DOI 10.5162/ETTC2024/B3.2

Flight Test Instrumentation System: Tailoring MBSE Methodologies for Prototype Aerial Systems Development

Claudia Pérez Rus[1], Stéphane Le Gal[2]
[1] Airbus Defence and Space, Avenida del Aeropuerto, s/n, 41020 – Sevilla, Spain
[2] Airbus Defence and Space, Avenida del Aeropuerto, s/n, 41020 – Sevilla, Spain
claudia.perez.rus@airbus.com
stephane.le-gal@airbus.com

Abstract

This paper investigates the application of Model – Based Systems Engineering (***MBSE***) methodologies to prototype aerial systems, with a specialized focus on the complex modelling requirements of Flight Test Instrumentation (***FTI***) system. Unlike conventional embedded aircraft systems, FTI possesses unique characteristics and challenges. Traditional design and implementation approaches often struggle to accommodate the specialized nature of FTI, leading to misunderstandings and inefficiencies in system communication and integration. New Airbus programs are bringing the shift towards MBSE, evidencing the necessity of adapting these methodologies to address the particularities of FTI. This article emphasizes the need for tailored modelling techniques capable of capturing the nature of FTI systems. This study proposes a structured framework that enhances system-wide visibility, facilitates seamless integration with other aircraft systems, and enables more effective collaboration among multidisciplinary teams. This research goes through the challenges of customizing MBSE methodologies to suit the specific requirements of FTI, ultimately contributing to more efficient, sustainable (***zero-paper***) and reliable ***prototype*** aerial system development processes.

Key words: MBSE, FTI, zero-paper, prototype

1. Introduction

Model – Based System Engineering (MBSE) methodologies are on the rise as a basis for system engineering activities. The definition of aerial systems can greatly benefit from this approach, evolving towards digitalized processes.

MBSE approaches integrate a set of models as analytical, system architectural, verification, mechanical, electrical, software models, among others. This document addresses the possibility of including Flight Test Instrumentation (FTI) system in the system architectural model in particular. This type of model emphasizes how pieces fit together into a consistent whole and are used to capture functions, behavior, structure, components, objects, information flows, interfaces, interactions and scenarios [1].

During the definition of aerial systems, FTI system are developed in parallel, ensuring that the system can gather all required data during flight tests to successfully reach certification.

FTI systems can benefit of MBSE methodologies in various aspects which will be later described. Nevertheless, the different lifecycle, together with the particular nature of FTI interfaces towards other systems, leads to the necessity of a tailored approach to MBSE to cover prototype development requirements.

1.1. State of the Art

The research referenced in [1] introduces the MBSE development of civil aircraft and then establishes an integrated platform for civil aircraft development based on this process. Moreover, the article referenced in [2] deals with simulation models for MBSE applied to aircraft systems, mostly focusing on the technical discipline of fluid dynamics with embedded hardware and software. It mentions how as the use of MBSE increases, the usage of flight test data for model validation increases. It is also highlighted how the proper definition of flight test installations on aircraft are critical to this duty. Continuing the literature research, [3] approaches the creation of flight test scenarios based on model - based system engineering artifacts represented in System Modelling Language (SysML). In the current literature, there is a lack of concrete studies covering how FTI systems can be introduced into system architectural models from early design stages, benefiting from MBSE methodologies.

1.2. Objectives

The objectives of this paper are to:

- analyze the crucial interfaces between serial aerial system and FTI definition processes,
- evaluate the opportunities and difficulties of applying MBSE methodologies to FTI systems,
- and propose key aspects to be tailored when including FTI in the system architectural model.

1.3. Structure

To meet the previously enumerated objectives, this article is structured as it follows:

- **Model Based System Engineering:** This first introductory chapter summarizes the principles of MBSE. The differences with respect to traditional system engineering approaches are gathered.
- **Flight Test Instrumentation System:** This second introductory chapter describes the basics of a flight test instrumentation system.
- **FTI Specific Development Process:** This chapter defines the how FTI systems are usually defined and developed. Interactions during aerial system and FTI definition processes are analyzed.
- **Opportunities of applying MBSE to FTI:** Based on the previous one, this chapter identifies how MBSE can bring opportunities to enhance FTI definition and development process.
- **Difficulties of applying MBSE to FTI:** This chapter identifies difficulties to be taken into account when tailoring MBSE methodologies to FTI.
- **Prototype and serial system: Variants:** This chapter covers the fact that the architectural model of a prototype evolves in parallel to the serial system one.
- **System architecture model structure:** This chapter addresses FTI presence in the overall model structure.
- **System architecture model views:** The model of an architecture can be approached from different views, which is handled along this chapter.
- **Modelling of internal FTI:** This chapter focusses on a model of the internal FTI architecture.
- **Modelling of FTI interfacing with other systems:** This chapter covers the integration of FTI architecture into an architecture model of the overall aerial system of which FTI is part.
- **Conclusion:** This concluding chapter collects future steps, lessons learnt and overall conclusions of tailoring MBSE methodologies for prototype aerial systems development.

