ISSN : 2347 - 2243

Indo - American Journal of Life Sciences and Biotechnology

www.iajlb.com
Email : editor@iajlb.com or iajlb.editor@gamil.com

ISSN2347-2243www.iajlb.com
Indo-Am. J.ofLifeSc& Bt.,202- *Vol.4,Issue.1,jan2016*

Indo-American Journal of Life Sciences and Biotechnology

Text to Audio Conversion System with Efficient Portable Camera Methods for Reconfiguring Tests of Composite Web Services on the Fly

Amiri Mahalili

ABSTRACT

A suitable testing strategy for service-oriented applications will be implemented in this project. And boost the efficiency of the online service. In this article, I outline an architecture that can adapt to shifting requirements in terms of service operation, argument types, and service composition. My model atomic and composite web services undergo random testing at runtime as part of the testing system's proof of concept. in order to maintain the testing infrastructure's availability while a new test subject is being configured automatically.

Keywords: *Software Testing, Service Oriented Architecture (SOA) Testing, Runtime Analysis, and Code Smell.*

1. INTRODUCTION

Web service and software reliability may be determined via rigorous testing and verification of quality of service. The quality of service (QoS) assurance provided by existing software testing methods is insufficient for service-based applications.

While it is possible to expose a Web service as an atomic unit with merely an interface for interaction, most web services are really composites of several services, some of which may even be hosted by unaffiliated third parties. Such web services and applications are difficult to evaluate because to issues such as dynamic composition and a lack of oversight over the web services' offered components. This article provides [1] to demonstrate the importance of testing during runtime.[2] Real-time detection of changes to service interfaces and composition, including component addition, deletion, and modification.

University Of Madras, IDE Building, Chepauk, Chennai, Tamil Nadu 600005, India

MOTIVATION & RELATED WORK

Accessing a website's contents for display is just half of what the World Wide Web entails now that a new idea known as web services has been introduced. We merely sent some input to the online service, and he gave you the results without you having to download and install any software.

It's possible that web service has either atomic or composite. It has to be tested to ensure its dependability. Composite web services are complex and challenging, but you must test them now. Many researchers these days are focusing on making web service testing tools more trustworthy. Mutative hange, as defined by King and Ganti [3]: the modification of an existing part while keeping some of its original functioning. Modifications that add anything new are called additive changes.

2. LITERATURE REVIEW

The purpose of this literature review is to compare and contrast the various methods currently in use for anticipating users' actions.

Integration testing technique for Web services, shown by Z. Hong and Z. Yufeng[4], involves the cooperative invocation of numerous test service partners during runtime. The included testservices were used to improve the testing procedure as a whole. Existing testing frameworks were said to have difficulties with SOA application testing. Due to the implementation being concealed from the users, there are no software artifacts. Test execution can't be controlled because of interactions between distributed components. Components give an opaque interface, making it difficult to see internal activities.

We propose using runtime or live testing methodologies for SOA applications based on the research of Hong and Yufeng. Runtime testing of SOA requires testing approaches that may evolve with the system being evaluated. Runtime testing needs for SOA applications were discovered by Bai et al. [5], and an adaptive testing framework was designed to help with that. Based on their prior work [6] and the work of Tsai et al. [7], Bai et al. suggested an adaptive test framework that used a test broker architecture. They employ an enhanced version of the UDDI service registry as a test broker. They implemented a Web service under test as a feedback procedure, which provided a response to a specific test and guided further testing steps.

Because Web services don't provide such details to end users, they can't see how a service is put together. Testing them in a black-box environment is unrealistic. A business process's output for a given set of parameters must be checked against the process's internal logic to ensure it is correct. There are now available testing tools for service composition. Although several testing methods may be used to the task of evaluating a service's composition, Bucchiarone et al. [12] claim that none of them are advanced enough to deal with the complexity of the composition's activities.

