ISSN: 2278 – 7798
International Journal of Science, Engineering and Technology Research (IJSETR)
Volume 2, Issue 12, December 2013

A STUDY OF TESTING TECHNIQUES FOR WEB APPLICATIONS

Dr. Manju Kaushik
Associate Professor, Computer Science and Engineering, JECRC University, Jaipur
Garima Singh
Research Scholar, Computer Science and Engineering, JECRC University, Jaipur

ABSTRACT: *There are so many techniques developed and implemented to test web applications. As web industry is growing very fast, and many applications are running for business whether it customer to customer, customer to business all need to be tested effectively for good results. Here in this paper we are discussing different types of web applications testing techniques like regression testing, user-session based testing, penetration testing, reliability, mutation, smoke, parallel, recovery, installation, compatibility and user interface testing.*

KEYWORDS: Web Applications, Web Testing Techniques, Test Cases

INTRODUCTION

Web testing is an approach in which any application is tested to find out the bugs or error so as to correct them in the initial stage of their detection and to verify that the application is functioning correctly. There are so many techniques to test functional and non functional aspects of web applications. We are discussing some of the testing techniques which are performed to test web applications.

REGRESSION TESTING

The proposed regression testing technique is automated, which means that we run the test cases manually one time and later in regression testing, the test cases are automatically executed and validated. Researcher highlights the algorithm for this technique and the basic structure of a custom tool to be used in this technique [1].

The basic steps of the technique are as follows:

1. Create test cases for the application and specify input data.
2. Use the developed tool with the embedded browser to run the test cases recording the visited URLs and the submitted arguments and form values.

3. While running the test, the developed tool saves HTML output for later comparison and validation in the regression testing stage.
4. In the regression testing step, the tool executes the sequence of saved URLs automatically and collects the output values specified in step 3. After executing the test cases, the tool compares the output values collected in this step with those collected in step 2 and it provides the user with the sections that produced different output.

The tester will have to analyze those differences manually.

USER-SESSION-BASED TESTING

In user-session-based testing, data is collected from users of a web application by the web server. Each *user session* is a collection of user requests in the form request and name-value pairs (e.g., form field data). A base request for a web application is the request type and resource location without associated data (e.g.,*GET /servlets/authentication/Login.jsp*). More specifically, a user session is defined as beginning when a request from a new IP address reaches the server and ending when the user leaves the web site or the session times out. To transform a user session into a test case, each logged request of the user session is changed into an HTTP request that can be sent to a web server. A test case consists of a set of HTTP requests that are associated with each user session [2].

PENETRATION TESTING

Penetration testing has been a common technique used to test network security for many years. It is also commonly known as black box testing or ethical hacking. Penetration testing is essentially the “art” of testing a running application remotely, without knowing the inner workings of the application itself, to find security vulnerabilities. Typically, the penetration test team would have access to an application as if they were users. The tester acts like an attacker and attempts to find and exploit vulnerabilities.

RELIABILITY TESTING

Users expect a web application to be available when needed and that any transactions performed will be executed consistently. Again, it is important to remember that reliability must be measured from the perception of the user. Testing must reflect the expectations of the customer.

For example, customers using an on-line trading application expect their trades to be successfully executed in seconds on a consistent basis.

MUTATION TESTING

The purpose of mutation testing is to assess the quality of test suite in terms of failure detection. Here we can change small syntactic changes to a program. The updated program is called a Mutant. If a test suite is able to detect the deviation, the mutant is said to be killed and mutation score is the number of killed mutants divided by the number of mutants. Test Suite is the collection of test cases [3].

SMOKE TESTING

Smoke testing is used to check the testability of the application. It is also called 'Build Verification testing or link testing. This type of testing check whether the application is ready for further major testing and working, without dealing with the finer details.

PARALLEL TESTING

This type of testing is done by comparing results from two different applications like manual verses automated testing.

RECOVERY TESTING

This type of testing is done to know how fast the application is able to recover against any type of failure like server failure.

INSTALLATION TESTING

This type of testing identifies the way in which installation procedure leads to incorrect results.

COMPATIBILITY TESTING

This type of testing determines if an application under supported configuration perform as expected, with various combination of hardware and software packages.

USER INTERFACE TESTING

This type of testing is performed to check how user friendly the application is? The user should be able to use the application, without any assistance by the system personnel.

SOME MORE FUNCTIONAL TESTING OF WEB APPLICATION TESTING TECHNIQUES [4]:

The previous section provided an overview of WA testing. This section will describe recent state-of-the-art works on WA functional testing approaches.

