World Scientific Research Journal Volume 8 Issue 12, 2022
ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

Software Development of Airworthiness Oriented Piston Aeroengine Control System

Honglei Lu, Lihua Zhu, Xiaoming Shan, Chun Zhang, Zhicheng Qi

AECC Aeroengine Control System Institute, Wuxi, China

Abstract

As an important part of the general aviation field, the research and development process of piston aeroengine control software needs to meet not only the normal functional requirements, but also the airworthiness requirements. Firstly, this paper expounds the background of piston aeroengine development. And explains the basic requirements of airworthiness. On this basis, referring to DO-178C airworthiness standard, a safety critical software development method meeting the objectives of airworthiness development process is proposed. According to the process activities required in the airworthiness standard, carry out the processes related to software high level requirements (HLR), architecture design, low level requirements (LLR), source code and software integration in the process of piston aeroengine software development.

Keywords

Safety critical software; Airworthiness; Piston engine.

1. INTRODUCTION

1.1. Status of The General Aviation Industry

The general aviation industry has a wide range of applications, including all civil aviation activities except scheduled passenger and cargo flights [1]. There is still a big gap between China's general aviation industry and the international level. However, after the Civil Aviation Administration of China (CAAC) issued the "Administrative Measures for General Airport Classification" in 2017, the construction of general airports is gradually taking off, and the general aviation industry has gradually attracted market attention.

With the large application and scope of general aviation aircraft, there are a variety of corresponding power devices. From the perspective of engine mechanical system, the general aviation aircraft can cover turbofan, turboprop, turbojet, turboshaft, piston engine, among which piston engine accounts for 45% [1]. Therefore, it is of great significance for the research and development of piston aeroengine. From the perspective of piston engine, its production and manufacturing technology has been relatively mature due to its extensive application in the automotive industry. However, its additional considerations for air conditions and changes in fuel quality still require multiple technologies (lightweight, reliability, supercharged matching, spray combustion, etc.) [2]; on the other hand, the research and development of aeroengine is highly related to safety. Whether it has security qualification requires the certification of the local Civil Aviation Administration.

1.2. Aeroengine Software Airworthiness

For the research and development of safety-critical systems, in addition to focusing on the requirements and functions, it is also necessary to consider the problems arising in the research and development process, to ensure that the process is controllable. Especially for its software

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

R&D, problems such as inconsistency of multi-level requirements, ambiguity of requirements, unexpected errors introduced in the development process, etc., need to be controlled through certain technical and management methods. For software development of the civil aviation industry, the DO-178C series standards [3] are generally required and accepted by Federal Aviation Administration (FAA) as a software compliance method [4]. In addition, for some new R&D technologies, it has added additional considerations, such as DO-330 standard [5] for tool qualification, DO-331 standard [6] for model-based development and verification. In China, the construction of airworthiness system is still in the development stage, and there is still a big gap compared with the international level. As airworthiness is a system construction rather than a single point of problem improvement, it needs to consider the safety aspects from a system perspective, such as the compliance of plans, requirements and standards, compatibility with target computers, traceability of evidence, accuracy of algorithms, etc.

The development of piston aero-engine control software should not only consider the development of the functional requirements of the engine itself, but also consider its safety and airworthiness goals in the development process.

2. METHODOLOGY (MODEL BASED DESIGN)

Compared with the traditional software development methods [7], the model-based development method has the characteristics of rapid requirements capture, automatic code generation, and high R&D efficiency. It has gradually become the preferred method in embedded application software development and its compliance with airworthiness objectives can also be satisfied [8]. As shown in Figure-1, the application layer software in this article uses a model-based method for development.

For software development phase, based on model development and verification, the relationship between requirements and models can be established through the lifecycle data management platform. Furthermore, models can directly generate the corresponding source code through the code generator. In addition, the code generator needs to pass DO-330 standard for tool qualification, which can ensure the consistency of source code and model.

