User Session based Test Cases Prioritization for Web Applications Testing

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

Abstract

Nowadays, web application is crucial for most daily activities that rely on the web services. To improve the quality of reliable web services, web application testing has been used in finding faults under test. User session based testing is one approach to create test cases with real user data that are collected from daily user logs. The main considerable problem is how to reduce the cost of web application testing in time without service interruption. However, real user data is reduced as the test cases, all the reduced test cases cannot be executed completely under time constraints in practice. In this paper, the test cases prioritization criterion is proposed to schedule the test cases in order to improve the rate of fault detection. This criterion is based on two factors, frequency of sequences and dependent count of web pages. Most current techniques use a random method to prioritize test cases when they have the same priority. The proposed prioritization method considers this fact that there are multiple test cases with the same priority.

Keywords: user session based testing, test cases prioritization, web application testing

1. Introduction

Web application testing is not only an expensive process in terms of time and money but also crucial part for web application. There are several reasons why it is important. First, daily web application services must be available 24/7. Second, the nature of web applications are frequently changed and upgraded due to security attacks and maintenance changes. Furthermore, web application testing must be performed completely in time without service interruption. Testers can detect faults early in the test execution life cycle because failures in this domain cause in losses of web based business or organization.

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. A user session based test case is a sequence of base requests and parameter name value pairs. The advantages of user session based testing are less dependent on heterogeneous system and can generate test cases that reflect actual user behavior. But, there is a considerable issue that is collecting, analyzing and replaying the large amount of test cases generated from user session data [2]. Many researchers presented various reduction and prioritization techniques to solve these issues.

There have been several strategies to prioritize test cases for web applications. We expand on previous work to propose prioritization criteria for web application testing. Entropy bases test cases reduction approach is previously proposed for user session based testing. Even the reduced test suites can be large to execute in some commercial system. Therefore, test cases prioritization method is proposed based on multiple sequences of base requests. The purpose of test cases prioritization lies in ordering test cases based on a particular technique [3]. In this system, multiple base request sequences (sequences of size 2) are ordered with proposed criteria to determine which reduced test cases run at first. We consider frequency of user access requests and dependency of web links by structural analysis. There are two main parts in proposed system: (1) criterion to prioritize user session based test cases and (2) evaluate proposed criterion by using fault detected rate.

Section 2 presents related works concern with user session based test cases reduction and prioritization approaches. In section 3, background theory in user session based testing and test cases prioritization. Section 4 describes our proposed system and evaluation of proposed system in. The conclusion of proposed system is described in section 5.

2. Related Works

S. Roongruangsuwan and J. Daengdej [4] proposed two new efficient prioritization methods to address the problem of failing multiple test cases prioritization and same priority cases. This study proposed four prioritization factors to schedule test cases.

S. Sampath and R. Bryce presented ordering the tests in a reduced suite to increase its rate of fault detection [5].There are several approaches to order reduced test suites using prioritization criteria for the domain of web application.

Sampath et al. [2] 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.

The studies [6] explored a method of estimating dependencies automatically and using them to arrange the test suite. The authors depicted some limitations of an approach to testing Web applications automatically and introduce some ideas for improving upon it.

Another method presented by A. K. Upadhyay and A. K. Misra [7], is prioritization test cases using clustering approach in software testing. This study applied dendrogram methods for prioritization method and also showed the significant improvement in fault detection rate by prioritized test cases.

3. Background Theory

3.1. User Session based Testing

User session based testing is a type of black box 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) [8]. 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 [9]. The key advantage is the minimal configuration changes that need to be made to the web server to collect user requests [2]. Sample process of user session data collecting is shown in figure 1.

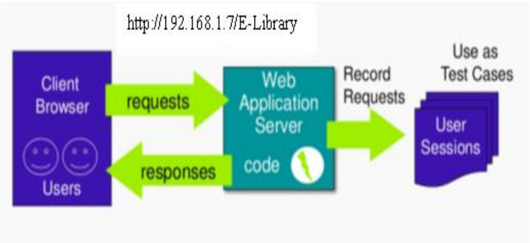


Figure 1. User Session based Test Cases

The user session based test cases can be viewed the perspective of the different test requirements. The test requirement data can be extracted from collected log data. The summarized test requirements are described as follows [10].

