International Journal for Modern Trends in Science and Technology
Volume: 03, Issue No: 01, January 2017
ISSN: 2455-3778
http://www.ijmtst.com

Allocation of Test cases using Severity and Data Interactivity

Pankaj Dalal[1] | Dr.Asheesh Shah[2]

[1]Research Scholar, Mewar University, Chittorgarh, Rajasthan. India.
[2]Pro Prisedent, Mewar University, Chittorgarh, Rajasthan. India.

To Cite this Article
Pankaj Dalal and Dr.Asheesh Shah, "Allocation of Test cases using Severity and Data Interactivity", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Issue 01, 2017, pp. 67-73.

ABSTRACT

Software quality is prime concern of software development among all stack holders. The Reliability is most concern matrix among quality matrixes for all software application. The reliability is measured by fault occurrence during software testing phase. The allocations of test cases as well as basis of allocation are key component of qualitative usage. The software testing by randomly selected test cases and software operations for performing testing is common in practice. The pervious researcher identified this problem and provided various solution of allocation, and basis of allocation, the basis of allocation using the Software Operational Profile (SOP) is one of them. The researchers have used the tradition SOP which has many demerits. Therefore the quality of software could not be assured or predicted. Here we will talk about another dimension to be added in allocation of test cases.

KEYWORDS: *Use case Model, Use case Operational profile, Data interactivity. Severity.*

I. Introduction

The random testing not tested some inputs, some of which may be standard and more important to the system's operation. The other factors are severity and number of interactive data in a function. Therefore many times sever functions, function with more data or more usage occurrence functions are left out from testing. At the same time test case allocated are the functions also uneven. The testing with all functions with all input is practically not possible. The researchers have given model to overcome such demerits of random testing. The testing using SOP. The functions are prioritized with occurrence usage and further classified with severity. The Musa has classification with usage probability where Leung and Wong included severity with usage probability test case allocation but both has use traditional SOP.

Operational profile is a quantitative characterization of a system usage, plays an essential and important role in reliability improvement through testing. Operational profile simply describes over all operations and their probabilities of occurrence. It focuses on functional requirement of the product and the relative importance of different uses. The traditional SOP has five-step procedure as (1) identification of the customer profile, (2) establishing the user profile, (3) defining the system-mode profile, (4) determination of the functional profile, and (5) determination of the operational profile itself has defined by Musa[1].

The traditional approaches for the development of SOP consider software usage data as crisp probability estimates. The quantitative usage data may be collected for the development of SOP from one or more combinations of different sources such as, (1) Existing field record data obtained from previous release or similar system, (2) Data obtained from system logs or marketing professionals, (3) Information acquired from software requirements documents and (4) Expert's opinion etc. [1].

There many difficulties in estimating the operational profile. An operational profile is an estimate of the relative inputs frequency of various usages. The frequency derived by user dependent usage analysis. There are situations under which deriving an accurate Operational profile may not be possible. The practitioners are often encountered problems are:

1. The new software may not have any customer base for the system and rely on guesstimates of occurrence probabilities for various features.

2. The new features exists a user base for the existing version of the system and rely on guesstimates of the occurrence probabilities of the new features.

3. The feature granularity may be more or less lines per feature than the average, may more features have more than average lines of code hence resulting in lesser lines of code per feature.

There are also problems in usage data collected from sources mention above, firstly the information from existing record of field data or from system logs or from marketing professionals may have high variations of software usage, secondly the sources like existing record of field data or the system log data considers software usage as point estimates.

A qualitative analysis may be useful to obtain the occurrence possibility estimates from expert opinion [7] [8], There is no evidence of step by step approach for the development of SOP through actor of the system present in the literature. The other some common problems observed with the traditional SOP development models are: 1. requires prediction of probability by expert's opinion, are difficult to quantify and selecting appropriate distribution through statistical methods. 2. SOP may not render the real situation for large systems in case of insufficient data. This discussion leads to an inaccurate operational profile. Hence the estimation of operational profiles is a difficult and error-prone task.

The test cases to each operation are designed based on their occurrence probability [2]. In the development stage of a software product, its reliability can be estimated by analyzing data obtained from software tests. The test case allotment can based using operational profile [3]. The testing using the accurate operational profile leads to more effective to allocation of test cases. Therefore accurate Software operational profile is needed for testing software.

II. Use Case Based SOP Model

The operational profiles is a "customer or user-centred" approach. The use case also is a "customer or user-centred" agreement between all stack holder of the system. Hence use case based operational profile facilitates to prepare the test case for the improved testing over the randomly selected test case testing. The operational profile is basically the probable operation perform by the user. There may be one or more user of the system [6].

