Test Case Reduction Approach in User Session Based Testing for Web Application

Hsu Mon Maung
University of Computer Studies, Mandalay
hsumon77@gmail.com

Abstract

Web application testing is crucial to provide reliable services for the fast growing demand on web application. Testing, designing and generating test cases are challenging task because web application is complex and changeable. Among test case generation, user session based test case generation has been researched to generate test cases by the use of user session data. In user-session based testing, a tester captures user accesses during deployment to create user session, which are then replayed as test cases. This paper describes the test case reduction approach for analyzing, and replaying the large number of test cases generated from user session data. Entropy gain theory is used to generate user session based test cases. Moreover, this system combines the user session data with the internal structure analysis of the application for ensuring the reliability of web application. The rate of fault detection of the test suite is measured to evaluate the effectiveness of proposed system.

Keywords*:* user session based testing, entropy measure, web application testing

1. Introduction

Web applications have been dramatically increased and most daily activities rely on the services provided by them. The qualities of these applications are central role because it may be great impact on daily activities. Efficient and effective testing of web application is crucial for reliable services. Furthermore, the nature of web applications are frequently changed and upgraded due to security attacks and user preference changes. Web application testing is to uncover the content errors, navigation error, and compatibility issues etc. Testing must be performed completely in time without service interruption. User session data has been recently used to create test cases. For web application system, field data has the additional advantage because the usage data is independent of the underlying implementation and server technologies [4]. User session based testing is an automated approach to enhance an initial test suite with real user data. The logged user sessions are collected as a set of use case which is behaviorally related events accessed by user through the system. In user session based testing, test suite reduction is main considerable point of executing the test sequences that are not only smaller in size but also equivalence in effectiveness to an original test suites. A major problem with user session based testing is the cost of collecting, analyzing, and replaying the large number of test cases generated from user session data [7]. Automated cost-effective test strategies are needed to provide reliable, secure and usable web application. Given a set of user sessions, there are techniques of producing test data for web applications [2, 3]. The main points of user session based testing are the selection and reduction methods of test case suites. This paper describes the test case reduction approach by applying entropy gain theory and analyzing the internal structure of web application. This approach requires white box analysis phase that determine the structure of web application. The dependent pages are extracted from web application structure and sorted by time stamp from log data. The highest entropy gain of user session is selected as a test case and then dependent links that are accessed by the user are considered as a test case to cover the overall web application structure. The related work is described in section 2. In section 3, this paper describes the web applications, user session based testing and test case generation. Section 4 discusses the proposed system and the metric for evaluating the testing effectiveness. The paper is concluded in section 5.

2. Related Work

Elbaum et al. [2] studied several techniques for using user session data gathered as users operate Web applications to help test those applications from a functional standpoint. The studies showed user session data can be used to produce test suites more effective overall than those produced by the white-box techniques considered. The authors also discussed additional issues relevant to the use of user-session data in testing Web applications.

Sampath et al. [8] explored the possibility of using concept analysis for achieving reduction and scalability in user session based testing of web applications. The method only considers the base request. The studies showed the low coverage of the base requirement, including statement coverage, fault coverage and base request coverage. The authors also admitted the importance of request data and ordering. But the concept analysis cannot take the factors into consideration.

Another method is generating test cases using a model of user sessions that requires less space than the original user sessions [10]. The studies explored a method of automatically estimating these dependencies and using them to arrange the test suite. The authors depicted some limitations of an approach to automatically testing Web applications and introduce some ideas for improving upon it.

Ebrahim Shamsoddin-Motlagh reported a survey of recent research to generate test case automatically. Those are presented from UML based, graph based, formal methods, web application, web service, and combined [11].

3. Background Theory

3.1. Web Application

A web application consists of a set of Web pages and components that form a system that executes using Web server(s), network, HTTP, and browser(s) in which user input (navigation and data input) affects the state of the system. A Web page can be either static—in which case, the content is the same for all users—or dynamic such that its content may depend on user input. Web applications may include an integration of numerous technologies, third-party reusable modules, a well-defined layered architecture, dynamically generated pages with dynamic content, and extensions to an application framework. Figure 1 shows how a simple Web application operates [2]. Large Web-based software systems can require thousands to millions of lines of code, contain many interactions between objects, and involve significant interaction with users [8].

