International Journal of Computer and Communication Technology

Volume 8 | Issue 2 Article 15

April 2017

Different Test Cases for e-Governance using Web Services

Sachidananda Patnaik
Department of Computer Science Berhampur University, Bhanja Vihar Berhampur - 760007 India,
sachi_patnaik2004@yahoo.com

Ajita Kumar Misro
Department of Computer Science Berhampur University, Bhanja Vihar Berhampur - 760007 India,
misroajit@gmail.com

Recommended Citation
Patnaik, Sachidananda and Misro, Ajita Kumar (2017) "Different Test Cases for e-Governance using Web Services," *International Journal of Computer and Communication Technology*: Vol. 8 : Iss. 2 , Article 15.
DOI: 10.47893/IJCCT.2017.1416
Available at: https://www.interscience.in/ijcct/vol8/iss2/15

Different Test Cases for e-Governance using Web Services

Sachidananda Patnaik, Ajita Kumar Misro
Department of Computer Science
Berhampur University, Bhanja Vihar
Berhampur - 760007 India
sachi_patnaik2004@yahoo.com, misroajit@gmail.com

Abstract*—SOA is an ideal architectural solution for resolving the adaptation, interoperability and dynamic requirements of information services in e-government applications. Construction and test Web Service solution for e-government is becoming more and more important for e-government adoption. Test cases play an important role in software applications and tracing of bugs in many software projects. In web services there are different techniques and software tools used for accessing the information however sometime, data is updated from different remote locations and give common information. There are different approaches for service composition, ranging from manual to semi-automatic to fully automatic. However it is quite complex when taking independently developed services and integrate them. In e-governance, there are many applications, which work concurrently and give different values to different persons at an instance, which can be traced improperly. In this paper we propose how a service is working properly which is having many services in e-governance applications. Web sites are of different servers and databases but they can integrate by using web services i.e. by using SOA. In SOA architecture a service can test to other service as service requester and what the services are tested as service provider. The trust of service testing is an important criteria while testing a service about its performance, efficiency and accurate.

I. INTRODUCTION

To day, Service Oriented Architecture [SOA] is being adopted by many IT organizations as software infrastructure because it promises to make them more agile and efficient. Due to loosely coupled nature of SOA that enables service components to evolve without needing costly reworks in existing deployments for the IT organizations by e-Governance initiative [3]. Web Applications run large-scale software applications for different commercial domains, organization as well as in many government applications. Web Service provides a common communication infrastructure that is supported by many web application vendors and requesters.

Testing a web service is quite complex work and required many legal aspects before going to test in e-governance applications [1]. One of the most difficult aspects of the web service development is the complexity involved in conducting the effective system testing. In government organizations many different types of works are going on every day having different purposes. Construction of Web Service based test solution is becoming more important for e-Government adoption. A problem that limits the growth of web services is the lack of trustworthiness by the service requesters because they can only see the WSDL document but not the implementation or any source code.

A test case from software engineering point of view is a set of conditions under which a tester can test an application having certain parameter and pre-conditions. A service subscriber has to use different software engineering testing because the design and implementation details of Web Service are not available. Web Service is passing messages and data in structured way.

One of the important thing for designing a test case for web service should be good programming language and communication protocol for transmitting both data and instructions. In this paper, we propose how different test cases are available from software engineering point of view that is applied in e-Governance projects and applications. The benefits of testing include dramatically increased test execution efficiency, better quality, better phase containment, increased speed of processing and reduced cost of both testing and bug fixing.

Though SOA is being increasingly implemented both as green field (top down) and legacy modernization (bottom up), there is no clear cut testing methodologies designed specifically for SOA applications. New approaches and methodologies are necessary to verify and validate applications based on SOA concepts.

When it comes to testing SOA based applications, one has to look beyond functionality and performance (load) testing. SOA testing requires testing of interfaces and services that might bring together diverse systems and platforms, along with other performance (latency) and security related aspects.

II. RESOURCE REQUIREMENT

To design a web service test case first of all we have to know about web service and web service based languages with its syntax and semantics that we can be of help to design a series of test cases. The term "Web Service" is also used in different ways by many professionals who have only passing familiarity with the technology. Hence, Web Service can be defined as an Internet based modular application that uses the Simple Object Access Protocol (SOAP) for the communication and transfer of data in XML format through the Internet with certain protocol. The purpose of XML is to transmit the messages and data from different services having

in any format that includes legacy applications. The Web Service Description Language (WSDL) is used to describe about the way of access Web Service and their operations on data.

