International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 1, January 2014

Model Based Testing in Web Applications

Nazish Rafique[1], Nadia Rashid[2], Saba Awan[3], Zainab Nayyar[4]

[1, 2, 3, 4]Department of Computer Engineering, College of Electrical and Mechanical Engineering, Rawalpindi, Pakistan

Abstract: ***Web applications have become an important aspect of life. It is used to support many activities like online information sharing, for business transactions and many other related things. Due to their high usage, they require high performance to be guaranteed. Therefore more emphasis should be done while testing web based applications. For this, model based testing has been introduced to find errors in web applications. In this paper, the basic models which are used in model based testing are discussed. Further, how these models can be used for testing the web based applications and a case study will be explained with the help of case study.***

Keywords: Web applications, testing, model based testing, finite state machines.

1. Introduction

Testing is the major phase while developing software because it makes the software much better by finding errors and make several improvements in the system. Testing web applications is a very important process as they are the fastest growing area in software engineering which provides several activities such as many business transactions, academic studies etc. Web applications are based on multi tier architecture so they require to be tested on each and every phase but in an automated manner to get the accurate results [1].

Model based testing is an automated testing which describes the behavior of the software in the form of state and transitions. The interface of the software shows the states of the software system and the menus, buttons and commands which changes or enhances the functionality of the software is the transition or change. In model based testing, test cases are generated on the basis of the model constructed from the application. This method can be used to check the access provided to multiple users as well because in web applications and java based interfaces there are multiple machines involved in operating these applications and multiple users perform parallel actions at a time on the single application from various machines and from several locations their might be one or more server is involved for providing services to the clients [2].

There are two modes of execution in model based testing: online execution and offline execution. In offline execution, test cases are generated and comparisons are made on the basis of the actual response and the expected response of the application and the results are recorded. An appropriate number of test cases are written and they are executed later. In online mode, the test cases are generated by continuously interacting with the application under test. In this, inputs are given, responses are being checked and the attributes given by the application under test are compared with the expected results of the model. Test cases are generated and executed at the same time. Web applications are usually based on sequential events. Few years back the web testers tests the web applications manually which involves the regression testing which was not only time consuming but results were inaccurate model based testing provides an automated way for web testing and thus obtained accurate results. Web testing is also performed using event flow graphs in which the every event act as a parent node for the next events generated from the parent event [3].

Model based testing is a black box testing in which code is not taken under consideration but only the functionality of the application is considered. Model based testing combine dumb monkey testing in which the tester writes the script which generates the application with random inputs and static testing which generate scripts that give only required inputs [4]. The major single drawback of model based testing is that it is a costly method [5]. Web applications are usually heterogeneous in nature and their major features are interoperability, fault tolerance, scalability, reliability and transparency they mostly support the distributed environment [6]. The remaining paper organizes as follows section 2 is related work, section 3 explanation of model based testing techniques, 4 example of model based testing on web applications a mini case study scenario, section 5 is conclusion and section 6 is references.

2. Related Work

Model based testing is used to generate test cases after modeling the system. These test cases describe the behavior of the system. Model is developed after gathering the information from requirements or specification document. This model does not require including all the functionality of the system rather it can be developed by using functional requirements. It can be constructed at any abstract level. Test cases are then generated from the model of the system. In test suite, inputs are given the actual data which of the system for the purpose of testing the system outputs. Then compare actual outputs with the desired outputs of the system. On this basis of which, the system is evaluated by verifying its conformance with the requirements, model and code. Through this technique, we can find the errors present in the system. Decisions are made whether to modify the existing model, or need to generate more test cases, reliability estimation [7]. Figure 1 explains the model based testing approach in detail.

