A Mobile Agent-Based Tool Supporting Web Services Testing

Jia Zhang

Abstract The Web services technology has received significant momentum in recent years, because it allows people to easily utilize and integrate existing software applications to rapidly create new business services. However, how to ensure the trustworthiness of a Web services-oriented system remains a big challenge. One critical issue is how to test a Web service in an effective and efficient manner. In this paper, we report our design and development of a novel mobile agent-based method oriented to Web services testing. We seamlessly integrate an existing testing tool (HP LoadRunner) with a mobile agent technology (IBM Aglet) to build a practical environment for testing Web services. We also report an automatic test case generation algorithm, which analyzes published Web service interfaces and creates boundary value-based, fault injection-equipped test cases.

Keywords Web services testing · Mobile agent · Testing platform

1 Introduction

A Web service refers to a programmable Web application with a standard interface and is universally accessible using standard network protocols. Its interoperability provides users a practical way to quickly create a new service by combining existing services. It is one central reason why this emerging technology has caught significant momentum from both academia and industry in the recent years.

However, the wide adoption of the Web services technology has encountered a bottleneck, because it remains a big challenge as how to ensure the trustworthiness of a Web services-
based system. To tackle this challenge, one key prerequisite is how to test a Web service, especially from client side. The reason is obvious: eventually a client will not really trust a system unless he/she can fully test it.

The last 50 years of software development history has established an independent research branch called software testing, which contains a wealth of theories, technologies, methodologies, and tools to guide the verification process of a software product. In the recent years, significant efforts have been conducted to adapt the traditional testing technologies toward testing Web applications remotely over the Internet [1]. A number of tools have been developed. Among them, HP’s LoadRunner [2] is a well-known performance and load testing software for examining system behaviors. In addition to testing traditional software products deployed in a local environment, LoadRunner has added some features to automate Web application testing.

However, these Web application-oriented testing technologies may not be suitable for measuring and testing Web services. The model of Web services poses significant challenges on Web application testing due to its unique nature and properties. First, unlike screen-centric Web applications, Web services are Web components only accessible via interfaces published in standard Web services-specific interface definition languages (e.g., WSDL) and accessible via standard network protocols (e.g., SOAP). Therefore, how to design test cases for a Web service using its limited information exposed remains a challenge. Second, one major goal of adopting the Web services technology is to dynamically compose existing Web services as components to quickly construct a new service. Therefore, Web services testing usually imply that multiple service components have to be tested in an efficient and lightweight (i.e., without maintaining dedicate network connections throughout a testing process) manner. Third, how to mitigate the overhead caused by the Web services-specific transport protocols (e.g., SOAP) deserves special attention [3].

The issue of Web services testing has caught significant attention in the recent years. For example, the latest version of LoadRunner (9.0) is also moving toward testing Web services. However, the current version of LoadRunner can only test single Web services with all test cases prepared ahead of time. Meanwhile, it requires to keeping the network connection throughout the entire testing process, which is not efficient and may not be even possible in the wireless network. Our previous work applies the mobile agent technology on Web services testing for higher performance and reliability [4]. A mobile agent is a composition of computer software and data that are able to travel from one computer to another on the network, and autonomously and automatically continue their execution on the destination computer [5–7]. IBM Tokyo Research Lab publishes Aglets Software Development Kit (ASDK) [8] that is an open source mobile agent development package written in Java. Our mobile agent-based Web services testing method can also handle testing multiple Web services with predefined testing scenarios and testing paths. However, our previous work requires that users write test code and embed them into mobile agents before they migrate.

This research is a continuous effort, aiming to build a practical method for Web services testing. Our contribution is two-fold. First, we seamlessly integrate an existing automatic software testing tool and the mobile agent technique to create an effective and efficient testing platform to facilitate Web services testing processes. In this project, we integrate IBM Aglet technology and HP LoadRunner as a proof of concept. On one hand, we apply the mobile agent technology to enhance the ability of a popular software testing tool (LoadRunner) on Web services testing; on the other hand, we integrate our mobile agent-based Web services testing approach with an existing tool to make it more practical and powerful. Second, we propose an automatic test case generation algorithm, which automatically analyzes published