Chapter XIV
Enhancing Web Service Discovery and Monitoring with Quality of Service Information

Christian Platzer
Vienna University of Technology, Austria

Florian Rosenberg
Vienna University of Technology, Austria

Schahram Dustdar
Vienna University of Technology, Austria

ABSTRACT

Web services provide a fundamental technology for developing service-oriented systems by leveraging platform-independent interface descriptions (WSDL) and a flexible message encoding (SOAP). Beside the functional description, quality of service (QoS) issues are currently not part of the Web service standards stack, although they provide valuable metadata of a Web service such as performance, dependability, security, or cost and payment. This additional information can be used to greatly enhance service discovery, selection, and composition. As a result of the latest research that is dedicated to this area, this chapter deals with the various ways of describing, bootstrapping, and evaluating QoS attributes. A strong focus is laid on client-side QoS assessment and the arising problems. Furthermore, a method to analyze Web service interactions by using our evaluation tool and extracting important QoS information without any knowledge about the service implementation will be presented and thoroughly explained. Usually, taking performance measures for a specific Web service requires access to the service implementation or at least the server machine where it is hosted. This chapter will address a way to bootstrap the most important performance and dependability values form the client’s perspective, and therefore overcoming these restrictions.
INTRODUCTION

QoS descriptions for Web services are gaining importance and are heavily investigated by different research groups. Their availability is a fundamental requirement to solve many problems researchers are currently encountering. These problems include dynamic composition, search and discovery of Web services, and performance-based selection of services.

Currently, the most severe drawback when dealing with Web services is a common lack of quality of service (QoS) attributes and their present values. These descriptions are necessary to appraise the value of a Web service in different respects. On the one hand, they can describe performance-related characteristics like availability, latency, and response time, and on the other hand, they are utilized to provide the necessary values for provider-specific settings like cost, security requirements, and similar values. With the current method to specify Web services by using WSDL descriptions, no technique to attach both, provider- and consumer-specific QoS values, exists. For the provider side it is simply because no common way to attach this valuable information to either a WSDL description or a service registry has yet been established.

The case is different for consumer-specific values, especially the performance-related aspects, because it simply does not make sense in most cases. A provider that guarantees a response time of 20 milliseconds for a Web service will run into serious trouble to fulfill this guarantee when a client operates on a slow modem connection, for instance. Furthermore, by appointing their own performance values, the providers would be forced to announce best-case values or even tempted to declare better values than they can possibly produce. Apart from the aspects mentioned above, QoS parameters related to performance are required by various researchers because they can be used to optimize workflows or enhance the results of search engines. For the time being, those very specific measurements must be assessed by the client invoking the service or a third party.

In this chapter, a framework is introduced which provides the possibility to assess QoS attributes for a given set of Web services. The approach is automated and is able to bootstrap and constantly monitor QoS parameters for existing services that are currently lacking such valuable descriptions. A tool, based on this approach, allows categorizing service repositories according to the most important characteristics, finally enabling QoS-driven development. Furthermore, existing standards are used to attach the bootstrapped values to a document description. Not all of these standards are applicable and can be used for the purpose of QoS propagation, while others have to be adopted to fit in the requirements discusses in the following chapter. First of all, we will discuss a proper QoS model which encompasses the most relevant nonfunctional properties of a Web service.

QoS MODEL

This section describes the QoS model and how it is used to express quality attributes of Web services. It is important to keep in mind that most of the presented attributes are dynamic and site-dependent. As a result, the produced values cannot be seen as global attributes, but as site-specific statistics with a strong local context. This is an intended behavior, because the parameters, influenced by the local conditions, increase the significance of the whole value. We assume two Web services, for example, named A and B, with the same implementation and the same hardware. The Web service consumer is located at a remote place where the routing of the actual IP packets is the only difference between the two services. Therefore, A may respond faster than B in this case, while B could be the faster service when queried from another place.
Related Content

Towards Usable Application-Oriented Access Controls: Qualitative Results from a Usability Study of SELinux, AppArmor and FBAC-LSM
www.igi-global.com/article/towards-usable-application-oriented-access/64346?camid=4v1a

Supply Chain Disruptions and Best-Practice Mitigation Strategies
www.igi-global.com/article/supply-chain-disruptions-best-practice/70232?camid=4v1a

Identification and Authentication for RFID Systems
www.igi-global.com/chapter/identification-authentication-rfid-systems/75514?camid=4v1a

Computational Complexity Analysis for a Class of Symmetric Cryptosystems Using Simple Arithmetic Operations and Memory Access Time
www.igi-global.com/article/computational-complexity-analysis-class-symmetric/78530?camid=4v1a