

A REVIEW ON CODE COVERAGE ANALYSIS

SARITA PATHY¹

P.G. Dept. Of Computer Science and Application
Jyoti Vihar, Sambalpur University, Burla, Sambalpur, Odisha, India
¹pathy.sarita60@gmail.com

SANTOSH PANDA²

P.G. Dept. Of Computer Science and Application
Jyoti Vihar, Sambalpur University, Burla, Sambalpur, Odisha, India
²Santosh.panda@live.in

SARADA BABOO³

P.G. Dept. Of Computer Science and Application
Jyoti Vihar, Sambalpur University, Burla, Sambalpur, Odisha, India
³baboo.sarada@gmail.com

Abstract-The quality of any software predominantly depends upon the software testing. Code coverage testing is one such technique that plays an important role to achieve the software for better quality and maintenance. The code coverage evaluates the testing effectiveness and provides data on different coverage items. Code coverage analysis measure what amount of source is covered by text execution. Using structural coverage, architectural coverage and requirement coverage the complete software testing can be measured. This review focus on techniques of structural coverage testing that is also known as code coverage testing. The code coverage can be done by some special analysis tools which are known as code coverage tools.

Keywords: Branch Testing, Code Coverage Testing, Code Coverage Tools, Statement Testing.

I. INTRODUCTION

Software testing is a process to identify defects and alleviate their associated effects, which is used to specify the quality of software. Software testing plays an important role in the software development process. One of the important aims of the software testing process is to inform a high percentage of testing coverage for a given program [1]. Testing coverage represents criteria that are used to measure the completion of the testing process. To obtain test coverage various testing activities can be performed. Code testing coverage shows which parts of code, for a given program, is checked by at least one test case [2].

The program source code is measured by the code coverage for a particular test suite. As compare to low code coverage the high code coverage is thoroughly tested and has the lower chances of software bugs [3]. The code coverage provides the percentage of verified code and also the uncovered area of code. It also gives the information to the user about the status of the verification process. Different types of metrics can be used to calculate code coverage.

II. APPLICATION AREA OF CODE COVERAGE ANALYSIS

- The test quality is maintained during phases of software development.
- By testing our code we know how robust our tests.
- To increase coverage, code coverage creates the additional test cases.
- Code coverage finds the parts of the program are not covered by the test cases.
- The quality of the product is determined by the quantitative measurement of code coverage.

III. DEMERITS OF CODE COVERAGE ANALYSIS

- The code coverage analysis can only measure the code itself that is what has been written and not measures what has not written in the software.
- The structure based techniques does not said anything about the specified functions which has not been implemented or omitted from the specification ,it only defines the structures which is already there [4].

IV. ISSUES AND CHALLENGES ON CODE COVERAGE

Although statements and branch coverage is available universally and reliable but various coverage tools are handle only statement coverage. There are various limitations of code coverage tools in the coverage it performs. Other coverage like condition and path coverage will not provide the complete information.

V. BASIC DEFINITIONS AND CONCEPTS

Code coverage uses in the following testing process

A. Black box testing

It does not required to know any internal knowledge of code .Black box testing examines the functionality of a system without having the knowledge of internal logic of code and the tester only knows the inputs and what excepted outputs.

B. White box testing

White box testing is also called as glass testing and structural testing. It concerns with the internal knowledge of coding. Code coverage is a white box testing methodology. The code coverage is mostly used in module testing and it is also used in regression testing and integration testing [5].

VI. CODE COVERAGE METRICS

There are various types of code coverage metrics are exists.

A. Statement Coverage or Line Coverage

The main idea behind the statements coverage is that it is very difficult to determine the error in the statements unless a statement is executed; it is very difficult to achieve 100% statements coverage [6].

Example:

```
Int gcd(m,n)
Int m,n;
{
While (m!=n)
{
if (m>n) then
m=m-n;
else
n=n-m;
}
Return m;
}
```

By choosing the test set {(m=5, n=5), (m=6, n=5), (m=5, n=6)} all statements of the program executed at least once. We can find out the statement coverage in the following way

Statement coverage= (total statements exercised/total number of executable statements in program)*100

There are four types of statements are as follows

1. Simple statement is also known as sequence flow.
 2. Selective or conditional statements like if then else are also known as two way statements.
 3. Nesting of if else statements or switch statements are also known as multi way statements.
 4. Iterative statements like while, do, until, repeat, for statements are also known as loop statements.
- 1 .Simple statements are not containing any type of loop and branch. In simple statements the test cases are designed in such a way that, each statement can be executed sequentially from top to bottom.
 2. For two away statements at least one test case is designed for else part and one test case is designed for then part.
 3. In a multi way statements possible switch cases are tested. Test cases are designed to cover all possible switch cases in multi way statements.
 4. The most commonly error can be occurred in the iterative statements due to improper defining the boundary condition and terminated condition of the loop.

