Integration of different MBSE approaches within the design of a control maintenance system applied to the aircraft fuel system

Integration of different MBSE approaches within the design of a control maintenance system applied to the aircraft fuel system

Claudio Pessa, Massimo Cifaldi
Leonardo Aircraft
Finmeccanica S.p.A
C.so Francia 426, 10146, Torino, Italy
claudio.pessa@leonardocompany.com
massimo.cifaldi@leonardocompany.com

Eugenio Brusa, Davide Ferretto,
Ksenija M.N. Malgieri, Nicole Viola
Department of Mechanical and Aerospace Engineering
Politecnico di Torino
C.so Duca degli Abruzzi 24, 10129, Torino, Italy
eugenio.brusa@polito.it, davide.ferretto@polito.it,
nicole.viola@polito.it

***Abstract*—The design of a Control Maintenance system (CMs) for an airborne platform deeply deals with the mission, the on–board systems interfaces and the identification of their behaviour in operation. This paper describes how the Model Based Systems Engineering (MBSE) was applied to an industrial test case to perform the functional design of an innovative CMs to be integrated with the aircraft Fuel system (Fs). The impact of different approaches applied when modelling the two systems and their integration through the SysML was investigated. As the IBM Rational Rhapsody® tool was used, the Harmony® methodology was applied to the CMs, while a MBSE customized approach was implemented for the Fs, even to cope with some differences in coupling an avionic system to a physical one.**

Keywords*—*MBSE, avionics, interoperability of models, material product development, design methodology.

I. Introduction

Aeronautics is a typical field of application of the Systems Engineering (SE) [1], since aircraft includes many interconnected on–board equipment. They are deeply integrated within the whole aircraft, thus increasing its complexity, especially because of some safety–critical issues [2]. The trade–off of system layout and the selection of functionalities are crucial issues of design, being aimed at fitting requirements and customer needs. Avionic systems based on electronic devices have to be integrated with material components of the on–board systems, thus motivating an intrinsic multi–disciplinary approach in modelling a number of interfaces [3]. Several suppliers and divisions of the same industry might be involved in this process, thus leading to apply different approaches. The SE assures an effective modelling solution as well as a strong requirements traceability and allocation process. The Model Based Systems Engineering (MBSE) [1] fits the need of sharing models, tools, software and a standard reference language as the SysML [4–6]. Moreover, a continuous tracking of changes is even guaranteed. This paper investigates the development of an innovative avionic on–board system for health monitoring, being interacting with flying and ground crews and even with other material systems in operation. The MBSE was applied and the IBM Rational Rhapsody® was used to develop the functional modelling, from the requirements elicitation to the definition of system functions and of architectural elements. However, as the proprietary Harmony® approach [7] was applied to the avionic system, a non–proprietary approach was assessed to develop the fuel system. Some critical issues about the interoperability of models were found as the two systems were connected.

II. The case study

A new avionic system, namely the Control Maintenance (CMs), able to monitor the health of aircraft on–board systems and to support the on–board crew in operation as long as the ground crew over maintenance activities is herein analyzed. Its mission includes two operational modes ‘in–flight’ and ‘ground’, respectively. It shall be able to detect failures of other on–board systems, to provide some clear checklists to pilots and to save failure data for the ground maintenance and troubleshooting activities (Fig.1).
Innovation of this product consists in reducing the crew workload related to the checklists management, providing a more user friendly interface and reducing the reaction time to an unexpected event. To ensure an effective operation the system is highly integrated within the aircraft. Therefore, in this study, the functional design of the Fuel system (Fs) was selected as an example of on–board system to check the capabilities of the avionic system. Those capabilities concern the collection and computation of several data (e.g. discrete and analogue inputs) coming from many sources (e.g. sensors, control units, electronic circuitries, etc.). The effectiveness of the MBSE approach was even tested in case of implementation of different design techniques.

AKNOWLEDGMENT

This activity was performed within the frame of the “CRitical sYSTem engineering AcceLeration” (CRYSTAL), funded from the ARTEMIS Joint Undertaking under grant agreement n. 332830. Authors acknowledge the research project CRYSTAL for funding this study and all of Partners for their fruitful cooperation within the frame of the CRYSTAL community.

