2011 International Conference on Telecommunication Technology and Applications
Proc .of CSIT vol.5 (2011) © (2011) IACSIT Press, Singapore

A Study on Test Coverage in Software Testing

Muhammad Shahid[1], Suhaimi Ibrahim and Mohd Naz'ri Mahrin

Advanced Informatics School (AIS), Universiti Teknologi Malaysia
International Campus, Jalan Semarak, Kuala Lumpur, Malaysia
smuhammad4@live.utm.my, suhaimiibrahim@utm.my, mdnazrim@utm.my

Abstract: Test Coverage is an important indicator of software quality and an essential part of software maintenance. It helps in evaluating the effectiveness of testing by providing data on different coverage items. Although much research effort has been put on how to get coverage information by either code based testing or requirement based testing, not much has been paid to measure and analyze the coverage by covering maximum number of coverage items. This paper provides a study of the current test coverage researches conducted by other researchers for test coverage in software testing. By looking at the existing approaches; gaps and uncovered measurement of Test Coverage can be explored further.

Keywords: Code Coverage, Coverage Measurement, Software Testing, Test Coverage, Traceability

1. INTRODUCTION

Software testing is considered now as an essential activity in software maintenance life cycle. It is a practice often used to determine and improve software quality. Testing activities also include obtaining the test coverage. "Coverage is the extent that a structure has been exercised as a percentage of the items being covered. If coverage is not 100%, then more tests may be designed to test those items that were missed and therefore, increase coverage"[1]. Test coverage can help in monitoring the quality of testing, and assist in directing the test generators to create test cases that cover areas that have not been tested before [2].

The output of coverage measurement can be used in several ways to improve the testing process. It can also provide the user with information on the status of the verification process. It can help to find holes in the testing, i.e. areas that are not covered [2]. Test coverage also helps in Regression testing, test case prioritization, test suite augmentation and test suite minimization. Although much research attention has been given to test coverage measurement and analysis, but there is a need to cover all three granularity levels of test coverage items, i.e. fine- grain, medium-grain and course-grain in detail.

This paper reports on a study to find the research on current approaches related to test coverage. The study presented here is that involves a search of the literature to determine what sorts of studies addressing the systematic review question have been carried out, where they are published, in what databases they have been indexed, what sorts of outcomes they have assessed, and in which populations [3]. This study is aimed at presenting an overview of research concerning test coverage over the period 2000 – 2010 and identifying probable gaps in research about test coverage.

[1] Tel.: + 60 16 2242627; fax: +60 3 26930933
E-mail address: ssheikh_85@hotmail.com

This paper is organized as follows. Section II describes the research method in conducting the study. Section III reports the result and discussion about study. In section IV, we have identified some threats to validity. Finally section V concludes the work.

2. RESEARCH METHOD

2.1 Research Objective

The objective of this study is to summarize current state of the art approaches related to Test Coverage based on the following main questions:

- What are the main research areas in test coverage addressed by other studies?
- What are the current approaches of test coverage findings?
- What is the number of publications per year?

2.2 Information Sources

This study was conducted from May to November 2010. The literature search was carried out by searching in a set of selected repositories. The repositories used were ACM Digital Library, Science Direct, IEEExplore, SciVerse Scopus, Google Scholar and Springer Link. Although the total number of papers was high, most of the papers were redundant. The paper having same title was returned by different repository search engines. Therefore, we eliminated these papers.

2.3 Search Criteria

Different keywords were defined as a search string based on the research questions. These include Code Coverage, Software Testing, Test Coverage, and Traceability. Using the above keywords, a search string was created and used for different repository search engines. The year range 2000 - 2010 was included in the search.

3. The Systematic Process

The essential process steps of our systematic study are shown in Figure 1. Our systematic review process started with defining research question as stated in section II (A). Next step was to define the searching keyword. The keywords are listed in section *C*. Using the keywords, searching was carried out on the selected repositories using the provided search engine. A set of inclusion and exclusion criteria were defined for the selection process. These criteria were then used in the study process after searching.

