The Design Assurance Method of Reused Airborne Software

J F Xu
Electronics Department, Shanghai Aircraft Design and Research Institute, Shanghai, 201210, China
xujunfeng1@comac.cc

Abstract. Reusing software data from different sources and different reuse methods will affect the full compliance of civil aircraft airborne software. To achieve cost and risk reduction in reused software project, firstly classification of reused airborne software is confirmed and assessment elements are recognized; secondly the assessment framework of software reuse is established and assessment activities and sequences are specified; lastly the reused software design assurance requirements are given from the perspective of certification. This study can provide support for stakeholders of reused software certification to recognize and assess reused software and generate certification strategy in the early stage of project.

1. Introduction
Facing the increasing cost and difficulty of large and complex software development, especially in the safety critical domain, software data reuse is an effective approach to improve software development efficiency and software quality.

In the aviation domain, most software projects are derived from existing systems. In actual projects, due to different software sources and development bases, there are different types and degrees of reuse of civil aircraft airborne software. When not all software data can be completely reused, suppliers tend to demonstrate compliance as new developed software, and inexperienced suppliers may therefore reduce the adequate assessment of reused software data compliance and face certification risks and change costs later in the project. Therefore, a full assessment of software reuse early in the project can help stakeholders such as software developers, integrator, and applicant to assess project risks and obtain airworthiness approval.

2. Background

2.1. Software reuse study status
Developers in various fields have conducted a lot of research on the application of software reuse technology: Li X.Y. proposed a software reuse framework for electronic information systems[1]; Zheng Y. et al, Ren G.S. et al, Ma F.Y. et al, Hao W.W. et al, Li M.Y. et al studied the software reuse technology on flight control system software, monitoring software, flight simulator generation software, ray-board software and on-board software[2][3][4][5][6]; Zhang X.G. et al. applied software reuse techniques to software testing[7]. In the field of civil aircraft, Luo Q.Q. proposed a specification standard and implementation guidelines covering the DO-178C standard defined software life cycle to reduce the difficulty of DO-178C implementation[8]; Li K. proposed a reusable airborne head-up display system architecture based on software reuse mechanism and ARINC661 architecture[9].
For the airworthiness certification of reused software in the civil aircraft domain, FAA Advisory Circular AC 20-148 provides guidance on the airworthiness approval of Reusable Software Components (RSC)[10]; the CAST-22 report applies the concept of RSC to tool qualification[11]; Zhao C.X. et al. address the airworthiness certification requirements for reusable components of integrated modular avionics systems[12].

2.2. Definition and classification of reused airborne software

Airborne software reuse is the process of implementing or updating an airborne software product for civil aircraft systems using existing software data. When reusing existing software data in a new civil aircraft project, the characteristics of the existing software data and the reuse method will affect the compliance demonstration process of the new software project.

The reused software data can be software components, software requirements, software design, source code, or other software life-cycle information (including plans, standards, test data, and tool qualification data). Sources of reused airborne software include Commercial Off the Shelf (COTS) software, software from previous projects, software developed per DO-178() or non-DO-178() standards. Software can be reused into existing systems, similar systems, or different systems in a variety of forms: changed as required or completely unchanged.

According to the sources of reused software and software reuse forms, summarize the classification of reused airborne software in Table 1.

| No. | per DO-178()? | Is COTS? | Examples |
|-----|---------------|----------|----------|
| 1   | Y             | Y        | Reuse COTS RTOS with DO-178() package |
| 2   | Y             | N        | Reuse software developed per DO-178() in similar/new system |
| 3   | N             | Y        | Reuse libraries provided by compiler |
| 4   | N             | N        | Reuse software developed per military or automotive standards |

3. Method

3.1. Reused software assessment framework

In order to reuse an existing software data into a civil aircraft product and demonstrate complete compliance to the Authority, the reused software should be assessed and the associated compliance strategy should be identified and documented in the software planning documents. The reused software assessment framework is shown in Figure 1.
The software reuse evaluation steps are described in Table 2.

| Step | Description |
|------|-------------|
| 1    | Assess the approval status of the software in a qualified civil aircraft, engine, or propulsion. |
| 2    | Assess the software compliance with the PDS definition to reuse the existing approval letter. |
| 3    | Assess the DO-178() standards followed by the software in a previous project. |
| 4    | Assess the software level compliance with the current project requirements. |
| 5    | Assess and identify the changes occur in the software or development environment. |
| 6    | Identify and assess the changes affected data and activities through change impact analysis and develop a re-verification plan. |
| 7    | Assess the compliance with project installation requirements. |
| 8    | For software reused in a new installation, perform a comprehensive assessment of the expected reuse environment to identify affected data. |
| 9    | Analyze the reused software and its supporting information to identify gaps based on DO-178C[13] objectives. |
| 10   | Perform baseline updating to close the gap between the reused software and the DO-178C objectives and perform additional verification activities as necessary. |
| 11   | Archive software reuse strategy, applicable principles and guidelines in the planning document, and get approval from the Authority. |
3.2. Reused software design assurance requirements

AC 20-115D[14] takes DO-178C as an acceptable software development assurance method, when applying DO-178C as a compliance method for current projects, the reused software should meet the following requirements.

3.2.1. For software developed per DO-178(). The software should comply with the requirements of current project:

- If the software has been approved in previous civil aircraft, then the target system, installation method and the software level should meet the requirements of current project as compared to the previous approval.
- If the software has not been approved, the software should demonstrate compliance with the current project requirements. If any changes on installation, software development environment and software functionality exist, or software level does not meet the requirements, applicant should perform a gap analysis and gap compensation in accordance with DO-178C Section 12.1.2, 12.1.3, 12.1.4 and perform additional verification activities as necessary to meet the additional DO-178C objectives.
- If the software uses new technologies, additional compliance should be demonstrated, such as DO-331[15] for model-based development techniques, DO-332[16] for object-oriented techniques, DO-333[17] for formal methods, DO-178C parameter data items requirements for configuration data, AC 20-115D section 10 for tool qualification.

