SUMMARY OF Ph.D. DISSERTATION

School	Student Identification Number	SURNAME, First name
School of Science for Open and Environmental Systems		NAGURA, Masataka

Title

Design and Implementation of a Mechanism Supporting the Orchestration of Heterogeneous Distributed Components

Abstract

Distributed component technologies are widely used for developing applications that communicate over the network. An application program using such a technology, e.g., EJB (Enterprise JavaBeans), CORBA (Common Object Request Broker Architecture) and Web Service, consists of a server program that provides a service and a client program that calls the server programs over network. Using one component technology, different services can be invoked using the same calling procedures. Therefore application development can be simplified; an application invokes existing services and combines them as software components. Critical issues, however, exist at application development time and at runtime.

First, at application development time, we may need to combine services provided by different vendors. Since each service is developed independently, possibly using different component technologies, we need to consider the heterogeneity of these components.

Second, at application runtime, applications should be able to flexibly use many services deployed on many types of terminals (such as mobile and embedded appliance). Thus, distributed components need to be executed through Plug and Play, so they can automatically work with other components on the network, by just connecting the terminal that provides them to the network.

We propose a mechanism to support the development and the operation for applications orchestrating heterogeneous distributed components. To solve the first issue, we propose adapters to absorb the heterogeneity between components. Developers first describe workflow information for statically combining components. This is used to automatically combine services consisting of heterogeneous distributed components, and generate the application that calls these components through the generated adapter. To solve the second issue, the generated application is executed on a platform provided by our proposed mechanism. This platform includes middleware for discovering and using components existing on a local network. They discover required components at runtime and invoke discovered components. Services consisting of heterogeneous distributed components are combined dynamically.

The adapter that absorbs the heterogeneity between components incurs about 30 msec overhead for component processing time. When applications discover and use components at runtime, heterogeneity is absorbed by the platform, and it does not incur overhead for absorbing the heterogeneity. But in this case, there is an overhead of about 330 msec for discovering required components. Note that this overhead can be shortened by changing parameter values.