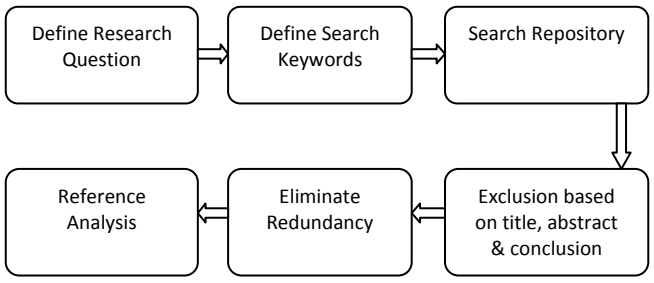

Fig. 1. Mapping Study Process

Once the list of research papers was obtained, the papers not matching our keywords were excluded after reading the title, abstract and conclusion. At the same time, the papers having same title returned by different repositories were removed to eliminate redundancy. Final analysis was conducted on the base of references to ensure that referenced papers were not missed out. Reference analysis is important as some of the publications might have been missed out during the keywords based search using search engines and during exclusion based on title and abstract. By looking at the reference at the end of each paper, relevant papers that are missed earlier can be included in the mapping study process.

4. Result and Discussion

A total of 47 research papers related to Test Coverage were returned by this study searching process. The research papers were then categorized and classified into 6 groups which are survey and study of test coverage, frameworks for test coverage, test coverage items, methods/algorithms, software reliability modeling for test coverage, and others. Figure 2 illustrates the breakdown of research papers according to the six categories.

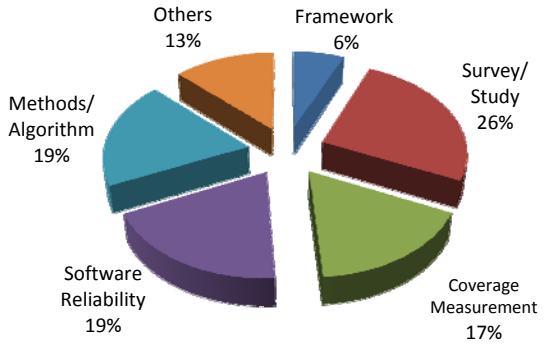

Fig. 2. Breakdown of Research Papers Category

Based on Figure 2, coverage measurement, software reliability issues and studies and surveys are contributing about the equal share. Share for framework development is only 6%.

Figure 3 shows the number of research publications made per year. Research publications on test coverage grew rapidly after year 2004. As of November 2010, there are 5 published papers in this year. This number may increase, since the literature search was conducted between May to November 2010 and more relevant studies not yet published or indexed by digital libraries.

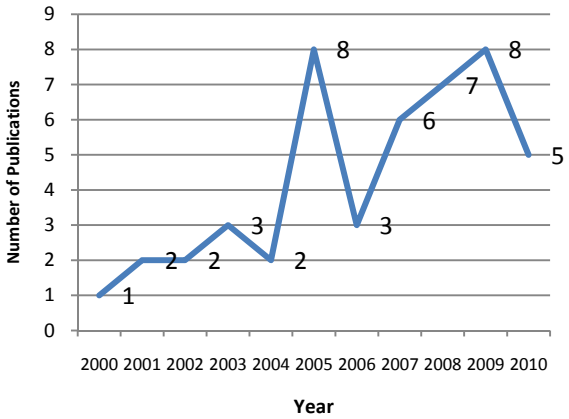

Fig. 3. Number of Research Publications per Year

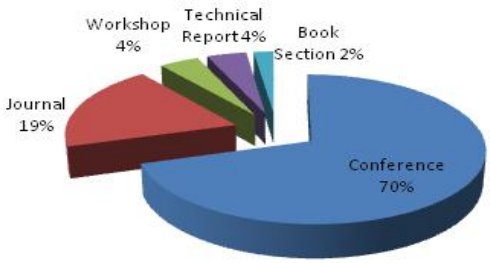

Fig. 4. Breakdown of Publication's Forum Type

The publications were also classified based on different forum types. The forum types are workshops, conferences and symposiums, journals, technical reports and book chapters. A large percentage of the publications come from conferences and symposiums (70 %), followed by journals (19 %). Workshops and technical reports shared 4 % each. This is illustrated in Figure 4.

The following section will discuss the findings of the study for each focus area as categorized in Figure 2.

4.1 Survey/Study

There are three survey type literature related to test/code coverage. The researchers focused on code coverage as a stopping criterion, compared different coverage based tools and looked on different techniques available for generating test cases to satisfy test coverage criteria.

Several researchers studied test coverage analysis related to test effectiveness, test suit improvement and software reliability estimation. Some believe that visualizations reduced the variability in the number of test cases; developers wrote by changing the standard developers used to evaluate their test effectiveness. The entire above researchers are tabulated in Table 1.

