Simulation-Based Verification of Avionic Systems Deployed on IMA Architectures

Tiyam Robati, Amine El Kouhen, Abdelouahed Gherbi, John Mullins

Simulation-Based Verification of Avionic Systems Deployed on IMA Architectures

Tiyam Robati*, Amine El Kouhen†, Abdelouahed Gherbi*, John Mullins‡
* Dept. of Software and IT Engineering
École de Technologie Suprieure, Canada
emails: Tiyam.robati.1@ens.etsmtl.ca, Abdelouahed.gherbi@etsmtl.ca
† Dept. of Engineering and Computer Science
Concordia University, Canada
email: elkouhen@encs.concordia.ca
‡ Dept. of Computer and Software Engineering
École Polytechnique de Montreal, Canada
email: john.mulins@polymtl.ca

***Abstract*—To build reliable avionic applications, we interconnect Integrated Modular Avionics (IMA) architectures with Time-Triggered Ethernet (TT-Ethernet). These systems have direct impacts on human lives where the failure is unacceptable. Therefore, verification is an important issue to ensure the safety and the performance of the system. The integration of IMA architectures is a very complex and challenging engineering task. To cope with complexity and to perform verification, a model-based approach, which endows engineering teams with a methodology and an adequate tooling is of a paramount importance. To design IMA architectures interconnected with TT-Ethernet, we have proposed an extension of the AADL language in previous works. In this paper, we present a simulation-based verification of our extension and show how it can be simulated using a discrete event simulation environment called DEVS Suite. The main advantage of this technique is to perform cycle-accurate simulation of the complex avionics systems, which cannot be undertaken by model checking techniques. The tool demonstration video is available at: http://youtu.be/hwgN-a-7rzw.**

I. Introduction

Avionic systems are safety-critical systems which should meet strict safety, reliability and performance requirements and where no deviation is admissible. To build such systems, the Integrated Modular Avionics (IMA) architecture is an alternative of federated architectures, where each function is designed and deployed to use its exclusive resources. The IMA architectures are distributed using a communication infrastructure, which should also be able to meet the same level of safety and performance requirements.

For this, Time-Triggered Ethernet permeates a system with robust fault detection characteristics across a wide range of fault hypotheses. This is rewarded with the schedule of TT-Ethernet, because all message transmissions must occur according to the schedule, therefore any deviations (e.g. transmission at the incorrect time) are easily detected.

The advantages of this infrastructure are numerous. The combination of IMA and TT-Ethernet enables the error isolation provided at the level of the modules through the partitioning and at the level of the network using different data traffics. Moreover, TT-Ethernet enable the safe integration of data traffics with different performance and reliability requirements.

The focus of this work is on the verification of avionic systems deployed on IMA architectures and inteconnected with TT-Ethernet. These systems are complex and very challenging. In order to control the complexity of such systems, we proposed in [7], a model-based approach for IMA architectures interconnected with TT-Ethernet, using AADL modeling language.

To verify our models, we chose a simulation-based approach, which is the widely-used technique to ensure the correctness of a system [1], [11]. This approach allows to design the behavior of a dynamic system by describing its reaction to external stimuli, such as the function of time [11].

In order to develop a simulation for our systems, one way is to move from an informal conceptual space to an abstract mathematical specifications [1]. The abstract mathematical specification is supported by a Discrete Event Formalism called (DEVS) [9]. This formalism can guarantee that the network is running according to the TT-Ethernet specifications by ensuring its scheduling properties. Toward this goal, we take advantages of model transformations, by transforming an AADL model which represents an IMA architectures interconnected with TT-Ethernet to a model conformant to DEVS formalism.

II. Background

The Integrated Modular Avionics (IMA) architecture is based on resource sharing between functionalities ensuring their isolation in order to prevent any interfaces between them [3]. Mainly two ARINC standards define IMA systems. The first standard is ARINC 653 [4], which focuses on a modular real-time architecture for avionics systems. Each functionality of the system is implemented by one or a set of functions distributed across different modules. A module represents a computing resource hosting many functions. Functions deployed on the same module may have different criticality levels. For safety reasons, the functions must be strictly isolated using partitions. The second standard supports the communication network in IMA. It is defined by either ARINC 664 [5] or the Time-Triggered Ethernet (TT-Ethernet) [6] depending on the need of predictability.

