Testing of Web Applications Based on Model Driven Architecture

Harsimrat Kaur[1], Janpreet Sigh[2]
[1]Assistant Professor, Punjab Business School, Fatehgarh Sahib, (Punjab), India
[1, 2] Assistant Professor, Department of Computer Science, Lovely Professional University, Phagwara, (Punjab), India
[1]roop1231@gmail.com, [2]Janpreet16@gmail.com

Abstract: **Web applications currently make up one of the largest growth areas in software Web [3]. During the analysis phase of web application development web model of the web application is created. This web model represents all the components of the web application that are to be tested. After generating the web model the very next step that is taken for testing the web application is to generate test cases. Basically** actual **testing begins on the test cases [9, 15]. This testing is followed up by regression testing if we bring any feasible change in the web model. . The regression testing verify whether previous model functionality will regress or fail by marking which previous test cases are valid or discarded in the updated model [12]. The updated web model is then used to generate the new test cases.**

I. Introduction

Web testing is the name given to software testing that focuses on web applications. Complete testing of a web-based system can help in addressing the issues before the system is revealed to the public [2]. Quality of the web site is very important to the user otherwise the user will leave for the different site if the web site is too complex and is of low quality. So methodologies for adequately analyzing, constructing, understanding, testing and maintaining web applications are essential. The criteria for testing websites are: Timeliness, Quality, Accuracy and Consistency, Response Time and Latency, and Performance. Issues such as the security and load and performance of the web application are the basic functionalities to be considered. But all these features are difficult to achieve because of certain constraints and challenges that make Web testing is a difficult task [6]. The problems that can occur depend on the application, technologies used, settings and environment, user software configurations etc. The current testing field focus is on techniques for unit–level testing and sub system testing [17, 18]. However web applications testing lack systematic testing methods and techniques to cope with challenges and constraints of testing. Currently the common strategies and techniques for web testing are adopted and usually driven by specific tools, metrics or immediacy rather than the objectives to achieve.

II. Objectives

1. Identifying and analyzing the domain areas of Web applications.
2. Representation of Web applications by using suitable Web Architecture.
3. To identify the key attributes for Web Testing.
4. To select a specific testing strategy to be applied on the Web components.
5. Application of Web Testing techniques at different levels.
6. Verification and validation of results.

III. Proposed Methodology

Thus our proposed framework for testing of web application is as follows:

