Integrated Intelligent Research (IIR)

International Journal of Computing Algorithm
Volume: 06 Issue: 01 June 2017 Page No.1-4
ISSN: 2278-2397

An Implementation of Enhanced Combinatorial Interaction Testing Software (ECITS)

S.Bhuvana[1], M.V.Srinath[2]
[1]Research Scholar, Department of Computer Science, S.T.E.T Womens College, Mannargudi, Thiruvarur(D.t.).
[2]Research Advisor & Director, Department of Master of Computer Application, S.T.E.T Womens College, Mannargudi, Thiruvarur(D.t.).

Abstract - A 3-way strategy on Combinatorial Interaction Testing is a hopeful approach to handle the difficult to capture bugs. To verify the accuracy and reliability of software, it should be tested from simple to complex scenario. Currently software structures are composite and have a lot of potential arrangements. Testing should be done in such a way to capture all the difficult scenario bugs in different combinations. This paper mainly focuses on the implementation of a 3-way strategy on Enhanced Combinatorial Interaction Testing Software (ECITS). It combines the distinct probable use of 3-way strategy on even communication and adjustable asset communication, as well as both input andoutput based communication. With a purpose to support engineers to create well-versed result by the use of a3-way strategy on different applications, the paper describes stepwise illustration of its Software development customization was done with the modification of run-time and assemble time tendency, authorizing customers to create measured variations, such that their code conveys onflexible structures for ideal web servers (e.g., Apache), databases (e.g. SQLServer), application servers (e.g., Visual Studio) or working environment applications (e.g., MS Word) that have some or even frequent results. To relax on after that will have significant features of other huge amount of strategies, as graphic.The system space is basically the analyzer selections for the compiler GCC and has 4:6 1061 preparations, and the 50% of the Apache server that dominates archive side by side grows to the operations. Enhanced Combinatorial Interaction Testing Software is that it which can is a cost-adequate preparation of the entirely structure performance and it was formed by the locations of 3 or else fewer adoptions. We guess though that the preparations of several of these perform will remain not actually strained as of surprising hiding influences. Despite the fact validation the exactness of the structure done in its entire strategy area is excessive and fascinating, complete testing of entire plans is essentially impractical. For example the sum of plans develops exponentially by the total sum of system results, the number of possible schedules generally route on the extreme side of the available around resources, to route they consider cases in an appropriate method. Unique research method, mentioned as Enhanced Combinatorial Interaction Testing Software (ECITS), systematically examines the plan space and tests especially the selected provisions. ECITS events work through principal starting of perfect structure plan area the established of large sequences in which it can be collected. Frequently, this typical integrates an assembly of system selections every possible test will shows a slight amount of credibility locations, and an assembly of structure extensive break strains that is a part of systems, recognized after the previous test that to be inacceptable. Assumed this typical, ECITS systems next list a set of genuine preparations, a 3-way strategies cluster, practical implementation. The proposed ECITS catches the error combinations in every single testing in java program. This typical model considers nearly the day functionalism review for errors, and also reports in which segment and batch, an error has been occurred. The concerned batch contracts by the errors in the segment and the segment is once more informed and then ended for testing. The segment is promoted to try and capable to be gauged report is generated which proves the significance of the testing. Test moments of predicted background provide developed outcome as well as difference and the prevailing outline.

I. INTRODUCTION

confidential which every attainable combination of credibility locations for every combination of 3 results performs at smallest once.Finally, the structure is patterned successful.Its test suite on each preparation confidential the casing displays. A preliminary components procedure to fading the outcomes of surprising covering effects. Now that work originally considered a combinatorial testing model that plans to promise that every single research has a sensible chance to test every recent bit and its grateful combinations have excellent locations. We formed a censure determined adaptable combinatorial testing method to appear in this regulation for preparation. The importance of this approach, we identify latent covering influences, undo their possible details, and subsequently yield innovative plans that manage the presumable disappointment source.

II. TESTING

Since 1980 combinatorial testing has been connected in various fields and applications. For example, ADA compiler was first tested by combinatorial scope in 1985. Tested PELMOREX (Weather forecast generator) by pairwise combinatorial scope in 1992 and created the OATS (Orthogonal Exhibit Testing Strategy) framework for test case generation. Numerous papers have been published that share the experience of utilizing CT in practice. In year 2000 uses CT to test the ground framework of satellite communication. In this paper that, how CT improves the quality and efficiency of internet protocol testing. In 1996 that CT can be utilized to test network interface has been moreover utilized in other application too. In year 2000 proposed a procedure for testing GUI objects. Empirical study of this procedure appears that, if we follow CT frameworks then a less number of test cases can moreover detect the tests of the GUI. It Created test cases with the help of CT to test the email framework with AETG. Configuration testing and browser testing were moreover evolved as the child of CT.