TT-Ethernet extends the Ethernet IEEE standard 802.3x, in order to support time-triggered services. Therefore, TT-Ethernet establishes a system-wide time base, implemented through a clock synchronisation of the end systems and switches. TT-Ethernet consists of three different types of traffic, which enable supporting distributed avionic applications with different performance and reliability requirements. Namely: Time-Triggered (TT) traffic, Rate-constrained (RC) traffic and Best-Effort (BE) traffic. Each of these traffics has specific characteristics, which are out of the scope of this paper.

AADL is a standard architecture description language (SAE Standard AS5506) [2]. Which provides an extensible core language with a precise semantics. AADL has been extended to support the modeling of IMA with an Annex for ARINC 653 [4]. This extension is one of the reason that we choose AADL as the modeling language in our work. We have presented in previous works an extension for AADL to support the modeling of IMA architectures interconnected using TT-Ethernet. This work supports the networking aspects of IMA architecture, where the conceptual elements ARINC 653 has been presented by ARINC 653 annex [4]. In particular, we present a metamodel for the domain of IMA and TT-Ehernet. We provide a concrete textual syntax based on this metamodel, which enables the system engineers to describe a full IMA-based systems using TT-Ethernet.

In order to simulate and analyse an AADL model, some tools have been proposed such as *cheddar* [17]. This tool cannot support the schedulability of TT-Ethernet nor the specification of specific behaviors of simulated systems. For this, we chose the DEVS formalism [12] that provides a rigorous common basis for discrete-event and continuous-time modeling and simulation [16]. It is presented as an extension to Finite State Automata. It allows describing the behavior of system in two levels, which are *atomic* DEVS and *coupled* DEVS. The behavior of a discrete-event system is described with *atomic* DEVS, which uses Finite State Automata to produce output event from the reaction to input event. A *coupled* DEVS represents overall system as a network of coupled components. These components can be *atomic* DEVS or *coupled* DEVS in their own right [12].

The metamodel of DEVS is built on top of Eclipse Modeling Framework (EMF) and DEVS formalism [1]. Knowing that our previous tooling set is also built on top of the Eclipse ecosystem [2], our motivation to chose DEVS was mainly for interoperability purposes.

DEVS metamodel is divided into structural and behavioral parts where both of them should be defined for atomic and coupled models. The metamodel of DEVS considers formal DEVS models as well as their simulation models developed for given simulator [1]. DEVS simulation environment is a simulation environment for hierarchical and parallel DEVS models [10].

III. Proposed Approach

Figure 1 illustrates the overall architecture of our framework. This approach takes advantage of model transformation where a set of target model is automatically generated from a set of source model.

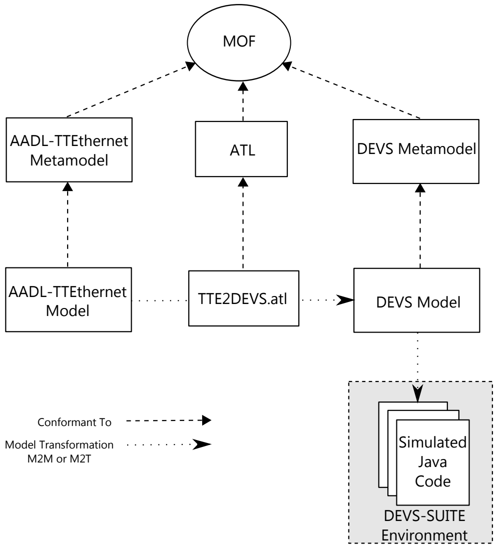


Fig. 1. Overall Architecture

In our case, the source model is an AADL model that represents IMA architecture interconnected with TT-Ethernet. This model is conformed to the metamodel of AADL-TTEthernet presented in our previous work [7]. The metamodel of IMA architecture interconnected by TT-Ethernet captures the main concepts and characteristics of the Time-Triggered Ethernet and describes the structural aspect of a distributed IMA architecture with it. It aims to explain all the concepts specified by this standard. The global information about the network elements and the underlying implementation is described in Figure 2.

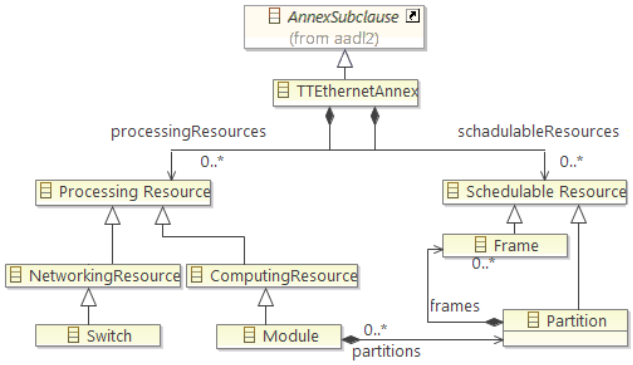


Fig. 2. Overview of AADL-TTEthernet metamodel

- *Schedulable resources* represents all the elements, which uses the network scheduler. These resources can be the partitions hosted by module, the data transferred through the network for example Frames.
  Partition is a group of time slices in a major frame (MAF) on a module. According to ARINC 653 standard [4], each function executes periodically within a partition where it is isolated from all others sharing the core module. Frame is a unit of transmission, a data packet of fixed or variable length, encoded for digital transmission over a communication link.

