The 2nd International Symposium on Aircraft Airworthiness (ISAA 2011)

Coverage analysis of airborne software testing based on DO-178B standard

LU Yinghui\textsuperscript{a*}, ZHAO Yuerang\textsuperscript{b}, HOU Xiaojun\textsuperscript{a}, SUN Yan\textsuperscript{c}, BAO Yu\textsuperscript{c}

\textsuperscript{a}School of Aeronautics, Northwestern Polytechnical University, Xi'an, Shaanxi, P.R.China, \textsuperscript{b}Commercial Aircraft Corporation of China, Ltd Shanghai, P.R.China \textsuperscript{c}AVIC Wuxi 607 Institut, Wuxi, Jiangsu, P.R.China

Abstract

Software testing is one of the most important ways to protect and enhance civil aviation safety and reliability of software on airborne equipment. Among software testing, test coverage analysis is absolutely necessary. Therefore, based on DO-178B standard, this paper studies the method of software testing coverage analysis. With the example of TCAS software testing established in the test environment of hardware and the testing tools of software, we complete the software test coverage analysis.

© 2011 Published by Elsevier Ltd. Selection and/or peer-review under responsibility of Airworthiness Technologies Research Center NLAA, and Beijing Key Laboratory on Safety of Integrated Aircraft and Propulsion Systems, China Open access under CC BY-NC-ND license.

Keywords: Software testing, Airborne equipment, DO-178B, Test coverage analysis.

1. INTRODUCTION

Recent years, airborne electronic equipment is moving towards the direction of highly integrated. Therefore, the security and reliability of software is of great importance during the software development process. According to the requirements of DO-178B standard, in addition to the development process to verify all the sub processes, should also prove that the verification work is already done well enough, which put forward the concept of software test coverage.

* Corresponding author. Tel.:+86-029-15991279726.
E-mail address: pipilyh@126.com
2. Theoretical Principles of Coverage

2.1. Overview of DO-178B standard

DO-178B technical standard, a set of standards of the aviation industry, was proposed by RTCA (Radio Technical Commission for Aeronautics). It was released in 1992, known as “air systems and equipment certification in the software requirements”. Experts from manufacturers and the aviation industry quality certification authority were involved in writing the standard design. The standard gives the development process of embedded software on aviation systems and equipments to ensure the development of the software functions correctly, credible in security and can meet the air worthiness requirements.

2.2. White-box testing theory

The testing method of coverage is white-box testing, also known as structural testing of logical driven testing. Shown in Fig. 1, it is working under the premise that the internal work process of the product is known. With test, it can detect whether the inner movement of the product is in accordance with the provisions of specification and with the internal structure of the program testing procedures, and can detect whether procedures of each pathway required can be scheduled to work properly.

![Fig. 1. white-box testing](image)

The cover standards of White-box testing method are logical coverage, loop coverage and basic path testing. Logical coverage includes different coverage standards in the following: statement coverage, decision coverage, condition coverage, path coverage, decision/condition coverage, multiple condition coverage and modified condition/decision coverage.

2.3. Testing coverage analysis

Test coverage analysis is divided into two steps, including demand-based coverage analysis and structural coverage analysis. The first step is to analyze the link between test and software requirements, to ensure that the selected test cases would meet specific standard. The second step is to ensure that demand-based testing procedures to achieve the code structure.

In DO-178B technical standard, the onboard software is divided into five different levels according to the impact to the aircraft safety system caused by the failure of software. A-level, the highest security level, should have the most rigorous test. The software definitions are:

Level A: Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a catastrophic failure condition for the aircraft.

Level B: Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a hazardous/severe-major failure condition for the aircraft.
Level C: Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a major failure condition for the aircraft.

Level D: Software whose anomalous behavior would cause or contribute to a failure of system function resulting in a minor failure condition for the aircraft.

Level E: Software whose anomalous behavior would cause or contribute to a failure of system function with no effect on aircraft operational capability or pilot workload.

2.4. Requirement-based test coverage analysis

According to different software levels, the criteria we take to carry out structural coverage analysis are different.

This coverage analysis is intended to justify the effectiveness of the use of requirement-based testing methods on validating software requirements. This analysis may require some additional test cases. This requirement-based test coverage analysis mainly in the:

- Each test has a corresponding software requirements.
- The selected test cases should meet the normal and robust standard.

2.5. Structure-based test coverage analysis

Structural coverage analysis is intended to ensure that coding structure has not been verified by the requirement-based testing step. Requirement-based test cases may not fully verify coding structure, so that we take structural coverage analysis and add some additional steps to meet the structural coverage. DO-178B standard made a number of recommendations for guidance for this method of analysis:

1. Structural coverage analysis should ensure that coverage of the structure compatible with the software level.

2. Structural coverage analysis should be performed on the source code, unless the software is of Level A and the compiled object code cannot be directly traced back to source code. Then, there should be additional verification on the object code to determine the correctness of the generated code sequence. Compiled object code array are those cases that cannot be directly traced back to check the object code of source code.