International Journal of Computing Algorithm
Volume: 06 Issue: 01 June 2017 Page No.1-4
ISSN: 2278-2397

III. RELATED WORK

A) Cuckoo Exploration Constructed Pairwise Testing (PT) for Combinatorial Testing

In this research the authors analyzed to find the efficiency of software, for which pairwise cuckoo search[6] strategy is used. Optimal test suite is formed by collecting the optimal test cases. Optimal test case means every input pair can test the code by its test cases effectively. With this approach to form the optimal set of test cases, first step is to generate the binary combination from the input parameters then interaction element list is formed accordingly. Using pairwisethat generates the initial test case randomly and find the best test case till the interaction list is empty. Finally add the best test case to test suite.

B) Applying Combinatorial Testing Strategy of CIT

In this implementation the authors introduce and analyze CT to check a combinatorial test creation tool named ACTS[12]. At first, they wish to increase knowledge and visions regarding, in what way to apply combinatorial testing in repetition section. And then they wish to estimate the reliability of CT applied to a real life system. ACTS provides interface via command line CLI and an equally sophisticated realistic operator interface. The principle test of this research was to create a perfect model and the given input range by establishing restrictions and principles. After the perfect model was planned, they produced test cases via ACTS, which stayed formerly later used to test ACTS. The outcomes of this research displays the input range displaying can be a major responsibility, and desires to be sensibly accomplished. The execution results display that the combinatorial testing is valuable in accomplishing great program analysis and error recognition.

C) Combinatorial Interaction Testing Using the Tree Strategy

To minimize the test size and generate highly recommended one to reduced test suite, 3-way testing strategy interaction is added to existing “A Tree Based Strategy for Test Data Generation and CostCalculation”[3]. Two algorithms are mainly used in this implementation. Tree construction algorithm that generates the needed test case and an interactive calculation algorithm constructs 2 or 3 way test suites which include all possible input combinations.

D) Combinatorial Interaction Test Using Hyper heuristic Search

Most recently, the researchers present the hyper heuristic algorithm[15] for CIT samples. It provides a single common approach for solving same problem of different classes. Experiments conducted with this algorithm proved that it can compete with good solution across constrained and unconstrained issues. Different tuning (low, medium and high) of algorithm and various stages of algorithm (Early, middle, late) gives better results cost wise and has the ability to learn and adopt the search.

E) Sequence Based Combinatorial Testing

In this paper focuses on SCAT[8], a sequence based t-way testing method. Main objective of SCAT is to produce minimal test suite size. It uses TCGA (Test Case Generation Algorithm) to select the optimal test case from pool of test case for final test suite. TCGA can cover the most uncovered tuples. Experimental outcome shows that it gives good results in test suites size when compared with different existing strategies.

IV. PROPOSED METHODOLOGY

Enhanced Combinatorial Interaction Testing Software (ECITS) testing has attracted both academia and industry. Several software engineers have indicated that combinatorial testing could dramatically diminish the number of tests while remaining compelling for detecting programming faults. Moreover, combinatorial testing is relatively easy to apply. As of TF technique, combinatorial testing does not necessitate investigation of source code, which is regularly troublesome for practical applications. In the direction of applying CT, a set of limitations, as well as their possible values, need to be identified. This data is regularly as much as ease to obtain than an operational model as required by numerous other TF techniques. After the parameters and their values are identified, the actual test Era procedure can be completely automated, which is the main key to industrial acceptance. Combinatorial testing relook has made significant progress in recent years, and proceeds to make progress each day, particularly in the bearings outlined in the past section. With these progresses, combinatorial testing is expected to be completely integrated with the existing testing forms and become an imperative device in the toolbox of programming practitioners. The wide use of combinatorial testing will help to significantly diminish the cost of programming testing while increasing programming quality. It will moreover progress the productivity of programming engineers by reducing the time and energy they spend on testing.

