



Large legacy systems design maintainability through modeling

Mohamed Habib Essoussi, Paul Vivot, Lucas Deloye, Davi Henrique De Sousa Pinto

► To cite this version:

Mohamed Habib Essoussi, Paul Vivot, Lucas Deloye, Davi Henrique De Sousa Pinto. Large legacy systems design maintainability through modeling. ERTS 2024, Jun 2024, Toulouse, France. <hal-04614391>

HAL Id: hal-04614391

<https://hal.science/hal-04614391v1>

Submitted on 17 Jun 2024

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

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



HAL Authorization

Large legacy systems design maintainability through modeling

MOHAMED HABIB ESSOUSSI

Airbus Operations S.A.S
+33 6 08312231

mohamed-habib.essoussi@airbus.com

Paul VIVOT

Airbus Operations S.A.S
+33 6 07601685

paul.vivot@airbus.com

Lucas DELOYE

SII
+33 5 34615959

lucas.deloye.external@airbus.com

Davi Henrique de Sousa Pinto

Airbus Operations GmbH
+49 15111097167

davi-henrique.sousa-pinto@airbus.com

Abstract. Model-Based System Engineering (*MBSE*) and particularly Model-Based Product Line Engineering (*MBPLE*) now stands as the new standard for systems engineering at Airbus Group. Indeed, the Airbus *MBSE* Architecture Framework (*R-MOFLT*) and its feature-based product line engineering framework extension (*MBPLE4MOFLT*) are widely deployed on Research & Technology projects. This paper tackles the applicability of such enablers to large legacy systems. As such, it outlines a proof of concept on redesigning a legacy system using *MBPLE4MOFLT* as a new product line based on several in-service variants definitions that have been designed over the last four decades following document-based ways of working. As such, the interoperability between these ways of working and the new digital assets is essential to achieve this migration on one side and, once migrated, to ensure backwards compatibility with the official process, on the other side. To this aim, besides using existing data hubs between *Cameo Systems Modeler* and *Rational DOORS*, the Airbus *MBSE SysML* profile has been extended with further customizations to fit the new product line design golden rules. Wizards are also proposed to ease authoring and impact analysis. Finally, a new plugin has been developed to automate the variability propagation throughout variable assets and to ensure consistency between the variability handled with *MBPLE4MOFLT* and the requirements applicabilities handled in *Rational DOORS*.

Keywords. MBSE, MBPLE, system engineering, industrial application, avionics.

1. Introduction

Recently, Airbus has launched a new Flight Management System (nFMS) product line to

capitalize on four decades of designs and improvements across Airbus aircraft families.

This product line aims to reduce non-recurring costs and time to market while preparing for the future. With *MBSE* being already deployed in the "Zero Emission" (ZEROe) Demonstrator[2][5], the question was to assess the methodology maturity against the new product line needs. It is crucial in our approach to underscore that unlike ZEROe, nFMS does not commence from a clean state, but rather inherits a substantial legacy. As such, a proof of concept has been conducted, prior to an eventual industrial deployment as an enabler of this development. The proof of concept also questioned the actual need of *MBSE*. It explored how *MBSE* can assist system designers who may not be familiar with it in their day-to-day tasks in the context of a high-pressure delivery rate. The objective was to maintain interoperability with the existing process while preserving the current format for sharing artifacts with suppliers.

Consequently, the primary focus shifted swiftly to the challenge of effectively managing a large volume of textual requirements. The emphasis was on organizing them within a robust specification, prioritizing solid variability management over modeling the system behavior. Model execution hence falls out of the scope.

Nevertheless, we explored enhanced collaborative work methods, digital continuity, document generation, and impact analysis through *MBSE*. This paper delves into these aspects, demonstrating *MBSE*'s role as a key enabler in large legacy systems, using an industrial use case.

The rest of the paper is organized as such: section 2 presents the methodology, from the state of the art down to the tailoring and customizations. Section 3

outlines the current process of designing a flight management system at Airbus. Section 4 is the proposed approach within the framework of the project assumptions. Section 5 exposes the first results with benefits, challenges and limitations then concludes.

2. Methodology

INCOSE's visions for 2020[12] and 2025[13] correctly predicted and anticipated the widespread adoption of *MBSE* as a de-facto way of deploying Systems Engineering for complex systems.

