The Journal of Systems and Software 93 (2014) 217–228

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

A trustworthy QoS-based collaborative filtering approach for web service discovery

Szu-Yin Lin [a], Chin-Hui Lai [a,*], Chih-Heng Wu [b], Chi-Chun Lo [b]

[a] Department of Information Management, Chung Yuan Christian University, Taoyuan County, Taiwan
[b] Institute of Information Management, National Chiao Tung University, Hsin-Chu, Taiwan

ARTICLE INFO

Article history:
Received 13 September 2012
Received in revised form
19 November 2013
Accepted 23 January 2014
Available online 3 February 2014

Keywords:
Service selection
Service discovery
Quality of Service (QoS)
Collaborative filtering

ABSTRACT

Many network services which process a large quantity of data and knowledge are available in the distributed network environment, and provide applications to users based on Service-Oriented Architecture (SOA) and Web services technology. Therefore, a useful web service discovery approach for data and knowledge discovery process in the complex network environment is a very significant issue. Using the traditional keyword-based search method, users find it difficult to choose the best web services from those with similar functionalities. In addition, in an untrustworthy real world environment, the QoS-based service discovery approach cannot verify the correctness of the web services' Quality of Service (QoS) values, since such values guaranteed by a service provider are different from the real ones. This work proposes a trustworthy two-phase web service discovery mechanism based on QoS and collaborative filtering, which discovers and recommends the needed web services effectively for users in the distributed environment, and also solves the problem of services with incorrect QoS information. In the experiment, the theoretical analysis and simulation experiment results show that the proposed method can accurately recommend the needed services to users, and improve the recommendation quality.

1. Introduction

Due to recent advances in ubiquitous and distributed computing, many distributed network services which process a large quantity of data and knowledge are available in the cloud computing network environment, and provide useful data and applications to users based on Service-Oriented Architecture (SOA) and web services technology. Because some service providers may provide services with similar functionalities, data, and knowledge, it is difficult for users to find relevant and the best services from many different types of services. Therefore, recommending an appropriate web service to fulfill a user's requirements from a large number of services is a great challenge.

In SOA, many studies have focused on the service discovery problem (González-Valenzuela et al., 2011; Koloniari and Pitoura, 2004). The traditional method of web service discovery is Universal Description Discovery and Integration (UDDI), which works as an intermediary between service providers and users. In UDDI systems, service providers can publish meta-information of services for locating web services, while users can send their requirements as queries to search for required services. Although the OASIS UDDI Specification Technical Committee voted to complete UDDI's final standard version in late 2007 and close the project of UDDI, UDDI systems are still most commonly found inside companies or inter-companies, where they are used to dynamically bind client systems to implementations. Because the number of services is increasing constantly, methods of service selection and service discovery are necessary for improving search efficiency. However, the methods of service discovery provided by UDDI can only satisfy users' functional requirements (Xu et al., 2007; Yan and Piao, 2009), and ignore users' non-functional requirements. Generally, functional requirements define what a system is supposed to do, whereas non-functional requirements define how a system is supposed to be. In recent years, non-functional requirements are also called quality attributes. Quality of Service (QoS) is usually employed to describe non-functional attributes (e.g., response time, availability, throughput, price, cost, reliability, etc.) of web services (Raj and Sasipraba, 2010). The QoS values are user-dependent because users may give different values based on their demands. Consequently, QoS can be applied in solving the problem of service selection when choosing between services which have similar functionalities (Canfora et al., 2008; Ko et al., 2008).

Providing the QoS information of web services could help users to select an appropriate service from many with the same functionality. However, there are some problems for service discovery in QoS. Because some QoS information in UDDI may be untrustworthy

* Corresponding author. Tel.: +886 3 2655413; fax: +886 3 2655499.
E-mail addresses: chlai@cycu.edu.tw, chinhui.lai@gmail.com (C.-H. Lai).

0164-1212/$ – see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jss.2014.01.036

(such as outdated information of services, or unavailable services), UDDI-discovered web services approach may not meet users' non-functional requirements. Essentially, traditional QoS-based query methods cannot verify the correctness of QoS information. Some methods, i.e., Web Services Level Agreements (WSLA) (Keller and Ludwig, 2003), reputation-enhanced methods (Ali et al., 2006; Maximilien and Singh, 2005; Wishart et al., 2005; Xu et al., 2007), and collaborative filtering methods (Manikrao and Prabhakar, 2005; Zheng et al., 2009) are proposed. WSLA is a standard for service level agreement compliance monitoring of web services. It allows users to specify the QoS performance metrics associated with a web service application (such as service selection or discovery), desired performance targets, and actions that should be performed when QoS performance is not met. The reputation-enhanced methods aggregate the QoS ratings of services as web service reputation to reflect the overall QoS for ranking the services during the service discovery process. Such service reputation also provides a general estimate for the reliability of service providers. Considering the web service reputation may increase the probability of finding the reliable services.

