Airborne Software Data Coupling and Control Coupling Analysis

Hui Hua1*, Xinai Zhang2, Yi Zhao3 and Ran You4

1*Shanghai Aircraft Design and Research Institute, Shanghai, 201206, China
2Shanghai Aircraft Design and Research Institute, Shanghai, 201206, China
3Shanghai Aircraft Design and Research Institute, Shanghai, 201206, China
4Shanghai Aircraft Design and Research Institute, Shanghai, 201206, China
*Corresponding author’s e-mail: huahui@comac.cc

Abstract. As a part of software structural coverage, the data coupling and control coupling analysis help to evaluate the implementation of requirements and the code quality to some extent. This paper researches the data and control coupling definitions, analysis methods and the differences of requirements between DO-178B and DO-178C, thus providing reference for airborne software verification.

1. Introduction
The RTCA/DO-178B released in 1992 provides an acceptable design assurance method for airborne software. In July 2017, the Federal Aviation Administration (FAA) and the European Aviation Safety Agency (EASA) jointly issued an advisory circular[1] recognized RTCA/DO-178C as a software design assurance method, which is based on DO-178B and has made many improvements of software development objectives and activities, including the requirements for software data and control coupling analysis.

DO-178B/C requires several activities and objectives to form a complete verification of airborne software, including software documents and design data review, software testing, and test coverage analysis. Among these, software testing is used to demonstrate that the software satisfied its requirements and to demonstrate with a high degree of confidence that errors leading to unacceptable failure conditions have been removed, while test coverage analysis is a two-step process involving requirements-based coverage analysis and structural coverage analysis. The first step analyzes that all the software functions are correctly realized. The second step confirms that the requirements-based tests exercised the code structure to the applicable coverage criteria. Data coupling and control coupling is one of structural coverage analysis objectives, and represents to some extent the function implementation degree and code quality.

2. Data & Control Coupling

2.1. Data Coupling
Data coupling determines the dependence of a software component on data not exclusively under the control of that software components[2-7]. From a practical standpoint, data coupling is the transaction of data values from a point in the source code where the value is set to a point where the value is referenced.
in a software program.

The data coupling determines the degree of the data sharing and transaction between the software components, more precisely, the dependence of data usage between the software components. The weaker the data coupling is, the more independent the software component will be. However, DO-178B/C does not give a definition of the “software component”, which means it is the developer to define the software component function and scale according to the software architecture, and it is not required to analysis the data transmission inside the software component.

2.2. Control Coupling

Control coupling determines the manner or degree by which one software component influences the execution of another software component [2,7]. From a practical standpoint, control coupling is epitomized by the call-tree of the software. Exception and interrupt handing schemes may be influential in execution of another software component, but this is typically not a component of data coupling. Exception handling is generally not a planned call, but is related to a reaction to a data condition that is not properly handled by the base code, so exception handling is not considered as control coupling. Interrupts are a mechanism for reacting to external data and/or events, and is generally not considered as control coupling.

The control coupling determines the degree of the calling sequence between the software components, the weaker control coupling is, the more independent the software component will be. The same as data coupling, there is no definition of “software component”, and it is not required to analysis the call inside the software component.

2.3. Importance of software data & control coupling analysis

Coupling relationship is an important indicator to measure the reliability of software architecture. For safety critical and high-reliability airborne software, data coupling and control coupling analysis are particularly important [3].

If the interface characteristics of the software component are not restricted, the data and control coupling of one software component to other software components can cause problems in the operation of other software. In addition, when the data and control signals from other software components have typical operating characteristics, if they are not considered properly, they may cause operating problems to the coupled software components, thereby making them in abnormal mode [4].

3. Airborne Software Requirements for Data & Control Coupling

3.1. DO-178B/C Requirements for Data & Control Coupling

For DAL A/B/C software, DO-178B/C Annex A Table A-7 objective #8 requires [5]: Test coverage of software structure (data coupling and control coupling) is achieved.

DO-178B 6.4.4.2.c requires: the analysis should confirm the data coupling and control coupling between the code components.

DO-178C 6.4.4.d requires: Test coverage of software structure, both data and control coupling, is achieved. Which can be achieved by the following activities. 6.4.4.2.c requires: Analysis to confirm that the requirement-based testing has exercised the data and control coupling between code components. And 6.4.4.3 guides structural coverage analysis resolution, which may reveal code structure including interfaces that were not exercised during testing, namely shortcomings in requirements-based test cases and procedures, inadequacies in software requirements, extraneous code (including dead code), and deactivated code.

DO-178B and DO-178C has different activities requirements of data and control coupling, that DO-178B requires the structural coverage analysis if used to confirm data and control coupling between code components, while DO-178C requires the structural coverage analysis is used to confirm that requirements-based testing has exercised the data and control coupling between components. The updated verbiage points are to use of test coverage measurement technologies, similar to other structural coverage objectives, and the intent behind is to ensure that the applicants perform a sufficient amount
of hardware/software integration testing and/or software integration testing [6].

The structural coverage analysis of data and control coupling between code components should be achieved by assessing the results of the requirements-based tests. As the data and control coupling coverage requirements are dictated by the software requirements at both high-level and low-level, and by the software architecture, so the pass of the requirements-based tests can represent that the requirements are successfully implemented into the software, so are the planned data transaction and function calls.

