

Sujet de master recherche « Architectures logicielles distribuées » 2006–2007

Intégration de l'architecture logicielle COSA au sein de la démarche MDA

Encadrant principal : Tahar KHAMMACI
courriel : Tahar.Khammaci@univ-nantes.fr
tél. : 02 51 12 58 34

Co-encadrant(s) : Mourad Oussalah

Objectif du stage

COSA (Component-Object based Software Architecture) est un modèle de description d'architecture logicielle [1]. C'est un modèle hybride, basé sur la modélisation par objets [2] et la modélisation par composants [3]. La première contribution de ce modèle est la définition des connecteurs en tant qu'entité de première classe pour traiter les interactions entre les composants. Par conséquent, il encourage la réutilisation non seulement des composants mais également des connecteurs. La deuxième contribution est l'amélioration de la réutilisabilité et l'extensibilité des éléments pour soutenir les évolutions dynamiques et statiques des éléments architecturaux. COSA décrit une solution pour réconcilier les besoins des architectures à base de composants avec les notations orientées objets et rejoint ainsi les travaux de Garlan et de son équipe [4][5]. Par ailleurs, MADL est un méta-méta modèle pour l'architecture logicielle. Il permet d'une part, d'unifier tous les concepts architecturaux tout en facilitant leur manipulation et leur réutilisation et d'autre part, de comparer et de transformer différents langages de description d'architectures [1].

La démarche MDA [6] a été proposée par l'OMG. Elle est présentée comme un produit dans la lignée des préoccupations de l'OMG en matière d'intégration et d'interopérabilité. MDA apporte une réponse à la multiplication des intergiciels. L'idée de MDA est de proposer un modèle « stable », indépendant de l'intergiciel, à partir duquel il est possible de dériver différents outils. L'objectif est de capitaliser et de réutiliser ce modèle au lieu de migrer incessamment, de façon laborieuse, d'un intergiciel à l'autre. MDA est basée sur les standards de modélisation de l'OMG : UML, CWM et MOF. Parmi les (nombreux) avantages de la démarche MDA cités par l'OMG, on peut signaler la facilité à gérer l'interopérabilité entre applications développées à l'aide du même « core model ». L'OMG présente MDA comme un cadre général de métamodélisation dans lequel le modèle est considéré comme une entité de première classe. Les opérations applicables à ces modèles (PIM et PSM essentiellement) sont la manipulation et la transformation entre modèles. On dénombre classiquement les transformations « PIM2PIM », « PIM2PSM », « PSM2PSM », etc.

Par ailleurs, la démarche MDA est très souvent associée à la notion de profil UML. Ce dernier constitue le fondement de la démarche MDA. Ainsi, un profil UML est une spécification conforme à l'identification d'un sous-ensemble du méta-modèle UML (qui peut être le méta-modèle UML entier) et à la définition de règles de bonne construction, en plus de celles contenues dans le sous-ensemble du méta-modèle UML. De telles règles permettent de décrire un ensemble de contraintes à l'aide du langage OCL (Object Constraint Language [7]), afin de définir un élément du méta-modèle. Ainsi, la démarche MDA présente un certain nombre de points intéressants. Du point de vue général, elle allie les avantages de la méta-modélisation et de la standardisation mise en œuvre par l'OMG. Si le nombre de modèles de base « core model » disponibles est faible, il n'est cependant pas figé. Le dispositif MDA est « extensible » par le biais de nouveaux modèles, d'autant que la démarche MDA laisse en suspens de nombreux challenges. L'activité d'intégration à proprement parler, la prise en compte d'applications héritées (legacy applications) de précédents développements et la gestion de l'interopérabilité entre intergiciels sont des points cités fréquemment par

l'OMG au nombre des pistes à explorer. Dès lors, il est envisageable de positionner MADL sous la forme d'un nouveau modèle de base MDA.

L'intégration de l'architecture COSA au sein de la démarche MDA rentre dans le cadre des nouvelles orientations des travaux de recherche de la communauté des architectures logicielles, en direction des travaux de l'OMG. Ces deux communautés ont travaillé en parallèle ces dernières années, la pauvreté d'UML en matière de concepts d'architectures logicielles expliquant probablement celà (du moins en partie). Aujourd'hui, l'intégration de la notion de composants dans UML 2.0, peut aussi être interprétée comme un signe de cette évolution et laisse entrevoir la possibilité de définir des architectures à base de composants (ADL) dont le passage vers des plates-formes d'exécution adaptées sera facilité. Ainsi, ce sujet se situe dans cette dernière tendance. Nous souhaitons allier les avantages de l'approche ADL et de l'approche MDA/MOF de l'OMG. De l'approche ADL, nous retenons les notions et mécanismes inhérents aux composants, connecteurs et architectures. Quant à l'approche MDA/MOF basée sur la métamodélisation, elle présente l'avantage de pouvoir considérer les modèles (dans notre cas, architectures, méta-architectures et méta-méta-architectures) comme entités de première classe, et de pouvoir leur appliquer diverses opérations, en particulier, des transformations permettant l'implantation du système sur différentes plates-formes d'exécution.

Travail à réaliser

Il s'agit de proposer un nouveau profil UML (PIM) dédié à l'architecture logicielle COSA dont l'originalité, outre le fait qu'il traite un problème de méta-architecture peu étudié dans les ADLs, résiderait dans la nature de sa spécialisation. En effet, les profils UML existants sont dédiés à un type d'application (systèmes distribués, temps réel, etc.), alors que celui de l'architecture COSA serait « orthogonal », spécialisé dans une activité particulière de modélisation de langages de description d'architectures. Le fait de pouvoir représenter les méta-architectures grâce à MADL avec la notation UML permet de réutiliser naturellement les outils de modélisation graphique UML qui lui sont associés. Plus concrètement, l'approche « UML Profile » consiste à étendre le méta-modèle UML en ajoutant de nouveaux concepts des langages de description d'architectures, et en particulier aux concepts de l'architecture COSA. Cette opération se fonde sur l'utilisation de stéréotypes, de valeurs marquées et de contraintes OCL. L'avantage reconnu de cette approche est l'utilisation des outils UML.

Références

- [1] A. Smeda, Contribution à l'élaboration d'une métamodélisation de description d'architecture logicielle, Thèse de Doctorat de l'Université de Nantes, Juin 2006.
- [2] Object Management Group, UML 2.0 Superstructure Specification : Final Adopted Specification, disponible sur <http://www.omg.org/docs/ptc/03-08-02.pdf>, August 2003.
- [3] N. Medvidovic et R. Taylor, A Classification and Comparison Framework for Software Architecture Description Languages, IEEE Transactions on Software Engineering, Vol. 26, No. 1, pp. 70-93, 2000.
- [4] D. Garlan, Software Architecture and Object-Oriented Systems, Proceedings of the IPSJ Object-Oriented Symposium 2000, Tokyo, Japan, 2000.
- [5] D. Garlan, S. Cheng et J. Kompanek, Reconciling the Needs of Architectural Description with Object-Modeling Notations, Science of Computer Programming Journal, Special UML, Edition 44, Elsevier Press, pp. 23-49, 2002.
- [6] D. Frankel, Model Driven Architecture : Applying MDA to Enterprise Computing, Wiley, Indianapolis, USA, 2003.
- [7] J. Warmer et A. Kleppe, The Object Constraint Language : Precise Modeling with UML, Addison-Wesley Publishing, Reading, Massachusetts, 1998.