HAL
open science

An Extension for AADL to Model Mixed-criticality Avionic Systems Deployed on IMA architectures with TTEthernet

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

To cite this version:

Tiyam Robati, Amine El Kouhen, Abdelouahed Gherbi, Hamadou Sardadouna, John Mullins. An Extension for AADL to Model Mixed-criticality Avionic Systems Deployed on IMA architectures with TTEthernet. 1st Workshop on Architecture Centric Virtual Integration @ the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS 2014), Carnegie Mellon University - Software Engineering Institute, Sep 2014, Valencia, Spain. pp.14-27. hal-01054881

HAL Id: hal-01054881

https://inria.hal.science/hal-01054881v1

Submitted on 9 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Extension for AADL to Model Mixed-criticality Avionic Systems Deployed on IMA architectures with TTEthernet

Tiyam Robati[1], Amine El Kouhen[1], Abdelouahed Gherbi[1], Sardaouna Hamadou[2], and John Mullins[2]

[1] Dept. of Software and IT Engineering, École de Technologie Suprieure, Canada
{tiyam.robati.1@ens.etsmtl.ca, amine.elkouhen@etsmtl.ca, abdelouahed.gherbi@etsmtl.ca}
[2] Dept. of Computer and Software Eng, Ecole Polytechnique de Montreal, Canada
{firstname.lastname@polymtl.ca}

Abstract. Integrated modular avionics architectures combined with the emerging SAE TTEthernet standard provides a strong infrastructure for the deployment of mixed-critical avionic applications having stringent safety, reliability and performance requirements. The integration of such systems is a very complex and challenging engineering task. Therefore, a model-based approach, which endows system engineers with a methodology and the supporting tools to cope with this complexity, is of a paramount importance. In this research paper, we present an extension for the standard architecture and analysis modeling language AADL to enable modeling integrated multi-critical avionic applications deployed on TTEthernet-based IMA architectures. In particular, we present a metamodel which extends the core AADL metamodel with concepts and constraints relevant for this domain, we define the concrete textual syntax for this extension and we outline the implementation of this extension using the Open Source AADL Tool Environment (OSATE). Finally, we illustrate our AADL extension using a case study based on the Flight Management System.

Keywords: AADL, Time-Triggered Ethernet, AFDX, IMA

1 Introduction

On-board avionic systems are safety-critical systems which should meet strict safety, reliability and performance requirements. These systems have traditionally been engineered using what is called a federated architectures approach, where each function is designed and deployed to use its exclusive resources. This approach is however costly in terms of equipments and wiring. The Integrated Modular Avionics (IMA) architecture is an alternative approach, which is based a consolidation of resources [21]. This is achieved through resources sharing between functionalities. With IMA different avionic functions having different criticality levels (e.g. control functions and comfort functions) share the same

hardware resources leading to mixed-criticality systems. Moreover, IMA architectures are distributed using a communication infrastructure, which should also be able to meet the same level of safety and performance requirements.

Ethernet is a widely used standard network (IEEE 802.3) which is not only used as infrastructure for classic office systems but is increasingly supporting industrial and embedded systems due to the high bandwidths it provides. However, Ethernet does not meet strict time and safety critical applications. Several extensions to enhance the predictability of Ethernet have been developed. One of these extensions is the Avionic Full Duplex AFDX standard ARINC 664 [11]. AFDX is a deterministic real-time extension of Ethernet based on an static bandwidth scheduling and control using the concept of virtual links. The SAE standard TTEthernet [3] is the most recent Ethernet extension based on the time-triggered communication paradigm [14] [19] to achieve bounded latency and low jitter. A TTEthernet network implements a global time using clock synchronisation and offers fault isolation mechanisms to manage channel and nodes failures. TTEthernet integrates three data flow: Time-Triggered (TT) data flow which is the higher priority traffic; Rate Constrained (RC) traffic, which is equivalent to AFDX traffic, and Best Effort (BE) traffic. This makes TTEthernet suitable for mixed-criticality applications such as avionic and automotive applications where highly critical control functions such as a flight management system cohabit with less critical functions such as an entertainment system.

