

Assemblage automatique et adaptation d'applications à base de composants

Guillaume Grondin^{*,**}, Noury Bouraqadi^{*} et Laurent Vercouter^{**}

^{*}Département IA, École des Mines de Douai
941 rue Charles Bourseul – B.P. 10838, 59508 Douai Cedex, France
{grondin,bouraqadi}@ensm-douai.fr
<http://csl.ensm-douai.fr>

^{**}Centre G2I, École des Mines de Saint-Étienne
158 cours Fauriel, 42023 Saint-Étienne Cedex 02, France
vercouter@emse.fr
<http://www.emse.fr/~vercouter>

Résumé. Dans cet article, nous introduisons MADCAR, un modèle de moteurs dédiés à la construction et à la reconfiguration dynamique et automatique d'applications à base de composants. Dans MADCAR, la description d'une application regroupe la définition des configurations valides et les règles de transfert de l'état de l'application lors des adaptations. Cette description est découplée de toute implémentation et peut donc être réutilisée avec différents jeux de composants. Partant d'une description d'application, un moteur MADCAR construit un problème de contraintes dont la résolution permet le choix de la configuration cible et des composants à utiliser. Ce choix prend en compte le coût de la configuration cible et son adéquation avec les ressources disponibles. Afin d'assurer la cohérence de l'application, le moteur utilise les règles de transfert d'état pour initialiser les attributs des composants de l'assemblage cible à partir des attributs des composants de l'assemblage de départ.

Summary

In this paper, we introduce MADCAR, a model of engines for dynamic and automatic (re)assembling of component-based software. In MADCAR, an application description consists of the definition of some valid configurations and the state transfer rules to apply during adaptations. This description is decoupled from any implementation and can therefore be reused with other components. Given an application description, a MADCAR engine builds a constraint solving problem that makes it possible to choose an appropriate configuration and the components to assemble. This choice takes into account the cost of the target configuration with respect to the available resources. To ensure the application consistency, the engine relies on the state transfer rules to initialize the component attributes of the target assembly using the component attributes of the source assembly.