The MIMOSA Frame of Reference for the Certification of Military Embedded Modular Architectures

Virginie Wiels[1], Pierre Bieber[1], Frédéric Boniol[1], Rémi Delmas[1], David Doose[1], Guy Durrieu[1], Olivier Poitou[1], Thomas Polacsek[1], Vincent Louis[2], Florian Many[2], Ghilaine Martinez[2]

1. ONERA/DTIM, 2 avenue Edouard Belin, BP 74025, 31055 Toulouse cedex

Firstname.Lastname@onera.fr

2. DGA/TA, Division Systèmes Informatiques Embarqués 47 rue St Jean - BP 23 - 31131 Balma Cédex
FRANCE

Firstname.Lastname@dga.defense.gouv.fr

ABSTRACT

This paper presents the current results of the MIMOSA project, collaboration between DGA-TA and ONERA on the subject of embedded modular architectures. This project is building a model-based frame of reference for the certification of military embedded modular architectures.

1.0 INTRODUCTION

Embedded modular architectures are progressively appearing inside new aircrafts such as A400M but also for avionics renovations. In civil aeronautics, Integrated Modular Avionics (IMA) are used by Airbus (A380 and future A350), Thales Avionics (Sukhoï RRJ100 and ATR72) and will be used by Eurocopter for future generation of helicopters. Embedded modular architectures represent a significant technological gap. The complexity of these architectures necessitates adapted methods and tools for their design, development, and certification.

This paper presents the current results of the MIMOSA project, collaboration between DGA-TA and ONERA on the subject of embedded modular architectures. This project is building a model-based frame of reference for the certification of military embedded modular architectures. This frame is composed of three models:

- A central data model defining the main concepts of embedded modular architectures and their relationships;
- A requirement model focusing on two specific viewpoints: safety and real-time;
- An argumentation model linking each requirement to the associated means of compliance.

The goal of this frame of reference is not to design and develop embedded modular architectures, but rather

- To precisely formalize requirements (essential properties) of this kind of architectures (requirements part of the frame of reference) and
- To define in front of these requirements the acceptable means of compliance (including model-based analyzes) and the way they are composed (argumentation part of the frame).

This frame of reference could be used by DGA to assess architectures proposed by industrial companies with respect to certification objectives.

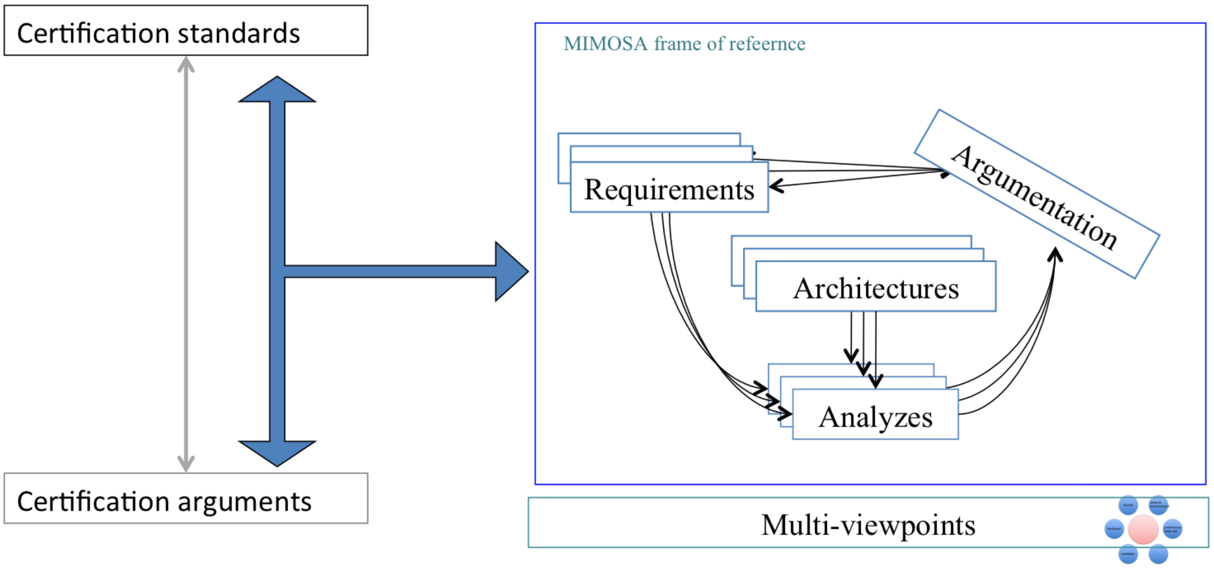

Figure 1: MIMOSA Frame of reference and its use to assess certification arguments

This paper is organized as follows. Section 2 describes the current status of the architecture model. Section 3 presents the requirement model. Section 4 discusses existing work that could be used for the argumentation model. Section 5 concludes the paper by describing work in progress inside the MIMOSA project.