The focus of this research work is on avionic applications deployed on IMA architectures interconnected using TTEthernet. The advantages of this infrastructure are numerous. First, the IMA modules enable the resource sharing. Second, the combination of IMA and TTEthernet enables the error isolation provided not only at the level of the modules through the partitioning but also the level of the network using different data traffics and the concept of virtual links. Third, TTEthernet enable the safe integration of data traffics with different performance and reliability requirements. However, these systems are on the other hand complex and the integration of diverse applications with mixed-criticality levels having strict real-time requirements is very challenging. In order to control the complexity of such systems, a model-based approach, which provides the systems engineers with a methodology and the supporting tools to accomplish correctly and efficiently this integration, is required. A key element of such approach is a modeling language which allows the engineers to express the system at a convenient level of abstraction and to interface with sophisticated formal analysis techniques to verify safety and performance properties of the system.

AADL is a well-established standard modeling language in the domain of real-time critical systems. AADL has been extended to support the modeling of IMA with an Annex ARINC 653 [2]. However, there is no support for AADL to model the networking of IMA modules through the recent technology TTEthernet. We present in this paper an extension for AADL to support the modeling of IMA architectures interconnected using TTEthernet. In particular, we present a metamodel for the domain of IMA and TTEhernet. We provide a concrete

textual syntax based on this metamodel, which enables the system engineers to describe a full IMA-based avionic systems interconnected with TTEthernet. We have implemented this extension in the framework of the Open Source AADL Tools (OSATE2)[5] . We illustrate the expressiveness of this extension through it application to model a subsystem of the the Flight Management System [18].

This paper is organized as follows: In Section 2, we introduce the concepts of IMA and the main features of the TTEthernet standard. We describe in Section 3 the main components of the proposed extension metamodel and discuss the rational behind its design. We outline the implementation of the proposed extension in the framework of OSATE in Section 4. We show the application of the proposed extension with an illustrative example in Section 5. In Section 6, we succinctly review the most close related research works to ours. We conclude the paper and outline our ongoing and future research work in Section 7

2 Background

In order to make this paper as self-contained as possible, we briefly introduce in this section the main concept of IMA and TTEthernet.

2.1 Integrated Modular Avionic Architecture (IMA)

The main idea underlying the concept of IMA architecture [21] is the sharing of resources between some functions while ensuring their isolation to prevent any interference between them. Resource sharing reduces the cost of large volume of wiring and equipment while the non interference guarantee is required for safety reasons. The IMA architecture is a modular real-time architecture for avionics systems defined in the standard ARINC653 [12]. Each functionality of the system is implemented by one or a set of functions distributed across different the 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 partitioning of these functions is two dimensional: spatial partitioning and temporal partitioning. The spatial partitioning is implemented by assigning statically all the resources for the partition being executed in a module and no other partition can have the access to the same resources at the same time. The temporal partitioning is rather implemented by allocating a periodic time window dedicated for the execution of each partition.

2.2 Time-Triggered Ethernet (TTEthernet)

The new SAE Time-Triggered Ethernet standard (TTEthernet) [3] specifies time-triggered services extending the Ethernet IEEE standard 802.3. TTEthernet is based on the Time-triggered communication paradigm [13] and therefore establishes a system-wide time base implemented through a synchronisation of the clocks of the end systems and switches. This results in bounded latency and

low jitter. TTEthernet integrates both time-triggered and event-triggered communication on the same physical network. TTEthernet limits latency and jitter for time-triggered (TT) traffic, limits latency for rate constrained (RC) traffic, while simultaneously supporting the best-effort (BE) traffic service of IEEE 802.3 Ethernet. This allows application of Ethernet as a unified networking infrastructure. It supports therefore the deployment of mixed-criticality applications at the network level.

3 Metamodel Extending AADL capability to model TTEthernet

In this section, we present the metamodel for our extention to AADL in order to support the modeling of TTEthernet, which will henceforth be called the AADL-TTEthernet metamodel. This meta-model captures the main concepts and characteristics of the TTEthernet standard. The AADL-TTEthernet metamodel will enable building a set of tools to perform the design and analysis of distributed IMA architectures using TTEthernet as communication infrastructure. We have designed the AADL-TTEthernet metamodel using the Eclipse Modeling Framework (EMF), which is also used to specify the AADL Core metamodel. This allows for a seamless integration of the AADL-TTEthernet in OSATE2 environment [5] in terms of dependencies and embedded Java API. On the other hand, using the same formalism (i.e. Ecore) to specify the two metamodels eases the expression of the domain concepts dependencies and simplifies the navigation between them.

