Test Case PrioritizationUsing Fuzzy Logic for GUI based Software

Neha Chaudhary,
IT Department,
JSS Academy of Technical Education, Noida, India

Om Prakash Sangwan,
School of ICT, Gautam Buddha University,
Greater Noida, India

Yogesh Singh,
Vice-Chancellor,
MS University, Baroda,
Gujarat, India

***Abstract*-Testing of GUI (Graphical User Interface) applications has many challenges due to its event driven nature and infinite input domain. It is very difficult for any programmer to test for each and every possible input. When test cases are generated using automated testing tool it uses each and every possible combination to generate test cases hence generates numerous number of test case for any GUI based application. Within a defined time frame it is not possible to test every test case, that is why test cases prioritization is required. Test-case prioritization has been widely proposed and used in recent years as it can improve the rate of fault detection during the testing phase. Very few methods are defined for GUI Test case prioritization that usually consider single criteria for assigning priority for the test case which is not sufficient for the consideration of that test case as more fault revealing. In this paper we have proposed a method for assigning weight value on the basis of multiple factors as one of the criteria for test case prioritization for GUI based software. These factors are: The type of event, Event Interaction, and Parameter-value interaction coverage-based criteria. In the proposed approach priority is assigned based upon these factors using fuzzy logic model. Experimental results indicate that the proposed model is suitable for prioritizing the test cases of GUI based software.**

Keywords-Graphical user Interface; Prioritization; Test Suite; Fuzzy Model.

I. INTRODUCTION

Testing is widely recognized as a key quality assurance (QA) activity in the software development process. Research in testing has received considerable attention in the last two decades [2,8,20,14]. Testing of graphical user interfaces (GUIs) was a neglected research area till last decade [4]. Graphical User Interface (GUI) constitutes as much as 45-60% of the total software code in any software, so testing of GUI is a very important concern [3,16]. Most of the test case generation techniques require human involvement and are resource intensive. Many automated approaches were proposed for test case generation but in practice capture replay tools are used [17]. So generation of test cases is a costly effort. Rapid prototyping model is followed for GUI development which involves continuous modifications in software versions [1]. Due to event driven nature of GUI it takes sequence of events as input and after change of state generates new sequence of input as output [6, 12, 19]. For different set of state and combination of inputs GUI generate different output [5, 20]. It would be difficult to manage all the combinations for testing as number of combination grows exponentially with the number of events. Running all GUI test cases and then fixing all bugs may be time consuming and delaying the project completion. This would require that the test developed for one version should be reusable across various versions [1, 4]. It is important to prioritize the test cases that uncover the most faults as fast as possible in the testing process. So prioritization of test suite is a challenging area [7, 9,13,18]. In this paper multiple factors are considered for the assignment of weight value for test suite.

This paper is organized as follows: Section II describes the research background for the proposed work. Section III describes the factors affecting the fault detection capability of test suite. Section IV introduces the concept of the proposed fuzzy model. Section V discusses about the experimental design. The results are displayed in section VI and conclusion and future work is presented in section VII.

II. RESEARCH BACKGROUND

The significant work is done by Renee C. Bryce and Atif M. Memon for test suite prioritization by interaction coverage. Test suite for GUI based program is prioritized by t-way interaction coverage and rate of fault detection is compared with the fault detection by other prioritization criteria [9]. Experimental results shows that test suits with the highest event interaction coverage benefit the most and test suits that has less interaction coverage does not benefit in using this prioritization technique.

In this approach only event interaction coverage criterion is taken as a measure for prioritizing test cases, there could possibly be significant effect of type of event in a test case, which will affect the rate of fault detection.

Atif M. Memon & Renee C Bryce provided a single abstract model for GUI and web application testing. In this approach test cases are prioritized by set of count based criteria, set of usage-based frequency and set of interaction based criteria [10]. The results show that test case prioritization by 2-way (interaction based criteria) and PV-LtoS (Parameter count based criteria) provided best improvement in the rate of fault detection for GUI based software. The main drawback of this technique is that the combination of different prioritization criteria is used and it is said that this is more effective than a single criterion.

However in order to cover web applications and GUI applications various factors need to be added and they add complexity to the process which can be avoided if specific criteria for web based application and GUI based application would be used.

