

Optimizing the Benefit of Virtual Testing with a Process-Oriented Approach

Jann-Eve Stavesand, Sören Reglitz, and Andreas Himmler
dSPACE GmbH

ABSTRACT

In the aerospace industry, methods for virtual testing cover an increasing range of test executions carried out during the development and test process of avionics systems. Over the last years, most companies have focused on questions regarding the evaluation and implementation of methods for virtual testing. However, it has become more and more important to seamlessly integrate virtual testing into the overall development process. For instance, a company's test strategy might stipulate a combination of different methods, such as SIL and HIL simulation, in order to benefit from the advantages of both in the same test process. In this case, efforts concentrate on the optimization of the overall process, from test specification to test execution, as well as the test result evaluation and its alignment with methods for virtual testing. Furthermore, software tools have to be suitable for virtual testing and the entire tool chain has to be adaptable so it can be used for different test methods for various applications that have different users.

This paper introduces an exemplary validation and verification process according to ARP4754A and describes how virtual testing contributes to a more efficient overall development process. The ARINC 653 software architecture serves as an example for demonstrating the benefits of using different virtual test environments. In order to coordinate and manage the design, implementation and test process, all test and simulation data is stored centrally using a data and test management tool that reduces complexity for an improved overview. Examples from real projects will illustrate this process-oriented approach.

CITATION: Stavesand, J., Reglitz, S., and Himmler, A., "Optimizing the Benefit of Virtual Testing with a Process-Oriented Approach," *SAE Int. J. Aerosp.* 10(1):2017, doi:10.4271/2017-01-2114.

INTRODUCTION

Virtual testing is well-established in the aerospace industry. Hardware-in-the-loop (HIL) simulation plays an important role, as it is the most industry-proven method for virtual testing. New demands that arise from the evolving architectural design of E/E systems and from process engineering conditions have their effects on this well-known method. From a technical perspective, virtual testing has to consider that E/E systems are more interconnected and functionalities are implemented across multiple electronic control units (ECUs) [1]. The systems are linked to the test environment, and the high number of system versions and variants significantly increases the volume of data, which in turn has to be handled and managed. The environments for virtual testing have to be suitable for testing these sophisticated ECU networks to enable simulation-based verification. From the perspective of process engineering, the virtual test environments also have to be able to comply with process descriptions of standards and recommended practices, such as ARP4754A [2]. Most of the functionalities of the advanced avionics systems are safety-related and the development and verification has to be planned and executed with the appropriate thoroughness. The methods used for development have to meet these requirements and have to be suitable to transfer the standards' generic process descriptions to real applications.

To efficiently structure and manage the development processes for E/E systems while complying with standards, the whole process from specifying and executing the test to evaluating the test results has to be considered. The tools and verification environments have to be integrated into an overall process. For example, using a high-end simulator is only one aspect of many. The simulator also has to be integrated into an overall verification process to address the new demands. The integration requires a tool chain that can be adapted to various processes and applications used by different users. The tool chain also has to enable users to efficiently customize the process for testing functional safety. Furthermore, the tasks and activities within these processes have to be optimized, i.e., automated, to minimize the influence of human error, for example. This requires interfaces between and supported by suitable tools.

This paper demonstrates that a central test and data management tool is essential for the previously described process-oriented approach. The data management tool makes it possible to store the relevant test artifacts throughout the entire test process, making the test workflows traceable and transparent by means of automation or at least partial automation of the test activities. The data management tool provides access to all data relevant for automation and makes it possible to use it along the entire development process as well as for monitoring tasks. Real examples demonstrate how using the development and test methods presented in this paper improves the efficiency of well-established test methods.

The paper gives an overview of the typical challenges that occur during the development and testing of E/E systems regarding workflow artifacts. It focuses on different simulation scenarios in the test process and identifies artifacts for potential reuse. Furthermore, the paper demonstrates why a central data management tool is indispensable for achieving reusability and traceability of the artifacts. It depicts what additional advantages a central data management tool can provide for automatic test configuration. Lastly, it describes how demands and recommendations from standards like ARP4754A [2] and ARINC 653 [3] can be met efficiently by focusing on process engineering issues and aspects of process tailoring.