B. Branch Coverage Analysis:

In branch coverage analysis each branch in the program is executed at least once during testing, branch indicates the loop, if statements, while statements etc [6].

example:

```
If(r == true)
{
```

```

hello
}
else
{
Good morning
}

```

The advantages of branch coverage is when a program can jump, it jumps to all of its possible destinations.

C. Block Coverage

An alternative of Statement coverage is Block coverage, every code block contains a known set of statements .In test quality point of view there is a one to one relationships is established in between statement and block coverage.

D. Path Coverage

Path coverage testing is used to coverage every possible logical path in the program at least once [7]. A path is a unique series of logical conditions that means, full path coverage analysis require attempt every combination of logical conditions within the program. For example, in code below

```

if (a)
{
working
}
if (b)
{
Working else
}

```

In this a full path coverage analysis is done by setting the value of (a,b) as (0,0), (0,1),(1,1), (1,0).

E. Condition Coverage

It is used for Boolean expression, to check whether it is TRUE OR FALSE for each condition in the sub expression of a conditional statement. It checks the conditions independently of each other.

F. Function Coverage Analysis:

It involves each and every function or procedure call of the code at least once, it does not said anything about what was executed inside it.

G. Relational Operator Coverage

In most cases the faults originate due to misuse of relational operators like interchanging of operator or incorrect boundary values [7].

VII. SOFTWARE TESTING TOOLS

Code coverage tools are eclipse plug-in that integrate with Junit testing frameworks. These tools are available for public used that is both for industrial and research fields. The tools are as follows.

Ecobertura: it is a free eclipse plug-in that used to calculate the percentage of line and branch coverage metrics for package, class and for over all of program. The source code are executing through Junit frameworks, this tools show the result of testing coverage process. The details of the coverage can be viewed with different colors in the source file [8].

Percentage of code coverage can be evaluated within the eclipse IDE interactive development environment by selecting the "Cover As" from the run menu.

JaBuTi: JaBUTi is a complete toolsuite for understanding and testing Java programs and Java-based components. JaBUTi can be used to perform coverage analysis using different testing criteria, and find outs the faults. JaButi provides a way to evaluate the coverage with respect to both control flow and data flow based criteria [9].

CodeCover: CodeCover is a free white-box testing tool, it measures statement, branch, loop, Boolean coverage. It provides the open language interface for Java and COBOL. It eliminate the duplicate test cases from a test suite. CodeCover run on Command line such as Linux, Windows, Mac OS [10].

Emma: EMMA is an open-source java tool that is used for measuring and reporting Java code coverage. It supports class, method, line, basic block (statements) coverage. EMMA can detect when a single source code

line is covered only partially. It provides analysis report in plain text, HTML, XML format . This tool generates a variety of reports and charts in HTML, XML, and CSV standard [11].

JCover: It is a code coverage testing tool for JAVA. It is used for statement and branch Coverage analysis, and it computes Method, Class, File, and Package coverage. Jcover can gather test coverage measures of applications whose source code is available, or to work with compiled class files bytecode [12].

VIII. A REVIEW OF CODE COVERAGE ANALYSIS

This chapter presents an overview of the existing method and technique to testing and code coverage analysis. First, we discuss the previous related work done by researchers on the topic of code coverage analysis.

A.M.R. Vincenzi *et al* purposed a technique that based on control flow, data flow based coverage criteria and a software testing tool named JaButi that supports the testing of java programs .control flow testing is described using all nodes and all edges criteria with the help of the def-use graph representation of a program. Data flow based testing is explain based on all-uses criteria comprising of all p-uses and c-uses criteria [13].

Srinivas Nidhra et al purposed that a techniques that related to both black box and white box testing. It can derive the test data and test case for white box testing methods such as branch testing, statement testing, condition and multiple condition testing and equivalence partitioning and the boundary value analysis in black box testing and derive the testing techniques along with the case situation and their advantages [14].

Saran Prasad et al purposed a new Test Suite Reduction approach technique in which the data set can be cuts short on the basis of some criteria in iteration and of computational complexity of test suite reduction problem. By using multiple coverage criteria like line, branch ,function coverage reduce the real world regression test suite, without any loss of percentage of code coverage and fault detection capability[15].

Ilona Bluemke et al purposed the concept of DFC, an eclipse piug-in JaButi tool then designed and implement supporting data flow testing of java methods. Data flow based testing helps to finding the error which has not been detected by black box testing. The design test cases are depends on the number of detected errors.DFC also checks whether the test cases are covered all specific path or not. Methods are called by using objects jaButi tools. In future it extends to the intra method criteria to inter method criteria [16].

