

Tool Chain for Avionics Design, Development, Integration and Test

Martin Halle

Institute of Aircraft Systems Engineering (FST)
Hamburg University of Technology (TUHH)
Hamburg, Germany
Martin.Halle@tuhh.de

Frank Thielecke

Institute of Aircraft Systems Engineering (FST)
Hamburg University of Technology (TUHH)
Hamburg, Germany
Frank.Thielecke@tuhh.de

Index Terms—avionics, tool chain, IMA, design, development, integration, test

Abstract—The design, development, integration and test of avionics systems is a complex task. Several national and European projects aimed at improving the methods and tools for new IMA platforms. Since about 12 years, the Institute of Aircraft Systems Engineering of the Hamburg University of Technology continuously contributed to such projects. This paper gives an overview about the tool chain that has been developed so far and addresses a new extension in the field of avionics tests and its automation that will be developed in on-going and future projects.

I. INTRODUCTION

Avionics are based on a generic, modular platform (Integrated Modular Avionics, IMA [1]) and serve system applications with the computing and I/O resource needs. Over the years and different aircraft programmes (i.e. B777, B787 or A380, A350), the avionics system has been further developed towards a distributed platform with different types of computing modules, different types of I/O and more and more applications running on IMA.

In the future, it is likely that IMA will expand into other areas like cabin and flight control, but also new capabilities like multi-/many-core processors and I/O technologies like wireless or optical communication or combined I/O concepts like data-over-power will be introduced. Such technologies will be the enabler for a modern avionics platform and will increase the freedom for the platform- and system-designers. However, the burden of handling the complex overall design space will increase, too. Manual design methods will likely become too error-prone or even impossible but at least non-optimal.

Ongoing research of the Institute of Aircraft Systems Engineering (FST) of the Hamburg University of Technology (TUHH) addresses an approach to an avionics-centred double-V process as shown in figure 1.

The double-V stems from the idea to have a model-based seamless tool-chain that supports the development process not only by the tools but also by enabling early validation and test. Using simulations and models that are derived from data and information based on the current level of detail available, a digital twin of the avionics platform allows its validation at any time in the development process. The FST develops

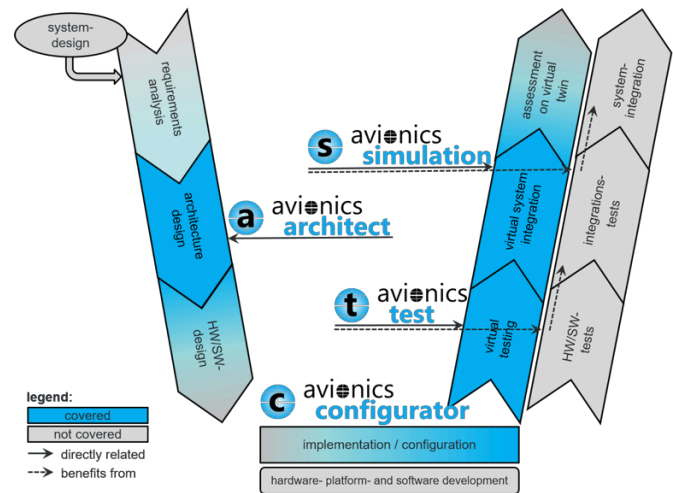


Fig. 1. Avionics double-V-process

a seamless tool-chain to demonstrate possible methods and automatise process steps as much as possible. New in the tool-chain is deriving re-usable and mostly generic tests procedures for different test-platforms.

While the FST has a strong background in system testing and virtual integration [2] [3] [4] [5] [6], partially including IMA [7], so far, IMA has been either provided as-is or was not considered at all. Therefore, the influence of the IMA platform on the system function and vice versa in preliminary system design was hard to investigate. Because of that, the seamless tool-chain is extended to allow IMA platform simulations hosting system applications on virtual IMA modules to allow studying and test the functional behaviour of the system functions with respect to new IMA approaches.

The paper is organised as follows: First, the different tools of the seamless-tool chain that already exist and how they fit into the double-V are explained. Then, the approach for simulation based avionics test will be explained. The paper ends with a summary and outlook.

II. AVIONICS ARCHITECT

When starting to design a new avionics platform or updating an existing one a lot of decisions have to be made. What systems/system applications will utilise IMA; what resources

require these applications; what I/O needs to be supported by the platform as well as where and what installation locations can be used. To to derive a valid architecture, additionally system- and certification constraints have to be take into account.

