Aurora: An Architecture for Dynamic and Adaptive Work Sessions in Open Environments

Manolis Marazakis, Dimitris Papadakis, and Christos Nikolaou

Department of Computer Science, University of Crete
and
Institute of Computer Science, FORTH,
PO Box 1385, GR 71110 Heraklion, Greece.
e-mail: {maraz,dimpapa,nikolau}@ics.forth.gr
WWW: `http://www.ics.forth.gr/pleiades/projects/Aurora`

Abstract. This paper presents the Aurora architecture for network-centric applications, complementing the CORBA and WWW/Java frameworks with support for composition of services. The architecture addresses the requirements of dynamic open environments with multiple autonomous service providers. The paper presents a detailed technical design of the run-time infrastructure that enables on-demand composition of services. Aurora is based on a *container framework*, which provides the basis for dynamic and adaptive composition, as well as detailed monitoring and tracking. Work sessions are implemented as networks of active containers. A distinguishing feauture of Aurora is that such networks can be inspected and manipulated at run-time.

1 Introduction

In open and dynamic environments, such as the Internet, a *network-centric* application paradigm is becoming popular, as a means to utilize widely distributed application components and information resources that are made available by autonomous providers. In such environments, dynamic configuration and composition are key requirements, as the basis for both *coordination*, which entails structured processes involving mainly automated activities, and *collaboration*, which entails mainly unstructured processes with significant human interaction. Current distributed application frameworks do not adequately support these requirements, as they assume well-structured applications where software components are developed according to given specifications and interact in predefined configurations. Applications are structured as networks of components with static interconnections and there is no support for dynamic adaptation.

CORBA [1] is a comprehensive distributed application framework that provides mechanisms and support services for developing applications as collections of distributed objects. The World Wide Web (WWW) [2], although it started as a distributed service over the Internet facilitating the dissemination of documents, is evolving towards becoming a complete platform for applications that combine the functionality and services of multiple components, which are owned and managed by autonomous authorities. The introduction of the Java programming language [3] has contributed to this objective by offering support for *active content*. Furthermore, several vendors have proposed development frameworks (see for example Netscape's ONE, IBM's NCF, and Oracle's NCA) that extend the CORBA and the WWW/Java functionality by introducing additional middleware services.

We view all such approaches as being to a large extent complementary, each being more suitable for certain applications. Therefore, we propose a framework within which they can be combined, and extended towards supporting coordination and collaboration and accommodating the dynamic aspects of open environments. *Aurora* [4] is an architecture we are developing towards that end, aiming for a unified treatment of the problem of supporting network-centric applications via a set of common middleware services [5].

The remainder of this paper is organized as follows: In Section 2 overviews the *Aurora* architecture and application model. Section 3 describes the *Aurora* session management infrastructure, which relies on a container framework and a scripting language, and supports dynamic and adaptive work sessions. Section 4 provides a comparison of our approach with related research. Finally, Section 5 summarizes the ideas expressed in this paper, and reports on the current state of our development effort.

2 Overview of *Aurora*

Dynamic open environments exhibit certain distinguishing features related to autonomy, consistency requirements, dynamics of the execution, and human factors [6]. Each service provider is *autonomous* in the way it provides services to a task that is part of a workflow. It is to be treated as a "black box" whose details of operation are not known to, and cannot be controlled by, its clients. Each service provider is responsible and can manage only its own services.

By providing a publicly accessible interface, a service provider allows unpredictable interactions between its local application programs that implement its services with *any* other program that is aware of this interface and the way to use it. A client may have to determine where a service provider is located before using its services. A service provider may allow different access privileges to different clients. Therefore, a service provider encapsulates services and resources in software components which export well-defined interfaces for use by clients who, in general, do not know, and cannot control, the details of operation. Clients can be either users, represented by interaction tools (such as WWW browsers) or other service components, when complex services have to be composed of "primitive" services.

Aurora is an open architecture for network-centric applications that supports dynamic configuration and composition of components, as well as detailed on-line monitoring. Figure 1 illustrates the basic components of the *Aurora* architecture. The following subsections describe these components.