Test Case Generation Techniques

Anchal Chauhan
Student, M.Tech., DCSE, KIET Ghaziabad (U.P.)

Ravi Shankar Singhal
Assistant professor, DCSE, KIET Ghaziabad (U.P)

Pramod Kumar Yadav
Associate professor, DCSE, KIET Ghaziabad (U.P)

Abstract — *Software testing can be done automatically or manually with the use of many different testing tools. Many techniques are available in market for generating test cases. This paper contains different test case generation techniques such as random, goal oriented, specification based, sketch diagram based and source code based.*

Keywords — *Test case generation, techniques, specification based, sketch diagram based, source code based*

I. Introduction

Software testing is an important part of SDLC (software development lifecycle). It is helpful in finding human errors and also logical errors. It helps to make sure that the software we are working over is working properly or not and also it can be released to market or not. If proper testing of software is not done before releasing it then this can harm organization business as it will not satisfy need of customer and reviews will be bad and also it can cause theft of user's private data. Nowadays software's are tested automatically because it is accurate and fast. Many techniques are present to generate test cases. But for generating test cases, testers first understand the requirements and specifications with the help of requirement document. Then they prepare test cases with the use of test case template where a test case template is a kind of document which contains details about test case ID, test case description, expected result and input.

II. Steps for test case generation

1. Finding all constraints from starting to end node where a constraint is described as an algebraic expression that dictate conditions of variables.
2. In order to reduce the test cases, variable having highest value is given maximum value and variable having lowest value is given minimum value.
3. then, variables are assigned with the constant value at each node.
4. Finally, a table including all possible test case is created.

III.Techniques

Various techniques are available for test case generation such as random, specification based, sketch diagram based, goal oriented technique and source code based technique. In brief we will discuss all these techniques but mainly we will discuss about specification based, sketch diagram based and source code based with their advantages and disadvantages in this paper.

A. Random Technique

Random technique discovers a set of test cases which are based on the assumptions involving fault distribution.

B. Goal Oriented Technique

This technique determines those test cases that covers selected goal. For example-statement of branch and irrespective of path taken.

C. Specification Based Technique [1]

Technique which uses specification document to generate test cases is called specification based technique. Its advantages-

1. Software tester have information related to the functionality of software and do not have to extract it.
2. Specification document is useful for deriving expected result for test data.
3. Tests can be generated concurrently with designing and implementation.

Disadvantages-

1. Difficulty in conducting the formal analysis.
2. This technique involves more manual effort in generation of test cases.

D. Sketch Diagram Based Technique [1]

Techniques which are used for generation of test cases from model diagrams are sketch diagram based techniques. For example- UML use case diagram, UML sequence diagrams, UML state diagrams OR technique that involves UML (unified modeling language) to generate test cases.

Its advantages-

1. Can be easily automated
2. This method saves time and resources
3. Used in web applications.

E. Source Code Based Test Case Technique or Path Oriented Test Case Technique [1]

Techniques which uses control flow diagrams to determine set of path to be covered and make proper test cases for those paths.

Its advantage is that it is more often do not lead to selection of the infeasible path.

IV. Literature Review

In 1994, Weyuker presented a method for test case generation from Boolean logic specification. After this, many criteria were also introduced by him for system level testing. [2]

Then in 1996, Blackburn and busser presented a language T-Vec in which Blackburn expressed state based functional specification in the software. T-Vec language given by Blackburn and busser was helpful to solve disjunctive normal form constraints for generating tests. [3]

In 1997, kancherla tried specification based testing which involves use of automated theorem prover for generation of test cases. He used PVS (prototype verification system) which was a new method for testing software which was based on formal specifications. [4]

After this, in 1999 cunning and rozenblit in their study presented automatic generation of the test cases from specification requirements for systems which are event oriented and real time. They tried heuristic algorithm for automatic generation of test cases. Heuristic algorithm used by them involved greedy search method which can follow distance based search if needed in test case generation. [5] Offut in 1999 gave model for making test inputs from the state based specifications. [7] Pargas and yang used genetic algorithm and CFG for test case generation in web applications. [6] [8]

In 2000, Ryser and Glinz derived test cases from UML use cases or scenarios and state diagrams. [9] Cavarra also presented modelling architecture for model based testing and verification. [10] Then in 2001, Tran does all his research use of model checking for generating test cases. He generated test cases by using mutation analysis. [11] Rayadurgam and heimdahl also gave method for structural coverage criteria [23]. Heuman presented use cases to generate test cases. He introduced three step process for generation of test cases from use case: 1) generated full set use case scenario for each use case 2) then for every scenario determine at least a test case and condition that can execute 3) for every test case, determine data values with whom to test. [22] El- Far and Whittaker in the same year generated test cases from finite state machines. [12]