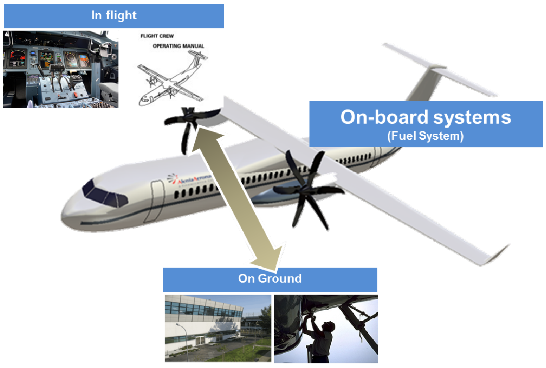

Fig. 1. Sketch of the case study

The functional design of the CMs and of the Fs was in fact performed through the IBM Rational Rhapsody® tool [7]. The CMs design was driven by the Harmony® methodology whilst the Fs was studied by following a customized SysML modeling approach. A key issue of the MBSE is the trade–off among some proposed system architectures, being aimed at finding the best one, able to match requirements and functionalities. The process must prevent any prejudice of the user about some existing solutions, i.e. each designer shall be able to select a configuration based only upon a deep analysis of system requirements. Therefore the effectiveness of the MBSE approach was investigated when predicting the functional behavior of some non–homogeneous system (avionic and mechanical), within a realistic industrial scenario. The interoperability of those models was then assessed as a crucial issue of this study.

III. SYSTEM ANALYSIS AND DESIGN

To describe the different approaches used for the implementation, the system design and analysis are herein presented separately for the CMs and the Fs, although the design process refers to the so–called "V–diagram" [1].

A. Control Maintenance system design

The CMs was developed from a preliminary elicitation of requirements to identify some functional scenarios compatible with those related to the aircraft operations. They were derived by applying the Harmony® methodology, i.e. by following the sequence of activities described by SysML diagrams as in [7]. Those operational scenarios were used to derive a set of logical capabilities. Only once that the aircraft capabilities were preliminarily established, the analysis of the CMs functionalities could be performed. Two different Rhapsody® models were built up to define, separately, the aircraft and the CMs (i.e. system) layers. To move from the first to the second model a hand–off process was required. For both layers the Requirement and Functional Analyses, and the Design Synthesis were performed. Requirements and use cases were first instantiated, then the Functional Analysis characterized the use cases through the typical SysML diagrams [4] and a functional architecture of system capabilities was found, as a "black box", being solution independent. The system architecture was defined within the Design Synthesis, by allocating the system operations on physical instances and associating then some capabilities to a specific configuration of the System of Interest (SoI). This approach usually allows distinguishing among the functions required and the material components which will be used, being dependent upon the selected technological solution. Block Definition Diagrams (BDDs) and updated Activity Diagrams (ADs) were plotted to perform the allocation and to build a "white box" view of the SoI. It allowed exploring several solutions, based on different kinds of real components, when available. Those two layers correspond to two steps of the analysis. In practice, the design of the CMs was based upon two strategies, concerning the technical activity and its implementation. The technical activity includes the analysis performed by following the process described by the "V–diagram", from the aircraft to the system level, while the implementation is strictly related to the modeling process applied, i.e. the Harmony® in this case (Fig.2).

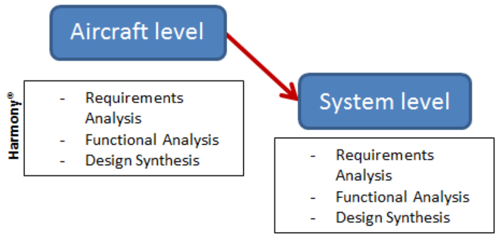

Fig. 2. Design methodology applied to the avionic system

B. Fuel system design

The design of the CMs needs deriving requirements and capabilities from the higher level of the aircraft. In principle, it starts only when the aircraft model is ready. In this case, a customized approach was applied for the Fs, by following some typical practices of the commercial aircraft design [8]. To avoid a detailed description of the whole aircraft layer, a list of requirements was written by resorting to the aircraft ones, and used as an input for the system level analysis. This approach allowed investigating the subsystems and setting up a proposal for the whole aircraft configuration. The technical activity was performed through a development process typical for aircrafts [9,10].

Requirement, Functional and Logical Analyses were used to define a preliminary allocation of system functions to a logical architecture, which was then modeled through a Physical Analysis to find a physical structure of the fuel system (Fig.3). Two main differences can be highlighted between this methodology and the Harmony® one. A new layer, namely the Logical Analysis, was used to allow a smooth transition between the "black box" and the "white box" views (see Sec. IV). The structure of the Functional Analysis phase was even changed. A deeper analysis of the use cases and scenarios was performed, since the use case analysis is applied in aeronautics to refine the requirements coming from the aircraft level [8,9]. Creating a set of scenarios, aimed at better describing the use cases through their realization, is moreover useful in identifying some interactions between the system and several external actors (see Sec. IV).

