

Test Case Prioritization Based on Specific Events

Ms. Snehal D. Shinde

Department of Computer Engineering
Bharati Vidyapeeth's Deemed University College of
Engineering, Pune.
snehalmp14@gmail.com

Dr. Shashank.D. Joshi

Department of Computer Engineering
Bharati Vidyapeeth's Deemed University College of
Engineering, Pune.
sdj@live.in

Abstract— Event-Driven Software (EDS) system changes its state according to arrival of events for example graphical user interface and web framework. So due to there are number of events generated by users waiting in queue, this system is raise issue for testing. Until now, there are more efforts taken for testing this issue but these efforts are not collective. In this project work, our try is to give collective solution for graphical user interface and Web frameworks combined. We designed model to test graphical user interface and web application combined by using test cases prioritization. Main objective is here to deploy this model to prioritize test cases based on events. Our proposed work shows that graphical user interface and Web-based frameworks, gives same behavior even after prioritization. To test stand-alone GUI and Web-based frameworks based on shared prioritization function, and prioritization criteria's. This generic approach is enough to study develop and test a unified theory for all kinds of Event Driven Software systems. This paper articulates all the details regarding our proposed system through following sections.

Keywords-Combinatorial interaction testing, covering arrays, event driven software (EDS), t-way interaction coverage, test suite prioritization, user-session testing, web-application testing, GUI testing.

I. INTRODUCTION

Event-driven software plays an important role in today's software systems. Most of customer specific software's has a GUI that interacts with the system user through some events like mouse clicks or keystroke. Similarly, web frameworks also depend on particular network protocols and embedded software. These types of frameworks are quite different which require qualitative testing. Many existing techniques are trying to work on fault detection for sequential ordering test cases. In most of the organizations for graphical user interface and web, resource working on testing for performance improvement is complex. These frameworks are always communicating some important features. Regarding this both areas when user interacts, they generate order of events, which disturb working of system. Traditional system shows that they have not worked yet on this method because it involves lot of steps for incoming events so testing each event is quite time consuming. Up to today there are many techniques has been developed for GUI and Web frameworks testing. Above-mentioned are common things in GUI and web frameworks testing but problem related to both of these solved separately because two causes. First, there are not adequate tools and frameworks to study these problems for researchers. Second, if single model is formed for this two applications it will perfectly work for event driven system because it separate attributes which are not necessary in functional test. So there is necessary to combine this two applications as one model to test, test suite.

In Generalized model, we can use different no. of criterions for both GUI applications and web-based applications. Reminder of this paper gives brief description of this approach with, section II gives background details, section III gives detailed description of system section IV shows result analysis and we summarize in section V by drawing conclusion.

II. RELATED WORK

This section surveys some existing methodologies related to our topic. Arne Michael et al.[1] presented approach for test cases methodology. This paper describes design implemented for black box testing methodology of web frameworks. Here to sake of web frameworks control flow integrated with information flow. This model test strategy provides handy test cases, which easily accommodated in automated testing tools and test oracle by model. Case study presented in paper to that depicts the effectiveness in a web frameworks designing project, most importantly in maintenance circumstances. Saswat Anand et al. [2] presented automated testing approach. The designed app roach desired only for smart phones. They developed methodology for input events, to achieve this concolic testing method has been used which gives automatic events sequentially .system is specific for android platform and results has been presented on five real time applications. Result analysis show that the presented method is efficient than traditional concolic evaluation. In similar year there are many research conducted on testing approach. In august 2012, research presented paper on

performance improvement for test prioritization [3]. In this paper, testing performance extended from single testing to more than one test case combined in one model. After that in December 2012 author Hong Mei et al. [4] proposed a static approach for testing. This concept is specifically for junit test case prioritization. This framework intended to work without need of runtime coverage information and operate through static analysis based on test cases graph. This approach does not need any execution of instrument code and test cases. Sreedevi Sampath et al. [5] proposed approach to order the tests in a composite suite to improve its sustainability against fault. This test suite evaluated in scenarios where constraint specific to time. If evaluation of test cases paused at start from the composite suite, best test case executed. J. Praveen Kumar et al have proposed same concept to prioritize test with single model by combining gui and web applications together. In next research, author Om Kumar et al. proposed an approach to prioritize number of test cases with equal priority.

