An Efficient Service Discovery Method and its Application

Shuiguang Deng, Zhejiang University, China
Zhaohui Wu, Zhejiang University, China
Jian Wu, Zhejiang University, China
Ying Li, Zhejiang University, China
Jianwei Yin, Zhejiang University, China

ABSTRACT

To discover services efficiently has been regarded as one of important issues in the area of Service Oriented Computing (SOC). This article carries out a survey on the issue and points out the problems for the current semantic-based service discovery approaches. After that, an information model for registered services is proposed. Based on the model, it brings forward a two-phase semantic-based service discovery method which supports both the operation matchmaking and operation-composition matchmaking. The authors import the bipartite graph matching to improve the efficiency of matchmaking. An implementation of the proposed method is presented. A series of experiments show that the method gains better performance on both discovery recall rate and precision than a traditional matchmaker and it also scales well with the number of services being accessed.

Keywords: Bipartite Graph Matching, Service Discovery, Service Matchmaking, Web Service

INTRODUCTION

Background

Nowadays, as the Internet has become the main platform on which enterprises carry out businesses globally, the environment of enterprise applications will be characterized by frequently changing market demands, time-to-market pressure and fierce competition. Therefore, it requires that the enterprise business systems should provide more flexibility than present-day systems can afford. The key to tackle this challenge completely is to utilize a kind of novel software system architecture which is required to be distributed, loose-coupled and reconfigurable. Fortunately, these requirements can be best addressed by Service-oriented Architecture (SOA).

SOA is an architectural style whose goal is to achieve loose coupling among interacting services and to build software systems by

DOI: 10.4018/jwsr.2009071305
composing services (Papazoglou & Georgakopoulos, 2003). It provides greater flexibility and agility while allowing business systems to use heterogeneous resources efficiently and effectively. Web services technology has been regarded as the preferred implementation vehicle for SOA. A Web service is a software entity that supports interoperable application-to-application interaction over Internet. At present, the accelerating creation and use of Web Services in enterprises informatics is a major trend (Kalogeras, Gialelis et al., 2006). Thus, more and more Web services are published in Internet by enterprises to accelerate the cooperation with their partners. For example, in the scenario of supply chain management, a manufacturer receives an order to deliver some merchandise to a retailer. In order to accomplish this business, the manufacture finds possible suppliers and selects the best available service provided by one supplier. However, due to the highly distributed and dynamic environment, Web services may be located at different enterprises and come and leave at any time without prior warning. In that context, no one is likely to have the detailed knowledge of all Web services in advance. As a result, one of great challenges is how to discover the suitable Web services accurately and quickly. Thus, service discovery, which aims at retrieving services advertised in a repository that match a user’s goal, has allured much attention both from industry and academy.

**Problem**

Currently, there is a good body of work on service discovery. Among the work, the effort of semantic Web service from the semantic Web community has been regarded as the most promising way to retrieve services in an accurate and automatic way. Based on related ontology languages and inference engines, semantic Web services provide machine understandable descriptions of what services do and how they achieve their goals (McIlraith, Son et al., 2001). Semantic Web service discovery utilizes semantic matchmaking to check whether an advertised service satisfies a user’s request by computing the similarity degree between the description for the service and the one for user’s request. If the similarity degree exceeds some threshold value specified by the user, the service is returned as a candidate for the user. Due to the accurate and unambiguous description of a service’s functionalities and a user’s request both enhanced by semantics, semantic Web service discovery tends to get good recall rate and precision. However, they can achieve even better performance with the following two factors taken into consideration.

1. **Not all inputs are compulsory for each output**

According to the most frequently cited semantic Web service matchmaking algorithm proposed by Paolucci, Kawamura et al. (2002), an advertised service matches a request if the request provides all the inputs (possibly more) needed by the service while the service generates all the outputs (possibly more) needed by the requester. In other words, a successful matching demands that the request provides all the inputs of the advertised service to get any output of the service. This requirement has been widely accepted by most semantic Web service matchmaking methods.

However, the successful matching criteria are too strict and may lead to some unwanted situations. Consider an abstract scenario between an advertised service $S$ and a request $R$, where $S$ has two inputs ($a$ and $b$) and two outputs ($o$ and $p$), and $R$ specifies one input ($a$) and one output ($o$). According to the above successful matching criteria, $S$ does not match $R$ as $R$ cannot provide the input $b$. But for $S$, maybe the input $b$ is optional for the output $o$. In this case, $S$ should be a candidate service for $R$ but it is excluded. Consider a real weather report service (http://www.webservicex.com/global-weather.asmx?WSDL). It has an operation named `GetWeather` that returns `WeatherResult` on receiving a `CityName` and a `CountryName`. However, this operation also serves well if the
An Adaptive Approach to Optimizing Tradeoff Between Service Performance and Security in Service-Based Systems
[www.igi-global.com/article/adaptive-approach-optimizing-tradeoff-between/55237?camid=4v1a](www.igi-global.com/article/adaptive-approach-optimizing-tradeoff-between/55237?camid=4v1a)

Data Science and Big Data Analytics in Financial Services: A Case Study