2.0 ARCHITECTURE MODEL

The goal of the architecture model is to represent the fundamentals concepts of an IMA architecture and to support the expression of requirements attached to such a platform. We first analyzed two existing visions of an IMA architecture:

- An industrial version defined in the Scarlett project [2]
- A version extracted from DO-297/ED-124 [1]

These two visions are close to each other, a large number of concepts are common (except sometimes for the name, for example avionic function versus aircraft function) even if the industrial version is somewhat more concrete, integrating technological choices. For example, the notion of resource is refined in a set of hardware elements (CPM, RDC, FieldBus, etc) while, for DO-297, a resource is a more abstract element (hardware or software or data).

The architecture model we propose aims at reconciling the two visions and providing solid grounds for the requirements and argumentation models. The architecture model is defined using an Ecore meta-model. We will not present it exhaustively, but we highlight the important concepts that have been included and their relationships.

2.1 Resource

As mentioned previously, the notion of resource is defined slightly differently in Scarlett and DO-297. In order to reconcile the two visions, we have proposed an intermediary notion of Element that is refined into Software, Hardware and Database. Hardware is itself refined into Equipment and Alimentation, and a non-exhaustive list of Equipments is considered. We also kept the notion of Component as defined in DO-297, which is a special case of Resource and to which is attached the notion of Configuration.

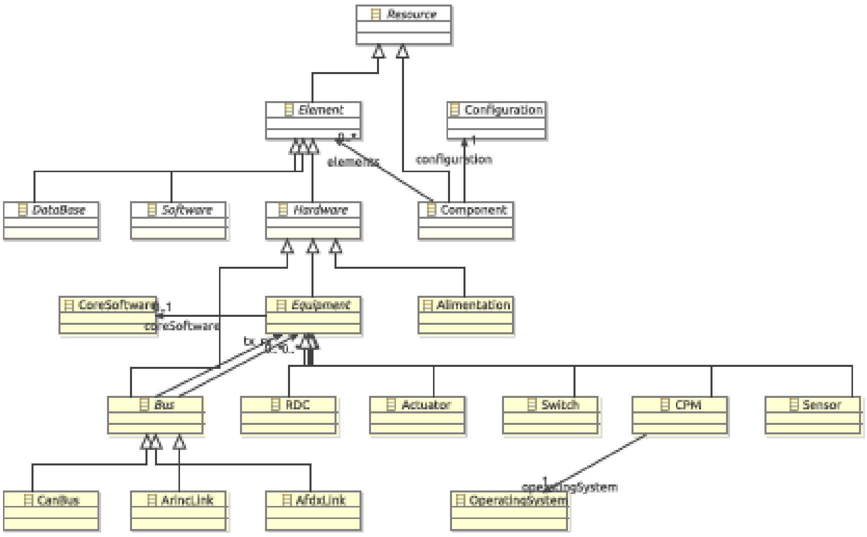

Figure 2: meta-model for the notion of Resource

2.2 Function

Functions and their architecture are not described in details in Scarlett and DO-297. However, in the MIMOSA framework, we need to be able to handle the notion of application and the allocation of functions on the platform. In our model, an aircraft function is decomposed into several functions. An application implements a function and is hosted by a partition. We have introduced the notion of Parameter. Applications produce and receive Parameters. This notion will be refined further on during the project when needed.

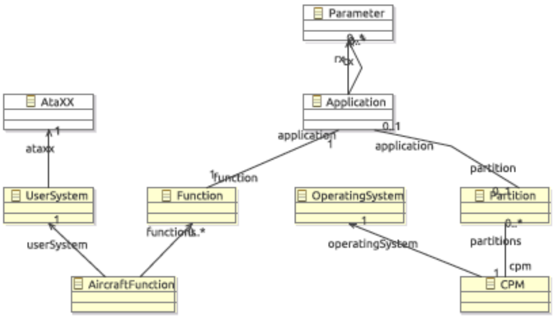

Figure 3: meta-model for the notion of Function

2.3 Module and Platform

Scarlett and DO-297 differ in the way they define a module. In the industrial vision, a module is a set of configured hardware elements (hardware + resident software + core software). In the certifier vision, a module is a grouping of resources and a module can be composed of several modules. We propose to use a combination of these two notions. In the MIMOSA model, a Module is an assembly of Applications and Resources useful for this application and can be composed of other modules. Finally, a Platform is characterized by a set of Modules.

