ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

Web Services Hash Table based Automation Testing

Chetan Phalke, Prof. K.D.Bamane

Masters of Engineering Student , Dept. of Computer Engineering, Alard College of Engineering & Management, Maruje, Savitribai Phule Pune University Pune, India

Assistant Professor, Dept. of I.T., Dr. D.Y.Patil College of Engineering, Pune, India

ABSTRACT: Testing a service based application with more suitable approach is introduced in this paper. Paper has an architecture that responds to changes of service operation , service composition changes and allow all Footprints by Tree. Our test system is able to identify changes that occur in service operations and operational arguments in a service description of a test candidate. Automated reconfiguration is used to support the continuous operation of the testing systems during a test candidate change. Our proof-of-concept test system performs runtime testing on our model atomic and composite web services using a random testing technique with previous footprints records.

KEYWORDS: ATTCWSIF ,Footprints, Runtime testing, SOA, Software Testing, Hash Table

I. INTRODUCTION

Recent Days service-based applications become for trendy and useful for verification and validation. The popularity of service-based applications makes it increasingly important that effective means for verification and validation of them are available. Due to only interface is available for web services it ts difficult to test service oriented application because there are many chance of third party services. Due to such difficulty Testing technics become more important . The dynamic execution environment of Web services means that runtime testing is necessary for verifying their ongoing reliability during deployment and production use.

Despite the very large number of available Web services (WS), each one, taken alone, has a relatively limited functionality. Generally, user requests cannot be satisfied by a single web service. Thus, WS should be combined to full fill user requirements. The combination process of WS is called WS composition. Current testing approaches involve several manual tasks and, hence, are error-prone and costly. In fact, the analysis approach is accomplished by detecting, manually, critical code locations in order to verify its vulnerability. Software engineers have to investigate these program places manually to decide for themselves whether this part of the program may or may not violate the property at hand. However, making this kind of analysis is almost very subtle and complex. Furthermore, composite WS impose additional analysis complexity as they are consumed by a plethora of clients having different desires. This provides frequent changes within composite services.

In this paper, we extend previous work [1] on runtime testing of atomic Web services. Our methods allow: Identification of service composition changes, including addition, deletion, and modification of a component, interface changes at runtime , and leave footprints . We have implemented a system to demonstrate these methods, and apply them to a small composite Web service.

II. RELATED WORK

Project research focus is on testing service oriented applications, which use atomic and composite Web services. Such web services maybe composed by thirdparty . Testing must not only take place during development, but at runtime. According to King and Ganti [3] test System will affect by Additive change: An introduction of a new component interface or an implementation. Reductive change: A removal of an existing implementation or an interface from a cloud application. Mutative change: A transformation of an existing component while retaining some of the existing functionality of that component. For any runtime testing system required artifacts related that web service.

DOI: 10.15680/IJIRCCE.2017. 0507056

Hong and Yufeng [4]Identify by collaborating multiple web services test service partners invoked at runtime so it has Lack of software artefacts due to implementation being hidden from the users. Lack of control over test execution due to distributed component interactions.

Runtime SOA Testing 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. SOA applications and introduced an adaptive testing framework which can continuously learn and improve the built-in test strategies as Bai et al. [5]

Mark B. Cooray, James H. Hamlyn-Harris, and Robert G. Merkel[1] used service composition details to identify deviations in composition and used WSDL service descriptions as a framework for generating test cases[1][2]. We also used service composition details to identify deviations in composition.

Existing Testing approaches involve several manual tasks, Specifically Testing a web services. Due to this manual task overall testing process becomes error-prone and costly.And we notice that Existing Architecture is lengthy withoutEffective traceability of web services

III. PROPOSED ALGORITHM

A. *Description of the Proposed Algorithm:*

Aim of the proposed algorithm is to prepare hash table to minimize search web service artefacts delay.

Input: WSDL
Output: Hash table H
Algorithm: Parse WSDL and store it in hash table.

1. Select the <operation> element under the element. Select the name of the services name concate with operation and assign it as the key for the hash table <key, value> pair. Operation Name is the unique key value for the Hash table.
2. Create an empty List L which will store all the values related for a particular Key.
3. Select the <input message> element and count the number of <sequence> elements associated with the <input message>. This count will give the number of parameters for the selected operation name.
4. Add the count to the List L.
5. Select the <input message> element and using the input message name extract the parameter names and their respective types for the selected operation name.
6. Add the parameter names and their respective values to the List L.
7. Select the <output message> element and using the output message name extract the return type for the selected operation name.
8. Add the return type value to the List L.
9. Link the List L as the value part for the selected operation name key in the Hash table.
10. Repeat Steps 1-9 for all the <operation> elements in the WSDL

IV. .PSEUDO CODE

```
Step 1: For each key opkey in delta hash table H, Repeat step 2..
Step 2: If hash table H contains opkey.
Step 3: then
Step 4: Compare the List of values associated with the opkey.
Step 5: then
Step 6: If all the values do not match, add the opkey to List L1.
Step 7: Else.
Step 8: Add the opkey to List L2.
Step 9: Select Test case according L1 and L2.
```

