

Proposition de stage de Master

Jean-Bernard Stefani

Titre

Composants et Coalgèbres

Thématique

Langages de programmation, Sémantique

Laboratoire

INRIA – Centre de Grenoble-Rhône-Alpes LIP

Ville et pays

Grenoble, France Lyon, France

Equipe

Equipe-projet INRIA SARDES Equipe LIP Plume

Noms et adresses électroniques des directeurs de stage

- Jean-Bernard Stefani (responsable de l'équipe SARDES) – jean-bernard.stefani@inria.fr
- Filippo Bonchi (équipe Plume) – filippo.bonchi@lip.fr

Contexte

L'équipe-projet SARDES développe une approche de la construction de systèmes logiciels dynamiquement configurables (notamment systèmes d'exploitation et systèmes distribués) qui combine programmation par composants et programmation réflexive. L'équipe a en particulier développé le modèle de composants Fractal [5]. Ce modèle de composants a été formellement défini dans le langage de spécification Alloy [8] et a fait l'objet de plusieurs implantations (dont l'implantation de référence en Java et une implantation en C).

Objectifs du stage

Le but du stage est d'amorcer une étude sémantique de la notion de composants logiciels telle que proposée dans le modèle Fractal en utilisant une approche coalgébrique. La notion de coalgèbre [9] fournit une approche très générale de la notion de comportement, qui généralise la notion de système de transition, ainsi que les notions standards de bisimulation et de logiques modales associées. Les travaux de Bonchi et Montanari en particulier [4, 3] ont montré qu'une approche coalgébrique est tout à fait adaptée pour définir la sémantique de systèmes réactifs. Quelques travaux [1, 7, 6, 2] adoptent une telle approche pour formaliser la sémantique de composants logiciels. Malheureusement, ils ne considèrent que des assemblages de composants

essentiellement statiques, c'est-à-dire qui ne peuvent être modifiés en cours d'exécution comme c'est le cas avec des assemblages de composants Fractal.

Le stage cherchera donc à définir une sémantique catégorique coalgébrique du modèle de composants en s'attachant notamment aux aspects suivants:

- La définition de différentes formes de composition, y compris de formes dynamiques où un composant peut être retiré d'un assemblage en cours d'exécution.
- La possibilité de définir des assemblages de composants autorisant du partage (le graphe de composition est un graphe et non une simple forêt) et du paramétrage.

Connaissances attendues

Sémantique des langages de programmation, théorie des catégories, calculs de processus.

References

- [1] L. Barbosa and J. Oliveira. State-based components made generic. *Electronic Notes in Theoretical Computer Science*, vol. 82, no.1, 2003.
- [2] L.S. Barbosa, S. Meng, B.K. Aichernig, and N. Rodrigues. On the Semantic of Componentware: A Coalgebraic Perspective. In Z. Liu and H. Jifeng, editors, *Mathematical Frameworks for Component Software - Models for Analysis and Synthesis*, volume 2 of *Component-Based Software Development*, chapter 3. World Scientific, 2006.
- [3] F. Bonchi and U. Montanari. Coalgebraic symbolic semantics. In *3rd International Conference Algebra and Coalgebra in Computer Science (CALCO)*, volume 5728 of *Lecture Notes in Computer Science*. Springer, 2009.
- [4] F. Bonchi and U. Montanari. Reactive systems, (semi-)saturated semantics and coalgebras on presheaves. *Theor. Comput. Sci.*, 410(41), 2009.
- [5] E. Bruneton, T. Coupaye, M. Leclercq, V. Quema, and J.B. Stefani. The Fractal Component Model and its Support in Java. *Software - Practice and Experience*, 36(11-12), 2006.
- [6] S. Meng, B. K. Aichernig, L.S. Barbosa, and Z. Naixiao. A Coalgebraic Semantic Framework for Component-based Development in UML. *Electr. Notes Theor. Comput. Sci.*, 122, 2005.
- [7] S. Meng and L.S. Barbosa. Components as coalgebras: The refinement dimension. *Theor. Comput. Sci.*, 351(2), 2006.
- [8] P. Merle and J.B. Stefani. A formal specification of the Fractal component model in Alloy. Research Report RR-6721, INRIA, France, 2008.
- [9] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, vol. 249, 2000.