ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 4, April 2016

Event- Driven Software Testing – An Overview

Sunayana Chaudhury[1*], Abhishek Singhal[1] and Om Prakash Sangwan[2]
[1]Department of Computer Science & Engineering, AMITY School of Engineering & Technology, AMITY University, Sector 125 NOIDA-201 313 (UP) INDIA
[2]Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar-125 001, Haryana INDIA

***Abstract*— Event Driven Software (EDS) are one of the widely used software's which considers a large number of events invoked by users and present with new sequence of events as the output. It comprises both the types i.e. GUI based and web based software's. The testing is a very challenging task as a large number of events can be invoked by the users so it can be an exhaustive job for the tester. This paper provides a deep study about Event Driven Software and their testing for bug free software. In this we use prioritization criteria to make the testing much more convenient as the nature of EDS is large amount of events invoked by the users through an interface**

Index Terms*— Event Driven Software (EDS), GUI testing, Test Case Prioritization, t-way Interaction Coverage, web application testing

I. INTRODUCTION

The quality of software is assured by testing the software development, application and its progression. While testing, test cases are usually created and followed by their execution on an Application under Test (AUT) [1]. Event Driven Software (EDS) takes large number of test cases so, we require new testing techniques for undertaking EDS. Owing to the nature of event driven it takes a number of inputs invoked by the user and consequent upon state change it generates a new combination of inputs denoted as the new output. Different outputs are generated for different combination of inputs which EDS generates. It is very tedious job to test all the possible combinations as the growth of the events is exponential. The examples of EDS are Graphical User Interfaces (GUI's) and Web Applications. It gives flexibility to user's as for example considering in GUI, user can input the GUI events (mouse clicks, typing in text field) differently varying on the number, execution order and the type. EDS is widely used as the software interacts with the user through the sequence of events generated by the user and performs accordingly. In this paper we will cover all the aspects related to EDS and testing of EDS using prioritization as a criterion.

II. RELATED WORK

The Event Driven Software (EDS) approach was proposed by Xiao-ling [2] by way of testing technology of Event Driven Software and analyzed the differences between Event Driven Software and the traditional software. It is procedure oriented and based on above analysis the mechanism of event driven and its effect on software testing was unveiled. The event coverage criterion was also defined. The event executing rules such as concurrent event, ordinal event, non ordinal event and predecessor event were described [2] which were followed by method of testing according to the rules.

Bryce and Memon [3] postulated testing model for both web and GUI applications. In this model test case prioritization was achieved by employing of interaction based criterion, frequency based on the usage and as well as the count [3]. The data demonstrated that the prioritization through a two-way (criteria based on interaction) and PV-L to S (criteria based on Parameter count) gave the perfect results in detecting the fault rates for the GUI applications. However, main disadvantage lies in the fact that multiple criteria for prioritization are combined to give the final output but in many instances it has been found that multi criteria is more useful rather than considering a single criterion.

Huang et al. [4] reported both weighted event flow graph and GUI Test case prioritization, were employed for ranking the GUI test cases which are non-weighted. For assigning weights, events classification is on the basis of their importance throughout the application [4]

Bryce and Memon [5] put forwarded interaction coverage which helps in test suite prioritization. The t-way interaction coverage and comparison was performed by fault detection rate criteria for the prioritization of GUI based program Test suites. The test suits with maximum coverage of event interaction advantage the maximum, whereas, those having lesser coverage of interaction was a disadvantage as there was no benefit after employing this prioritization technique [5].

Rothermal et al. [6] discussed about the prioritization of test cases for regression testing which increases the effectiveness of test cases for implementation in meeting performance objective. One of it can be the fault detection rate which quantifies how fast a fault is observed throughout the process of testing. Therefore, rate of fault detection is much more fast and allows the developer in fixing the faults quickly. Prioritization involves one application of regression testing which is retesting once modifications are done in the software. An advantage of prioritization is that the information is comprehended about the past executed test cases to get the ordering of the test cases. Several techniques

for regression testing were used for extracting useful information by prioritizing test cases [6].
Yu and Lau [7] postulated the prioritization using the faults found by the test cases which straight away uses the information regarding ability of fault detection. Based on the proposed fault model interaction between the test cases and the faults is used for test case generation [7].

