

# Assessment and Comparison of Fuzzy Based Test Suite Prioritization Method for GUI Based Software

Neha Chaudhary

Ph.D. Scholar, Gautam Buddha University,  
Greater Noida, India

O.P. Sangwan

Guru Jambheshwer University of Science & Technology,  
Hisar, India

**Abstract**—The testing of event driven software has significant role to improve overall quality of software. Due to event driven nature of GUI based software many test cases are generated and it is difficult to identify test cases whose fault revealing capability is high. To identify those test cases test suite prioritization is done. Various test suite prioritization methods exists for GUI based software in literature. Prioritization methods improve the rate of fault detection. In our previous work we have proposed a fuzzy model for test suite prioritization of GUI based software. In this method priority is assigned on the basis of multiple factors using fuzzy model. These factors are: The type of event, Event Interaction, and Parameter-value interaction coverage-based criteria. Using this method a test oracle will be organized in the descending order of its effectiveness. In this paper we are evaluating the proposed fuzzy model and comparing results with other prioritization methods.

**Keywords**—Test suite prioritization; Fuzzy Model; Comparison of prioritization methods

## I. INTRODUCTION

Testing is a quality assurance activity [10]. Testing of GUI impose many challenges due to its event driven nature and interaction with the user [9]. Event driven software takes sequence of events as input after change of state generate new sequence of input as output [1, 8]. Many test cases are generated to cover combination of all events. To improve the effectiveness these test cases generated for event driven software a prioritization method are developed. The aim of this prioritization is to reorder the test cases according to fault revealing capability of test cases.

This paper is organized as follows: Section II describes the research background for the proposed work. Section III describes Fuzzy model. Section IV discuss about evaluation of fuzzy based test case prioritization method. Section V covers assessment of effectiveness of test suite. Conclusion is presented in section VII.

## II. RESEARCH BACKGROUND

The problem of test suite prioritization is widely explored by many researchers. Few researchers have significant contribution in this area. One imperative work is proposed by Memon Atif M, Bryce Renee C. (2007). In this work test cases are prioritized using event interaction coverage. Rate of fault detection improves for the test suits which have the larger number of 2 way and 3 way interaction. Huang Chin-Yu et al. (2010) in their work proposed three weight based method. These methods are Equal weight, Fault prone weight and Random weight. They have classified events and assign weight value based on their importance. Results shows that high to low Adjusted weight method find more fault. Another work by Bryce Renee C. et al. (2011) proposed cost-based combinatorial interaction coverage metric as 2way interaction coverage and cost-based 2way interaction coverage. According to experimental results the difference in APFDC between 2way and cost-based 2way for CPM was less than 3%. Wei Sun et al. proposed a multi-objective test case prioritization strategy is proposed to combine two objectives: statement coverage and event coverage for GUI applications. Experiment shows that multi-objective has better performance than both single-objective strategies.

## III. FUZZY MODEL

This model was proposed in our previous work [5]. In this model multiple factors are considered for prioritization. Major components of GUI are identified as prioritization factors. These factors are: The type of event, Event Interaction, and Parameter-value interaction coverage-based criteria. A fuzzy model is developed for test suite prioritization which considers these factors as input and assigns weight value to each test case. These factors are categorized as very low, low, medium, high and very high. Impact of these factors are categorize in five categories as very low, low, medium, high and very high. According to this impact priority is assigned. Random tiebreaking is used when weight value comes same for multiple test cases. Total 125 rules are generated for fuzzy model. Centre of gravity (COG) is used for defuzzification.

