Evaluation of Automated Testing Coverage: a Case Study
Testing of Secure Wireless Connection Software

S. Vilkomir\textsuperscript{1}, P. Tips\textsuperscript{2}, D. Parnas\textsuperscript{1}, J. Monahan\textsuperscript{2}, T. O'Connor\textsuperscript{2}

\textsuperscript{1}Software Quality Research Laboratory (SQRL), University of Limerick, Ireland
\textsuperscript{2}Dell Products, Limerick, Ireland

Abstract

We discuss methods used to increase the effectiveness of automated testing by assuring that all major partitions of the input domain are adequately tested. Our case study is the testing of wireless secure connection software to guarantee high quality products. The software tested is intended to provide secure wireless connections between a computer (with either an external wireless card or an internal mini pc\textsuperscript{i} wireless card) and any supported Dell access point.

We consider the following aspects:

1. Approach to automated testing of wireless secure connection software. Testing of wireless secure connection software is required every time that a new model of a wireless card or access point is launched. Previously, testing has been manual using a fixed set of different cryptography keys as test cases. The project being described uses automated testing. The initial test set is kept for the purpose of regression testing and additional, randomly generated, test cases are used to ensure thorough test coverage. Our approach to development of test scripts and their structure is discussed.

2. Formalisation and tabular representation of requirements to test cases. The principal attributes of test cases are:
   \begin{itemize}
   \item Type of security algorithm (WEP - WPA)
   \item Symbols of security keys (ASCII - HEX)
   \item Length of security keys (40 bit - 104 bit)
   \item Validity (valid - invalid), etc.
   \end{itemize}
To develop a tabular representation of test case requirements, we divided the attributes into two orthogonal classes and used one class as row headers and the other class as column headers. Cells of the table represent the different groups of test cases that should be used during testing. For this product, we have identified 102 different groups of test cases. Test requirements are specified by putting integers, representing the required number of tests, into each cell. We present an example of such a table.

3. Using formalized requirements for test coverage evaluation. To analyze coverage of the initial test set for regression testing, we classify the tests according to the table and display the coverage in tabular form. Where a cell entry is zero, special consideration is required. Often this means that the corresponding situation has been overlooked and we need to add test cases to improve coverage.

4. Dynamic test coverage evaluation during automated testing. We use the same tabular representation of requirements to test cases to generate random test cases during automated testing. The table is automatically filled in during testing to reflect the actual number of random test cases for every group of test cases. This allows us to evaluate the level of testing coverage during automated testing.

5. We discuss our future plans to use test coverage information for reliability assessment.