H.singh et al. used the control-flow and data-flow criteria to support the testing of Java programs (Java bytecode) aimed at intra-method structural testing of code and its components based on various testing criteria. Used a testing tool, named JaBUTi (Java Bytecode Understanding and Testing), which supports the application of such criteria for testing Java programs and components, which is used for structural testing and code coverage analysis [17].

Nan Li, Xin Meng et al. Experiment on 31 java code coverage tool and found four of them is active and support to measure the branch coverage. Here used one byte code instrumentation tools elemma, two source code instrumentation tools code cover and clover, analysed using this tool found that byte code instrumentation is not a valid technique to measure branch coverage. So more tools can be support statement coverage [18].

Muhammad Shahid et al purposed a new algorithm for test case prioritization that is based on the code coverage of the test cases. Purposed a regression testing and regression test prioritization techniques and tools based on coverage metrics. A new regression test cases priority technique that arranges the selected test cases based on their code coverage percentage and find outs the faults earlier [19].

IX. SUMMARY OF VARIOUS CODE COVERAGE ANALYSIS

Sl.no	Source	Year	Auther	Methodology	Results	Future scope
1.	IJESA	2005	A.M.R. Vincenzi et al.	control-flow and data-flow based coverage criteria, line coverage.	Data flow and control flow based criteria is used to measure the java source code. JaButi tool provides the coverage reports with respect to for each class, report and method. It can be used by both the component developer and the component user to test Java-based components and systems.	Future the JaButi apply to industry projects to find out cost and benefits of control and data flow based criteria in testing java programs and components and performed integration testing.
2.	IJESA	2012	Srinivas Nidhra	Statement, branch, condition coverage ,control flow ,data flow, code walk through techniques.	Finding and removing logical errors and achieving the full path coverage and clearly explain different testing techniques along with a case situation and their advantages.	future work is to check the usability and usefulness of each and every technique from state-of-practice.
3.	IJAIC	2012	Saran Prasad et al.	Line ,branch, function call coverage	developed a new Test Suite Reduction approach which is innovative in the sense that it cuts short the data set on the basis of some criteria in iterations and does comparisons and helps in reducing the computational complexity of test suite reduction problem.	
4	IJESA	2013	Ilona Bluemke et al.	DFC(data flow coverage) tool, data flow based technology.	Data flow testing of java classes, provide opportunities to find error that may not be uncovered by black box testing.DFC also checks the specific path derived from the test cases are covered.	Apply DFC to industrial projects to found cost and benefits in data flow based criteria of testing of java programs. Extend the <i>intra-method</i> testing criteria to wider, <i>inter-method</i> level, so more error could be detected.

5		2011	H.singh	Control flow,data flow criteria, JaButi tool	done the code coverage analysis mostly on All-Nodes-ei criteria. Obtain high code coverage of Java programs in minimum test runs.	In future the code coverage analysis of Java programs based on other criteria like All-edges-ei, All-uses-ei, All-Pot-Uses-ei, All-Nodes-ed, can be implemented, JaBUTi, can further be implemented on programs containing GUI components.
6	ISSRE	2013	Nan Li, Xin Meng et al	Branch coverage, EclEmma CodeCover and Clover tool.	Bytecode instrumentation is not a valid technique to measure branch coverage. More tools can be support statement coverage.	Perform this in similar s statement coverage. The second is to extend the study to more tools. A third is to try to measure which tool will help testers design better quality tests.
7.	IJESA	2014	M.Shahid et al	Regression testing techniques	Finds on test case prioritization techniques for code coverage. Prioritized test cases can save lot of efforts, time and find out faults earlier.	In future the algorithm is applied on large system to verify its correctness.

Methodology	Advantages	Disadvantages
Control flow and data flow testing using JaButi tool.	Does not require any source code only requires java byte code. Generating different testing reports and evaluates quality of current test set.	Does not measure the implementation aspects of the components.
Black box and white box testing techniques(equivalence class portioning,code walk through,control flow,data flow testing)	Finding and removing logical errors and helps to selecting a test inputs with a high chance of detecting a defect. Achieving a full path coverage.	This technique does not validate for industrial perspective.
Data flow testing using DFC(data flow coverage)tool	Identifying the errors from the test cases and check the specific paths derived from test cases are covered.	Time consuming, not fully automatic.
Test case prioritization in Regression testing	Increasing the effectiveness of testing, finding faults and saving time and cost.	Does not use for the large software systems, difficult to select the test cases.
Branch coverage using EclEmma,CodeCover,Clover tool	Each branch of the program is executed. Gives maximum path coverage.	Only limited tool support the branch coverage.