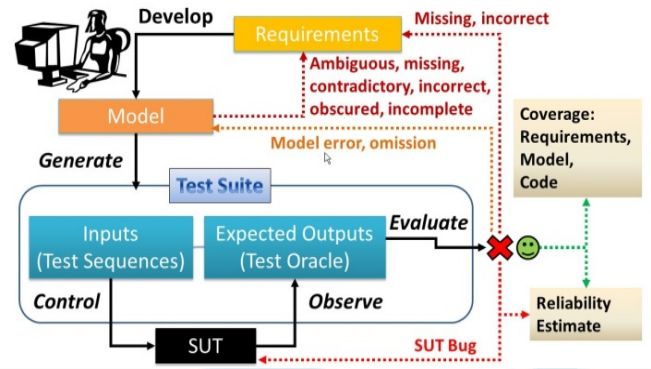

Figure 1: Model Based Testing Approach

Kung, Liu, and Hsia in [8] proposed model based generation techniques. Object Relation Diagrams, Object State Diagram, a Script Diagram and a Page Navigation Diagram models were used in that technique. Basically in that technique the authors used source code for the model generation and testing them. Main drawback in their technique was that they used source code for testing for web applications and this was not helpful in testing whole system. This cannot be applied on different types of web applications [9]. It was confined to a specific web based application. However, the technique discussed in this paper does not use source code for testing.

Lee and Offutt in [10] proposed an approach in which generation of test cases were based on mutation analysis. In this approach, test cases were only based on module interactions with each other that are implemented on XML.

In [11], Liu proposed a tool known as WebTestModel, in this technique web application components were considered as objects. For this, they generated test cases based on the data flow among these objects.

In [12], the authors have proposed Reweb and Test web tools. These tools were applied on source code for the purpose of testing. To represent components in web applications and to show their interaction, UML class diagrams were used in this approach. To construct the model of the system, Reweb tool was used. It was consisted of two phases: 1) to model the static and dynamic characteristics. Second phase worked on forms and the pages linked with these forms. The tester manually selected the test inputs which included the data used to fill the forms. The main drawback in this approach was that it models the complex part of the system manually and automation was done only for simple Parts. After this, Test Web tool used the extracted model from the Reweb Tool and then using source code generates test cases.

In [13], Finite state machines model have been proposed by Andrew, Alexander and Offutt. Hierarchical FSM were used to characterize the structure of web based applications. Each page of the web application represents a node in the finite state machines. Test cases were generated to represent sequence of states having some parameters which were required by states in the finite state machine. Low level test cases were combined together to form high level test cases.

In [14], a technique was proposed to investigate the source code and then generate test cases by using this information. The purpose of investigating the source code was to obtain information about interfaces which included: parameters for input, information related to domain, and map related to user navigation actions. The main drawback of this technique was that for generating test cases, analysis of source code is required. This may limit us to use it for specific programming languages only and does not support modern technologies.

3. Model Based Testing Techniques

3.1 Finite State Machines

Finite State Machine is a model used in model based testing for web applications. The basic purpose of using this model is to test the behavior of web based applications. This model contains states and transactions among them. Actions are performed to change the state. There are finite set of states that represents the system under testing. The model is represented in the form of graph where nodes represent the page of the web applications and transactions are represented by edges which shows page navigation [14]. We can represent the model of a web application by using finite state machine model. For this, there are some rules which we have to follow:

- States shown in the model of the application should be associated with one of the pages of the application
- Sub-state represents the individual tab of each page.
- State of change can take place when actions are performed on any web page.
- Transition in the finite state machine can only take place when some action is performed on the pages of web application [14].

In figure 2 an example of finite state machines is shown in which the home page transits to services and about.

Figure 2: State and Actions Representation in FSM

3.2 Event Flow Graph Model

The event flow model consists of two phase. In the first phase, encoding of each event is done in the form of

preconditions. i.e.: the current state of the event in which it executes and the effects on it after the change of state occurred. Second phase represents the order of events which are executed on web application page and is represented as a set of directed graph. For web applications these two phases are considered to be compulsory for testing. For test cases generation, these preconditions and effects of the states can be used [3].

3.3 Unified Modeling Language Class Diagram:

Class diagram contains classes and relationship of these classes with each other. Relationships are of three different types; generalization, association and the last one is aggregation. Domain problems are determined by representing classes, associations will show the relationships among these problems. Generalization describes the problem from bottom up/top down techniques. Aggregation is much more similar to association. The test cases generated for the web based application through UML class diagram cover the following criteria: [15]

- Association-end multiplicity.
- Generalization.
- Class attributes [15].