Input Validation Testing

The first work proposed a white box technique that implemented input validation testing of WAs [5]. User inputs are partly essential in determining the accuracy of an application's output. Even if each input in a WA is validated, the control mechanism itself needs to be tested to ensure that it is performing as expected. Thus, the need of

test approach in this particular area arises. This approach produce test cases for input validation testing using TIVT (Tool for Input Validation Testing), a prototype testing tool from Georgia Institute of Technology. The test cases are obtained by recovering input validation model in the form of Validation Flow Graph from the respective WA's program source code automatically. The approach effectively detected most of the expected errors, with a few tester managed to detect all expected error in a WA. While the input validation area remains as one of the important features in a WA testing that has received fair amount of attention from researchers, other feature-oriented WA testing also has potentials for exploration.

Navigation Flow & Control Flow Model Testing

Ricca and Tonella [6] propose an approach for white box testing that is more suitable for static WAs. This approach focused on the navigation model at the higher level and control flow model for the lower level of a WA. Test cases are defined through structural coverage of a WA obtained by tracing its navigation and control flow. However, the work is far from producing a definitive conclusion, as there is still room for further improvements and exploration. Nuo et al. [7] later extended the approach in the aspect of utilizing navigation model as test model in a proposed model-driven methodology. A framework is also proposed to support the methodology and mainly features a modeler and tester. Even so, automatic generation of test data is only partly achievable and the flexibility of the test model can be emphasized through enabling rules establishment form modifying models.

Event Flow Testing

Another proposal introduces a new testing coverage for WA testing using event flow testing technique [8], data flow testing technique is applied to an enhanced dependence graph called event-based dependence graph model which is newly introduced. However, this technique exploits the event-driven feature in the .NET environment and is only particular for testing WAs in the said environment.

Object-Oriented Model Testing

Di Lucca et al. [9] propose a two-stage black box testing approach which comprises of a unit test (testing a single page) as the first stage and integration test (testing a set of Web pages) as the second stage. The approach utilizes a WA object-oriented model as test model through the implementation of decision tables. The test models are then tested using a testing

toolkit called WAT (Web Application Testing) which automated the testing process. Although the tool managed to automate testing process and reduce test time, this approach is only applicable for pages with feasible paths.

Finite State Machine (FSM) Model

Andrews et al. [10] utilizes FSM for modeling software behavior and deriving test cases from them. Called FSM Web technique, it is realized by building FSM model of a WA's subsystem, the subsequences of a state in FSM is combine to generate the complete test. Furthermore, the author proposes an automated tool that can support certain phases in the approach. However, the proposed tool is only conceptually sound. Therefore realization of the proposed tool is requisite before its effectiveness in supporting the approach can be determined empirically.

Input, Output, Preconditions and Effects (IOPE)

Paradkaret al.[11] proposed a test approach that is specified at testing web services

Gray box testing is a hybrid of white box and black box testing. Its primary aim is testing a piece of software against its specification but using some knowledge of its internal working [9]. This testing strategy is well-utilized in a WA functional test approach in the aspect of manipulating captured data from user sessions to produce test suites and includes them into a white box testing implementation for fault detection purposes [12]. By comparing and combining both techniques, the effectiveness of the techniques is presented through finding different types of fault.

Still, the possibility that certain aspect of the approach can be used in a general WA testing must not be ruled out. This approach makes use of the Precondition and Effect pairs in the IOPE information that is usually present in a web service and refines them based on fault models to generate testing goals. The testing process is automated using the planner component and the test cases are then validated by a verification sequence generated in the approach. The adequate savings in effort for requirement coverage and effectiveness of the fault detection is apparent in the approach. The work can be extended by making the approach adaptable to general WA to some extent.

User Session Data Manipulation

Test Suite Reduction

Sampath et al. [2] expands the works by further proposing a technique on minimizing the size of test suite by clustering the user-sessions using concept analysis. Compared to the original test suite that has not been minimized, the reduced test suite produces lesser result. However, the author states that it is not a significant loss and the notion of reducing test suites can be explored further with other alternate algorithms which may produce a higher fault detection capability.

Functional Scenario Testing

Huang et al. [13] proposed that the UML extended activity diagram is utilized to produce test cases. The work is focused on the functional scenario testing aspect of a WA as functional scenario can be depicted by the activity diagram. The test cases are essentially in the form of testing codes based on Http Unit, a web testing tool. The generation of the test cases is done using Web Application Scenario Automated Testing Tool (WASATT). The test codes can then be compiled and run by the testers. Huang et al. proposed that this approach can potentially reduce artificial errors which are usually present when testers try to self-

Test Case Design Approach

The test case design approach which is proposed by[15] utilizes UML sequence diagram of a WA to generate three types of test cases which will be tested level by level, starting with the smallest unit (single web page testing), followed by mutual web page testing and lastly integrated web page

defined test models inaccurately. The work can be further extended to completely model all functional properties of WA, accurately validate web page's dynamic semantics and accommodate the behavior of concurrent user access.