Table 1. Survey and study table

Mode	Author	Year	Issues Addressed
Survey	Ben Smith, Laurie Williams [4]	2008	Code Coverage as a stopping criterion for unit testing
	Qian Yang et al. [5]	2007	Coverage-based testing tools
	Prasanna et al. [6]	2005	Test case generation techniques to satisfy test coverage criteria
Study	Audris Mockus et al. [7]	2009	Test effectiveness
	Silva et al. [8]	2009	Code coverage analysis
	Y. Wei [9]	2008	Coverage as Testing effectiveness
	Anna Derezińska [10]	2008	Test suits improvement
	Jun-Ru et al. [11]	2007	MC / DC coverage criterion
	Stefan Berner et al. [12]	2007	Impact on testing
	Lloyd, Malloy [13]	2005	Test coverage adequacy
	Joseph Lawrance et al. [14]	2005	Code coverage visualizations
	Yong Woo Kim [15]	2003	Systematic approach of coverage analysis

4.2 Framework

Three authors have proposed frameworks for test coverage measurement and analysis, one by Sakamoto et al. (2010), the other by Matteo Bordin et al. (2009), and third by Misurda et al. (2005). Sakamoto [16] proposed a framework for consistent and flexible measurement of test coverage, called the Open Code Coverage Framework (OCCF) that supports multiple programming languages. Their framework provides guidelines to support several test coverage criteria. Moreover, OCCF let users expand features to add user-defined test coverage and new programming language. Matteo Bordin [17] proposed a framework, Couverture, which provides a virtualized execution platform for cross-compiled application on the host machine. Couverture is able to measure structural coverage of object and source code without requiring any form of application instrumentation with a single execution of the cross-compiled application and test suites. Misurda, [18] described a new scalable and flexible framework for testing programs with a novel demand-

driven approach based on execution paths to implement test coverage. He used dynamic instrumentation technique on the binary code that can be inserted and removed on-the-fly to keep performance and memory overheads low.

4.3 Test Coverage Items

A lot of research papers focused on different coverage items to measure and analyze test coverage. There are about 12 coverage item types like statement, branch, block, decision, condition, method, class, package, requirement, and data flow coverage. Based on the papers gathered, it can be observed that different researchers have targeted different coverage types but only two have used requirement coverage for test coverage analysis. This type of coverage can be explored further to look deeply into coverage analysis and measurement. Table 2 classifies the approaches based on coverage measurement.

Table 2 . Classification of Approaches based on Test Coverage Measurement

Author	Year	Test Coverage Items
Koochakzadeh et al. [19]	2010	Method, Class, Package
Faizah and Suhaimi [20]	2009	Method, Class, Package, Requirement
Angeletti et al. [21]	2009	Branch
Kapfhammer et al. [22]	2008	Code Coverage, Date Flow Coverage
Lingampally et al. [23]	2007	Branch, Block, Method, Predicate
Mehdi et al. [24]	2005	Line, condition, Method
Lormans et al. [25]	2005	Requirement
Diaz et al. [26]	2004	Branch

4.4 Methods/Algorithms

Some researchers developed new methods, models and algorithms to increase and improve the quality of the code coverage. Tsai et al. (2007) defined a coverage relationship model for test case selection and ranking for multi-version software. He proposed a model based adaptive test (MAT) case technique. Li and Asaf used their models to improve the quality of the coverage analysis. In his study, Gao proposed dynamic test coverage analysis solution to monitor API-based component validation. James and Mary presented two new algorithms for test-suite reduction and one new algorithm for test-suite prioritization that can account for MC/DC when reducing and prioritizing test suites. We have shown these authors, year and the issues they discussed in their methods/algorithms in Table 3.

Table 3. Methods/Algorithms

Author	Year	Issues Addressed
Krishnamoorthi et al. [27]	2009	Test case prioritization
Chen et al. [28]	2009	Test data generation for branch coverage
Tsai et al. [29]	2007	Test case selection
J. Jenny Li [30]	2005	Increase code coverage Reduce testing cost
Gao et al. [31]	2005	Test coverage analysis for component validation
Asaf et al. [32]	2004	Improvement for coverage analysis
James and Mary [33]	2003	Test-suite reduction Test-suite prioritization
Tikir et al. [34]	2002	Dynamic instrumentation
Atif Memon et al. [35]	2001	GUI coverage criteria