3.2. User Session based Testing

User session based testing makes use of field data to create test case, which has the great potential to effectively generate test case that can effectively detect residual faults [5]. A user session based test case is a sequence of HTTP request (GET or POST) containing base requests and name value pairs that are recorded when a user accesses the application [6].

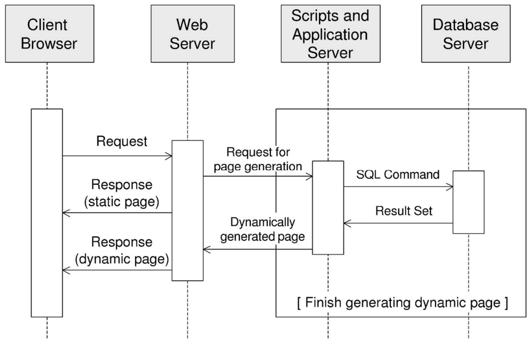
Client Browser
Web Server
Scripts and Application Server
Database Server
Request
Request for page generation
SQL Command
Response (static page)
Result Set
Dynamically generated page
Response (dynamic page)
[Finish generating dynamic page]

Figure 1. Sequence Diagram of Web Application

The web server records each interaction with user in log files. The sample log data of web server looks like the followings:

```
192.168.30.15--
[18/Feb/2009:16:17:41−0500]“GET/apps/onlinelibrar
y/Login.jsp    username=hsu&password=hsu305105”
“HTTP/1.1” 200 66468 “Mozila/4.0 (compatible;
MSIE 6.0; Windows  NT 5.1)”
```

In this sample log, the base request is Login.jsp and parameter value pairs are username=hsu and password=hsu30510. The example test case is shown in Table 1.

Table 1. Example Test Case

Testcase1
Def.jsp
Login.jsp&name=hsu&password=hsu305105
Sbook.jsp&bookid=005
Vres.jsp
Logout.jsp

When a request from a new IP address arrives at the server, a user session is identified as initial and when the user leaves or session time out, the user session is identified as the end. The 30 minutes is taken to identify the user session. The test case reduction techniques are based on different criterions, such as covering all base requests in user session data. The reduction method can be not only smaller test suite size than original suite but also fault finding effectiveness.

3.3. Test Case Generation

A test case is defined as a set of inputs, execution conditions, and expected results developed for a particular purpose. A test suite is a related collection of test cases. A test strategy is an algorithm or heuristic used to create test cases from a representation, an implementation, or a test model [1]. Various test case generation strategies have been proposed to generate test cases from static models of web applications. Test suite generation consists of two costs: parsing the server log and generating suites [4].

In user session based testing, the number of access logs can be gigabytes in a day and the logs are redundant because users access the application in the same way. A large number of user session data may be available from various web application systems. Given the collected URL and name value pairs, there are many ways in which test cases could be generated [2]. We need to reduce the size of test suites while maintaining their effectiveness. In this proposed system, the entropy based link analysis within single web site is applied to analyze the base request of user session data. Entropy values of links that are accessed by users are analyzed to reduce test cases for web application. Shanon's information entropy E can be expressed as [9]:

$$E(H) = -\sum_{i=0}^{n} P_i \log P_i \qquad (1)$$

where P_i is probability of $link_i$ that are accessed by users and n is the number of links of web site. Entropy values of link are derived from base request data in the relation table.

4. Proposed System

The proposed test case generation approach is based on the entropy concept for measuring the user session data of web application.

4.1. Subject Application

In the proposed system, the log file data of Apache Tomcat server 5.5 is used to generate the test cases. The application, OnLineLibrary system is used on web application testing. The OnLineLibrary functionalities are divided into two types: user activities and admin activities. This system provides the ability of register, login, viewcatalog, viewreservedlist, searchbook etc. to the users. It uses JSP for its front-end and a MySQL database backend. Our proposed system concentrates on functionalities that are accessible to the users and not the administration function.

