ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Survey on Collaborative Testing of Web Services

Chirra Sharath[*1], D.Basawaraj[2]
[*1] PG Student, Department of CSE, CMR Institute of Technology, Hyderabad, India
[2] Associate Professor, Department of CSE, CMR Institute of Technology, Hyderabad, india
chirrasharath@gmail.com

Abstract

The foremost realization of Service-Oriented Architecture (SOA) is Web Services (WS). It defines a framework for nimble and adaptable amalgamation among self directed services based on Internet open standards. SOA allows composition of distributed applications free from their platform and thus reduces the cost of such compositions and makes them easier and faster to develop. However, testing has been confronted due the vibrant and collaborative nature of WS. This paper analyses the key techniques present in testing of WS and shows the strength and weakness of current web service testing technology.

Keywords: Ontology, Semantic Web Services, Service Oriented Architecture, Testing, Web Service.

Introduction

Service Oriented computing is a new computing paradigm that utilizes services as the lightweight constructs to support the development of rapid, low-cost and easy composition of distributed applications. This is a widely used definition of SOC. The concept of a "service"is comparatively elusive and consequently harder to describe. The definition adopted for this survey is: Services are autonomous, platform-independent computational elements that can be described, published, discovered, orchestrated and programmed using standard protocols to build networks of collaborating applications distributed within and across organizational boundaries.

Service-Oriented Architecture (SOA) and its Web implementation Web Services (WS) promote [2] an open standard-based and loosely coupled architecture for integrating applications in a distributed heterogeneous environment. Such applications are characterized by service orientation, task distribution, and collaboration among development parties, run-time behavior and open standards for interfacing among their components. Web Services are acknowledged another standard in constructing programming provisions dependent upon the Internet what's more open guidelines. This standard has altered the way we check out the Internet from being an archive of information into an archive of Services. By utilizing Web Services, associations can guarantee that their provisions will correspond with those of their business friends and clients regardless of the fact that they are utilizing diverse modifying dialect or stages. Testing might be utilized to take care of this issue; by evaluating the quality characteristics of a Web Service under test [4], the certainty or the reliability of the requester's expansion or diminish as per the test effects.

However, software testers are confronted with great challenges in testing. Testing of web services classifies into four categories: functional, regression, integration and non functional testing. The remainder of this survey is organized as follows. Section 2 briefly explains the SOC paradigm, SOA. Section 3 narrates the various underlying technologies in testing the web services . Section 4 discusses the issues related to testing web services. Section 5 concludes the survey.

Service Oriented Computing

SOC shifts the traditional understanding of software application design, delivery and consumption. The idea of SOC is that it ought to be able to create a more systematic and a more efficient way of building distributed applications [1]. The vision underpinning this idea is to establish a world of loosely coupled services, able to rapidly assemble dynamic business processes and applications. There are two primary characteristics of SOC applications through which these advantages are:

1. In SOC, all the services must comply with the interface standards so that services are guaranteed to be platform-independent.
2. Service descriptions must enable the automation of integration process (search, discovery and dynamic composition).

One of the barriers to enterprises' transition to SOC systems is denoted by the heightened importance of the issue of trust. This issue has many dimensions such

ISSN: 2277-9655
Impact Factor: 1.852

as correct functioning of service, service security and also Quality of Service (QoS). Testing provides one potential approach to the issue of trust. Testing is important to assure the correct functioning of service-oriented systems that, by nature, have the ability to dynamically select and use services. In order to confirm the correct functioning of a service-oriented system, interoperability among all its components and integration of these components must be adequately tested.

SOC may also require more frequent testing than traditional software [3]. The possibility of changes to a service-oriented system increases with the number of services involved. With every change to a service, the service-oriented system needs to be tested. Testing service-oriented systems for changes presents other challenges, such as when to test for changes and which operations are affected by the changes. These problems occur when changes are made to services of other businesses. Because of the reasons stated above, service-oriented systems require more effective and efficient testing than traditional software systems. Unfortunately, most of the existing testing approaches for distributed systems are not adequate enough or applicable to SOC due to the limitations SOC brings.

