

A Survey on Technique for Dynamic Test Reconfiguration for Composite Web Services

Mr.Chetan N Phalke¹, Mrs.Priti Vyawahare²

¹ P.G. Student, Department of computer Engineering , ACEM Pune, Maharashtra, India

² Asst. Professor, Department of computer Engineering , ACEM Pune, Maharashtra, India

Abstract – This project will introduces a testing policy that is appropriate for testing service-based applications. And enhance web service performance. I describe an architecture that responds to changes of service operation , arguments and changed service composition. To prove test system performs runtime testing by random testing technique on my model atomic and composite web services. to support the continuous operation of the testing systems during a test candidate change done by Automated reconfiguration.

Key Words: Code smell , Runtime testing, SOA Testing, software testing.

1 INTRODUCTION

Reliability is important for Web Service & Software which calculated by Testing and verifying QoS. Currently available software testing techniques not provide the level of QoS guarantee that is required for service-based applications.

Web services can be exposed as an atomic Web service where only an interface is available for interaction; But mostly web service is made by composite different services , many of which may be provided by third parties. Such web services ,application are complicated to test, having problem like composed dynamically , there is no control on component entity which offer by web services

Run time testing is important so this paper provide[1][2] Identification of service composition changes, including addition, deletion, and modification of a component, and interface changes at runtime.

2 Motivation & Related Work

WWW is just not to get Access of any particular website for display contain, information but recently new concept is added on this is software which run on the Internet call web service. We have just submit some input on that web

service he het result you without giving downloading & install that particular software.

That web service may have Atomic or composite. To increased its reliability we have to Test it. But Test at the ru time that composite web service is complicated & difficult thing. Now Days may of researcher are trying to improve the reliability of Testing tool work on the web service. King and Ganti [3] have identified Mutative hange: A transformation of an existing component while retaining some of the existing functionality of that component. Additive change: An introduction of a new component interface or an implementation.

3. Literature Review

The focus of literature review is to study and contrast the available technique to predict the users' future movements. Z. Hong and Z. Yufeng [4] shows integration testing mechanism for Web services by collaborating multiple test service partners invoked at runtime. The participating test services were used to assist the overall testing process. They reported challenges for existing testing frameworks when dealing with SOA application testing: Lack of software artifacts due to implementation being hidden from the users. Lack of control over test execution due to distributed component interactions. Lack of ways to monitor internal behavior, since the components provide an opaque interface.

Based on Hong and Yufeng's work, we suggest that runtime or live testing strategies should be used for testing SOA applications. To make runtime testing of SOA practical, it is necessary for testing methodologies to adapt as the tested system changes. Bai et al. [5] identified runtime testing requirements for SOA applications and introduced an adaptive testing framework which can continuously learn and improve the builtin test strategies. The proposed adaptive test framework by Bai et al. used the concept of a test broker architecture based on their previous work [6] and work by Tsai et al. [7]. The test broker used in their work is an extension to the UDDI service registry. In their work, a Web service under test was used as a feedback process, which responded to a given test and helped to control testing activities based on the response received.

A service composition is not visible to the users of the service as the Web services do not expose composition information. Running black-box testing on them is impractical. It is necessary to verify the internal logic of a business process to determine if the output received for a given set of arguments is valid. Tools to conduct service composition testing exist. According to Bucchiarone et al. [12] even though several testing techniques can be used when testing service composition, the techniques are not sufficiently mature to handle the complex operations in a composition.

In Web services, a primary method of automated test generation is through the interface specification. For example, WSDL-based test data generation is a widely used technique [10]. However, there are specifications composed in other languages, which could be used for test case generation. The concrete protocol and data format specifications for a particular port type constitutes a reusable binding, where the operations and messages are then bound to a concrete network protocol and message format. In this way, WSDL describes the public interface to the Web service. WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL file using for example XML over HTTP.

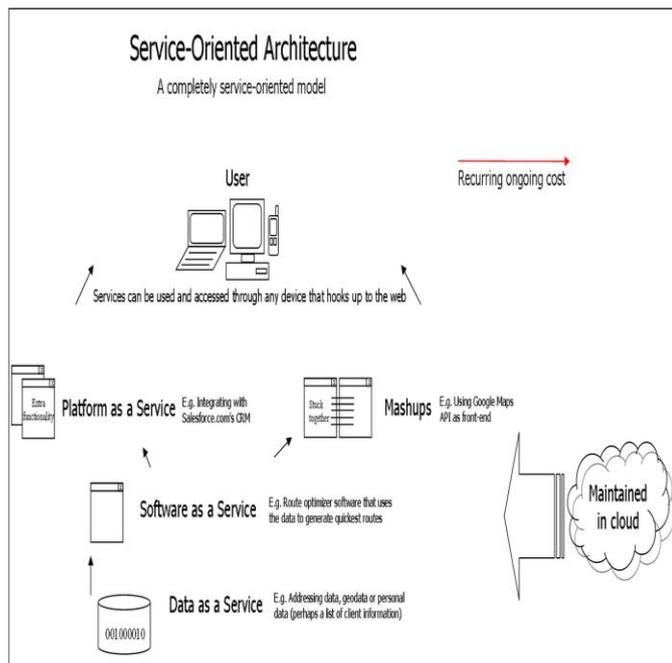


Fig 1 Web services in a service-oriented architecture.