4.2. Proposed System Design

In this section, the steps of proposed system that generate the test cases are described. The overview frame work of proposed system is shown in Figure 2. In preprocessing step, the structure of web application is first analyzed to extract the dependency pages. A dependent graph depicts the data and link dependent relationship between the web pages in web application. Figure 3 shows neatly the relationship between pages in the OnLineLibrary application.

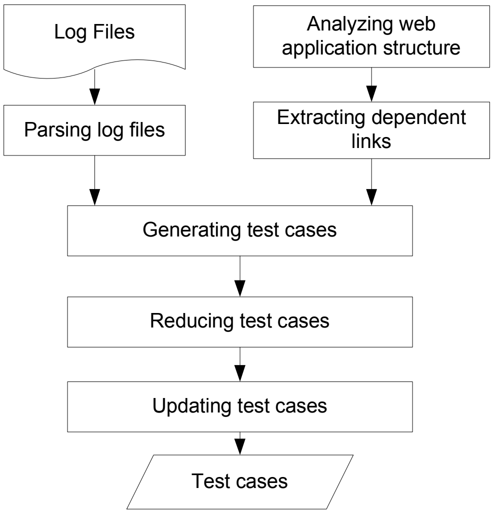

Figure 2. Framework of Proposed System

In step 1, log files are collected to record user interaction with web application. The originating IP-address, time stamp, the request, cookie information and GET and POST data in each user request are collected. The unwanted view of data in the log files is removed.

In step 2, the access logs are parsed according to their field respectively. After parsing the web server's access logs, the output of parser is further edited to create the binary relation table as shown in Table 2.

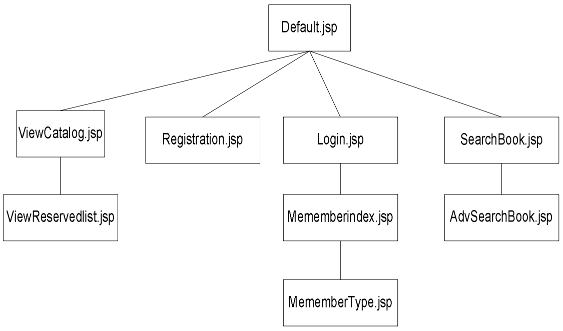

Figure 3. Example of Partial OnLineLibrary System

In step 3, the test cases are generated by applying the entropy gain value.

In step 4, there are two steps to reduce test cases. Firstly, maximum entropy gain of user from user session that is covered for most or all URLs in web application are selected. According to Table 2, the entropy gain of user3 is highest. Thus, the user 3 is selected as a test case. Secondly, the selected user session as a test case is needed to check whether it covers all dependent relationships among web pages or not. If the test case doesn't cover the dependency pages, the next user that has second highest entropy value needs to select for whole web application. There are two users that have same entropy value (U5 and U6) in Table 2. In this case, we need to consider the time of base request that is accessed by user.

Table 2. Sample Relation Table

user	Def.jsp	Vcat.jsp	Vres.jsp	Sbook.jsp	Reg.jsp	Login.jsp
U1	1	1	1	0	0	0
U2	1	1	0	1	0	0
U3	1	1	1	1	0	1
U4	1	1	1	0	0	0
U5	1	1	1	0	1	0
U6	1	1	0	1	0	1

For example, if U5 is selected as a test case, we will lose the events of U6 according to the time stamp in Table 3. Thus, the dependent pages that are accessed by users are sorted by ascending order as shown in Table 3.

Table 3. Time Order of Based Requests

User	Time Stamp	Base Request
U1	00:30	Login.jsp
U2	00:32	Login.jsp
U3	00:36	Login.jsp
U4	00:40	Login.jsp
U5	00:41	Login.jsp
U6	00:46	Login.jsp

Example test cases are:

U3 – Def.jsp – Vcat.jsp – Vres.jsp – Sbook.jsp – Login.jsp

U6 – Def.jsp – Vcat.jsp – Sbook.jsp – Login.jsp

U3 U U6 covers the all base requests of web application structure.