2. Model Based System Engineering

The traditional way of working in systems engineering is document centric: the primary focus is to create and manage various documents such as requirements documents, design specifications, interface control documents, etc. In this approach, the information is primarily conveyed through text, diagrams, tables, and other traditional document formats. The changes and updates of the system are managed by updating the relevant documents, which can sometimes lead to version control challenges and inconsistencies. See Fig. 1.

Fig. 1: Document centric approach schema

MBSE is a model centric approach: the documents are replaced by a model, that graphically represents the system, its components, interactions and requirements. Instead of relying on textual documents, MBSE employs standardized modelling languages like SysML (which stands for System Modelling Language). The model serves as master source of information, enhancing traceability and consistency. See Fig. 2.

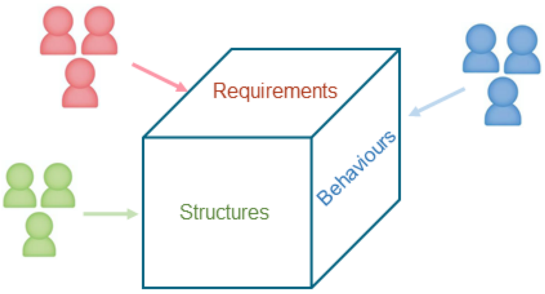

Fig. 2: Model centric approach schema

MBSE is an approach that:

- Improves communication and collaboration across disciplines,
- Improves ability to predict system behavior,
- Ensures design data consistency and completeness,
- Allows to perform cross domain impact analysis of changes,
- Supports interface management, enabling efficient system integration.

The International Council of Systems Engineering (INCOSE) formally defines MBSE as the formalized application of modelling to support system requirements, design, analysis, verification and validation, beginning in the conceptual design phase and continuing throughout development and later life cycle phases.

3. Flight Test Instrumentation System

Main purpose of a FTI system is to gather flight test data - in a safe and secure manner - which are later used to validate and certify the aerial system (e.g., aircraft, Remotely Piloted Aerial System (RPAS), Unmanned Aerial Vehicle (UAV), etc.).

When designing a new aerial system, the prototypes need to be fitted with a considerable amount and variety of instrumentation, whose installation may impact the system design from the very beginning.

As depicted in Fig. 3, FTI systems core functions are aiming to:

- Acquire flight avionics data and aircraft structural/environmental data,
- Record flight test data on-board and on-ground for offline data analysis,
- Encrypt flight test data (if needed),
- Transmit on-board data by telemetry means to FTGS (Flight Test Ground Station) for real-time monitoring during test flights,
- Contribute to execution of specific flight tests by the means of special instrumentation, as for example:
    - Envelope expansion by using flutter excitation system,
    - Real-time monitoring of weapons jettisoning by using video cameras,
    - …

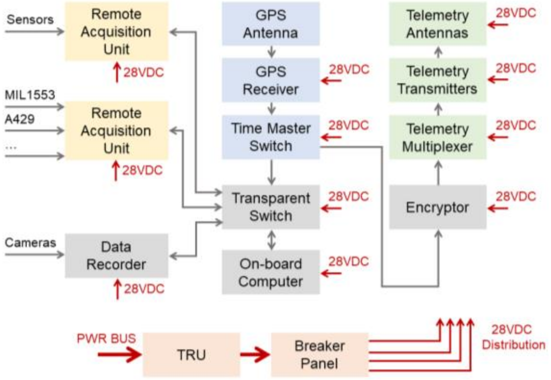

Fig. 3: FTI conceptual block diagram

Additionally to these core functions, FTI system relies on its own support functions which guaranty FTI proper operation:

- Power management and distribution,
- Time synchronization,
- Data management and distribution,
- Health status monitoring.

FTI is usually not considered as a L2 system (see Section 8 for the definition of a L2 system) itself because of its particularities and for being only implemented on prototypes. Therefore, it has a very specific lifecycle which is described in the following section.

4. FTI Specific Development Process

Following conceptual design phase, any aerial system development will follow standard system engineering approach as depicted in Fig. 4 (known as V model). This illustrates main system engineering principles which goal is to deliver the suitable system on-time, minimizing risk, controlling cost and quality along the whole project. It relies on:

- Properly capturing customer needs,
- Defining and validating system functional requirements early in the project,
- Proceeding with system and subsystems design,
- Building the right system and subsystems,

- Performing coherent and exhaustive verification and validation of the whole system.