Table 4. Software Reliability Modeling Study

Author	Year	Issues Addressed
Jinxia et al. [36]	2010	Integrated test coverage
Smidts et al. [37]	2009	Software fault content and location
Haifeng et al. [38]	2008	Fault detection
Xia et al. [39]	2007	Estimation improvement
Xia [40]	2006	Failure intensity function
J.-Y. Park et al. [41]	2006	Coverage growth function
H Pham et al. [42]	2003	Integrated test coverage
Malaiya et al. [43]	2002	Relation between testing time, coverage, and reliability
M. Chen et al. [44]	2001	Testing time reduction

4.5 Software Reliability Modeling

The study regarding software reliability using test coverage has gained much attention in recent years. Many new models have been proposed for software reliability estimation. Most of the researchers believe that time is not the only factor that affects the failure behavior of the software but other important factors must be added to predict the accuracy of software reliability models. Test coverage is believed as an essential factor to enhance these models. Table 4 lists down the authors involved, year, and issues addressed in developing the software reliability models using test coverage. All studies used test coverage with time and other factors to define software reliability models.

4.6 Others

Other research works relating to test coverage measurement and analysis includes a strategy by Ricardo et al. (2010) [45] for test coverage analysis of UML state machines which produces a colored UML model showing the elements covered. Rauf et al. (2010) [46] used genetic algorithm in MATLAB to explore automated GUI test coverage. Takahashi et al. (2008) [47] used coverage criteria for concurrent software testing. The concurrent coverage criteria intend to find concurrent software specific defects, such as race conditions. Gupta and Jalote (2008) [48] used mutation analysis to experimentally evaluating effectiveness and efficiency of coverage criteria for testing.

Whalen et al. (2006) [49] defined test coverage metrics on high-level formal software requirements to support structural or white box testing. Gupta et al. (2000) [50] presented a new program execution based approach to generate input data for branch coverage.

5. Threats to Validity

First, keyword searching and reference analysis were used to get the required research papers. The irrelevant papers were then excluded after reading the title, abstract and conclusions. However, there is a possibility that some papers may be missed due to the above searching and exclusion method.

Second, judgmental errors may disturb in classifying the papers into each category.

6. Conclusion

This study paper has described the research methods, discussed the results of the study and threats to the validity of the study. The systematic process was described in terms of the research questions defined, searching keywords used, the exclusion and inclusion criteria. The results of the study was classified into several categories and analyzed. The paper has shown the areas of research within Test Coverage that have been done by answering the questions that were defined initially. Most of the research papers are from conference and paper proceedings, which indicate that more work needs to be done in order to improve the current state of research in test coverage measurement and analysis.

7. References

[1] ISTQB, "International Software Testing Qualification Board" version 2.0, 2007 www.istqb.org

[2] Grinwald, R., Harel, E., Orgad, M., Ur S, Ziv, A., "User Defined Coverage – A tool Supported Methodology for Design Verification", IBM Research Lab, Haifa Dac, San Francisco, CA USA, 158-163, 1998

[3] Barbara Kitchenham, Stuart Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering, EBSE Technical report 2007, Keel University, UK.

[4] Williams, B. S. a. L. (2008). "A Survey on Code Coverage as a Stopping Criterion for Unit Testing.", Technical report (North Carolina State University. Dept. of Computer Science), TR-2008-22.

[5] Qian Yang, J. Jenny Li, David M. Weiss, "A Survey of Coverage-Based Testing Tools", Published in *The Computer Journal (2009*) , volume 52 (5): pp. 589-597.

[6] M.Prasanna, S.N. Sivanandam, R.Venkatesan, R.Sundarrajan, "A SURVEY ON AUTOMATIC TEST CASE GENERATION", published in Academic Open Internet Journal (AOIJ), Volume 15, 2005

[7] Audris Mockus, N. N. (2009). "Test Coverage and Post-Verification Defects: A Multiple Case Study.", In the 3rd International Symposium on Empirical Software Engineering and Measurement, 2009. ESEM 2009.

[8] Silva, L. and S. Soares (2009)."Analyzing structure-based techniques for test coverage on a J2ME software product line", In 10th Latin American Test Workshop, 2009, LATW '09. Pp.1-6

[9] Y. Wei (2008), "Is Coverage a Good Measure of Testing Effectiveness?", Chair of software engineering ETH Zurich, CH-8092 Zurich, Switzerland.

[10] Derezińska, A. (2008). "Experiences from an Empirical Study of Programs Code Coverage.", Book section, Advances in Computer and Information Sciences and Engineering 2008, 57-62, DOI: 10.1007/978-1-4020-8741-7_11

[11] Jun-Ru, C. and H. Chin-Yu (2007)."A Study of Enhanced MC/DC Coverage Criterion for Software Testing", In 31st Annual International Computer Software and Applications Conference, 2007 COMPSAC 2007

[12] Berner, S., R. Weber, et al. (2007). "Enhancing Software Testing by Judicious Use of Code Coverage Information**",** In 29th International Conference on Software Engineering, 2007. ICSE 2007..

