

Reconfiguration dynamique d'architectures basées sur des composants Fractal réparties

Encadrant principal : Thomas Ledoux

Courriel : ledoux@emn.fr

Tel. : 02.51.85.82.19

Co-encadrant(s) :

Problématique et contexte

Les développeurs d'applications sont aujourd'hui confrontés à des contextes d'exécution de plus en plus variables. D'une part, on trouve une grande diversité de plates-formes couvrant un large spectre en terme de ressources disponibles (des systèmes embarqués aux grilles de calculs). D'autre part, le contexte d'exécution d'une application donnée évolue de plus en plus pendant son exécution (disponibilité des ressources matérielles et logicielles, nomadisme, etc.).

Le besoin de construire des applications qui s'adaptent au contexte n'est pas nouveau. Cependant, cela est généralement fait de façon ad hoc, en tentant d'anticiper au moment du développement les futures conditions d'exécution de l'application, et en intégrant directement dans celui-ci les décisions d'adaptation : le mélange du code métier et du code chargé d'adapter celui-ci implique une complexité accrue de l'application, qui rend son développement et sa maintenance difficile (Dowling *et al.*, 2001). On aimerait pouvoir *modulariser* le code chargé de l'adaptation et *l'intégrer* dynamiquement dans le code métier afin de garantir un découplage spatio-temporel entre les deux types de code.

S'inspirant d'une approche de programmation par aspects (Kiczales *et al.*, 1997), notre proposition est fondée sur Safran (David, 2005), un système pour le développement d'applications adaptatives basé sur le modèle de composants Fractal (Bruneton *et al.*, 2004). Safran est constitué (i) d'un langage dédié permettant de programmer *l'aspect d'adaptation* sous la forme de politiques réactives, et (ii) du support d'exécution nécessaire au *tissage* et à l'exécution de ces politiques (aspects) dans les composants Fractal (programme de base).

Objectif du stage

Actuellement, le support d'exécution de Safran repose d'une part sur WildCAT (David *et al.*, 2005), un système permettant de détecter les évolutions du contexte d'exécution (quand adapter ?), et d'autre part sur FScript, un langage dédié pour la reconfiguration dynamique consistante de composants Fractal (comment adapter ?).

Cependant, l'infrastructure Safran reste minimale puisqu'elle concerne le développement d'applications adaptatives relativement centralisées. En effet, la limitation principale de l'implémentation actuelle est qu'elle n'aborde pas la problématique de la distribution.

L'objectif de ce stage est d'étendre les principes de Safran pour permettre l'adaptation d'applications Fractal réparties. Les perspectives de ce travail devront apporter – si possible – une contribution à la gestion automatique des ressources dans les applications de type *grid computing*.

Travail à réaliser

Plusieurs extensions à différents niveaux de Safran sont possibles.

D'une part, il est possible d'enrichir FScript pour supporter la migration de composants ou la manipulation de connexions distribuées (via Fractal RMI par exemple). Une telle extension nécessiterait de prendre en compte la nature distribuée des transactions de reconfigurations.

D'autre part, de nouveaux domaines contextuels WildCAT peuvent être implémentés pour pouvoir partager les informations entre les différents nœuds d'une application.

Enfin, l'aspect d'adaptation lui-même pourrait être étendu pour permettre à des composants distants de s'adapter de façon coordonnée (Cherfour *et al.*, 2003).

Références

- Bruneton *et al.*, 2004 – E. Bruneton, T. Coupaye, M. Leclerc, V. Quema, J-B. Stefani. « An Open Component Model and Its Support in Java ». 7th ACM International Symposium on Component-Based Software Engineering (CBSE7), Edinburgh, Scotland, May 2004.
- Cherfour *et al.*, 2003 – Cherfour D., André F., « Auto-adaptation de composants ACEEL coopérants », *CFSE'3, Conférence Française sur les Systèmes d'Exploitation*, La Colle sur Loup, France, October, 2003.
- David, 2005 – David P.-C., Développement de composants Fractal adaptatifs : un langage dédié à l'aspect d'adaptation, Phd thesis, Université de Nantes / École des Mines de Nantes, July, 2005.
- David *et al.*, 2005 – Pierre-Charles David and Thomas Ledoux, «WildCAT: a generic framework for context-aware applications », MPAC 2005, ACM Digital Library, Grenoble, France, November 2005.
- Dowling *et al.*, 2001 – Dowling J., Cahill V., « The K-Component Architecture Meta-Model for Self-Adaptive Software», in , A. Yonezawa, , S. Matsuoka (eds), *Proceedings of Reflection 2001, The Third International Conference on Metalevel Architectures and Separation of Crosscutting Concerns, Kyoto, Japan*, vol. 2192 of *LNCS*, AITO, Springer-Verlag, p. 81-88, September, 2001.
- Kiczales *et al.*, 1997 – Kiczales G., Lamping J., Mendhekar A., Maeda C., Lopes C. V., Loingtier J.-M., Irwin J., « Aspect-Oriented Programming », *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, vol. 1241 of *LNCS*, Springer-Verlag, June, 1997.