4. Example of Model Based Testing on Web Applications a Mini Case Study Scenario

In this section model based testing is explained with the help of an example of an online shopping system by considering its home screen. Figure below shows a home screen of online shopping system to illustrate the application.

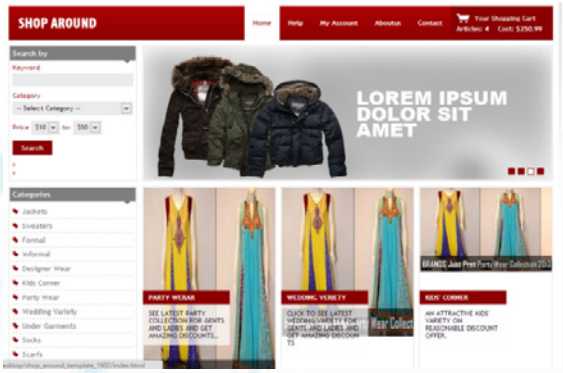

Figure 3: Screen Shot Of Online Shopping System

Each new icon of the page moves to the new page. Each page represents a state and the menu icons of the page shows the transitions. In the table below these transitions are represented.

Table 1: Tabs and Menus of Home Screen

From State	Transition	Target State
Home Screen	News Tab	News Screen
Home Screen	Images Tab	Images Screen
Home Screen	Home Tab	Home Screen
Images Screen	Home Tab	Home Screen
Images Screen	News Tab	News Screen
Images Screen	Images Tab	Images Screen
News Screen	Home Tab	Home Screen
News Screen	Images Tab	Images Screen
News Screen	News Tab	News Screen

Figure 4 shows the UML class diagram [15] of online shopping system. In which there are classes which have a generalization, composition and aggregation [15] relationship. Customer and administrator classes are inherited from user class. The user class has user id, password and login status having string as data types and a procedure verify Login () of type bool. Aggregation is present between shopping cart and product as shopping cart contains the products which are ordered by the customers. Composition is present between shopping cart, customer, orders, and shipping Info, Order Detail, Category and product classes.