Here we take example of USOP [6] to illustrate over work. The library member participates in the use cases are registration for new member, login and logout, query books availability, raise the request for book, check the status of a request, cancels the request of book, and pay the subscription or fine . The final table shown as below [6]:

III. Use Case Based Software Operational Profile

The five step procedure prescribed by Musa requires:

1. Customer profile – There is one general customer library.
2. User profile- Here is the two users of the system.
3. System modes – A single mode operation is treated as it may have web or client server architecture.
4. The table 4 shows the functions of the actor-Member.
   The functions are the use cases and these functions are in path of execution of on session.
5. The table 1 shows the operations of the each

The possible error is classified and assumes to be of equal chances of each error occurrence in the table 1.

Table 1: Functional & Operational profile

Functions (Use case ,	No. in	Probability of usage	Path Type	Total Operation	Probability of usage
Query	9	0.2	T	Request Matched	0.1
			A	Request query not Matched	0.05
			A	Request for purchase	0.05
Request	5	0.11	T	Book Request registered	0.055
			A	Requested Book Available	0.0275
			A	Requested Book Not Available	0.0275
Status	5	0.11	T	Requested Book Available	0.055
			A	Requested Book Not Available	0.0275
			A	Request Not Registered	0.0275
Cancel	6	0.13	T	Cancel Request Successful	0.065
			A	Request Not Registered	0.065
Deposit	7	0.15	T	Deposit Book Successful	0.075
			A	Delay Deposit	0.01875
			A	Book Lost	0.01875
			A	Book ID not in Member Account	0.01875
			A	Book Damaged	0.01875
Issue	7	0.15	T	Book Issue	0.075
			A	Not a Valid Member	0.0375
			A	No of Book Exceed Max Limit	0.0375
Fine	7	0.15	T	Deposit amount Successful	0.075
			A	Deposit amount Not Successful	0.0375
			A	Requested Book Not Available	0.0375
Valid Path 17		Valid Events 46			

IV. Methodology

1 Identifying the Severity:

The some of the Software are develop for high risk applications. If it fails, it proves fatal, some time loss of human life also. The railway, Air trafficking system are the popular risky example of fatal prone applications. Such critical tasks are classified as Catastrophic, severe and normal task, in some case may be classified in two, critical and normal tasks.

2 Weight the Severity:

The Catastrophic severity must weight double then Severe and Normal with lowest weight. The total weight is one then may be distributed 0.6:0.3:0.1 respectively. The further severity is added in the table 1. Functional & Operational profile. The Normal is low severe task. Therefore, normal task is assigned by 0.1 weights.

3 Identify Data interaction:

The data interaction needs to associate with task. The applications require data to process and generate results. There are many task having dense data interaction, hence it is required to associate with operational profile. The task without data interaction has lesser chance of failure then the task with data interaction. Data interaction is error prone. Therefore, it is required the special attention. This data interaction is associated with frequency of occurrence.

4 Weight Data interaction:

The tasks are assigned weight according to the data interaction. The same weight scheme 0.6:0.3:0.1 is adopted for High, Medium, & Low interactivity. The High is interactivity, the more is chances of error, and therefore needs more test cases. The Nil or Low interactivity needs less test case. This further weight scheme can be granule and weight can be very such that if interactivity is slightly lesser then high, the weight could be 0.5 or 0.45. The every task must be tested though having low or nil interactivity. Therefore weight 0.1 must be default weight for low or nil interactivity.

Table 2: Functional & Operational profile with Severity and interactivity

Functions (Use case, Path)	Operations	Probability of usage	Severity	Severity Weight	Interactivity	Interactivity Density
Query	Request Matched	0.1	Normal	0.1	Low	0.1
	Request query not Matched	0.05	Normal	0.1	Low	0.1
	Request for purchase	0.05	Normal	0.1	High	0.6
Request	Book Request registered	0.055	Normal	0.1	Medium	0.3
	Requested Book Available	0.0275	Normal	0.1	Low	0.1
	Requested Book Not Available	0.0275	Normal	0.1	Low	0.1
Status	Requested Book Available	0.055	Normal	0.1	Low	0.1
	Requested Book Not Available	0.0275	Normal	0.1	Low	0.1
	Request Not Registered	0.0275	Normal	0.1	Low	0.1
Cancel	Cancel Request Successful	0.065	Normal	0.1	Low	0.1
	Request Not Registered	0.065	Normal	0.1	Low	0.1
Deposit	Deposit Book Successful	0.075	Normal	0.1	Low	0.1
	Delay Deposit	0.01875	Normal	0.1	Low	0.1
	Book Lost	0.01875	Normal	0.1	Low	0.1
	Book ID not in Member Account	0.01875	Normal	0.1	Low	0.1
	Book Damaged	0.01875	Normal	0.1	Low	0.1
Issue	Book Issue	0.075	Normal	0.1	Medium	0.3
	Not a Valid Member	0.0375	Normal	0.1	Low	0.1
	No of Book Exceed Max Limit	0.0375	Normal	0.1	Low	0.1
Fine	Deposit amount Successful	0.075	Severe	0.3	Medium	0.3
	Deposit amount Not Successful	0.0375	Normal	0.1	Low	0.1
	Requested Book Not Available	0.0375	Normal	0.1	Low	0.1

