Journal of King Saud University – Computer and Information Sciences (2015) 27, 323–333

King Saud University

Journal of King Saud University – Computer and Information Sciences

www.ksu.edu.sa
www.sciencedirect.com

Bootstrapping quality of Web Services

Zainab Aljazzaf *

Department of Information Science, College of Computing Sciences and Engineering, Kuwait University, Kuwait

Received 18 June 2014; revised 13 November 2014; accepted 9 December 2014
Available online 29 June 2015

KEYWORDS

Services;
Web Services;
Quality of services;
Bootstrapping;
Service Oriented Architecture

Abstract A distributed application may be composed of global services provided by different organizations and having different properties. To select a service from many similar services, it is important to distinguish between them. Quality of services (QoS) has been used as a distinguishing factor between similar services and plays an important role in service discovery, selection, and composition. Moreover, QoS is an important contributing factor to the evolution of distributed paradigms, such as service-oriented computing and cloud computing. There are many research works that assess services and justify the QoS at the finding, composition, or binding stages of services. However, there is a need to justify the QoS once new services are registered and before any requestors use them; this is called bootstrapping QoS. Bootstrapping QoS is the process of evaluating the QoS of the newly registered services at the time of publishing the services. Thus, this paper proposes a QoS bootstrapping solution for Web Services and builds a QoS bootstrapping framework. In addition, Service Oriented Architecture (SOA) is extended and a prototype is built to support QoS bootstrapping. Experiments are conducted and a case study is presented to test the proposed QoS bootstrapping solution.

1. Introduction

To build a distributed application from Web Services, the application developer, or service requestor, may need to select Web Services from different service providers. Because there are many Web Services with similar *functionalities*, service requestors need to differentiate among them. The only differentiating factor between similar Web Services may be their *non-functional* properties, which can be considered as criteria for service selection. Quality of services (QoS) has been used as a non-functional property for selecting services (Papazoglou et al., 2006; Maximilien and Singh, 2004; Dragoni, 2009; Ying-Feng et al., 2006; Huhns and Singh, 2005; Zhang et al., 2012; Kalepu et al., 2003; Kim and Doh, 2007; Liu et al., 2004; Zheng et al., 2014; Rajeswari et al., 2014).

For example, one approach in service selection (Ran, 2003) involves the case where the Web Service registry can capture the QoS provided by the service provider and the QoS required by the service requestor and accordingly match the two while discovering the service, to select the best match from services with similar functionality. In this scenario, a service requestor may need a service that has a low response time, which is considered a QoS property of the service. Therefore, a service with

* Tel.: +965 24633223.
E-mail address: dr.zainab@ku.edu.kw.
URL: http://www.isc.ku.edu.kw/.

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.jksuci.2014.12.003
1319-1578

a lowest response time will be selected by the requestor from many similar services with different response times.

Many studies have been conducted to examine QoS compliance by monitoring Web Services or collecting quality ratings from the users (Papazoglou et al., 2006; Kim and Doh, 2007; Zhang et al., 2012; Liu et al., 2004; Huhns and Singh, 2005; Kalepu et al., 2003; Zheng et al., 2014). Thus, considering the issues related to the Web Services and distributed paradigms, QoS is an important contributing factor to the evolution of distributed paradigms.

Service providers may register their Web Services claiming the services' QoS. However, the Web Service brokers need to justify the conformance of the QoS to the published specification. Many studies have been conducted to justify the QoS at the time of selecting the services or at run time of services (Maximilien and Singh, 2004; Dragoni, 2009; Zhang et al., 2012; Kalepu et al., 2003; Kim and Doh, 2007; Liu et al., 2004; Zheng et al., 2014). To the best of our knowledge, few attempts were done to justify the QoS once new services are registered and before any requestors use them. Therefore, there is a need to ensure the QoS, especially for new services (new comer) that no service consumer has tried using before, and for which the justification of the QoS measurements is not available a priori, so called Bootstrapping QoS. Bootstrapping is the process of evaluating the QoS of the newly registered services at the time of publishing the services.

In this paper, a solution for bootstrapping QoS is introduced that assesses the QoS attributes for the newly registered Web Services. The main contribution lies in the automated approach for QoS bootstrapping at the Publish time and before any requestor requests the Web Service. As a result, the justification of QoS for new Web Services will be available at the time of publishing the Web Services, thus:

- there is no need to test the Web Services at the time of Finding the service;
- the Finding operation will be faster;
- increases the opportunity for new Web Services to be selected by requestors;
- increases the level of trust in such services.

Accordingly, this work proposes a QoS bootstrapping solution for Web Services that includes a QoS model, QoS bootstrapping framework, SOA extension to support bootstrapping QoS, and prototype. Consequently, experiments are conducted to evaluate the bootstrapping solution.

The rest of the paper is organized as follows. The background is presented in Section 2. Section 3 presents the related work. The proposed QoS bootstrapping technique is included in Section 4 and it covers QoS model and QoS bootstrapping framework. Section 5 covers the SOA extension to support bootstrapping QoS. The prototype is presented on Section 6. The experiment, evaluation, and case study are demonstrated in Section 7. Section 8 concludes the paper.

2. Background

This section provides the background about Web Services and QoS and service-oriented computing paradigm, as given below.

2.1. Web Services and quality of service

The development of a distributed software system requires the interaction of services and the use of resources from diverse organizations throughout the Web. A service is "a discrete unit of business functionality that is made available through a service contract" (Rosen et al., 2008), which includes a service interface, service documents, service policies, and QoS. Services perform functions from simple requests to complicated business processes.

Services can be implemented using Web Service technology. Web Services are an emerging technology that enables applications running from different machines over the Web to integrate and exchange data regardless of their platform, hardware, operating system, and languages (Papazoglou, 2012).

A Web Service is defined by the World Wide Web Consortium (W3C) as "a software application identified by a URI, whose interface and binding are capable of being defined, described and discovered by XML artefacts and supports direct interactions with other software applications using XML based messages via Internet-based protocols" (Ferris et al., 2004). Similarly, IBM defines a Web Service as "self-contained, self-described, dynamically discovered applications with Internet-based interfaces" (Yuan and Long, 2002).

Accordingly, a Web Service is a new breed of Web applications, which is modular and can be published, located, and invoked across the Web. Once a Web Service is deployed, other applications can discover and invoke it. Web Services are loosely coupled, platform-neutral, reusable, and distributed software components (Yuan and Long, 2002).

Web Services are based on common standards, such as Extensible Markup Language (XML), and existing technologies, such as Hypertext Transfer Protocol (HTTP). The key to Web Services' success is the open standards that facilitate the interoperability among different parties (Yuan and Long, 2002). Web Services technology's main protocols include Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL), and Universal Description, Discovery, and Integration (UDDI) (Papazoglou, 2012).

SOAP is an XML-based standard messaging protocol, using HTTP as a means of transport and for circumventing the firewalls. WSDL is the service representation language used to describe the interface of and access to Web Services. This description includes the operations and parameters, location, and invocation protocol of the Web Services. UDDI is a cross-industry directory standard for the description, publication, and discovery of Web Services. This standard stores the Web Service interfaces described by WSDL, categorizes Web Service information, and allows searching the directory for Web Services.

Regarding quality and QoS, the international quality standard ISO 8402 (part of the ISO 9000) describes quality as "the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs". Hoyle (2005) defines quality as "the degree to which a set of inherent characteristics fulfills a need or expectation that is stated, general implied or obligatory". The author elaborates that "quality is thought of as conformance to specification regardless of whether the specification actually meets the needs of the customer or society" Hoyle (2005). The W3C describes the