In the work done by Chin-Yu Huang et al. on GUI Test case prioritization, weighted event flow graph was used for solving the non-weighted GUI test cases and ranked GUI test cases based on weight scores. In order to assign weights, events are classified based on their importance in the GUI application [11, 15]. In this technique weight summation of termination event and unrestricted focus event is equal to that of restricted focus event which requires further research in this area. In this approach the effect on fault detection based on event interaction with other event need to be explored further. Weight value of each interaction would also have impact on fault detection ability of test cases which was not considered in this approach.

III. FACTORS FOR TEST CASE PRIORITIZATION

Weight value will be assigned by considering following factors:

- Type of event
- Event Interaction
- Count based criteria

In following section we will elaborate different criteria considered for assigning weight values:

- Type of event

Type of event, a test suite is covering has significant impact on the fault revealing capability of test case.

According to the literature survey events are classified as following five types, restricted-focus event, unrestricted-focus event, termination event, menu-open event, and system-interaction event. Event weight has been assigned on the basis of importance of specific type of events [15]. This categorization of events is given in table 1.

TABLE 1: EVENT WEIGHT ASSIGNMENTS

Event type	Weight Value
Restricted-focus event	5
System-interaction event	4
Termination event	3
Menu-open event	2
Unrestricted-focus event	1

- Event Interaction

In event driven software event interaction makes the program to follow a different execution path that may reveal faults in the system. In our proposed method Priority is assigned to those test cases which have large number of parameter value interaction [9].

- Count-based criteria

Since the GUI is the collection of events, number of actions performed with events, set of parameters and number of windows. It is very important that test suit that provides maximum count coverage should be given higher importance then the test suit that provide low coverage. So another factor that will be considered is the count of number of windows, actions or parameter values that a test case may cover.

IV. PROPOSED FUZZY MODEL

Fuzzy logic is a convenient way to map an input space to output space. In this paper we have proposed a fuzzy model with three inputs, namely Type of event, Event Interaction, Count based criteria. Figure1 shows the fuzzy model. The proposed model consists of three inputs and provides a crisp value of priority using Rule Base.

Fuzzy Inference System (FIS) is the process of formulating the mapping from a given input to an output using fuzzy logic. This will use Mamdani's fuzzy inference method which is most commonly seen fuzzy methodology as shown in Figure 2.

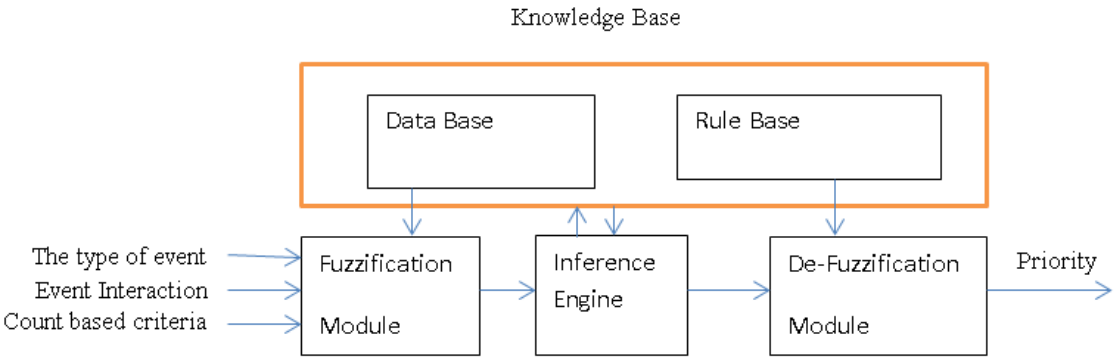

Figure 1: Fuzzy Model for Prioritization

After the fuzzification process, there is a fuzzy set for each output variable that needs defuzzification. The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is singleton number. Further centroid method will be used for defuzzification. Centroid method will return the centre of area under the curve.

