

# Administration and Deployment Tools in a Message-Oriented Middleware

Vivien Quéma<sup>1</sup>, Roland Balter<sup>2</sup>, Luc Bellissard<sup>2</sup>,  
David Féliot<sup>2</sup>, André Freyssinet<sup>2</sup>, Serge Lacourte<sup>2</sup>

<sup>1</sup>INRIA Rhône Alpes  
38334 Saint-Ismier Cedex, France  
Vivien.Quema@inrialpes.fr

<sup>2</sup>ScalAgent Distributed Technologies  
38432 Echirolles Cedex, France  
Roland.Balter@scalagent.com

## 1 Introduction

The use of the Internet leads to emerging interactive services that are cooperating with existing enterprise information systems. Developing these new services is a challenging issue both from the software engineering point of view and from the run-time point of view. It is recognized today that message oriented middleware (MOM) is well-suited to support these services. However MOM-based environments are suffering from a lack of tools for building and operating distributed applications.

The ScalAgent infrastructure aims at filling this gap. It provides — besides a MOM — an asynchronous component-based programming model and a set of tools allowing the description, the configuration and the administration of applications made of components interacting via the MOM.

## 2 Hierarchical component model for asynchronous applications

The component model is dedicated to the construction of asynchronous distributed applications. Components communicate by sending and receiving messages. This component model is reflexive: a component is composed of two parts: a functional part, called *SCBean*, and a container, called *SCContainer* that represents a meta-level causally linked to the functional level, allowing the management of the component's lifecycle and non-functional properties. Moreover, this model is hierarchical: an assembly of *SCContainers* may be manipulated as a special *SCContainer*, called *SCController* which, can itself be part of an other *SCController*.

## 3 Integrated configuration and administration environment

The ScalAgent platform provides an Architecture Description Languages (ADL) that allows the description

and configuration of distributed applications built using the component model presented in section 2. It also provides a set of tools allowing the deployment and the administration of applications described using the ADL.

## 4 The message-oriented middleware

The run-time infrastructure provides an asynchronous messaging service that guarantees messages delivery, ordering of messages, security, etc. The MOM is implemented as a set of servers organized in a bus architecture. Each server hosts *SCBean* components.

As an exemple, Joram, the open source JMS implementation is built on top of the MOM [1].

## 5 Future work

We are working on the MOM configuration. Our objective is to use both the description of the application and a description of the system (machines characteristics, network topology) — using an Hardware Description Language (HDL) — to allow a fine-grained configuration of the middleware. This should allow to automatically deploy a customized middleware and to optimize the application performances.

## References

- [1] JORAM: Java Open Reliable Asynchronous Messaging, 2002. Objectweb, <http://www.objectweb.org/joram/>.
- [2] V. Quéma et al. Application-Driven Customization of Message-Oriented Middleware for Consumer Devices. In *Workshop on Software Infrastructures for Component-Based Applications on Consumer Devices*, 2002.
- [3] ScalAgent Distributed Technologies, <http://www.scalagent.com>.