4 Proposed Work

Proposed System focuses on testing cloud software applications, which use atomic and composite Web services. Cloud applications are often composed of opaque services supplied by third party. These opaque services are highly dynamic. Testing must not only take place during development, but at runtime (during the operation of the system). When performing runtime testing of such applications, test artifacts are required to be updated instantaneously in response to changes. In the proposed system they integrate with the web services and software testing.[1]

The test system architecture has four main layers. The Service Interface Layer provides interaction with stakeholders external to the test system. The test system administration component accepts new test requests from service providers and the interface also provides a test policy update facility. The Service Interface Layer also contains an external coordinator component that interacts with the clients when invoking a Web service and uses secure access links for test-related administrative-transactions. The System Operations Layer contains the core logic that implements testing mechanisms, change 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.

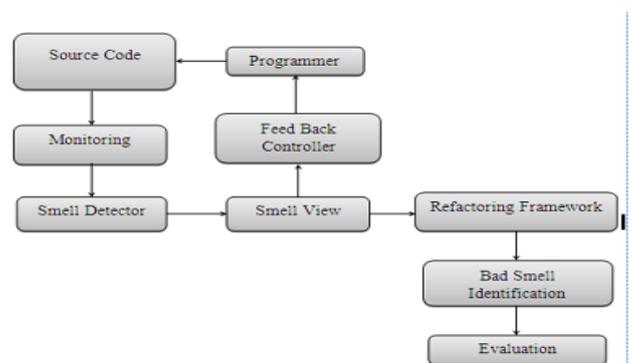


Fig 2: Architecture Diagram.

5 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
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

6 CONCLUSION

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

7 REFERENCES

- [1] Mark B. Cooray, James H. Hamlyn-Harris, and Robert G. Merkel, "Dynamic Test Reconfiguration for Composite Web Services" IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 8, NO. 4, JULY/AUGUST 2015
- [2] M. B. Cooray, J. H. Hamlyn-Harris, and R. G. Merkel, "Test reconfiguration for service oriented applications," in Proc. IEEE Int. Conf. Utility Cloud Comput., 2011, pp. 300-305.
- [3] T. M. King and A. S. Ganti, "Migrating autonomic self-testing to the cloud," in Proc. Int. Conf. Softw. Testing, Verification, Validation Workshops, Paris, France, 2010, pp. 438-443.
- [4] Z. Hong and Z. Yufeng, "Collaborative testing of web services," IEEE Trans. Service Comput., vol. 5, no. 1, pp. 116-130, Jan.-Mar. 2012.
- [5] X. Bai, C. Yinong, and S. Zhongkui, "Adaptive web services testing," in Proc. 31 Annu. Comput. Softw. Appl. Conf., 2007, pp. 233-236.
- [6] X. Bai, Z. Cao, and Y. Chen, "Design of a trustworthy service broker and dependence-based progressive group testing," Int. J. Simul. Process Modell., vol. 3, pp. 66-79, 2007.
- [7] W. T. Tsai, R. Paul, Z. Cao, L. Yu, and A. Saimi, "Verification of web services using an enhanced UDDI server," in Proc. 8th Int. Workshop Object-Oriented Real-Time Dependable Syst., 2003, pp. 131-138.
- [8] X. Bai, X. Dezheng, D. Guilan, T. Wei-Tek, and C. Yinong "Dynamic reconfigurable testing of service-oriented architecture," in Proc. 31st Annu. Int. Comput. Softw. Appl. Conf., 2007, pp. 368-378
- [9] D. Brenner, C. Atkinson, O. Hummel, and D. Stoll, "Strategies for the run-time testing of third party web services," in Proc. IEEE Int. Conf. Service-Oriented Comput. Appl., 2007, pp. 114-121.
- [10] M. Chunyan, D. Chenglie, Z. Tao, H. Fei, and C. Xiaobin, "WSDLbased automated test data generation for web service," in Proc. Int. Conf. Comput. Sci. Softw. Eng., 2008, pp. 731-737.
- [11] X. Bai, D. Wenli, T. Wei-Tek, and C. Yinong, "WSDL-based automatic test case generation for web services testing," in Proc. IEEE Int. Workshop Service-Oriented Syst. Eng., 2005, pp. 207-212.
- [12] A. Bucchiarone, H. Melgratti, and F. Severoni, "Testing service composition " in Proc 8th Argentine Symp. Softw. Eng., Mar delPlata, Argentina, 2007, pp. 1-16.