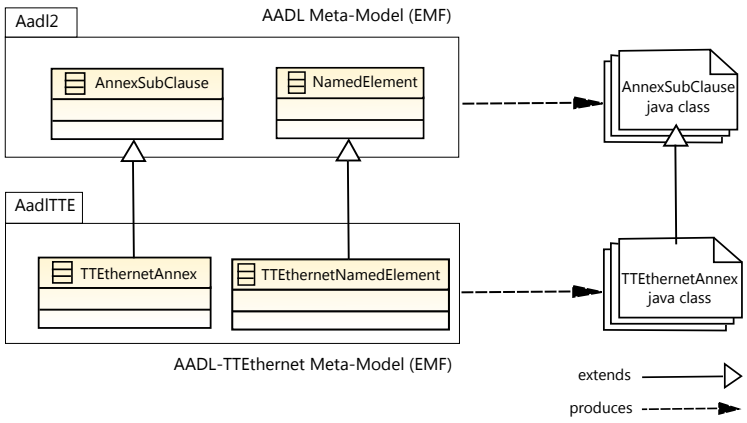

Fig. 1. AADL-TTEthernet meta-model dependencies

Our AADL-TTEthernet metamodel describes the structural aspect of a distributed IMA systems interconnected using TTEthernet and makes explicit all

concepts specified by this standard. The EMF framework generates automatically the Java implementation classes corresponding to the metamodel objects as it is shown in Figure 1. In order to extend AADL with our metamodel it is required to attach a TTEthernet model to an AADL component and to link the objects of our TTEthernet extension with AADL core objects. This is achieved by the implementation of the OSATE2 extension mechanism, which requires to link the TTEthernetAnnex concept in our metamodel to the AnnexSubclause concept of the AADL core as it is shown in Figure 1. This figure shows also how we use the EMF/Ecore inheritance mechanism to express the dependencies between the two metamodels. Consequently, a *TTEthernetAnnex* extends an *AnnexSubclause* and an *TTEthernetNamedElement* extends a *NamedElement*.

In the metamodel, the TTEthernetAnnex concept, which links as shown in Figure 2 the metamodel to the AADL core metamodel, represents the overall model of a TTEhernet-networked IMA system which will undergoes different analysis to verify safety and performance properties. The global information about the network elements and the underlying implementation is described in the *TTEthernetAnnex* concept.

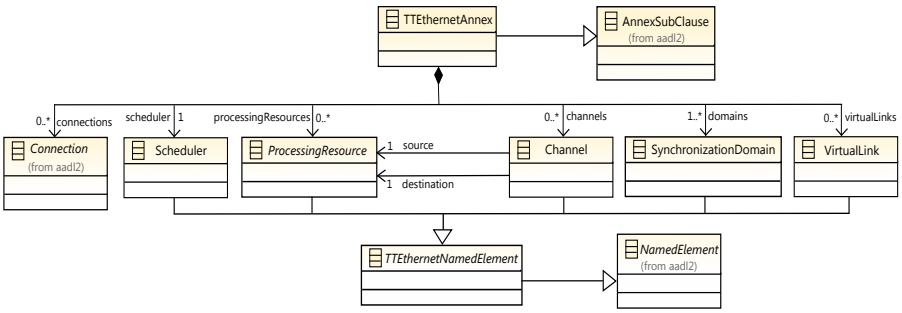

Fig. 2. AADL-TTEthernet meta-model overview

The *TTEthernetAnnex* is composed of the following (as depicted in Figure 2):

1. The *Synchronization domains* concept of TTEthernet standard. TTEthernet supports system-of-system communication by introducing *Synchronization domains* and *Synchronization priorities* as shown in Figure 3. *Synchronization domains* specify independent subsystems with respect to their synchronization. All the resources configured to belong to the same synchronization domain should synchronize with each other and components belonging to different synchronization domains in one TTEthernet network do not synchronize their local clocks.
2. The *Scheduler* is the entity in TTEthernet that is capable of producing a schedule, which should be compliant with the scheduling constraints of TTEthernet. These constraints are depicted in figure 3.

3. The *Processing Resources* represent active hardware components in a network. They can be *Computing Resources* such as *Modules* (i.e. end systems) or *Networking Resources* such as switches as shown in Figure 4. All processing resources have *features* which can be parameters, access to physical buses, or ports (i.e, interfaces for frames inputs and outputs). A processing resource can be a synchronization master and can then transmits its local time to synchronize the whole network as shown in Figure 3. Several processing resources can be aggregated into logical groups called clusters as shown in Figure 4. A *Cluster* is associated with one synchronization domain. Each single cluster can establish and maintain synchronization by itself.