DOI: 10.15680/IJIRCCE.2017. 0507056

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

V. SIMULATION RESULTS

Fig1 shows result analysis of between existing system and ATTCWSIF proposedsystem. In analysis Blue Bar represent Existing system and Red on represent ATTCWSIF.This Figure shows comparison on 5 parameter. Parameter one is Test case selection in bothsystem same no of test case are selected. Second parameter is Test Automation is done onlyin ATTCWSIF proposed system it was not present in existing system. Third parameter issearching specific element in web service wsdl. Proposed system take very less time than existingsystem. Last parameter is footprints it was in existing system but not shown graphically touser.

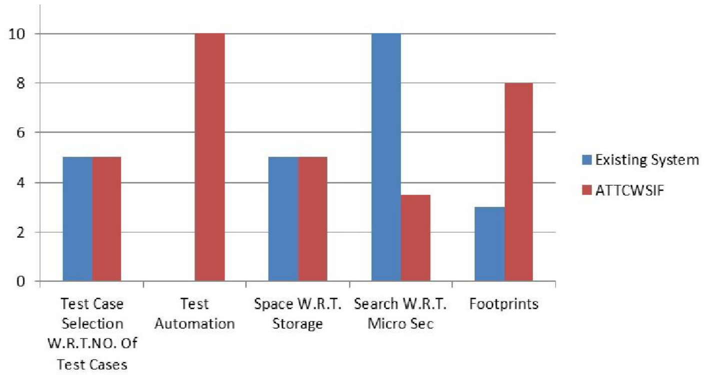

Fig1..Result Table

VI. CONCLUSION AND FUTURE WORK

This research work, investigate the problem of handling complex data type generation, Automation test case selection and perform automation in testing. Changes in operation are identify easily due to hash table. Comparator Algorithm find out difference and selected list of test cases. Finally testing is done with help of Selenium testing framework and report get generated by proposed system with graphical footprints of service

REFERENCES

1. Mark B. Cooray, James H. Hamlyn-Harris, and Robert G. Merkel, Dynamic Test " Reconguration for Composite Web Services ",IEEE TRANSACTIONS ON SERVICES COMPUT ING, VOL. 8, NO. 4, JULY/AUGUST 2015.
2. T. Masood, A. Nadeem and S. Ali, "An automated approach to regression testing of web Services based on WSDL operation changes Emerging Technologies(ICET) " , IEEE 9th International Conference 2013.
3. M. B. Cooray, J. H. Hamlyn-Harris, and R. G. Merke, "Test reconguration for service oriented applications " , in Proc. IEEE Int. Conf. Utility Cloud Comput., 2011, pp. 300305
4. T. M. King and A. S. Ganti, " Migrating autonomic selftesting to the cloud " , in Proc. Int. Conf. Softw. Testing, Verication, Validation Workshops, Paris, France, 2010, pp. 438443.
5. Z. Hong and Z. Yufeng, " Collaborative testing of web services " , IEEE Trans. Service Comput., vol. 5, no. 1, pp. 116130, Jan.Mar. 2012.
6. X. Bai, C. Yinong, and S. Zhongkui, "Adaptive web services testing " , in Proc. 31 Annu. Comput. Softw. Appl. Conf., 2007, pp. 233236.
7. 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. 6679, 2007.
8. W. T. Tsai, R. Paul, Z. Cao, L. Yu, and A. Saimi, " Verication of web services using an enhanced UDDI server " , in Proc. 8th Int.Workshop Object-Oriented Real-Time Dependable Syst., 2003, pp. 131 138.
9. X. Bai, X. Dezheng, D. Guilan, T. Wei-Tek, and C. Yinong, " Dynamic recongurable testing ofserviceoriented architecture " ,in Proc. 31st Annu. Int. Comput. Softw. Appl. Conf., 2007, pp. 368 378.
10. 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. 114121.
11. Antawan Holmes and Marc Kellogg, "Automating Functional Tests Using Selenium " Proceedings of AGILE 2006 Conference (AGILE'06) 0-7695-2562-8/06 2006
12. X Bai, WDong, WT Tsai, Y Chen, WSDL-based Automatic Test Case Generation for Web Services Testing, IEEE International Workshop on Service-Oriented System Engineering
13. (SOSE), Planned Parenthood.215- 220. IEEE Computer Society, (2005), Los Alamitos

BIOGRAPHY

Chetan Nandkumar Phalke is a Student of Master of Engineering at SavitribaiPhule Pune University Pune, India. She received Bachelor of Computer Engineering degree in 2012 from Shivaji University, Kolhapur, MS, India.

Prof. K.D.Bamane is Assistant Professor in Information Technology Department at, Dr. D.Y.Patil College of Engineering,Pune,MS, India, His research interests are Networking ,Software Architecture and Distributed Operating System.