# An Efficient Test Cases Reduction Approach in User Session Based Testing

Hsu Mon Maung and Kaythi Win

***Abstract*—Web application testing has been used in finding various faults in order to improve the quality of reliable web services. Among test cases generation approaches, user session based testing is an approach to create test cases with real user data. However, real user data usage is extremely large and executing all the test cases can be time consuming in practice. This paper describes the test cases reduction approach for analyzing and replaying the large number of test cases generated from user session data. The structural analysis of web application with the user session data is used to ensure test results of web services. The entropy gain theory is applied in test cases reduction of proposed system to get the best test cases that cover all user accesses of web application. Executing all the tests in a reduced test suite can still be time consuming in practice. Therefore, test cases prioritization is proposed to reorder the reduced test cases with the goal of improving the rate of fault detection. The effectiveness of test cases reduction approach is evaluated by fault detection rate.**

***Index Terms*—Web application testing, user session based testing, test cases reduction, entropy measure.**

## I. Introduction

As the web application usage has been dramatically increased and most daily activities rely on the services provided by them, the qualities of these applications are central role. Efficient and effective testing of web application is crucial for reliable services. Testing must be performed completely in time without service interruption. Web application testing is a very expensive process in terms of time and resources due to the nature of web application. Testing, designing and generating test cases are challenging tasks because web application is complex and changeable.

User session based test cases generation has been recently researched to generate test cases by the use of user session data. For web application system, field data has the additional advantage because the usage data is independent of the underlying implementation and server technologies [1]. 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 according to their behavior of the related events accessed by user through the system. In user session based testing, a tester captures user accesses during deployment to create user session which are then replayed as test cases. 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 [2]. The main considerable points of user session based testing are the selection and reduction methods of test suites because the collected daily user logs are million gigabytes that are impossible to replay as test cases. 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 determines the structure of web application. The reduced test suites are able to execute the test sequences that are not only small in size but also equivalent in effectiveness to an original test suite. In order to improve the rate of fault detection, test suite prioritization modifies the order of test cases. In this paper, frequency and time based test cases prioritization criteria is proposed in which the test cases are ordered based on occurrence and time of base request sequences.

In the remainder of this paper, the related work is described in section II. Section III discusses the user session based testing and test cases reduction. The proposed system and a metric for evaluating the effectiveness of proposed system are described in Section IV. The paper is concluded in Section V.

## II. Related Work

In this section, several researches related with testing web application are described.

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 [3].

S. Sampath and R. Bryce presented ordering the tests in a reduced suite to increase its rate of fault detection [4].There are several approaches to order reduced test suites using prioritization criteria for the domain of web application.

Elbaum *et al.* [5] 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.* [6] explored the possibility of using concept analysis for achieving reduction and scalability in user session based testing of web applications. 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.

The studies [7] explored a method of estimating dependencies automatically and using them to arrange the

Manuscript received May 5, 2014; revised July 7, 2014.

Hsu Mon Maung is with the University of Computer Studies, Mandalay, Myanmar (e-mail: hsumon77@ gmail.com).

Kaythi Win is with University of Computer Studies, Mandalay, Myanmar (e-mail: kthiwin11@gmail.com).

DOI: 10.7763/IJIET.2015.V5.608

test suite. The authors depicted some limitations of an approach to testing Web applications automatically and introduce some ideas for improving upon it.

## III. Background Theory

### A. User Session Based Testing

A user session based test case is a sequence of user requests in the form of base requests and parameter name value pairs (eg, form field data) [4]. 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 [8]. The key advantage is the minimal configuration changes that need to be made to the web server to collect user requests [6]. The example web server log is as the following:

192.168.1.87 - - [23/Jan/2014:14:25:05 +0630] "GET /E-Library/components/com_jevents/views/default/assets/images/evback.jpg HTTP/1.1" 200 364 "Mozilla/5.0 (Windows NT 6.0; rv:26.0) Gecko/20100101 Firefox/26.0"

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. A time out interval of 30 minutes is taken to identify the user session.

### B. Test Cases Reduction

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 [9]. We need to reduce the test suite sizes while maintaining the web application to their cost effectiveness. In this paper, entropy based test cases reduction approach is proposed. Shanon's information entropy E can be expressed as [10]:

$$E(H) = -\sum_{i=0}^{n} P_i \log_2 P_i \tag{1}$$

where $P_i$ is the probability of link$_i$ that are accessed by users and $n$ is the number of links of web site. The reduction method can reduce smaller test suite size than original suite and can improve fault finding effectiveness.

## IV. Proposed System

Our proposed system consists of three main phases: generating user sessions data, reducing test suites and prioritizing reduced test suites for subject application. The overview framework of proposed system is shown in Fig. 1.

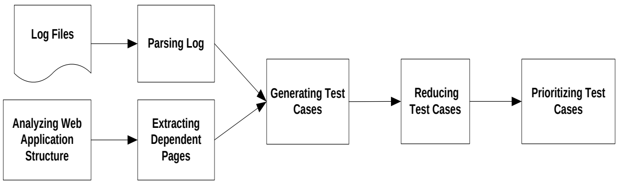

Fig. 1. Framework of proposed system.