Considering its order of priority, a frame could be *Protocol Control Frame* (PCF), *Time Triggered (TT)* frame, *Rate Constraint (RC)* frame or *Best Effort (BE)* frame.

- *Processing Resources* represents active hardware components in a network. All processing resources have *features* that can be parameters, access to physical buses or ports (i.e. interfaces for frames inputs and outputs). A processing resource can be *Networking Resources* such as *Switches* or *Computing Resources* such as *Modules*.

The next step consists of transforming an AADL-TTEthernet model to a DEVS model according to some transformation rules written in the ATL language. The execution environment for ATL is provided by Eclipse framework [8]. The target model, provided by ATL transformation, is an intermediate model that can be used in the future to perform directly the model simulation realized by DEVS simulation environment.

The target metamodel is the metamodel of DEVS. The simplified version of this metamodel is described in Figure 3. *eAtomic* and *eCoupled* are the main classes of DEVS metamodel. The atomic and coupled models perform a simulation model. The simulation models implement abstract models. Every component in the system that includes a behavior is implemented as an atomic model where the Finite State automata of this specific behavior is defined. A coupled DEVS model contains all its sub-components, that can be either atomic or coupled. For the letter, it describes the way of coupling them together. For that, *eCoupling* classes of DEVS metamodel are used. They support the internal connection between the subcomponents of a coupled model as well as the connectivity of other coupled models. More details about this metamodel can be found in [1].

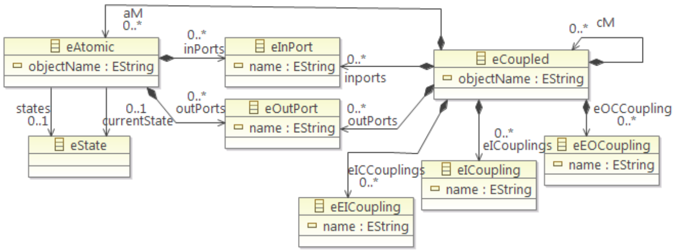


Fig. 3. DEVS metamodel

The DEVS model resulted from the transformation step represents an IMA architecture interconnected with TT-Ethernet using AADL that refined with DEVS model formalism. The behaviors of that model is added into its implementation to provide the simulatable model for DEVS simulation environment. We use Acceleo [13], to generate the java code from the DEVS model. Acceleo is a pragmatic implementation of the MOF Model to Text Language (MTL) standard [13]. Finally, the Java code corresponding to our DEVS model, is loaded and simulated by means of DEVS simulation environment. In the following section we present a case study that demonstrates the feasibility and efficiency of our approach as well as its implementation details.

IV. Simulation of the Navigation & Guidance System

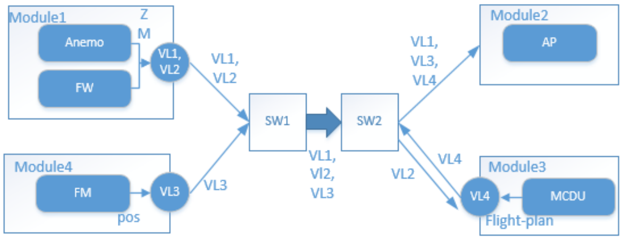
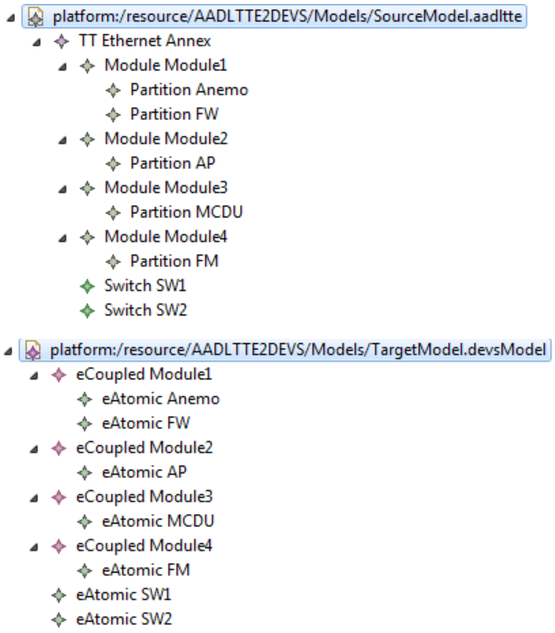