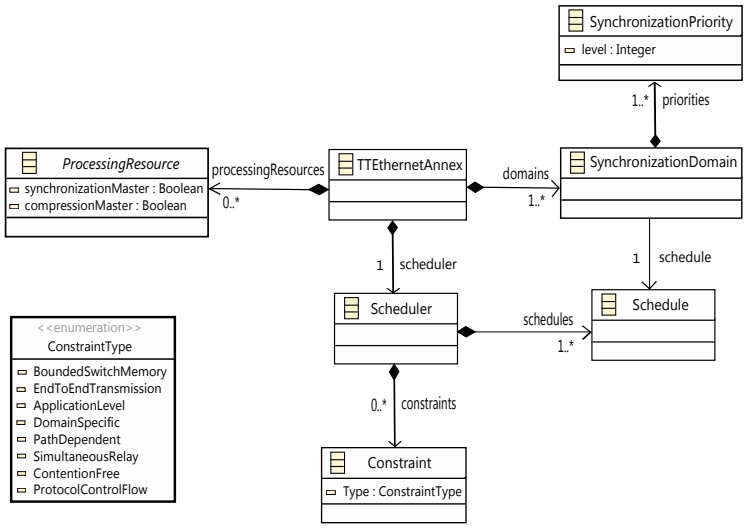

Fig. 3. Synchronization elements

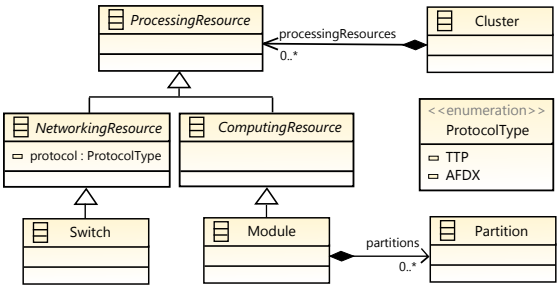

Fig. 4. Processing resources

The AADL-TTEthernet metamodel captures the different possible links between the components of a TTEthernet Network. These links can be either physical ones such as connections or logical ones such as channels or virtual links.

- A *Connection* is a link between two physical ports, usually realized as a copper or optical fiber cable. A connection may be unidirectional or bidirectional.
- A *Channel* is a logical connection from one source processing resource to another processing resource destination. A Channel is defined to map multi-cluster architectures.
- A *Virtual Link* is a logical link defined by ARINC 664 standard [11]. Each virtual link is associated with a dedicated maximum bandwidth, specified by the minimum frame interval, called bandwidth allocation gap, and the maximum frame length.

The *Schedulable Resources* represents all the elements which are managed using the network scheduler. These resources can be the partitions hosted by module, the data transferred through the network (i.e, Frames), the networks communication channels or the virtual links as shown in Figure 5.

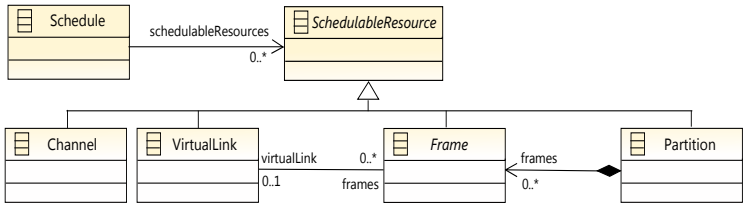

Fig. 5. Schedulable resources

A *Frame* is unit of transmission, a data packet of fixed or variable length, encoded for digital transmission over a communication link as depicted in Figure 6. Considering its order of priority, a frame could be *Protocol Control Frame* (PCF), *TT* frame, *Rc* frame or *BE* frame.