For such purposes, a model-based methodology has been developed [8] that allows to formulate and capture these requirements is a formal way that allows to be further processed and to derive an optimised IMA platform. The requirements are captured in a rather generic way and can either be input manually or imported as tables which contain:

- Software tasks and their attributes like resource requirement (I/O, memory, redundancy/segregation constraints, ...);
- Signals to be exchanged between tasks and attributes like periodicity or bandwidth
- Physical system peripherals like sensors or actuators and their location as well as attributes like weight, dimensions, ...;
- Devices that can host tasks and provide resources or are required for I/O like switches. Additional attributes can be captured like weight, cost, power supply and others;
- The anatomy of the aircraft or vehicle to describe installation locations for devices or peripherals and cable routes including attributes like capacity, volume, available resources and alike.

The information is structured and linked based on a meta-model in an Eclipse-based [9] application. The software-framework that implements the methodology and provides a graphical frontend to the user is called *Avionics Architect* and shown in figure 2.

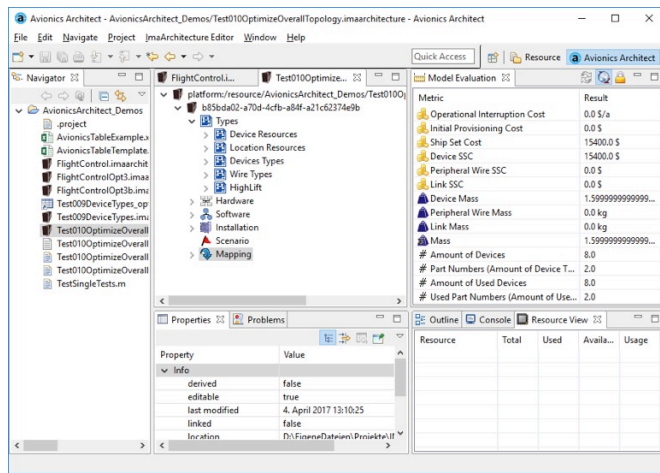


Fig. 2. Avionics Architect

In the V-model its use is in the left-hand side when the requirements are captured. This includes the requirements for the IMA platform to e.g. derive the specification for IMA modules but also the requirements of the system applications to achieve a common understanding and integration database between the integrator and the system departments.

Similar tools from platform suppliers have been developed [10] [11] but they are usually limited to the modules of that

supplier and do not allow for an optimisation at aircraft level.

III. AVIONICS CONFIGURATOR

After the design of the avionics platform is done the function and configuration development starts. Besides the actual system software applications the configuration for IMA modules plays an important role. It consists of thousands of parameters that define the application (partitions) as well as physical and logical I/O parameters. For specific IMA modules several other parameters like for combinatorial logic are included, too. Configuration tools are provided by the respective module suppliers for their dedicated IMA modules whereas the actual configuration is managed by the OEM by means of a database and configuration documents. These configuration documents are often hand-crafted using tools like Excel in comma-separated-values (CSV) format.

Due to the fact that this is often error-prone, a new model-based concept for creating and managing configuration data at aircraft level has been developed by FST [12] [13]. For such purposes, a model-based configuration management concept and software-framework namely *Avionics Configurator* was developed and is shown in figure 3.

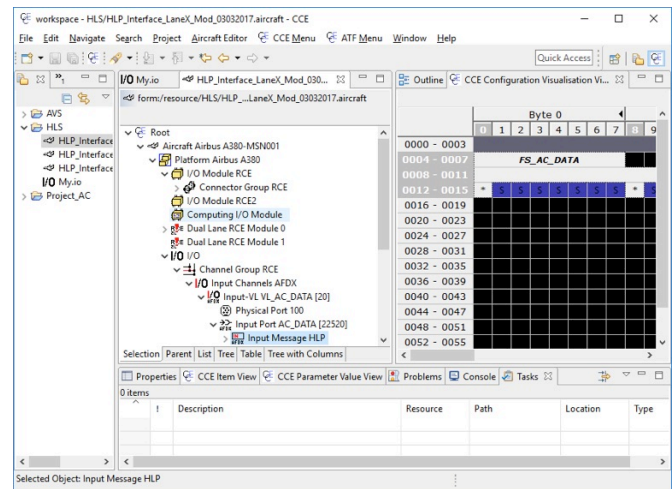


Fig. 3. Avionics Configurator

It allows to capture all configuration parameters in a supplier independent format in one tool. It replaces the need for table-based editing with duplicated information by using a linked meta-model and guided input. Graphical visualisations and model-based verification of the users input improve the consistency of the configuration data early in the development process. It is not a replacement for the qualified tool-chain of the module supplier though, but can export the input files needed for these tools e.g. for a qualifiable validation. In the V-model its use is currently in the implementation phase.