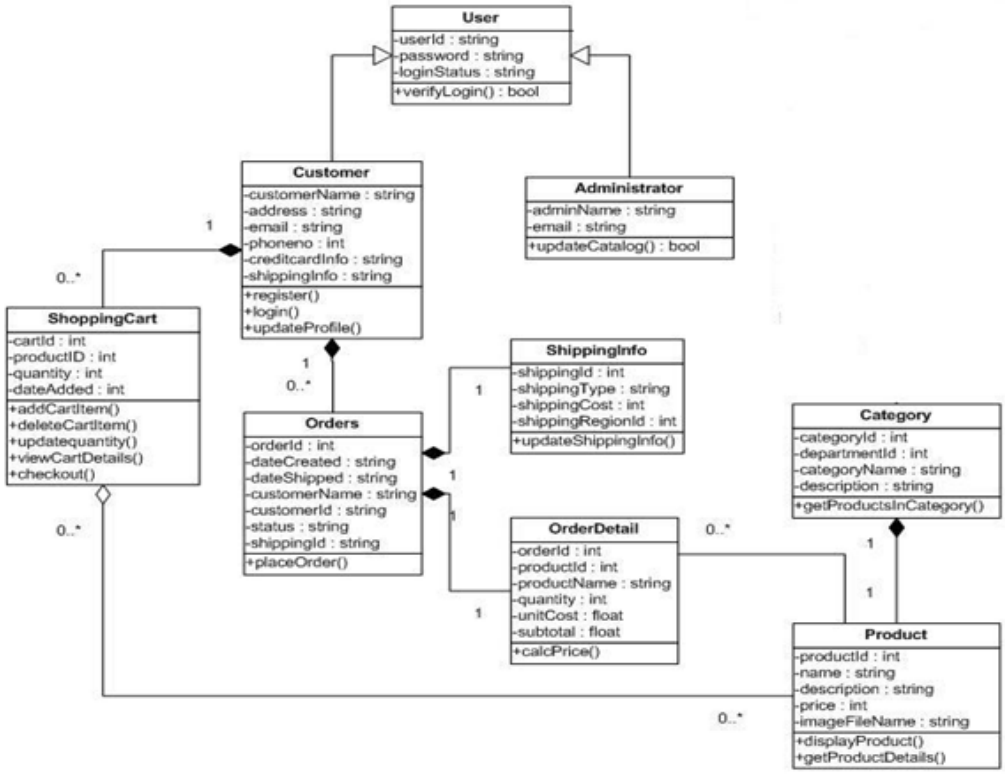

Figure 4: Class diagram of online shopping system

The name of the state diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system. A State chart diagram describes a state machine. Now to clarify it state machine can be defined as a machine, which defines different states of an object, and these states are controlled by external or internal events [14]. In the state machine diagram below the states and transitions which occur in the online shopping system are shown in which the boxes represents the states and arrows represents the transitions occur in the system.

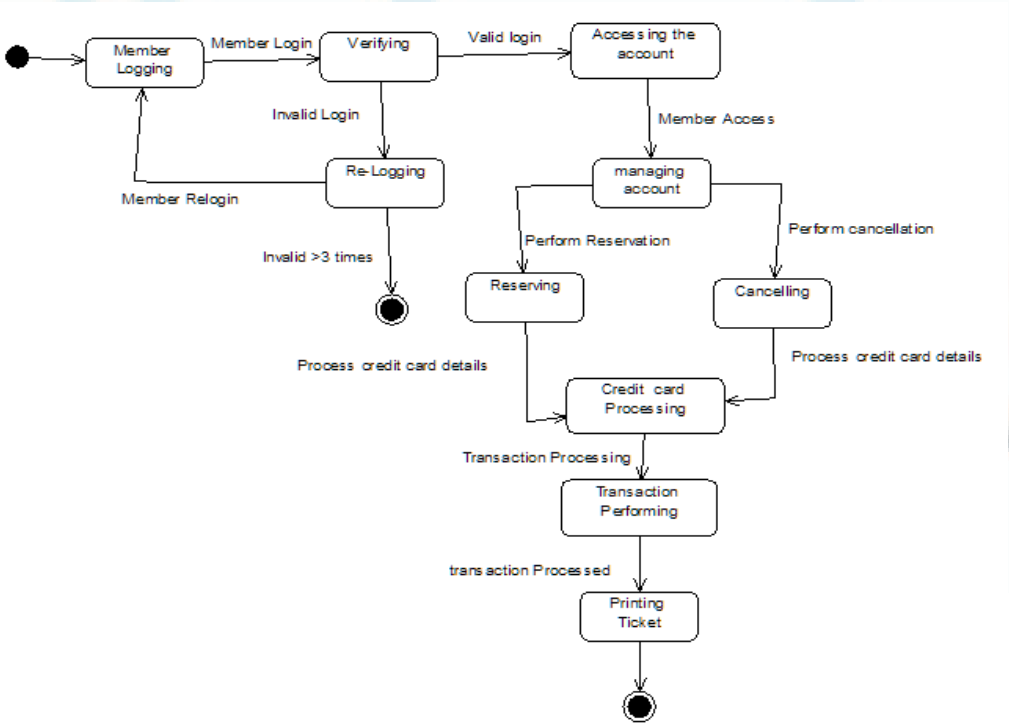

Figure 5: State machine diagram for Online Shopping System

The use-case concept was introduced by Ivar Jacobson in the object-oriented software engineering method. A use-case diagram is a graph of actors, a set of use cases enclosed by a system boundary, communication (participation) associations between the actors and the use cases, and generalization among the cases. Use case diagrams show how users interact with the system. Use case diagrams describe what a system does from the standpoint of an external observer. The emphasis is on *what* a system does rather than *how*. Use case diagrams are closely connected to scenarios. This is the reason due to which use case diagrams are considered as a part of model based testing because along with the interaction of users with the system they also shows the different aspects of the systems [16].