SOA based test for web applications means bypassing the web browser and working with web sites through a designed programs. The programs are working and performing their work in the background for the other service. The services are used to test different other services having different parameters. SOA allow application builders search and discover services from service brokers and use services from different service providers. Because service brokers aren't responsible for the quality of the services they refer to or host, service trustworthiness isn't guaranteed. Traditional dependability techniques—correctness proof, fault-tolerant computing, model checking, testing, evaluation, and so on—can improve individual services' trustworthiness. However, implementers must redesign these techniques to handle the dynamic applications composed of services at runtime.

III. EXISTING APPROACHES FOR SERVICE TESTING

Existing approaches for testing a service is based on both white-box and black box testing methods. While white-box testing techniques are based on automatic test case generation from FSM models. In fact, most service requesters would provide just the minimal amount of implementation details for their components, sufficient to allow interoperability with the rest of the platform. Furthermore, white-box techniques would not allow addressing the distributed nature of such platforms. On the other hand, existing black-box testing approaches such as TTCN-3 (Testing and Test Control Notation) are specified typically at a lower level of abstraction, which makes their usage for testing services both less efficient and costly, although the benefits of using such test-specific notations for test development have already been demonstrated in many instances [5]. In previous work [6], we proposed to generate reusable code snippets and libraries of the target test notation based on test patterns to address those concerns. However, as we evolved in our work, it became more and more obvious that what was needed was a model-driven test development process allowing such tests to be specified at a high level of abstraction.

The production of high quality software product requires application of both defect prevention and defect detection techniques. A common defect detection strategy is to subject the product to several phases of testing such as unit, integration and system testing. These testing phases consume significant project resources and cycle time. As software companies continue to search for ways for reducing cycle time and development costs while increasing quality, software-testing processes emerge as a prime target for investigation. In average software testing needs from 40% to 85% of the whole development life cycle. Even if testing is very cost intensive, surveys showed that testing in every stage of the development cycle reduces the whole cost for a system. Testing can help to find errors in the system at an early stage and therefore time and cost intensive rectification of defects can get reduced. In some areas automated testing of software parts is much more efficient and easier than manual testing, even if we like to point out that this is not always the case. Sometimes the configuration for automated tests needs more time than a manual test would do. The current state in software development is that testing is widely recognized as a key success factor, but that automated tests are rarely used [10].

As our research shows, there are tools for automated testing on the market but most of them focus on unit and regression testing. There is less research on how distributed objects; spread around several different machines can be tested. When we work in the area of SOA sometime we face problem that testing can be very crucial, extremely time consuming, and error-prone. Sometimes it needs more time to set up a test environment and to analyze the results than to write the source code itself. Whenever a change in one of the source code parts arises a new test has to be done and very often there is no stable test environment. This then leads to the problem of setting up the whole test environment for every single change. In such cases the investment for testing can explode. There are no reliable surveys which provide information on how much was invested in integration technique in the last years, but IDC predicted that in 2003 50 billion dollars will be invested into software covering integration and there is an upward forecast for the next years. We assume that a large proportion of this investment was due to manual testing the components [9].

If testing can be automated in a desirable way then this would be a big economy of time without decreasing the quality of the software. The approach we are presenting in the following sections tries to show how a test tool could be designed to automate tests for SOAs.

In government organizations there are different departments having their own rules and regulation to process a document in the form of cases. A case is put up or solved in hierarchical manner with in a given time. But the case should be solved by the government rules and regulations, which has already been designed. Government systems have many projects those are implemented by

different departments. A department has many sections and a section has many activities. Sometime, the activities are interrelated with each other and have many interdepartmental activities. A service can be defined a part in a department i.e. a section of a department can provide service to the citizens with Government rules and regulations. A department may have many different services and services are interoperable.

The sections of departments have their own business logics, which are independent to solve the cases or problems by existing rules and regulations. The rules and regulation are designed and amended by the Government for smooth working for the public. So, the rules can be technically designed and termed as a service for automation of the cases. In offices, the cases or problems are flowing from top to bottom and after solving that again goes from bottom to top for finalization. But, sometime the cases are pending due to lake of further intervention which cannot be noticed by the citizen.

IV. TEST MODEL FOR E-GOVERNANCE

Testing activities such as planning, analysis, design and execution must happen throughout the entire SOA project life cycle. There is a model that can be represented as, V-Model suitable test methodology enforces testing discipline throughout the project life cycle. The project starts with defining the User Requirements [11]. The V-Model would recommend that the Business Acceptance Test Criteria for these defined requirements are defined and agreed before moving to the start of the technical design phase.