Because *Avionics Architect* and *Avionics Configurator* share the same philosophy of meta-modelling and also the same modelling language (Ecore from the Eclipse Modelling Framework, [9]) in [14] a methodology has been presented that allows to create configuration stubs directly from the architecture data using a formal model-to-model transformation. Thus, all configuration-relevant information that was already captured

during the architecture phase will be derived automatically following the philosophy of a seamless tool-chain [15].

IV. AVIONICS SIMULATION

Knowing the architecture of an IMA platform, the functions, the I/O types and signals between function blocks and the configuration of system applications, virtual integration and functional validation becomes possible.

When it comes to functional validation of system applications, often the algorithms behind these applications are developed in Matlab/Simulink or similar. The timing behaviour of the IMA platform and the I/O interfaces need to be considered as good as possible for functional validation. To address this issue, a simulation-framework namely *Avionics Simulation* has been developed at FST that consists of Matlab/Simulink-based models to emulate the behaviour of IMA platforms and communication interfaces with respect to their timing, nominal and faulty behaviour [16]. It is shown in figure 4.

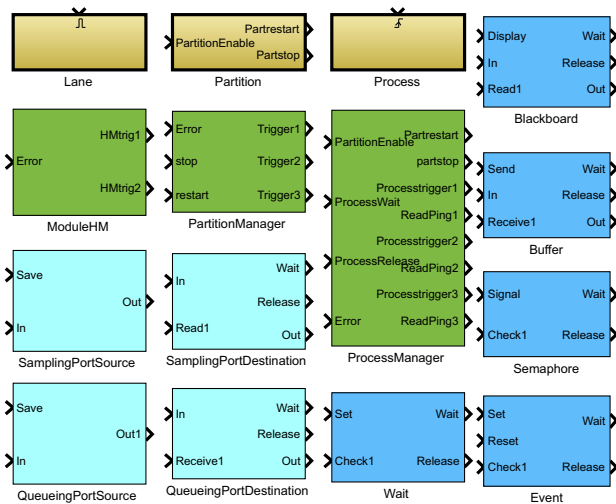


Fig. 4. Avionics Simulation

The simulation-framework consists of models for IMA modules, schedulers, health management and I/O blocks. The latter for IMA module internal functions (like ARINC 653 ports, buffers or blackboards [17]) but also external I/O like AFDX, CAN or analogue/discrete busses. For a seamless tool-chain, a model generator takes the architectural information from the *Avionics Architect* and the configuration details from *Avionics Configurator* to generate an overall simulation model stub that consists of the allocated IMA modules, partitions for the system applications and the communication between the IMA modules (AFDX network) including the logical signals. Technically this is done using the automation interface of Matlab/Simulink. Embedding the developed system functions into this model is demonstrated in [18]. It allows for simulation-based, virtual early validation studies of system applications under consideration of the IMA platform at aircraft level. In the V-model its use is in right-hand side and can already start, when hardware is not yet available.

V. AVIONICS TEST

A new project continues the work and aims at model-based or hybrid virtual testing in a more systematic and automated manner. As already mentioned, using the architectural and configuration data, simulations can be derived that are used for nominal and failure case testing. As explained in [18], functional tests can be executed on these models. So far, the tests were manually derived and executed. The overall goal of such tests is to ensure functionality of system applications on the designed platform in early design and development stages. Thus, design limitations of the platform can be found. Consequently, such functional tests should be re-usable as soon as hardware and/or equipment becomes available.

To do so, an at least semi-automatic derivation of test cases and a test engine (*Avionics Test*) that can conduct and document these tests is desired. Although system requirements can be captured in *Avionics Architect*, this type of information is not yet consequently used for test automation although it is already available in a structured, model-based and machine-readable fashion. Alternatively, requirements databases like Doors could be used.