### A. Test Cases Generation

This system combines the user session data with the internal structure analysis of web application as gray box testing because test cases generation totally based on user session data is not ensure for the reliability of web application. Therefore, the structure of web application is first analyzed to extract the dependent pages as a preprocessing step for ordering access time of these pages.

In generating user sessions phase, we collect the web server's access logs to record the GET and POST requests, IP address, cookie information and time stamps. Then, the user's access logs are parsed according to their field respectively. The output of parser is further edited to create the binary relation table as shown in Table I. In this table, 1 is user's access link and user's unvisited link is 0.

TABLE I: Sample Relation Table

| Users | Index.php | Login.php | Showcat.php | Tbook.php | View.php | Reg.php |
|---|---|---|---|---|---|---|
| U1 | 1 | 0 | 0 | 1 | 1 | 0 |
| U2 | 1 | 0 | 1 | 0 | 1 | 0 |
| U3 | 1 | 1 | 1 | 1 | 1 | 0 |
| U4 | 1 | 0 | 0 | 1 | 1 | 0 |
| U5 | 1 | 0 | 1 | 1 | 1 | 0 |
| U6 | 1 | 1 | 1 | 0 | 0 | 1 |

### B. Entropy Based Test Cases Reduction

In second phase, the entropy based reduction method is proposed to reduce test cases for testing web site. The higher entropy value leads to the more URLs covered. We calculate the entropy value based on dependent pages of structural analysis.

The total number of links is chosen as the base logarithm to normalize the entropy value to the range [0, 1]. Therefore, the entropy equation becomes as follows:

$$E(H) = -\sum_{i=0}^{n} P_i \log_n P_i \tag{2}$$

where $n$ is the total number of links.

For example, the entropy value for base request accessed by user is calculated by using entropy equation (2):

$$E(U_1, U_2, U_4) = -\sum_{i=0}^{3} \frac{1}{3} \log_6 \frac{1}{3} = 0.61314 \tag{3}$$

$$E(U_3) = -\sum_{i=0}^{5} \frac{1}{5} \log_6 \frac{1}{5} = 0.89824 \tag{4}$$

$$E(U_5, U_6) = -\sum_{i=0}^{4} \frac{1}{4} \log_6 \frac{1}{4} = 0.77371 \tag{5}$$

As the example calculation, low entropy indicates that few links are covered and high entropy indicates that more links are covered. The entropy value of each user is shown in Table II.

The maximum entropy value of user that covers all or most URLs is selected as a test case. According to Table II, the entropy gain of user 3 is highest, and thus it is selected as a test case. If selected test case is not covered all URLs in typical application, we need to select second highest entropy value of user until all URLs are covered.

In some cases, there are two users that have same entropy values. In this case, we need to consider the time of base request that is accessed by user. For example, the second highest entropy, U5 and U6 have the same entropy value. If U5 is selected as a test case, we will lose the events of U6 according to the time stamp in Table III. Therefore, U6 is selected as next test case.

TABLE II: RESULTS OF ENTROPY VALUE COMPARISON

| Users | Entropy(H) | No. of Base Request |
|---|---|---|
| U1 | 0.61314 | 3 |
| U2 | 0.61314 | 3 |
| U3 | 0.89824 | 5 |
| U4 | 0.61314 | 3 |
| U5 | 0.77371 | 4 |
| U6 | 0.77371 | 4 |

Example test cases are:

U3: {Index.php - Login.php - Showcat.php - Tbook.php - View.php}

U6: {Index.php - Login.php - Showcat.php - Reg.php}

The test suite (U3 U U6) covers all base requests in web application structure. Both coverage URLs and access time are considered in reducing test cases.

TABLE III: TIME STAMP ORDER OF BASE REQUESTS

| Users | Time Stamp | No. of Base Requests |
|---|---|---|
| U1 | 00:30 | Login.php |
| U2 | 00:32 | Login.php |
| U3 | 00:36 | Login.php |
| U4 | 00:40 | Login.php |
| U5 | 00:41 | Login.php |
| U6 | 00:46 | Login.php |

### C. Test Cases Prioritization

Test cases prioritization is the process of scheduling the execution of test cases according to some criteria to satisfy a performance goal [4]. The reduced test suites are prioritized based on several criteria in order to improve fault detection rate. Most current techniques use a random method to prioritize test cases when they have the same priority.

In the prioritization phase, proposed criteria consider this fact that there are multiple test cases with the same priority. Two factors, frequency (Feq) and time (T) are considered to order the test cases. In the case that the test cases have same frequency, the second factor T (time) is considered to prioritize test cases because the latest user event gives the update information. Only sequences of base requests are considered ignoring the parameter-value pairs. The procedures of proposed prioritization criteria are:

Step 1: The frequency table is constructed to order the test suites based on occurrences of base request sequences, $S_i$.

Step 2: Order the sequences ($S_0$, $S_1$, …, $S_i$) in decreasing order of frequency.

Step 3: Select the test cases that have maximum occurrence of sequences for execution before other test cases.