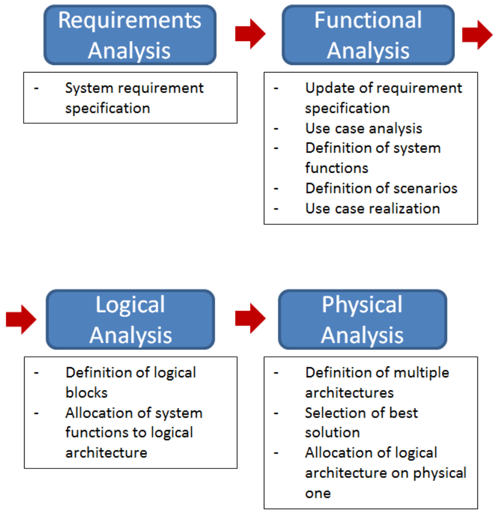

Fig. 3. Design methodology for the fuel system

IV. Implementation - The Central Maintenance System

A. Requirements Analysis for the CMs: aircraft level

Once that a list of the aircraft requirements was composed by the manufacturer design unit, it was written and classified by resorting to the IBM Rational Dynamic Object–Oriented Requirements System (DOORS®). It allows enabling traceability and analysing the change impact, while it acts as a database where requirements are created, updated and managed. Requirements were classified through their functionality, usability and interfaces. Some operational "storyboard" requirements were even derived, by looking at the interaction with pilots, ground crew and operators, being stakeholders and final users. This approach allowed highlighting some actions performed by external actors. Storyboards are effective in modelling the aircraft functional scenarios, while system requirements are usually focused on a single system. This first set of requirements allowed starting the Harmony® process within Rhapsody®. Requirement Analysis was driven by the use cases, being exploited to clarify where each requirement could have an impact on the system for a given scenario. At aircraft level, two use cases were proposed (Fig.4). The stakeholders in this case are only external actors, i.e. pilots, ground operators and the Customer Maintenance System (CMS), to be distinguished from the CMs.

B. Functional Analysis for the CMs: aircraft level

The Functional Analysis was then performed to identify the system functions and operations. Each use case was analysed separately. The use case 'Manage failures and maintenance', for instance, was first described through a "Black Box" Activity Diagram (BBAD) (Fig.5).

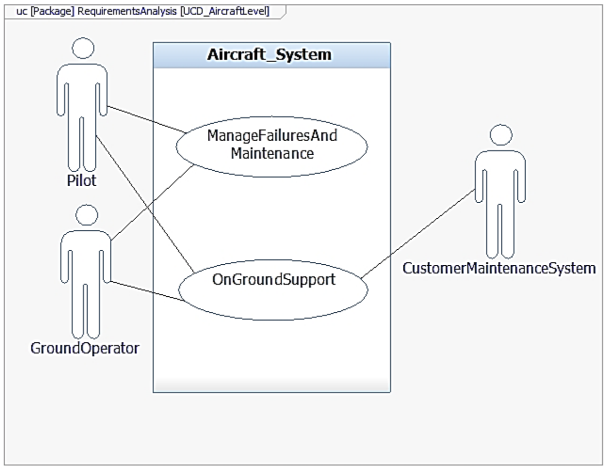

Fig. 4. Use cases for the avionic system at aircraft level

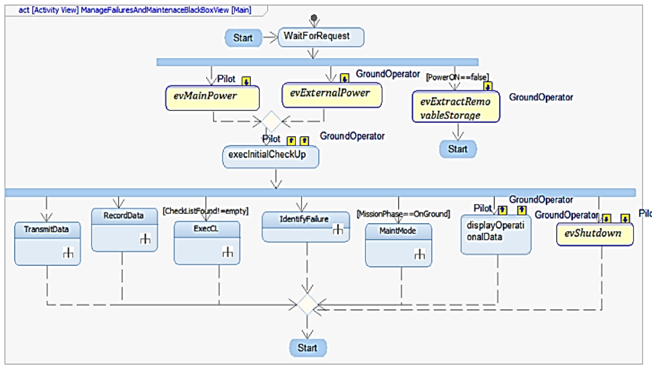

Fig. 5. Black box activity diagram for 'Manage failures and maintenance'

It provides a flow of actions performed by the system, defines the main interactions with actors and all of events foreseen for the use case. It is referred to as "Black Box" since actions are not yet instantiated. Data were used to draw the Sequence Diagrams (SDs), where activities are represented as a function of time. It is worthy noticing that when paths described by the BBAD are sufficient, all of operational scenarios are represented by the SDs. Relations between the system and the actors are described by the activity and sequence diagrams. They are then instantiated into state views via the State Machine Diagrams (SMDs), which are able to communicate with each other thanks to the instantiation of interfaces within the related Internal Block Diagrams (IBD). All those diagrams provided a complete description of the system functional behaviour and capabilities (Fig.6) and were used to perform functions/product allocations during the Design Synthesis.

C. Design Synthesis for the CMs: aircraft level

The allocation of operations performed at aircraft level to a set of logical capabilities was used to define the system architecture. The rationale is based on the idea that logical capabilities are actually use cases at system level (Fig.7), as described in Section IV.D.