To conduct a meaningful test and test automation, more information is needed. At FST, a generic test environment for avionics systems is about to be established. The principle is shown in figure 5.

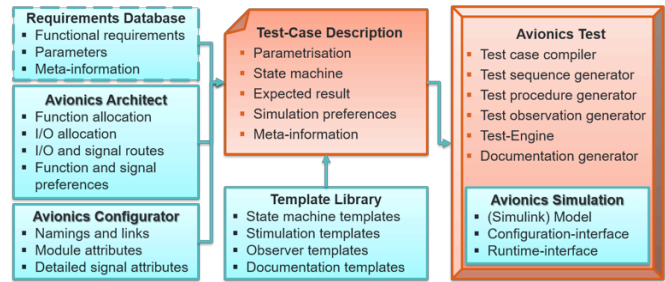


Fig. 5. Avionics Test

Assume an aircraft door system with 5 proxy sensors and avionics system functions hosted on an IMA modules to read their states and to visualise a consolidated state in the cockpit. For a test case of the function that validates a "cabin door closed and locked" scenario, the following data is obtained from the respective sources:

- From a requirements database the functional requirements needed for the test case are derived. That is, what and how many proxy sensors must be in what state to confirm the door is closed. Additionally meta-information like test case ID and other information for traceability are obtained.
- From the architecture model, the function and I/O allocation including the signal path physical wiring are obtained. This also includes the instances of IMA modules hosting the respective functions or sub-functions.
- From the configuration model, attributes like periods and detailed signal attributes like sampling times, type an size of data are obtained. This also includes the

concrete signal names and protocol encapsulation (i.e. functional data set structures with signal positions for AFDX messages).

- From the architecture and configuration model, the model of the IMA platform is derived and instrumented with system applications for simulations as explained earlier.

To use the simulation model for testing, interfaces for configuring the simulation itself, different block parameters like buffer sizes or signal names and methods to inject test-procedures and observers are required and need to be implemented. Furthermore, a runtime-interface that controls the simulation, injection and recording of data is needed.

To formulate the test cases and test sequences in a Matlab/Simulink compatible fashion they shall be expressed in Simulink/Stateflow, similar to the SCXML notation [19]. A library that consist of parameterisable standard test steps as state machine templates is under development. Using the Matlab/Simulink automation interface, such templates can be instantiated to simplify the generation of the test procedures. Similar to that, common input patterns for stimulations and output sinks for observation are provided through other libraries. The generated test-procedure is expressed in Stateflow (an example is shown in figure 6) and connected to the system- and IMA platform simulation via input- and output-ports.

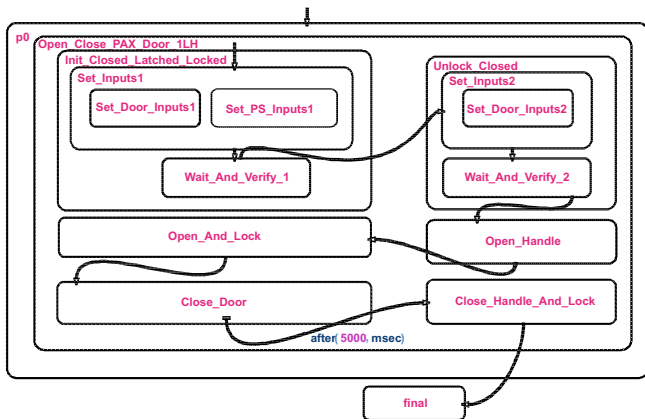


Fig. 6. Avionics Test Procedure

Preliminary prototypes show the feasibility of the approach. However, a lot work still needs to be done. In the end, this approach shall enable to conduct functional tests across different IMA platforms using different technologies without the need to rewrite or reconfigure the tests manually. For hybrid testing the integration of hardware test-benches should also not affect the tests.

VI. SUMMARY AND OUTLOOK

This paper focusses on a new project at FST towards Avionics Testing. For many years the FST developed a sophisticated tool-chain for the design, implementation and simulation of IMA platforms including and focussing on system function applications hosted on avionics. Consequently, a new approach for automatic testing of system applications using a seamless tool-chain is under development and has been introduced

hereby. The scientific question in this stage is how far does this concept work and what type of tests can be accomplished to an useful extend. Also, the institute is seeking for a standardisation of system tests including avionics. Some of the remaining issues are how to formalise data formats and data management. Other questions are how to derive test relevant parameters that are often not expressed in machine readable format like timing behaviour constraints. For test automation, besides Matlab/Simulink also existing HITL test-systems available at FST are investigated.

