SAVOIR MODEL-BASED AVIONICS.

Barthélémy Attanasio[(1)], Federico Di Domenicantonio[(1)], Francesco Colombi [(1)], David Peña Hidalgo [(2)]
[(1)]Thales Alenia Space, e-mail: barthelemy.attanasio@thalesaleniaspace.com

***ABSTRACT* – In the space domain, avionics is by nature a challenging part of the system, as it is distributed over different sub-systems and involves different disciplines (DHS, AOCS, FDIR, SW, HW, RAMS, …). For many years, the SAVOIR Working Group tried to elaborate a set of documentation that has been successfully released to the industry with requirements defining the need of the avionics (functional architecture and also equipment). It has been proposed to use a Model-Based System and Software Engineering approach, in particular thanks to Capella, to have a common understanding of the SAVOIR functional architecture including new technologies. Thanks to its experience and its continuous efforts in this area, Thales Alenia Space can draw a global picture of how to use the Capella tool and how to understand the SAVOIR reference architecture model.**

KEYWORDS – MBSE, Capella, Avionics

1. Introduction

The avionics architecture of a spacecraft can be well summarized at functional level by the SAVOIR reference architecture:

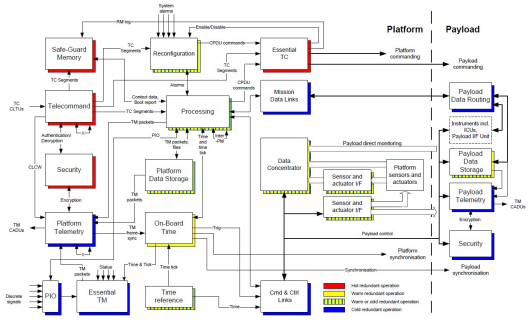

Figure 1: SAVOIR Avionics functional block diagram

This block diagram has a lot of advantages but is not driven by functional chains. There are several units which handle all these avionics functions : On-Board Computer, Remote Terminal Unit, Instrument Computer Unit, Mass Memory Unit… This diagram only gives a glimpse at a very complex design architecture which is then formally converted in an avionics design process, starting from the expression of functional needs to the HW and SW implementation with a lot of interactions between engineering disciplines (RAMS, SW, HW, …).
The space engineering world and more specifically the avionics activities expressed its interests in using more and more MBSE tools to optimize the work performed. Using models as references for several activities brings a high added value to this job.
This paper explains how Thales Alenia Space used Model-Based System Engineering for modelling the SAVOIR functional reference architecture.

2. Logical Architecture

The first step was to develop the model at logical layer of the SAVOIR functional reference architecture. This led to a Logical Diagram which is almost equivalent to the block diagram of the Figure 1 but with functions instead of blocks; functions that can be refined with the detailed requirements.

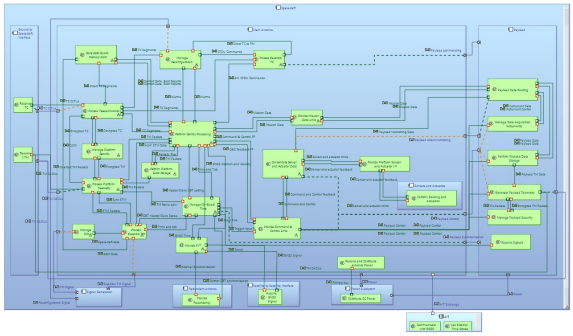

Figure 2: SAVOIR Avionics LAB

After having detailed all the functions in sub-functions to refine the logical model, several functional chains have been created in order to explain the interactions between the functions. Each diagram is centered in one of the main avionics functions defined by SAVOIR (basically one main function is one section of any SAVOIR document).

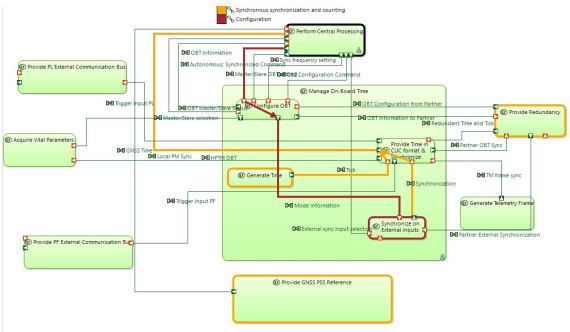

Figure 3. "Manage OBT" LDFB

The previous figure presents an example of functional diagram with main avionics functions and one avionics function detailed in sub-functions. Functional chains are

included to explain which functions are linked together and their logic of data exchange.

3. Use of the Model for Requirement Checking

The first use of the Capella SAVOIR model is to check that it is consistent at logical level to have a common understanding of the functional architecture and that is fulfil the functional requirements.

The process to integrate requirements in the model is well mature thanks to the Open-Source Capella Requirement Viewpoint and once they are loaded in the model, it remains to build the link with functions that fulfil the corresponding requirement.

