

Compte rendu de la conférence Models2006

Florian Prud'homme, I3S CNRS

1. Intérêt pour mon travail

La conférence Models2006 m'a permis d'approfondir la notion de modèle, l'UML, et les différents degrés de formalisation en fonction du langage de modélisation. J'ai vu une approche intéressante bien que limitée, qui relie modèles avec la programmation par aspect, qui permet la séparation des préoccupations.

J'ai vu ensuite des méthodes de visualisation dans le domaine de la sécurité, et la fiabilité des systèmes, avec un aperçu des vérifications formelles.

Beaucoup d'efforts sont fait pour rendre naturel aux utilisateurs la représentation des connaissances sous forme de modèles standardisés.

J'ai découvert également certains aspects sur les difficultés pratiques et le problème de validation à l'utilisation, ainsi que des outils d'éditeurs et de transformation de modèles.

Tout ceci rentre dans le cadre de ma thèse et m'aidera donc dans l'établissement de mon état de l'art.

2. Panels

1.1. Keynote

Titre: Immune System Computation and the Immunological Homunculus

Irun R. Cohen

Cohen explique que le système immunitaire de l'homme possède un modèle moléculaire du corps humain, qu'il appelle l'homunculus (petit homme en latin) immunologique. Ce modèle est constitué de plusieurs centaines de paramètres, et permet au système immunitaire d'appréhender efficacement la complexité d'un corps humain. Ce modèle est en partie hérité de la mère, l'autre partie est généré progressivement à partir de la naissance. L'assistance a soulevé une question : pourquoi le système immunitaire se trompe-t-il parfois, provoquant par exemple de l'arthrite. Cohen affirme que c'est la question du XXIe siècle. Pour y répondre, il faudra que les biologistes affinent leur modèle du système immunitaire, c'est-à-dire, décrire un méta-modèle de l'homunculus.

1.2. Evaluer UML

La présentation de "Uses and abuses of the stereotype mechanism in UML1.x and 2.0" a été annulé.

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. Les résultats sont difficilement appréciables, car ils reposent sur un test sur 27 étudiants répartis en 3 groupes : le premier sert de témoin, et modélise un cas en UML sans convention de modélisation, le deuxième doit modéliser ce même cas en UML en utilisant les conventions, et le troisième doit modéliser en utilisant les conventions et des outils de vérification. Les résultats ne sont pas significatifs, et ne montre pas un intérêt particulier à utiliser les conventions de modélisation, excepté pour la syntaxe. Il est à noter que ces conventions ont été sélectionnées à partir de la littérature, en fonction de leur pertinence pour le projet d'étudiant. Les auteurs ont essayé d'extraire des données quantitatives à propos de la qualité représentative des modèles UML. Pour essayer de normaliser les mesures, les auteurs divisent la quantité d'erreurs trouvées par des outils automatiques, par le nombre de classes contenues dans le modèle. 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

L'auteur cherche à améliorer la précision d'UML et sa compréhensibilité suivant plusieurs critères. Ces critères sont la possibilité de vérification automatique de la sémantique, ou encore la possibilité de s'accorder sur des idées complexes.

Il n'y a pas d'accord parfait sur la signification d'un diagramme UML. L'auteur prétend qu'il faut donc allier la précision des mathématiques, avec de la psychologie cognitive, afin de rendre facilement compréhensible les diagrammes. L'auteur propose d'utiliser la logique dynamique pour vérifier la satisfaisabilité de diagrammes de séquence.

Un intervenant dans la salle, membre de l'OMG réagit en expliquant que les industriels ne cherchent pas à avoir une définition formelle de la sémantique d'UML. Le côté informel d'UML est peut-être une raison de son succès. L'auteur répond que l'on peut simplifier sa sémantique « universelle » en fonction des domaines des industriels.

1.3. Programmation par aspects

A Slice of MDE with AOP: Transforming High-Level Business Rules to Aspects

Cibrán, D'Hondt Belgium, France

Les logiciels conçus pour un domaine particulier consistent en un noyau de fonctionnalités techniques et des règles métiers. La logique des règles métiers a tendance à évoluer rapidement, et il faut donc séparer les règles du noyau de l'application. L'auteur démontre que seule la programmation par aspect permet de séparer efficacement les règles métiers du cœur de l'application. Une solution a été programmée en utilisant AspectJ. Un effort tout particulier a été porté sur l'ordonnancement des aspects, et la gestion des conflits.

Domain Models are NOT Aspect Free

Rashid, Moreira UK, Portugal

Cette présentation est une réponse très critique d'un article présenté à Models2005, et qui démontre point par point l'inconsistance de l'affirmation : "Models are Aspect Free". L'auteur reprend chaque point de l'article cité, pour le critiquer en détail. Finalement, on se rend compte que le problème principal consiste à définir qu'est-ce que l'on entend par « aspect ».

1.4. 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 inconsistances dans des diagrammes UML et de résoudre ces inconsistances par des règles de transformation sur des graphes et en analysant les dépendances séquentielles.