REFERENCES

- [1] H. Butz, "The Airbus approach to open integrated avionic modular avionics (IMA)," in *Workshop on Aircraft System Technologies (AST)*, 2007.
- [2] J. Grymlas, T. Kreitz, and F. Thielecke, "Scenario-based testing of multifunctional fuel cell systems using the virtual integration platform VIPER," in *International Conference on Fundamentals and Development of Fuel Cells (FDCC), Toulouse 3 - 5 February 2015*, Toulouse, 2 2015.
- [3] T. Kreitz, R. Doering, S. Jäger, and F. Thielecke, "System-level virtual integration study of an aircraft electrical power network," in *Proceedings of the 5th International Workshop on Aircraft System Technologies (AST 2015)*, Hamburg University of Technology. Shaker Verlag, 2015.
- [4] T. Kreitz and F. Thielecke, "Virtual integration of an all-electric flight control system architecture and the aircraft electrical power distribution network," in *SAE 2016 Aerospace Systems and Technology Conference*. Hartford, CT, USA: SAE, 9 2016.
- [5] J. Grymlas, "Systemautomatisierung für den multifunktionalen Betrieb von Brennstoffzellen in Verkehrsflugzeugen," phdthesis, Hamburg University of Technology, 12 2017.
- [6] L. Nordmann, F. Thielecke, P. Lücken, and M. Hamm, "Virtual testing of aircraft hydraulic systems," in *Recent Advances in Aerospace Actuation Systems and Components, May 30 - June 1, 2018, Toulouse, France*, 2018.
- [7] S. Benischke, "Modellbasierte Entwicklung eines verteilten Antriebssystems für ein multifunktionales Landeklappensystem," phdthesis, Hamburg University of Technology, 6 2018.
- [8] B. Annighöfer, "Model-based architecting and optimization of distributed integrated modular avionics," phdthesis, Hamburg University of Technology, 3 2015.
- [9] D. Steinberg, F. Budinsky, M. Paternostro, and E. Merks, *EMF: Eclipse Modeling Framework (2nd edition)*, 2nd ed. Addison-Wesley Longman, Amsterdam, 2008. [Online]. Available: <http://my.safaribooksonline.com/9780321331885>
- [10] M. Fumey, "Avionics systems EVT: Early validation tools," in *SAE Technical Paper*. SAE International, 10 2011.
- [11] E. Stav, "A sirius based function modelling tool for the avionics domain," in *Eclipse Neon DemoCamp, Trondheim*, 2016.
- [12] M. Halle and F. Thielecke, "7konfigurationsmanagement für integrierte modulare avionik."
- [13] —, "Next generation ima configuration engineering – from architecture to application," in *34th Digital Avionics Systems Conference (DASC)*, 2015.
- [14] —, "Model-based transition of IMA architecture into configuration data," in *35th Digital Avionics Systems Conference (DASC)*, 2016.
- [15] —, "Evaluation of the ASHLEY seamless tool-chain on a real-world avionics demonstrator," in *36th Digital Avionics Systems Conference (DASC)*, 2017.
- [16] B. Annighöfer, H. Höber, and F. Thielecke, "An easy-to-use real-time AFDX simulation framework," in *35th Digital Avionics System Conference*, Sacramento, CA , USA, 9 2016.
- [17] ARINC, *ARINC 653 Standard, "Avionics Application Software Standard Interface"*, Aeronautical Radio Incorporated (ARINC), 2006.
- [18] H. Höber and F. Thielecke, "Eine simulationsbasierte Methode zur frühzeitigen Validierung von Avionik-Plattformen," in *Deutscher Luft- und Raumfahrtkongress (DLRK) 2018*. Friedrichshafen: Deutsche Gesellschaft für Luft- und Raumfahrt, 9 2018.
- [19] World Wide Web Consortium (W3C). (2015) State chart XML (SCXML): State machine notation for control abstraction. [Online]. Available: <https://www.w3.org/TR/scxml/>