4 Implementation of the TTEthernet Extension for AADL

Textual Syntax for the TTEthernet Extention for AADL The definition of a textual syntax is provided by a grammar (i.e, a set of rules which define the composition of a language). In order to translate the textual syntax to its corresponding model, a lexer, a parser as well as a component for the semantical analysis (type checking, resolving of references, etc.) are required. The backward

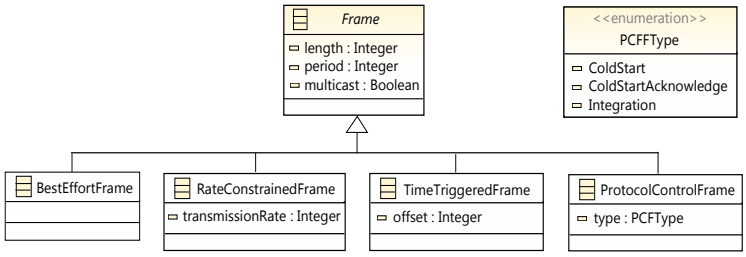

Fig. 6. Frame kinds

transformation, from model to text, is provided by an emitter. All the components can be generated using the grammar $\Longleftrightarrow$ meta-model mapping definition [10]. Figure 7 demonstrates the selected framework to define textual syntax of our extension. It employs the data provided by the mapping definition used to generate the parser, emitter and an editor for the corresponding language to the metamodel. This editor can then use the generated parser and emitter to modify the text and the model. Therefore it is responsible for keeping the text and the model in sync, e.g., by calling the parser upon any changes on the text. Based on this mapping definition, several features of the editor can be generated, such as syntax highlighing, autocompletion or error reporting. To build

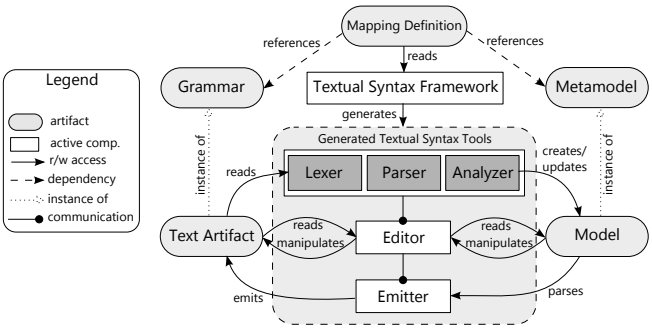

Fig. 7. General structure of a textual syntax framework

the textual editor tool for our AADL-TTEthernet extension, we used the xText framework [8]. It implements the textual syntax according to an extended BNF. Figure 8 shows an excerpt of this xText grammar. In this xText framework, the AADL-TTEthernet metamodel concept is mapped to a Java implementation, where the TTEthernet objects names are used as class names. All attributes are implemented as private fields and public get- and set- methods. The composition relationships are realized in the same way as attributes and contribute to the constructor of the class. All classes support the Visitor pattern [9] to traverse the abstract syntax along the composition relationships [15].

The analyzer module scans the Abstract Syntax Tree (AST) and checks the semantics of the AADL-TTEthernet model. First, it proceeds to a resolution phase (e.g, naming resolver), which links TTEthernet objects to their corresponding AADL objects. In order to achieve this phase, we use the visitors (e.g, java classes) provided by OSATE2 to retrieve AADL objects. For the sake of the implementation of our AADL-TTEthernet extension, we have developed the visitors required to navigate through the AADL-TTEthernet AST. This phase adds information to the AST and makes its use easier.

```
1  grammar org.osate.ttethernet.xtext.Aadltte
2
3  import "http://ca.estmtl.aadl2/aadltte/1.0"
4  import "http://aadl.info/AADL/2.0" as aadl2
5  import "http://www.eclipse.org/emf/2002/Ecore" as ecore
6
7  Partition returns Partition:
8    'Partition' name = ID
9        'frames' ':'frames += Frame*
10   'end' ID ';';
11
12 Frame:
13   RateConstrainedFrame | TimeTriggeredFrame | BestEffortFrame
       | ProtocolControlFrame
14 ;
15
16 SynchronizationPriority returns SynchronizationPriority:
17   'Synchronization Priority' name = ID
18     'level' level = Integer ';'
19   'end' ID ';';
```

Fig. 8. xText grammar overview for AADL-TTEthernet

Integration of the AADL-TTEthernet Compiler to OSATE2 Sublanguages are included into AADL specifications as annex subclauses. The latter may be inserted into AADL component types and AADL component implementations of an AADL model. OSATE2 currently provides four extension points that can be used to integrate a sublanguage into the tool environment. These extension points are designed to support parsing, unparsing, name resolution / semantic checking, and instantiation of annex models. From the AADL-TTEthernet EMF meta-model in the EMF framework, we generate the AADL-TTEthernet builder factory to build and manipulate TTEthernet objects used in the compiler. The compiler plug-in contains two modules: a parser/lexer and an analyzer. The integration of the AADL-TTEthernet plug-in is a two-steps process. First, we link the AADL-TTEthernet plug-in to the OSATE2 annex plug-in

using the Eclipse extension points mechanism. The annex plug-in defines extension points which allow to plug-ins be connected together as depicted in Figure 9. Second, we have to register our parser in the OSATE2 annex registry. As the AADL-TTEthernet metamodel becomes a part of the AADL description and the AADL-TTEthernet textual syntax tool is connected to OSATE2 registry, the AADL-TTEthernet plug-in is directly integrated and driven by OSATE2.