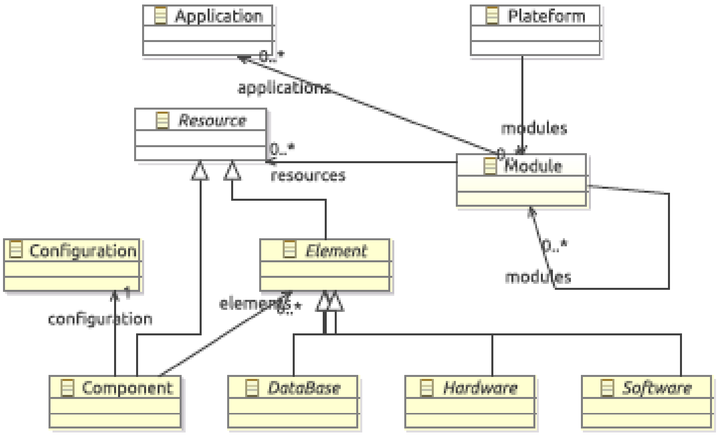

Figure 4: meta-model for the notions of Platform and Module

2.4 Relationships

The MIMOSA architecture model also includes a model of the relationships between the different notions (typing, cardinalities, etc) that will not be presented in this paper.

3.0 REQUIREMENT MODEL

The goal of the MIMOSA Requirement model is to represent requirements that can be expressed for functions hosted on an IMA platform at different levels (from very early high level requirements until concrete detailed requirements including requirements on the IMA platform). We aim at providing several levels of requirements and at introducing design choices and elements of solution only when necessary and the latest possible. In particular, we are interested in identifying the precise role of partitioning in the satisfaction of requirements. We focus on two specific viewpoints: safety and real time.

In the MIMOSA project, the approach we adopted to define the requirement model is to start from a case study (terrain avoidance system), to define categories of safety and real-time requirements on this case study, and then to identify how to express the requirements using the notions defined in the architecture model (possibly extending it when necessary).

We will not present the case study here. We only describe the categories of requirements that have been identified up to now for safety and real-time viewpoints.

3.1 Safety requirements

Safety requirements can be classified following the decomposition presented in Figure 5.

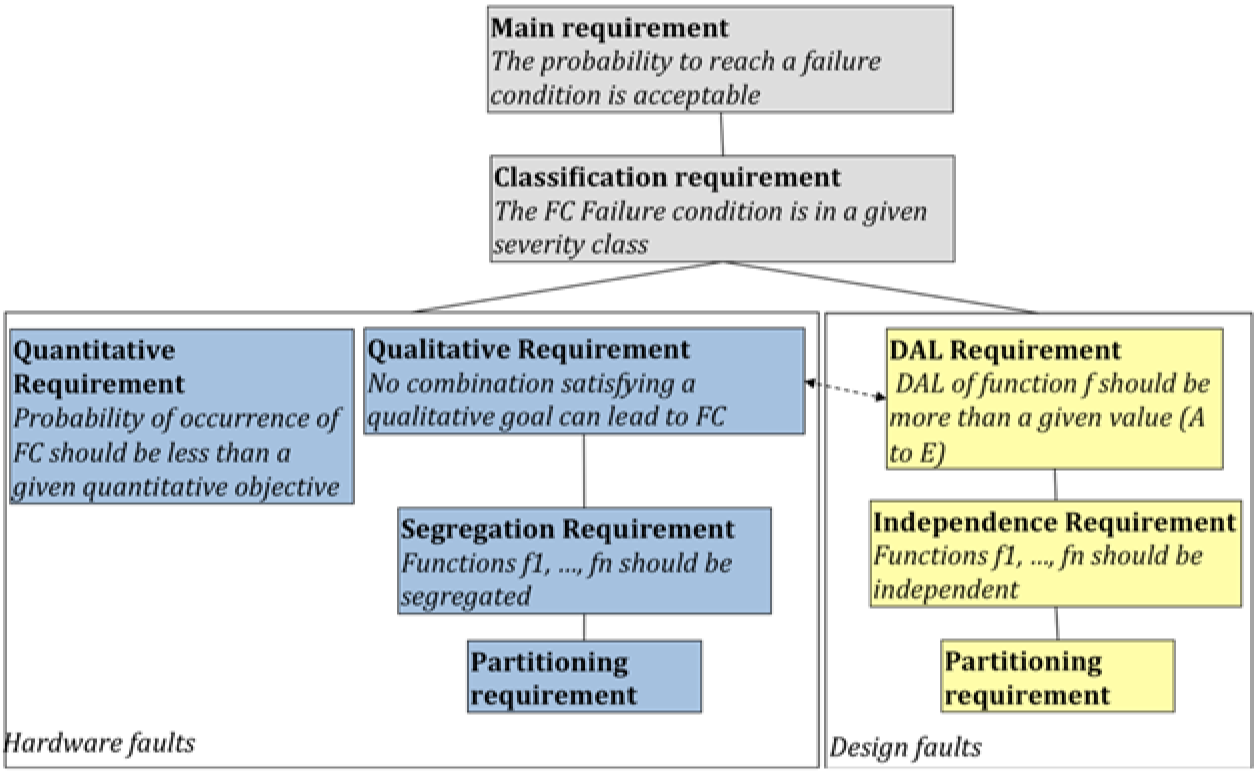
Main requirement
The probability to reach a failure condition is acceptable
Classification requirement
The FC Failure condition is in a given severity class
Quantitative Requirement
Probability of occurrence of FC should be less than a given quantitative objective
Qualitative Requirement
No combination satisfying a qualitative goal can lead to FC
Segregation Requirement
Functions f1, …, fn should be segregated
Partitioning requirement
Hardware faults
DAL Requirement
DAL of function f should be more than a given value (A to E)
Independence Requirement
Functions f1, …, fn should be independent
Partitioning requirement
Design faults