Gerrard [8] reported testing and prioritization of the GUI applications in which the fault detection effects are dependent on the way of interaction of a given event with another event and requires further exploration. The weight value assigned to will surely have an impression on ability of detecting fault for a given test case.

III. GUI AND WEB BASED APPLICATIONS: SIMILARITY (GENEARALISED MODEL):

GUI-Based Applications:
GUI acts as an interface between the software and the users. The interaction of user with the software through series of events invoked by the user and the software present with new sequence of events as the output which is recognized by changed GUI's widgets. GUI testing is stated as inspecting the full application by generation of only GUI inputs to find the errors and failures in the application.

Test cases are generated using tools i.e. replay/capture tools such as Win Runner which provider very less automation [9], as we need test cases which are run to test the application and find out the errors if present.

A Graphical User Interface (GUI) is a system which by nature is hierarchical and behaves as a front end for the software which accepts user invoked inputs and system generated events and in return produces an output.

Web Applications:
A web application contains pages which can be accessed by the users by using a web browser through a network which connects the two. A web page can be either static or dynamic. There can be million lines of code using languages such as Java Script, ASP, Java Servelts and HTML for programming.

The languages used can be differentiated into two as client-side languages and server-side languages. Java script falls into client-side language and ASP, Java Servelts falls into server-side languages.

Event generation in a web application can be generated by two ways:
1) In client side code an event gets triggered which bring changes in the user interface displayed to the user without code execution on server side like hovering over a link it makes the color of the link changes.
2) In client side code an event gets triggered which makes the execution of code on server side like pressing a submit button after filling a form.

Web application testing is stated as inspecting the full application by generation of inputs which are URL based to find the faults. It is a manual task. The tool used is a capture-replay tool which captures the tester action throughout the application and then replay on the web application [10].

Generalized Model:
The user-centric nature between Graphical User Interface (GUI) and Web Application makes them similar to each other and develop a single platform for both GUI and Web applications for test suite prioritization. Considering both the as 'window' .We have a pair <*parameter_name*,*value*> as parameter-values. The widgets present in the window are known as parameters and their settings are known as values and the term 'action' being considered as the series of interactions on a single window done by the users before switching to the new one.

Action listeners are the event handlers for a GUI application to be implemented. Consider an event like clicking a button an action event performed by the user which results into an action message which is sent to all the action listeners that have been in the application. We can create test cases from the given application to design a single model. The generated test cases are a series of the actions performed by the user. For each, a value for one or more parameters in the pair set is set by the user.

Similarly, for Web applications, the page is considered as a window. In case of GUIs, the widgets present on the window are considered as 'parameters', and 'values' being their settings. Different number of criteria's can be used when developing a single platform for both GUI and Web applications for test suite prioritization.

IV. FACTORS AFFECTING PRIORITIZATION OF TEST CASE

Here assignment of weight value is done by taking the following factors [11] into account:-

•Event Type
• Interaction of the Event
•Coverage of Event
The following is the description of various factors given above:-

• **Event Type** – The type of event plays a major role on the fault detecting potential of a test case generated by user input. As per earlier reports events can be broadly classified into five major types as Unrestricted-Focus Event, Menu-Open Event, Termination Event, System-Interaction Event and Restricted-focus Event. Based on the importance of each type of events weight value has been assigned to various types of events. For better understanding assignment of weight values is presented in Table 1.

TABLE 1: WEIGHT VALUES ASSIGNMENT TO VARIOUS EVENTS [12]

Type of Events	Weight Value Assigned
Unrestricted-Focus Event	1
Menu-Open Event	2
Termination Event	3
System-Interaction Event	4
Restricted-Focus Event	5

• **Interaction of Event** – Interaction of event in an Event Driven Software directs the program to follow an execution path which is quite different and has the potentiality of divulging new faults in the given system. Here in the anticipated technique the test cases which have high value of the event interaction are assigned the Priority [13].

•**Coverage of Event**- In order to obtain best results test suits which encompasses maximum coverage of event is given additional significance than the test suits providing lesser coverage. Therefore, it will include actions or parameter values as well as the window counts, which gets covered by the test case.