A) ECITS Using 3-ways

In this paper we received computational procedure which is based on a searching procedure to make test suite. The proposed procedure begins by building the test-tree based on the information parameters and values. The estimation develops the covering array, for all possible blends of Information variables. After that the cost exhibit corresponding to the number of test cases (or leaf nodes) is made and initialized to some high value. Then, the cost estimation begins. The estimation first calculates the maximum cost or maximum number that can be secured by any test case for the given set of parameters and values. Then it iterates to calculate the cost of each leaf hub which represents the test case, in a sequential order. The cost of any leaf hub or test case is equal to the number of 3-ways that it covers in the covering array. Once it reaches a leaf hub with the maximum cost, it erases leaf hub from the list of leaf hubs Created by the test-tree and includes hub or test case into the new list which holds all the test cases that are to be included in the test suite. It moreover erases all the sets that this test case has secured from the covering array. 3-way covering exhibit can be created as in

B) ECITS Algorithm

Input: Java Code

Step 1:Assign number of lines in a file as variable ‘n’

Step 2:For each line ‘i ‘in a file till n value**.** Find the function is present in each line and set it to array variable ‘m’ and append itnd for.

Step 3:Calculate the number of functions in array variable‘m’ and assign it to variable ‘fn’

Step 4: For each function 'j' till 'fn' timesCalculate the number of attributes and assign it to variable 'g'end for

Step 5:For each variable 'k' till 3 time

If the number of attributes 'g' is greater than 3

Exit the program

Else

Find the data type of attribute and assign it to variable 'f'Do combination of data type of attribute end if

end for

Output: combination

Table 1

Connecting Information Events	Arrangement Tuples Produced	Connecting Informatio n Events	Arrangement Tuples Produced
A,B,C	A□B□C A□C□B B□C□A B□A□C C□A□B C□B□A	A,D,E	A□D□E A□E□D D□A□E D□E□A E□A□D E□D□A
A,B,C	A□B□D A□D□B B□A□D B□D□A D□A□B D□B□A	B,C,D	B□C□D B□D□C C□B□D B□D□B D□B□C D□C□B
A,B,E	A□B□E A□E□B B□E□A B□A□E E□A□B E□B□A	B,D,E	B□D□E B□E□D D□B□E D□E□B E□B□D E□D□B
A,C,E	A□C□E A□E□C C□A□E C□E□A E□A□C E□C□A	C,D,E	C□D□E C□E□D D□C□E D□E□C E□C□D E□D□C

V. CONCLUSION

In this paper we broaden and progress our past strategy, to support 3-ways testing quality interactions. The proposed procedure is based on ECITS algorithms. A 3-way construction estimation which develops the possible test cases and an iterative cost estimation that develops effective 3-way test suites, which spread all possible combinatorial interactions between Information components.

Table 2: Comparison Result

Program	Loc	No.of functions	Exec. Time	No.of Com
p1	1127	20	342	4.00E+09
p2	1267	22	451	4.00E+11
p3	1567	18	312	3.00E+06
p4	1765	16	286	6.00E+09
p5	2346	27	540	6.00E+15

We have implemented and checked five different java programs containing various LOC (Length of Code) and also it has several functions. As per our result we get different execution time and combinations of that program. Those details are illustrated at table 2.

For Example:

In java program p1 contains a total of 1700 line of code on which 1127 are valid LOC. The remaining are blank lines and comments, which are neglected. In those lines there are around 28 functions, and of that only 20 functions contains three input arguments, which has been taken as input and processed for various combinations. The performance result &testing efficiency of input java programs is shown in figure 5.1 & 5.2.

The proposed procedure works well for diverse test quality values, and can produce an effective and reduced test suite size. Our procedure has been one of the best results for most of the test. In this future work we will be developing not only the three way construction estimation. It will require "n" number of tuples, which are used to generate the test cases and the loading test cases are minimal amount because it will reduce the timing of test case generation.