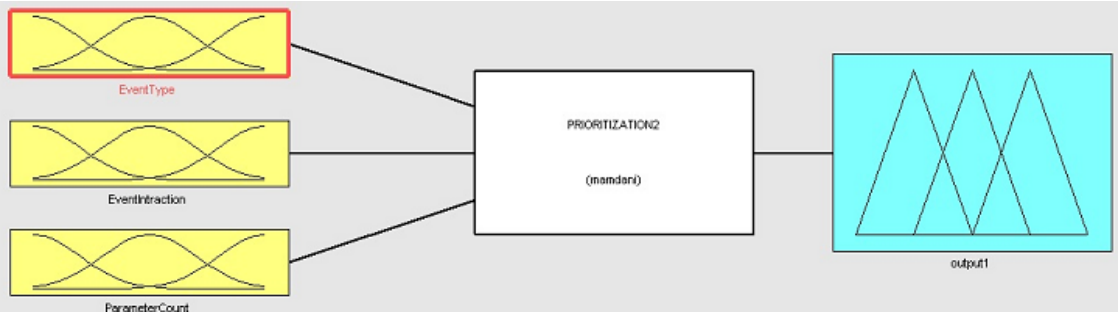

Figure 2: Fuzzy Inference System: Priority Model

V. EXPERIMENTAL DESIGN

In order to fuzzify the inputs, we have selected following membership functions for the Type of event, Event Interaction and Count based criteria and they are shown in figure 3-5. GUI events are classified into five categories and they have different fault revealing capabilities. Weight value of test case will be calculated by taking summation of weight according to categorization. That weight will be divided into five states (linguistic variables) i.e. very low, low, medium, high and very high as shown in figure 3. The input variable Event Interaction has been divided into five levels i.e. very low, low, medium, high and very high as shown in figure 4.

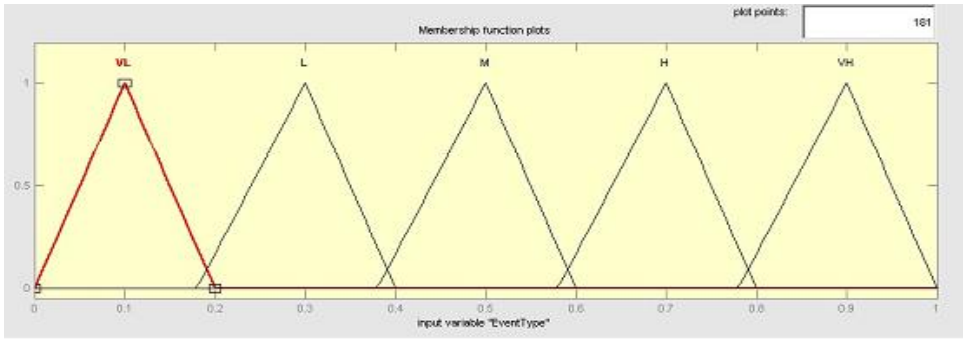

Figure 3: Fuzzification of Input Variable Event Type

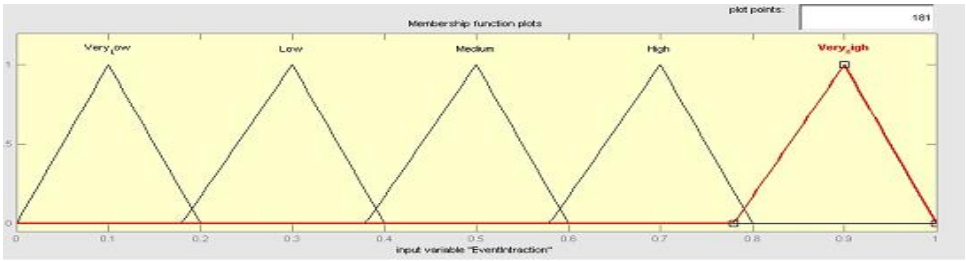

Figure 4: Fuzzification of Input Variable Event Interaction

Similarly the input variable count has been divided into five states i.e. very low, low, medium, high and very high as shown in figure 5.

The output variable priority is classified as very low, low, medium, high and very high. Similarly priority has five membership functions as shown in figure 6:

A. Rule Base and Evaluation Process

When input data is fuzzified, processing is carried out in fuzzy domain. The model integrates the effects of multiple factors type of event, Event Interaction and Count based criteria into a single measurable parameter that will define the priority of test case, based on the following knowledge/rule base. The rule base can further be advanced by creating more ranges (fuzzy sets) for the input variables. All inputs and outputs are fuzzified as shown in figure 3 to 6. All possible combinations of inputs were considered that will create 5^3 i.e. 125 sets. The priority for all 125 combinations is classified as very low, low, medium, high & very high by expert judgment. This indicates to formulation of 125 rules for the fuzzy model and some of the rules are presented below:

1. If value assigned for Type of event is low, Event Interaction is low and Count based is low then priority will be low.