Fig. 4. The Navigation & Guidance system

This example is a simplified navigation and guidance system, which is one of the safety-critical function in IMA architecture [14]. Figure 4 depicts this example which is composed of four modules and two switches. The *Autopilot* (AP) module elaborates flight command to reach a defined attitude (the attitude is defined by the next way-point of the flight plan), the *Multifunction Control Display Unit* (*MCDU*) presents an interface between the system and crew. The *Flight Management* (*FM*) sends periodically to *AP* the next way-point*(pos)* to reach. The *Flight Warning* (*FW*) reports to *MCDU* the equipment status *(sens-stat)*. The last module which is *Anemometer* (*Anemo*) computes and broadcasts speed (M) and attitude (Z) to *AP*.

We assume that Z and M are two critical parameters in this example and encapsulated in TT frames. They are transmitted in two distinct frames, by VL_1 from *Anemo* which is the partition of module 1, to *AP* which is the only partition of module 4, via SW_1 and SW_2. The AADL-TTEthernet model and its corresponding DEVS model, are depicted in Figure 5.



- platform:/resource/AADLTTE2DEVS/Models/SourceModel.aadltte
  - TT Ethernet Annex
    - Module Module1
      - Partition Anemo
      - Partition FW
    - Module Module2
      - Partition AP
    - Module Module3
      - Partition MCDU
    - Module Module4
      - Partition FM
    - Switch SW1
    - Switch SW2
- platform:/resource/AADLTTE2DEVS/Models/TargetModel.devsModel
  - eCoupled Module1
    - eAtomic Anemo
    - eAtomic FW
  - eCoupled Module2
    - eAtomic AP
  - eCoupled Module3
    - eAtomic MCDU
  - eCoupled Module4
    - eAtomic FM
  - eAtomic SW1
  - eAtomic SW2

Fig. 5. TT-Ethernet Model (top) and the corresponding DEVS model (bottom)

In the next step we produce the corresponding Java code of the target model using Acceleo [13]. Then, we add the behavior of IMA architecture interconnected with TT-Ethernet to the generated Java code. Finally, we load it as a model into DEVS simulation environment. Figure 6 demonstrates the simulated model of navigation and guidance example.

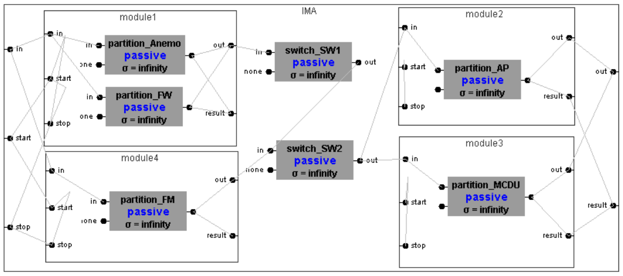


Fig. 6. Simulation graph for the navigation & guidance system

For the verification purpose, we can define different simulation scenarios based on the system requirements. In our scenarios we are verifying the scheduling properties that, if held true, will guarantee that the network is running according to the TT-Ethernet specifications. The scheduler of TT-Ethernet request specific constraints and properties, which are presented in [15]. One of those constraint is the *contention-freedom*. We assume that the scheduler is produced by the system and we verify it to ensure the satisfaction of this constraint.

To guarantee the *contention-freedom*, an End-System (ES) dispatches a frame only after the previous frame completely delivered by the receiver ES. For that, we create *Job1* and *Job2* to check that if *job2* is dispatched only after the reception of *Job1* by its corresponding destination. Figure 7 demonstrates the output of our case study. As it can be captured from Figure 7, in 10.0, Module1 relays on *Job1* that is received by the output of IMA at 40.0. After this reception, module1 in the same time (40.0) relays on *Job2* that is received by the output of IMA in 70.0. That means, even that module1 has two Jobs in its schedule to dispatch, but it relays on second one only after reception of first one. As it is requested by contention-freedom constraint.