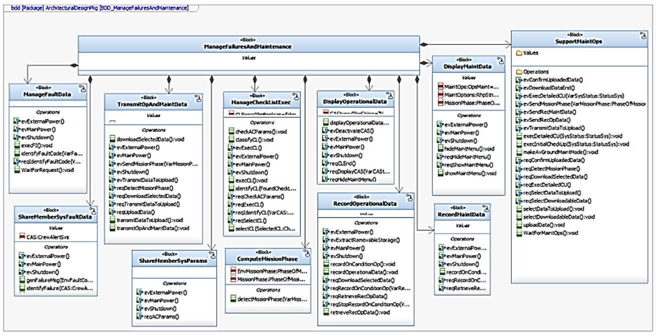

Fig. 6. Block definition diagram with system capabilities architecture

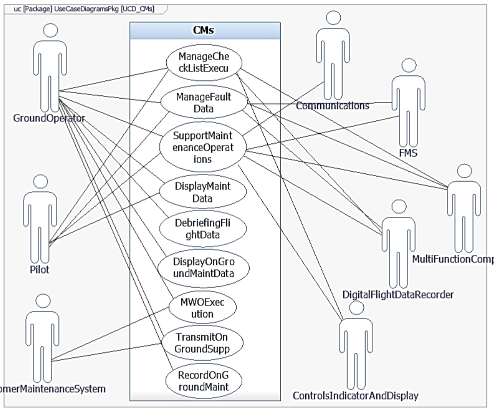

Fig. 7. Use case diagram for the system level analysis

Operations were allocated to system capabilities, through a 'White Box', being a version of the Activity Diagram (WBAD) enriched with 'swimlanes'. This element is based on a partition of the diagram which separates the actions and associates them to the capabilities. The 'white box' process drove to create a new version of each diagram of the Functional Analysis (SD, IBD, SMD) where relations with capabilities were specified, thus defining the configuration of the SoI. Finally the system behaviour and the model could be validated by executing the SMDs. A formal simulation was activated highlighting each system state in sequence, under the control and triggering of some Rhapsody® panel diagrams, where the user sets up some main parameters (Fig.9).

D. Requirements Analysis for the CMs: system level

The system level was then investigated. A hand–off procedure was required to make available the results of previous level to the new one. First at all a new project was built up in Rhapsody®, focused on the system. This model required accessing to capabilities represented by the BDD (Fig.6) and their allocations, defined within the 'white box' analysis at aircraft level. System level analysis is aimed at developing the logical capabilities in order to define system architecture and to identify the interfaces useful for system characterization. Requirements defined by the customer specifications were already allocated on the logical capabilities. Therefore their coverage was intrinsically assured by tracing the relations between operations and contents of the CMs model at system level. A main hand–off output was that logical capabilities (Fig.6) became the use cases at system level and the ports identified within the aircraft level IBD specify the actors involved (Fig.7).

E. Functional Analysis for the CMs: system level

The Functional Analysis of the CMs was performed for each use case as in previous step. A 'Black Box' view was developed by looking at the imported 'White Box', which describes the interfaces with external actors. As for the aircraft level a set of behavioural diagrams was created. The IBD was used to define all the communication nodes (ports and interfaces) between the system and the actors. A SMD was exploited to simulate the system behaviour.

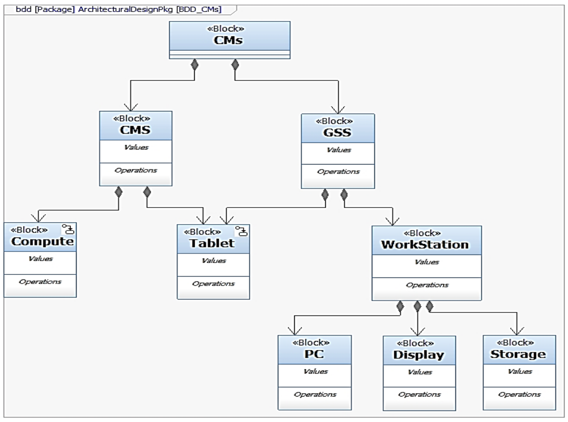

Fig. 8. CMs architecture (BDD)

F. Design Synthesis for the CMs: system level

The physical architecture was defined through some BDDs (Fig.8). The system operations were allocated on each block, by following the 'white box' analysis, as in Section IV.C. A formal simulation of the system could be performed. Main elements of the CMs actually are the on–board computer and a tablet, being exploited as a user interface, therefore panel diagrams allowed the execution of the SMD even through some visual rendering of those devices (Fig.9).

