Chapter 19
A Model-Based Approach to Configure and Reconfigure Avionics Systems

Julien Delange
TELECOM ParisTech, France

Laurent Pautet
TELECOM ParisTech, France

Fabrice Kordon
Université P. & M. Curie, France

ABSTRACT
Aircraft manufacturers have been moving toward the Integrated Modular Avionics (IMA) approach to reduce the number of dedicated boxes in the aircraft. Standards such as DO178B or ARINC 653 must be followed during design, configuration or certification of IMA systems. Productivity and costs must also be improved while preserving conformance to standards. For instance, development process of avionics systems involves several system representations and representation transformations are done manually. Moreover, the complexity of new generation of safety-critical systems has also increased the complexity of their development. The authors present their component-based approach which relies on an appropriate modeling language (AADL) combined with modeling patterns to represent, configure and deploy an IMA system. It reduces costs by detecting errors earlier and prevents specifications revisions. Their code generator reduces memory footprint and improves code coverage. One last benefit is a possible automatic certification.

INTRODUCTION
Historically, aircrafts have been designed and built using a federated design approach, where functions are isolated from others by confining them into dedicated boxes. However, in a recent past, aircraft manufacturers have been moving toward the Integrated Modular Avionics (IMA)
A Model-Based Approach to Configure and Reconfigure Avionics Systems

A typical avionics system architecture is designed as a federated architecture of dedicated boxes. Each box can be a hardware/software configuration totally different to the others. The applications are physically protected from one another in such a way that failures of one or more applications do not affect others. However, such architectures are expensive to build in terms of space, weight, and power requirements. Moreover, adding new functions means adding new boxes. Thus, modifications and improvements usually convey serious drawbacks in terms of maintenance and operating costs.

To overcome those disadvantages, a new avionics architecture model, known as Integrated Modular Avionics (IMA), has been developed. IMA helps to reduce the number of dedicated boxes in the aircraft to a single computing platform running multiple applications. This approach reduces the space, weight, and power requirements of the aircraft, reduces spares holding, and therefore reduces complexity, costs, time, etc. In this environment, application code can be separated and tested independently, with the benefit of reducing the overall cost of software certification while maintaining safety requirements.

One of the major goals of the IMA approach is to enforce system configuration and incremental integration of new functionalities into a pre-existing system. Moreover, in order to exhibit fault tolerance in presence of errors giving any operational scenario, fault detection and health monitoring become a central point in IMA systems. Such systems must be reconfigurable to remain operational when a system component fails.

In order to provide an effective IMA system, standards have to be followed to allow for methodical development, validation and testing, as well as providing a standards-based Application Programming Interface (API) to allow for software portability and modularity. The ARINC 653 specification (Aeronautical Radio, Inc. 1997) is now a de-facto standard in Avionics. It provides a framework for building an operating system environment to support IMA. It specifies behavioral aspects such as scheduling and communications, as well as programming aspects such as an API set to call ARINC 653 methods.

However, it is currently hard to configure an IMA system. These safety-critical systems embed more and more functionalities year after year so that complexity of their development increases especially for design, configuration or certification. Due to their criticality, they must be carefully designed and comply against restrictive standards. For instance, adding new functions means reconfiguring the ARINC653 operating system environment so that both resource requirements and availability as well as integrity requirements are satisfied. It also means this new configuration has to be qualified and certified with regards to avionic software guidance documents (such as DO178B). This emerging complexity shows the need for a development and configuration process has to be defined to assist designers, developers or engineers and build the new generation of safety-critical systems.

To address these issues and ensure configuration correctness, requirements must be verified at the earliest stages of the development process to avoid errors traditionally detected during tests, integration or production. To do so, configuration is validated and its corresponding implementation code is automatically generated instead of being manually written by developers. This also avoids all errors introduced by “traditional” methods. In addition, generating configuration from validated artifacts eases their certification against standards (such as DO178B), and reduces their associated costs.

For that purpose, we design a dedicated process that automatically validates avionics system configuration, generates its code, builds the system and checks its requirements enforcement during its execution.
Related Content

Capturing Consumer Preference in System Requirements Through Business Strategy
[www.igi-global.com/article/capturing-consumer-preference-in-system-requirements-through-business-strategy/103315?camid=4v1a](www.igi-global.com/article/capturing-consumer-preference-in-system-requirements-through-business-strategy/103315?camid=4v1a)

An Ontology Based Representation of Software Design Patterns
[www.igi-global.com/chapter/ontology-based-representation-software-design/29408?camid=4v1a](www.igi-global.com/chapter/ontology-based-representation-software-design/29408?camid=4v1a)

Towards a Method Family Supporting Information Services Co-Creation in the Transdisciplinary Context
[www.igi-global.com/article/towards-a-method-family-supporting-information-services-co-creation-in-the-transdisciplinary-context/80196?camid=4v1a](www.igi-global.com/article/towards-a-method-family-supporting-information-services-co-creation-in-the-transdisciplinary-context/80196?camid=4v1a)

Increasing the Accuracy of Software Fault Prediction using Majority Ranking Fuzzy Clustering
[www.igi-global.com/article/increasing-the-accuracy-of-software-fault-prediction-using-majority-ranking-fuzzy-clustering/120519?camid=4v1a](www.igi-global.com/article/increasing-the-accuracy-of-software-fault-prediction-using-majority-ranking-fuzzy-clustering/120519?camid=4v1a)