1. Limitations in observability of service code and structure due to users having access to service's interface only.
2. Lack of control due to independent infrastructure on which services run and due to provider being the only control mechanism over service evolution.
3. Dynamicity and adaptiveness that limit the ability of the tester to determine the web services that are invoked during the execution of a workflow.
4. Cost of testing:

Service Oriented Architecture

SOA is described as a strategy for building service-oriented applications. Its aim is to provide services that can be used by other services. In SOA, there are three main participants; a service provider, a service user and a service broker. These three participants perform the three fundamental SOA actions; publish, find and bind. Figure 1 illustrates this foundational SOA concept, its participants and the operations among participants.

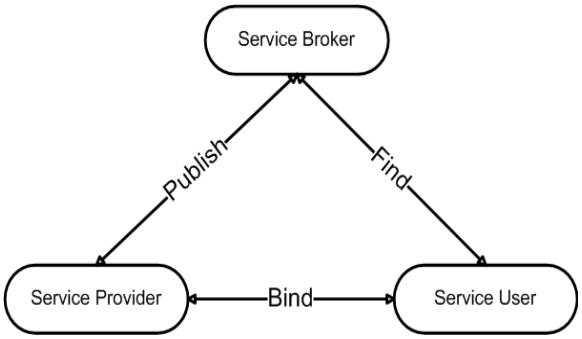

Figure 1 Service Oriented Architecture

Service Oriented Computing

A web service is defined as "a software system designed to support interoperable machine-to-machine interaction over a network" by W3C. There are different web service styles such as Representational State Transfer (REST) web services and Simple Object Access Protocol (SOAP) [5] web services. They are all based on the SOA but differ in the interfaces that they use. Figure 2 describes the web service architecture and its core specifications that are used in performing each SOA operation. It also illustrates the way in which web services implement the general concept of SOA.

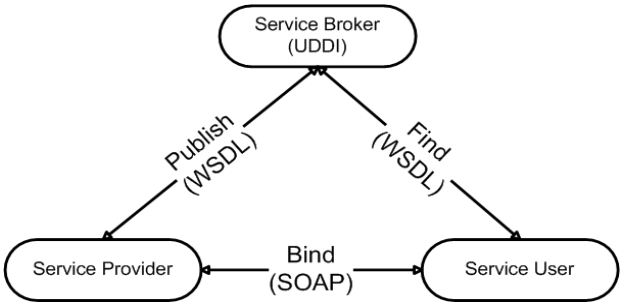

Figure 2 Web Service Architecture

Web Service Testing

Testing web services at the client side is one of the major problems that slows down the wider use of web services. The increased attention to SOC and web services is also reflected on their testing [6]. Testing web services includes testing of the basic web service functionality, web service interoperability, some of the SOA functionalities, QoS and load/stress testing.

ISSN: 2277-9655
Impact Factor: 1.852

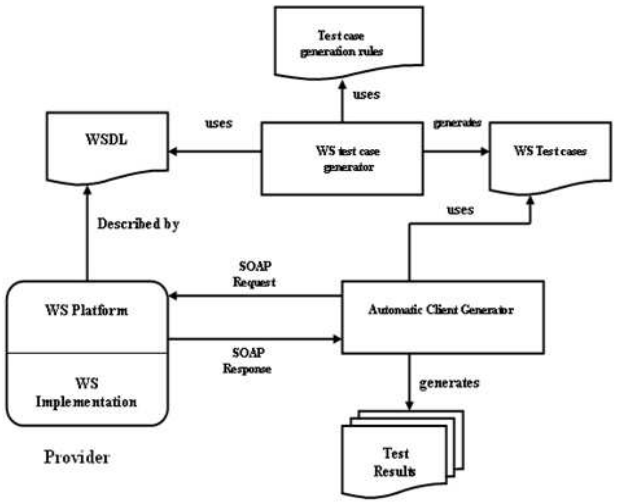

Figure 3 Web Service Testing Architecture

A. ***Service Oriented Computing***

Tester: A tester refers to a particular party who participates in a test activity. Generally speaking, testers can be human beings, organizations, and software systems. In the service oriented framework, T-services perform the test tasks, thus they are testers, too. It can be an atomic Tservice, or a composition of T-services. One important property of tester is its capability, which reflects the capability to perform test tasks.

Activity: There are various test activities including test planning, test case generation, test execution, result validation, adequacy measurement, and test report generation, etc.

