ICAS 2020 SHANGHAI
International Council of the Aeronautical Sciences
32nd Congress of the International Council of the Aeronautical Sciences
September 6-10, 2021
Pudong Shangri-La, Shanghai, China

MODEL-BASED VIRTUAL INTEGRATION TEST METHOD FOR AVIONICS SYSTEM

Shaofan Zhu, Peipei Xing*

COMAC Beijing Aircraft Technology Research Institute, Beijng 102211, China

Abstract

In this work, a model-based virtual integration test method is proposed, which attempts to solve the problems in the IV&V(Integration, Verification&Validation) process of complex avionics systems, such as exponential increase in complexity, difficulty in collaborative design, complex dynamic cross-linking of integrated testing, and low manual test efficiency. The proposed method is decomposed into several parts that are detailed to go from operational concept to the process, method and environment of virtual integration test of avionics domain. To achieve this work, virtual test elements are firstly defined, then identify the three major abilities required by the virtual integration test. Define the whole process and method from system architecture design to integration test. Finally, develop the basic operational environment for virtual integration test.

Keywords: model-based; system engineering; virtual component; integration tesst; avionics system

1. General Introduction

With the rapid increase in the complexity of modern aircraft avionics systems, in the face of hundreds of thousands of signal data interactions and thousands of data buses, the coding workload is extremely huge. When requirements change rapidly and system failures continue to occur, the traditional software-in-the-loop approach is even more stretched. Hardware-in-the-loop testing is an integrated testing method based on software-in-the-loop on a hardware test bench. In the process of system integration and verification, it is also necessary to solve the problems of large amount of electronic interface communication protocol testing, low manual test efficiency, and complex dynamic cross-linking tests. At present, all model-in-the-loop tests are mostly closed-loop tests for a single model such as Simulink, which has the limitation of a single model type and is difficult to integrate.

Avionics system is a relatively special field, which is different from the majors of structure and materials. It focuses on testing the functional and logical characteristics of avionics systems. The existing tools and software on the market focus on the physical characteristics of system testing, while the tools with special functional and logical characteristics of the avionics system are very rare. Therefore, conventional tools are difficult to realize the early verification and testing of model-based avionics systems. A new development environment is needed to complete the virtual simulation and comprehensive verification of avionics system, so as to improve the automation degree of integration testing and shorten the development cycle.

2. Virtual Test Overview

2.1 Definition of Virtual Test

In the traditional V-shaped research and development model, the integration stage needs to be executed after the detailed system design and implementation, so the system integrator can only start the integration when the software and hardware are completed[1]. However, in virtual integration and testing, the system or device under test is abstractly expressed as a Virtual Component (VC)[2]. At the same time, in order to enable testers to perform multi-virtual component integration testing, this paper introduces the concept of Virtual Test Bench (VTB). The virtual test is a customized modeling test process based on the avionics system development process system. It specifies the virtual elements of the avionics system and the interoperability framework of components, and confirms the correctness of the system functions and interfaces through dynamic simulation of the model, so as to realize the rapid evaluation of the model and the evaluation of the system options.

2.2 Definition of Virtual Test Related Elements

- **Virtual Component (VC):** A generalized integrated component of avionics virtual test, which is the building block of a Virtual Bench: a simulation model that represents an equipment.
- **Virtual Test Bench (VTB):** The most important part of the avionics virtual test, which realizes the integration test from multi-level architecture design input to virtual component. In the form of multiple virtual component integration packages, it covers all the information required for the test (interface, databus, link relations). It also satisfies the seamless connection of virtual components to real software/hardware test benches in virtual testing or semi-physical testing.
- **Virtual Component Requirement (VCR):** It defines the test scope of the virtual component to be tested in the virtual test bench, including the interface to be tested and the external environment.
- **Virtual Component Specification (VCS):** As an extension of the VCR, this interface definition not only inherits the basic information such as the relevant test range, interface and external environment in the VCR, but also defines the detailed definition of each interface to be tested in the virtual component. The interfaces defined at different levels are different. At the logical architecture layer, the interface definition only describes the corresponding basic information flow and simple bus information. At the physical architecture layer, the interface definition will reach the physical level ICD.
- **Simulation Execution Engine (SEE):** A runtime software that carries and executes virtual components to realize real-time simulation testing of virtual components in the virtual test bench.

2.3 Analysis of the Operational Concept of Virtual Test

As shown in Figure 1, the virtual integration test process corresponds to the logical architecture and physical architecture layers of the system development cycle. Based on the constructed virtual integration test process, method and environment, the requirements validation for logical design, physical design, subsystem design, and system integration of the avionics system are completed. It can provide design input for the system implementation stage and verification support for the system integration phase.

