Sujet de master recherche ■ Architectures logicielles distribuées ■ 2006–2007

Etude et mise en place de protocoles de composants Kmelia

Encadrant principal : Pascal Andre
courriel : `Pascal.Andre@univ-nantes.fr`
tél. : 02 51 12 59 65

Co-encadrant(s) : Gilles Ardourel, Christian Attiogbé

Cadre du travail

Une des préoccupations de notre équipe est l'étude des mécanismes d'analyse, de spécification et de développement rigoureux de composants logiciels ayant des propriétés formellement exprimées. Dans ce cadre nous avons proposé un modèle à composants basé sur les services : le modèle Kmelia. Le modèle est assorti d'un langage de spécification ayant le même nom [1, 2, 3]. Dans l'approche Kmelia, un composant (abstrait) a un invariant ($InvC$) et offre plusieurs services. Chaque service S précondition ($preS$) et une postcondition ($postS$) et un comportement sous forme d'automate. Le comportement décrit l'enchaînement des actions d'un service (actions élémentaires, échanges de messages, invocation de service). L'équipe développe à des fins d'expérimentation des travaux de recherche, un environnement nommé COSTO (*Component Study TOolbox*[1]).

Pour développer des composants sûrs, on souhaite garantir en les vérifiant, les propriétés des composants et des services. Plusieurs propriétés sont identifiées ; par exemple la composabilité (qui est la combinaison de l'interopérabilité statique et de l'interopérabilité dynamique), préservation de l'invariant, substituabilité, etc. Nous avons étudié et développé des outils pour vérifier certaines propriétés, par exemple la composabilité. Un point méthodologique important consiste à réutiliser les résultats et les outils existants par ailleurs dans le domaine de la vérification formelle ; nous avons ainsi utilisé les Lotos/CADP et Mec pour analyser des composants et services spécifier en Kmelia.

Dans le but de faciliter l'accès au service, on se propose d'introduire des protocoles dans Kmelia. Ces protocoles servent de mode d'emploi des services et mettent en évidence des dépendances d'enchaînement entre services. Ainsi tel service doit être précédé de tel autre service. Un protocole peut jouer le rôle d'une session, d'une vue ou d'une classe d'usager. Pour différentes raisons nous avons choisi de représenter les protocoles par des services particuliers en Kmelia.

Objectif du stage

Le but de ce stage est d'étudier l'intégration des protocoles en Kmelia à la fois au niveau théorique par une formalisation de la notion et des propriétés particulières qu'on peut y associer, et au niveau pratique en complétant la plateforme COSTO par des outils de description et de vérification de protocoles.

Etudier les *propriétés d'ordonnancement* des services dans les protocoles. Etudier l'expression et la transformation systématique en B (ou un autre langage formel) de telles propriétés. Développer (en Java) un module qui s'intégrera dans la boîte à outils COSTO.

[1] `http://lina.atlanstic.net/fr/equipes/team10/Kmelia/`

Travail à réaliser

S'imprégner rapidement du modèle Kmelia et notamment des mécanismes de composition de services. Etudier les travaux de l'équipe autour des protocoles [4], et en particulier les spécificités des protocoles vis-à-vis des services en expérimentant sur des exemples.

Etudier différentes façon d'expliciter les différentes propriétés d'ordonnancement des protocoles (obligations de preuve en B ou Z, logique temporelle, algorithmes...). Parmi les garanties à produire pour assurer la correction d'un composant, il y a la cohérence des protocoles. La détection d'incohérence dans les protocoles fait partie des vérifications nécessaires pour assurer la correction d'un composant. Un protocole r d'un composant C est *incohérent* si un des enchaînements de services qu'il décrit peut être impossible du seul fait de l'enchaînement. Les deux causes d'incohérence suivantes peuvent être détectées :

- L'existence de chemins gardés sans alternatives menant à l'état final du protocole, dans le cas ou l'expression de ces gardes ne peut être évaluée à vrai.
- L'existence dans un protocole d'une séquence d'appels $s_i - s_j; s_k$ telle que la prise en compte des postconditions de s_i à s_j implique la négation de la précondition de s_k, c'est-à-dire qu'un des services appelé avant s_k et dont les effets n'ont pas étés remis en cause empêche le déroulement correct de s_j. Par exemple, si le service `connexion` possède une précondition *notconnected* et une postcondition *connected*, la séquence `connexionconnexion` rend le protocole incohérent, ainsi que toute autre séquence ne contenant aucune modification de *connected* entre deux appels de `connexion`.

Pour l'analyse des suites de séquences infaisables, nous envisageons pour l'instant des passerelles avec des prouveurs de théorèmes de la logique du premier ordre tel que Atelier B, Z-EVES, HOL...

Implanter ces idées dans le prototype COSTO.

Références

[1] Christian Attiogbé, Pascal André, and Gilles Ardourel. Checking Component Composability. In *5th International Symposium on Software Composition, SC'06*, volume 4089 of *LNCS*. Springer, 2006.

[2] Pascal André, Gilles Ardourel, and Christian Attiogbé. Vérification d'assemblage de composants logiciels Expérimentations avec MEC. In Michel Gourgand and Fouad Riane, editors, *6e conférence francophone de MOdélisation et SIMulation, MOSIM 2006*, pages 497–506, Rabat, Maroc, April 2006. Lavoisier.

[3] Pascal André, Gilles Ardourel, and Christian Attiogbé. Spécification d'architectures logicielles en Kmelia : hiérarchie de connexion et composition. In *1ère Conférence Francophone sur les Architectures Logicielles*, pages 101–118. Hermès, Lavoisier, 2006.

[4] Pascal André, Gilles Ardourel, and Christian Attiogbé. Protocoles d'utilisation de composants : Spécification et analyse en Kmelia. In (Eds.), editor, *Soumis à LMO 2007*, 2007.

[5] B. Meyer. The Grand Challenge of Trusted Components. In *Proceedings of 25th International Conference on Software Engineering*, pages 660–667. IEEE Computer Society, 2003.

[6] Robert Allen and David Garlan. A Formal Basis for Architectural Connection. *ACM Transactions on Software Engineering and Methodology*, 6(3) :213–249, July 1997.

[7] D.M. Yellin and R.E. Strom. Protocol Specifications and Component Adaptors. *ACM Transactions on Programming Languages and Systems*, 19(2) :292–333, 1997.

[8] Jean-Raymond Abrial. *The B-Book Assigning Programs to Meanings*. Cambridge University Press, 1996. ISBN 0-521-49619-5.

[9] John Wordsworth. *Software Engineering with B*. Addison-Wesley, September 1996.

[10] Pascal André and Alain Vailly. *Spécification des logiciels ; Deux exemples de pratiques récentes : Z et UML*, volume 2 of *Collection Technosup*. Editions Ellipses, 2001. ISBN 2-7298-0774-8.