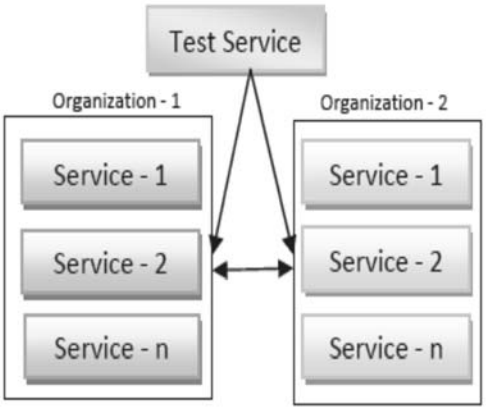

Figure 1. Different Ways of test

Before moving to the next phase/level of technical design, the model recommends test criteria defined for that level of technical requirements, and so on. The V-Model is simply encouraging the project team to continually determine how they would successfully test the project deliverables.

In e-Governance, there are many components for smooth working for common people. The components are Technical Components, Social Components and Cultural Components and Physiological Components. The various components of e-Governance and their inter relationship with each other taken as most essential while we designing our proposed test case model along with interoperability of service components like information, interactive transactional as well as different and finishing matrixes like process logic, accounting, computation, security and self services. These components are categorized as per the administration point of view [9].

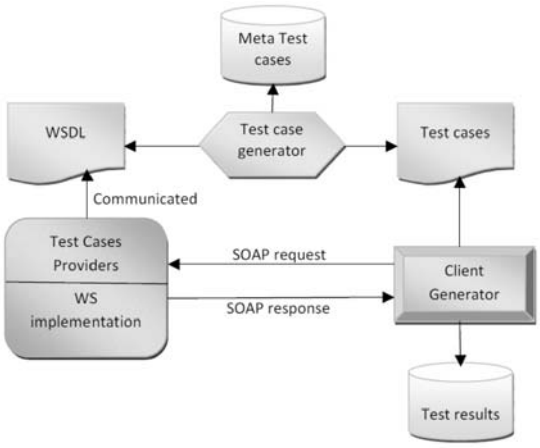

Figure 2. SOA Based architecture for test cases

The Test-Model is a suitable test methodology to deliver SOA projects for the following reasons:

- This model helps from top-down approach with respect to defining the business process requirements, high-level functional, technical design, security etc.
- It then performs a bottom-up test approach that individual functions within a service can test easily. SOA is 'loosely' coupled services and that is why a bottom up test approach is helpful.
- The levels reflect different viewpoints of testing on different levels of detail.
- The model helps in testing throughout the entire Software Development Life Cycle and projects of different components.

The above points help while testing any components of e-Governance or from organization point of view. A component is a function or process logic which is a part of a service.

V. TESTING STRATEGY

In e-Governance applications there are several components, which are running simultaneously in different location with different workloads and having different activities. There are different servers located at different locations having different configurations.

A. Service Verification

The service provider has designed a service and that is published in the UDDI for requesting by the Service requesters. The service requesters request the services according to their requirements. By using service verification some facility is provided to test these services to guarantee that the service used is a proper one.

VI. SOFTWARE TESTING FOR E-GOVERNENCE

Testing like Black Box, White Box & Gary Box are essential for deploying robust, scalable, interoperable and secure Web Services in e-governance.

A. Black Box Testing

Black Box testers do not have access to the source code and are oblivious of the system architecture. A Black Box tester typically interacts with a system through a user interface by providing inputs and examining outputs without knowing where and how the inputs were operated upon. In Black Box testing, target software is exercised over a range of inputs and the outputs are observed for correctness.

B. White Box Testing

White Box testing refers to the technique of testing a system with knowledge of the internals of the system. White Box testers have access to the source code and are aware of the system architecture. A White Box tester typically analyzes source code, derives test cases from knowledge about the source code, and finally targets specific code paths to achieve a certain level of code coverage [12].

C. Gray Box Testing

Gray Box testing refers to the technique of testing a system with limited knowledge of the internals of the system. Gray Box testers have access to detailed design documents with information beyond requirement documents. Gray Box tests are generated based on information such as state-based models or architecture diagrams of the target system.

VII. TOOLS FOR TESTING

SOA testing involves the ability to test SOAP and XML based messaging against a service endpoint in order to assess the robustness, reliability, and resilience of the service. Comprehensive testing of a service endpoint involves 4 primary focus areas: Functional, Performance, Interoperability, and Security. There are different testing tools available including open source tools for testing a service.

A Web service is, after all, a collection of procedures or business process logics exposed to the unforgiving thoroughfare of the Net (be it inter- or intra-). In this roundup, three tools that purport to verify that Web services do what they are supposed to do, that they resist graceless failure, and (in some cases) that they conduct themselves with efficiency. The tools are *soapUI*, *TestMaker*, and *WebInject*. All are open source, and are available for free download and incorporation into next Web services project.