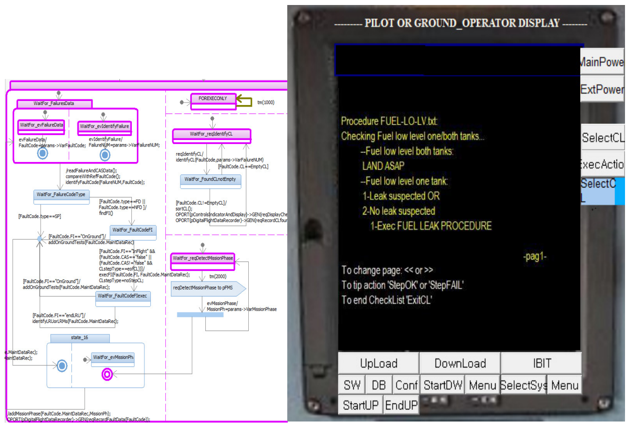

Fig. 9. Simulation of the on-board computer

V. Implementation - The Fuel System

A. Requirements Analysis for the fuel system

As the Fuel system was designed by a different unit, it was developed through a tool–independent approach. The aircraft level analysis was represented by a high level requirements specification managed within the IBM Rational DOORS® as previously described. However in that test case a complete aircraft model was unavailable. This was due to a preliminary priority constraint that decided to reuse a fuel system already operative whilst the aircraft configuration could be innovated. Therefore some standard requirements were organized by fitting the ASD S1000D for the aerospace systems [11]. They preliminarily defined the aircraft layout, number and type of engines, number and location of tanks, some interfaces with other systems, etc. Other requirements at system level were added by resorting to the stakeholder analysis, existing technologies, industrial and domain practices as long as some interviews with customers.

B. Functional Analysis for the fuel system

In the following, the use cases at system level were set up, together with the related UCDs (Fig.10) and then new requirements were derived by analysing the operational scenarios. For each use case some SDs were drawn as in Fig.13, where the "Feed LH engine" context is presented. It is worthy noticing that within this approach the use cases are 'goals that actors want to reach by using the system' [5]. So far the CMs looks like an external actor for the Fs and has some dedicated objectives and use cases. Requirements were linked to use cases as previously done for the CMs, but system functions were derived directly by deducing the usage dependencies between system and use case (Fig.11).

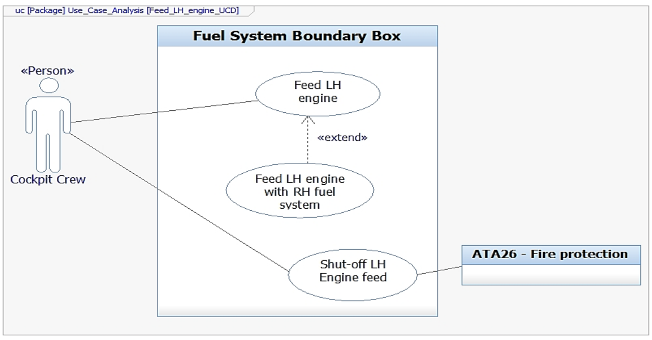

Fig. 10. Example of use case diagram for fuel system

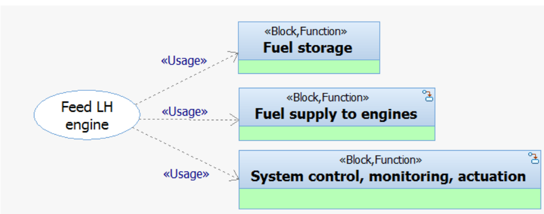

Fig. 11. Example of connection between use cases and functions

In practice, it was investigated what kind of function the system has to provide to reach the goal related to a use case. This approach allows a direct connection to a functional tree of the Fs that can be used then for the safety assessment process [10]. This system actually includes several issues related to Reliability, Availability, Maintainability and Safety (RAMS), being critical for the design of the aircraft. Specialists of RAMS actually use this list of functions as an input for the Functional Hazard Assessment (FHA). It can be remarked that the integration of processes described by the SAE ARP4754A and ARP4761 [9,10] poorly matches with the Harmony® methodology, since it might look stiff for a straight implementation of the FHA, because, for example, of the lack of a real functional tree, and other safety assessment tools.