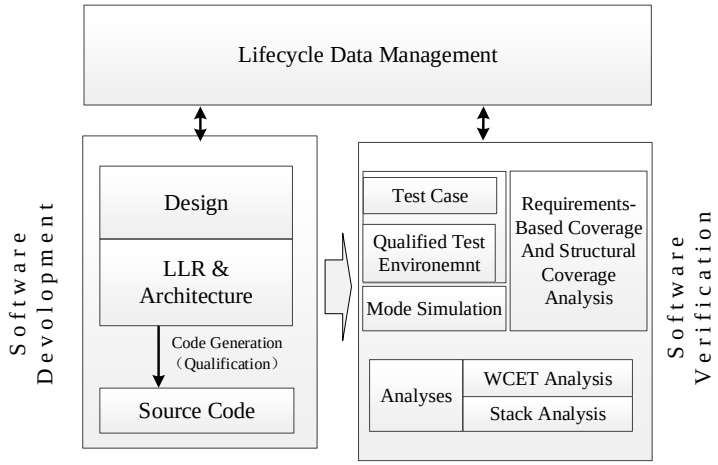

Figure 1. Application layer software development based on MBD

For software verification stage, both model coverage and code coverage can be collected by means of test and model simulation to validate requirements coverage and redundancy

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

structure validation; Worst-Case Execution Time (WCET) and stack analysis can also be used to verify code performance requirements and compatibility with the target computer [9].

3. AIRWORTHINESS OBJECTIVES

Before software development, the compliance of system functions is generally managed through the ARP4754A standard [10] to manage the system requirements process, of which the safety requirements are guaranteed by ARP4761 standard [11]. As shown in Figure-2, the software shall meet the system requirements and decompose the system requirements from the perspective of software. Considering the particularity of airborne safety critical software development and reducing the major failure due to non-expected errors, a series of activities are specified for the development of airworthiness software. It involves software planning process, development process (including requirements process, design process, coding process, integration process), verification of outputs of requirements process, verification of outputs of design process, verification of outputs of coding process, verification of outputs of integration process, verification of verification process results, configuration management process , quality assurance process, etc.

The objectives of the planning process, it is necessary to define various plans and standards of the software life cycle; The software development process needs to define various activities in the process from requirements to source code and object code; The verification of the requirements process is mainly to verify the correctness and verifiability of the output of the requirements process. Verification of usability and other aspects; similarly, verification of the integrity and correctness of design, coding, and integration process; The verification of verification process results is mainly to ensure the correctness of verification and the coverage of requirements and structures, so as to prove that all requirements have been realized, and there is no extra code and function redundancy; the configuration management process is mainly to ensure the orderliness of project development, change management, software release and problem handling mechanism, etc.; the quality assurance process is mainly to ensure that the implementation of the project meets the predefined activities. So, the objective of the entire DO-178C is mainly to ensure that the functions required by the system are correctly implemented in the software development process, and no additional errors are introduced.

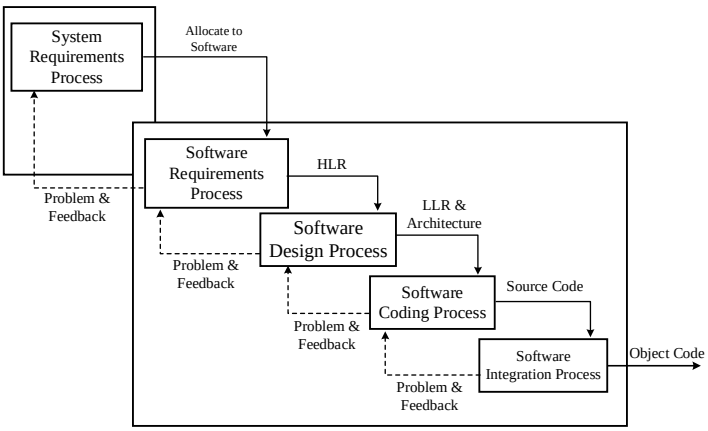

Figure 2. Information flow in software development process

In addition, for the verification process, it can be divided into three ways: review, analysis, and testing. The review can be carried out in the form of peer expert inspection, in the form of a checklist, etc.; the analysis can be carried out in a deterministic and repeatable way with the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

help of special tools on the function, performance and other aspects of software components; testing can be carried out by establishing test requirements, test cases, test procedures, and confirming the test results after implementation.

Based on the software development of piston aeroengine as an example, this paper mainly introduces the airworthiness goals that need to be met in the process of software development. As shown in Table-1, there are 7 objectives that need to be met during the development process, which involves different development stages.

