The Bio-Networking Architecture
Bi-weekly report #4 (July 22, 2002): Flexible Interface Definition for Service Components

PI: Tatsuya Suda (suda@ics.uci.edu)
University of California, Irvine
http://netresearch.ics.uci.edu/bionet/

Introduction

In the Bio-Networking Architecture, complex applications emerge through autonomous interactions of multiple service components (called "cyber-entities"), similarly to individuals cooperating to achieve a complicated task in the real biological world. This concept of composing an application through an autonomous interaction of multiple service components at runtime is called Dynamic Service Composition.

Dynamic Service Composition reduces development time as a large-scale complex application can be autonomously built from simpler service components. It also provides flexibility and adaptability to applications. For example, depending on a specific type of a display device available to a user (e.g. small PDA display or full color PC display), Dynamic Service Composition allows creation of applications using a service component that matches the capability of the display device (e.g., simple interface for a PDA display, colorful and high resolution interface for a PC display). In addition, it is also possible to use service components based on various resource constraints and system configurations (e.g., CPU speed, memory size, network bandwidth) in dynamically creating applications so that applications functionality and performance adapt to the context.

Since service components may be designed and developed independently by different designers, it is important to ensure that such independently-developed service components can communicate and interact to collectively provide an application. The PI is currently developing a set of specifications (referred to as the Flexible Interface Definition in this report) that defines interface of a service component in a flexible manner so that service components with different interfaces can communicate and interact with each other. The following describes the progress we made to date in describing interface of service components.

New Achievements

The design of Flexible Interface Definition has been proposed, and a measurement using Web Service has been conducted to verify the effectiveness of Flexible Interface Definition.

The PI has also recently published works regarding the current design and effectiveness of the proposed Flexible Interface Definition for Dynamic Service Composition:

- In a poster presentation entitled "Loose Interface Definition: An Extended Interface Definition for Dynamic Service Composition" at the First Annual Symposium on Autonomous and Intelligent Networked Systems Symposium

Flexible Interface Definition

In the proposed Flexible Interface Definition, an interface of a service component is defined as a set of elements, namely, name, type, and value. This approach of defining an interface as a set of elements is similar to the method definition in object-oriented languages and the function definition in procedural languages, where a method (function) is defined as a set of variable definitions (arguments).

In the Flexible Interface Definition, the following three techniques are used. First, Flexible Interface Definition allows each element to have multiple names so that multiple synonyms represent the same content. For instance, a human-being can easily understand that the 'surname' and 'last name'

represent the same meaning. However, machines using simple keyword matching cannot. Flexible Interface Definition allows multiple names (e.g., 'surname' and 'last name') to be associated with a single element so that they are recognized as synonyms representing the same content. Second, Flexible Interface Definition supports optional elements so that two service components having similar (but not exactly the same) interfaces can communicate. For example, using Flexible Interface Definition, a white page service component which returns first and last names of a person can communicate with a biographic dictionary service component which accepts first, middle and last names of a person, as the middle name is defined as an optional element. Third, Flexible Interface Definition disregards the order of the elements in the interface when comparing two interfaces, because a set of elements can be, and often are, defined in different orders. For example, a name of a person can be described either in the order of 'last name, first name' or in the order of 'first name, last name.' Flexible Interface Definition ignores the order of the elements (e.g., last name, first name) and considers representations with different element orderings equivalent. The three techniques explained above increases flexibility of interface and improve the possibility of communication between two service components that are implemented individually by two different designers.

In order to verify the effectiveness of Flexible Interface Definition that the PI is currently developing, the PI has conducted a simple measurement using existing Web Services. Web Service technology refers to a set of protocols for a service component to discover other service components and communicate with them via XML/HTTP. In the measurement, the number of pairs of Web Services (i.e., service components) that can communicate is compared without and with Flexible Interface Definition applied onto their interfaces. The measurement results shown in Figure 1 illustrate that applying Flexible Interface Definition doubles the number of pairs of Web Services that can communicate. This result indicates that Flexible Interface Definition is effective to increase interface flexibility.

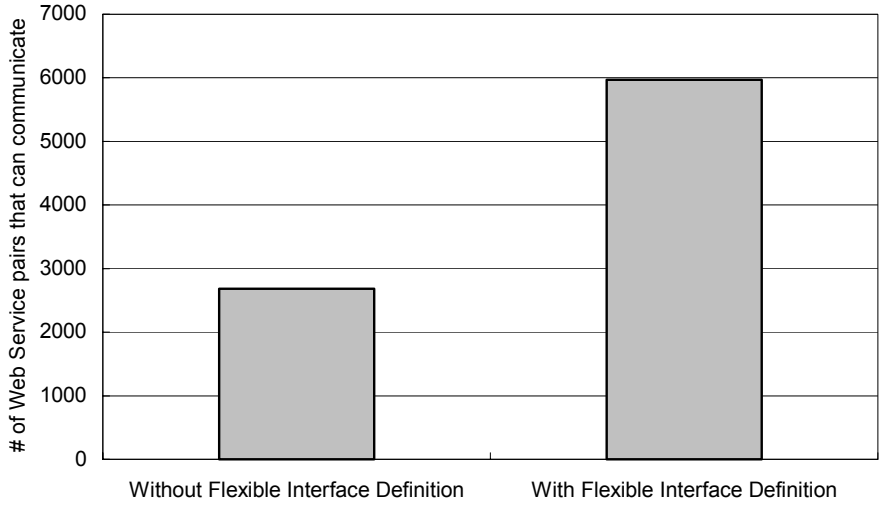

Figure 1: Web Services without and with Flexible Interface Definition applied