

Sujet de master recherche « Architectures logicielles distribuées » 2005–2006

Génération d'une spécification à partir d'un automate symbolique avec actions

Encadrant principal : Jean-Claude ROYER
 courriel : Jean-Claude.Royer@emn.fr
 tél. : 02 51 85 82 05

Objectif du stage

On se propose de décrire des composants par un automate symbolique et des actions comme dans la figure ci-dessus. La description est très proche des Statecharts d'UML mais en beaucoup plus simple. Cette figure représente graphiquement un composant banque qui reçoit une demande de transfert d'une somme du compte d'un client vers un autre. Il s'agit d'un automate à nombre d'état fini avec deux ajouts : des gardes permettant de contrôler le déclenchement des transitions et une description fonctionnelles des actions associées à chaque transition.

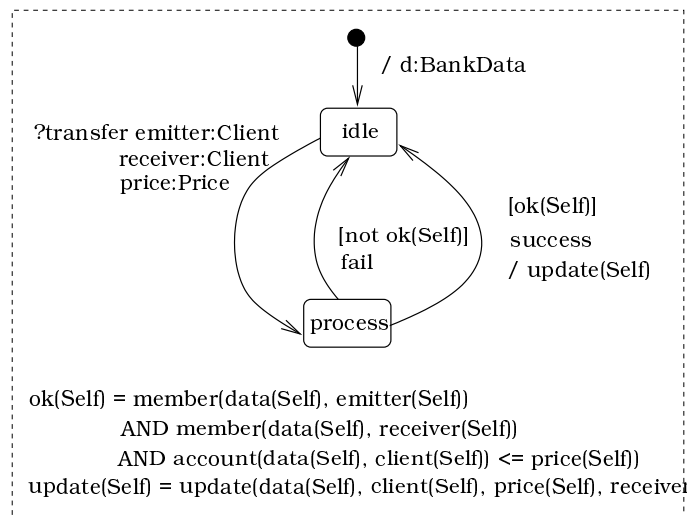


FIG. 1 – Un exemple d'architecture à analyser

Le problème est ensuite de générer un code correspondant à cette description. La génération de code peut-être de plusieurs natures, un travail de thèse en cours s'intéresse à du code Java exécutable. Dans ce travail la génération vers un environnement de spécification et de vérification serait un complément très appréciable. Parmi les cibles que l'on peut avoir en tête il y a par exemple, le système PVS ou encore le couple LOTOS/CADP.

Travail à réaliser

Le premier travail consiste à analyser les éléments essentiels des ADL (par exemple [3]) et des langages de composants. L'approche définie dans [4, 2, 1] devrait être étudiée pour en dégager les

limites et éventuellement s'en inspirer. Après ce premier travail bibliographique le déroulement proposé est :

- Définir une grammaire (ou en spécialiser une déjà existante) pour le format textuel des STS avec actions.
- Choisir un environnement cible adéquate, principalement un langage de spécification assez complet (PVS, LOTOS, ...).
- Etudier l'approche proposée dans [1], pour les composants atomiques, la simplifier, la compléter et l'améliorer si il y a lieu.
- Rédiger les principes de la génération et l'implémenter.

Les outils de représentation et d'analyse basés sur XML pourraient être une bonne base de travail.

Jean-Claude Royer, Ecole des Mines de Nantes, Département Informatique, B220 - 02 51 85 82 05, Jean-Claude.Royer@emn.fr, <http://www.emn.fr/jroyer>

Références

- [1] Michel Allemand and Jean-Claude Royer. Mixed Formal Specification with PVS. In *Proceedings of the 15th IPDPS 2002 Symposium, FMPPTA*. IEEE Computer Society Press, 2002.
- [2] Olivier Maréchal, Pascal Poizat, and Jean-Claude Royer. Checking Asynchronously Communicating Components Using Symbolic Transition Systems. In R. Meersman, Z. Tari, and al, editors, *On the Move to Meaningful Internet Systems 2004 : CoopIS, DOA, and ODBASE*, volume 3291 of *Lecture Notes in Computer Science*, pages 1502–1519. Springer Verlag, 2004.
- [3] Nenad Medvidovic and Richard N. Taylor. A classification and comparison framework for software architecture description languages. *IEEE Transactions on Software Engineering*, 26(1) :70–93, 2000.
- [4] Jean-Claude Royer and Michael Xu. Analysing Mailboxes of Asynchronous Communicating Components. In D. C. Schmidt R. Meersman, Z. Tari and al., editors, *On the Move to Meaningful Internet Systems 2003 : CoopIS, DOA, and ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 1421–1438. Springer Verlag, 2003.