Table 1. Software Development Process Airworthiness Objectives (DO-178C [3])

Index	Objective Description	Software Process
A2.1	High-level requirements are developed	Requirements Porcess
A2.2	Derived high-level requirements are defined and provided to the system processes, including the system safety assessment process.	Requirements Porcess
A2.3	Software architecture is developed.	Design Process
A2.4	Low-level requiremnts are developed.	Design Process
A2.5	Derived low-level requirements are defined and provided to the system processes, including the system safety assessment process.	Design Process
A2.6	Source Code is developed.	Coding Process
A2.7	Executable Object Code and Parameter Data Item Files, if any, are produced and loaded in the target computer.	Integration Process

4. DEVELOPMENT PROCESS

4.1. Requirement Process

The requirements process mainly involves the two objectives of A2.1 and A2.2 in Table-1. The basic activities include: on the one hand, the system requirements (including functional requirements, interface requirements, safety requirements, etc.) allocated to the software level need to be decomposed to ensure that each system requirement allocated to the software has corresponding high-level software requirements; on the other hand, the requirements themselves should meet the defined requirement standards, and should not describe the details of design and verification; finally, for the requirements derived from high-level requirements, incorrect or ambiguous contents in system requirements, should be feedback to the system process to ensure the normal development of functions and the evaluation of safety.

Requirements can be edited in the requirement platform, mainly including the maintenance and management of requirements at all levels. For high-level software requirements, it is mainly about the decomposition of system requirements and the description of what to do from the software perspective. High-level requirements for piston aeroengine control software, mainly including signal processing, engine state management, fuel injection management, fault handling, etc. Furthermore, the control of a piston aeroengine is different from that of vehicle piston engine. In addition to normal functions, it is necessary to consider the requirements of system fault tolerance.

4.2. Design Process

The design process mainly involves A2.3, A2.4 and A2.5 in Table 1. Its basic activities include: on the one hand, the architecture and low-level requirements should meet high-level requirements and conform to the defined design standards; On the other hand, it is also

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

necessary to consider the handing of failure modes, the consistency of data flow and control flow of each software component, and its impact on safety; finally, the derived requirements involved in the design process and problems found in the design process should be feedback to the upper-level process for clarification or correction.

After completing the high-level requirements, the software architecture can be built according to the high-level requirements during the design process. The piston engine control software is processed as follows: First, the hardware driver module obtains real-time data from the hardware and communication. And transfers the original data to the application layer software. Secondly, the application layer software processes the original data (including signal calibration, signal filtering, fault diagnosis, etc. Further, after the signal is pretreated, the application layer software transfers valid data to the engine status management module, and manages the demand torque according to the engine status and external conditions.

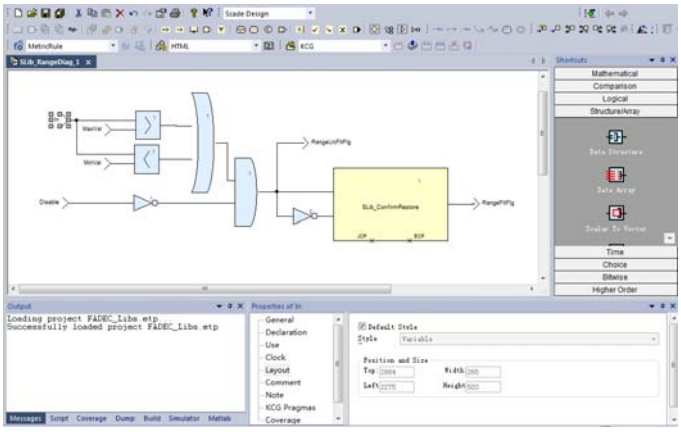

Figure 3. Application layer software development

As shown in Figure-3 after the above software architecture design is completed, the software can be further detailed designed to meet the functional requirements required in high-level requirements.

4.3. Coding and Integration Process

The software coding process focuses on the A2.6 objective in Table-1. Since the application layer is model-based, the source code can be automatically generated in the model-based design environment, and the code generator that has been qualified, ensures that the source code implements the low-level requirements and conforms to the software architecture and coding standards. For other handwritten code, the low-level software requirements can be established through the requirements management platform. Furthermore, source code can be developed according to the low-level requirements, and corresponding verification activities can be carried out.