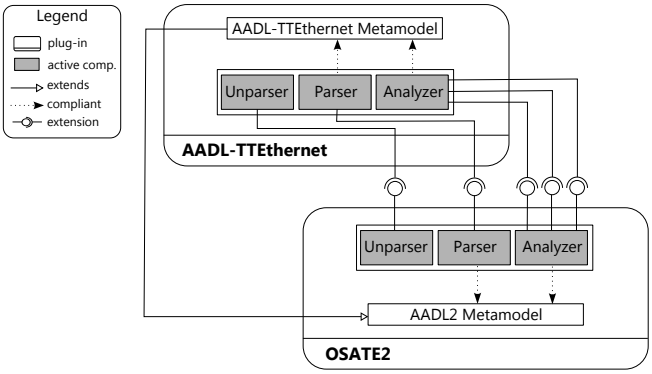

Fig. 9. AADL-TTEthernet plug-in integrated to OSATE2

5 An Example: A Model of a Subsystem of the Flight Management System

In this section, we illustrate the modeling of a distributed IMA system based on TTEthernet as a communication network using our extension for AADL. In order to do so, we use as a subsystem of the Flight Management System presented in [18]. This subsystem controls the display of static navigation information in the cockpit screens. The structure of the considered FMS subsystem in terms of modules and the partitions they host is shown in Figure 10. In the original version of the system considered in [18], the system is interconnected using AFDX. In our context, the modules are instead interconnected using TTEthernet. The AFDX data traffic in the original system corresponds to the RC traffic in the TTEthernet context. Table 1 shows a subset of the virtual links used in the FMS subsystem with their corresponding characteristics including the Bandwidth Allocation Gap (BAG), the sender modules of the VLs and the corresponding receiver modules.

This subsystem can be modeled using our TTEthernet extension for AADL as follows. The extension is a sub-language for AADL, which can be included in *system implementation* of the AADL model of this system. The concrete textual syntax of AADL-TTEthernet extension provides several new reserved words, which correspond to the main concepts of the metamodel described previously,

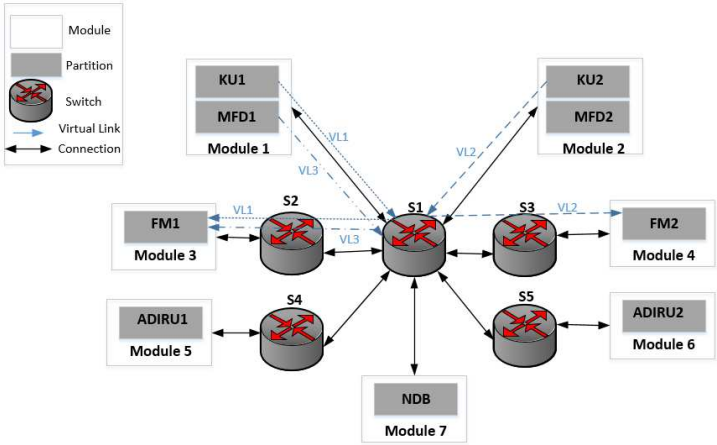

Fig. 10. A Subsystem of the Flight Management System

Virtual Link	Source	Destination	BAG	Direction
VL_1	KU_1	FM_1, FM_2	32	$\{S_1, S_2\}$, $\{S_1, S_3\}$
VL_2	KU_2	FM_1, FM_2	32	$\{S_1, S_2\}$, $\{S_1, S_3\}$
VL_3	FM_1	MFD_1	8	$\{S_2, S_1\}$

Table 1. Virtual Links details

such as module, switch, partition, connection, virtual link, Time-triggered frame, Rate Constraint frame and Best Effort frame. An excerpt of the model of the FMS subsystem using our AADL-TTEthernet is shown in Figure 11. The full specification of the model can be downloaded at [20].

