

Survey of Web Testing Techniques

Sonal Anand

M.Tech (Computer Science)
USIT, GGSIPU
New Delhi, India

Anju Saha

Assistant Professor
USIT, GGSIPU
New Delhi, India

ABSTRACT

This paper presents a survey of various web application testing techniques and provides a comparison of techniques, concluding with benefits and limitations of each.

Keywords

Web application testing; testing technique; testing methodology; web applications, testing criteria; testing concept, feature of testing

1. INTRODUCTION

IEEE has defined software testing as the process of evaluating a software system to verify that it satisfies specified requirements [11]. Web application testing is a specialization of software testing. It focuses on identification of errors in a web application to verify whether the output meets the specification. Web applications are dynamic and interactive, as compared to traditional applications. Therefore, traditional testing techniques cannot successfully be incorporated in testing web applications. A variety of web application testing techniques has been proposed. The organization of the paper is as follows. Section 2 provides the literature review. In Section 3, a classification of the techniques has been provided, by categorizing them into the types of testing approaches, along with the features focused upon. In Section 4, the techniques have been compared, based on the testing feature, concept, testing method, target, testing level and the approach used. In Section 5, the benefits and limitations of each technique has been discussed. Section 6 discusses the future scope of the survey.

2. LITERATURE REVIEW

In this section, a review of various web application testing techniques is given.

Qian Zhongsheng [2] has proposed a web testing technique based on probable FSM $Q = (S, \Sigma, \delta, PR, s_0, F)$ where S is a set of states (web pages), Σ denotes a set of transitions among pages, δ is a single valued transition function, PR is set of probabilities of transitions (execution frequency of various parts of application may vary), s_0 is the unique initial state and F is the set of final states. Test cases are generated by first calculating the independent paths from state transition diagram based on cyclomatic complexity, where the probability of whole path is the product of probabilities of individual transitions in that path. Possible test cases with probabilities above threshold can be generated to cover frequently occurring operations.

Qian, [3] Mia and Zeng have proposed a web testing model

that constructs a Page Flow Diagram (PFD) of web pages and transitions followed by conversion into a Page Test Tree (PTT), a spanning tree, from which a test translator is employed to generate test cases, and finally execute them to generate test report.

Lei Xu and Baowen Xu [4] introduced a framework for web testing beginning with requirement analysis based on object model, interactive relation model and architectural model, to test case generation via combinatorial method, followed by execution of test cases.

Wang Hai-rui, LI Ya and LI Xin-jie's mobile agent based model for quality testing which provides a single framework for distributed applications [5]. The mobile agent moves through heterogeneous networks and communicates with hosts and other agents. The approach begins with "Initialization" phase wherein web page of agent server is accessed, and the agent server assigns values to agent to perform automated testing on its own. The next step is to "Dispatch" the agent in cyber space. Then in "Execution" phase, the agent performs automated testing. In "Report" phase, the sever status is reported by agent to client.

Kung, Liu and Hsia have presented an object oriented approach for web testing [6], to capture structural and behavioral test artifacts of web applications. The entities are represented as objects, and their structures, relationships and dynamic behaviors are described. An object oriented Web Test Model (WTM) has been proposed which represents artifacts from object, behavior and structure features. For object perspective, entities are depicted by Object Relational Diagram (ORD). For behavior perspective; a Page Navigation Diagram (PND) describes the navigation behavior among pages. For structural perspective, flow graphs are used to describe control flow and data flow information of scripts or functions in web applications.

Guangzhou Jiang and Shujuan Jiang have presented a quick test model for performance testing [7], based on testing flow of web applications. It introduces a new performance index called successful request rate apart from other concepts like number of users and response time in order to test the performance of web application. It also contributes a testing method, combined with LoadRunner tool to provide effective solution to quick performance testing. The quick testing process begins with planning the test, followed by LoadRunner script creation, scenario definition, execution and result analysis.

Song, Miao and Chen have proposed a model based on web frameset and browser interactions [8]. Web framesets are employed in applications where layouts of some pages are identical. The behavior of browser has a critical impact on

functioning of web applications, since an application may be functionally sound but may not work due to an irregular functioning browser. The process begins with modeling of application with framesets, followed by considering browser interactions along with framesets, modeling the web navigation, formalizing the navigation model, and finally generating and executing test cases.