Fig. 4 points out particularities of FTI system development process, which is characterized by a much more time-constrained lifecycle compared to aerial system development one:

- Even if system testability requirements shall be considered from the beginning, L1/L2 systems requirements toward FTI come late in the development phase as a certain design maturity is usually required to properly assess qualification and certification needs, and therefore the needed flight test instrumentation. Consequently, FTI requirements definition and validation arise once aerial system is already in design phase, requiring many interactions to follow design changes (❶).
- In design phase, FTI can anyhow be impacted by systems late design modifications, originated from implementation/realization issues, implying requirements update and therefore quick FTI system definition update (❷).
- As FTI is mainly based on COTS and well-known equipment, implementation/realization of FTI system is much more agile than for a standard L2 systems.
- Anyhow, FTI verification phase may require involvement of other L2 systems (combined tests ❸). This imposes FTI system to be already available during L2 systems verification test, shortening its own verification phase.
- Same occurs for FTI system validation as FTI is involved in aerial system verification process (❹).
- Finally, as FTI is a key contributor to whole system validation and qualification, it has to be fully validated at this point of the project (❺).

In conclusion, FTI development process is much shorter than aerial system development one, requiring a lot of interactions with various stakeholders, agility and synchronization efforts to guaranty consistency of the end-to-end definition. In this regard, opportunities arise for FTI to benefit from MBSE features, which are gathered in the following Section 5.

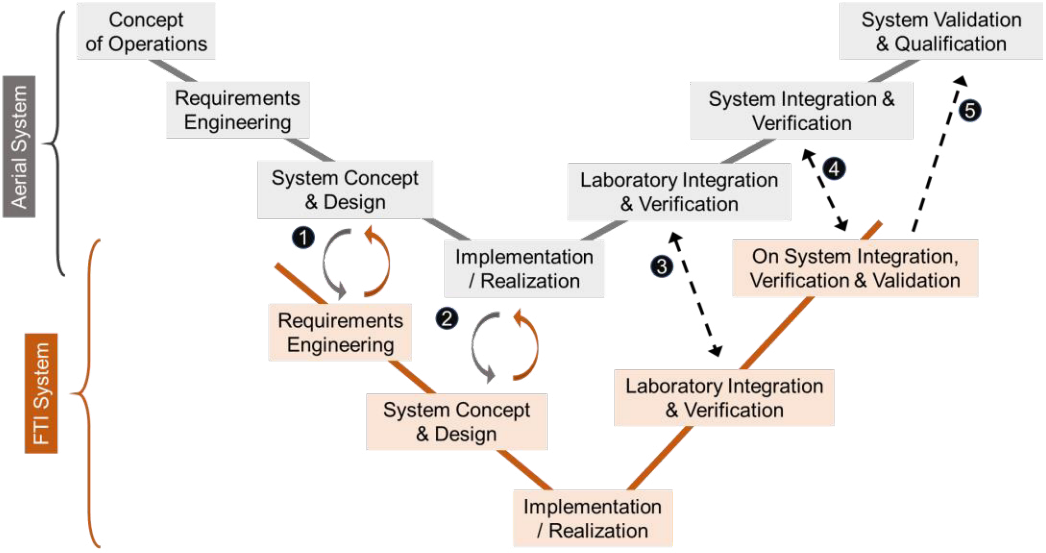

Fig. 4: FTI Development process

5. Opportunities of applying MBSE to FTI

Observing in detail the previously described FTI development process, it is noticeable that there is room for improvement, especially regarding:

- **FTI Visibility:** In a traditional FTI development process, FTI is visible to other systems at late stages of the definition phase.
- **System Visibility:** The visibility of other systems is arduous for FTI as different aspects are defined in a multitude of individual and non-interrelated documents. This implies difficulties in cross – domain

analysis regarding security and safety (among other things) of the architecture as a whole.

- **FTI Integration in the system:** In a traditional FTI development process, FTI is likely to be integrated once the system is defined, which can lead to inconsistencies.
- **FTI Design Review:** In a traditional approach, the verification of the FTI design is done by means of a manual review of the documentation that defines the system.

Tab 1 compiles a comparison between traditional and model – based system engineering regarding the previously enumerated points. Based on that, the opportunities of applying MBSE to FTI development process are extracted and summarized in the right – hand side column.