2. If value assigned for Type of event is medium, Event Interaction is medium and Count based is medium then priority will be medium.
3. If value assigned for Type of event is low, Event Interaction is high and Count based is high then priority will be medium.
.
.
.

- If value assigned for Type of event high, Event Interaction is medium and Count based is high then priority will be high.

…..
125. If value assigned for Type of event very low, Event Interaction is high and Count based is very low then priority will be low.

All 125 rules are inserted and rule base is created. A rule is fired based on the particular set of inputs. In this model Mamdani style inference has been used.

The output of test case priority has been observed using rule viewer for particular set of inputs using MATLAB fuzzy Tool Box as shown in figure 7.

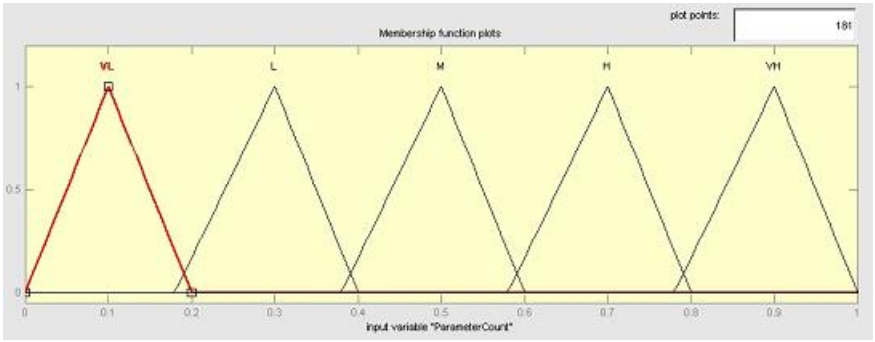

Figure 5: Fuzzification of Input Variable Count

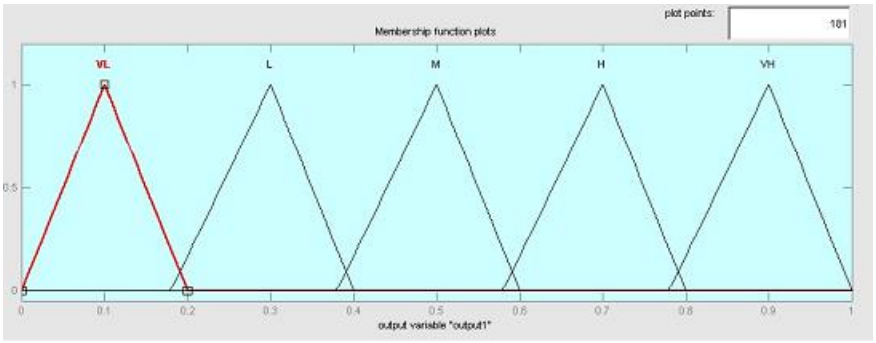

Figure 6: Fuzzification of Input Variable Priority

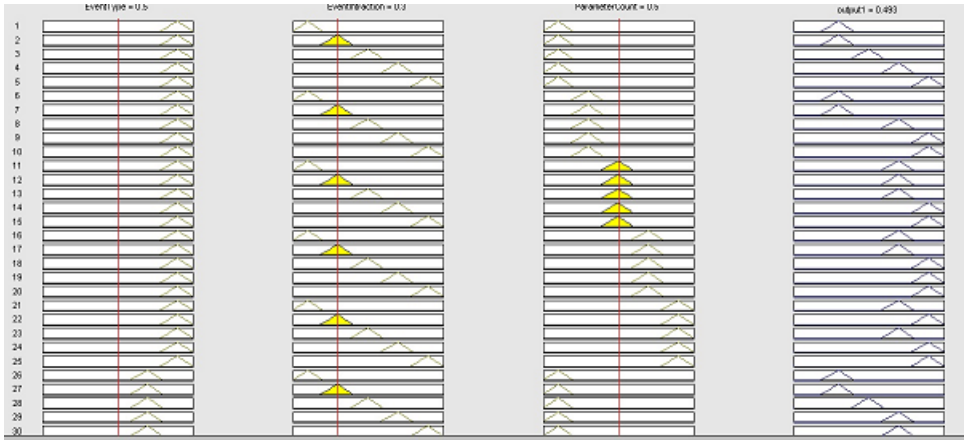

Figure 7: Rule Viewer for the Priority Model

VI. EXPERIMENTAL RESULTS

For example we have following crisp value inputs to the model: type of event =0.5, Event Interaction =0.3 and Count based criteria=0.5.

These inputs are provided for the fuzzification module and after fuzzification of given value we find the type of event =0.5 belongs to the fuzzy set low with membership grade 0.9 and belongs to fuzzy set medium with membership grade 0.9 and with high it has membership grade 0.72.

For event Interaction =0.3 belongs to the fuzzy set low with membership grade 0.9 and belongs to fuzzy set medium with membership grade 0.72 and with high it has membership grade 0.72. Count based criteria=0.5 belongs to the fuzzy set low with membership grade 0.9 and belongs to fuzzy set medium with membership grade 0.9 and with high it has membership grade 0.72. With these input values we find that rules given in table 2 will be considered:

First rule assigns the priority low to an extent of 0.9 and second rule gives priority medium to an extent of 0.72 and the third rule gives priority high to an extent 0.72 this is shown in the figure 8.

TABLE 2: TEST SUITE PRIORITY CALCULATION FOR A GIVEN INPUT SET

The type of event (.5)	Event Interaction(.3)	Count based criteria(.5)	Priority	Membership Grade for Test Case Priority
Low	Low	Low	Low	Min(0.9,0.9,0.9)=0.9
Medium	Medium	Medium	Medium	Min(0.9,0.72,0.9)=0.72
High	Medium	High	High	Min(0.72,0.72,0.72) =0.72

A. Defuzzification

After getting the fuzzified output as specified in previous section, we defuzzify them to get the crisp value of the output variable priority [21]. Transformation of the output from fuzzy domain to crisp domain is called defuzzification. In this model we defuzzify using centre of gravity (COG) method of the aggregate output 1, 2 and 20. X axis centroid points for all three variables are 2.9, 4.9 & 6.9 the final value for GOG is 4.84.

The effect of these rules is also observed by simulating the model using fuzzy logic tool box of MATLAB. The priority for the given input values comes out to be 0.493 which is the same as calculated from COG method.

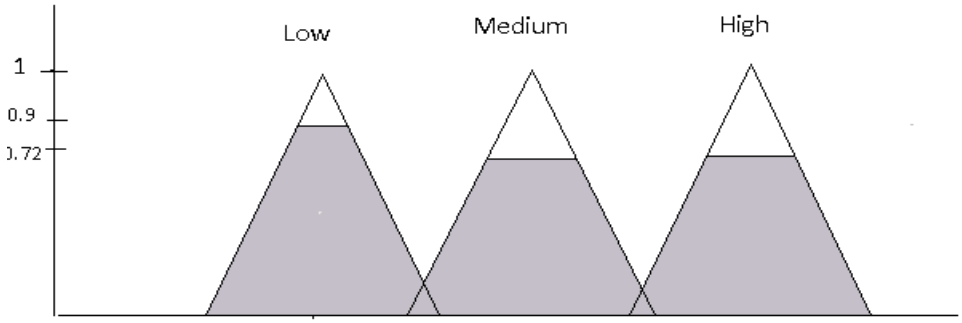

Figure 8: Output computation for Test Case Priority

VII. CONCLUSION

The problems of test-case prioritization have been explored in this paper to improve the rate of fault detection effectiveness for GUI based software. We have proposed a fuzzy based technique to assign priority of test case. Priority of test case will be assigned as very low, low, medium, high and very high. In this technique three factors namely Type of event, Event Interaction, Count based criteria are considered to assign weight values for test cases. Impact of these factors are categorize in five categories as very low, low, medium, high and very high. Experimental results shows that the proposed fuzzy model is proved to be an effective approach for test case prioritization for GUI based software.