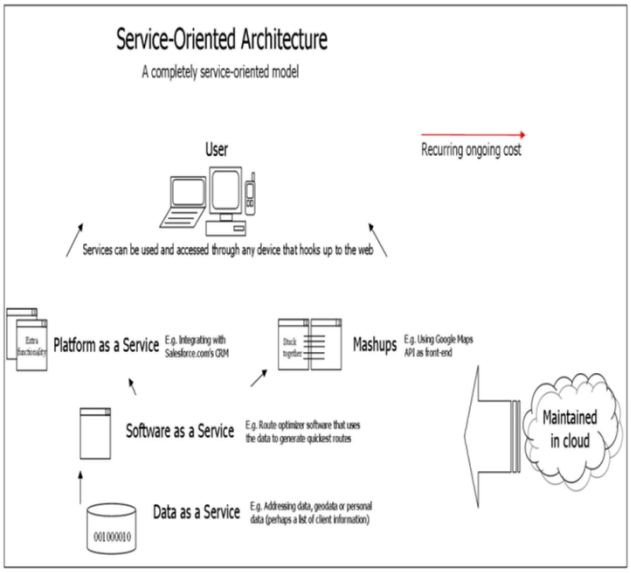
The interface specification is a major tool for automated test creation in Web services. Examples of popular methods include WSDL-based test data creation [10]. However, there are additional language standards that might be utilized to generate test cases. Operations and messages are tied to a concrete network protocol and message format, and the specifications for that port type make up a reusable binding. The WSDL file specifies the Web service's public face. Web services via the Internet often include the usage of WSDL, SOAP, and an XML Schema. By reading the WSDL file, a client software may learn what methods are supported by a Web service. The WSDL file incorporates XML Schema for any custom data types required. One of the WSDL file's actions may then be called by the client using SOAP, for instance via XML over HTTP.

3. PROPOSED WORK

Cloud-based software applications that make use of atomic and composite Web services are the primary target of the proposed system's testing efforts. Many cloud-based apps are built from indistinct third-party services. These hidden services are always evolving. Runtime (during actual use of the system) testing is just as important as testing during development. Test artifacts for such applications must be kept up-to-date in real-time in reaction to changes in order to pass runtime testing. They are connected to software testing and online services in the proposed system [1].

The architecture of the testing system consists of four basic levels. The Service Interface Layer facilitates communication with parties outside of the test environment. New test requests may be submitted by service providers using the interface, and test policy updates can be made as needed. An external coordinator is part of the Service Interface Layer that communicates with clients when they call a Web service and employs encrypted connections for administrative transactions related to testing. The System Operations Layer is where all the meaty logic for things like testing techniques, stateful

Fig. 1. Web services in a service-oriented architecture.



management, and system reconfiguration operations. The Service Management Layer provides system administration components and test scheduling components. The administration components are responsible for handling test service registrations and test policy updates. The scheduling components perform test allocations and trigger tests based on the schedule in the test policy. The Resource Access Layer provides data access logic for test data retrieval and storage.When the user sends request to the service provider for code smelling technique. For Enhancement we will enhance the work by implementing code smelling technique.

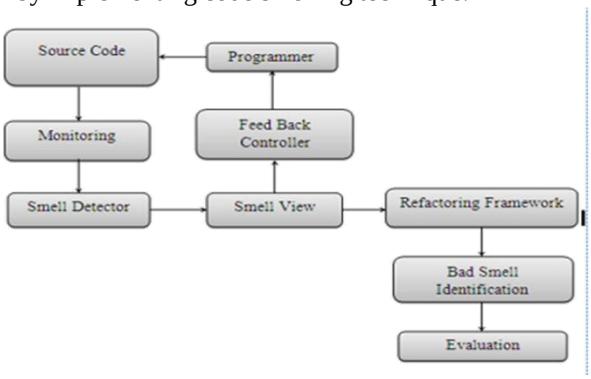


Fig. 2. Architecture Diagram.