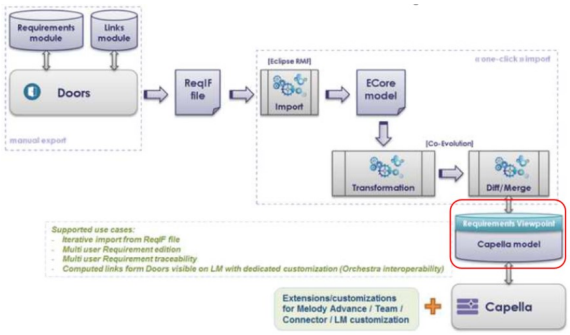

Figure 4: Requirement Import Process

Thanks to that, all the objects populating the Capella model have been linked to the already existing requirements, that can become mainly Logical Functions, Logical Exchanges or Logical Chains. Several diagrams (LDFB) can be defined for a given function in order to underline a different logical chain or to highlight a different functionality or operation of the system.

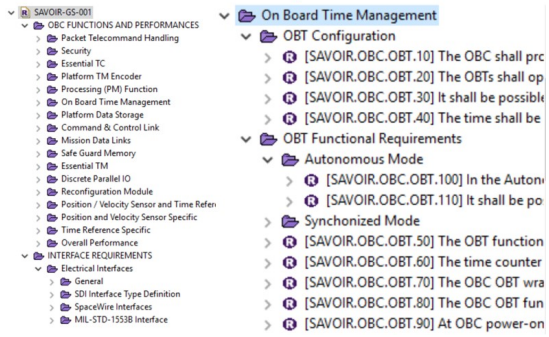

Figure 5: Requirements linked in the model

Interesting outcomes can result such as the compliance matrix or the requirement traceability with parents requirements. This is something very useful because the use of the model can generate automatically this kind of document which are often a source of time consuming work with long discussions.

Requirement	Allocation status (If field is empty, then no allocation)
[SAVOIR.OBC.TM.590] *Output data rate stability*	Encode TM *As* LogicalFunction TM CADUs *As* FunctionalExchange TM CADU to EGSE *As* FunctionalExchange TM CADU to Partner *As* FunctionalExchange
[SAVOIR.OBC.TM.600] *TM Time Strobe test point delay*	Encode TM *As* LogicalFunction TM Time Strobe Pulse *As* FunctionalExchange
[SAVOIR.OBC.PM.10] *No of Processor Module functions*	Provide Redundancy *As* LogicalFunction Perform Central Processing *As* LogicalFunction

Figure 3: Traceability Matrix Automatically Generated

4. Physical Architecture

After having mapped all the requirements in the Logical Architecture, the transition to the Physical Architecture is performed, by allocating the Logical elements to the Physical ones (functions, exchanges and chains). By using the REC/RPL Capella functionality, two different physical architectures are presented (based on IPAC and NG_Ultra), showing how the same Logical Architecture can be implemented in several ways.

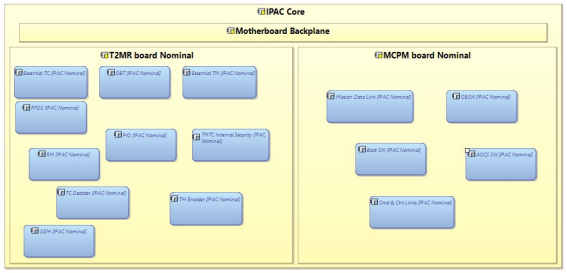

Figure 6. IPAC Component allocation

This proves that from the same functional architecture, several architectures can be derived with their corresponding HW architecture and HW/SW functions sharing.

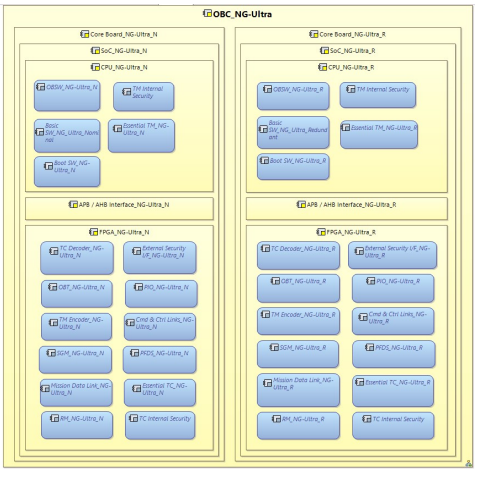

Figure 5. OBC NG Ultra Component allocation

A specific focus has been done for the OBC specification and the OBC detailed design but the RTU unit have also been modelled and unfortunately, the data storage was not studied in details. This approach can be extended to other sub-systems.

5. SAVOIR Refinement Proposal

The last step of this work was to look at the SAVOIR specification with a Model Based perspective. Defining the Logical Architecture gives the opportunity to propose a restructuration of the SAVOIR specification, by listing two kind of requirements:

- *Redundant requirements*: there are requirements, especially translated in functional exchanges, that are present in different parts of the specification and that are mapped to the same object. Only one requirement per object is necessary.
- *Requirements which are too low level*: some requirements, even if they are in the generic OBC specification, are too much linked to the actual implementation of the model. They have been reworded, deleted, or moved in another document.

6. Conclusion

The study is a proof of concept of how an already defined avionic architecture can be modelled using a Model-Based approach, and how the process itself can provide a feedback on matters that would not be highlighted when working in a traditional way.

Acknowledgments
The authors would like to acknowledge the support of the European Space Agency, the involvement of ESA Technical Officer in the study, their advises and their confidence in Thales Alenia Space MBSE teams. This enabled to make the MBSE community grown and having coherency between the space ecosystem from a common model to be used in space programs.