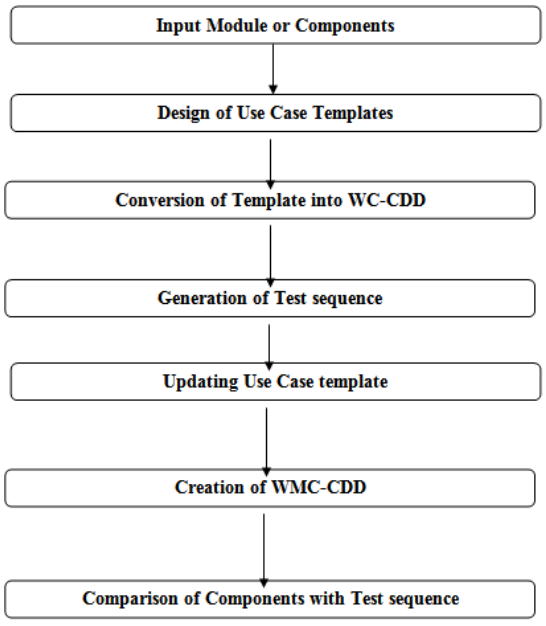

Fig1: Proposed Methodology

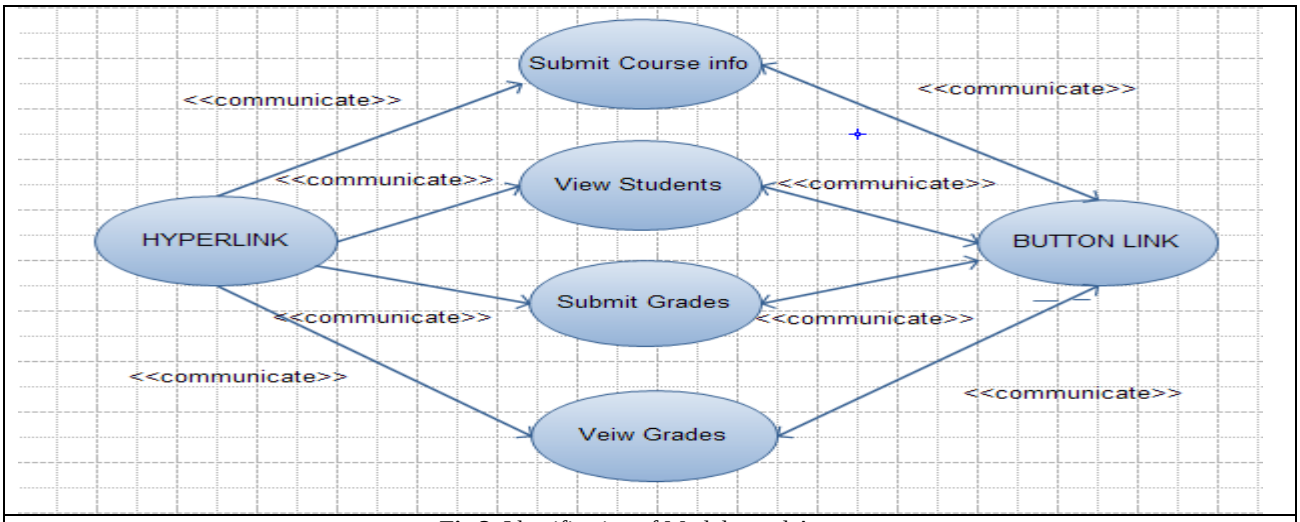

Fig 2. Identification of Modules and Actors

Table I Introduction about Use Cases

Use case name:	It includes the name of the use case.
Actor:	An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks.
Preconditions:	List any activities that must take place, or any conditions that must be true, before the use case can be started.
Post conditions:	Describe the state of the system at the conclusion of the use case execution. Post-conditions can be numbered like preconditions.
Base Flow of events:	Provide a detailed description of the user actions and system responses that will take place during execution of the use case under normal, expected conditions.
Alternate flow of events:	Describe any anticipated error conditions that could occur during execution of the use case, define how the system is to respond to those conditions and cover behaviour of an exceptional scenario

The proposed methodology is a combination of these seven steps. In the very beginning the web application is modeled into the high level use cases diagrams called UCTMs (Use Case Transition Models).

In the second step use case diagram is designed into use case templates that show the functionality of Web application. Testing method is implemented by transferring the textual description of use cases into WC- CDDs (Constraint Directed Diagrams). Thereafter the test case sequences are generated by traversing the WC-CDD. In the next phase we have worked to generate the regression Testing Suite by inserting the changes in the use case template and converting WC-CDD into MWC-CDD. The test sequences so obtained from it are then validated by giving the statistics like which old test sequences are reused, how many are updated or modified and newly added. This will lead to reduce the time while regression testing, as we need not to test sequences again which are used without any modifications

A. Design of use case Template

B. Conversion of Template into WC-CDD (Web component Constraint directed Diagrams)

In order to get the test scenarios conveniently, we transform the textual description of use cases to the WC- CDD. Because the flow of events are the most important part for generating test cases from use cases, we only transform the flow of events of use cases to the WC- CDD.

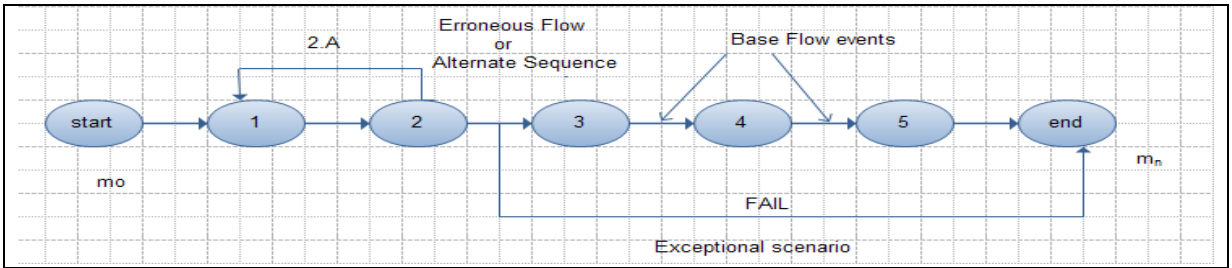

Fig. 3. Test Case Generation

- The vertexes in WC-CDD denote actions in base flow of events.
- The arcs with constraint ".A" correspond to actions in alternate flow of events.

In order to control combinatorial explosion, we propose the CSC (Constraint Scenario Coverage) criterion that means in CDD must be covered at least once.

C. Generation of Test sequences

This step gives the output as the test sequences by traversing the WC-CDD. The number of generated test scenarios from WC-CDD satisfying the constraint scenario coverage is less than traditional methods.

Table II
Test Sequences

Main flow or successful path	Start, 1, 2, 3, 4, 5,6,7,8, end.
Exceptional scenarios	Start, 1, 2, 1, 2, end. Start, 1, 2, 3, 4, 3, 4, end. Start, 1, 2, 3, 4, 5, 6, 7, 6, 7, end.
Erroneous scenarios	Start, 1, 2, 1. Start, 1, 2, 3, 4, 3. Start, 1, 2, 3, 4, 5, 6, 7, 6.

D. Insertion of changes in Use case Templates

Dynamic Insertion of any number of changes in the use case template adds flexibility to cope with the changing user requirements.

- ✓ Addition of new base flow event.
- ✓ Addition of alternate flow event.
- ✓ Deletion of any flow event.

