

Goal-based testing of semantic web services



M. Shaban Jokhio^a, Jing Sun^{a,*}, Gillian Dobbie^a, Tianming Hu^b

^a Department of Computer Science, The University of Auckland, New Zealand

^b Department of Computer Science, Dongguan University of Technology, China

ARTICLE INFO

Article history:

Received 8 December 2015

Revised 29 November 2016

Accepted 30 November 2016

Available online 1 December 2016

Keywords:

Formal methods

Model checking

Mutation analysis

Web Service Modeling Ontology

Semantic web services.

ABSTRACT

Context: Recent years have witnessed growing interests in semantic web and its related technologies. While various frameworks have been proposed for designing semantic web services (SWS), few of them aim at testing.

Objective: This paper investigates into the technologies for automatically deriving test cases from semantic web service descriptions based on the Web Service Modeling Ontology (WSMO) framework.

Method: WSMO goal specifications were translated into B abstract machines. Test cases were generated via model checking with calculated trap properties from coverage criteria. Furthermore, we employed mutation analysis to evaluate the test suite. In this approach, the model-based test case generation and code-based evaluation techniques are independent of each other, which provides much more accurate measures of the testing results.

Results: We applied our approach to a real-world case study of the Amazon E-Commerce Service (ECS). The experimental results have validated the effectiveness of the proposed solution.

Conclusion: It is concluded that our approach is capable of automatically generating an effective set of test cases from the WSMO goal descriptions for SWS testing. The quality of test cases was measured in terms of their abilities to discover the injected faults at the code level. We implemented a tool to automate the steps for the mutation-based evaluation.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

The past decade has witnessed a rapid progress in Web Services (WS), which provide an open platform with a set of XML-based standards. However, due to the fact that XML is only a syntax-based notation, it can not be used to document complicated contextual relationships among the web entities and be fully understood by software tools. To this extend, semantic web services (SWS) add meta data and semantics to the present WS technologies using the ontology description languages. Since the ontology associates the strict meaning of terms, it is easier for software tools to understand and more accurately process SWS.

While various frameworks have been proposed for the design and development of SWS, few of them aim at SWS testing. In this paper, we study the issue from an end user's perspective. Traditionally, a web service is written and provided by a web service

provider, together with the corresponding web service specification, e.g., the WSDL or OWL-S description. The web service specification is used by the web service tester for designing test cases. The test cases are then executed on the web service and the test result report is generated. One could observe that during the whole process of web service testing, the end user does not have an input in any phase. The reason why there is no place for the end user during testing is that web service specification and end user goal are not distinguished from the very beginning in the design of traditional web service architecture. As a consequence, the web service tester is checking if the web service meets the provider's own specification, rather than the user requirements. Apparently, since a provider's web service specification only describes the behavior of web services from his/her own point of view, it may not necessarily coincide with the end user's understanding/expectation of that web service. Thus, the testing of web services based solely on the provider's specification is unable to ensure the correctness of web services from the end user's perspective. The term *correctness* here means that it will fulfil all of the user requests and thus, satisfy the user requirements completely. In any software testing approach, the objective is to ensure the correctness of the soft-

* Corresponding author. Fax: +6493737453.

E-mail addresses: mjok002@aucklanduni.ac.nz (M.S. Jokhio), j.sun@cs.auckland.ac.nz (J. Sun), gill@cs.auckland.ac.nz (G. Dobbie), tmhu@ieee.org (T. Hu).

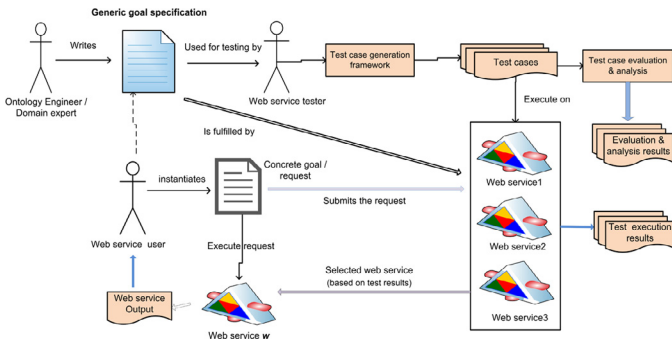


Fig. 1. A high level overview of the proposed solution.

ware for the end users who use it. However, since the present approaches of web service testing are mainly based on the provider's specification, they may not ensure correctness for the end users.

In 2005, the Web Service Modeling Ontology (WSMO) was introduced as a conceptual framework for semantically describing all relevant aspects of SWS in order to facilitate the automation of services over the Web [4]. In particular, WSMO explicitly separates web service specifications from the user requirements with the help of the top level element, called the *WSMO goal*, in its conceptual framework. Thus, in the scenario of WSMO based SWS testing, we do have a chance to check if the web service really meets the user requirements.

In this paper, we focus on two parts of goal based SWS testing: test case generation and evaluation. Specifically, we apply model checking to derive test cases for SWS from goal specifications. Since the quantity of test cases does not necessarily add value to the testing, it is important to measure their quality. Therefore, to make the subsequent test case evaluation independent of the model-based generation approach, we proposed a mutation-based solution for such a purpose. Furthermore, to make this approach scalable and practical, we implement a tool to automate all the evaluation steps. Finally, we applied our approaches to the well-known Amazon E-Commerce Service, a web service that exposes Amazon's product data and E-Commerce functionality. The case study validated the effectiveness of the proposed solution and the usefulness of the implemented tool.