Figure 5: categories of safety requirements

For example, on the case study, one of the main safety requirements is that the failure condition “system in error” is classified as hazardous. This classification requirement is decomposed into

- A quantitative requirement: probability of occurrence of this failure condition is less than 10^{-7}/FH;
- A qualitative requirement: no single fault should lead to this failure condition;
- A DAL (Development Assurance Level) requirement concerning one of the functions: the DAL for VerticalSpeed should be at least equal to B.

3.2 Real-Time requirements

Classification of real time requirements is not as clearly identified as for safety ones. Indeed objectives and activities of the safety assessment process are described in ARP-4754 [3], while no specific standard addresses real-time requirements. Real-time requirements are necessary to achieve some of the functional requirements at system and software level, they are also part of platform specification. In MIMOSA, an objective is thus to propose a classification of real-time requirements that is compatible with certification standards and current industrial practice.

The current classification is presented in Table 1.

Table 1: classification of real time properties

Functions	• **-** End to end Latency • - End to end Freshness • - End to end Reactivity • - Temporal consistencies		
One task on the Platform	• WCRT (Worst Case Response Time) • Maximum Jitter wrt specified period	**Data on the network**	• WCTT • Maximum Jitter
One task alone	Resource usage • WCET (Worst Case Execution Time) • Stack usage	**Data alone**	• Elementary WCTT (Worst Case Traversal Time)

At the highest level, functions are required to meet end-to-end temporal constraints. For instance let us consider the terrain avoidance function. It computes flight control orders (such as aileron angles) in order to maintain a constant altitude. For that purpose, it takes as entry an image of the ground in front of the aircraft provided by the radar. The function has to meet freshness and latency requirements. In normal mode, at any time the aileron position must reflect a ground image acquired at most 100ms before. In other words, the worst-case freshness of the aileron angle must be less than 100ms. In case of radar failure, the function has to apply an emergency up order to the flight control surfaces at most 100ms later. The end-to-end latency from the radar failure to the emergency up order must be less than 100ms.

At platform level, the function is then refined into software tasks executing on specific equipments communicating through channels and buses. The functional requirements (freshness and latency) are also refined at the task level into platform requirements such as:

- Worst case response time requirement for each software task involved in the function, i.e., maximal duration between an input and the related output taking into account the scheduling strategies and all the tasks running concurrently on the same platform;
- Maximum jitter with respect to the period of each task, i.e. the greater variation of the duration between two consecutive output productions;
- Maximal traversal time of each message involved in the function through the communication means (buses, network, etc).

These requirements are finally refined into lower level requirements for each task: the maximum execution time of each task when running alone on the platform (without any perturbation coming from other tasks or the operating system), and the maximum traversal time of each message alone through the communication means.

4.0 ARGUMENTATION MODEL

The goal of the Argumentation model is to represent the different means of compliance used to justify the satisfaction of the requirements. The Argumentation model is an essential part of the MIMOSA frame of reference. It organizes the various elements (formal and informal) that contribute to the justification of requirements.