Tab 1: Opportunities of applying MBSE to FTI

	Traditional SE	Model - based SE	Opportunity
FTI Visibility	Late visibility	Enhanced visibility from early design stage	FTI design can be optimized being part of the system as a whole. The L2 systems can define their architectures considering interfaces with FTI.
System visibility	Multitude of individual and non-interrelated documents	Integrated picture of the whole system, including cross – domain system properties such as safety criticality, vulnerability, etc.	Support cross – domain analysis (safety, security, etc.).
FTI Integration in the system	FTI is integrated post system definition.	Anticipated FTI interfaces definition for further integration within the overall system	Enable earlier definition of interfaces towards FTI. Systems can consider the implication of FTI interconnections and mechanical installations in their design.
FTI Design Review	Manual review of the documentation.	Formal model validation rules	Support automatic consistency checks.

6. Difficulties of applying MBSE to FTI

A change towards MBSE inevitably implies the emergence of difficulties and challenges. The obstacles that may be encountered by the nature of both MBSE and FTI are identified and listed below:

- **Methodology change:** The changes in processes towards digitalization imply high anticipation and cost, time and resource efforts.
- **Robustness vs agility:** On the one hand, MBSE is a robust and formal approach to define a system. On the other hand, the nature of FTI requires agile processes. A trade – off between robustness and agility has to be found.
- **End-to-end visualization:** The end-to-end describes a process that takes a system from the very beginning to the end of its definition. Following MBSE methodologies, the system is defined in an integrated tool chain.

Modelling particularities of intrusive FTI interfaces: FTI has very particular interfaces towards other systems as it needs to gather transparently data within sub-systems interconnection (e.g., avionic bus tapping), without compromising system function. This requires specific and unclassical modelling technics.Tab 2 compiles a comparison between traditional and model – based system engineering regarding the previously enumerated points. Based on that, the difficulties of applying MBSE to FTI development process are extracted and summarized in the right – hand side column.

Tab 2: Difficulties of applying MBSE to FTI

	Traditional SE	Model - based SE	Difficulty
Methodology Change	Well known skills, tools and methodologies	Acquisition of new skills + adaptation of existing tools and methodology or development of new ones	High adaptation and anticipation required, which is translated into cost, time and resource efforts. Risk of not reaching sufficient maturity versus project milestones.
Robustness versus Agility	A robust system engineering methodology can be easily tailored for FTI for increased agility with a minimum of constraints.	Robustness is ensured through well-defined methodology and integrated tool chain. Agility is possible, but limited by a constrained framework.	Anticipation needed to tailor the methodology to be able to cope with agile implementation for FTI. Modelling rules have to be extended and customized beforehand.
End-to-End Process	The end-to-end process does not require the system to be defined in integrated tools.	The end-to-end process requires the system to be defined in an integrated tool chain.	The increased number of tools and the interdependencies between them make difficult the visualization of the end-to-end process and the understanding of the implications of a change along the whole tool chain.
Modelling particularities of intrusive FTI interfaces	Managed directly together with system owner and FTI.	Definition of specific modelling rules and responsibilities are required to be able to manage intrusive FTI interfaces.	Definition of serial and prototype system in the same model makes difficult to segregate both architectures in the case of intrusive FTI interfaces.

7. Prototypes and serial system: Variants

There are two important concepts in product line engineering and configuration management: variants and versions. On the one hand, versions are sequential revisions replacing each other; on the other hand, variants exist in parallel.

A prototype is developed in parallel to the serial system: it can be seen as a variant of the product. Fig. 5 provides a visual representation of this concept.

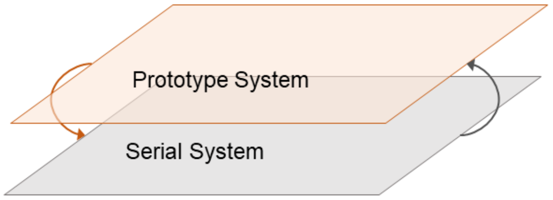

Fig. 5: Prototypes as variants of serial systems

The need for prototypes responds to high-level requirements: the system must have the functionality of being able to be tested. This justifies the existence of the particularities of the prototypes in the model as a concrete variant.

To use the same system architectural model to represent a set of system variants requires a high configuration control effort. As said before, a prototype responds to additional high-level functions than those of serial systems. A prototype evolves in different versions together with the aerial system. As prototypes development occurs in parallel of serial system development, both system architectures must be synchronized so that coherency is kept over whole system definition, including FTI.

8. System architecture model structure

A common approach in the development of complex systems involves conceptually decomposing the system into a multi-level hierarchy of interacting systems [4]. An aerial system could be decomposed in the following layers:

- **L0 System:** System of Systems (SoS). It represents the System of Interest (SoI) in its context. For example, a RPAS embedded in its operational environment. It answers to stakeholder requirements and operational concept analysis. The actors (e.g., Air Traffic Controller (ATC) and external systems (e.g., Ground Positioning System (GPS)) are also defined.