- base: Form of a base request
- Seq: Form of base sequences of size 2
- Name: Form of base request and parameter name
- Name_value: Form of base request, parameter name and parameter value
- Seq_name: Form of size 2 sequences of base request and parameter name

3.2. Test Cases Prioritization

Test cases prioritization techniques become more significant when the time to replay all the tests is insufficient under test. These techniques prioritize and schedule test cases in order that attempts to maximize some objective functions. [4]. Rothermel et al.[11] define test cases prioritization problem as follows:

Given: T, a test suite;
PT, the set of permutations of T;
F, a function from PT to the real numbers;

Problem: Find T′ € PT such that (∀T″) (T″€PT) (T″ ≠ T′) [f (T′) ≥ f (T″)]. Here, PT is set of all possible prioritizations of T and f is a function that, applied to any such ordering. In prioritization process, test cases are executed according to some criterion to satisfy some performance goal. Some prioritization criteria are length based, code based, frequency based and other possible criteria. Most criteria focus on the goal of increasing the rate of fault detection earlier in the testing process.

Frequency of user access requests can be obtained from user session data. The proposed criterion consider occurrence of multiple requests as one factor to prioritize test cases. The second factor dependent count of web pages is taken to solve same frequency cases. The dependent links are extracted by using structural (white-box) analysis.

4. Proposed System

Web application may have a large number of users and corresponding large number of user sessions. The user interactive web logs are stored in web servers. The testers extract user session from web server logs and use them as test cases for testing web application.

In reduction techniques, the criteria create smaller test suite than original suite but test cases are in no specific order. Therefore, we propose new multiple sequences prioritization criteria by expanding our previous reduction work to be more efficient in fault finding. The overall process of proposed system is shown in figure 1.

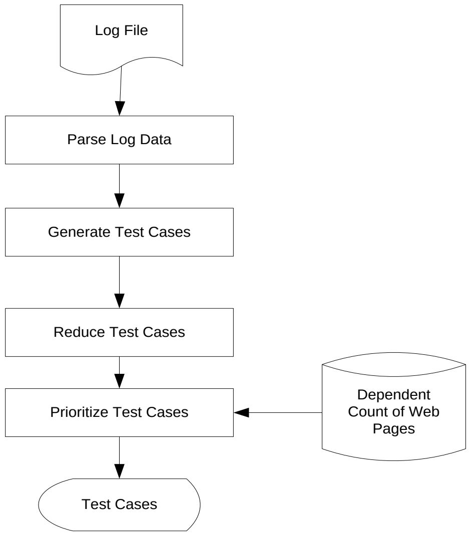


Figure 1. Overall Process of Proposed System

4.1. Test Cases Generation

In test cases generation process, we collect the daily usage log of specific web application. The user session data is parsed to extract necessary information such as IP address, GET, POST methods, time stamp and cookie information. The unwanted and redundant data are removed. The sample log data that used in our system is shown in figure 2. The user access log is converted into test cases in the form of http requests that can be sent to the web server.

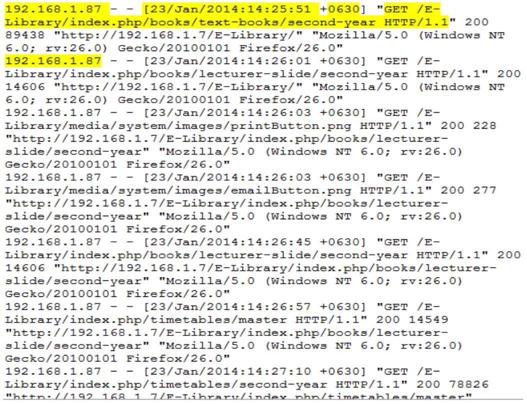
4.2. Test Cases Reduction

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.

In this phase, Shanon's entropy gain theory is applied to reduce test cases [12].