III. PROPOSED WORK

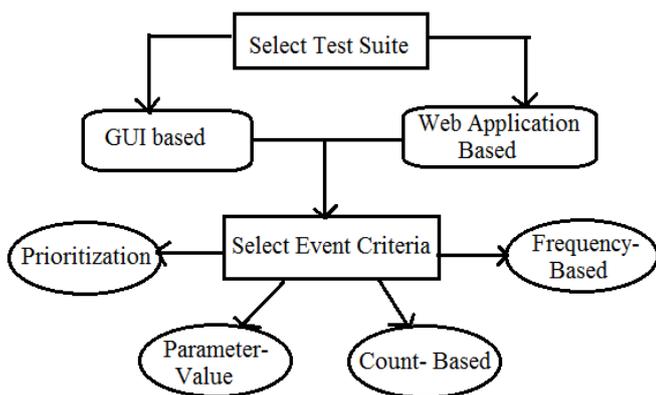


Figure 1: Architecture Flow

In this system, our motive is to design a test prioritization framework for event driven system where we can perform on different test cases in single model. To achieve our goal we have used some criteria's as Prioritization, frequency based, parameter values and count based. By selecting existing among GUI or web frameworks components, we can apply any one criterion on that to perform testing. When particular criteria has executed, the appeared window will show sequence of prioritized test cases. All this depicted in next section.

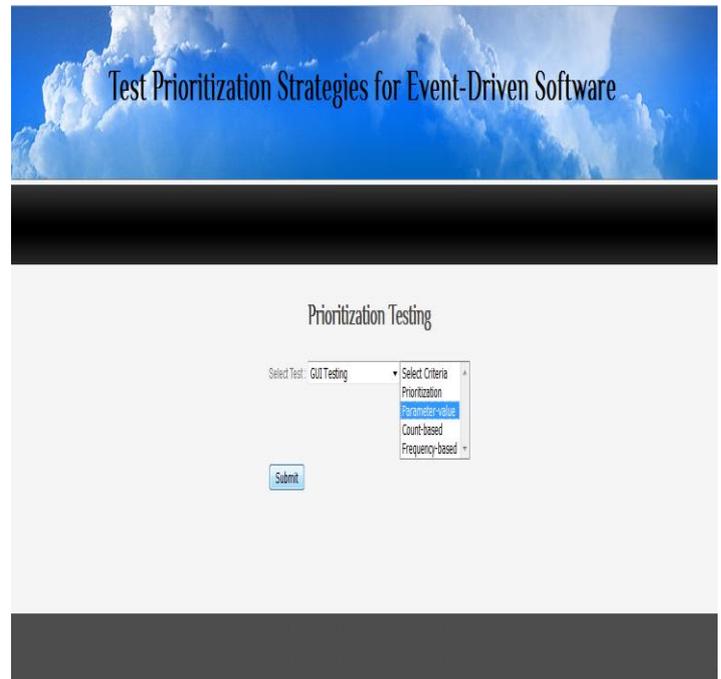
IV. RESULT ANALYSIS

In this project, our approach has been base on GUI and Web frameworks test suite according that we have set some events criteria. Event criteria used to perform testing on test cases specific to GUI or web application.

For result analysis, we have used two existing test suit GUI and web based framework. When home page appears or

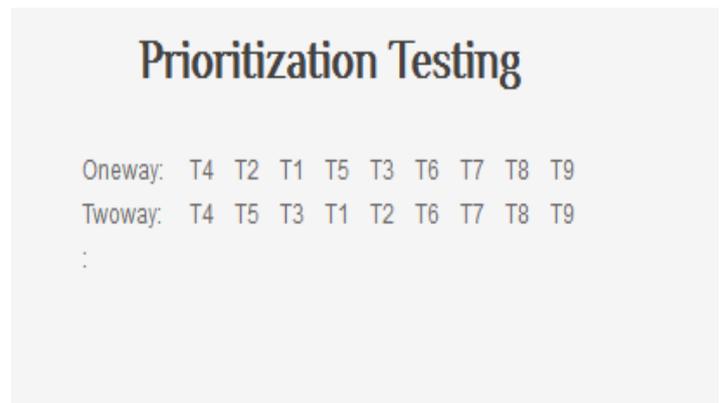
we enter in to system, we need to select required framework first. Based on selected framework some events will occur, these events will prioritize test cases according its type. Reminder of this section showing details of system view one by one.

1. GUI based



This is home page where among list select test of web application and GUI, GUI testing has been selected.

a. Parameter value



One way: The 1-way criterion selects a next test to maximize the number of parameter-values that do not appear in previously selected tests.