PROCESS-ORIENTED TESTING

A detailed test process is an inherent part of the development process of E/E systems, as it gives users the required confidence in the quality of the embedded software. Several factors are considered when assessing software quality. This paper mainly emphasizes the factors that are most important for the quality of embedded software of ECUs, e.g., reliability, compatibility, performance efficiency, and functional suitability [4]. The software of the avionics systems in the aerospace industry has to be reliable, which means that the systems have to be robust against induced faults to ensure functional safety, for example. The interfaces of the software, and especially software components in the embedded software, have to be compatible with different integration scenarios. Therefore, the interfaces of the software components have to be clearly defined to enable an efficient integration along the development process and facilitate incremental software development. In addition, the software has to be performance-efficient so it can be executed on the ECUs. The execution time on the hardware has to be reliable and, in most cases, the hardware must be real-time-capable to ensure the reliability of safety mechanisms and safety-critical program sequences in the embedded software. The functional suitability of the software for the intended tasks is particularly important. A high software quality is essential for building a safe and efficient E/E system. Therefore, and because E/E systems have become increasingly complex over the last years, the evaluation of the software quality has to be structured and managed efficiently. In consequence, the quality management for the embedded software and related hardware constitutes a considerable challenge for manufactures of aircraft systems today [5]. To improve test efficiency and thus quality management itself, the whole development process has to be taken into account, because artifacts of the entire development process are used for the test tasks. The test process has to follow a clearly defined taxonomy, such as the one stated in ARP4754A. This means that the optimization of verification activities has to start at the beginning of the development process and continue throughout the whole process, from the initial requirement of the E/E system to the final test phase before production starts (Figure 1). Therefore, each phase of the development process, including the relevant test tasks, has to be scrutinized with regard to data exchange, the provision of standard interfaces, requirements management, etc.

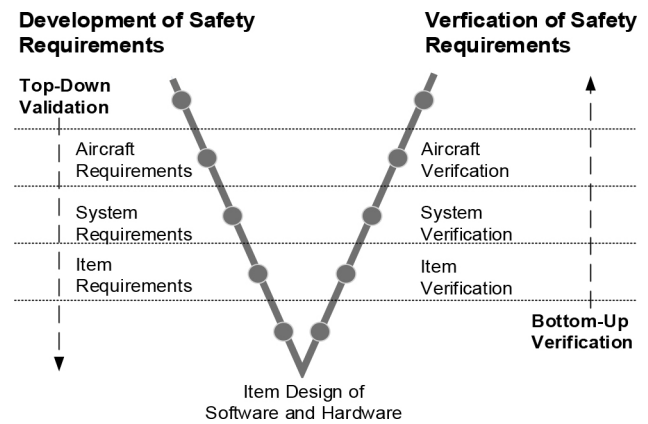


Figure 1. The validation and verification process according to ARP4754A.

The V-cycle has long been established in various industries as the model for development and test processes [6]. It describes the required validation and verification tasks. In general, the right branch of the V-cycle contains the individual test phases and integration levels for testing E/E systems. Integration gradually moves from single software components to the entire software running on networked avionics systems. This involves many steps, which differ in test object (system under test), test objective, test phase, test environment¹, and the related work packages. The main challenge is to coordinate these steps and use them consistently and reliably throughout the entire verification process. This way, the high number of tests can be carried out efficiently within the available development time. The input from the left branch of the V-cycle is needed to execute the tests. It includes behavior models for virtual test methods, requirements for test designs, etc. This input also has to be managed. On the one hand, this is a challenge, because extensive planning is required. On the other hand, it is an opportunity, because managing the input helps avoid redundant work products in the development process. For example, models are developed once and can be reused throughout the development cycle.

The model for the verification process introduced by ARP4754A (Figure 2) describes the general verification workflow recommended for aircraft systems, including the required inputs and generated outputs as well as dependencies on the upstream development process [2]. Development tasks preceding the verification process include the formulation of requirements according to the associated development assurance level of the function or item as well as its design process, which results in a testable implementation. The verification process is applied iteratively at different integration levels, such as the item, system and aircraft integration levels.