3.2.2. For software not developed per DO-178(). A gap analysis should be performed based on the DO-178() objectives to identify unmet objectives:

- For gaps identified by the gap analysis, use the alternative methods and their combinations given in Table 3 to obtain equivalent confidence levels.
- For COTS software, the DO-178C equivalent level of confidence should be obtained in accordance with the guidance in section 12.4 of DO-278A[18].
- If all information of the software is inaccessible, the approval level should be limited to D or E.
- If the source code of the software is not available, the maximum approval level should be limited to level D.
- If the software needs to show compliance up to level A and B, the identified gap will be large and the reuse strategy should be reconsidered.

Table 3. Alternative methods.

| Method              | Guide                                                                 |
|---------------------|-----------------------------------------------------------------------|
| Service History     | Provide evidence in accordance with DO-178C Section 12.3.4 to ensure that: service history has relevance, service history is adequate, service history has a problem reporting process. |
| Process Recognition | Establish the DO-178C compliance matrix by mapping the existing process to DO-178C process. |
| Reverse Engineering | Develop high-level abstractions based on low-level abstract data by well-defined plans, experienced team, closely communication with original developers or experts, and establishing clear and reasonable traceability relationships. |
| Function Restrictions| Typical function restriction methods include: run-time checks, deactivated code, built-in restrictions, etc. |
| Architecture Mitigation | Typical architectural mitigation methods include: |
|                      | - Software partitioning.                                             |
|                      | - Safety monitoring.                                                 |
| Additional Tests     | Typical tests include:                                               |
|                      | - Exhaustive input test.                                             |
|                      | - Comprehensive robustness testing.                                  |
|                      | - System level testing.                                              |
|                      | - Long-term immersion test.                                          |
|                      | - Multiple independent users run the system.                         |
4. Conclusion
This paper first analyzes the reuse sources and reuse methods of civil aircraft airborne software, and summarizes the classification of airborne software reuse. On this basis, the assessment elements for reusing existing software data into civil aircraft products are identified, and the assessment framework for airborne software reuse is given. Further, the certification requirements of reused airborne software are proposed from the perspective of practical review. The research can help to assess the compliance of reused software and develop a certification strategy at the early stage to reduce the potential certification risk and change cost.

Acknowledgments
The author would like to thank the Electronics Department of Shanghai Aircraft Design and Research Institute for sponsorship of this research, and Master Shan J. M. of Shanghai Ocean University for data collection of this work.

References
[1] Li X.Y. (2016) Technology Analysis and Application of Software Reuse to Electronic Information System. In: 2nd Workshop on Advanced Research and Technology in Industry Applications (WARTIA 2016). Dalian. pp. 1531-1536.
[2] Zheng Y., Wei P., Qiao T. (2012) Software reuse techniques for flight control systems. Aviation Computing Technology, 42(4): 104-109.
[3] Ren G.S., Yang J.D., Xu W.X. (2003) Design of monitoring software based on software reuse technology. Journal of Chongqing University, 26(6):18-21.
[4] Ma Y.F., Zhou B.S. (1992) A software generation system for flight simulators based on software reuse techniques. Journal of Beijing University of Aeronautics and Astronautics, 3(013): 91-98.
[5] Hao W.W., Wu X.G., Niu Y. (2001) Ray-board software reuse technology. Firepower and Command Control, 36(11):197-205.
[6] Li M.Y., Peng P. (2011) Research on on-board software reuse technology. Shanghai Space, 28(5): 63-67.
[7] Zhang X.G., Liu R.S. (2009) Research on the application of software reuse in aviation embedded software testing. Aviation Computing Technology, 39(3): 87-91.
[8] Luo Q.Q. (2017) Design and implementation of software reuse platform for flight control system based on model library. University of Electronic Science and Technology, Chengdu.
[9] Li K. (2017) Design and implementation of an airborne head-up display system based on software reuse mechanism. University of Electronic Science and Technology, Chengdu.
[10] US Department of Transportation Federal Aviation Administration. (2004) AC 20-148: REUSABLE SOFTWARE COMPONENTS. FAA, USA.
[11] Certification Authorities Software Team. (2005) Certification Authorities Software Team (CAST) position paper 22: reuse of software tool qualification data across company boundaries (applying the reusable software component concept to tools). FAA, USA.
[12] Zhao C.X., Yan F., Zhang F., et al. (2016) Analysis of airworthiness certification requirements for reusable software components of integrated modular avionics systems. Electro-Optics and Control, 23(9): 63-67.
[13] RTCA SC-205. (2011) DO-178C Software Considerations in Airborne Systems and Equipment Certification. RTCA, Washington.
[14] US Department of Transportation Federal Aviation Administration. (2017) AC 20-115D: Airborne Software Development Assurance Using EUROCAE ED-12() and RTCA DO-178(). FAA, USA.
[15] RTCA SC-205. (2011) DO-331 Model-Based Development and Verification Supplement to DO-178C and DO-278A. RTCA, Washington.
[16] RTCA SC-205. (2011) DO-332 Object-Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A. RTCA, Washington.

[17] RTCA SC-205. (2011) DO-333 Formal Methods Supplement to DO-178C and DO-278A. RTCA, Washington.

[18] RTCA SC-205. (2011) DO-278A Software Integrity Assurance Consideration for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems. RTCA, Washington.