V. PRIORITIZATION CRITERIA

We can use various criteria for prioritizing the test cases generated for EDS testing. Prioritization of the test cases which have maximum fault revelation in the testing process is very important. Hence, prioritizing the test cases is one of the challenging part of research for maintaining QA in Software testing [14,15,16,17].

According to the prioritization techniques for the test cases, test cases are prioritized using priority as a parameter; in other words, it can be defined as a test suite say T, now PT being the number of possible permutations of T, and a function from PT to real numbers which is f [18]. We need to find T'ϵ PT, such that (V T‖)(T‖ ϵPT)(T‖ ≠T')[F(T')≥ f(T‖)].

Following are few strategies:

A) Using Parameter Values for Systematic Prioritization

Web application pages contain parameters for which *values* are specified the users.

Interaction Coverage using Parameter-value: In 2-way criterion next test is selected where the number of t- way parameter-value interactions gets maximized among the pages which occur in a test.

Parameter-value counts determining the Length: During a user session, parameter values are specified by the users. The Prioritization is done by considering the number of parameter values in a test case including duplicates. There are two categories which select those tests first for number of parameter values is highest ,called PV-L to S and the second one is called PV-S to L in which we prioritize but vice-versa by selecting those first which are having smallest number of parameter-values.

B) Random Prioritization

Random Prioritization just selects the next test in a random manner. In this, we keep selecting randomly until no more test cases are left to be selected for the prioritized test suite.

C) Prioritization by Test Length

Prioritization of length is done using base requests which by selecting the next test whose count of base requests including duplicates are the highest. The request generated in a test case roughly explains to what extent the test case is using the application code. The ordered test suite rate of fault detection can be impacted by the ordering of test cases which consider their length. The ordering of test cases can be done in two manners one is (Req-LtoS) descending and other is (Req-StoL) ascending order of the length, where length is the count of base requests for a test case

D) Prioritization based on Frequency

We can do the prioritization based on the frequency count of the page accessed sequences which can be seen in the test case. The failures have more impact on reliability as perceived by the application users; we consider on considering the test cases the frequency of accessing the application rises, which can improve the fault detection rate of the test suite which is ordered.

Depending upon the number of times the unique page sequences being accessed in the throughout test suite, we construct a frequency table. The considered sequences are in terms of base requests it means the parameter-value pairs are ignored and those sequences which involve interactions between Java servlet pages and JSP only; it means sequences that contain static HTML pages are not included.

Using the frequency table, we can do prioritization of test cases in two manners:

Most Frequently Accessed Sequence (MFAS): It consider the most frequently accessed sequence of request say 'as' and orders test cases according to the count 'as' is visible in the test case in decreasing order.

All Accessed Sequences (AAS): This approach considers all sequences for the test suite ordering. Considering every sequence, 'as', starting with accessed sequence most frequently occurring ; the maximum occurring test cases for the execution are selected for execution among others.

VI. CONCLUSIONS

Graphical User Interfaces (GUI's) and Web Applications are different areas but they both have some kind of similarity which allows creation of an exclusive single model for testing the EDS. Prioritization criteria makes the testing much more convenient as the nature of EDS is large amount of events

invoked by the users through an interface considering test cases having higher fault revealing capabilities. We concluded that the prioritization through a two-way (criteria based on interaction) and PV-L to S (criteria based on Parameter count) gave the perfect results in detecting the fault rates for the GUI applications. For web application frequency based technique is more suitable. Thus, we presented a single exclusive model for both the GUI and web-based applications which are the examples of an event-driven software system. The model results test orders which are more effective and results into better testing of EDS.

REFERENCES

[1] Gerrard Paul, "Testing GUI Applications" EuroSTAR, Edinburgh UK, 1997.

[2] DING Xiao-ling, DING Chun and HOU Young-Hong "Utilizing transient/constant blunders to create robotized test prophets for event driven programming" In Proceedings of Automated Software Engineering 2004 The International Conference on (ASE'04) (Linz, Austria, Sept. 2004).