- **L1 Systems:** Systems of Interest (SoI). These are major systems, as for example, the aircraft or the ground control station composing an RPAS.
- **L2 Systems:** Major Subsystems, as for example the electrical system of an aircraft. The definition of these systems answers to requirements that emerge during preliminary design phases.
- **L3 Systems:** Equipment, composing a L2 System. For instance, a battery of the electrical system of an aircraft.

In the system's hierarchical decomposition, FTI can be located at L2 level, but it is only part of a concrete variant of the system: the prototype.

Observing the prototype as a variant of the serial system as presented in Section 7, there are implications at all layers of the system (L0, L1, L2). Fig. 6 depicts graphically how the context of a prototype differs from the one of the serial systems. Note how there are also specific Flight Test (FT) actors and external systems that will be interacting with the FTI, as for example Flight Test Engineers (FTE) or the Flight Test Ground Station (FTGS).

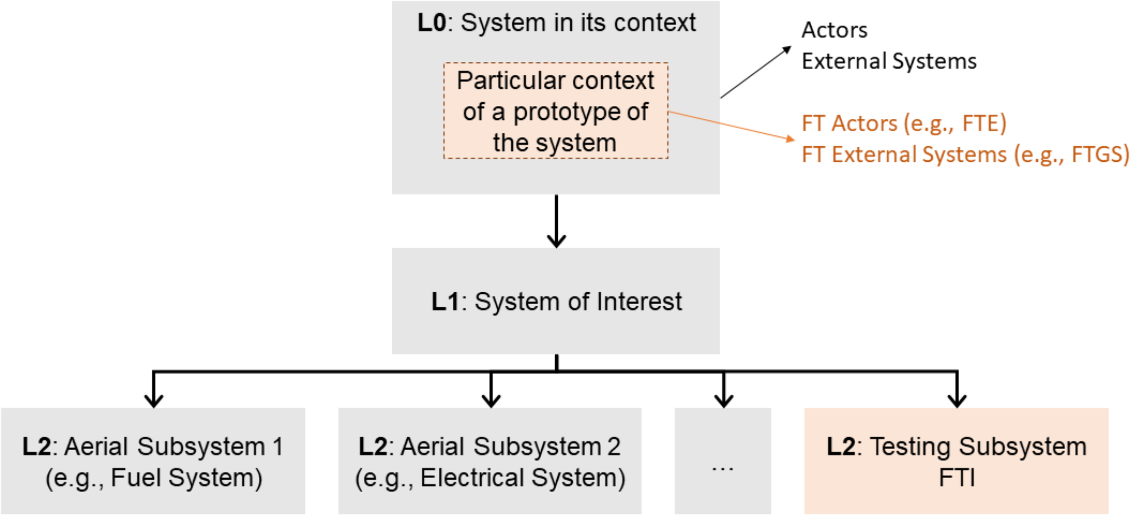

Fig. 6: FTI in the hierarchical structure of the system

9. System architecture model views

In usual MBSE methodologies to define a system architecture, the model can be observed from different views:

- **Requirements view:** A system can be observed as a set of requirements that, by *well-formed textual "shall" statements […] communicate in a structured, natural language what an entity must do […]* [5].
- **Functional view:** A system can be observed by a set of interrelated system functions, defining the capabilities, the services, or the functions provided by the system.
- **Logical view:** A system can be observed by a set of interrelated logical systems that compose the logical architecture, which consists in *decomposing and partitioning the system into logical elements. […] The elements interact to satisfy system requirements and capture system functionality. Having a logical architecture mitigates the impact of requirements and technology changes on system design* [6].
- **Physical view:** A system can be observed by allocated and interconnected hardware and software resources.

Fig. 7 shows how the previously defined views are traced and linked one to each other [7] [8]. System functions are defined **satisfying** functional requirements. These are **allocated** to logical systems. And these latter systems are an **abstraction** of the physical systems, that are defined and interrelated in the physical view. This tracing enhances impact analysis; for example, if a system is replaced by another, one can traced back, knowing which logical system, system function and requirement may be affected by the change.