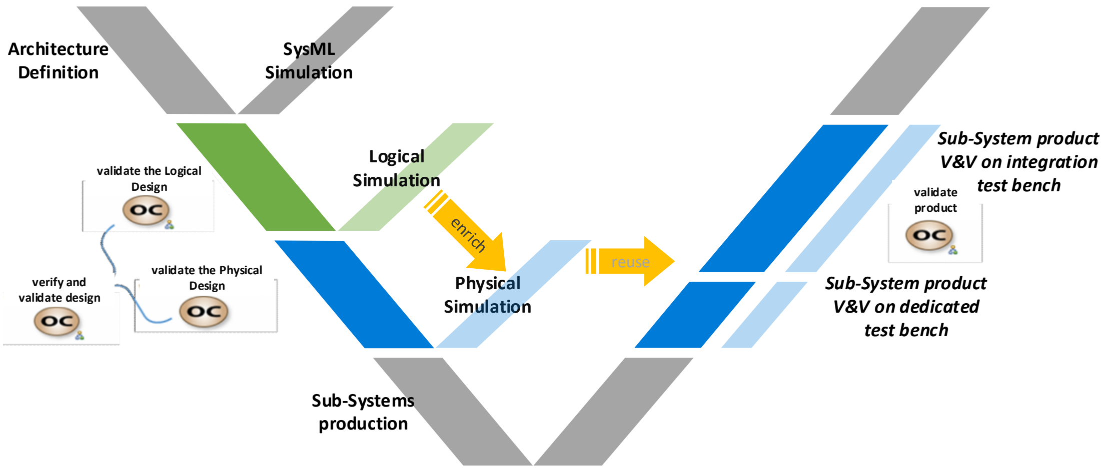

Figure 1 – Virtual integration test capability in the development cycle.

By capturing the requirements of model-driven avionics system early test technology, we can get the following three capabilities for virtual integration testing of avionics system:

- **Design-oriented test verification capability**

The virtual integration test of avionics system should be able to define the minimum test scope and related test procedures according to the scale of the system under test and the fidelity of virtual components required by it. In general, the following capabilities are required for design-oriented simulation test verification:

- Logic design validation: it can verify the subsystem functions and function exchanges defined in the logic design, and define the information flow and the mapping relationship between the function flow and the information flow at the same time, and generate the corresponding functional interface definition file (FICD);
- Physical design validation: it can verify the physical elements, physical interfaces (A429, CAN, etc.) and related physical connections defined in the physical design;
- Subsystem validation: it can verify the sub-system environment, so as to prepare for the subsequent construction of a virtual test bench;
- System integration validation: based on the constructed virtual components and virtual test bench, it can realize the simulation validation of system integration.

- **Product-oriented validation capability**
  On the hybrid test bench, reusable virtual components and real components can be used for integration testing to achieve validation of real products.
- **Provide a virtual integrated environment**
  Building a virtual integrated testing environment, so that all subcontracting suppliers can participate in the realization of early model-based verification and validation activities.

3. Virtual Integration Test Process Design

3.1 Virtual Integration Test Phase Division

On the right side of the V-shaped model of the avionics system, a virtual integrated testing system is introduced to open up the relationship between top-down design and virtual testing, so that the overall development process data is homogenized and expressed as a model. The virtual integration test process of the avionics system is shown in Figure 2, which is mainly divided into three parts:

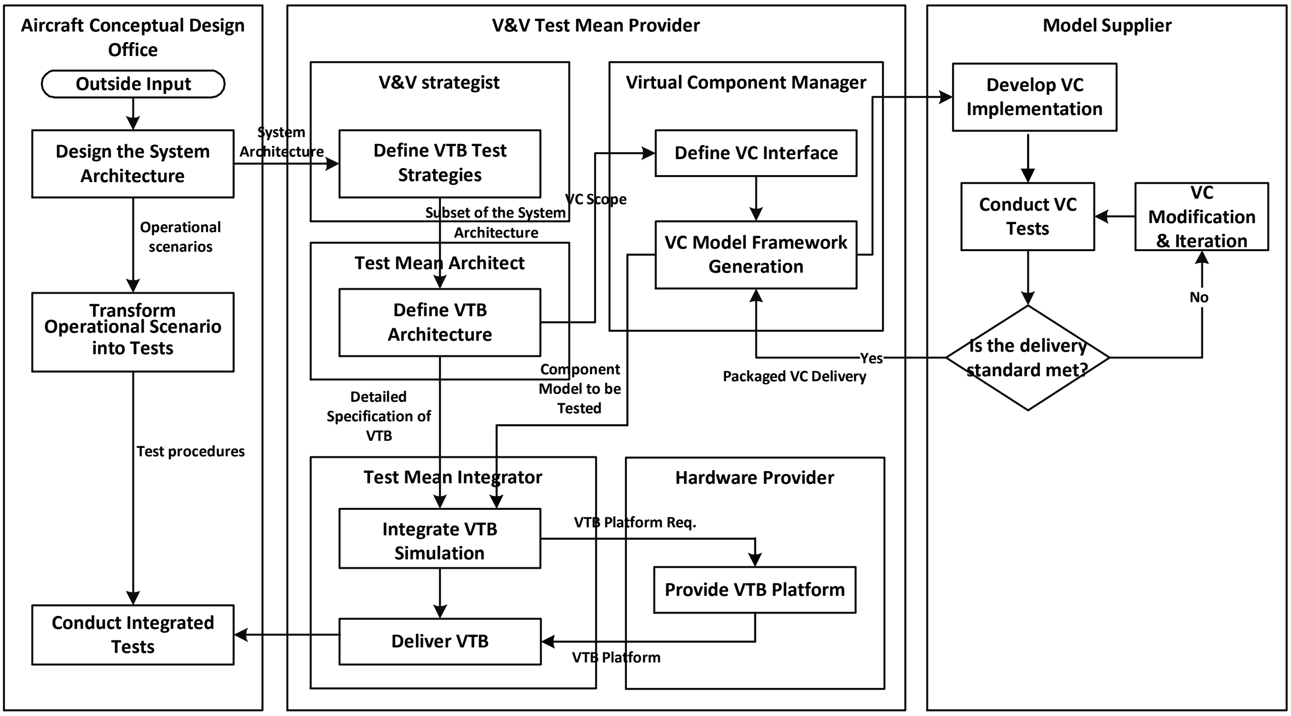

Figure 2 – The process of virtual integration test.

- System architecture input and virtual test execution guided by the conceptional design office;

- The model-driven avionics system virtual integrated test system builds the simulation architecture of the tested object system, automatically generates the virtual test framework, and configures multi-model integrated test resources;
- Internal realization of virtual components by model supplier**.**

The above three parts have strong coupling characteristics and can be traced back in both directions. The verification environment depends on the iterative input of the conceptional design office and model suppliers. The virtual integration test process of the model-driven avionics system is divided into seven stages. The following describes each stage one by one.

3.2 System Architecture Design

The design of the system architecture is the responsibility of the design team. They mainly design the logical architecture of the system, which is divided into the following five stages:

- Capture logical architecture: use mainstream modeling tools such as Cameo Systems Modeller, Capella, etc. to build logical architecture for the system under test[3];
- Confirm the continuity of the system architecture: identify logical architecture interface information and connection relationships;
- Check for errors or missing architecture information: check the logical architecture structure and non-business-level errors of interface information, and find out whether there is missing architecture information;
- Define the baseline of the logical architecture: define the baseline of the logical architecture to facilitate the rapid iteration of subsequent architecture models;
- Visualization of ICD module: convert the logic architecture to a visual ICD module, which facilitates the establishment of subsequent verification strategies and simulation architecture.

3.3 V&V Strategy Establishment

The construction of the V&V strategy of the virtual test bench is the responsibility of the verification and validation team, which is mainly to define and refine the requirements of the virtual test bench. It divided into the following four stages:

- Simulation integration architecture initialization: the verification and validation team will implement the simulation framework initialization according to the system or interface to be tested;
- Identify the simulation integration architecture interface: according to the inherited logical architecture information, identify other subsystems that are cross-linked with the simulation model;
- Edit and improve the simulation architecture: edit the simulation architecture according to actual business needs, including adding and deleting corresponding connections, etc.;
- Define the simulation test interface: according to the simulation test goal, screen each simulation module interface under the simulation architecture to reduce those unnecessary tested interfaces, thereby reducing the weight of subsequent integration and verification.

3.4 Test Mean Architecture Design

The test mean architecture design is completed by the test design team, which specifically involves two types of activities:

- Define VCR, which is the basis for building a virtual test bench;
- Define the simulation solution. The test design team selects the simulation engine configuration according to the defined VCR, and automatically divides all modules in the simulation architecture to generate the simulation specification definition of the virtual component, and prepare for the detailed definition and configuration of virtual components in the next step.

3.5 Virtual Component Definition & Management

The detailed definition of the virtual component is divided into the following two types: 1) the interface definition of the virtual component; 2) the behavior definition of the virtual component.