To realize goal based testing of SWS, we use the WSMO framework. Fig. 1 gives the overview of the proposed solution. In order to understand how it works, we need to understand the three different roles involved: the *web service tester*, the *domain expert* or an *ontology engineer*, and a *web service user*. The *web service tester* is responsible for web service testing and quality assurance. This is done by generating concrete test cases and executing them on the semantic web services. A *domain expert* or an *ontology engineer* is a person who has knowledge of the domain related to the user's requirements and masters expertise to model user requirements using a WSMO goal specification and WSMO ontology language, i.e., WSML. Finally, a *user* or a *web service user* uses GUI-based tools that the WSMO framework provides to instantiate the generic goal specification and define his or her own concrete goal requests. Thus, the users are only expected to specify their concrete requests with the help of tools, but they are not expected to specify or write the generic WSMO goal specification of their own.

In detail, the whole system proceeds the following way. The goal specification is designed and published on a public repository by an ontology engineer that represents the generic user requirements that can be fulfilled by a set of semantic web services. For example, a goal *Book A Flight* represents the generic user requirements for booking a flight and can be fulfilled by several web services. Such a goal specification is used by web service testers to

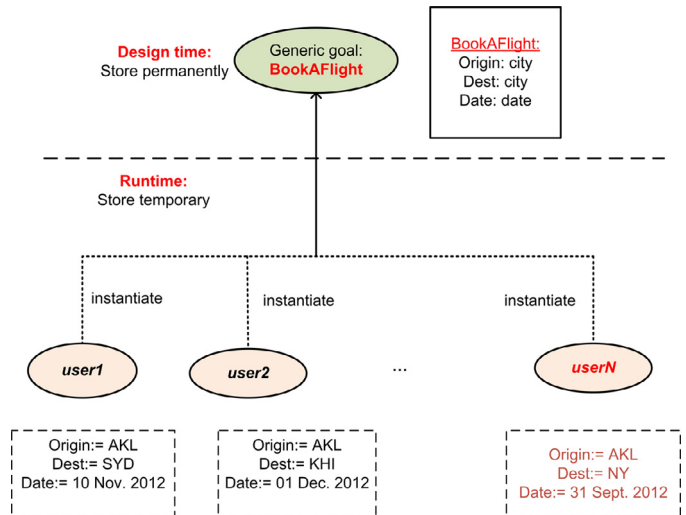


Fig. 2. Different users requirements instantiated from the generic goal specification.

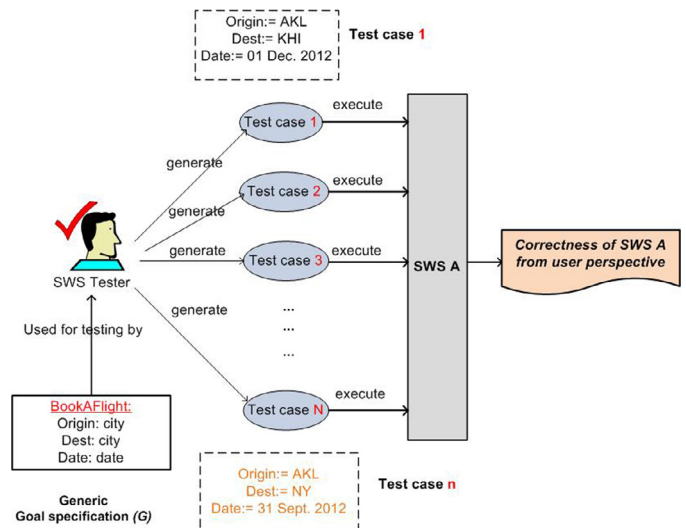


Fig. 3. SWS testing based on the proposed approach.

generate a set of test cases, which is then executed on the selected web services and the test reports are recorded. A web service user finds the appropriate goal to meet his or her requirements and instantiates it using the GUI tools to define his concrete requests. The user submits the concrete request for achieving the goal. The semantic web service execution environment may select a particular web service that fulfills the user requirements. This selection can be based on the test result reports, if no other selection parameters are specified by the user. The user uses the selected web service to execute his/her request and gets the required response.

In the proposed solution, we can see that the SWS tester is now really testing the selected web service from the user perspective. On one hand, as shown in Fig. 2, all these user requests are instantiated from a generic goal specification and each instance represents a concrete user request. For instance, it is possible that, by mistake, a user can instantiate a request with an invalid flight date. On the other hand, as shown in Fig. 3, since the SWS tester uses the same goal specification to generate the test suite, he can perfectly imitate the users to simulate user requests, including those critical ones. Thus, if a web service passes the comprehensive enough test suite generated by the tester, we are confident

Download English Version:

<https://daneshyari.com/en/article/4972327>

Download Persian Version:

<https://daneshyari.com/article/4972327>

[Daneshyari.com](https://daneshyari.com)