Assessment on Various Code Coverage Tools

R. Sivaguru
Assistant Professor
Department of Computer Science and Engineering
Narasu's Sarathy Institute of Technology

G. V. Kanimozhi
Assistant Professor
Department of Computer Science and Engineering
Narasu's Sarathy Institute of Technology

S. Alaudeen Basha
Associate Professor
Department of Computer Science and Engineering
Narasu's Sarathy Institute of Technology

Abstract

Software Testing one of the challenging tasks is to select the test inputs. Code coverage techniques are used to express the amount to which the code of a program is tested. Many coverage tools are available in working environment. Here we have studied seven code coverage tools and out of which one tool was actually evaluated for their proposed features. A comparative study is presented on the basis of the set criteria.

Keywords- Software Testing, Code Coverage, Code Coverage Tools, Code Coverage Criteria

I. INTRODUCTION

Software testing is a process of assuring a program is bug free and also that it performs the intended functions which are error free. Software testing is used to indicate the software quality. Testing activities include obtaining the test coverage. Code coverage method is a way of ensuring that your tests are really testing your source code. When you performed your tests you are actually inspection that you are getting the estimated results.

The output of coverage measurement can be used in several ways to improve the testing process. It also gives the information to the user about the status of the verification process. It can help to find areas that are not covered.

Test coverage is used to measure how the software is tested and developers use it to point out their assurance in the willingness of their software. “A Survey of Coverage-Based Testing Tools” studies and compare 17 coverage-based testing tools mostly focusing on, but not limited to, coverage depth. All tools included in this survey have coverage measurement capability. This survey compares tools released before 2007 for three important coverage tool characteristics.

There are several tools in order to facilitate the software testing process, and they have different functionalities. Our objective in this paper is to study the tools with code coverage capabilities which are released after 2007.

We selected test tools with code coverage capabilities. We have selected 7 tools out of which we have evaluated 1 tool and all other tools are studied and compared based on the literatures available.

This paper organized as follows. The section II describes the overview of the coverage. Section III describes the 5 code coverage tools. In section IV we have compared these tools based on three measurement criteria: supported programming languages, coverage measurement criteria, programming instrumentation and automation. Finally section V summarizes the work.

II. CODE COVERAGE

Code Coverage method is the process of finding areas of a program which are not exercised by set of test cases, which creates additional test cases to increase coverage and determines quantitative measure of code coverage. Coverage based testing tool can be applied to any stage of testing including unit, integration or system testing.

Code coverage provides quantification of coverage related test progress, it prioritize the testing by selecting those tests that has largest incremental gain in coverage. It detects redundant cases and removes those cases since these much time to execute repeatedly.

By using the code coverage testing process can be improved and cost of correcting the errors can be reduced. Benefits of Code Coverage measurement:

- To sustain the test value over the life cycle of a project
- It will teach how well test our code
- It helps to finding areas of a programs not exercised by total test cases
- To determining a quantitative measure of code coverage quality of the particular application and product.

Drawback of Code Coverage measurement:

- Many of the measurements function are not been implemented

All rights reserved by *www.grdjournals.com*

- Structure-based techniques cannot say anything about new things. It only looks at a structure which is already present.
- The code itself it will perform.
- It cannot declare anything about the software that has not been written.

III. CODE COVERAGE TOOLS

A. OPEN CODE COVERAGE FRAME WORK (OCCF)

There are many programming languages and coverage criteria exist, and every coverage measurement tools support programming language and the coverage criteria. So many tools exist and they have various programming languages that lead to difference between the existing tools. To overcome the diversity of existing tool, a novel approach for measuring the coverage for multiple programming languages called open code coverage framework.

B. DYNAMIC CODE COVERAGE (DCC)

A Dynamic Code Coverage is an easy-to-use tool that indicates which source code is exercised during one or more executions of a program. This information is invaluable in determining how thoroughly a test suite exercises a program.

C. PET, JPET

Partial evaluation based test case generation (PET) for byte code transforms java byte code to equivalent CLP code, and generates test cases from it. It is implemented in program log, jPET is an eclipse Graphic User Interface (GUI) for it.

D. RANDOOP

Randoop tool are executes the sequences of it creates, using the results of the execution of the create assertions that capture the actions or the program and to catch bugs. Randoop generates unit testing using feedback directed random test generation.

E. AUTOMATIC ROBUSTNESS COVERAGE ANALYSIS TOOL (AURORA)

AURORA is a tool that provides testers with the ability of computing the complete coverage achieved by a certain test suite ps over a program t in an automated way. The tool accepts code transformations defined by means of the TXL language, and uses standard coverage measurement libraries to compute the coverage achieved by ps on t, and using the transformations it automatically computes the fragility indexes.

F. JAVACODECOVERAGE (JACOCO

JavaCodeCoverage is a byte-code analyzer tool for test coverage analysis for Java software which neither requires neither the language grammar nor the source code. An important aspect of JavaCodeCoverage is that it stores the coverage information for individual test case thereby facilitating detailed coverage analysis. Another important aspect of JavaCodeCoverage is that it records all vital code-elements and test coverage information in open source database software MySQL

G. EVOSUITE

To find defects in software, needs test cases that execute the software methodically and oracles that appraise the correctness of the practical performance when organization these test cases. EvoSuite is a tool that unhappily generates test cases with assertions for classes printed in Java code. Achieve this, EvoSuite tool apply a crossbreed approach that generates and optimizes complete test cases towards filling a coverage criterion. For the created test suites, EvoSuite tool suggests potential oracles by adding small and valuable sets of assertions that quickly summarize the current activities; these assertions allow the developer to perceive deviations from expected activities, and to capture the current activities in order to protect against future defects flouting this behavior.