In 2003, Nilawar and Dascalu researched for testing of web based applications. They presented test cases that contains: set of test inputs, expected result and execution conditions. [13] In the same year, Beydeda and Gruhn presented an activity oriented approach for testing of web applications. [21]

In 2005, Sinha and Smidts introduced new technique which was model based to determine critical domain requirements. This technique is based on system modelling by the use of Domain Specific Language (DSL). [14]

In 2006, Antonio provided a mutation approach for object constraint language. In that year, Santiago also used state charts to generate test cases automatically. [15]

In 2008, Reza introduced testing technique for web applications which was model based. He focused on front end design and web application testing. [19]

In 2012, Elber zhager presented mapping study over existing approaches.

In 2015, A. Jalila, D.J. Mala, M Eswaran used OCL predicates to find early identification of software defects so that quality of software can be made better or improved.

Summary of literature review in tabular form—

Researcher name	**Year of research**	**Research field**	**Technique used**
Weyuker	1994	Presented Boolean logic specifications method.	Specification based technique
Blackburn and Busser	1996	Presented state based functional specifications of the software which is expressed in T-Vec language	Specification based technique
Kancherla	1997	Used automated theorem prover to generate test cases	Specification based technique
Cunning and Rozenblit	1999	Used heuristic algorithm for generating test cases	Specification based technique
Pargas	1999	Used genetic algorithms	Source code based technique
Offut	1999	Gave model for making test inputs from state based specifications	Specification based technique

Researcher name	**Year of research**	**Research field**	**Technique used**
Ryser and Glinz	2000	Gave method for system test	Sketch diagram based technique
Cavarra	2000	Presented modelling architecture for model based testing and verification	Sketch diagram based technique
Tran	2001	Test case generation using model checker	Specification based technique
Rayadurgam and heimdahl	2001	Gave method for structural coverage criteria	Specification based technique
Heumann	2001	Presented use cases to generated test cases	Sketch diagram based technique
El Far and Whittaker	2001	Presented test cases from finite state machines	Sketch diagram based technique
Nilawar and Dascalu	2003	Generated test cases for web based applications by explaining unified modelling language	Sketch diagram based technique
Sinha and Smidts	2005	Presented model based testing method to identify critical domain requirements	Sketch diagram based technique
Antonio	2006	Gave a mutation approach for OCL(object constraint language)	Specification based technique
Yang	1999	Used CFG for web applications	Source code based technique
Nilsson	2006	Used heuristic driven simulation for generating test cases	Specification based technique
Santiago	2006	Used state charts for automatic generation of test cases	Sketch diagram based technique
Reza	2008	Used state chart for web applications	Sketch diagram based technique
Turner	2008	Presented an activity oriented approach for testing web application	Source code based technique
Elberzhager	2012	Presented mapping study over existing approaches	Specification based technique
A Jalila, DJ Mala, Eswaran	2015	With use of OCL predicates tried to find early identification of	Specification based technique

V. Conclusion And Future Work

This paper includes different test case generation techniques year wise. We have tried to present a brief description of literature survey of test case generation, also presented it in tabular form.

Future work may include improving existing techniques or integration of two or more techniques.

ACKNOWLEDGMENT

I would like to express my gratitude and regards to Dr. Vineet Sharma, Professor & Head, Department of Computer Science and Engineering, KIET Ghaziabad for their guidance and motivation. I also express my heartful thanks & gratitude to Dr. A. P. Shukla, Professor and M. Tech Coordinator, Department of Computer Science and Engineering, KIET Ghaziabad for their valuable support throughout my research work.

My friends have also helped me with personal and professional guidance. They taught a great deal for scientific research and life.

Last but not the least, I want to thank my parents. Nobody is more important to me than my family members. Their love, support and guidance helped me so much in doing this research work. They are role models for me.

REFERENCES

[1] Nicha Kosindrdecha and Jirapun Daengdej "A test case generation technique and process" Journal of Software Engineering 4(4) 265-287 2010, Academic Journal Inc., 2010 ISSN 1819-4311.

[2] Weyuker, E., T. Goradia and A. Singh, "Automatically generating test data from a Boolean specification" IEEE.Trans. Software Eng., 199420 353-363.