Structural coverage analysis should ensure the data link and control link between the encoded components.

3. TCAS Software Test Coverage Analysis.

3.1. Requirement test coverage analysis

In the TCAS software requirements management, we use the DOORS software as a connecting link between test software and test procedure. The relationship between software requirements and test procedures is a direct link. During software requirements review process; we can clearly make sure whether each requirement has a test program to be connected with. While the testing program is being accreditation, we could see the requirements corresponding to the test program, and can also check whether the test case meet the demand and whether additional test is needed to achieve the coverage. Typically, software requirements and test procedures is not one to one, their correspondence is shown in Fig. 2:
Fig. 2. correspondence for software requirements and test procedures

Through the DOORS software’s function of traceability reporting, we can get a link report between requirements and test program, with which we could easily obtain the coverage of software requirement. With the assessment of the test program, and test cases added in time, the coverage requirement of software could be met.

3.2. Structural coverage analysis

Choosing test cases based on requirement, we can easily meet the demand of requirement coverage. As for meeting the demand of coding structural coverage, additional test cases should be taken into consideration. According to safety level of aviation software, a standard for structural coverage analysis was set by DO-178B. This standard has been the guiding principle for structural coverage of TCAS software.

(1) Test for decision coverage of TCAS software

In different modules of TCAS software, the security requirement varies, of which the majority are Level-C. For these modules, its coverage should meet the statement coverage standard. Yet the CAS logical module set in the center of TCAS software, is on Level-B, for these reasons the coverage should not only include statement coverage, but also decision coverage.

While in the coverage test of TCAS software, LDRA Testbed software is used. LDRA Testbed, a unique quality controlling tool developed by LDRA Company, has a powerful function in providing source code testing and analysis for validation and verification of application software. It helps to improve the necessary reliability, robustness, and possible zero defects of software. LDRA Testbed meet Level-A defined by DO-178B standard. It has been successfully applied in software testing in China’s “ShenZhou” Spacecraft project. Through instrumentation, it could be more easily to monitor the information flow of control to analyze the coverage, after running the source code. Fig. 3 shows the coverage testing process.
Based on copying the source code, LDRA Testbed inserts probes in key points of the program. The probe code inserted has three main tasks to complete:

A. Creating and opening execution history file.
B. Write execution history information into the file.
C. Close the file.

After compiling the source code, and load the compiled file onto the target computer, we can then complete program testing and data collection. LDRA Testbed takes dynamic analysis on the execute history file. While doing this, it could choose different coverage level according to the module being tested. Here we choose Level B which is fit for CAS module (Fig. 4). After finishing analysis, LDRA Testbed would show the coverage report.

Fig. 3. specific process of coverage test

Fig. 4. choose proper condition for the test
(2) Analysis for decision coverage of TCAS software

The statement coverage and decision coverage of CAS module would be clear, once the coverage report was get. Dealing with functions that have not reach 100% of coverage requirement, with analyzing the coverage report, we can handle this by adding testing cases or removing dead code. By these means, sometimes however, the coverage requirement would still not be met. For example, code respond that need to interrupt RAM, or tests need to be analyzed. In this case, we take artificial coverage analysis to make sure whether it meets the requirement.

Here are some coverage reports of a few functions in CAS module:

Example 1: the file to be tested is XXXClass.cpp. Fig. 5 (a) shows the coverage report, and Fig. 5 (b) shows the coverage conclusion.

Fig. 5. (a) coverage report; (b) coverage conclusion
The report of example 1 shows that all the statement coverage of the three functions are 100%, which suggest the result of coverage analysis meets decision coverage (Level B) required by DO-178B.

Example 2: the file to be tested is XXXDateClass.cpp
According to Fig. 6 (a), the coverage analysis report of XXXDataClass.cpp, both the statement and branch of function XXXDataClass::connect in target file have not been fully covered.

In Fig. 6 (b), the statement coverage conclusion of function XXXDataClass::connect, shows that 8 of 12 executable statements are executed, and 4 are left. In branch/decision coverage report in Fig. 6 (b), the function has three branches, of which only one is covered.

Fig. 6 (c) shows coverage report conclusion, from which we can get history information of structural coverage test of the three functions included in the file to be tested. The report could not only tell the number of uncovered lines and branches, but also finds the accurate position of them. With this information, we would be more confident in adding testing cases, besides, it could help us in detecting and removing dead code. In example 2, we can see that by adding test cases in the next test, the coverage of target function is 100%. Fig. 7 shows the details.
4. Conclusion

Coverage analysis is an indispensable part of aviation software testing. With requirement and structural coverage analysis, the coverage of TCAS software would meet the guidelines of DO-178B standard. By using DOORS software and LDRA Testbed, most work of analyzing, like quickly locating the uncovered code, or making the result of coverage analysis more accurate, could be completed automatically.

References