Li, Mio and Qian have proposed UML Based approach [9] that models a large web application as hierarchical profile use case diagrams called Use Case Transition Model (UCTM). Traversing the UCTM from top to down, each use case is described as a sequence diagram, which automatically converts it into a Restricted Message on Vertex Graph (RMOVG). A vertex in RMOVG represents one message in sequence diagram. According to Constraint Message Coverage (CMC) criteria, each message must be traversed at least once. Test cases generated from RMOVG satisfy CMC and result in reduced number of test cases.

Turner, Park, Kim and Chae have proposed an activity oriented technique [10] for automated test code generation. The naming convention used is “user-activity-object”. An activity B is dependent on an activity A if the application state of B requires the running of activity A. The developer begins by identifying activities in a web application, followed by developing a test activity graph depicting dependent and independent activities. Finally, an activity test algorithm is employed to generate test cases.

3. CLASSIFICATION OF TECHNIQUES

In this section, we classify the web application testing techniques based on the type of approach followed, viz. finite state machine based, object based, flow based and agent based approach.

3.1 Finite State Machine approach

Qian Zhongsheng has proposed a web testing technique based on probable FSM $Q = (S, \Sigma, \delta, PR, s_0, F)$ where S is a set of states (web pages), Σ denotes a set of transitions among pages, δ is a single valued transition function, PR is set of probabilities of transitions (execution frequency of various parts of application may vary), s_0 is the unique initial state and F is the set of final states. Test cases are generated by first calculating the independent paths from state transition diagram based on cyclomatic complexity, where the probability of whole path is the product of probabilities of individual transitions in that path. Possible test cases with probabilities above threshold can be generated to cover frequently occurring operations. PFSM is a functional testing technique, based on the concept of execution frequency of various parts of applications, which tests the usability feature, while ignoring the other features including object behavior, quality and browser interaction. It is best suited for applications where some web pages are more frequently traversed than others.

The other technique that follows the finite state machine approach is frameset –browser interaction technique by Song, Miao and Chen. Web framesets are employed in applications where layouts of some pages are identical. The behavior of browser has a critical impact on functioning of web applications, since an application may be functionally sound but may not work due to an irregular functioning browser. The process begins with modeling of application with framesets, followed by considering browser interactions along with framesets, modeling the web navigation, formalizing the navigation model, and finally generating and executing test cases. This technique is again a functional

testing technique, but with a different feature, behavior of the browser. As a concept of testing, it takes page navigation into account, rather than execution frequency used by the previous technique. This technique is applicable in web applications with web pages having identical layouts, and thus incorporating the use of framesets for designing.

3.2 Object based approach

Kung, Liu and Hsia have presented an object oriented approach for web testing, to capture structural and behavioral test artifacts of web applications. The entities are represented as objects, and their structures, relationships and dynamic behaviors are described. An object oriented Web Test Model (WTM) has been proposed which represents artifacts from object, behavior and structure features. For object perspective, entities are depicted by Object Relational Diagram (ORD). For behavior perspective; a Page Navigation Diagram (PND) describes the navigation behavior among pages. For structural perspective, flow graphs are used to describe control flow and data flow information of scripts or functions in web applications. Best suited for object oriented applications, it is a functional testing technique which tests the object, by taking object navigation, state and behavior as the concepts of testing. However it ignores the browser behavior and quality features.

Lei Xu and Baowen Xu introduced a framework for web testing beginning with requirement analysis based on object model, interactive relation model and architectural model, to test case generation via combinatorial method, followed by execution of test cases. Combinatorial method begins by identifying various parameters of testing, followed by comparison of default method and a pair-wise combination of test cases, proving that later is the more efficient technique to generate test cases for an application with a very high number of parameters of testing. This is an object based functional testing with object behavior as the concept of testing. This technique guarantees a reduced number of test cases for web applications having a very high number of parameters, as explained in the next section.

Turner, Park, Kim and Chae have proposed an activity oriented technique for automated test code generation. The naming convention used is “user-activity-object”. An activity B is dependent on an activity A if the application state of B requires the running of activity A. The developer begins by identifying activities in a web application, followed by developing a test activity graph depicting dependent and independent activities. Finally, an activity test algorithm is employed to generate test cases. Activity oriented is an object based, functional testing technique taking into account the dependent and independent web application activities, with usability as the feature of testing. It does not however support browser behavior and quality features. It can be well applied in applications that can be divided into a set of dependent and independent activities.