Test-First Design Approach

Antawan and Marc[14] provide an acute description of Selenium, a functional test tool for WA Selenium, an open-source project for in-browser testing uniquely offers a possible test-first design of WA for its users, a customer acceptance test and an automatic regression test bed for web-tier. As an agile testing tool, Selenium's capabilities in writing and maintaining test scripts surpasses other testing tool such as Canoo Web Test, Http Unit and Quick Test Professional. Furthermore, Selenium allows its users to write test codes in a variety of programming languages, thus eliminating the need of having different test tools for WAs developed in different languages and eases testing task for testers. The author further reports that while Selenium's current performance is adequate, its future potential in becoming a powerful WA testing application is reflected by the constant growth of Selenium's active community.

testing. The information of the test case was derived solely from sequence diagram and testing was performed with the aid of a testing tool called OnlineTestWeb. The approach demonstrated was fairly simple and straightforward and easy to comprehend. Unfortunately the tool used was no longer available on the Internet.

REFERENCES

1. Hamzeh Al Shaar and RamziHaraty, "Modeling and automated black box regression testing of web applications", *Journal of Theoretical and Applied Information Technology*, **Vol. 4 No 12**. Page 1182-1198, (2005).
2. S. Sampath, R. Bryce, G. Viswanath, V. Kandimalla, and A.G. Koru, "Prioritizing User-Session-Based Test Cases for Web Application Testing," In proceedings of IEEE International conference on Software Testing, Verification, and Validation, Page. 141-150 (2008).

3. Alexander Pretschner, TejeddineMouelhi, Yves Le Traon, "Model-Based Tests for Access Control Policies", *International Journal of software testing, verification and validation*, **Vol.3**, page 9-11 (2008).

4. M.Y. Suhaila, W.K. Wan Mohd Nasir, "An Outlook of State-of-the-Art Approaches Testing of Web Application" In proceedings International multiconference of engineers and computer scientists, **Vol. 1,** Page 16-18(2011).

5.Liu, H. and T.H.B. Kuan, "Testing input validation in Web applications through automated model recovery*"Journal of Systems and Software*, **Vol. 2. No. 81**. Page 222-223, (2008).

6. Ricca. Filippo and Tonella, "A 2-Layer Model for the White-Box Testing of Web Applications", in proceedings international conference on telecommunication energy, Chicago, Page 11-19, (2004).

7. Nuo Li, Qin-qin Ma, Ji Wu, Mao-zhong Jin, Chao Liu, "A Framework of Model-Driven Web Application Testing", In proceedings of 30th Annual International Computer Software and Applications Conference, Chicago, Illinois, Page.157-162, (2006).

8. Mansour, N. and M. Houri, "Testing web applications.

Information and Software Technology", *Information and software technology,* **Volume 48, Issue 1,** Pages 31–42 (2006).

9. Giuseppe A. Di Lucca and Anna Rita Fasolino, "Testing Web-based applications: The state of the art and future trends", *Information and Software Technology***Vol.48**, page 1172–1186 (2006).

10. Anneliese A. Andrews, Je Offutt, Roger T. Alexander, "Testing Web Applications by Modeling with FSMs", *Software & Systems Modeling,* **Vol. 4, Issue 3**, Page 326-345 (2003).

11. Paradkar, A.M., Sinha A, William C, "Automated Functional Conformance TestGeneration for Semantic Web Services", IEEE International Conference on web services, Salt Lake City, UT, Page 110- 117(2007).

12.Elbaum, S., S. Karre, and G. Rothermel, "Improving web application testing with user session data", In proceedings on International conference on software engineering, Portland, Page 49-59, (2003).

13. Huang, C.-H. and H.Y. Chen, "A tool to support automated testing application scenario", In proceedings of International conference on system, man, cybernetics, United States, Page 2179 – 2184, (2007).

14. Antawan, H. and K. Marc,"Automating Functional Tests Using Selenium", in Proceedings of the conference on AGILE,IEEE Computer Society, Minneapolis, Minnesota, Page 270-275 (2006).

15. Cho, Y., W. Lee, and K. Chong, "The Technique of Test Case Design Based on the UML Sequence Diagram for the Development of Web Application", in proceedings of International conference on Computational Science and its Applications – ICCSA, Singapore, Page 9-12 (2005).