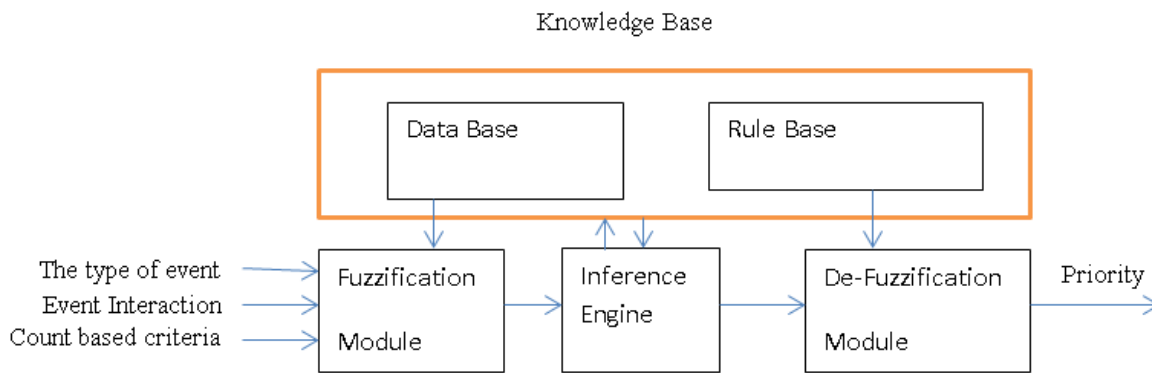


Fig. 1. Fuzzy Model for Prioritization [5]

A. Evaluation of fuzzy based test case prioritization method

For the evaluation of fuzzy based test case prioritization method application Terppaint 3.0 and some of its L1 and L2 test cases are considered. For each test case its parameter count, event sum and number of event Interaction covered are counted for the input of fuzzy model. These values are shown in table 1.

TABLE I. TERPPAINT 3.0 TEST CASES

Test Case	Parameter count	Event Sum	Event Interaction
T1	57	6	1
T2	95	14	1
T3	94	13	1
T4	112	16	1
T5	74	12	1
T6	58	10	1
T7	76	12	1
T8	75	12	1
T9	36	8	2
T10	77	5	2

A parser is developed to calculate the Parameter Count Event Sum and Event Intraction. These values are given as input to the fuzzy model and output value is generated. Output of fuzzy model is shown in table II as FIS output.

On the basis of output of fuzzy model priority is assigned to each test case. Maximum priority is assigned to those test cases

whose FIS output is highest and lowest priority is assigned to those test cases whose FIS output is lowest.

TABLE II. APFD FOR TERPPAINT 3.0

Test Case	Parameter count	Event Sum	Event Interaction	FIS Output	Priority
T1	57	6	1	0.22	10
T2	95	14	1	0.82	1
T3	94	13	1	0.811	2
T4	112	16	1	0.487	6
T5	74	12	1	0.565	4
T6	58	10	1	0.366	8
T7	76	12	1	0.383	7
T8	75	12	1	0.577	3
T9	36	8	2	0.232	9
T10	77	5	2	0.535	5

Once priority is assigned to test cases evaluation of prioritized sequence is done. In order to evaluate prioritized sequence faults identified by each test case in previous run is considered.

With the count of number of faults identified by the test case and its priority assigned by the fuzzy model its rate of fault detection has been calculated as shown in table 3.

TABLE III. PRIORITY TABLE FOR TEST CASES OF TERPPAINT 3.0

Test Case	Parameter count	Event Sum	Event Interaction	FIS Output	Priority	APFD Calculation
T1	57	6	1	0.22	10	10
T2	95	14	1	0.82	1	2
T3	94	13	1	0.811	2	6
T4	112	16	1	0.487	6	18
T5	74	12	1	0.565	4	12
T6	58	10	1	0.366	8	24

T7	76	12	1	0.383	7	42
T8	75	12	1	0.577	3	12
T9	36	8	2	0.232	9	45
T10	77	5	2	0.535	5	40

B. Assessment of effectiveness of test suite:

Average Percentage of Fault Detection (APFD) metric is considered for the assessment of effectiveness of test suite. This metric calculates rate of fault detection. Higher value of this metric represents effective prioritized sequence. APFD is defined as (Rothermel et al., 2001):

$$APFD = 1 - \frac{TF_1 + TF_2 + \dots + TF_m}{nm} + \frac{1}{2n} \quad (1)$$

Where  $TF_1$ : number of faults identified by first test case

$n$ : total number of test cases under consideration

$m$ : total number of faults identified by test cases

Final APFD value for TerpPaint 3.0 is 0.49. These results are further compared with APFD of Parameter count method and event weight method and results are shown in table 5.