Following the trend, the industry, academia and the systems engineering community have at their disposal several comprehensive frameworks, such as DoDAF¹, MoDAF², UAF³, Arcadia[14]. These frameworks are continuously improved, matured and tailored to achieve business and technical needs.

Airbus MBSE Framework. Airbus' own *R-MOFLT* Architecture Framework[3] has been developed to tackle the organization's challenges regarding SE and *MBSE*. It is based on the SysML language[15] and expands and combines existing frameworks to support Airbus' products, services and industrial systems.

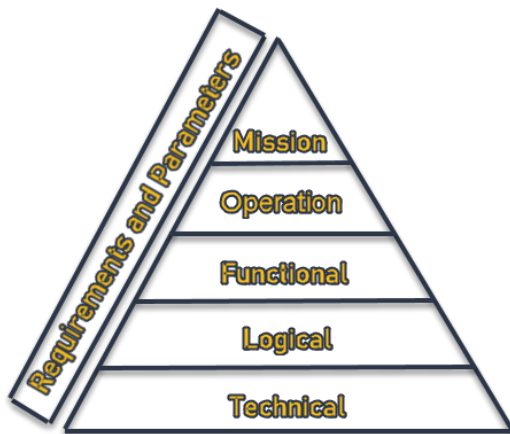


Figure 1: R-MOFLT Architecture Framework Scheme

Airbus MBPLE Framework. As part of providing an integrated solution, an *MBPLE*[1][4] approach based on the principles described in the ISO 26580[18] is being developed. It currently supports variability on architecture and requirements, but its ambition is to

provide holistic product line approach capabilities, by expanding existing digital assets to include variation points, while maintaining a single source of variability, through feature models. For the project described in this paper, two methods from the *MBPLE* framework have been used and tailored:

- **MBPLE4MOFLT.** Expansion of the *MOFLT* framework. It includes variability and is used to represent a product line system architecture (or 150% models). It includes a PLE configurator capable of generating product line variants (or 100% models).

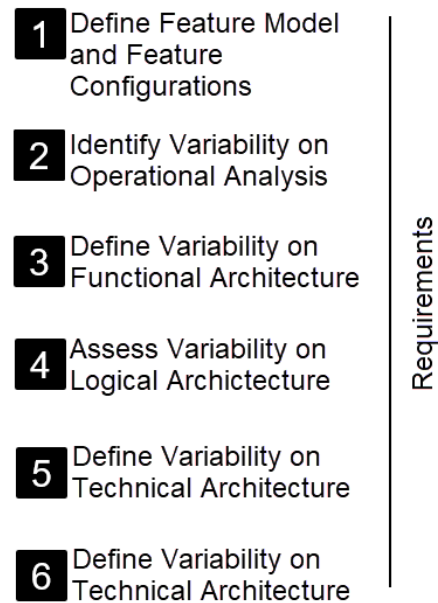


Figure 2: MBPLE4MOFLT scheme

- **MBPLE for Requirements and V&V Management (MBPLE4RVVM).** Expansion of the model-based requirements management methods. It includes existence (meaning a textual exists or not for a certain variant configuration) and parametric variability to textual requirements. This method is integrated and meant to be used with *MBPLE4MOFLT*.

Tailoring. The digitalisation and transition to a model based approach is an even bigger challenge in large and complex organizations such as Airbus. This puts a big emphasis not only on the framework itself, but also on organizational and transitional aspects[17]. Programmes have been running for decades and plenty of data is still stored in older means (hardware and/or software).

In this context, automations and tailoring were developed to interface with “non model-based”

¹ Department of Defence Architecture Framework (U.S. Department of Defence 2010)

² Ministry of Defence Architecture Framework (U.K. Ministry of Defence 2012)

³ Unified Architecture Framework (OMG 2020)

documents and data formats still required by some processes. These automations are better described in Section 4 - Proposed Approach.

3. Analysis of current practices

Within the Airbus framework, the newly developed nFMS adheres to the well-structured Airbus process titled "Define System Requirements and Design." This procedural guideline outlines a comprehensive set of deliverables, encompassing not only the development and process assurance plan but also a collection of artifacts that intricately delineate the system requirements, design specifications, and interface parameters.