Li, Mio and Qian have proposed UML Based approach that models a large web application as hierarchical profile use case diagrams called Use Case Transition Model (UCTM). Traversing the UCTM from top to down, each use case is described as a sequence diagram, which automatically converts it into a Restricted Message on Vertex Graph (RMOVG). A vertex in RMOVG represents one message in sequence diagram. According to Constraint Message Coverage (CMC) criteria, each message must be traversed at least once. Test cases generated from RMOVG satisfy CMC and result in reduced number of test cases. UML based technique is a functional testing technique, using use cases and sequence diagrams as the concepts for testing. It lays stress on the feature of object behavior and message passing

among objects.

3.3 Flow based approach

Qian, Mia and Zeng have proposed a web testing model that constructs a Page Flow Diagram (PFD) of web pages and transitions followed by conversion into a Page Test Tree (PTT), a spanning tree, from which a test translator is employed to generate test cases, and execute them to generate test report. Evaluating this approach begins with constructing a Page flow diagram that represents links between web pages traversed by a specific user at a specific time. K_i represents links and S_j represents web pages, where i and j are the number of links and web pages, respectively. Using a PFD-to-PTT algorithm, a table depicting the parent child relationship among nodes is created, which results in the formation of the Page Test Tree to represent successive relationships among nodes. This is a functional testing technique which tests the usability of the application. The applications where the navigation among pages can be easily identified are the best candidates for this web testing technique. It does not however, take into account, the quality attributes, and object and browser behavioral features.

Guangzhou Jiang and Shujuan Jiang have presented a quick test model for performance testing, based on testing flow of web applications. It introduces a new performance index called successful request rate apart from other concepts like number of users and response time in order to test the performance of web application. It also contributes a testing method, combined with LoadRunner tool to provide effective solution to quick performance testing. The quick testing process begins with planning the test, followed by LoadRunner script creation, scenario definition, execution and result analysis. Quick testing model is a non functional testing technique suited for applications that may experience failure in case of heavy loads, example unexpected outputs for an excessively higher number of users. The technique focuses on quality features of testing (performance testing), and uses the performance index called successful request rate as the concept for testing, unlike the

previous flow based approach.

3.4 Agent based approach

Wang Hai-rui, LI Ya and LI Xin-jie's mobile agent based model for quality testing which provides a single framework for distributed applications. The mobile agent moves through heterogeneous networks and communicates with hosts and other agents. The approach begins with "Initialization" phase wherein web page of agent server is accessed, and the agent server assigns values to agent to perform automated testing on its own. The next step is to "Dispatch" the agent in cyber space. Then in "Execution" phase, the agent performs automated testing. In "Report" phase, the sever status is reported by agent to client. Best applicable on distributed applications, agent based technique is also a non functional testing, emphasizing on quality features, including load, reliability and security testing. However it ignores the browser behavior and object behavior features.

4. COMPARISON

The following table represents the comparison of various techniques based on the features tested, concepts used and type (functional/non functional) of testing. The features refer to the attributes of the web application being tested, viz. usability, object behavior, quality and browser behavior. The concepts are the parameters along which web applications are tested, i.e. execution frequency, use case, object behavior, mobile agent, activities, and page navigation and performance indices. These techniques have been compared based on functional and non functional testing. While functional testing verifies that each function of the software application operates in conformance with the requirement specification [11], non functional testing demonstrates how well the product behaves [12]. Example, the mobile agent based web testing technique is a non functional testing technique that uses the concept of mobile agents to test the quality features including reliability and security in web applications.