[3] Sampath Sreedevi, Bryce Renee C. and Memon Atif M, "Test prioritization station for occasion driven programming and Building up a solitary model" Software Engineering IEEE Transaction, Jan 2010

[4] Huang Chin-Yu, Chang Jun-Ru, and Chang Yung-Hsin, "Outline and examination of GUI experiment prioritization utilizing weight-based strategies" The diary of Systems and Software 83, pp 646-659, 2010

[5] Bryce Renee C. and Memon Atif M "Interaction Coverage by Test Suite Prioritization by Test Automation" Workshop on Domain-Specific Approaches to Software, Dubrovnik, Croatia, 2007

[6] Rothermel, G., Untch, R.H. Chu, C. and Harrold, M J. "For Regression Testing Prioritizing Test Cases" IEEE Transactions Software Engineering 27: (10) 929-948, 2001

[7] Yu Yuen Tak and Lau Man Fai "Fault-based test suite prioritization for detail based testing" Inf. Software Technology 54, pp. 179-202, February 2012.

[8] Gerrard Paul, " Testing and prioritization of the GUI applications ", Edinburgh UK, 1997

[9] A.M. Memon and Q. Xie, "Studying the fault detection effectiveness of GUI test cases for rapidly evolving software," IEEE Transaction on Software Engineering, Vol-31, No. 10, pp. 884-896, 2005

[10] "Web Site Test Tools and Site Management Tools," http://www.softwareqatest.com/qatweb1.html, Apr. 2009

[11] Chaudhary, N. Sangwan, O.P. and Singh, Y. "Experiment Prioritization Using Fuzzy Logic for GUI based Software".Worldwide Journal of Advanced Computer Science and Applications. 3:(12). 2012.

[12] Memon Atif, Lou Soffa Mary, E. Pollock Martha, ―Coverage criteria for GUI testing,‖ in the proceeding of 21st International conference on software engineering, ACM press, pp 257-266, 1999.

[13] Bryce Renee C. and Memon Atif M, "Test suite prioritization by communication scope" Test Automation Workshop On Domain-Specific Approaches to Software September, 2007

[14] Rothermel G,.Elbaum S and Malishevsky A. G. "A group of exact studies: Test case Prioritization", IEEE Transactions on Software Engineering vol. 28 (2), pp. 159–182, 2002.

[15] Bryce Renee C. and Memon Atif M, "Test suite prioritization by communication scope" Test Automation Workshop On Domain-Specific Approaches to Software September, 2007.

[16] Memon Atif M. and McMaster Scott "Call-Stack scope for GUI test suite decrease" Software Engineering IEEE Transaction , Volume 34, Jan/Feb, 2008

[17] Memon Atif M. and McMaster Scott "Call Stack Coverage of the GUI test-suite diminishment", Proc., Seventeenth International Symposium on Software Reliability Engineering, Nov. 2006.

[18] Memon Atif M and Bryce Renee C. "Interaction Coverage by Test Suite Prioritization by Test Automation" Workshop on Domain-Specific Approaches to Software, Dubrovnik, Croatia, 2007.

Authors' Profiles

Sunayana Chaudhury is pursuing her M.Tech. degree in Computer Science Engineering from Amity University, Noida, B.Tech. in Computer Science Engineering from Guru Jambheshwar University of Science & Technology, Hisar, Haryana in 2014. Her area of research interest is Software Engineering.

Dr. Om Prakash Sangwan received his PhD in Computer Science & Engineering and Master of Technology (M.Tech.) degree in Computer Science & Engineering with distinction in Research Work from Guru Jambheshwar University of Science & Technology, Hisar, Haryana. He is also CISCO Certified Network Associate (CCNA) and CISCO Certified Academic Instructor (CCAI). His area of research is Software Engineering focusing on Planning, Designing, Testing, Metrics and

application of Neural Networks, Fuzzy Logic and Neuro-Fuzzy. He has numbers of publications in International / National Journals and Conferences. He is working as Associate Professor with Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, Haryana,

Abhishek Singhal is currently working as Assistant Professor-GradeIII, Department of Computer Science & Engineering ASET, Amity University, Uttar Pradesh. He is currently pursuing his Ph.D. from Amity University, Uttar Pradesh. His area of research is Software Engineering.