Classically, argumentation distinguishes three concepts:

- Claim: the property to be justified (requirements here),
- Evidence: the facts that will be used to justify the claim (analysis results, test results, expert knowledge, bibliographical reference, etc),
- Argument: strategy to combine different evidences in order to justify a claim.

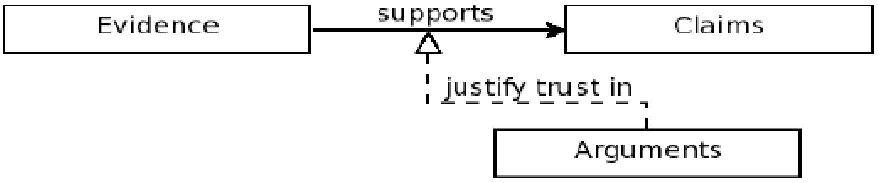

Figure 6: Argumentation concepts

In the MIMOSA project, we have studied three existing approaches: KAOS [4], GSN [5] and GM-VV [6]. KAOS and GSN are coming from the requirement engineering community, GSN is more common than KAOS in the industrial domain. GM-VV is a standard but the graphical notation is quite complex to use. We only give here a brief overview of GSN.

4.1 GSN (Goal Structured Notation)

GSN is a graphical notation designed to structure the representation of an argumentation. Figure 7 illustrates the notation on a very simple example. Key concepts are:

- Goal: what the argumentation aims at establishing (G1 on Figure 6), a goal can be refined in sub-goals (G1.1, G1.2, G1.3);
- Strategy: it is possible to add an explanation of the decomposition of a goal into sub-goals, this explanation is called strategy (Str1);
- Context: allows to add context information for some elements of the argumentation (C1);
- Justification: explains why an element of the argumentation is sufficient (J1);
- Assumption: facts that are outside the perimeter of the argumentation (not studied) but necessary for the argumentation (A1);
- Evidence: leave of the argumentation tree (Doc1).

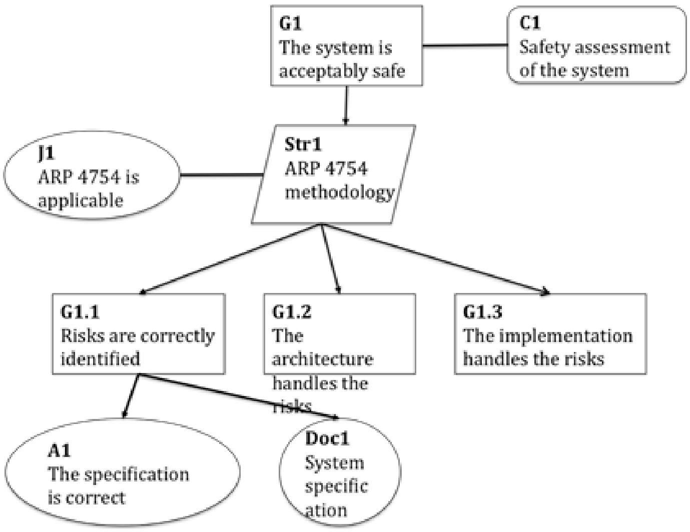
G1
The system is acceptably safe
C1
Safety assessment of the system
J1
ARP 4754 is applicable
Str1
ARP 4754 methodology
G1.1
Risks are correctly identified
G1.2
The architecture handles the risks
G1.3
The implementation handles the risks
A1
The specification is correct
Doc1
System specification

Figure 7: GSN example

In order to define the MIMOSA argumentation model, we need to identify the nature of the goals, evidences and arguments (strategies) specific to IMA architectures.

5.0 CURRENT WORK

The MIMOSA project is still under way. Current work addresses two main domains:

- Enriching the Architecture model with notions that are necessary to represent safety and real-time requirements and making the link with existing formal models for safety and real time analyses;
- Defining an Argumentation model dedicated to IMA certification: we are working on specific evidences and strategies adapted to the safety and real-time requirements we have defined, and more generally to IMA requirements.

REFERENCES

[1] DO-297/ED-124. RTCA and EUROCAE. Integrated modular avionics (IMA) development guidance and certification considerations.

[2] Scarlett project. http://www.scarlettproject.eu/

[3] ARP 4754. Guidelines for development of civil aircraft and systems.

[4] KAOS. http://www.info.ucl.ac.be/~avl/ReqEng.html

[5] GSN. http://www.goalstructuringnotation.info/

[6] GM-VV. http://www.sisostds.org/StandardsActivities/DevelopmentGroups/GMVVPDGGenericMethodologyforVVAintheM.aspx