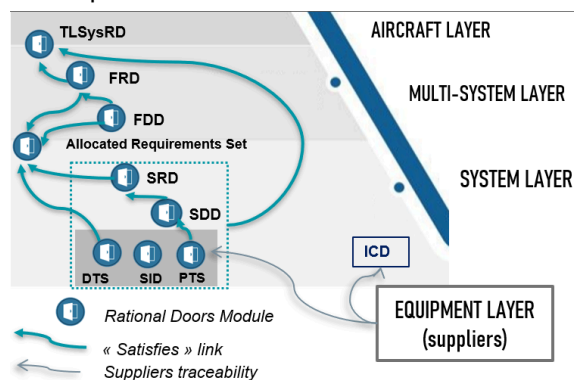


Figure 3: nFMS RDBS

Illustrated in Figure 3, these artifacts are systematically arranged within a Requirements Documents Breakdown Structure (RDBS), with the Detailed Technical Specification (DTS) and the System Interface Documents (SID) serving as the central focal points of this organizational framework at system layer. Requirements within the DTS and SID satisfy upper-level requirements cascaded from the multi-systems function layer such as the Function Definition Documents (FDD) and the Top-Level System Requirements Document (TL SysRD). Subsequently, the requirements embedded in the DTS and SID are disseminated to suppliers, accompanied by the Purchase Technical Specification (PTS) delineating the definition of equipment, encompassing both software and hardware. The seamless interconnection and cascade of these design artifacts is facilitated through the utilization of *Rational DOORS*[22]. Beyond the functional requirements, the system design incorporates the interface definition, meticulously documented within an Interface Control Document (ICD[9]) facilitated by the Albatros⁴ tool. This document comprehensively outlines the

physical characteristics of network messages, digital inputs and outputs labels and buses, as well as analog and discrete inputs and outputs signals.

A critical challenge arises due to the absence of digital continuity among the DTS, the SID, and the ICD. This lack of integration complicates impact analysis processes, introducing inefficiencies and time-consuming evaluations of potential design evolutions consequences.

Moreover, the current methodology falls short in addressing essential product line considerations, specifically in defining variability at the requirement level. This limitation poses a significant hurdle in achieving a comprehensive understanding of the system's adaptability to diverse configurations.

To overcome these deficiencies, the design office has developed ad-hoc golden rules tailored to nFMS as a multi-program and a multi-supplier product line, where the requirements applicabilities are set with the targeted configurations (Airbus aircraft families) and the stakeholders with the suppliers as depicted in the template below:

DTS FMS-xxx-xxx-xx-xx
<Requirement Statement>
Rationale: <rationale of the requirement>
Additional Information: <any additional info to support the understanding of the requirement>
Applicability: <aircraft configurations>
Change Rationale: <change rationale with regards to existing products>
Source: <pre-nFMS source>
Stakeholders: <list of suppliers>
Working Status: <working status>

Before integrating updates into the baseline DTS, the design office initiates a Specification Evolution Request (SER) outlining the changes towards the suppliers. This delta is then assessed for cost implications. Upon agreement between the involved parties, the delta is manually incorporated into the complete DTS.

4. Proposed approach

This section provides a comprehensive exploration of the pragmatic solutions implemented to address the design definition and maintainability challenges outlined in Section 3, employing *MBPLE4MOFLT*. Emphasis is placed on the significance of ad-hoc customizations tailored to deliver substantial value to the project.

a. Project assumptions

The objective of this proof of concept is to showcase the effectiveness of improving the authoring process for DTS and SIDs requirements. The primary focus is on establishing robust linkages between these

⁴ The official interface management tool at Airbus

requirements, facilitating smooth impact analysis. Furthermore, the goal is to forge connections with ICDs parameters, thereby establishing a digital continuity that extends from the functional and interface requirements to the intricate details of physical connectors within avionics computers backplanes.

As depicted in figure 4, to accompany the change smoothly, bridges and hubs (supporting the exchanges depicted as red arrows) are opted over disruptiveness so that interoperability with the current process is maintained. A primary motivation behind the adoption of this new approach is to ensure user-friendliness for designers unfamiliar with *MBSE*, aiming to seamlessly integrate with existing workflows for authoring whilst taking advantage of new automated time-saving impact analysis features. *MBSE* serves in our context as a facilitator for elevated requirement-based engineering.