[13] Lloyd, E. L. (2005). "A Study of Test Coverage Adequacy in the Presence of Stubs.", In the Journal of Object Technology, 2005, volume 4, pp.117--137

[14] Joseph Lawrance, S. C., Margaret Burnett‡, Gregg Rothermel (2005). "How Well Do Professional Developers Test with Code Coverage Visualizations? an Empirical study", In the IEEE Symposium on Visual Languages and Human-Centric Computing, 2005 pp.53-60

[15] Kim, Y. W. (2003). "Efficient Use of Code Coverage in Large-Scale Software Development.", In the Proceedings of the 2003 conference of the Centre for Advanced Studies on Collaborative research CASCON '03

[16] Sakamoto, K., H. Washizaki, et al. (2010). "Open Code Coverage Framework: A Consistent and Flexible Framework for Measuring Test Coverage Supporting Multiple Programming Languages**",** In the 10th International Conference on Quality Software, QSIC,2010, pp. 262-269

[17] Matteo Bordin, C. C., Tristan Gingold (2009). "Couverture: an Innovative Open Framework for code coverage analysis of safety critical applications.", Couverture Open R/epository at Open-DO.org, http://forge.open-do.org/projects/couverture

[18] Misurda, J., J. A. Clause, et al. (2005).”Demand-driven structural testing with dynamic instrumentation**”,** In Proceedings of 27th International Conference on Software Engineering, 2005 ICSE 2005, pp. 165-165.

[19] Koochakzadeh, V. G. a. N.,"An Empirical Evaluation to Study Benefits of Visual versus Textual Test Coverage Information.", in the fifth conference on The Testing: Academic and Industrial Conference, Practice and Research Techniques (TAIC PART), 2010

[20] Faizah Omar,S. I,. "A Software Traceability Approach to Support Test Coverage Analysis."In the proceedings of Third International Conference on Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services CENTRIC 2010.

[21] Angeletti, D., E. Giunchiglia, et al. (2009). “Automatic Test Generation for Coverage Analysis of ERTMS Software**”,**Inthe International Conference on Software Testing Verification and Validation, 2009. ICST '09.

[22] Kapfhammer, G.M., Soffa, m.L.(2008) “Database-Aware Test Coverage Monitoring”, Proceedings of the Ist India Software engineering conference ISEC’08, Hyderabad, India, ACM.77-86.

[23] R. Lingampally, A. Gupta, P. Jalote. "A Multipurpose Code Coverage Tool for Java," In Proceedings of the 40th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, 261b, 2007.

[24] M. Kessis, Y. Ledru, G. Vandome. “Experiences in Coverage Testing of a Java Middleware”, in Proceedings SEM 2005, Lisbon, Portugal. ACM, pp. 3945, 2005.

[25] Lormans M, D. (2005). "Reconstructing Requirements Coverage Views from Design and Test using Traceability Recovery via LSI**”,** TEFSE, Long Beach, California, USA, ACM 2005

[26] Eugenia Díaz, J. T., Raquel Blanco (2004). "A Modular Tool for Automated Coverage in Software Testing.", In the Eleventh Annual International Workshop on Software Technology and Engineering Practice, 2003.

[27] R. Krishnamoorthi *, S. A. S. A. M. (2009). "Factor oriented requirement coverage based system test case prioritization of new and regression test cases." ,Published in journal of Information and Software Technology, Volume 51, Issue 4, 2009

[28] Ciyong, Chen, Xiaofeng, Xu, Yan, Chen,“A new method of test data generation for branch coverage in software testing based on EPDG and Genetic Algorithm”,In the 3rd International Conference on Anti-counterfeiting, Security, and Identification in Communication, 2009. ASID 2009

[29] Tsai, W. T., Z. Xinyu, et al. (2007).”A Coverage Relationship Model for Test Case Selection and Ranking for Multi-version Software**”**, In 10th IEEE High Assurance Systems Engineering Symposium, 2007. HASE '07.

[30] Li, J. J. (2005).”Prioritize code for testing to improve code coverage of complex software**”,** In the 16th IEEE International Symposium on Software Reliability Engineering, 2005. ISSRE 2005, 10-pp..