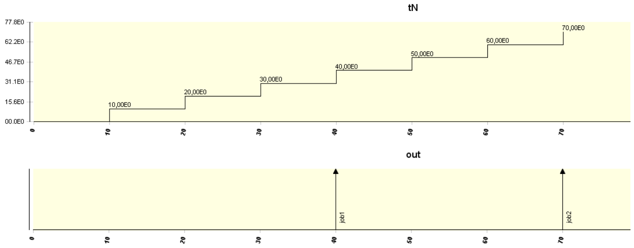


Fig. 7. Simulation results

V. Conclusion

In this demonstration, we have presented an AADL simulation tool using DEVS simulation environment. Our technique has been implemented and successfully tested on an example of IMA architecture interconnected with TT-Ethernet. It is enough mature to be tested on others IMA example, regardless the complexity of example.

The main motivation of this demonstration is to provide a methodology and a tooling set to assist system engineers to verify the designed system. Accordingly, we provide a modeling language that allows to express the system at a convenient level of abstraction and to interface with sophisticated simulation environment to verify the safety and the performance of the system. Practically, engineers can represent the intended system with an AADL model using our extension and then map it to the discrete event formalism with DEVS. This mapping conform to the corresponding metamodels in both sides. The behavior of the system is added to the mapped model in order to assemble the simulation framework of DEVS simulation environment. The simulation results is used to verify the system requirements.

In future work, we aim at providing other verification scenarios for our topic, in order to guarantee the communication network of IMA architecture is running according to the TT-Ethernet specifications.

References

[1] Hessam S. Sarjoughian and Abbas Mahmoodi Markid. 2012. *EMF-DEVS modeling*. In Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium (TMS/DEVS '12). Society for Computer Simulation International, San Diego, CA, USA, , Article 19 , 8 pages.

[2] Open source aadl tool environment (osatev2). http://www.aadl.info, 2015.

[3] Christopher B Watkins and Randy Walter. Transitioning from federated avionics architectures to integrated modular avionics. In *Digital Avionics Systems Conference, 2007. DASC'07. IEEE/AIAA 26th*, pages 2–A. IEEE, 2007.

[4] Aeronautical Radio Incorporated. *ARINC Report 653P0 Avionics Application Software Standard Interface, Part 0, Overview of ARINC 653*, 2013.

[5] Aeronautical Radio Incorporated. *ARINC Report 664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network*. AEEC, Maryland, USA, 2009.

[6] SAE Aerospace. *Time-Triggered Ethernet*, sae as6802 edition, 2011.

[7] Tiyam Robati, Amine El Kouhen, Abdelouahed Gherbi, Sardaouna Hamadou, John Mullins. An Extension for AADL to Model Mixed-Criticality Avionic Systems Deployed on IMA architectures with TTEthernet. ACVI at MoDELS, 2014.

[8] F. Jouault, F. Allilaire, J. Bzivin and I. Kurtev, ATL: A model transformation tool, Science of Computer Programming, Elsevier, vol. 72, no. 12, (2008), pp. 31-39.

[9] Bernard P. Zeigler, Herbert Praehofer, and Tag Gon Kim. Theory of Modeling and Simulation. Academic Press, 2000.

[10] DEVS-Suite Simulator. http://devs-suitesim.sf.net, February 2009.

[11] Generation of DEVS Modeling and Simulation Environment. Ernesto Posse, Jean-Sbastien Bolduc, Hans Vangheluwe, 2003/7.

[12] Bernard P. Zeigler. Multifacetted Modelling and Discrete Event Simulation. Academic Press, London, 1984.

[13] Acceleo. http://www.eclipse.org/acceleo.

[14] Michael Lauer, Jerme Ermont, Claire Pagetti, and Frederic Boniol. 2010. Analyzing end-to-end functional delays on an IMA platform. In Proceedings of the 4th international conference on Leveraging applications of formal methods, verification, and validation - Volume Part I (ISoLA'10), Tiziana Margaria and Bernhard Steffen (Eds.), Vol. Part I. Springer-Verlag, Berlin, Heidelberg, 243-257.

[15] W.Steiner. An evaluation of smt-based schedule synthesis for time-triggered multi-hop networks. volume vol., no., pp.375,384. Real-Time Systems Symposium (RTSS), IEEE 31st, 2010.

[16] Richard E. Nance. The time and state relationships in simulation modeling. Communications of the ACM, 24(4):173179, April 1981.

[17] http://beru.univ-brest.fr/ singhoff/cheddar/

[18] SAE Aerospace. *SAE Architecture Analysis and Design Language (AADL) Annex Volume 2: Annex B: Data Modeling Annex Annex D: Behavior Model Annex Annex F: ARINC653 Annex, AS5506/2*, 2011.