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

where P_i is probability of $link_i$ that are accessed by users and n is the number of links of web site. This process is proposed in our previous work to reduce test cases that are not only small in size but also equivalent in effectiveness to an original test suite. The concept in this reduction method is that the higher entropy value may lead to more URLs covered.



```
192.168.1.87 - - [23/Jan/2014:14:25:51 +0630] "GET /E-
Library/index.php/books/text-books/second-year HTTP/1.1" 200
89438 "http://192.168.1.7/E-Library/" "Mozilla/5.0 (Windows NT
6.0; rv:26.0) Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:26:01 +0630] "GET /E-
Library/index.php/books/lecturer-slide/second-year HTTP/1.1" 200
14606 "http://192.168.1.7/E-Library/" "Mozilla/5.0 (Windows NT
6.0; rv:26.0) Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:26:03 +0630] "GET /E-
Library/media/system/images/printButton.png HTTP/1.1" 200 228
"http://192.168.1.7/E-Library/index.php/books/lecturer-
slide/second-year" "Mozilla/5.0 (Windows NT 6.0; rv:26.0)
Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:26:03 +0630] "GET /E-
Library/media/system/images/emailButton.png HTTP/1.1" 200 277
"http://192.168.1.7/E-Library/index.php/books/lecturer-
slide/second-year" "Mozilla/5.0 (Windows NT 6.0; rv:26.0)
Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:26:45 +0630] "GET /E-
Library/index.php/books/lecturer-slide/second-year HTTP/1.1" 200
14606 "http://192.168.1.7/E-Library/index.php/books/lecturer-
slide/second-year" "Mozilla/5.0 (Windows NT 6.0; rv:26.0)
Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:26:57 +0630] "GET /E-
Library/index.php/timetables/master HTTP/1.1" 200 14549
"http://192.168.1.7/E-Library/index.php/books/lecturer-
slide/second-year" "Mozilla/5.0 (Windows NT 6.0; rv:26.0)
Gecko/20100101 Firefox/26.0"
192.168.1.87 - - [23/Jan/2014:14:27:10 +0630] "GET /E-
Library/index.php/timetables/second-year HTTP/1.1" 200 78826
"http://192.168.1.7/E-Library/index.php/timetables/master"
```

Figure 2. Sample Log Data of UCSM E-Library

4.3 Process of Proposed Prioritization Method

The reduction techniques are based on covering all base requests but test cases are in no priority to run first. In this system, the new prioritization criterion is proposed to be more efficient in fault finding of reduced test suites.

Test cases prioritization is the process of scheduling the execution of test cases according to some criteria to satisfy a performance goal [4]. This prioritization technique is mainly focused on two factors *feq* (frequency) of accessed sequences and *dept* (dependent count) of links. We consider the multiple sequence of size 2 in this system. First, the most frequently access of base request in the test suite are identified. The frequency table is constructed in decreasing order of occurrence of this sequence in the test suite. Sample frequency table for test cases is described in table 1.

Table 1. Frequency Table of Access Sequence

Sequence	Frequency	Test Cases
(A, B)	7	Tc1
(B, C)	7	Tc3
(A,D)	7	Tc6
(A, C)	4	Tc2
(C, D)	2	Tc4
(C, C)	1	Tc5

Most current techniques use a random method to prioritize test cases when they have the same priority. For example in table 1, the frequencies of the sequences (A, B), (B, C) and (A, D) are same and thus we need to select one of them to run first. In this case, the second factor, dependent count (*dept)* of each page is considered to prioritize test cases. Dependency of pages is the important factor for prioritization because this may increase the fault detection rate. The dependent counts of each page are listed in Table 2.

Table 2. Dependent Count of Each Page

Pages	LDC (link dependent count)
A.php	0
B.php	2
C.php	1
D.php	4
E.php	5
F.php	1
G.php	1

We select the test case that has maximum dependent count according to structural analysis because this count depicts the coverage of user visited pages. Thus, the proposed technique addresses the problem of choosing which test case is run first among multiple test cases with the same priority. In this case, for example, we select the test case according to dependent count of access sequence (TABLE 2) for prioritization test suite. Therefore, Tc6 is selected as the first test case to run because this dependent count is greater than other test cases. The prioritized test cases by applying proposed procedure in TABLE 1 is Tc6, Tc3, Tc1, Tc2, Tc4 and Tc5. In our experiment, the original user sessions are 65 and the reduced test cases from these user sessions are 32. All reduced test cases in test suite are prioritized based on proposed prioritization technique.

The procedures of proposed method are described in table 3:

Table 3. Procedure of Proposed Method

Input : set of test cases T set of base request sequences S (size 2) dependent count dept Output : prioritized test cases PT
Step 1: Construct frequency table based on occurrences of base request sequences, si. *Step 2*: Order the dependent count (*dept)* of each link in decreasing manner. *Step 3*: Order the sequences s_i € S in decreasing order of frequency. *Step 3*: Select the test cases ti € T that have maximum occurrence of sequences. *Step 4*: If the selected test case ti has the same frequency with another test case, select the test case that has maximum dependent count as the first priority. *Step 5:* Go to step 3 and 4 to order all test cases in the test suite.

4.4. Subject Application

We used the log data of UCSM's e-library system to generate the test cases on web application testing. The links of this web site are extracted by using online Buzzstrem [13] link extractor tool. The total number of links is 80.The functionalities of e-library system are divided into two types: user activities and admin activities. The functionalities that are able to user are considered to test in our proposed system. The administrator's function is ignored.