A. SoapUI 1.6

SoapUI 1.6 is a Java-based tool from Eviware. This version executes within its own stand-alone UI; the new 1.7 release includes plug-ins for the NetBeans, IntelliJ, and EclipseIDEs.

The user interface conforms to the architecture of the typical IDE: a navigation pane on the left, a content pane on the right, and additional properties panes tucked near the bottom [13].

SoapUI arranges work into projects. Each project is primarily identified by the interfaces that the project is built to test. Here, an interface is the “other end” of a URI (uniform resource identifier) pointing to a site that is exposing Web service methods. You can quickly generate a skeletal project by aiming an empty project at a Web service’s WSDL code; soapUI will accept WSDL from either a file or a Web service end point that transmits the WSDL for its services.

B. TestMaker

TestMaker has different versions but the latest version is TestMaker 5.5 and has many features for testing SOA services. TestMaker 5.5 is a major feature enhancement and bug fix release. The new software testing tool includes major feature improvements in the areas like Grid testing, SOA testing Cloud testing, Log testing.

C. WebInJet

WebInject is a free tool for automated testing of web applications and web services. It can be used to test individual system components that have HTTP interfaces (JSP, ASP, CGI, PHP, AJAX, Servlets, HTML Forms, XML/SOAP Web Services, REST, etc), and can be used as a test harness to create a suite of [HTTP level] automated functional, acceptance, and regression tests. A test harness allows to run many test cases and collect/report many results. WebInject offers real-time results display and may also be used for monitoring system response times [14].

WebInject can be used as a complete test framework that is controlled by the WebInject User Interface (GUI). Optionally, it can be used as a standalone test runner (text/

console application) which can be integrated and called from other test frameworks or applications.
The above tools are helpful to design from top to bottom of different components of services. A service component can be tested individually and the result can be taken as further input for other services.

VIII. CONCLUSION

Testing plays very important role in any type of applications. By testing, the system or application is good for longer time and can give high performance. By using SOA, we can design many number of test cases and that can implement in any systems with respective of hardware and software. Hence, in e-Governance application, there is much application software for different types of applications. Hence, it can be concluded that the SOA architecture is very much helpful for testing of different application of e-Governance.

REFERENCES

[1] A. Askaruinisa and A.M. Abirami, "Test Case Reduction Techniques for Semantic Based Web Services", *International Journal on Computer Science and Engineering (IJCSE) Vol. 02, No. 03, 2010, 566-576.*

[2] A. B. Arntzen and A. Krogsrud, "Web-Service Architecture: A Solution for e-government application".

[3] A.G.V. Fevudjio and I. Schieferdecker, "Availability testing for Web Services", *Telektronikk 1.2009 ISSN 0085-7130 © Telenor ASA 2009.*

[4] B.K. Aichernig, "Qualification of Open Source Software for e-Government", *e-Maco Report 2008, Version 1.0, January 2006.*

[5] C. Ghezzi and S. Guinea, "Run-Time Monitoring in Service Oriented Architecture", *http://www.springer.com/978-3-540-72911-2.*

[6] H.M. Sneed, "Testing and e-Government web Site". *Proceedings of the 2005 Seventh IEEE International Symposium on Web Site Evolution (WSE'05).*

[7] K. Pasklaeva-Shapira, "Transitioning from e-Government to e-Governance in the Knowledge Society: The Role of the Legal Framework for Enabling the Process in the European Union's Countries"

[8] L. Mei, Z. Zhang, W.K. Chan and T.H. Tse, "Test Case Prioritization for regression testing of Service-Oriented Business Application", *Track: Web Engineering / Session: Service Oriented Developemtn, WWW 2009 Madrid.*

[9] R. K. Das, S. Patnaik and A.K. Misro "Adoption of Cloud Computing in e-Governance", CCSIT 2011, Part III, CCIS 133, PP. 161 – 172, Springer Verlag Berlin Heidelberg 2011.

[10] S. Hanna and M. Munro, "Fault-based Web Services Testing", *Fifth International Conference on Information Technology: New Generations.*

[11] S.P. Lee, L.P. Chan and E.W. Lee, "Web Service Implementation Methodology for SOA Application", *1-4244-9701-0/06/$20.00 ©2006 IEEE.*

[12] Y. Zheng, J. Zhou and P.Krause, "An Automatic Test Case Generation Framework for Web Service", *Journal of Software, Vol. 2, No. 3, September 2007.*

[13] Wu Deng, Shuqin Liu and Jingjing Liu "Automation Testing Process Modeling Method of SOA-based Isomerous Software" *2009 International Conference on Industrial Mechatronics and Automation.*

[14] http://www.soapui.org/SOAP-and-WSDL/getting-started.html

[15] *http://www.webinject.org*