This information is of origin Airbus Defense and Space/Spain and does not contain any export controlled information
Airbus Amber releasable to ETTC
ETTC 2024– European Test & Telemetry Conference

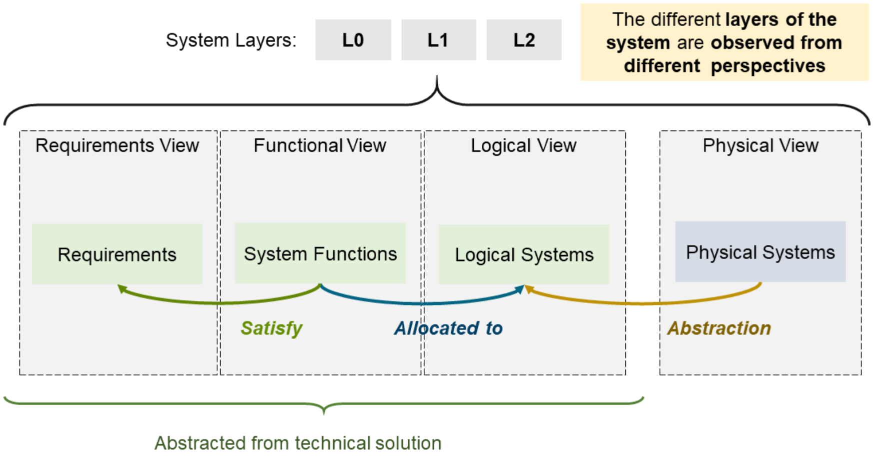

Fig. 7: Model views observed from model layers

10. Modelling of internal FTI

The layers of the system described in Section 8 can be defined through the different perspectives gathered in Section 9, and so can FTI also. Fig. 8 shows how FTI requirements can be traced down to FTI system functions using *satisfaction* links in the system architecture model.

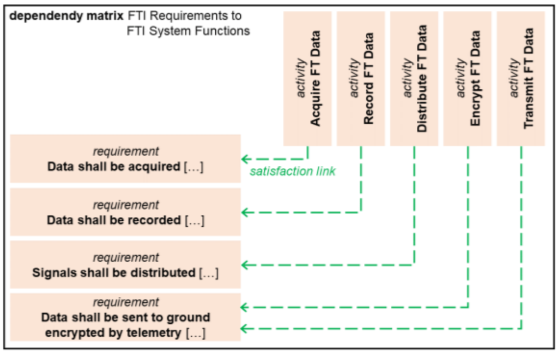

Fig. 8: System functions satisfy system requirements

Then, Fig. 9 depicts the modelling of the relationships between these system functions, defining the functional inputs and outputs of each function. The system functions are then allocated to logical system defining a logical architecture abstracted from the technical one. The black border shows the functional boundary of the system function "Provide *FT Data to Ground*", which is higher level than the ones represented inside. The lines that cross it represent functional flows coming/going from/to other higher level system functions outputs/inputs, or from external systems or actors.

To refine the logical architecture detailing the technological solution, concrete physical systems are defined realizing the logical ones, as shown in Fig. 10. Note that this tracing is done at all levels: L2 logical systems as an abstraction of L2 physical systems, and analogously to L3 logical and physical systems.

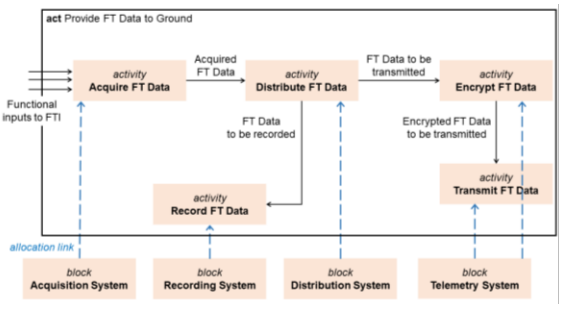

Fig. 9: Logical systems allocated to system functions

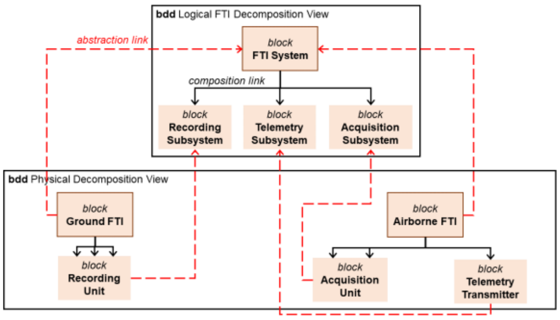

Fig. 10: Logical systems are an abstraction of physical systems

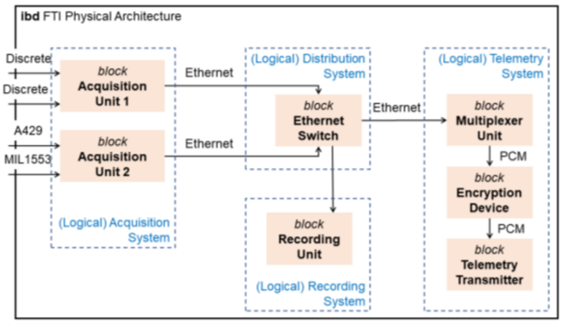

Fig. 11: FTI internal physical architecture

Once the physical systems are defined and traced up to the logical ones, the connectivity between them can be shown (see Fig. 11). It is depicted again how a group of physical systems constitute the realization of a logical one, which in turn has specific system functions allocated that satisfy concrete functional requirements. The black box represents the boundary of FTI physical architecture. The lines that cross it represent physical inputs/outputs coming/going from/to other systems, external systems or actors.