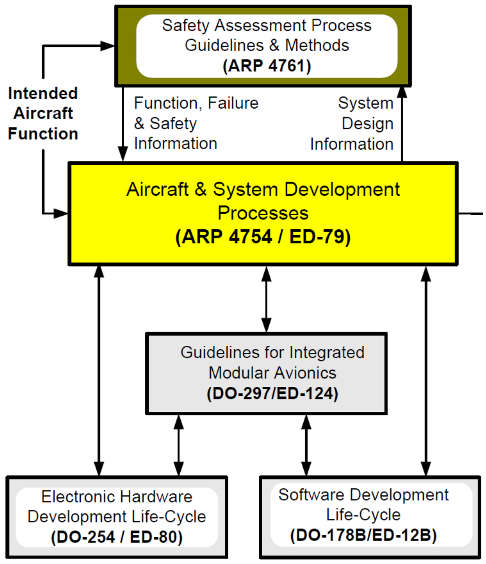

Fig. 12. Typical Aircraft and System Development Processes

In general, Safety assessment process is based upon a defined set of phases that use the data coming from the MBSE approach. Functions, Logical and Physical blocks can be used respectively for FHA, reliability allocation and prediction, involving Fault Tree Analysis (FTA), Reliability Block Diagrams (RBD) and, ultimately Failure Modes and Effects Analysis (FMEA). The results of the MBSE process can be thus very helpful in terms of data (System design information as indicated in Fig.12) provided to the safety and reliability experts that typically base their work upon the specifications stored in a document-based way. On the other hand, function, failure and safety information can be acquired by systems engineers from the RAMS staff to enrich the analysis and to keep the traceability with the safety and reliability aspects as well as to further characterize the operational background of the system. Operational scenarios are implemented through the SDs to include actors and functions, thus defining operations, messages and events exchanged among them over time (Fig.13). They provide a sketch of the logical architecture of system, being subsequently analysed.

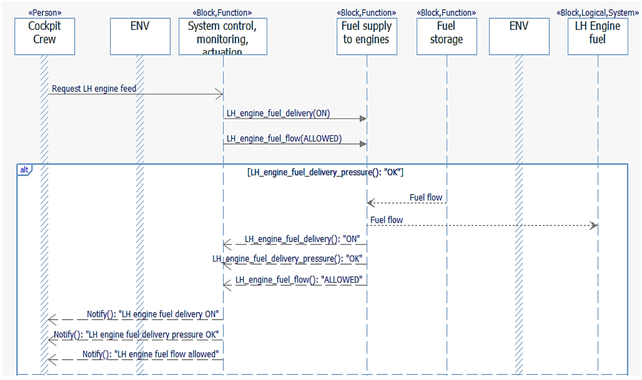

Fig. 13. First part of sequence diagram for left hand engine feed scenario

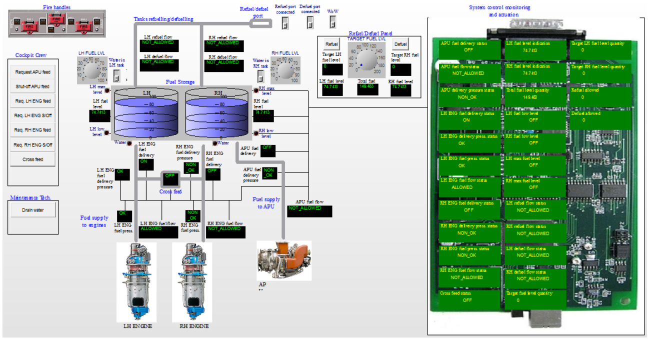

Fig. 14. Formal simulation of fuel system functional behavior

Once that operations were defined, system functions could be expressed by the SMDs and a formal simulation was performed. Connections among the SMDs were implemented through a main IBD, whilst a dedicated panel diagram was created to control the execution (Fig.14). During the functional analysis, messages and outputs to be sent to the CMs were defined and the simulation was prepared to provide these data for a next integration.

C. *Logical Analysis for the fuel system*

Logical Analysis is an intermediate level of abstraction between the Functional Analysis, where only functions are described, and the Physical Analysis, aimed at selecting the real components to build up the system. Logical subsystems are only entities that perform the system functions. Each logical block is linked to one, or more, functional block. Connections and topology among logical subsystems is established through some IBDs to define the system architecture, even during a dysfunctional analysis which might contribute to the trade–off process. These logical blocks can be used to start hypothesizing the levels of reliability of the system in terms of failure rates, Mean Time Between Failures (MTBF) and Mean Time To Repair (MTTR) basing on their breakdown and on typical reliability and maintainability characteristics of similar components, as explained in Section V.B. Logical blocks, in fact, may contain already some preliminary features that will be developed within physical analysis in detail. This is the real importance of including a logical analysis within the design process of this kind of aeronautical on-board system.

D. *Physical Analysis for the fuel system*

The Physical Analysis consists in representing the final system architecture and contains the allocation of the Logical Analysis and of the requirements. A BDD (Fig.15) is used to instantiate the hierarchy of subsystems and components whilst a matrix is used to check the requirements satisfaction, being assured by dependencies among them and the physical blocks. Traceability is complete and continuous, since all the initial requirements, eventually updated, are linked to the use cases, which are used by functions. These functions are allocated on logical blocks and instantiated as physical components. RAMS features are eventually updated and blocks are completely characterized.