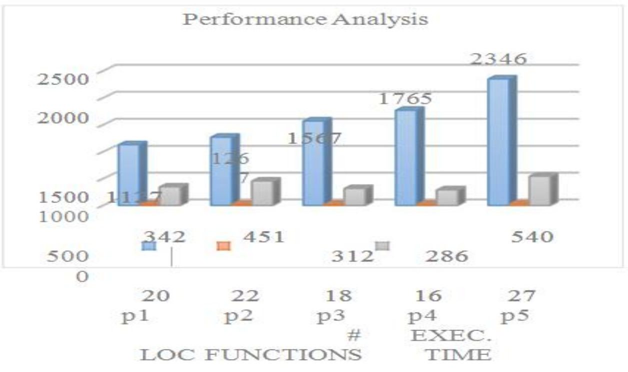

Fig. 5.1 Performance Analysis

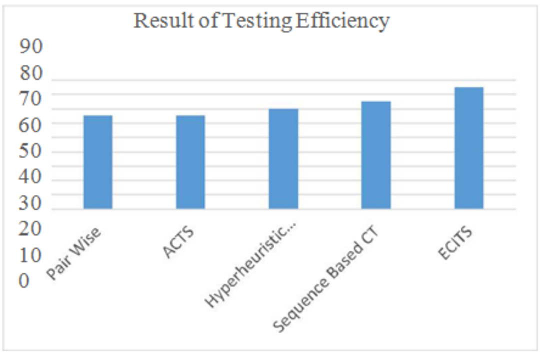

Fig. 5.2 Result of Testing Efficiency

References

[1] Abdullah, B., Nasser &Yazan, A., "A CuckooSearchBased Pairwise Strategy for Combinatorial Testing Problem", in JATIT, Vol.82.No.1, 2015.

[2] Borazjany,M.N., Yu, L., &Y., Lei,"CombinatorialTesting of ACTS: A Case Study", 2012 IEEE FifthInternational Conference on Software Testing.

[3] CemalYilmaz, SandroFouche& Myra, B., Cohen, "Moving Forward with Combinatorial Interaction

Testing”, in IEEE Computer, Vol-47, Issue-2, PP-37 – 45, 2014.

[4] CemalYilmaz, “Test Case-Aware Combinatorial Interaction Testing”, in IEEE SoftwareEngineering, Vol-39, Issue-5, PP-684 – 706, 2013.

[5] Charles Song, Adam Porter & Jeffrey S., Foster,“iTree: Efficiently Discovering High-Coverage Configurations Using Interaction Trees”, in IEEESoftware Engineering, Vol-40, Issue-3, PP-251 – 265, 2014.

[6] Christopher Henard, Mike Papadakis& Gilles Perrouin, “Bypassing the Combinatorial Explosion: Using Similarity to Generate and Prioritize T-Wise Test Configurations for Software Product Lines”, inIEEE Software Engineering, Vol-40, Issue-7, PP-650 – 670, 2014.

[7] D. M., Cohen, S. R., Dalal, M. L.,Fredman& G. C., Patton, “The AETG system: an approach to testingbased on combinatorial design”, in IEEEEvolutionary Computation, Vol-19, Issue-4, PP-575– 591, 2015.

[8] Ely Porat,Amir Rothschild, “Explicit Nonadaptive Combinatorial Group Testing Schemes”, in IEEE Information Theory, Vol-57,Issue-12, PP-7982 – 7989, 2011.

[9] Fang Hao, MuraliKodialam, T. V.,Lakshman&Haoyu Song, “Fast Dynamic Multiple-SetMembership Testing Using Combinatorial Bloom”Filters”, in IEEE/ACM Networking, Vol-20,Issue:1, PP-295 – 304, 2012.

[10] JustynaPetke, Myra, B., Cohen, Mark Harman & Shin Yoo, “Practical Combinatorial Interaction Testing: Empirical Findings on Efficiency and Early “Fault Detection”, in IEEE Software Engineering,Vol-41, Issue-9, PP- 901 – 924, 2015.

[11] Mohammad F. J.,Klaiba, Mohammad Subhi Al-batahb&Rashad, J.,Rasrasc, “3-way InteractionTesting using the Tree Strategy”, ICCMIT 2015,Procedia Computer Science 65 (2015) 845 – 852

[12] Myra B.,Cohen , Matthew B., Dwyer,“Constructing Interaction Test Suites for Highly-Configurable Systems in the Presence of Constraints: A Greedy Approach”, in IEEESoftware Engineering, Vol-34, Issue-5, PP-633 – 650, 2008.

[13] Renee C., Bryce, SreedeviSampath&Atif M.,Memon, “Developing a Single Model and TestPrioritizationStrategiesforEvent-DrivenSoftware”, in IEEE Software Engineering, Vol-37,Issue-1, PP-48 – 64, 2011.

[14] YueJia, &Myra,B.,Cohen, “LearningCombinatorial Interaction Test GenerationStrategies using Hyperheuristic Search”, 2015IEEE/ACM, Vol-1, PP-540 – 550.

[15] M., ZamriZahir Ahmad, R., Razif Othman &M.S.Aziz Rashid Ali, “Sequence CoveringArrayGenerator (SCAT) For Sequence BasedCombinatorial Testing”, in IJAER, Volume11,Issue: 8, PP: 5984-5991, 2016.