The management activities of virtual components are divided into the following five steps[4]:

- Generate virtual component by third-party software: avionics virtual integrated testing environment provides third-party open data interfaces, so that models in the form of Simulink, FMU, etc. can directly generate corresponding virtual components;
- Virtual component development and configuration: after the establishment of the virtual test bench V&V strategy and test method architecture design stage, the avionics virtual integrated testing environment automatically inherits the generated simulation architecture definition and simulation module information, and realizes the initial generation from the simulation module to the virtual component configuration;
- Detailed definition of virtual components: the detailed definition of virtual components includes the view of virtual component interface information and the generation of ICD in the form of XML. The avionics virtual integrated testing environment provides Linux and Windows operating system compilers to increase the flexibility of subsequent testing;
- Generation of virtual component implementation framework: the simulation design team automatically generates the framework code of the virtual component through the avionics virtual integrated testing environment. The code defines the detailed information of the virtual component interface and the functional framework of the virtual component's internal functional logic. The model supplier will participate in the internal business implementation in the future;
- Verify the delivery of virtual components: the virtual component management team conducts preliminary verification on the functions and interfaces of the virtual components delivered by the model supplier to ensure the correctness of the interfaces and functional logic before the simulation integration.

3.6 Virtual Component Implementation

Subsystem suppliers perform internal implementation and factory-level verification and validation according to the detailed description of virtual components provided by OEMs, so as to standardize the functional logic, improve the iterative efficiency of virtual components, and achieve high-quality rapid delivery.

3.7 Virtual Test Bench Design and Delivery

For the construction of a virtual test bench, it is first necessary to confirm the correctness of the matching between the VCR and the VCS, and then it is necessary to integrate multiple virtual components that have been implemented by the model supplier. On the basis of the original integrated simulation architecture interface connection, the avionics virtual integration test environment will provide corresponding integration interfaces for integrators to integrate external interfaces. After the interface integration is completed, the design of the virtual test hardware bench and the allocation of virtual components to the simulation execution engine will be carried out. Finally, the simulation execution engine is assigned to the virtual test hardware bench to complete the construction of the virtual test bench.

After the virtual test bench is built, it needs to be verified according to the VTB architecture design process, and finally delivered to the user terminal to carry the test implementation.

3.8 Integration Testing

The virtual integrated testing environment packs virtual components and VTB to realize the configuration of integrated test software. The software will automatically identify the internal logic, interface information and connection relationship of each virtual component, and provide a real-time simulation test bench. Technicians perform early virtual test activities based on the previously defined test boundaries and tested objects, and complete the generation and analysis of test cases based on the virtual test environment.

4. Virtual Integration Test Method Design

On the right side of the V-shaped model, according to the verification requirements for each design stage, the verification methods for the operational scenarios and the system analysis stage are summarized.

4.1 Operational Scenario and System Analysis Phase

Sort out the use scenarios of the integrated test bench, analyze the test scenarios, simulation scenarios, fault injection scenarios, test management scenarios and system operation scenarios, and realize the early distributed simulation test through the SysML-based distributed simulation environment. Based on this goal, the use scenarios and basic architecture of the integration test bench are summarized, as shown in Figure 3.

4.2 Logical & Physical Architecture Verification phase

As shown in Figure 4, in the logic architecture and physical architecture verification stage, for each logical architecture and physical architecture model, the simulation framework is built and the code of the simulation model is automatically generated through the digital simulation and comprehensive test environment, so as to complete the internal implementation of the simulation model. Then through the simulation model integration configuration, the simulation model is input into the integrated test environment for real-time simulation testing, and finally the verification of the logical architecture and the physical architecture is completed.

- The definition of virtual test strategy and simulation architecture;
- Automatic generation of virtual test component interface framework;
- Internal implementation and delivery test of virtual test components;
- The generation of virtual test system test integration and configuration;
- Model-based virtual test execution of avionics system.