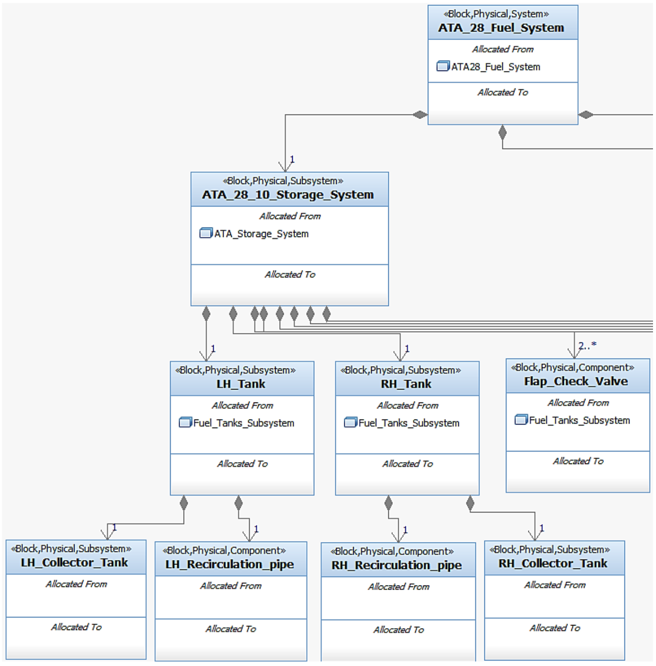

Fig. 15. BDD of the fuel system for Physical Analysis (detail)

VI. Systems Integration: Overview and Opportunities

The integration between the CMs and Fs is a typical situation where a computer-based system interacts with another one, which has low computational capabilities, being characterized by physical behaviour. In this case study, the CMs analyses the health status of the Fs providing fundamental information to the pilot, in order to take corrective actions during operations, and to the ground crew in case of inspections and maintenance activities. Apart from the connectivity aspects, that are not so stressed in the presented example, the whole system including CMs and Fs can be considered as a small and simple Cyber Physical System (CPS) that uses human operator as mean to control and correct Fs behaviour. The integration with the user was already sketched through the implementation of dedicated panel diagrams used to drive the formal animation of the state machines, in order to simulate input/output relationships (Fig.9 and Fig.14). However, an integration strategy shall be proposed in order to provide a coherent interface between CMs and Fs, to test the behaviour of the system, acting as a whole, and proving that the methodology used to define the features of the subsystems does not affect the final result. Concerning this aspect, a distinction shall be made in order to identify what kind of integration is required and what is its aim. In fact, if a simple merge of the Rhapsody® models is requested, in order to provide a representation of the assembly of CMs and Fs, the procedure available through the proprietary Diff–Merge® tool can be sufficient in order to have a logical model able to describe the whole scenario. This approach can allow the creation of a new model, made from the elements and properties of the previous ones, properly selected, which has to be in any case customized to enable a correct set of interfaces and to refine the available data. These checks are required both because the Diff-Merge® does not perform an organic integration process, being more focused on the identification of the elements that are not shared between the models and thus allowing the user to manually select the data to be kept and/or updated, and since the methodologies used have a considerable impact on the model structure and on tool capabilities. For example, the Harmony® process creates a specific architecture for the model structure, which usually is not present in case of a customized approach, both in terms of data and of organization of the links among the packages. Another choice could be focused on the adoption of one of the models as "Reference" inside the other one. This is a typical feature of the Harmony® methodology, used during the hand-off process [7]. However, this option is usually inhibited since the lack of coherence may be too high to proceed straightforward with the implementation of the final Rhapsody® project. For merge purposes, the use of the Diff-Merge tool is then considered as best solution, even if some manual adjustments can be required. On the other hand, if the integration is mainly required for simulation purposes, thus aiming at testing the complete behaviour of the assembly under specific inputs, a strategy based on Functional Mock-up can be very helpful. This approach is based on exposing a limited number of data, usually represented as interfaces with external world, that can be used as input/output architecture for other models to be integrated within the scenario. In this context, a promising interoperability standard is the Functional Mock-up Interface (FMI) [13,14,15] since, being based on the generation of standard Functional Mock-up Units (FMU) that expose only the variables involved, protecting internal algorithms, represents a flexible solution especially when a high number of tools is considered. Concerning its use within the simulation domain, this approach is generally addressed as heterogeneous simulation since different formalisms are executed together in a final environment, configured as host. With this simulation architecture, models can be exported from the original tools where they were created as standardized units FMU [16,17]. In this case, a model concept conceived to establish the interfaces among the different model parts, namely execution components, is highly recommended to improve reusability of the scenario and to state precisely their properties. A previous work about this topic was based on preliminary studies about integration of heterogeneous models [12], proving the effectiveness of the approach. An intermediate process can be set up by using a preliminary merge of the models before proceeding with the generation of the simulation architecture, either with interoperability standards or with proprietary adapters. In this case, for example, Rhapsody® capabilities can be used to perform a direct connection with Simulink® allowing an export of the IBDs containing the interfaces and reaching a level of architecture similar to the one obtained with the FMI [17]. Apart from the selected integration, which is related to the objective that the designer wants to reach, as mentioned before, this effort is justified by the need of verifying that the systems have been defined in the correct way, with a particular focus on the data exchanged, which result easier when looking at the big picture. Moreover, this integration strategies can be generalized in case of numerical simulation, allowing a connection between the functional and physical worlds, even if the definition of systems interfaces shall be managed carefully.