X. COMPARISON TABLE ON CLASSIFICATION OF APPROACHES ON TEST COVERAGE MEASUREMENT

Author	Year	Test Coverage Items
Koochakzadeh et al. [20]	2010	Method, Class, Package
Faizah and Suhaimi [21]	2009	Method, Class, Package, Requirement
Angeletti et al. [22]	2009	Branch
Kapfhammer et al. [23]	2008	Code Coverage, Data Flow Coverage
Lingampally et al. [24]	2007	Branch, Block, Method, Predicate
Mehdi et al. [25]	2005	Line, condition, Method
Lormans et al. [26]	2005	Requirement
Diaz et al. [27]	2004	Branch

CONCLUSION

After reviewing different papers it has presented the usability of test code coverage analysis. It is found that the techniques code coverage for statement and branch testing under white box testing is still to study the quality and the usability of these approaches using free and open source testing tools eCobertura under Java Eclipse IDE. Using statement and branch coverage researchers measure the quality of written code and checks the flow of different paths in the program and it also ensure that whether those path are tested or not.

REFERENCES

- [1] H. Do, S. Elbaum and G. Rothermel, "Supporting controlled experimentation with testing techniques: An infrastructure and its potential impact", Empirical Software Engineering: An International Journal, vol. 10, no. 4, pp. 405-435, 2005
- [2] K., H. Washizaki, et al. "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, pp. 262-269 QSIQ, 2010,
- [3] http://en.wikipedia.org/wiki/Code_coverage
- [4] Sneha Shelke, Sangeeta Nagpure "The Study of Various Code Coverage Tools" International Journal of Computer Trends and Technology (IJCTT) – volume 13 number 1 – Jul 2014
- [5] http://pjcj.net/testing_and_code_coverage/paper.html
- [6] R. M. Sharma "Tools And Techniques Of Code Coverage Testing", International Journal Of Computer Engineering & Technology (Ijcet), 2014
- [7] http://en.wikipedia.org/wiki/Code_coverage
- [8] Ecobertura. Available at: <http://www.ecobertura.org/index.html>
- [9] R. M. Sharma "Tools And Techniques Of Code Coverage Testing", International Journal Of Computer Engineering & Technology (Ijcet) 2014
- [10] Code Cover, <http://www.codecover.org>
- [11] Emma, <http://emma.sourceforge.net/>.
- [12] JCover, <http://www.mmsindia.com/JCover.html>
- [13] A.M.R. Vincenzia, et al. "Coverage testing of Java programs and components" Science of Computer Programming 56, 2005 .
- [14] Srinivas Nidhra1 et al. "black box and white box testing techniques –a literature review" International Journal of Embedded Systems and Applications (IJESA) Vol.2, No.2, June 2012
- [15] Saran Prasad "Regression Optimizer A Multi Coverage Criteria Test Suite Minimization Technique" International Journal of Applied Information Systems (IAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 1– No.8, April 2012 .
- [16] Ilona Bluemke, Artur Rembiszewski "Dataflow testing of Java programs with DFC" International Journal of Embedded Systems and Applications (IJESA), Institute of Computer Science, Warsaw University of Technology Nowowiejska 15/19, 00-665 Warsaw, 2013.
- [17] H.singh et al. "Code Coverage Analysis of Object-Oriented Programming" Bachelor of Technology thesis, Department of Computer Science and Engineering National Institute of Technology Rourkela, 2011
- [18] Nan Li, Xin Meng, Jeff Offutt, and Lin Deng "Is Bytecode Instrumentation as Good as Source Code Instrumentation: An Empirical Study with Industrial Tools" ISSRE 2013
- [19] Muhammad Shahid et al "A New Code Based Test Case Prioritization Technique " International Journal of Software Engineering and Its Applications Vol.8, No.6 (2014)
- [20] 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
- [21] 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.
- [22] Angeletti, D., E. Giunchiglia, et al. "Automatic Test Generation for Coverage Analysis of ERTMS Software", In the International Conference on Software Testing Verification and Validation, 2009.
- [23] Kapfhammer, G.M., Soffa, M.L. "Database-Aware Test Coverage Monitoring", Proceedings of the 1st India Software engineering conference ISEC'08, Hyderabad, India, ACM.77-86, 2008.
- [24] 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.

- [25] M. Kessis, Y. Ledru, G. Vandome. "Experiences in Coverage Testing of a Java Middleware", in Proceedings SEM , Lisbon, Portugal. ACM, pp. 3945, 2005.
- [26] Lormans M, D. "Reconstructing Requirements Coverage Views from Design and Test using Traceability Recovery via LSI", TEFSE, Long Beach, California, USA, ACM 2005
- [27] Eugenia Díaz, J. T., Raquel Blanco , "A Modular Tool for Automated Coverage in Software Testing.", In the Eleventh Annual International Workshop on Software Technology and Engineering Practice, 2003.