4. Data & Control Coupling Analysis method

4.1. Static Review
As stated in section 3.1 of this article, the data and control coupling coverage requirements are dictated by the software requirements at both high-level and low-level, and by the software architecture, so the reviews and analysis of the software requirements in respect of data and control coupling are the first activities to perform, including review the correctness and the consistency of the requirements, and reasonable decomposition from high-level requirements to low-level requirements.

Meanwhile, data and control coupling coverage requirements are implemented into source code, so the requirements can be identified through the use of static analysis of the source code. Main consideration includes the consistency between the source code and the requirements and architecture, and compliance of the code to the coding standard which may raise requirements or suggestions regarding the coupling degree, the software complexity or the use of global variables etc.

Manual data and control coupling may be used.

For data coupling, data dictionaries may be extracted from the requirements to indicate value definitions and reference to variables, furthermore, data dictionaries can keep consistence of the data usage between software component and deferent level of requirements. However, keeping the data dictionaries up to date can be problematic, errors in the data dictionaries may cause inconsistency between software component and raise problems of the data usage.

Function parameter lists are likely to be verified at code review. Scoped data references are also likely to be verified at code review, because scoped data is commonly assigned a value in relatively near proximity to the reference. While global data references are less likely to be fully verified at code review, as global data reference do not necessarily occur in proximity to the data value assignment. This isolation is what makes using code review as a full verification of data coupling characteristics difficult.

For control coupling, call tree information is commonly carried in the function headers, and keeping the function headers up to date can be problematic.

To make a full verification, the analyzed data and control coupling should be reviewed for correctness with respect to the data coupling indicated in the requirements and architecture.

4.2. Tool-based analysis
The traditional data coupling coverage was done in group review, once tests were written, and often executed, the software verification team got together and went down the list of data coupling instances to determine if one or more tests were designed to exercise the coupling. The coverage was essentially predicted based on test design, and was never measured. At that time, tools to measure data coupling coverage were not common.

The tool-based data coupling coverage measurement is now available. At present, it is common to use tools for other structural coverage like statement coverage, condition coverage, decision coverage, and MCDC, but it is still not common for data and control coupling. It is expected that data and control coupling tools will become ubiquitous in the industry, much as statement-based coverage tools have become universally used.

When using the tool to conduct the coupling analysis, tool qualification may be needed, and it is useful to evaluate the tool operational requirements that would apply, that:

1. Correctly identify data and control coupling coverage requirements as represented by the code.
2. Correctly identify achieved data coupling coverage.
3. Correctly aggregate coverage metrics across multiple test executions.

4.3. Mechanism of Data & Control Coupling
An example can show in a simplified way the mechanism of data and control coupling. In Figure 1, there are three components: a subprogram “Main”, which calls two subprogram “Calculate the Cabin Pressure” and “Open the Thrust Recovery Valve”. The calculated cabin pressure is passed between the two subprograms using a global variable “Cabin Pressure”.

![Figure 1. Example of data and control coupling](image)

In this example, there exists controls coupling between “Main” and “Calculate Air Speed”, and control coupling between “Main” and “Open the Thrust Recovery Valve”. There also exists data coupling between “Calculate Air Speed” and “Open the Thrust Recovery Valve”. To satisfy the objective stated in DO-178C/DO-278A section 6.4.4.d, all of the following should be invoked during integration testing:
1. The subprogram call from “Main” to “Calculate Air Speed”.
2. The subprogram call from “Main” to “Open the Thrust Recovery Valve”.
3. The Passing of “Cabin Pressure” from “Calculate Air Speed” to “Open the Thrust Recovery Valve”.

5. Conclusions
Data and control coupling analysis is important to evaluate that the code is correctly implement the software requirements regarding the data value transactions and function calls in software. The analysis can be done by static analysis and by tool, taking the requirements and the requirements-based testing as the same basis. It is recommended that the data and control coupling is considered early at the development of the software requirements.

References
[1] FAA. Airborne Software Development Assurance Using EUROCAE ED-12() and RTCA DO-178(), AC 20-115D, [2017-7-21].
[2] RTCA. Software Considerations in Airborne Systems and Equipment Certification, DO-178C. Washington, D.C: RTCA Inc, 2011.
[3] Xia Xiaofeng, Pang Yu, Chen Kelong. (2019) Implementation of Data Coupling and Control Coupling Coverage Object for Radio Interface Unit Software. Computer Applications and Software., 36: 34-38,44.
[4] Sun Jinghua, Liu Jianfang, Tong Yuewei, Hua Hui. (1994-2020) Research of Software Data Coupling and Control Coupling based on RTCA/DO-178C. In: China Academic Journal Electronic Publishing House. Shanghai. pp.44-48. http://www.cnki.net.
[5] Kong Deqi, Zhong shan. (2018) How to Satisfy Data Coupling and Control Coupling Object of DO-178C. Aeronautical Computing Technique., 48:56-58.
[6] RTCA. Supporting Information for DO-178C and DO-278A, DO-248C. Washington DC: RTCA Inc,
2011.

[7] RTCA. Software Considerations in Airborne Systems and Equipment Certification, DO-178B. Washington, D.C: RTCA Inc, 1992.