Merging Models with the Epsilon Merging Language (EML)

Kolovos, Paige, Polack UK

L'auteur propose un nouveau langage (EML) qui est centré sur la fusion de modèle. Il est basé sur Epsilon (*Extensible Platform for Specification of Integrated Languages for mOdel maNagement*). La fusion est séparée en plusieurs étapes : la phase de comparaison, de vérification de conformité, de fusion proprement dite, et ensuite celle de restructuration.

Package Merge in UML 2: Practice vs. Theory?

Zito, Diskin, Dingel Canada

L'auteur expose une formalisation de la notion de fusion de package présente dans la norme UML 2, ainsi qu'une classification très générale des mécanismes d'extension de modèle.

1.5. Analyzing Dynamic Models

Chair: **Susanne Graf** | 5-Oct 9:00 - 10:30

Analysis and Visualization of Behavioral Dependencies among Distributed Objects based on UML Models

Garousi, Briand, Labiche Canada, Norway

L'auteur introduit différentes métriques basées sur le nombre et le type de messages synchrones échangés pour mesurer la dépendance d'un objet par rapport aux autres objets du système. A partir de ces métriques, l'auteur construit des graphes hiérarchiques d'objets et de liaisons

entre objets, ce qui permet ensuite aux ingénieurs d'améliorer le système en introduisant de la redondance, ou en refondant les classes pour mieux répartir les charges. Les dépendances sont asymétriques, car on considère une notion de service requis/fourni.

Model Extraction Using Context Information Duarte, Kramer, Uchitel UK

Les auteurs proposent une approche "boîte blanche" et propose d'allier l'analyse statique avec une analyse dynamique (en faisant plusieurs essais d'exécution) pour modéliser les différents états atteignable par le programme.

Dynamic and Generic Manipulation of Models: from Inspection to Scripting

Tombelle, Vanwormhoudt France

Les auteurs mettent en évidence l'intérêt qu'il y a à manipuler des modèles avec un langage de script. Ils illustrent ces idées à l'aide de JavaScript appliqué au framework de modélisation EMF (Eclipse Modeling Framework) alternative au MOF.

1.6. MOF

Chair: **Pierre-Alain Muller** | 5-Oct 11:00 - 12:30

A Metamodeling Approach to Pattern Specification Elaasar, Briand, Labiche Canada, Norway

Ce travail est issu d'une collaboration entre IBM et l'université de Carleton d'Ottawa. Les auteurs présentent le Pattern Modeling Framework (PMF), qui permet de modéliser des spécifications de pattern et la détection de langages et de structures de modélisation compatible avec MOF. PMF fournit le langage Epattern, qui permet de spécifier précisément les patterns dans des métamodèles compatibles avec MOF. Les auteurs présentent ensuite une méthode systématique pour dériver un algorithme de détection à partir d'une spécification Epattern.

1.7. Risk, Trust and Dependability

Chair: **Antonia Bertolino** | 5-Oct 16:00 - 17:30

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 cout. 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.

Reusable MDA Components: a Testing-for-Trust Approach

Mottu, Baudry, Le Traon France

Les auteurs introduisent expérimentalement des erreurs dans le code, et vérifient ensuite si les contrats permettent de détecter les erreurs introduites. Ainsi, on peut mesurer un degré de confiance dans des composants MDA.

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 systèmes, et trouver des moyens plus fiables pour la conception des interactions entre le système et l'environnement.

1.8. Tool Environments

Chair: Paul Baker | 6-Oct 9:00 - 10:30

Applying Model Fragment Copy-Restore to Build an Open and Distributed MDA Environment

Sriplakich, Blanc, Gervais France

L'auteur propose une solution qui permet d'économiser de la bande passante lors de la modification d'un modèle à travers un système distribué. Les outils de modélisation possèdent chacun une copie du modèle. Lorsqu'un outil modifie ce modèle, seul un fragment du nouveau graphe est transmis aux autres éléments du système. Cette solution a pour objectif futur de permettre des modifications d'un modèle par plusieurs outils en parallèle.

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. Les capacités réflexives de Smalltalk sont utilisées, ce qui permet à un développeur de produire le code, et de décrire le modèle du domaine en même temps. Le développeur utilise donc les mêmes outils et environnement, ce qui n'est pas le cas dans l'approche MOF+Java, où MOF est utilisé pour spécifier un langage de domaine spécifique, et un sous-ensemble de Java est utilisé pour décrire le comportement, ou la sémantique opérationnelle.

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 à 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. Le XMI est une norme trop riche qui permet de sérialiser très différemment un même modèle.

1.9. Roundtrip Engineering

Chair: **Jean-Marie Favre** | 6-Oct 11:00 - 12:30

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

L'auteur présente un système qui permet de gérer plusieurs versions différentes d'un même modèle à travers une série de couche qui sont ensuite fusionnées. Cela permet d'explorer de nouvelles alternatives de conception à des fins d'évaluation.

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

L'UML est de plus en plus utilisé pour développer des systèmes embarqués critiques. Des avancées ont été réalisées dans la traduction des modèles UML en des langages de spécification formels, permettant ainsi à des modèles d'être analysé par des vérificateurs de modèles. Cependant, les développeurs ont du mal à maîtriser la complexité issue de cette formalisation. Les auteurs ont donc conçu un environnement appelé Theseus, qui permet d'interpréter visuellement les résultats d'analyse du vérificateur formel sous forme de diagrammes UML.