. Les risques sont évalués par les utilisateurs comme « faible, moyen ou élevé ». Chaque risque est associé à une conséquence et à un coût. Les auteurs évitent l'usage des probabilités dans l'évaluation de la fréquence d'un risque, car dans la pratique, les utilisateurs les évaluent mal.

Model-Driven Assessment of Use Cases for Dependable Systems Mustafiz, Sun, Kienzle, Vangheluwe Canada

Les auteurs présentent une approche par les modèles pour analyser les dépendances entre cas d'utilisation pour trouver la meilleure méthode dans la conception de l'interaction avec le système dans son environnement. Pour ce faire, les auteurs utilisent une extension des machines à état probabilistes pour modéliser les requis systèmes. Ces machines à état sont visualisées dans ATOM 3, un outil multi-metamodelles, pour en faire l'analyse statistique.

10 Tool Environments

Using Smalltalk as a Reflective Executable Meta-Language: An Experience Report Ducasse, Girba Switzerland

Les auteurs décrivent un environnement de reverse engineering, Moose, écrit en Smalltalk. Ils utilisent notamment la réflectivité de SmallTalk pour écrire le modèle qui correspond au code en même temps

que celui-ci est écrit. Le système d'analyse de code est donc lui-même écrit en smalltalk ce qui permet au développeur smalltalk de facile mettre à jour et personnaliser son environnement (via une api). Ici ils utilisent seulement smalltalk pour toute la description et manipulation du modèle ce qui n'est pas le cas avec les systèmes actuels (MOF + java)

UML model interchange in heterogeneous tool environments: an analysis of adoptions of XMI 2 Lundell, Lings, Persson, Mattsson Sweden

Les auteurs présentent une étude de compatibilité entre différents outils de modélisation a propos du *XML metadata Interchange (XMI)*, norme de l'OMG, et mettent en avant les difficultés pour passer du XMI 1 au XMI 2, ainsi que les problèmes de compatibilités entre outils d'entreprises différentes. Actuellement le standard XMI est bien trop complexe et permet a chacun de développeur d'outils de mettre en place son propre XMI.

11 Roundtrip Engineering

Layered Class Diagrams: Supporting the Design Process Hendrickson, Jett, Van Der Hoek USA

Le problème posé ici est la gestion de version entre modèles lors de conception itérative. Ce papier fait appel à une série de couche pour traiter ces versions qui pourront ensuite être mise en commun. D'un point de vue de l'ingénierie cela permet d'explorer plus facilement les alternatives possibles de modélisation (en phase de conception).

Enabling a Roundtrip Engineering Process for the Modeling and Analysis of Embedded Systems Goldsby, Cheng, Konrad, Kamdoum USA

On utilise de plus en plus des langages de modélisation dans les systèmes critiques : l'hypothèse étant qu'une majorité de code généré peut réduire les erreurs induites lors de la production. Certaines entreprises utilisent de plus en plus UML. UML doit être traduit vers des langages de spécifications formels pour être analysés par des vérificateurs (comme pour le langage B par exemple). La principale difficulté réside dans l'appréhension des modèles formel : en particulier lorsqu'il s'agit de les interpréter. Ce papier traite par conséquent d'un environnement pour visualiser les contraintes et formalisation introduite par la formalisation d'UML.

Conclusion

La conférence Models devient sûrement de plus en plus pluridisciplinaire. La grande diversité des ateliers proposés, mais aussi de l'origine des participants, montre bien que l'IDM offre un cadre intégratif pour tous les domaines de l'informatique. Le fait que 2 initiatives : ReMODD (et le Zoo of metamodel soit clairement orienté vers « un repository de metamodel » permettra d'offrir une meilleure interaction entre les divers communautés de recherche. Il en est de même pour l'enseignement de l'informatique, d'ailleurs un panel y a été consacré : la démarche de metamodélisation apporte tant un outil réflexion que de comparaison entre les technologies étudiées.