

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

## Passage de contexte par aspects

Encadrant principal : Rémi DOUENCE

courriel : douence@emn.fr

tél. : 02 51 85 82 15

Co-encadrant(s) :

### Objectif du stage

La programmation par aspects [1] permet de modulariser des préoccupations transverses (c.a.d. permet de réunir des morceaux de code qui concernent une même fonctionnalité). La programmation par aspects est particulièrement adaptée au passage de contexte qui nécessite d'ajouter du code dans une application de base pour :

1. collecter le contexte,
2. accéder au contexte,
3. et enfin effacer le contexte.

AspectJ [2] propose un seul opérateur `cflow` pour le passage de contexte. Nous avons récemment proposé [3] de définir des coupes à l'aide d'un langage logique : Datalog [4]. Ce langage permet de définir des coupes déclaratives qui expriment les relations entre différentes entités (objets) du programme de base. Nous l'avons utilisé pour ajouter des explications dans un solveur de contraintes [5], ainsi que pour enrichir l'interface du canevas de dessin JHotDraw [6]. Afin de faire les expérimentations, nous avons implémenté un petit interprète Datalog facile à embarquer. Dans un deuxième temps, pour des raisons de performance, nous avons défini des schémas de compilation que nous avons testé manuellement.

### Travail à réaliser

Notre implémentation de Datalog repose sur un algorithme de chaînage avant. Le travail consistera à étudier d'autres options :

- étendre notre implémentation pour y intégrer un algorithme de chaînage arrière [4]
- réutiliser un moteur Datalog industriel [7]

Ce travail produira un outil pour la programmation par aspect dédiée au passage de contexte. Cet outil devra ensuite être évalué en l'utilisant pour modifier le canevas JHotDraw.

Ce travail permettra de développer à la fois des compétences formelles (par exemple, algorithmes pour Datalog, transformations de programmes, compilation) ainsi que des compétences appliquées (par exemple programmation en Java, programmation en AspectJ, analyseurs syntaxiques, canevas).

### Références

- [1] G. Kiczales et al. Aspect-oriented programming. In *ECOOP'97 : Proceedings of European Conference of Object-Oriented Programming*, volume 1241 of *LNCS*, pages 220–242. Springer Verlag, 1997.

- [2] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. Griswold. An overview of AspectJ. In *ECOOP'01 : Proceedings of European Conference of Object-Oriented Programming*, volume 2072 of *LNCS*. Springer-Verlag, 2001.
- [3] R. Douence. Relational aspects for context passing : Beyond stack inspection. In *Soumis à publication*, 2005.
- [4] S. Ceri, G. Gottlob, and L. Tanca. What you always wanted to know about datalog. *IEEE Transactions on Knowledge and Data Engineering*, 1(1) :146–166, 1989.
- [5] R. Douence and N. Jussien. Non-intrusive constraint solver enhancements. In *First AOSD workshop on ACP4IS : Aspects, Components, and Patterns for Infrastructure Software*. University of Twente, 2002.
- [6] JHotDraw Site. Jhotdraw as open-source project. <http://www.jhotdraw.org/>.
- [7] Jess Site. Jess, the rule engine for the java platform. <http://herzberg.ca.sandia.gov/jess/>.