Moreover, collaborative filtering methods are widely used in many application domains. They are generally classified into user-based and item-based collaborative filtering methods based on similar users and similar items respectively to predict the values of items. For the web service discovery, the user-based collaborative filtering methods employ historical QoS information obtained from users who have similar QoS experience on the use of web services, and then predict the QoS performance of a web service for an active user. Similarly, the item-based methods use the similarity between the web services instead of the service users to predict a QoS value of a web service. The mentioned approaches have been proposed to solve the QoS incorrectness problem in service discovery, but they still cannot completely solve the untrustworthy QoS problem. Because of WSLA, it is impossible for users to sign the WSLA contract of every service provider when they query through a UDDI registry center. For reputation-enhanced methods, the reputation values of services may be incorrect because of malicious user feedback, malicious attacks, or inaccurate ratings given by users. Moreover, traditional collaborative filtering methods not only have the same problems as reputation-enhanced methods, but may also be unable to recommend the needed web services without users' rating data.

Therefore, this study aims to solve the following problems for service discovery in QoS: (1) QoS-based query methods cannot verify the correctness of the QoS information. (2) When users' ratings of web services are collected in query methods, it is necessary to deal with the problems of malicious users and malicious attacks. (3) Because feedback scores from users for web services are subjective, inaccuracies or errors in computing the reputation values of services may occur. (4) Traditional collaborative filtering approaches cannot recommend services corresponding to users' QoS requirements.

This paper proposes a trustworthy two-phase web service discovery mechanism based on collaborative filtering and QoS information. The first phase is used to filter out services with incorrect QoS values, while the second phase is employed to recommend services to users according to their non-functional requirements, which describe the correct QoS information for service consumers. Therefore, the research objectives in this paper are: (1) to recommend web services corresponding to services' functionalities and users' QoS requirements, (2) to filter out web services with incorrect QoS parameters, and (3) to propose a trustworthy platform for service discovery.

The rest of this paper is organized as follows: Section 2 describes the related works, including web services architecture, QoS storage for web services, QoS-based service discovery methods, and recommendation methods. In Section 3, we propose a trustworthy two-phase web service discovery mechanism. The details of our method are clearly illustrated. Section 4 shows the implementation of our proposed method for the experiment, and the evaluation of the experimental results. Finally, our conclusion and future works are discussed in Section 5.

2. Related works

In this section, we will discuss the related works on the basic framework and development of web services, QoS storage for web services, QoS-based web service discovery methods, and collaborative filtering for web service recommendations.

2.1. Web services architecture

The World Wide Web Consortium (W3C) describes web services as a software system to support interoperable machine-to-machine interaction over a network (Haas and Brown, 2004). For the B2C environment, the development of web services focuses on individual users who can only use the limited functions provided by a single computer. To create an operation process between the services within an enterprise and its partners in a B2B environment, the concept of Service-Oriented Architecture (SOA) is proposed. Thus, software applications, not only services, can communicate with each other based on SOA. These applications can exchange and integrate information based on SOA and XML-based web service techniques. W3C (World Wide Web Consortium) defines web services as an application that can use XML to describe a request and utilize URI (Uniform Resource Identifier) to identify the interface of the application and binding methods. Thus, the message between applications is transferred over the Internet using XML and standard Internet techniques. That is, the web service is regarded as a software component, and applies the HTTP protocol and open data format, e.g., XML, WSDL and SOAP, to communicate with other applications. Because web services technology is based on the Internet, it provides an effective means of communication. Even though it can be implemented by using different programming languages and platforms, it is possible to resolve the integration problems and difficulties in distributed systems. There are 3 kinds of roles in Service-Oriented Architecture, including service provider, service requester, and UDDI registry (Papazoglou, 2003). The service mainly provides the development of web services, descriptions of creating the web service, and the service registration in UDDI. UDDI, which is a mediator between service provider and service requester, receives registration requests from a service provider, and deals with queries from a service requester. Thus, a service requester queries and searches a specific service in UDDI when he has some demands. The process for obtaining a web service is as follows.

(1) *Registry*: A service provider provides the needed service information to UDDI.
(2) *Publish*: UDDI accepts the registration of a service by a service provider and informs other UDDIs.
(3) *Find*: A service requester sends a query requirement to UDDI.
(4) *Response of a service query*: UDDI replies with a query result to the service requester, and retrieves the service WSDL.
(5) *Invoke*: Based on the description of the WSDL, the service requester sends a request to a service provider.
(6) *Bind*: The service provider replies with the result of the request.

According to the above process, the "Find" step is important in service discovery to help service requesters discover the services they need. In the following sections, this paper will investigate and discuss the details of the problem associated with service discovery.