1. See [7] for a detailed definition of all software testing terms used in this paper.

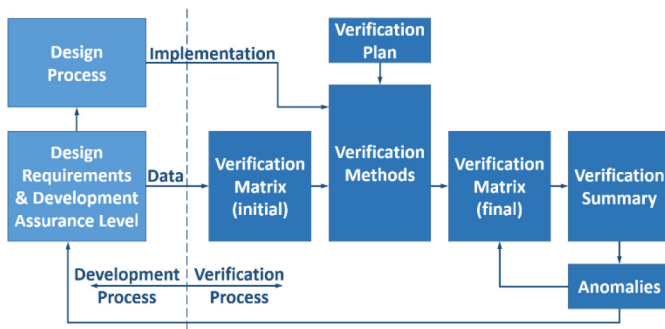


Figure 2. ARP4754A model of the verification process.

The process defines four basic verification methods for evidencing that the function or item under test, i.e., the test object, meets its functional requirements [8]:

- Inspection or review
- Analysis (e.g., modeling or coverage analysis)
- Test or demonstration
- Service experience

For the scope of this paper, ‘test or demonstration’ is of particular interest. According to ARP4754A, testing demonstrates that a test object fulfills its intended function and provides confidence that it does not perform unintended functions that impact safety. This can be achieved by using representative software or hardware together with either the physical system or appropriate validated models. This makes it possible to use different virtual test environments, depending on the test object and the test phase. The virtual test environments are introduced and described in detail in subsequent sections of this paper.

Depending on the development assurance level (DAL) of the test object, one or more verification methods have to be used. When aircraft systems are developed according to standards such as DO-178C for avionics software, each function or item is assigned a specific DAL, which depends on the function’s or item’s failure condition classification. For example, functions or items whose failure is classified as catastrophic or hazardous have DAL A or B. In this case, ARP4754A recommends test or demonstration and one additional verification method in order to show compliance with applicable airworthiness regulations. This underlines the particular importance of testing in the verification process of safety-critical functions or items in aircraft systems.

Each of the verification methods above requires specific inputs and generates outputs that have to be managed and linked to the overall development process in order to establish traceability between requirements and test results, reproducibility of the verification task and reusability of the related artifacts, e.g., simulation models and test scripts, for subsequent verification phases. Other outputs defined by ARP4754A include a specific set of verification data to provide evidence that the verification process was completed. The standard introduces verification matrices, plans, procedures and summaries to use as evidence. This evidence is one of the requirements for later certification credit.

A verification plan determines the verification methods to be used depending on the associated DAL to fulfill the requirements. It defines the success criteria for assessing the verification results and the related verification credit. In addition, the plan identifies the test object configuration and determines test equipment and facilities. In the context of this paper, this means that the verification plan defines which virtual test environment has to be used for specific test objects and how the related test activities contribute to verification credit. The verification matrix links requirements and associated functions from the development process to verification methods, procedures and results of the verification process. The verification summary additionally includes information about significant deviations from the plan, open problem reports and their impact on safety, supporting test data as well as references to the verification plan and matrix.

The integration of a central data management system in the development and verification process of aircraft systems provides essential support to efficiently manage the previously described verification data. It offers the necessary interfaces between all tools involved in the different processes, such as requirements engineering, configuration management and test automation tools. The general approach and additional benefits of a central data management system are described in subsequent sections of this paper.

In the model-based development of avionics systems, the verification process predominantly involves hardware-in-the-loop simulation as the virtual test environment. This type of validation can be complemented by PC-based simulation systems, which can be used in earlier development phases when hardware prototypes are not available yet.

In addition to the well-established test methods that use test benches or test flights, hardware-in-the-loop simulation is an efficient and industry-proven step for reliably managing the pending test tasks. However, there is still considerable potential for moving current flight tests to earlier test phases with the support of hardware-in-the-loop simulation. The functional scope and thus size and shape of hardware-in-the-loop systems differ depending on the validation task in the test process. For example, a different effort is required when validating single software components compared to partial networks in the final aircraft. Mechatronic test benches can also be enhanced with simulation-based methods to allow a model-based approach.

To increase the efficiency of these test methods, the tool chain has to support all artifacts needed for setting up simulation scenarios, e.g., models and parameters of the test objects, and it has to be able to manage the overall development and verification activities.

Test Objects & Evaluation Criteria

An essential part of structuring the test activities is scheduling test phases. Scheduling test phases is directly linked to the available test objects. Questions like “When is a software component ready for testing?” or “When will a first prototype of actuators be available?” have to be answered when planning the verification process and the test phases along the V-cycle (Figure 1). According to ARP4754A, the test objects can be the systems and subsystems under investigation, e.g., single electronic control units at the item