Step 4: If the test cases have the same frequency of sequences, the test cases are ordered in decreasing manner and test case containing latest time sequence is selected as first priority.

In this criterion, time and frequency of sequences are used to prioritize test cases. For example test suites in Table IV, test cases, Tc3, Tc5, Tc6 are selected first as it occurs maximum times of access sequences.

Notice that these test cases have same number of times. In this case, we select the test case according to the time stamp of access sequence (Table III) for prioritization test suite. Therefore, tc6 is selected as the first test case because this access time is later than other test cases. The prioritized test suite in Table IV is Tc6, Tc5, Tc3, Tc1 and Tc4.

TABLE IV: FREQUENCY TABLE OF EXAMPLE TEST CASES

| Sequences | Frequency | Test Cases |
|---|---|---|
| (Index.php, Login.php) | 5 | Tc6 |
| (Index.php, Login.php) | 5 | Tc3 |
| (Index.php, Login.php) | 5 | Tc5 |
| (Index.php, Login.php) | 3 | Tc1 |
| (Index.php, Login.php) | 2 | Tc4 |

### D. Subject Application

We used the log file data of e-library system to generate the test cases on web application testing. The functionalities of e-library system are divided into two types: user activities and admin activities. Our proposed system concentrates on functionalities that are accessible to the users and not the administration function.

### E. Dependent and Independent Variables of Proposed System

In our study, the independent variables are the test cases reduction and prioritization criteria, subject application, reduction requirements and original test suites. The dependent variables are fault detection rate, reduction time of test suites and space of reduced test suites.

### F. Evaluation Effectiveness of Proposed System

To evaluate the effectiveness of reduced and prioritized test suites, testers can use fault detection rate. A fault is defined as an incorrect statement, step, process, or data definition in a computer program [11]. We execute the correct version of application, and save the expected results. The different types of faults which are GUI faults, database faults and navigation faults are manually seeded to the application. The reduced and prioritized test cases are replayed to the faulty version of the application. After replaying test suite, the actual results are obtained. We compare the expected results and the actual results to determine that the test cases pass or fail. The following equation is used to evaluate the effectiveness of fault detection by test cases.

$$FDR = \frac{\text{Fault detected in testing}}{\text{Total number of faults}} \tag{6}$$

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

## V. CONCLUSION

In user session based testing of web system, web usage logs are very large and thus resulting in a large set of tests cases. Using large amount of test cases in testing is not practical within time constraint. In this paper, the new hybrid approach is proposed in which it combines the user session data and structural analysis of web site to generate reasonable test suite. The entropy based reduction technique creates test suites that are smaller than original test suites but the reduced test suites themselves can be very large. Therefore, the time and frequency based prioritization is proposed to reorder the reduced test suite. The fault detection rate can be improved by using this hybrid criterion. We have not yet fully compared our approach to current user session based testing techniques. In the future, the abilities of fault detection will be evaluated with other test cases reduction approaches.

## REFERENCES

[1] S. Sprenkle, E. Gibson, S. Sampth, and L. Pollock, "A case study of automatically creating test suites from web application field data," presented at TAVWEB06, Portland, Maine, USA, July 2006.

[2] S. Sprenkle, E. Gibson, S. Sampth, and L. Pollock, "An empirical comparison of test suite reduction techniques for user session based testing of web application," in *Proc. the 21st IEEE Int. Conf. Softw. Mainten.* Budapest. Hungary, 2005, pp. 587-596.

[3] E. Shamsoddin-Motlagh, "A review of automatic test cases generation," *International Journal of Computer Applications (0975 – 8887,)* vol. 57, no. 13, November 2012.

[4] S. Sampath and R. C. Bryce, "Improving the effectiveness of test suite reduction for user session based testing of web application," *Information and Software Technology,* vol. 54, pp. 724-738, 2012.

[5] S. Elbaum, S. Karee, and M. Fisher II, "Leveraging User-Session Data to Support Web Application," *IEEE Transactions on Software Engineering,* vol. 31, no. 3, March 2005.

[6] S. Sampth, V. Mihaylov, A. Souter, and L. Pollock, "A scalable approach to user session based testing of web applications through concept analysis," in *Proc. 19th Int. Conf. Automated Sofw. Eng,* Washington DC, USA, 2006, pp. 132-141.

[7] A. Pobletts, C. Cobb, and L. Simko, "CREU 2011 final report toward an effective data model and user session dependency model," 2011.

[8] S. Sampth and R. C. Bryce, "Prioritizating user session based test cases for web application," *ICST,* April 2008.

[9] H. M. Maung, "Test case reduction approach in user session based testing for web application," in *Proc. ICCA,* February 2013.

[10] M. C. Thomas and J. A. Thomas, *Elements of Information Theory,* Wiley-Interscience, 2006.

[11] R. Binder, *Testing Object-Oriented Systems,* Reading, MA: Addison Wesley, 2000.

**Hsu Mon Maung** received her bechelor of computer technology from Computer University (Meikhtila), Myanmar. She completed master course from Computer University (Taunggyi). Now, She is a Ph.D student in University of Computer Studies, Mandalay (UCSM), Myanmar. Her fields of interest are software engineering, software testing and validation.