V. Estimate No of Allotment of Test Cases using Various Models

The previous researcher Musa has given model for allotment of test cases based on operational profile. He had considered usage probability alone for computing no of test cases for each task. Thereafter Leung and Wong added criticality (severity) in their model with probability. In our proposed model we added Data interactivity factor to Leung and Wong model. The Musa and Leung & Wong had used traditional Software operational profile (SOP), where we have used operational profile (USOP) developed using Uses Case Methodology. The notation used here are Wi = Weight Calculated, Si = Severity, Di = Data interactivity, Pi = Usage Probability, i = task index.

1. Musa Model using USOP:

   Test case= Pi*250

2. Leung and Wong Model using USOP

   The weight is $$Wi = \frac{Si}{\sum_{i=0}^{n} SiPi} Pi$$

   AND Test case= Wi*Pi*250

3. Proposed Model using USOP

   The weight is $$Wi = \frac{SiDi}{\sum_{i=0}^{n} SiDiPi} Pi$$

   AND Test case= Wi*Pi*250

Table 3: Estimate No of Test Cases in MUSA, LEUNG & WONG with PROPOSED Model using USOP

Function	Operations	Pi	Di	Si	PROPOSED MODEL		MUSA MODEL	LEUNG & WONG MODEL	
					Weight	TEST CASE	Test case	Weight	TEST CASE
Query	Request Matched	0.1	0.1	0.1	0.047	12	25	0.087	22
	Request query not Matched	0.05	0.1	0.1	0.024	6	13	0.043	11
	Request for purchase	0.05	0.6	0.1	0.142	36	13	0.043	11
Request	Book Request registered	0.055	0.3	0.1	0.078	20	14	0.048	12
	Requested Book Available	0.0275	0.1	0.1	0.013	4	7	0.024	6
	Requested Book Not Available	0.0275	0.1	0.1	0.013	4	7	0.024	6
Status	Requested Book Available	0.055	0.1	0.1	0.026	7	14	0.048	12
	Requested Book Not Available	0.0275	0.1	0.1	0.013	4	7	0.024	6
	Request Not Registered	0.0275	0.1	0.1	0.013	4	7	0.024	6
Cancel	Cancel Request Successful	0.065	0.1	0.1	0.031	8	17	0.057	15
	Request Not Registered	0.065	0.1	0.1	0.031	8	17	0.057	15
Deposit	Deposit Book Successful	0.075	0.1	0.1	0.036	9	19	0.065	17
	Delay Deposit	0.01875	0.1	0.1	0.009	3	5	0.016	4
	Book Lost	0.01875	0.1	0.1	0.009	3	5	0.016	4
	Book ID not in Member Account	0.01875	0.1	0.1	0.009	3	5	0.016	4
	Book Damaged	0.01875	0.1	0.1	0.009	3	5	0.016	4
Issue	Book Issue	0.075	0.3	0.1	0.107	27	19	0.065	17
	Not a Valid Member	0.0375	0.1	0.1	0.018	5	10	0.033	9
	No of Book Exceed Max Limit	0.0375	0.1	0.1	0.018	5	10	0.033	9
Fine	Deposit amount Successful	0.075	0.3	0.3	0.32	80	19	0.196	49
	Deposit amount Not Successful	0.0375	0.1	0.1	0.018	5	10	0.033	9
	Requested Book Not Available	0.0375	0.1	0.1	0.018	5	10	0.033	9
Valid Path = 17 & Valid Events = 46, Total test Cases =250		1				261*	258*	1.001	257*

VI. Estimate No of Allotment of Test Cases using Various Models

The number of test cases to be allocated to the software operations is shown in Table 3. Results obtained from the proposed and the other models are compared using measures such as relative difference (RD), square of relative difference (SRD) and mean square relative difference (MSRD).

Where, Relative Difference (R.D)

$$= \frac{\text{Traditional-Proposed}}{\text{Traditional}}$$

Square of Relative Difference (SRD) = $(R.D)^2$ AND MSRD = $(R.D)^2$ / Total number of operations

6.1 Operation wise Comparison between Proposed with Musa and Leung Model

Comparison of results obtained from the allocation of test cases considering the model by MUSA and Leung & Wong with the proposed model are shown in Table 4.