Table 1: Comparison of Web Testing Techniques

		FEATURE TESTED				CONCEPT USED							FUNC/NON FUNC	
		USABILITY	OBJECT	QUALITY	BROWS ER	EXECUTI ON FREQ.	USE CASE	OBJECT	AGENT	ACTIVITY	PAGE NAVIG ATION	PERF INDEX	NON FUNC.	FUNC.
FSM APPROACH	FRAMESET- BROWSER				×						×			×
	PFSM	×				×								×
OBJECT BASED APPROACH	OBJECT ORIENTED		×					×						×
	COMBINA- TORIAL		×					×						×
	ACTIVITY ORIENTED	×								×				×
	UML BASED		×				×							×
FLOW BASED APPROACH	FLOW DIAGRAM	×									×			×
	QUICK TESTING			×								×	×	
AGENT BASED APPROACH	MOBILE AGENT			×				×				×		

5. CONCLUSION

The following table elaborates the benefits and limitations of the techniques discussed:

Table 2: Benefits and limitations

Technique	Benefits	Limitations
Probable FSM	The concept of probability of transitions results in reduced test cases	No general formula for frequency and threshold calculation
UML	Reduced test cases compared to conventional approaches	Requires partitioning in case of large web applications, Restricted to branch testing
Combinatorial	Significant reduction in test cases for large applications with large no. of parameters	Complexity and cost not justified for small applications
Mobile Agent	Reduced network traffic, balanced workload, high robustness	A proper Server-Agent coordination model required
Object Oriented	Every object has its own test methods and utilities, Testers can identify change ripple effects	Time consuming, complex process, requiring multiple models for testing in each perspective
Quick Testing	Practical, effective, accurate, complete, automated	Expensive license for LoadRunner, Configuration/installation issues
Frameset – Browser	Considers browser interactions that significantly affect application behavior	Redundancy in test cases, Framesets intricate the process of testing
Flow Diagram	All-pages-coverage guarantee every page tested at least once	Requires manual effort for construction of PFD
Activity Oriented	Few edges in activity graph lead to reduced test cases	Requires manual effort for identification of activities

Web applications develop more rapidly as compared to traditional applications and their multi-platform, dynamic and heterogeneous nature makes it difficult to test them through traditional testing methodologies. Each web testing technique discussed in the survey has its own criteria, target, feature, concept and methodology for testing. As explained in the last section, every technique is characterized with its own benefits and limitations, but none of them can be eliminated as they all find use depending on the characteristics of web applications.

6. FUTURE SCOPE

The comparative survey performed above may serve as a base for the future generation of web application testing engineers to focus their interest on specific testing techniques based on the nature of web applications.

7. REFERENCES

- [1]Rajiv Chopra, “Software Testing”, http://books.google.co.in/books/about/Software_Testing_2nd_Edition.html?id=uTJY3WQLLc-kC
- [2] Qian Zhongseng, “An approach to testing web applications based on probable FSM”, International Forum on Information Technology and Applications, 2009
- [3] Zhongsheng Qian, Huaikou Miao , Hongwei Zeng.” A Practical Web Testing Model for Web Application Testing”, Third International IEEE Conference On Signal-Image Technologies And Internet Based System
- [4] Lei Xu, Baowen Xu,” A Framework for Web Application Testing”, International Conference on Cyberworlds, 2004
- [5]WANG Hai-ru, LI Ya, LI Xin-jie,” A Mobile Agent Based General Model for Web Testing”, Second International Symposium on Computational Intelligence and Design,2009
- [6] David C.Kung, Chien-Hung Liu, Pei Hsia,”An Object Oriented Web Test Model for Testing Web Applications”, the 24th Annual International Computer Software and Applications Conference, 2000
- [7]Guangzhu Jiang, Shujaun Jiang,” A Quick Test Model of Web Performance Based on Testing Flow and its Application”, Sixth Web Information Systems and Applications Conference, 2009
- [8]Bo Song, Huaikou Miao, Shengbo Chen,” Considering Web Frameset and Browser Interactions in Modelling and Testing of Web Applications”, International Conference on Computational Intelligence and Software Engineering, 2000
- [9]Liping Li, Huaikou Miao, Zhongsheng Qian,” A UML-Based Approach to Testing Web Applications”, International Symposium on Computer Science and Computational Technology, 2008
- [10] David A Turner, Moonju Park, Jaehwan Kim, Jinseok Chae,” An Automated Test Code generation Method for Web Applications using Activity Oriented Approach”, International Conference on Automated Software Engineering, 2008
- [11] <http://www.guru99.com/functional-testing.html>
- [12]<http://www.slideshare.net/nworah/non-funtional-testing>