

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

Proposition et développement d'une couche de gestion d'évolution par la transformation de graphes

Encadrant principal : Dalila Tamzalit – LINA

Courriel : Dalila.Tamzalit@univ-nantes.fr

tél. : 02 51 12 58 50

Co-encadrant(s) : Mourad Oussalah – LINA

Objectif du stage :

Une architecture logicielle, quelquesoit le paradigme (objet, aspect, composant...) évolue pour répondre aux différents changements dans le système qu'elle décrit (ajout de nouveaux éléments, de nouvelles fonctionnalités, de nouvelles interactions ou tout autres suppressions et modifications). Ce travail s'intéresse à la problématique de l'évolution logicielle d'architectures à base de composants. Les systèmes à base de composants sont des systèmes développés par assemblage de composants réutilisables, préfabriqués et pré-testés (J2EE, .NET, CCM, Rapide, Wright, Acme...). Ils sont définis généralement par un ensemble de composants représentant les fonctionnalités du système et un ensemble d'interactions entre ces composants.

L'objectif est d'enrichir un projet en cours et de l'étendre en proposant et en développant un plug-in pour le modèle de gestion d'évolution SAEV. Ce plug-in sera une extension d'un atelier de transformations de graphes déjà éprouvé, l'atelier AGG écrit en Java.

Contexte :

Le travail vient faire la jonction entre le méta-modèle d'évolution SAEV et l'outil de transformation de graphes AGG:

- SAEV est un méta-modèle dont l'objectif est la gestion des évolutions de modèles d'architectures à base de composants. Il a été conçu au sein de l'équipe MODAL du laboratoire LINA et est basé sur le concept de *règles d'évolution*, d'*invariants* et du *gestionnaire d'évolution*:
 - Règle d'évolution : décrit une opération d'évolution sur un élément architectural donné. Elle fonctionne selon le principe ECA (Evènement/Condition/Action).
 - Invariant : associé aux éléments du méta-modèle. Il permet d'ajouter des contraintes supplémentaires à celles de conformité et doit être vérifié. Les invariants sont utilisés afin de garantir la cohérence globale de l'architecture logicielle.
 - Gestionnaire d'évolution : gère la propagation des évolutions au travers du déclenchement de règles appropriées.
- AGG est un outil pour la visualisation et la transformation des graphes typés et attribués. Ceci permet, par exemple, de représenter des diagrammes UML (p.e. les diagrammes de classes, les diagrammes d'état/transition) comme un graphe, et de les transformer en spécifiant des règles de transformation de graphes. Il s'appuie sur le même principe que SAEV : des évènements et des règles qui précisent les invariants et ce qu'il faut faire en respectant les invariants. Il est opensource et est développé en Java.

Travail à réaliser

Il s'agit de faire le lien entre SAEV et AGG. Plus précisément, il s'agit de :

- comprendre les concepts et le fonctionnement des deux (très similaires dans le principe)
- proposer une démarche unifiée et automatisée de description d'architectures à base de composants selon les concepts d'AGG.
- dériver de manière automatique les règles à partir des invariants exprimés.
- Développer la sur-couche AGG/SAEV.

Mots-clés

Evolution – modèle – Propagation – Transformation de graphes

Références

- N. Sadou, D. Tamzalit, and M. Oussalah. La problématique de l'évolution structurelle dans les «architectures logicielles a base de composants. » In 2 èmes journées sur l'Ingénierie Dirigée par les Modèles, pages 93–108, Lille, 2006.
- S. Ajila. Software maintenance : An approach to impact analysis of objects change. Software - Practice and Experience, 25(10): 1155–1181, 1995.
- <http://tfs.cs.tu-berlin.de/agg/xml.html>
- <http://tfs.cs.tu-berlin.de/agg>