11. Modelling of FTI interfacing with other systems

As shown in the previous section, the FTI architecture can be designed following an MBSE methodology. Nonetheless, it important to keep in mind that the real benefit is to integrate FTI architecture into an architecture model of the overall aerial system of which FTI is part.

Fig. 12 illustrates the fact that the FTI architecture implemented in the prototypes is defined on top of the serial system architecture. change.

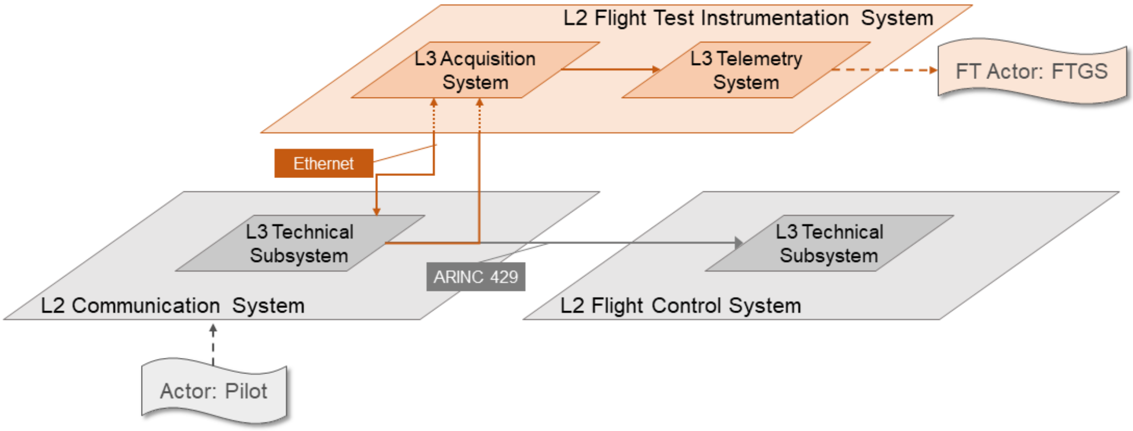

Fig. 12: FTI architecture on top of serial aircraft architecture

The same procedure as the one followed in Section 10 is adopted at the level of the complete aerial system. Fig. 13 shows a diagram analogue to the one in Fig. 10, but corresponding to the physical decomposition of the whole aerial system. In this example, also communication and flight control systems are shown in the composition tree.

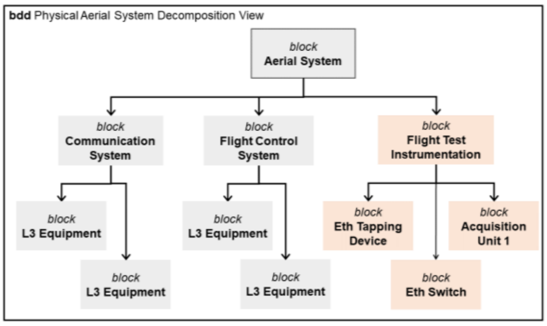

Fig. 13: Physical system decomposition view

The interconnections of FTI with other systems can be modelled: this is one of the key aspects from which FTI can benefit, since the communication and collaboration across disciplines is enhanced. Fig. 14 shows a diagram analogue to the one in Fig. 11, but one level of decomposition above. In this case, the black border depicts the boundary of the whole aerial system (instead of the one of FTI), so the lines that cross it come/go from/to external systems or actors (e.g., the Air Traffic Controller (ATC) or the FTE). The blocks inside the diagram are systems at the same decomposition layer of FTI (e.g., Communication System, Flight Control System (FCS), etc.).

Note the different types of interconnections shown in Fig. 14:

- The connection ❶ between FTI and FCS is “direct”. FCS actively provides data to FTI through that connection. In the serial system, that connection would simply not exist.
- The connection ❷ between FTI and FCS represents the splice of a harness. FTI sniffs the information that flows between the two L3 FCS equipment. In the serial system, that connection would disappear, but the connection between FCS equipment remains.
- In the connection ❸ between FTI and Communication System an intermediate equipment is installed to sniff the data

flowing between the L3 Communication System equipment. Note that the red (FTI) and black (serial system) connections are mutually exclusive: as in the previous connections types, the red one does not exist in the serial system. But the black also has to be removed in the prototype architecture.