VII. Conclusion

This study confirmed the effectiveness of the MBSE methodology even in presence of models developed by different users and at different levels. An overview of two different methodologies was introduced both to present examples of typical problems that can arise from the design of software-based and hardware-based systems and peculiar approaches adopted to face them in the related domains, and to demonstrate how a considerable effort in interoperability topics can notably reduce the issues related to the integration of models built with different technical processes. For the case study, the design and integration of an avionic system with a classical aircraft fuel system are presented. A straight implementation of the Harmony® approach was chosen for the avionic system whilst it resulted somehow difficult for a more physical one, like the fuel system, which requires, together with the MBSE analysis, a detailed Safety Assessment. The design process has then to be assessed and organized to cope with these needs. Some problems of compatibility were evaluated and possible solutions to overcome these issues were presented by resorting to proprietary and standard communication tools, like the FMI, although an assessment of this process has still to be completed to assure a full and replicable interoperability.

References

[1] D.D. Walden, G.J. Roedler, K.J. Forsberg, R.D. Hamelin, T.M. Shortell, "Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities",2015, INCOSE, San Diego CA (US).

[2] S. Chiesa, M. Fioriti, N. Viola, "Methodology for an Integrated Definition of a System and Its Subsystems: The Case-Study of an Airplane and Its Subsystems", in "Systems Engineering - Practice and Theory", B. Cogan (Ed.), InTech, 2012, ISBN: 978-953-51-0322-6.

[3] E. Brusa (Ed.), “Mechatronics: Principles, Technologies and Application”, Nova, New York, 2015, ISBN: 978-1-63482-801-7.

[4] The Object Management Group, “OMG Systems Modeling Language (OMG SysML™)”, Vs 1.4 2015, OMG Limited, Needham, MA (US).

[5] S. Friedenthal, A. Moore, R. Steiner, “A practical guide to SysML” ed. 2014. The MK/OMG Press, Waltham, MA (US).

[6] J. Holt, S. Perry, “SysML for Systems Engineering”, ed. 2008. The Institution of Engineering and Technology, London, UK.

[7] H.P. Hoffmann, “Systems Engineering Best Practices with the Rational Solution for Systems and Software Engineering”, 2011 vs. 4.1. IBM Corporation, Somers, NY (US).

[8] B. Di Giandomenico, C. Pessa, E. Valfré, I. Viglietti, “Alenia Aermacchi needs and experience on MBSE: the Crystal user scenario”, 23rd Conference of the Italian Association of Aeronautics and Astronautics, Torino, Italy, 2015.

[9] SAE ARP4754A, “Guidelines for Development of Civil Aircraft and Systems”, ed. 2010, rev. A. Warrendale. SAE International, PA (US).

[10] SAE ARP4761, “Guidelines and methods for conducting the safety assessment process on civil airborne systems and equipment”, ed. 1996, rev. A. Warrendale. SAE International, PA (US).

[11] ASD S-1000D, “International Specification for Technical Publications” – Aerospace and Defence Industries Association of Europe - current version.

[12] A. Mitschke, E. Brusa, A. Calà, D. Ferretto, C. Pessa, G. Bachelor, “Heterogeneous simulation based on standards: deepening interoperability in trade-off analysis approach for aeronautical application”, 23rd Conference of the Italian Association of Aeronautics and Astronautics. Torino, Italy, 2015.

[13] K.M.N. Malgieri, "Sviluppo di un Central Maintenance System innovativo grazie alla metodologia Harmony SE® in ambito MBSE", M.Sc. Thesis, Politecnico di Torino, 2015.

[14] C. Bertsch, E. Ahle, E. Schulmeister, “The Functional Mockup Interface seen from an industrial perspective”, Proc. 10th International Modelica Conference. Lund, Sweden, pp. 27-33, 2014.

[15] A. Garro, A. Falcone, “On the integration of HLA and FMI for supporting interoperability and reusability in distributed simulation”, Proc. of the Symposium on Theory of Modeling and Simulation (TMS’15), part of the SCS Spring Sim 2015 Conference, Alexandria, VA (USA), April 12-15, 2015.

[16] E. Brusa, A. Calà, S. Chiesa, F. De Vita, D. Ferretto, “Towards an effective interoperability of models within the ‘Systems Engineering’ applied to aeronautics”, Proc. INCOSE Conf. CIISE 2014. Rome, Italy, pp.38-47, 2014.

[17] E. Brusa, D. Ferretto, A. Calà, “Integration of heterogeneous functional-vs-physical simulation within the industrial system design activity”, Proc. 1st IEEE-ISSE Conf. Rome, Italy, pp. 303-310, 2014.