Artifact: . Various kinds of artifacts may be involved in test activities as input/output, such as test plan, test cases, test results, program, specification, and so forth [8]. The most important property of class Artifact is Location, whose value is an URL referring to the location of the Artifact. Each type of artifacts is a subclass of artifact, and inherits the properties from Artifact. The subclasses of Artifact can be added into the ontology using the ontology management services. Context. Test activities may occur in different software development stages and have various test purposes. The concept context defines the contexts of test activities in testing processes and test methodologies. Typically, the contexts include unit testing, integration testing, system testing, regression testing, etc.

Method: . For each test activity, there may be multiple applicable test methods. Method is a part of the capability and also an optional part of test task. Test methods can be classified in a number of different ways. For example, test methods can be classified into program-based, specification based, usage-based, etc. They can also be classified into structural testing, fault-based testing, error-based testing, etc. Structural testing methods can be further classified into

To deal with semantic complexity [7], ontology of software testing is used. In general, ontology defines the basic terms and relations comprising the vocabulary of a topic area as well as the rules for the combination and extension of the vocabulary. The Web Ontology Language OWL is a semantic markup language for publishing and sharing ontologies on the World Wide Web. It is designed for applications that need to process the content of information. It is a part of the growing stack of W3C recommendations related to the Semantic Web.

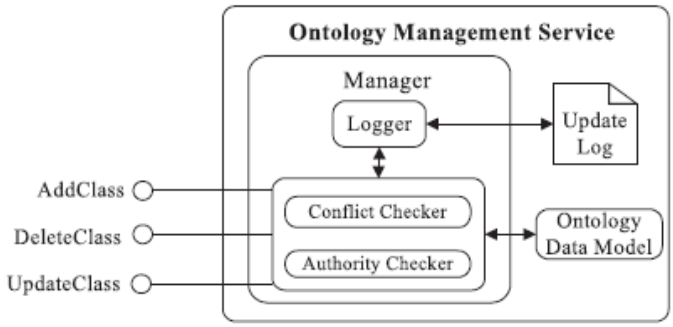

Figure 3. The structure of OMS

Conclusion

This paper describes about the fundamentals of service oriented computing and analyses the various testing strategies used in web services. Part 2 completely talks the architecture of service oriented computing. Part 3 discusses about the testing fundamentals. This survey paper helps to understand the ways of performing web service testing.

References

[1] Hong Zhu and Yufeng Zhang, "Collaborative Testing of Web Services" , IEEE Transactions on Service computing , Vol.1, No-5 , PP. 116-130.

[2] G. Canfora and M. Penta, "Service-Oriented Architectures Testing: A Survey," Software Eng.: Int'l Summer Schools (ISSSE 2006-2008), Revised Tutorial Lectures, A. Lucia and F. Ferrucci, eds., pp. 78-105, Springer- Verlag, 2009

[3] Samer Hanna and Malcolm Munro Department of Computer Science Durham University UK"Fault-based Web Services Testing" 2008 IEEE.

[4] Bertolino, A., & Polini, A. (2009). SOA Test Governance: enabling service integration testing

ISSN: 2277-9655
Impact Factor: 1.852

across organization and technology borders. IEEE International Conference on Software Testing Verification and Validation Workshops, 277-286. Doi: 10.1109/ICSTW.2009.39

[5] Wang, Y., Ishikawa, F., Honiden, S. (2010). Business Semantics Centric Reliability Testing for Web Services in BPEL. IEEE 6th World Congress on Services, 237-244. Doi: 10.1109/SERVICES.2010.88

[6] Mei, L., Chan, W.K., Tse, T.H. (2008, May 10-18). Data Flow Testing of Service-Oriented Workflow Applications, ICSE 08, Leipzig, Germany, 371-380. Doi: 10.1145/1368088.1368139

[7] Richard Hull Bell Labs Research Lucent Technologies, Jianwen Su Department of Computer Science UC Santa Barbara"ToolsFor Composite Web Services:A Short Overview" June 2005

[8] H. Zhu, "A Framework for Service-Oriented Testing of Web Services," Proc. 30th Ann. Int'l Computer Software and Applications Conf. (COMPSAC '06), pp. 679-691, Sept. 2006.