6 Related Work

AADL presents two extension mechanisms, namely the property sets and the sublangues (i.e. annex). Several AADL extensions based on these mechanisms are now standardized as official annexes. These include the Data modeling annex, ARINC653 annex, the AADL Behavor Annex [2] , and Error Model Annex [1]. In addition, some research work have focused on extending the language using these extension mechanisms or investigating alternative ways. The most close research works to ours are reported in [7] and [17] . J. Delange et al. [7] present an approach based on AADL, which covers the the modeling, verification and implementation of ARINC653 systems. The authors describe in the work the modeling guidelines elaborated in the ARINC653 annex of the AADL standard. This approach is supported by a tool chain composed of Ocarina AADL toolsuite, AADL/ARINC653 runtime POK and Cheddar scheduling tool. G Lasnier et al. [17] present an implementation of the AADL behavior annex as an extension plug-in to the OSATE 2. We have implemented our AADL TTEthernet extension

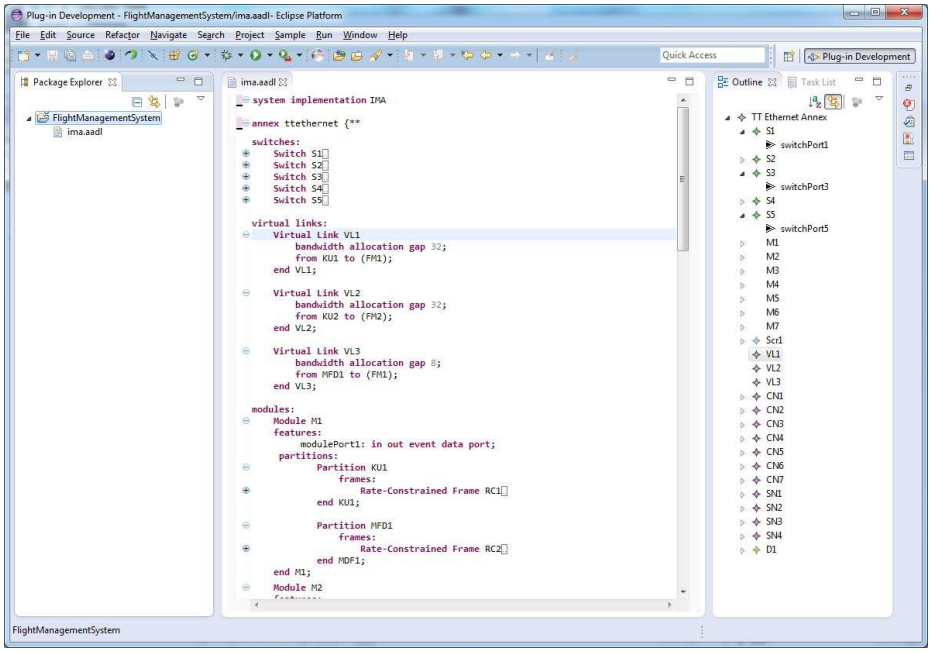

Fig. 11. Flight Management Subsystem Model using AADL TTEthernet Extension

using similar techniques. M. lafaye et al. [16] define a modeling approach based on AADL and SystemC, which aims at the design and dynamic simulation of a IMA-based avionics platform. This is component-based approach, which can be used to dimension the architecture taking into consideration the application to be deployed while achieving early platform validation. De Niz and Fieler discuss in [6] how to extend the AADL language to include new features for the separation of concerns (i.e. Aspects). Based on this research work, it seems that the AADL extension mechanisms do not support the separation of concerns and new aspect-like constructs and mechanisms are then investigated. G. Brau et al. present in [4] a model of a subsystem of Flight Management System using AADL and show how to establish important parameters in the AADL model including the virtual links characteristics for instance. To the best of our knowledge, there is no published research work, which addresses the modeling of the TTEthernet standard as networking infrastructure for IMA architecture, which is the contribution of this work.

7 Conclusions and Future Work

The IMA architecture combined with the new SAE standard TTEthernet as communication infrastructure provide a strong platform for the deployment of distributed avionic applications. The integration of mixed-criticality applications

on such platforms is a complex and challenging engineering activity. A model-based approach based on the SAE standard architecture language AADL provides the system engineers with the tools to cope with this complexity. Modeling TTEthernet infrastructure using AADL is the research gap that we are targeting in this work. We have presented in this paper our contribution to address this issue, which consists in an extension for the standard architecture and analysis modeling language AADL to enable modeling integrated mixed-criticality avionic applications deployed on TTEthernet-based IMA architectures. In our ongoing research work, we aim at formalizing this extension in the form of a new annex through the SAE standardization process. Moreover, we aim at defining a formal semantics for our extension to allow transforming the AADL models built using our extension to models that are suitable for analysis techniques that can be used to verify relevant safety and performance properties.