TABLE IV. COMPARISON WITH OTHER METHODS

Test Case	APFD Calculation for Parameter Count		% Event Coverage	Event Weight	APFD Calculation for Event Weight		APFD for Fuzzy based Prioritization Method	
	Priority	APFD			Priority	APFD	Priority	APFD
T1	9	9.00	0.80	1.61	9	9.00	10	10
T2	2	4.00	2.01	10.04	1	2.00	1	2
T3	3	9.00	2.01	10.04	2	6.00	2	6
T4	1	3.00	1.20	3.61	6	18.00	6	18
T5	7	21.00	1.61	6.43	3	9.00	4	12
T6	8	24.00	1.20	3.61	7	21.00	8	24
T7	5	30.00	1.20	3.61	8	48.00	7	42
T8	6	24.00	1.61	6.43	4	16.00	3	12
T9	10	50.00	0.80	1.61	10	50.00	9	45
T10	4	32.00	1.61	6.43	5	40.00	5	40
	Final APFD	0.51				0.47		0.49

For the comparison of results prioritization is done for the same test suite using various methods as shown in table 5 and their APFD value is calculated for the comparison.

TABLE V. COMPARISON OF VARIOUS PRIORITIZATION METHODS

Prioritization Method	APFD %
Fuzzy Method	0.49
Best order of test cases	0.66
Worst order of test cases	0.31
Event Weight Method	0.47
Random order of test cases	0.36

It is demonstrated in figure 2 that Fuzzy based prioritization method outperform most of the prioritization methods. APFD of Fuzzy based test case prioritization method is very close to best method of parameter count based method. But these results will very when number of parameters will be different in other applications.

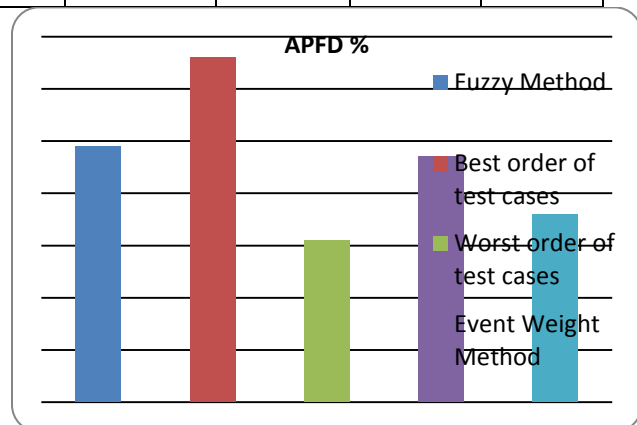


Fig. 2. Comparison of various test case prioritization methods

IV. CONCLUSION

A fuzzy based technique which was proposed in previous research to assign priority of test cases is evaluated in this paper. It is further compared with existing prioritization methods. Experimental results shows that fuzzy based

prioritization method outperform most of the prioritization methods. APFD of Fuzzy based test case prioritization method is very close to best order of test cases. 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] Memon Atif M, Bryce Renee C., "Test Suite Prioritization by Interaction Coverage", In Proceedings ACM SIGSOFT Conference, 2007.
- [2] Huang Chin-Yu, Chang Jun-Ru and Chang Yung-Hsin, "Design and analysis of GUI test-case prioritization using weight-based methods", The journal of Systems and Software vol. 83, 2010, pp. 646-659.
- [3] Bryce Renee C., Sampath Sreedevi, Pedersen Jan B., "Manchester Schuyler, Test suite prioritization by cost-based combinatorial interaction coverage", Published in International Journal of System Assurance Engineering and Management, 2011.
- [4] Wei Sun, Zebao Gao, Weiran Yang, Chunrong Fang, Zhenyu Chen, "Multi-Objective Test Case Prioritization for GUI Applications", Accepted to be published, 2011.
- [5] Neha Chaudhary, O.P.Sangwan, Yogesh Singh, "Test Case Prioritization Using Fuzzy Logic for GUI based Software", International Journal of Advanced Computer Science and Applications, Dec 2012, pp. 222-227.
- [6] Chu Chengyun, Rothermel Gregg, Harrold Mary, Jean and Roland H. Untch, "Test Case Prioritization: An Empirical Study", Published in the Proc. of the International Conference on Software Maintenance, Oxford, UK, 1999.
- [7] Memon Atif, "Automatically Repairing Event Sequence-Based GUI Test Suites for Regression Testing", ACM Transaction on Software Engineering and Method, Volume 18, Issue 2, 2008.
- [8] Hackner R., Daniel, Memon Atif, "Test Case Generator for GUITAR, International Conference on Software Engineering", 2008.
- [9] Memon Atif, Soffa Lou Mary, Martha E. Pollack, "Coverage Criteria for GUI Testing", Proc. of the 8th European Software Engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of Software Engineering, 2001, pp. 256-267.
- [10] Gerrard Paul, "Testing GUI Applications", EuroSTAR, Edinburgh UK. 1997.