IV. Conclusion

As we observed from the literature survey that the dynamic nature of websites requires updates and modifications time to time. During the development of websites the different modules are tested after development of those modules by creating their use cases and then by making test case for each use case. And when some modifications have to be done on websites during their modification process then we make test cases for only the modified aspects of the websites to test the performance of new inserted portions. But we ignore the fact that the new insertions may disturb the test case already generated for the existing modules as the modified portion may interact with the existing modules. Also this may lead to software disaster or unacceptable conditions by website users. I proposed a solution for the above problem that whenever there is a requirement to modify a website by inserting some new modules then during the testing of new modules when we create use cases for those new modules and generate test cases for each use case then we have to consider the previous use cases of the existing modules which may be affected with the introduction of new modules and we have to updated the test cases of those existing modules on the basis of their new use

cases then check the overall performance of website. This technique definitely will help to reduce the unwanted effects produced during the insertion of new modules.

References

[1] E.Hieatt, R.Mee, "Going Faster: Testing the web applications", IEEE Software, Mar.2002, pp.60-65.

[2] A. Andrew, J. Offutt, R. Alexander, "Testing Web Applications by Modelling with FSMs", Software and Systems Modelling. 2004, pp. 326-345.

[3] Jeff Offutt, Ye Wu, Xiaochen Du and Hong Huang, " Web Application Bypass Testing" , In Proceeding of the 28th Annual International Computer Software and Applications Conference (COMPSAC 2004), Oct. 2004, pp. 106-109.

[4] Y. Qi, D. Kung, E. Wong, "An Agent-Based testing Approach for Web Applications", In Proceedings of the 29th Annual International Computer Software and Applications Conference (COMPSAC'05), 26-28 July.2005, pp. 45-50.

[5] Adriana Carniello, Mario Jino, Marcos Lordello Chaim , "Structural Testing with use cases", JCS&T Vol. 5 (No. 2), Aug. 2005, pp. 100-106.

[6] Ghulam Mustafa, Abad Ali Shah, Khadim Hussain Asif and Amjad Ali, "A Strategy for Testing Web Based Software", Information Technology Journal 6(1): 74-81, 2007.

[7] Shengbo Chen, Huaikou Miao, Zhongsheng Qian , Shanghai Univ,Shanghai, "Automatic Generating Test Cases for Testing Web Applications", International Conference on Computational Intelligence and Security Workshops, Dec. 2007, pp.881-885.

[8] Atifah Ali, Aamer Nadeem, Muhammad Zohaib Z. Iqbal and Mohammad Usman, "Regression Testing based on UML Design Models", In 13th IEEE International Symposium on Pacific Rim Dependable Computing, Dec.2007, pp.85-88.

[9] Hassan Reza, Kirk Ogaard, Amarnath Malge, "A Model Based Testing Technique to Test Web Applications Using Statecharts", Fifth International Conference on Information Technology, Apr. 2008, pp. 183 – 188.

[10] N. Alshahwan, M. Harman, "Automated Session Data Repair for Web application Regression Testing", International Conference on Software Testing, Verification and Validation, Apr. 2008, pp. 298 -307.

[11] Huaikou Miao, Zhongsheng Qian and Bo Song, "Towards Automatically Generating Test Paths for Web Application Testing", International Symposium on Theoretical Aspects of Software Engineering, June. 2008, pp. 211 – 218.

[12] Javier J. Gutierrez, Maria J. Escalona, Manuel Mejias, Jesus Torres, Arturo H. Centeno " A Case study for generating Test cases from Use cases", In second International Conference on Research Challenges in Information Science, June. 2008, pp. 209-214.

[13] Liping Li, Huaikou Miao, Zhongsheng Qian, "UML based approach to testing web applications" International Symposium on Computer Science and Computational Technology, Dec. 2008, pp. 397-401.

[14] Liping Li, Huaikou Miao "An approach to modelling and testing applications based on use case", International Symposium on Information Science and Engineering, Dec.2008, pp. 506-510.

[15] Ananya Kanjilal Swapan Bhattacharya and Sabnam Sengupta, "Analysis of Object-Oriented Design: A Metrics Based Approach", In TENCON 2009 - 2009 IEEE Region 10 Conference, Jan. 2009, pp. 1-6.

[16] B. Kam, T. R. Dean, "Lessons Learned from the Survey of Web Applications Testing", Sixth International Conference on Information Technology: New generation, Apr.2009, pp. 125-130.

[17] Z. S. Qian, "An Approach to Testing Web Applications Based on Probable FSM", International Forum on Information Technology and Applications, May.2009, pp. 519-522.

[18] Gabriela Robiolo, Cristina Badano and Ricardo Orosco, "Transactions and Paths: two use case based metrics which improve the early effort estimation", In Third International Symposium on Empirical Software Engineering and Measurement, Oct.2009, pp.422-425.

[19] Ashalatha Nayak and Debasis Samanta, " Automatic Test Data Synthesis using UML Sequence Diagrams", In journal of Object Technology, vol.09 (No.2), March-April 2010, pp. 75-104.

[20] Hesham A. Hassan and Zahraa. E. Yousif, "Generating test cases for platform independent model by using use case model" In International Journal of Engineering Science and Technology, Vol. 2(12), 2010, pp. 7665-7675.