

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

Représentations et transformations des automates symboliques 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 plus lisible (enfin c'est un des buts à atteindre!). 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 fonctionnelle des actions associées à chaque transition.

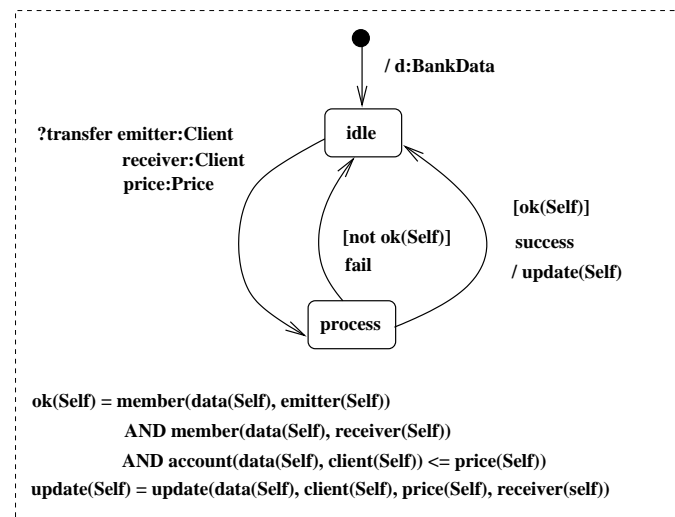


FIG. 1 – Un exemple d'architecture à analyser

Un travail de développement en Java a été initié récemment, son but est de fournir un environnement dédié à ce type de formalisme (STSlib). Le présent travail vise à compléter notre compréhension de ce formalisme et à l'enrichir avec des algorithmes adaptés.

Travail à réaliser

Le premier travail consiste à analyser les éléments essentiels des automates ou systèmes de transitions symboliques, le lien avec les algèbres de processus avec passage de valeurs est également

pertinent. Le groupe OBASCO a fait plusieurs études, en particulier des exemples d'utilisation sont décrits dans les articles suivant [4, 1, 3, 2]. Le déroulement proposé est :

1. Travail bibliographique, besoins et difficultés.
2. Définir une grammaire (ou en spécialiser une déjà existante) pour un format textuel concis et lisible, un format XML, une analyse lexicale et syntaxique.
3. Etudier les moyens généraux de transformation et de "simplification" et illustrer leurs intérêts.
 - Réduction à un état unique
 - Simplification des expressions de gardes
 - Traductions paramétrables vers les algèbres de processus
4. Expérimenter et implémenter au moins un des algorithmes proposés.

Les outils de représentation et d'analyse basés sur XML pourraient être une bonne base de travail ainsi que l'environnement Java sous Eclipse.

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

Références

- [1] 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.
- [2] Pascal Poizat and Jean-Claude Royer. Formal Specification in Korrigan of the Cash Point Case Study. Technical report, Ecoles des Mines de Nantes, 2006. <http://www.emn.fr/x-info/~jroyer>.
- [3] Pascal Poizat, Jean-Claude Royer, and Gwen Salaün. Bounded Analysis and Decomposition for Behavioural Description of Components. In Springer Verlag, editor, *FMOODS*, number 4037 in *Lecture Notes in Computer Science*, pages 33–47, 2006.
- [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.