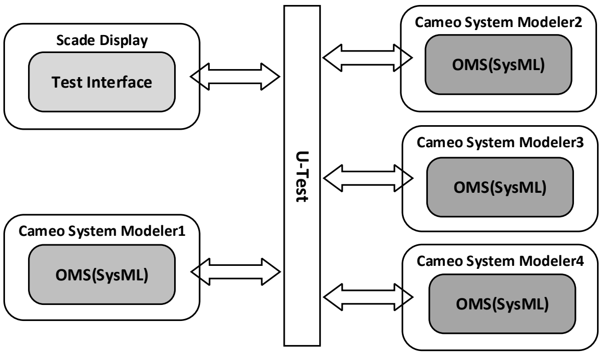

Figure 3 – Operational scenario simulation

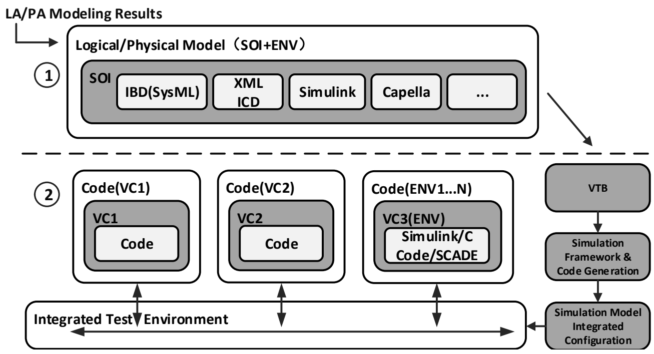

Figure 4 – Logical/Physical architecture verification

5. Virtual Integration Test Environment Development

Avionics Virtual Integration Platform (AVIP) mainly provides a basic operational environment for model-driven virtual integration and early verification of avionics systems[5]. The development process of this environment is divided into four parts: the development of model import capabilities, the model to virtual part allocation and visualization generation development, virtual component interface definition and framework generation and development, virtual test environment development and virtual test configuration generation[6]. Configure the data bus to the test environment to complete the test verification of the target model. The specific verification content includes the following four aspects:

- Quickly evaluate the new methods required for model development: through the dynamic function, interface and performance test verification of the system, the rapid evaluation of the system prototype is realized;

- Support model import and verification of ICD format, SysML, Capella, Simulink, MWorks;
- Virtual component interface information verification: It can generate the interface definition of each virtual component, and generate the interface information inherited by the model according to the needs of the test, includes two optional modes:
  - If the virtual components are all from the ICD file, the virtual component interface can be automatically assigned according to the connection relationship of the source model, and the interface is only related to the model corresponding to the virtual component under the test simulation architecture;
  - For other types of models, you need to manually connect the virtual component interface when the virtual test bench is generated;

  After the virtual component interface is generated, users can choose the framework mode according to their needs, including two optional modes:

  - Framework code based on C language;
  - Simulink framework (including external interfaces);

    Model suppliers can implement the internal implementation of virtual components according to different modes. At the same time, the verification environment needs to have certain compilation capabilities to achieve integrated compilation of the model framework and model implementation.
- Virtual integration test configuration verification:

  This virtual integrated test environment defines standards for the interaction of virtual components supported by existing commercial software, which can effectively alleviate the problem of excessive dependence on suppliers. At the same time, the environment also provides a virtual integration process and data interoperability, which can support up to 200,000 avionics system signals and more than 200,000 simulation parameters associated with airborne systems.

6. Conclusion

The supplier provides the main manufacturer with a system model framework (functional prototype) according to the system integration verification requirements, and then the main manufacturer can perform system-level virtual integration test of the "black box" model (virtual component) provided by the supplier according to the predefined integration process. The method proposed in this paper can confirm the correctness of system functions and interfaces through dynamic simulation of the model, realize rapid prototype verification of supplier delivery model and evaluation of system alternatives, provide design input for the system realization stage, and provide early verification support for the system integration phase.

7. Contact Author Email Address

Peipei Xing, email: xingpeipei@comac.cc

References

[1] Jiang X, Shaofan Z, Tang J. Model-based systems engineering for the design of civil aircraft avionics system. Proceedings of the 17th AIAA aviation technology, integration, and operations confer- ence. Denver, USA. Reston: AIAA; 2017, p.AIAA 2017-4395.

[2] Yue T, Briand LC, Labiche Y. An automated approach to transform use cases into activity diagramsProceedings of the European conference on modelling foundations and applications, Paris, France. Berlin Heidelberg: Springer; 2010. p. 337–53.

[3] Löding, H. & Peleska, J., Timed Moore Automata: Test Data Generation and Model Checking. International Conference on Software Testing, Verification, and Validation, 2010, Paris, pp. 449-458.

[4] Peleska, J., Industrial-Strength Model-Based Testing - State of the Art and Current Challenges, Petrenko A., Schlinglo H. (Eds.): Eighth Workshop on Model-Based Testing (MBT 2013), EPTCS 111, 2013, pp. 3-28.

[5] J. Peleska, "Model-based avionic systems testing for the airbus family," 2018 IEEE 23rd European Test Symposium (ETS), Bremen, Germany, 2018, pp.1-10.

[6] Weilkiens T. SYSMOD-The systems modeling toolbox-pragmatic MBSE with SysML. MBSE4U Publishing Organization; 2016.

Copyright Statement