

Advanced coverage reporting

1 Context

Several tools report statistics about how a test suite covers a program under test. For example, Jacoco summarizes coverage statistics about the coverage of a JUnit test suite, at different granularity levels (package, class, method): <http://www.eclemma.org/jacoco/trunk/coverage/>. However, current tools ignore a large number of dynamic information and do not report them. The goal of this project is develop a technique that instruments a test suite and a program in order to collect and report on dynamic information about test suite coverage.

2 Project description

Current statement or branch coverage is a boolean information: a test case covers or not a statement. Yet, when a statement is covered it can be covered several times, it can be covered "deep" in the execution, the statement can be buried in deep nested loops and conditions, it can be covered by several different test cases, a predicate can be evaluated with different values, etc.

The goal of this project is to develop a prototype tool that collects advanced coverage information for statements and predicates in a program, and generates reports about these data. Students should select several metrics mentioned above, or define some of their own. Then, they shall define what probes they must inject in the source code of the test cases and the program under test to collect the necessary information. Last, they shall develop a prototype and make it run on a couple of case studies.