A Metamorphic Relation-Based Approach to Testing Web Services Without Oracles

Chang-ai Sun, University of Science and Technology Beijing and Chinese Academy of Science, China
Guan Wang, University of Science and Technology Beijing, China
Baozhong Mu, University of Science and Technology Beijing, China
Huai Liu, Swinburne University of Technology, Australia
ZhaoShun Wang, University of Science and Technology Beijing, China
T. Y. Chen, Swinburne University of Technology, Australia

ABSTRACT

Service Oriented Architecture (SOA) has become a major application development paradigm. As a basic unit of SOA applications, Web services significantly affect the quality of the applications constructed from them. In the context of SOA, the specification and implementation of Web services are completely separated. The lack of source code and the restricted control of Web services limit the testability of Web services, and make the oracle problem prominent. In this context, can one alleviate the test oracle problem, or effectively and efficiently test such Web services even without oracles? It is an important issue which has not been yet adequately addressed. To address the challenge of testing Web services, the authors propose a metamorphic relation-based approach to testing Web services without oracles. The proposed approach leverages so-called metamorphic relations to generate test cases and evaluate test results. To make the proposed approach practical and effective, the authors proposed a framework taking into account the unique features of SOA, and developed a prototype which partially automates the framework. Three case studies are conducted to validate the feasibility and effectiveness of the proposed approach. The work presented in the paper not only alleviates the test oracle problem of testing Web services, but also delivers an effective and efficient test technique without oracles.

Keywords: Metamorphic Testing, Service Oriented Architecture (SOA), Software Testing, Test Oracle, Web Services

INTRODUCTION

Service Oriented Architecture (SOA) has been evolving as a mainstream software development paradigm where Web services are basic elements (Papazoglou, Traverso, Dustdar, & Leymann, 2008). A Web service often implements an application or part of an application, and is able to make a set of operations available to its consumers through the Web Service Descrip-
In the context of SOA, Web services can be implemented and owned by one organization, and published as an independent resource that is consumed by other organizations. To implement complex applications, Web services have to be loosely orchestrated to fulfill a business goal (Peltz, 2003; Sun, Khoury, & Aiello, 2011).

Let us consider an e-bookstore system constructed by several Web services. Among them, a Web service is responsible for the electronic payment. Usually, such a Web service is developed and owned by a third-party organization, such as a software company, a bank, or an independent commercial office. Due to the fact that the consumer (i.e., the e-bookstore system) can access the Web service only through its description (namely WSDL) and cannot look into the source code of the Web service, this results in some inconsistency issues. For example, some faults may exist in the implementation regardless how many efforts are spent on testing. Also, the service owner may update the implementation due to changes of the payment policy (specification); however, the service consumer may not realize the changes happening to the implementation or specification. This may result in the situation where the consumer invokes the latest version of implementation while holds the old version of specification. All these cases bring us one question, namely “how should we assure the consistency between implementation and specification of Web services?"

Software testing provides a practical and feasible approach to the question. The basic idea of testing is to select some test cases for executing program under test, and then evaluate the output against the expected correct output (namely oracle). If the actual output equals to the oracle, such a test succeeds; otherwise the test detects a fault. In the context of SOA, new unique features pose challenges for testing Web services. For instance, the specification and implementation of Web services are completely separated, and often service consumers cannot access the source codes of Web services they invoke. The lack of source code and the restricted control of Web services not only limit the testability of Web services, but also make white-box testing techniques inapplicable. One may argue that the lack of source codes is also present in the use of any normal software library, but they are totally different in that the implementation of a Web service can be written in any programming language and its specification is written in the XML style. Moreover, the test oracle problem, which means in some situations it is impossible or practically too difficult to decide whether the program outputs on test cases are correct (Weyuker, 1982), is even amplified. In the example of the electronic payment service, the consumer, in some situations, may not know exactly how much should be charged for a given input (i.e., book price). The problem becomes even worse when the payment involves charges for transfer between accounts or currency exchange. Testing Web service under SOA calls for new techniques (Bartolini, Bertolino, Elbaum, & Marchetti, 2009; Canfora & Penta, 2009; Farooq, Georgieva, & Dumke, 2008; Sun, 2011).

Although there exist many techniques for Web service testing in the literature (Bartolini, Bertolino et al. 2009, Bai, Lee et al. 2008, Lenz, Chimiak-opoka et al. 2007, Zhang and Qiu 2006), most of these existing testing techniques always assume the presence of test oracles and thus cannot address the oracle problem. Metamorphic Testing (MT) was first proposed by Chen,Cheung et al. (1998) as a testing approach without the need of oracles. The basic idea behind MT is to leverage the metamorphic property of Web services to generate test cases and evaluate test outputs. Metamorphic property is a kind of inherent properties of software, referring to that for a pair of test cases satisfying relation $R$, and then their corresponding outputs should also satisfy relation $R'$ during the testing, if there exist some tests where $R$ is satisfied while $R'$ is violated, then a fault is detected. In this way, one can employ MT to test programs without precise oracles. It has been shown that MT has successfully alleviated the test oracle problem (Dong, Xu, Chen, Nie, & Wang, 2009). However, new features of SOA prevent the traditional MT framework from testing Web services. Recall the e-bookstore...
Related Content

Pricing Utility Computing Services
[www.igi-global.com/article/pricing-utility-computing-services/3101?camid=4v1a](www.igi-global.com/article/pricing-utility-computing-services/3101?camid=4v1a)

A Metamorphic Relation-Based Approach to Testing Web Services Without Oracles
[www.igi-global.com/article/metamorphic-relation-based-approach-testing/64223?camid=4v1a](www.igi-global.com/article/metamorphic-relation-based-approach-testing/64223?camid=4v1a)

Using Distributed Semantic Catalogs for Information Discovery on Spatial Data Infrastructures
[www.igi-global.com/chapter/using-distributed-semantic-catalogs-information/51485?camid=4v1a](www.igi-global.com/chapter/using-distributed-semantic-catalogs-information/51485?camid=4v1a)
Web Services for the Global Earth Observing System of Systems
[www.igi-global.com/chapter/web-services-global-earth-observing/51494?camid=4v1a](www.igi-global.com/chapter/web-services-global-earth-observing/51494?camid=4v1a)