

# Adapting applications on the fly

Abdelmadjid Ketfi, Nouredine Belkhatir, Pierre-Yves Cunin

*Adele Team Bat C*

*LSR-IMAG , 220 rue de la chimie - Domaine Universitaire, BP 53  
38041 Grenoble Cedex 9 France*

## Abstract

When it is necessary to adapt an application, traditionally, the administrator must stop it to introduce the needed modifications. This static approach is not suitable for critical systems that have to be non-stop and highly available. With this kind of systems, the adaptation must take place at run-time and the administrated application cannot be entirely stopped. Existing approaches differ in their performance, their degree of automation, their consistency and in the granularity that they address. The granularity means the basic entity that can be adapted. Usual solutions focus on how to replace a procedure, a module or an object in a running application. Our work focuses on component-based applications.

Adapting a component-based application means adapting one or more of its components, and in general, adapting a component at run-time means disconnecting it from the application and connecting a new version of this component. The adaptation may imply the modification of the connections between components, the migration of components from an execution site to another or the modification of the services provided by a specific component. A dynamic adaptation may be performed to correct the faulty behavior of some components, in response to modifications affecting the execution environment, to extend or to improve the performance of the running application.

Many approaches like OLAN (1) DCUP (2) are based on component models in which each component is constituted by a functional part and a control part. This last part provides a set of services to administrate the component. In our approach we aim to separate the control part outside the component. The benefit is the possibility to apply our solutions to existing models that have not been designed to support the dynamic adaptation. Through a first experimentation on the JavaBeans component model (3), we define an adaptation process composed by several steps: 1) define the mapping rules between the old and the new component (correspondence between properties and services); 2) passivating the old and the new components; 3) transferring the state of the old component into the new one according to the mapping rules defined before; 4) disconnecting the old component and connecting the new one; 5) activating the new component. These steps may constitute the basic activities of an adaptation machine.

In order to validate our approach, we are currently working on a second experimentation on OSGi components. Some challenges are the management of the objects that belong to a component, the interface change between the old component and the new one and how to guarantee the consistency of the application when the adaptation is performed.

## ***Bibliography***

- (1) Bellissard L., De Palma N., Freyssinet A., Herrmann M., Lacourte S., « *An Agent platform for Reliable Asynchronous Distributed Programming* », SRDS'9), Lausanne, Suisse, October 1999.
- (2) Plasil, F., Balek, D., Janecek, R., « *DCUP: Dynamic Component Updating in Java/CORBA Environment* », Tech. Report No. 97/10, Dep. of SW Engineering, Charles University, Prague, 1997.
- (3) A. Ketfi, N. Belkhatir and P.Y. Cunin « *Dynamic updating of component-based applications* », accepted in SERP'02, June 2002, Las Vegas, Nevada, USA, <http://www-adele.imag.fr/Les.Publications/BD/SERP2002Ket.html>