In step 5, if new user session arrives, compare the entropy gain and time of accessing dependent pages with the existing test to create the update test suites.

In this proposed system, the large percent reduction in test suite sizes may lead to reduce the replay time of user sessions and oracle comparison time. The fault detection rate of reduced test suite can be achieved in high percentage.

4.3. Evaluate Generated Test Cases

The performance of generated test case is evaluated with respect to the detection of faults. Both original and reduced test suites are replayed through the original version (fault-free) of application and the expected results are recorded. Then different type of faults which are database faults, GUI faults and navigation faults are seeded into the copies of OnLineLibrary system. The original and reduced test suites were re-run through the fault seeded version and the actual results are computed. The expected output and the actual output are compared by utilizing oracle comparator. A number of detected faults and faults detected by the suites were recorded for every user session. The effectiveness of fault detection rate is evaluated by using the following equation.

$$FDR = \frac{faults\ detected\ in\ testing}{total\ number\ of\ faults} \qquad (2)$$

where *FDR* is fault detection rate of test cases.

5. Conclusion

The user session based testing has an advantage that actual user interaction with a web application can be recorded and used for testing the web application. Many user session based test reduction techniques have been quantified for web application testing. In this paper, the new hybrid approach is proposed in that it combines the user session data and structure analysis of web site to generate the reasonable test suite that covers as many faults as possible. This system uses the base request, the name value pair and the ordering of time stamp of base request accessed by the users. In our approach, the structure of web application is first analyzed to extract the dependent pages and to sort the time of request that are based on the dependency pages from the structural analysis. Then, entropy gain theory is applied to generate test cases by analyzing the gain of different user session data. Finally, in order to check the effectiveness of generated test cases, fault detection rate is theoretically evaluated. We have not yet fully compared our approach to current user session based testing techniques. In the future, the abilities of detection seeded faults will be evaluated compare with other test case reduction approaches.

References

[1] R. Binder "Testing Object-Oriented Systems", Addison Wesley, 2000.

[2] Sebastian Elbaum, Srikanth Karee and Marc Fisher II "Leveraging User-Session Data to Support Web Application", IEEE TRANSATIONS ON SOFTWARE ENGINEERING, VOL. 31, NO.3, MARCH 2005.

[3] G. A. DiLucca and A. R.Fasolino, "Testing Web-based applications. The state of the art and future trends" in Information and Software Technology, 2006, pages 1172-1186.

[4] Sara Sprenkle, Emily Gibson, Sreedevi Sampth and Lori Pollock, "A case study of automatically creating test suites from web application field data", Portland, Maine, USA, TAVWEB06 (Jul. 2006).

[5] T. Deenadayalan, V.Kavitha, S.Rajaraeswari, "Examing Web Application by Clumping and Orienting User Session", Journal of Computing Volume 2, Issue 6, 2010.

[6] Sreedevi Sampth and Renee C. Bryce, "Prioritizating User Session based Test Cases for Web application".

[7] Sara Sprenkle, Emily Gibson, Sreedevi Sampth and Lori Pollock "An Empirical Comparison of Test Suite Reduction Techniques for User Session based Testing of Web Application". In ICSM05: Proceeding of the 21st IEEE Int. Conf. Softw. Mainten. Budapest. Hungary, 2005, pp. 587-596.

[8] Sreedevi Sampth, Valentin Mihaylov, Amie Souter, and Lori Pollock "A Scalable Approach to User Session based Testing of Web Applications through Concept Analysis". In: ASE'04: Proc.19th Int. Conf. Automated Sofw. Eng. Washington DC, USA, 2006, pp. 132-141.

[9] Thomas M. Cover, Joy A. Thomas, "Elements of Information Theory", Print ISBN 0-471-06259-6 Online ISBN 0-471-20061-1.

[10] Anna Pobletts, Camille Cobb, Lucy Simko, "Toward an Effective Data Model and User Session Dependency Model", Conf, 2011.

[11] Ebrahim Shamsoddin-Motlagh, "A Review of Automatic Test Cases Generation", International Journal of Computer Applications (0975 – 8887) Volume 57– No.13, November 2012.