[3] Blackburn, M. And R. Busser, "T-VEC: A tool for developing critical systems." Proceeding of the Annual Conference on Computer Assurance, (ACCA'96), IEEE Computer Society Press, 1999 pp: 237-249.

[4] Kancherla, M.P., 1997.Generating test templates via automated theorem proving. Technical Report, NASA Ames Research Center.

[5] Cunning, S.J and J.W. Rozenblit, 1999. Automatic test case generation from requirements specification for real time embedded systems. IEEE Int. Conf. Syst. Man Cybernetics, 5: 784-789.

[6] Pargas, R.P., M.J. Harrold and R.R. Peck, "Test data generation using genetic algorithms" Software Testing Verification Reliability, 1999, 9: 263-282.

[7] Offutt, A.J., Y. Xiong and S. Liu, 1999. Criteria for generating specification-based tests. Proceedings of the 5th International Conference on Engineering of Complex Computer System, Oct. 18-22, Washington, USA., pp: 119-119

[8] Yang, J.T., J.L. Huang, F.J. Wang and W.C. Chu, "Constructing control flow based testing tools for web application" 11th Software Engineering and Knowledge Engineering Conference(SEKC'99), June 1999.

[9] Ryser, J. and M. Glinz, "SCENT: A Method employing scenarios to test cases for system test." Technical Report, http://portal.acm.org/citation.cfm?id=901553.

[10] Cavarra, A., C. Crichton, J. Davies, A. Hatman, T. Jeron and L. Mounier, "Using UML for automatic test generation" Oxford Univesity Computing Laboratory, Tools and Alllgorithms for the Construction and Analysis of Systems, TACAS'2000.

[11] Tran, H., "Test generation using model checking" European Conference on Software Maintenance and Reengineering, CSMR2001, http://www.cs.toronto.edu/~chechik/courses00/csc2108/ projects/4.pdf.

[12] El-Far, I.K. and J.A. Whittaker, "Model based software testing" http://143.225.25.115/~flammini/materiale/Modelbased%20Testing /ModelBasedSoftwareTesting.pdf. 2001.

[13] Nilawar, M. and S. Dascalu, "A UML-based approach for testing web applications" M.Sc. Thesis, University of Nevada, Reno. 2003.

[14] Sinha, A. and C.S. Smidts, "Domain specific test case generation using higher ordered typed languages from specification" Ph.D. Thesis, University of Maryland, 2005.

[15] . Antonio, P., P. Salas and B.K. Aichering "Automatic test case generation for OCL: A mutation approach" PROCEEDING OF 5th International Conference Quality Software, Jan. 2006, IEEE Computer Society, pp: 64-71.

[16] Nilsson, R., J. Offutt and J. Mellin, "Test case generation for mutation based esting of timeliness." Electronic Notes Theor. Comput. Sci., 164: 97-114.

[17]] Santiago, V., A.S.M. Do- Amaral, N.L. Vijay kumar, M.D.F., M. attiello-Francisco, E. Martins and O.C. Lopes, "A practical approach for automated test case generation using state charts." Proceedings of the 30th Annual International Computer Software and Applications Conference, Sept. 17-21, IEEE Computer Society, pp: 183-188

[18] Tuner, D.A., M. Park, J. Kim and J. Chae, "An activity oriented approach for testing web applications" Proceeding of the 23rd IEEE/ACM International Conference on Automated SoftwareEngineering, Sept. 15-19, Washington, USA., 2008, pp: 411-414.

[19] Reza, H., K. Ogaard and A. Malge "A model based testing technique to test web applications using state charts" Proceeding of 5th International Conference on Information Technology: New Generations, April 7-9, Las Vegas. 2008, pp: 183-188.

[20] Gupta, N., A.P. Mathur and M.L. Soffa, "automated test data generation using an iterative relaxation method" ACM SIGSOFT Software Eng. Notes, 1998, 23: 231244.

[21] Beydeda, S. and V. Gruhn, "BINTEST-Binary search based test case generation" Proceedings of Computer Software and Application Conference, Nov. 2003, Leipzing Univ., Germany Leipzig Univ., Germany, pp:28-33.

[22] Heumann, J., "Generating test cases from use cases" Rational Software. http://www.ibm.com/developerworks/rational/library/content/RationalEdge/jun01/GeneratingTestCasesFromUseCasesJune01.pdf.

[23] Rayadurgam, S.and M.P.E. Heimdahl, " Test –sequence generation from formal reuriment models." Proceedings of the 6th IEEE International Symposium on High Assrance System Engineering, Oct. Boca Raton, Florida, pp:23-23, 2001, 22-24.