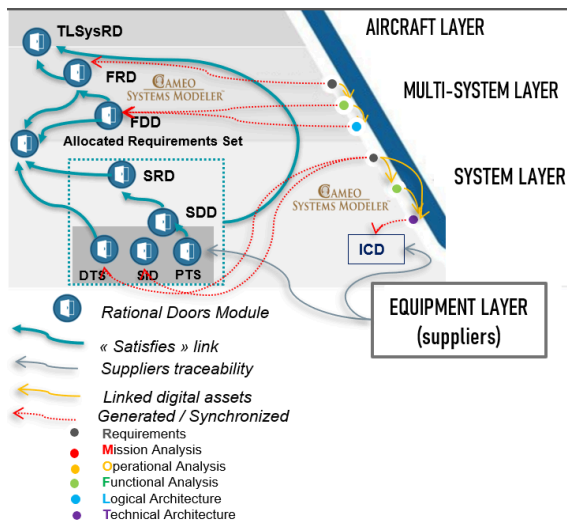


Figure 4: nFMS RDBS with R-MOFLT

b. Product Line Engineering

The management of the product line is facilitated through a rigorous framework comprising a feature model and a feature configuration. The feature model serves to formalize the pivotal factors influencing variability, delineating the software's adaptability. As described in Figure 5, these factors include specific features per supplier, cockpit HMI type, and aircraft type, thus ensuring both software adaptability and portability. The feature configuration, on the other hand, is determined by a triplet encompassing aircraft type (such as A320, A330, A350), aircraft variants (such as Future architecture or Legacy), and suppliers.

Legend									
Select									
ConfigurationSpace		27	27	28	28	28	28	28	28
nFMS Product Line Feature Model									
F0000_nFMS_Product_Line		✓	✓	✓	✓	✓	✓	✓	✓
F0001_Hardware_Adaptability		✓	✓	✓	✓	✓	✓	✓	✓
F0010_FMC		✓	✓	✓	✓	✓	✓	✓	✓
F0011_FMGECC		✓	✓	✓	✓	✓	✓	✓	✓
F0023_HMI		✓	✓	✓	✓	✓	✓	✓	✓
F0230_MFD		✓	✓	✓	✓	✓	✓	✓	✓
F0231_MCDU		✓	✓	✓	✓	✓	✓	✓	✓
F0022_Suppliers_Specificities		✓	✓	✓	✓	✓	✓	✓	✓
F0220_SupplierB		✓	✓	✓	✓	✓	✓	✓	✓
F0221_SupplierA		✓	✓	✓	✓	✓	✓	✓	✓

Figure 5: Extract from nFMS Feature and Configuration matrix

This structured approach enables the transformation of the model from 150% to 100% based on a predefined set of configurations. The 150% model is founded on the principles of clean and direct methodologies, as elaborated upon in greater detail in section 4.c. Moreover, the product line exhibits robustness against configuration evolution, adeptly accommodating changes such as supplier product evolution or the introduction of new aircraft models. Once both feature model and configuration are established, features are articulated as elementary constraints. To mitigate workload and minimize errors, features are categorized at function level and subsequently propagated to requirements, coupled with validation rules. To ensure interoperability with *Rational DOORS*, the configuration of aircraft and supplier specifications is systematically calculated at the requirement level, with the resulting data automatically integrated into the applicability and stakeholders attributes. This process is facilitated through a specialized plugin and additional customizations, which encompass derived properties implemented by opaque behaviors. The overarching objective of this automated tool chain is to forge a cohesive connection between *MBPLE*, leveraging digital assets as the definitive source of truth, and traditional Product Line Engineering (PLE) reliant on textual attributes within *Rational DOORS*.

c. Architecture

Modeling the functional breakdown through SysML Block Definition Diagrams (BDD) and system architecture via SysML Internal Block Diagrams (IBD) presented a strategic opportunity to mitigate complexity in the context of legacy large systems. The approach involved conceiving this extensive system as a hierarchical arrangement of functions, transcending conventional document-based representations. This method facilitated a holistic comprehension of the interconnections among functions and the corresponding functional flows, exemplified by the Airbus MBSE SysML Profile's extension of SysML proxy ports stereotypes.