TABLE 4: RESULT COMPARISON BETWEEN MUSA, LEUNG & WONG WITH PROPOSED MODEL

Function	P_i	D_i	S_i	Test case out of 250			MUSA			LEUNG		
				MUSA	PROPOSED	LEUNG	RD	SRD	MSRD	RD	SRD	MSRD
Query	0.2	0.6	0.1	50	81	39	-0.6	0.39	0.009	-1.08	1.16	0.026
Request	0.11	0.3	0.1	28	23	22	0.18	0.03	0.001	-.05	0	0.001
Status	0.11	0.1	0.1	28	8	22	0.71	0.51	0.012	0.64	0.41	0.009
Cancel	0.13	0.1	0.1	33	9	25	0.73	0.53	0.012	0.64	0.41	0.009
Deposit	0.15	0.1	0.1	38	10	29	0.74	0.54	0.012	0.66	0.43	0.01
Issue	0.15	0.3	0.1	38	31	29	0.18	0.03	0.001	-0.07	0	0.001
Fine	0.15	0.3	0.3	38	91	87	-1.4	1.95	0.043	-0.05	0	0.001
Valid Path = 17 & Valid Events = 46				253	253	253	0.53	3.98	0.09	0.7	2.41	0.057

Results show that the MSRD values between the models yield out to be 82.4% and 42.6% of MUSA and LEUNG & WONG respectively. The Query and Fine functions show increased MSRD due to the data interactivity. The rest of the functions are having around 1% justifiable difference. This high difference can also be justified due to the no of attributes. The clear variation shows the impact of factor considered with probability.

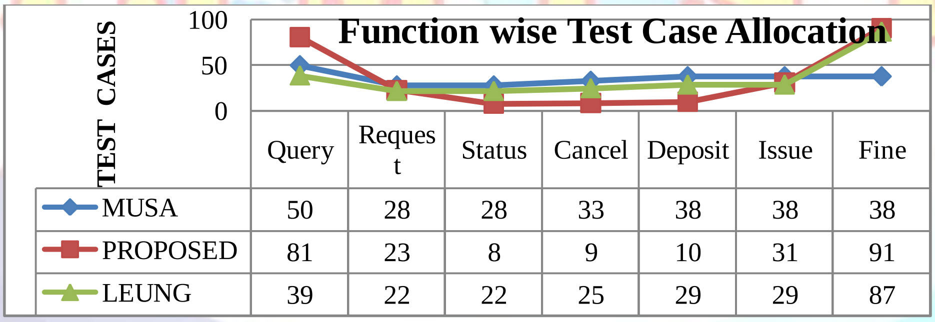

Figure 1: Test case allotment comparison of three Models

VII. CONCLUSION

In this paper, sensing algorithms based on the sample covariance matrix of the received signal have been proposed. Statistical theories have been used to set the thresholds and obtain the probabilities of detection. The methods can be used for various signal detection applications without knowledge of the signal, channel, and noise power. Simulations based on the narrow-band signals, captured DTV signals, and multiple antenna signals have been carried out to evaluate the performance of the proposed methods. It is shown that the proposed methods are, in general, better than the energy detector when noise uncertainty is present. Furthermore, when the received signals are highly correlated, the proposed method is better than the energy detector, even if the noise power is perfectly known.

REFERENCES

[1] J. D Musa, "Operational profiles in Software Reliability Engineering", IEEE Software, Vol. 10, Issue 2, Mar 1993, pp 14-32.

[2] J.D. Musa, The operational profile, in: S.Ozekici (Ed.), Reliability and Maintenance of Complex Systems, NATO ASI Series,vol. F154, Springer, Berlin, 1996, pp. 332–343.

[3] Sham Arora, R. B. Misra, Vinod M. Kumre, "Software Reliability Improvement through Operational Profile Driven Testing", Reliability and Maintainability Symposium, Proceedings IEEE, 24-27, 2005,pp 621-627.

[4] J.D. Musa, A. Iannino, K. Okumoto, Software reliability: Measurement, prediction, application, McGraw-Hill, New York,1987.
[5] J.D. Musa, "Software Reliability Engineering:More Reliable Software, Faster Development and Testing", McGraw-Hill, New York, 1998.
[6] Pankaj Dalal, R.B. Mishra, "Software Operational Profile Development using Use Case Model",
[7] Himanshu, P., Franklin, P., Everett, W., A Structured Approach to Improving Software-Reliability Using Operational Profiles, IEEE Proceedings of RAMS, pp 142–146, 1994.
[8] Musa, J. D., Software Reliability Engineering: More Reliable Software Faster and Cheaper, 2nd. Edition, Tata Mc-Graw Hill Publishers, 2005.