Z. Hong and Z. Yufeng [4]	collaborating multiple test service partners invoked at runtime.	Collaborative testing of web services	2012
X. Bai, C. Yinong, and S. Zhongkui [5]	adaptive testing framework which can continuously learn and improve	Adaptive web services testing	2007
M. Chunyan, D. Chenglie, Z. Tao, H. Fei, and C. Xiaobin[10]	automated test generation is through the interface specification	WSDLbased automated test data generation for web service	2008

4. SUMMARY OF LITERATURE REVIEW

Table 1: Summary

Author	Method	Application	Year
M. B. Cooray, J. H. Hamlyn-Harris, and R. G. Merkel[2]	WSDL service descriptions as a framework for generating test cases	Test reconfiguration	2011
Mark B. Cooray, James H. Hamlyn-Harris, and Robert G. Merkel [1]	Automated reconfiguration	Dynamic Test Reconfiguration for Composite Web Services	2015

5. CONCLUSION

This paper has provided a more current evaluation and updation of 'Dynamic Test Reconfiguration for CompositeWeb Services' That will make more reliable composite web service & reduce waiting time of the web service consumer.

REFERENCES

According to [1] "Dynamic Test Reconfiguration for Composite Web Services" by Mark B. Cooray, James H. Hamlyn-Harris, and Robert G. Merkel. July/August 2015 Issue of IEEE Transactions on Services Computing Volume 8 Number 4

"Test reconfiguration for service oriented applications," by M. B. Cooray, J. H. Hamlyn-Harris, and R. G. Merkel, was published in the proceedings of the 2011 IEEE International Conference on Utility Cloud Computing.

[3] "Migrating autonomic self-testing to the cloud," by T. M. King and A. S. Ganti, was published in the proceedings of the 2010 International Conference on Software Testing, Verification, and Validation held in Paris, France (pp. 438–443).

According to [4] "Collaborative testing of web services," by Z. Hong and Z. Yufeng, published in IEEE Trans. Service Comput., volume 5, issue 1, pages 116-130, January-March 2012.

"Adaptive web services testing," by X. Bai, C. Yinong, and S. Zhongkui, was published in the proceedings of the 31st Annual ACM Symposium on Applied Computing in 2007 (pp. 233-236).

Sixth, "Design of a Trustworthy Service Broker and Dependence-Based Progressive Group Testing," Int. J. Simul. Process Modell., volume 3, pages 66-79, 2007. Authors X. Bai, Z. Cao, and Y. Chen.

"Verification of web services using an enhanced UDDI server," by W. T. Tsai, R. Paul, Z. Cao, L. Yu, and A. Saimi, published in Proceedings of the Eighth International Workshop on Object-Oriented Real-Time Dependable Systems, 2003, pages 131-138.

[1] "Dynamic reconfigurable testing of service-oriented architecture," by X. Bai, X. Dezheng, D. Guilan, T. Wei-Tek, and C. Yinong, in Proceedings of the 31st Annual ACM/IEEE/ACM Symposium on Theory and Practice of Software Engineering (pp. 368–378), 2007.

"Strategies for the run-time testing of third party web services," by David Brenner, Christopher Atkinson, Oliver Hummel, and David Stoll, was published in the proceedings of the 2007 IEEE International Conference on Service-Oriented Computing Applications.

"WSDLbased automated test data generation for web service," by M. Chunyan, D. Chenglie, Z. Tao, H. Fei, and C. Xiaobin, was published in the proceedings of the 2008 international conference on computing, science, and software engineering (Proc.

"WSDL-based automatic test case generation for web services testing," by X. Bai, D. Wenli, T. Wei-Tek, and C. Yinong, was published in the proceedings of the 2005 IEEE International Workshop on Service-Oriented Systems Engineering.

Reference: [5] "Testing service composition" by A. Bucchiarone, H. Melgratti, and F. Severoni in Proceedings of the Eighth Argentine Symposium on Software Engineering, Mar del Plata, Argentina, 2007. Pages 1-16.