4.5 Evaluation of the Proposed System

Fault detection technique can be used in evaluating the efficiency of prioritized test cases. The original version V is a non faulty version of the application in which there is no faults and changes. Different types of faults are manually seeded into the application. This version is called modified version V′ that has been changed by developers or there is seeded faults. In our system, two types of faults are used to inject into web application. These are [14]:

Form faults: faults in the application code that controls, modifies and displays name-value pairs in forms,

Link faults: faults in the application code that change the page pointed to by a URL.

The prioritized test cases may be run with both V (original version) and V′ (modified version). The actual results are obtained when these test cases are replayed into modified version and expected results are obtained from original version. Both expected and actual results are compared with the comparator to determine how many faults are detected by test cases.

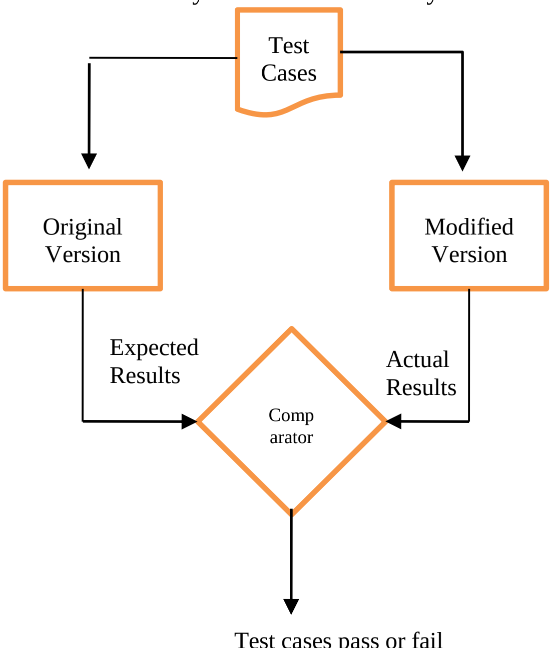


Figure 3. Test Cases Replaying Process

The following equation is used to determine the ability of fault detection rate by test cases [6].

$$FDD = \frac{tf_1 + tf_2 + ... + tf_n}{|T| * |F|} \qquad (2)$$

FDD (fault detection density) is defined as the ratio of the sum of the total number of faults detected by each test case and the total number of test cases, normalized with respect to the total number of fault detected. T is a set of test cases and F is a set of faults detected by test cases in T. tf_i is the number of faults detected by t_i. FDD value of 1 for a test suite indicates that each test case in the suite detects every fault [10]. The fault detection density of our proposed system will be evaluated in the future.

5. Conclusion

Large amount of user session based test cases in web application testing is not practical within time constraint. In this paper, a new prioritization method is presented to make more efficient the fault detection rate of reduced test cases. The proposed method can be applied with multiple base request sequences and also same priority cases. In this system, dependent relationships between pages are considered to enhance the fault detection rate. Our system is scalable and flexible with system evolution. We have not yet fully compared our approach to current user session based test cases prioritization techniques. In the future, the abilities of fault detection will be evaluated with other 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," Portland, Maine, USA, TAVWEB06 (Jul. 2006).

[2] 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.

[3] S. G. Elbaum, A. G. Malishevsky, and G. Rothermel "Test Case Prioritization: A Family of Empirical Studies" IEEE Transactions on Software Engineering, 28(2): 2002.

[4] S. Roongruangsuwan and J. Daengdej "Test Cases Prioritization Techniques",Journal of Theoretical and Applied Information Technology, JATIT 2010.

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

[6] A. Pobletts, C. Cobb, L. Simko, "Toward an Effective Data Model and User Session Dependency Model", Conf, 2011.

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

[8] A. K. Upadhyay and A. K. Misra "Prioritizing Test Suites using Clustering Approach in Software Testing". ISSN: 2231-2307, volume-2, Issue-4, September, 2012.

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

[10] S. Sampth and R. C. Bryce, "Prioritizating User Session based Test Cases for Web application".

[11] S. Sampath, R. C. Bryce, "Improving the Effectiveness of Test Suite Reduction for User Session based Testing

[12] of Web Application", Information and Software Technology 54 (2012), 724-738.

[13] Rothermel et al.[, "Prioritizing Test Cases for Regression Testing", Transactions on Software Engineering 27 (10) (2001).

[14] H. M. Maung, "Test Cases Reduction Approach in User Session based Testing for Web Application", ICCA, (2014).

[15] http://tools.buzzstream.com

[16] Y. Huang, and Lu Lu, "A Methodology for Test Suite Reduction in User Session based Testing", Journal of Coverage Information Technology, vol 5, no 9, November 2010.