IV. COVERAGE MEASUREMENT CRITERIA

All tools support coverage measurement capability. It consists of supported languages, program instrumentation, coverage measurement table, automation. We have studied 7 code coverage tools out of these we have evaluated java code coverage tool and all other tools studied based on the available literature.

A. Supported Languages

Every tool supports programming languages. Some of them support only java, some of them support only C/C++, and some of them support both java and C/C++, some of them supports
FORTRAN, C#.NET. Table I shows a list of tools and the languages they support

The selection of supported languages reflects each company's target industries. EvoSuite is a tool that mechanically generates test cases and classes written in Java code. Java Code Coverage is a byte-code analyser tool for test coverage analysis for Java software which requires neither the language grammar nor the source code.

To overcome the diversity of existing tools, a novel framework developed for consistently and flexibly measuring the coverage supporting multiple programming languages, called Open Code Coverage Framework (OCCF).

All rights reserved by *www.grdjournals.com*

Table 1: Support Language

Tool Name	*C/C++*	*Java*	*Other*
OCCF	☐	☐	☐
DCC	☐	--	--
PET,jPET	--	☐	--
Randoop	--	☐	--
AURORA	--	☐	--
JaCoCo	--	☐	--
EvoSuite	--	☐	--

B. Program Instrumentation

Code coverage testing tools imprison coverage information by monitoring program execution. Execution is monitored by inserting probes into the program before or during its execution. A probe is typically a few lines of code that, when executed, generate a record or event that indicates that program execution has passed through the point where the probe is located. There are two kinds of overhead associated with instrumenting a program with probes:

- The off-line overhead

It cannot be used source code is not available. They are most efficient in terms of compilation time but less portable.

- The run time overhead

The tools which are provided for system software or embedded software, they tend to focus on reducing the run time overhead, so their tools can be usable in real time environment.

The EVOSUITE tool implements the approach presented for generating JUnit test suites for Java code. EVOSUITE works on the byte-code level and collects all necessary information for the test cluster from the byte-code via Java Reflection. This means that it does not require the source code of the SUT and in principle is also applicable to other languages that compile to Java byte-code.

OCCF inserts instrumentation code into source code. The abstract syntax trees of source code for most programming languages have similar structure. Thus OCCF provides a reusable common code to insert instrumentation code through AST's by utilizing the similarities.

Table 2: Program Instrumentation

Tool Name	*Source code Instrumentation*	*Byte Code Instrumentation*
OCCF	☐	--
DCC	☐	
PET,jPET	--	☐
Randoop	--	☐
AURORA	--	☐
JaCoCo	--	☐
EvoSuite	--	☐

Table 3: Coverage Measurement Criteria

Tool Name	*Statement/ Line*	*Branch/ Decision*	*Method/ Function*
OCCF	☐	☐	--
DCC	☐	☐	☐
PET,jPET	☐	☐	☐
Randoop	☐	--	☐
AURORA	☐	--	--
JaCoCo	☐	☐	☐
EvoSuite	--	☐	☐

V. AUTOMATION

Automation Testing is used to repeats the test scenarios that were performed physically, speedily and repeatedly. Automation of testing process includes number of steps such as test case generation, test execution and creation of test oracles It increases the entire test coverage; develop accuracy, saves time and money in association to manual testing. Automated test generation tends to be linked with code coverage, i.e. the goal of generating test automatically can easily be linked to the goal of increasing coverage. EvoSuite is a tool that mechanically generates test cases and classes written in Java code.

All rights reserved by *www.grdjournals.com*

VI. CONCLUSION

We have studied 7 code coverage-based testing tools. Our Study includes the comparison of three features: Code coverage measurement, Coverage criteria, Automation Out of these we have evaluated java code coverage tool are evaluated basis on the paper. Java code coverage tool effectively used for bug place identification as well as decision/condition coverage evaluated both as true and false.

REFERENCES

[1] 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.
[2] 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.
[3] G. Fraser and A. Arcuri, "Evolutionary Generation of Whole Test Suites," Proc. 11th Int'l Conf. Quality Software, pp. 31-40, 2011.
[4] G. Fraser and A. Arcuri, "Evosuite: Automatic Test Suite Generation for Object-Oriented Software," Proc. 19th ACM SIGSOFT Symp. and the 13th European Conf. Foundations of SoftwarEng., 2011
[5] R. Lingampally, A. Gupta, P. Jalote. "A Multipurpose Code Coverage Tool for Java," In Proceedings of the 40thAnnual Hawaii International Conference on System Sciences, IEEE Computer Society, 261b, 2007.
[6] Angelo Gargantini, Marco Guarnieri and Eros MagriAURORA: AUtomaticRObustnesscove Rage Analysis Tool in 6th IEEE International Conference on Software Testing, Verification and Validation - Testing Tools Track (ICST 2013)
[7] Angelo Gargantini, Marco Guarnieri, Eros MagriExtending Coverage Criteria by Evaluating their Robustness to Code Structure Changes in 23rd International Conference on Testing Software and Systems (ICTSS 2012 - Acceptance rate: 33%)

All rights reserved by *www.grdjournals.com*