

Reconfiguration dynamique d'une architecture logicielle à l'aide de BZR

Type de contenu : Texte

Titre(s) : Reconfiguration dynamique d'une architecture logicielle à l'aide de BZR ; BATISTA, Thais ; BUISSON, Jérémy ; SLT SENELLART de VRIERE, Eric

Autre(s) responsabilité(s) : BATISTA, Thais (Directeur de thèse)
BUISSON, Jérémy (Directeur de thèse)

SLT SENELLART de VRIERE, Eric Promotion Chef de bataillon Bulle (2010-2013) (Secrétaire)

Editeur, producteur : Ecoles Militaires de Saint-Cyr Coëtquidan

Description matérielle : 1 CD

Note sur le contenu : mémoire

Note de thèses et écrits académiques : Filière Scientifique - Option Informatique Promotion Chef de bataillon Bulle Date de soutenance : 01/01/2013

Résumé ou extrait : **PRESENTATION** : Lorsqu'un logiciel est modifié, il est souvent nécessaire de redémarrer l'application associée afin de s'assurer que tous les changements effectués sont pris en compte et respectent les contraintes imposées au programme. Or, redémarrer n'est souvent pas pratique, et même parfois très déconseillé comme pour les systèmes de sécurité. Notre objectif est donc de chercher à optimiser une reconfiguration logicielle à l'aide d'un contrôleur afin de s'assurer qu'elle se fasse correctement. Ce contrôleur doit vérifier que la mise à jour soit compatible et possible au niveau architecture du système. **CONSTRAINTES** : Nous avons travaillé avec deux langages qui jusque là n'ont pas été conçus pour fonctionner dans un même programme. Aussi, le problème consiste à faire interagir un programme écrit en ACME avec un contrôleur BZR par l'intermédiaire de variables communes. Ceci est réalisé avec l'aide d'un compilateur qui permettra d'insérer le code généré dans un middleware. Bien que ce dernier soit toujours en développement, nous voulons arriver au code compilé. **RESULTATS OBTENUS** : En recoupant les prédicats des deux scripts (ACME et Heptagon) tout en respectant les contraintes imposées à l'architecture du logiciel, il devient possible de les faire communiquer et ainsi de s'assurer que le contrôleur a effectivement une main mise sur les reconfigurations. Enfin, on s'aperçoit de la possibilité de mettre en parallèle ces deux langages à l'aide du compilateur hec, ce qui permettrait ultérieurement d'accroître le domaine d'utilisation de Heptagon/BZR dans les reconfigurations logicielles. **LIMITE** : La principale limite rencontrée lors de notre travail est que le middleware avec lequel le code généré par le compilateur devait être utilisé n'était pas encore au point. Aussi, il n'a pas été possible d'implémenter le script obtenu. **CONCLUSION** : Grâce à ce projet, nous avons pu constater la possibilité de faire en sorte qu'un langage de programmation réactive puisse contrôler le bon déroulement d'une reconfiguration d'une architecture logicielle. Bien qu'il nous manque le middleware nécessaire, il est clair qu'une fois compilé, les automates Heptagon et le contrat BZR peuvent effectivement s'assurer que toute erreur puisse être évitée. Avec un contrôleur suffisamment complexe (contrairement à l'exemple simplifié employé pour illustrer notre argumentation), une architecture bien plus complexe pourra voir ses

reconfigurations prise en compte par le contrôleur seul.

Sujet(s) : architecture : informatique

configuration informatique

logiciel

mise à jour

programme informatique

système : informatique

système de sécurité