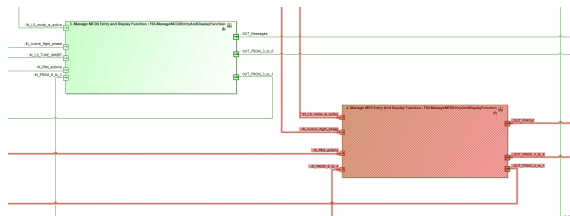


Figure 6: Extract of nFMS 150% functional architecture with a model transformation preview for a particular variant with Pure::Variants[19]

The functional analysis adhered to the direct method (150%, c.f., figure 6), as advocated in [2]. The resultant model enabled the establishment of hyperlinks from specific parts of requirements' textual statements to functional flows. This serves a dual purpose: validating requirement allocations for enhanced organizational structure and enabling metachain navigation essential for impact analysis, thereby enhancing design maintenance (c.f., 4.b). Figure 7 illustrates the process wherein functional flows, originally designated as generic and referenced in the cross-programs DTS requirements, undergo specialization to manifest as specific parameters tailored for each aircraft within the ICD[9]. In this context, the functional interface, initially representing a generic parameter, extends specific functional interfaces defined as variant assets. This relationship ensures that the former strictly inherits the variant functional flow properties of the latter.

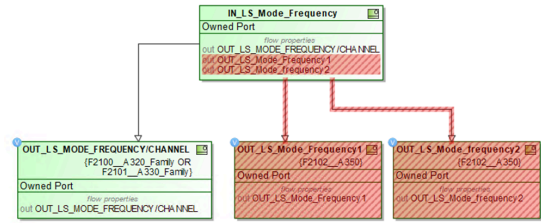


Figure 7: Example of functional flow with associated ICD specific parameters with a model transformation preview for a particular variant with Pure::Variants

As exemplified in figures 8 and 9, these ICD parameters are conveyed then through nested ports which found representation in technical interfaces—conforming to the Airbus MBSE SysML Profile's extension of SysML proxy ports stereotypes—defined within the technical architectures.

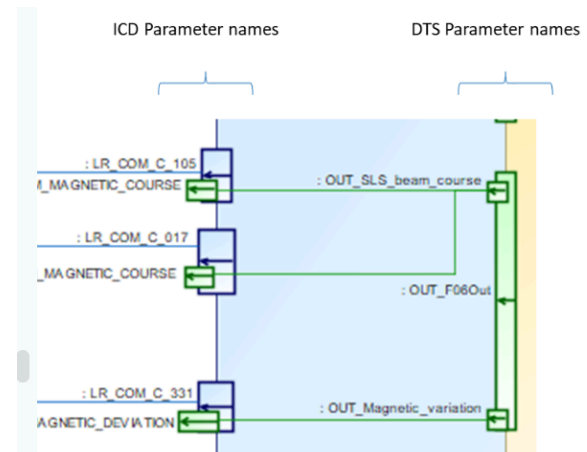


Figure 8: Example of two ICD parameters delegated to one DTS generic parameters and realized by two different digital (A429[10]) labels

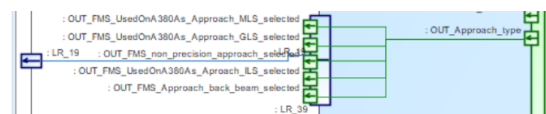


Figure 9: Example of 6 ICD parameters delegated to one DTS generic parameters and realized by one Functional Data Set (FDS) of an AFDX[11] message (avionics network)

Each variant followed the clean method[2]. Its technical interfaces, encompassing system inputs like physical connectors, digital labels, buses, network message structures, ports, and physical links, formed a crucial connection between the Functional (F-layer) and Technical (T-layer) components. This linkage supported digital continuity, allowing for bidirectional impact analysis. Figure 10 shows an example of a specific functional

flow (ICD parameter “LS_MODE_FREQUENCY/CHANNEL”) conveyed by a digital label (A429 label “033”) and transferred on a data bus which is connected to FMGC (Flight Management and Guidance Computer) backplane pins number “LM10A/10B”.

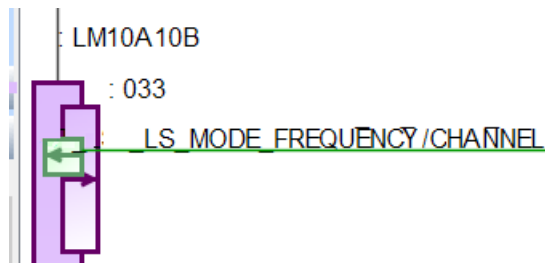


Figure 10: Example of an ICD parameter in F-Layer and T-Layer

d. Enhanced Requirement-based Engineering

Our approach is adaptable to legacy workflows, encompassing existing processes and exchanges with suppliers. Its primary objectives are to enhance engineers' comprehension and to support the 150% specification completeness.