The integration process focuses on the A2.7 objective in Table-1. Its main activities include: on the one hand, compile and link the source code to generate executable object code; On the other hand, software integration is done in a certain form to ensure functional integrity and interface correctness. Software and hardware integration is performed in the target computer environment to verify compatibility and performance impact with the target computer. On this basis, inadequate and incorrect inputs need to be feedback to the upper process for clarification and correction; finally, changes to requirements, architecture, or source code during development should be strictly restricted and managed in formal airworthiness products to ensure traceability, rationality, and compliance with objectives.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

Model checkers can be used for model integration. It checks for errors in software models and architecture through a set of built-in rules, including incorrect or redundant variable definitions, type consistency, consistency of timing, consistency of control flow and consistency of data initialization. The result after passing the checker is shown in Figure-4.

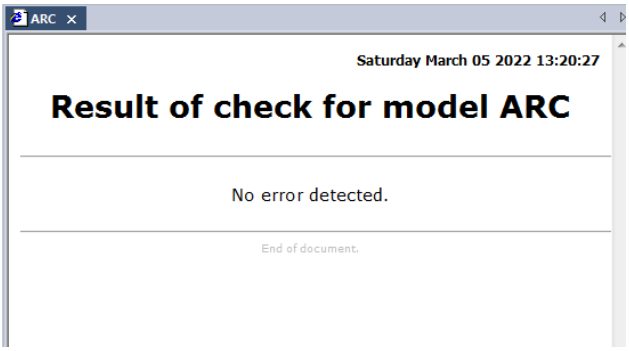

Figure 4. Model checker-based rule checking

Model generation code is integrated with handwritten code through the compiler and associated configuration files. Address space of code and warning during compilation are also analyzed. The compiled executable object code is then loaded into the target computer for testing to ensure its correctness.

4.4. Traceability of the development process

Although the data traceability objective is not specified during the development process, the corresponding activities have clear required in the DO-178C standard. It is convenient to assess whether the requirements have been fully implemented, and the visibility of derived requirements and redundant functions. On the other hand, it is also easy to track changes in requirements by establishing traceability between requirements and models at each level.

Traceability between configuration items can be established and visualized through the life cycle management platform. For the traceability between requirements, the traceability between system requirements and software high-level requirements, between software high-level requirements and some software low-level requirements can be carried out. On the other hand, the traceability between models and high-level requirements can also be established. The traceability between handwritten code and low-level software requirements can be carried out through code comments.

5. CONCLUSION

This paper expounds the software development based on piston aeroengine from the airworthiness point of view. Based on the DO-178C standard, carry out software high-level requirements, architecture, models, low-level requirements, source code and integration process to achieve the purpose of meeting the airworthiness objectives (mainly 7 development process objectives).

REFERENCES

[1] Dong Yanfei, Huang Ming and Li Ruiqi, “Review of general aviation engine development”, Journal of Xi'an Institute of Aeronautics, 2017, Vol. 35 (5), p8-13.

[2] Pan Zhongjian, He Qinghua and Yang Jing, “Development status of piston aviation heavy oil engine s”, Science and Technology Guide, 2013, Vol. 31 (34), p65-68.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202212_8(12).0024

[3] DO-178C, "Software Considerations in Airborne Systems and Equipment Certification", Washington DC: RTCA Inc, December 2011.

[4] DO-330, "Software Tool Qualification Considerations", Washington, DC: RTCA, Inc., December 2011.

[5] DO-20-115D, "Airborne Software Development Assurance Using EUROCAE ED-12() and RTCA DO-178()", FAA, USA, 2017.

[6] DO-331, "Model-Based Development and Verification Supplement to DO-178C and DO-278A", Washington, DC: RTCA, Inc., December 2011.

[7] Yang Lisha, Wang Hui. "Model-based design of embedded software", Application Research of Computers, 2004, 21(12):76-78.

[8] Jean-Louis Camus, Bernard Dion. "Efficient development of airborne software with SCADE suite", Esterel Technologies, 2011.

[9] Leanna Rierson. "Developing safety-critical software: a practical guide for aviation software and DO-178C compliance". CRC Press, 2017.

[10] ARP4754A, "Guidelines for Development of Civil Aircraft and Systems", Warrendale, PA: SAE Aerospace, December 2010.

[11] ARP4761, "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment", Warrendale, PA: SAE Aerospace, December 1996.