References

1. SAE Aerospace. *SAE Architecture Analysis and Design Language (AADL) Annex Volume 1: Annex A: Graphical AADL Notation, Annex C: AADL Meta-Model and Interchange Formats, Annex D: Language Compliance and Application Program Interface Annex E: Error Model Annex,AS5506/1*, 2011.
2. 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.
3. SAE Aerospace. *Time-Triggered Ethernet*, sae as6802 edition, 2011.
4. Guillaume Brau, Jérôme Hugues, and Nicolas Navet. Refinement of aadl models using early-stage analysis methods: An avionics example. Technical Report TR-LASSY-13-06, Laboratory for Advanced Software Systems, 2013.
5. CMU/SEI. Open source aadl tool environment (osatev2). http://www.aadl.info, 2014.
6. Dionisio De Niz and Peter H Feiler. Aspects in the industry standard aadl. In *Proceedings of the 10th international workshop on Aspect-oriented modeling*, pages 15–20. ACM, 2007.
7. Julien Delange, Laurent Pautet, Alain Plantec, Mickaël Kerboeuf, Frank Singhoff, and Fabrice Kordon. Validate, simulate, and implement arinc653 systems using the aadl. In *ACM SIGAda International Conference on Ada*, pages 31–44. ACM, 2009.
8. S. Efftinge. *Xtext reference documentation.* http://www.eclipse.org/gmt/oaw/doc/4.1/r80xtextReference.pdf, 2006.
9. Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-oriented Software.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1995.
10. Thomas Goldschmidt, Steffen Becker, and Axel Uhl. Classification of concrete textual syntax mapping approaches. In Ina Schieferdecker and Alan Hartman, editors, *Model Driven Architecture - Foundations and Applications*, volume 5095 of *Lecture Notes in Computer Science*, pages 169–184. Springer Berlin Heidelberg.
11. Aeronautical Radio Incorporated. *ARINC Report 664P7-1 Aircraft Data Network, Part 7, Avionics Full-Duplex Switched Ethernet Network.* AEEC, Maryland, USA, 2009.

12. Aeronautical Radio Incorporated. *ARINC Report 653P0 Avionics Application Software Standard Interface, Part 0, Overview of ARINC 653*, 2013.
13. G. Kopetz, Hermann; Bauer. The time-triggered architecture. volume vol.91, no.1, pp.112,126. Proceedings of the IEEE, 2003.
14. Hermann Kopetz and Günther Bauer. The time-triggered architecture. *Proceedings of the IEEE*, 91(1):112–126, 2003.
15. Holger Krahn, Bernhard Rumpe, and Steven Vlkel. Integrated definition of abstract and concrete syntax for textual languages. In Gregor Engels, Bill Opdyke, DouglasC. Schmidt, and Frank Weil, editors, *Model Driven Engineering Languages and Systems*, volume 4735 of *Lecture Notes in Computer Science*, pages 286–300. Springer Berlin Heidelberg.
16. Michaël Lafaye, David Faura, Marc Gatti, and Laurent Pautet. A new modeling approach for ima platform early validation. In *Proceedings of the 7th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software*, pages 17–20. ACM, 2010.
17. Gilles Lasnier, Laurent Pautet, Jérôme Hugues, and Lutz Wrage. An implementation of the behavior annex in the aadl-toolset osate2. In *IEEE ICECCS*, pages 332–337. IEEE Computer Society, 2011.
18. M.Lauer. Une méthode globale pour la vérification d'exigences temps réel-application á l'avionique modular intgrée, 2012. Thse de Doctorat, Institut National Polytechnique de Toulouse.
19. Roman Obermaisser. *Event-triggered and time-triggered control paradigms*, volume 22. Springer, 2004.
20. Tiyam Robati, Amine El Kouhen, and Abdelouahed Gherbi. Flight management subsystem model using aadl ttethernet extension. http://profs.etsmtl.ca/agherbi/ima.aadl, 2014.
21. 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.