Static Estimation of Test Coverage

Tiago L. Alves
Universidade do Minho, Portugal, and
Software Improvement Group, The Netherlands
t.alves@sig.nl

Joost Visser
Software Improvement Group
The Netherlands
j.visser@sig.nl

Abstract

Test coverage is an important indicator for unit test quality. Tools such as Clover can compute it by first instrumenting the code with logging functionality, and then log which parts are executed during unit test runs. Since computation of test coverage is a dynamic analysis, it presupposes a working installation of the software.

In the context of software quality assessment by an independent third party, a working installation is often not available. The evaluator may not have access to the required software libraries or hardware platform. The installation procedure may not be automated or documented.

We investigate the possibility to estimate test coverage at system, package and class levels through static analysis only. We present a method using slicing of static call graphs to estimate the actual dynamic test coverage and we identify the sources of imprecision. The method estimates test coverage at method level, by computing all methods reachable from tests. Coverage at method level is then used as basis for estimating class and package coverage.

We validate the results of static estimation by comparison to actual values obtained through dynamic analysis using Clover. The comparison is done using 12 systems both proprietary and open-source, with sizes ranging from small to large. Additionally, we apply our technique to 52 releases of a proprietary system.

We report that static estimation of test coverage at system level is highly correlated with dynamic coverage, demonstrating that static estimation can be a good predictor for the actual coverage.