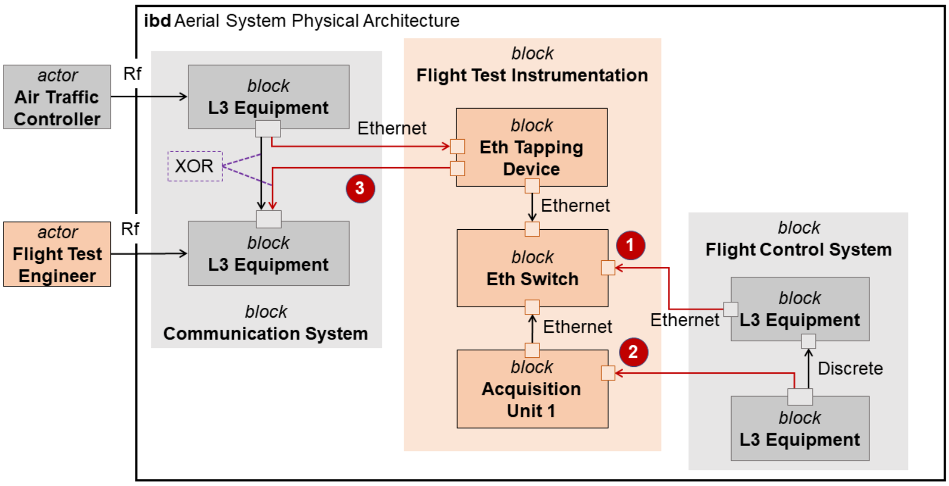

Fig. 14: Technical FTI interfaces towards other systems

Summing up, it is important to keep the serial and prototype architectures well decoupled.

12. Conclusions

Model - based systems engineering methodologies bring with them the **need to adapt to new ways of working**. It is necessary to familiarize oneself with the characteristics of the methodology, **understanding the opportunities (and challenges)** that it brings to the definition of complex systems.

As a FTI engineer, you need to make a deep study of the gaps that you will encounter when trying to bring your system definition into a model - based environment. It is a good idea to think deeply what parts of FTI definition can benefit from this process, which are, for example, the interconnections towards other systems and the repetitive part of the architecture (excluding sensors).

As a model - based system engineer, understand how different FTI lifecycle, as well as its operational concept. For this reason, FTI requires a tailored MBSE approach that enables agility in processes and tools. This has to be reflected at all levels of the system. Moreover, at a physical layer, there are very “FTI specific” types of interconnections with other systems. Awareness of specific processes, methods and tools towards prototypes particular needs and development lifecycle has to be achieved from the very beginning of the project

To conclude, this paper addresses just one piece of the complete MBSE environment: the system architectural model. Further assets are required for the complete definition of the FTI, from detailed measurement chain definition, pin-to-pin design, electrical wiring establishment, etc. All these elements have to be taken into account for the realization of the system as a whole. This will be subject to further publications.

References

[1] L. Hart, “Introduction to Model-Based Systems Engineering (MBSE) and SysML”, in Delaware Valley INCOSE Chapter Meeting, Delaware Valley, USA, 30th July 2015

[2] W. Bi, W. Wang, A. Zang and Y. Wang, “Extending Model-Based Systems Engineering into Integrated Platform Designed for Civil Aircraft Development”, 8th August 2021; doi: 10.1590/jatm.v13.1218.2021

[3] J. Alvarado and T. Bradley, “Developing Model-Based Flight Test Scenarios”, *Journal of Test and Evaluation*, vol. 44, nº 4, December 2023

[4] N. Chiriac, K. Hölttä-Ottom D. Lysy and E. Suk Suh, “Three Approaches to Complex System Decomposition”, in 13th International Dependency and Structure Modelling Conference, Cambridge, Massachusetts, USA, September 2011.

[5] INCOSE, “Guide to Writing Requirements”, 2022.

[6] INCOSE, “INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities”. Wiley, 2015.

[7] Magic Draw, Accesed on May 18th 2024 [Online], Available: https://docs.nomagic.com/plugins/viewsource/viewpagesrc.action?pageId=36311025

[8] “What is MBSE? Introduction to Model-Based Systems Engineering”. Capella | Open Source MBSE Tool. Accessed on May 18th 2024. [Online] Available: https://mbse-capella.org/what_is_mbse.html