Two way: The 2-way criterion selects a next test to maximize the number of 2-way parameter-value interactions between windows.

b. Count based

Prioritization Testing

UniqueWindow:	T3	T4	T5	T1	T2	T6	T7	T8	T9
Action-LtoS:	T4	T2	T1	T5	T3	T6	T7	T8	T9
Action-StoL:	T7	T8	T9	T6	T3	T1	T5	T2	T4

Unique Window: In this criterion, we prioritize tests by giving preference to test cases that cover the most unique windows that previous tests have not covered.

Action count based: an action is a sequence that sets one or more parameter-values in a single window. The prioritization includes selecting the test cases with preference given to those that include the most number of actions.

Action-StoL gives priority to test cases with the smallest number of actions.

Action-LtoS gives priority to test cases with the largest number of actions.

Frequency Based

Prioritization Testing

Most-frequently Present Sequence:	T1	T2	T3	T5	T4	T6	T7	T8	T9
All present Sequence of Window(APS):	T1	T2	T3	T5	T4	T6	T7	T8	T9
Weighted Frequency:	T3	T4	T5	T1	T2	T6	T7	T8	T9

Most-frequently present sequence of windows(MFPS): It identify the most frequently present sequence of windows, in the test suite and order test cases in decreasing order of the number of times that window appears in the test case.

APS: the frequency of occurrence of all sequences is used to order the test suite. For each sequence, s_i , in the application, beginning with the most frequently present sequence, test cases that have maximum occurrences of these sequences are selected for execution before other test cases in the test suite.

Weight Frequency: the weight of a sequence of windows is measured by the number of times the sequence appears in the suite.

This all results showing test prioritization regarding GUI.

2. Web Frameworks based

a. Parameter value

Prioritization Testing

Oneyway:	T3	T4	T6	T2	T1	T5
Twoway:	T3	T4	T2	T5	T1	T6
:						

b. Count based

Prioritization Testing

UniqueWindow:	T2	T3	T4	T5	T6	T1
Action-LtoS:	T3	T4	T2	T6	T1	T5
Action-StoL:	T1	T5	T6	T2	T3	T4

c. Frequency based

Prioritization Testing

Most-frequently Present Sequence:	T1	T2	T3	T4	T5	T6
All present Sequence of Window(APS):	T1	T2	T3	T4	T5	T6
Weighted Frequency:	T2	T3	T4	T5	T1	T6

Web based criteria is same as GUI based application.

All this result shows web frameworks specific test case. This results shows that there is difference between sequences of test cases for both testing approach. In GUI, testing parameter value gives priority to T4 where in web application T3 have priority. Therefore, we can see we both approaches working perfectly in combine as model.

V. CONCLUSION

In this paper, we discussed an approach to test stand-alone GUI and Web-based applications based on shared prioritization function, and prioritization criteria's. Our design is generic enough to study develop and test a unified theory for all kinds of Event Driven Software. Result analysis show that according to selected events test cases are shuffled specific (GUI or Web) for test suite.

REFERENCES

- [1] Arne-Michael Torsel, Fachhochschule Stralsund, Zur Schwedenschanze, "Automated Test Case Generation for Web Applications from a Domain Specific Model", 2011 35th IEEE Annual Computer Software and Applications Conference Workshops, pp. 137-142
- [2] Saswat Anand, Mayur Naik, Hongseok Yang, Mary Jean Harrold, "Automated Concolic Testing of Smartphone Apps", 2012, pp.1-15
- [3] Polani Sri Gnana Kiran¹, M.R.Rajaramesh, "Improving the Performance of a Single Model and Test Prioritization Strategy for Event Driven Software ", International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 6, August 2012 ,pp. 40-44
- [4] Hong Mei, Dan Hao, Lingming Zhang, Lu Zhang, Ji Zhou, and Gregg Rothermel "A Static Approach to Prioritizing JUnit Test Cases", IEEE Computer Society Transactions On Software Engineering, Vol. 38, No. 6, November/December 2012, pp. 1258-1275.
- [5] Sreedevi Sampath, Renee C. Bryce b, "Improving the effectiveness of test suite reduction for user-session-based testing of web applications", Elsevier, Information and Software Technology 54 (2012), pp.724-738
- [6] J. Praveen Kumar, Manas Kumar Yogi, " A Survey on Models and Test strategies for Event-Driven Software", International Journal Of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 4 , august 2012, pp 1087-1091
- [7] Om Kumar C.U., P. Bhargavi, Vinod Kumar. K, "A Single Model for Event-Driven Software", ISSN (Print) : 2319 – 2526, Volume-2, Issue-2, 2013, pp.31-36