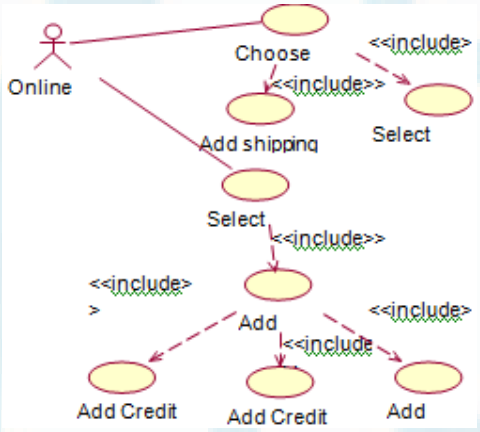

Figure 6: use case diagram of online shopping system

5. Conclusion

In this paper, we have discussed the model based testing in web applications. Different techniques have been described to model testing in web based applications. This technique of testing provides an easy way to find errors in these applications. A model can be constructed from requirements document which includes only few functional requirements. Test cases are generated and system conformance can be checked with the system model. In the end model based testing is also explained with the help of a case study.

References

[1] Arora A, Sinha M. Web Application Testing: A Review on Techniques, Tools and State of Art. International Journal of Scientific & Engineering Research, Volume 3, 2012.
[2] Jeffrey Feldstein. Model-Based Testing for Java and Web applications. 2006.
[3] Oluwaseun Akinmade & Prof. Atif M. Memon. Automated Model-Based Testing Of Web Applications. 2008.
[4] Christoph Neu. An introduction to Model Based Testing.
[5] HamidehHajiabadi, Mohsen Kahani. An Automated Model Based Approach to Test Web Application Using Ontology. 2010.
[6] Cyntrica Eaton and Atif M. Memon. Advances in Web Testing. In *Advances in Computers*, vol. 75, pages 281-306 2009.
[7] Hani Achkar. Model Based Testing of Web Applications. STANZ, pages 1-28, 2010.
[8] Kung, C. H. Liu, and P. Hsia. A model-based approach for testing Web applications. Twelfth International Conference on Software Engineering and Knowledge Engineering, 2000.
[9] D. Kung, C. H. Liu, and P. Hsia. An object-oriented Web test model for testing Web applications. IEEE 24th Annual International Computer Software and Applications Conference, pages 537-542, 2000.
[10] S. C. Lee, J. Offutt. Generating test cases for XML based Web component interactions using mutation analysis. In Proceedings of the 12th International Symposium on Software Reliability Engineering, pages 200-209, 2001.
[11] C. Liu, D. Kung, P. Hsia, and C. Hsu. Structural testing of web applications. 11th IEEE International Symposium on Software Reliability Engineering, pages 84--96, 2000.
[12] F. Ricca and P. Tonella. Analysis and testing of Web applications. 23rd International Conference on Software Engineering, pages 25-34, 2001.
[13] Andrews, J. Offutt, R. T. Alexander. Testing Web Applications by Modeling with FSMs. Software and Systems Modeling, pages: 326-345, 2005.
[14] AynurAbdurazik and Jeff Offutt. Generating Test Cases from UML Specifications. 1999.
[15] M. Wang, J. Yuan, H. Miao, G. Tan. A Static Analysis Approach for Automatic Generating Test Cases for Web Applications. International Conference on Computer Science and Software Engineering, 2008.
[16] Li Ye. Model Based Testing Approach for Web Applications. 2007.

Author Profile

Nazish Rafique has done BS (Software Engineering) from APCOMS, Rawalpindi, Pakistan in 2013. Currently she is doing MS (SE) from EME College, NUST, Pakistan.

Nadia Rashid has done bachelors in software engineering from Amy Public College of Management and Sciences, Rawalpindi, Pakistan in 2013.Now doing MS in software engineering from College of Electrical and Mechanical Engineering, NUST, Rawalpindi, Pakistan.

Saba Awan has done bachelors in software engineering from Amy Public College of Management and Sciences, Rawalpindi, Pakistan in 2013.Now doing MS in software engineering from College of Electrical and Mechanical Engineering, NUST, Rawalpindi, Pakistan.

Zainab Nayyar has done BS (Software Engineering) from APCOMS, Rawalpindi, Pakistan. Currently she is doing MS (SE) from EME College, NUST, Pakistan.