REFERENCES

[1] A. M. Atif, "Automatically repairing event sequence based GUI test suites for regression testing", ACM Transaction on Software Engineering and Method. Volume 18, Issue 2, Nov. 2008.

[2] M. J. Harrold. "Testing: a roadmap". In ICSE '00: Proceedings of the Conference on The Future of Software Engineering, ACM Press, New York, NY, USA, pages 61.72, 2000.

[3] B. A. Myers. User interface software tools. ACM Transactions on Computer-Human Interaction,1995.

[4] A. M. Memon. A Comprehensive framework for testing graphical user Interfaces. Ph.D. thesis, Department of Computer Science, University of Pittsburgh, July 2001.

[5] L. White and H. Almezen, "Generating test cases for GUI responsibilities using complete interaction sequences". In Proceedings of the International Symposium on Software Reliability Engineering, pages 110-121, Oct. 2000.

[6] R. K. Shehady and D. P. Siewiorek, "A method to automate user interface testing using variable finite state machines." In Proceedings of The Twenty-Seventh Annual International Symposium on Fault-Tolerant Computing (FTCS'97), Washington - Brussels - Tokyo, pages 80.88, June 1997.

[7] Elbaum S., Malishevsky A. G., Rothermel G.,"Test case prioritization: a family of empirical studies", IEEE Transactions on Software Engineering vol. 28 (2), pp. 159–182, 2002.

[8] A. M. Atif ,Mary Lou Soffa, Martha E. Pollack," Coverage criteria for GUI testing" Proceedings of the 8th European Software Engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of Software Engineering, pp. 256-267, 2001.

[9] Renee C. Bryce, Atif M Memon, "Test suite prioritization by interaction coverage" Domain-Specific Approaches to Software Test Automation Workshop September, 2007.

[10] Renee C. Bryce, Sreedevi Sampath, Atif M Memon, "Developing a single model and test prioritization station for event- driven software" IEEE Transaction on Software Engineering, Jan 2010

[11] Chin-Yu Huang, Jun-Ru Chang, Yung-Hsin Chang, "Design and analysis of GUI test-case prioritization using weight-based methods" The journal of Systems and Software 83, pp 646-659, 2010

[12] A. M. Atif ,Mary Lou Soffa, Martha E. Pollack," coverage criteria for GUI testing" Proceedings of the 8th European Software Engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of Software Engineering, pp. 256-267, 2001.

[13] Scott McMaster, Atif M. Memon,"Call-Stack coverage for GUI test suite reduction" IEEE Transaction on Software Engineering, Volume 34, Jan/Feb, 2008

[14] Jaymie Strecker , Atif M Memon ,"Relationships between test suites, faults, and fault detection in GUI testing" In ICST '08 Proceedings of the First international conference on Software Testing, Verification, and Validation, (Washington, DC, USA), 2008.

[15] Paul Gerrard, "Testing GUI applications", EuroSTAR, Edinburgh UK, 1997.

[16] A. Memon, A. Nagarajan, and Q. Xie, "Automating regression testing for evolving GUI software," J. Software Maintenance and Evolution: Research and Practice, Volume 17, no. 1,pp. 27-64, 2005.

[17] Atif M. Memon , Qing Xie "Studying the fault-detection effectiveness of GUI test cases for rapidly evolving software" IEEE Transaction on Software Engineering, Volume 31, no. 10, pp. 884-896, Oct. 2005

[18] S. McMaster and A. Memon, "Call Stack Coverage for GUI test-suite reduction", Proc., 17th International Symposium on Software Reliability Engineering, Nov. 2006.

[19] A. M. Memon, M. E. Pollack, and M. L. Soffa, "Hierarchical GUI test case generation using automated planning", IEEE Transactions on Software Engineering, pp 144–155, Feb. 2001.

[20] Daniel R. Hackner , Atif M. Memon ,"Test case generator for GUITAR" , International Conference on Software Engineering, (Washington, DC, USA), 2008.

[21] Yogesh Singh, Pradeep Kumar Bhatia, Omprakash Sangwan,"Predicting software maintenance using fuzzy model", published in SIGSOFT Software Engineering Notes, vol 34, July 2009.