[31] Gao, J., R. Espinoza, et al. (2005).”Testing coverage analysis for software component validation”, In the 29th Annual International Computer Software and Applications Conference, 2005. COMPSAC 2005.

[32] Asaf, S., E. Marcus, et al. (2004).”Defining coverage views to improve functional coverage analysis**”,** In the Proceedings of 41st Design Automation Conference, 2004.

[33] Jones, J. A. and M. J. Harrold (2003).”Test-suite reduction and prioritization for modified condition/decision coverage”, In the journal of IEEE Transactions on Software Engineering, Volume 29 Issue 3, March 2003.

[34] Tikir, M. M. and J. K. Hollingsworth (2002).” Efficient instrumentation for code coverage testing**”,** In Proceedings of the ACM SIGSOFT 2002 International Symposium on Software Testing and Analysis, pp.86-96.

[35] Atif Memon, “Coverage Criteria for GUI Testing”, In ACM SIGSOFT Software Engineering Notes. Vol. 26, no. 5, pp. 256-267. Sept. 2001

[36] Jinxia An, J. Z. (2010). "Software Reliability Modeling with Integrated Test Coverage.", In the Fourth International Conference on Secure Software Integration and Reliability Improvement (SSIRI), 2010 pp. 106 – 112.

[37] Smidts, C. (2009).”A Test Coverage-Based Model for Predicting Software Fault Content and Location during Multi-phase Functional Testing”, In International Conference on Advanced Software Engineering & Its Applications (ASEA 2009), Korea

[38] Haifeng, L., L. Qiuying, et al. (2008).”Software Reliability Modeling with Logistic Test Coverage Function**”,** In 19th International Symposium on Software Reliability Engineering, 2008. ISSRE 2008.

[39] Xia, C. and M. R. Lyu (2007). Software Reliability Modeling with Test Coverage: Experimentation and Measurement with A Fault-Tolerant Software Project**”,** In the 18th IEEE International Symposium on Software Reliability, 2007. ISSRE '07.

[40] CAI Xia, (2006), “Coverage Based Testing Stratigies and Reliability Modelling for Fault-Tolerant Software Systems”, in CSA, Dissertation Abstracts International. Vol. 68, no. 3. 2006

[41] J. -Y. PARK, (2006), “COVERAGE GROWTH FUNCTIONS FOR SOFTWARE RELIABILITY MODELING”, Book article, Advanced Reliability Modeling, II, Reliability Testing and Improvement, (pp 435-442).

[42] H Pham, (2003), “NHPP software reliability and cost models with Test coverage”, In the European Journal of Operational Research Volume 145, Issue 2, 1 March 2003, Pages 443-454.

[43] Malaiya, Y. K., M. N. Li, et al. (2002). “Software reliability growth with test coverage”, In Journal IEEE Transactions on Reliability. Vol. 51, no. 4, pp. 420-426. Dec. 2002.

[44] Mei-Hwa Chen, Michael R.,(2001), ”Effect of of Code Coverage on Software Reliability Measurement”, in journal IEEE Transactions on Reliability, vol.50, no.2, 2001, pp.165-170.

[45] Ricardo D. F. Ferreira, J. P. F., Ana C. R. Paiva (2010). "Test Coverage Analysis of UML State Machines.", In the International Conference on Software Testing, Verification and Validation Workshops, 2010. ICSTW '10. , pp. 284-289.

[46] Rauf, A., S. Anwar, (2010). “Automated GUI Test Coverage Analysis Using GA”**,** Proceedings of the 2010 Seventh International Conference on Information Technology: New Generations, pp. 1057-1062.

[47] Takahashi, J., H. Kojima, et al. (2008). “Coverage Based Testing for Concurrent Software**”,** In 28th International Conference on Distributed Computing Systems Workshops, 2008. ICDCS '08.

[48] Jalote, A. Gupta. P. (2008). "An approach for experimentally evaluating effectiveness and efficiency of coverage criteria for software testing.", In International Journal on Software Tools for Technology Transfer (STTT), Volume 10, Number 2, 145-160, DOI: 10.1007/s10009-007-0059-5

[49] Whalen, M. W., M. P. E. Heimdahl, et al. (2006). “Coverage metrics for requirements-based testing.”2006, In Proceedings of the 2006 International Symposium on Software Testing and Analysis, ISSTA 2006, pp. 25-35.

[50] Gupta, N., Mathur, A. P, Soffa, M. L. (2000), “Generating test data for branch coverage”, In The Fifteenth IEEE International Conference on Automated Software Engineering, 2000. ASE 2000.