Requirement authoring. The requirements authoring is supported by a wizard to streamline the automatic setup of attributes such as requirement labels (unique identifiers) and versions, ensuring adherence to the nFMS golden rules as sketched in the template defined in section 3. Engineers can focus on product design rather than formatting the specification. Allocation of requirements to functions is achieved through dependency links within traceability matrices. This linkage enables the propagation of function variation points, facilitating the automatic computation of applicability and stakeholder attributes (c.f. 4.b). Subsequently, validation rules are applied to ensure that requirements are defined at the appropriate level, verifying that referenced generic functional flows (if any) are indeed utilized by the function to which the requirement is allocated. With this method, we intend achieve a well organized DTS

The maintenance of the requirements set adheres to the SER process outlined in Section 3, facilitated efficiently through the use of branches. The existing baseline resides within the trunk, with each branch dedicated to a distinct SER. Upon validation of functional evolution with suppliers, the branch is merged into the trunk, establishing a new baseline. Only potential conflicts necessitate the expertise of a systems engineer; all other components can be integrated without any loss of information. While

Rational DOORS remains integral to Airbus's operational methodology, the baseline is also shared within its framework. In a branch, the requirement label is constructed in the creation wizard by taking the branch name and associating it with a unique number per branch to identify new requirements. Once merged into the trunk, another wizard is launched on the requirement set to define the final label of created requirements. Each label is built by retrieving the identifier of the section in which the requirement resides, providing a unique number within the section, and retrieving the baseline number of the trunk in *Teamwork Cloud* to construct the version. Branches serve not only to highlight SER disparities with its original baseline but also enable comparison with the current iteration, ensuring comprehensive tracking of system evolution.

Impact analysis. Beyond merely linking requirements across diverse documents as discussed in Chapter 3, our approach involves referencing model elements such as generic functional flows, specific functional flows or technical flows (c.f. 4.c). By leveraging HTML hyperlinks embedded within requirement attributes such as statements, additional information or rationale, facilitating the creation of a relational map spanning different layers of the system architecture. Integration of such hyperlinks in a requirement statement is illustrated in Figure 11. In this scenario, both the “SLS is active” (Satellite-based Landing System), and the “FAS data block” (Final Approach Segment) functional flow interface types are referenced.

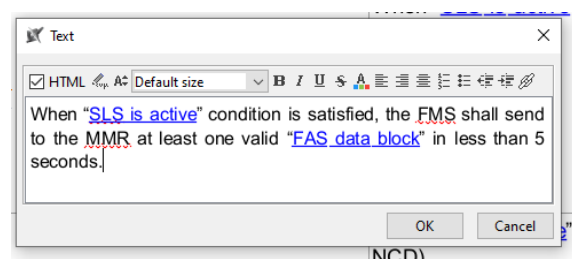


Figure 11: Requirement statement with hyperlinks

Through the utilization of MOFLT profile extensions and opaque behaviors, it is feasible to define a forward impact analysis, as illustrated in Figure 12. This analysis bridges the gap between requirements and functional flow interface types, along with their technical implementations. Notably, only the FAS data block is detailed in technical components as it is an ICD parameter (c.f. 4.c), while the “SLS is active”

Boolean is a generic specification-defined element whose technical definition remains undisclosed to Airbus.

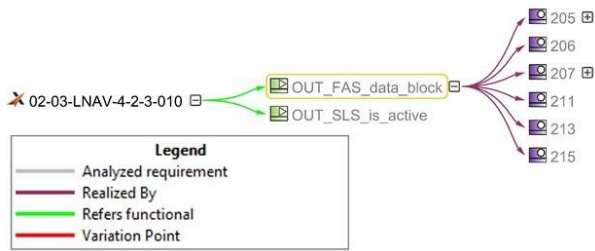


Figure 12: Example of a forward impact analysis (from a requirement)

Figure 13 demonstrates the capability to conduct a backward impact analysis, employing specific functional or technical flow in a manner akin to the forward analysis. For instance, all requirements affected by a system updating one of its interfaces with the FMS are enumerated.

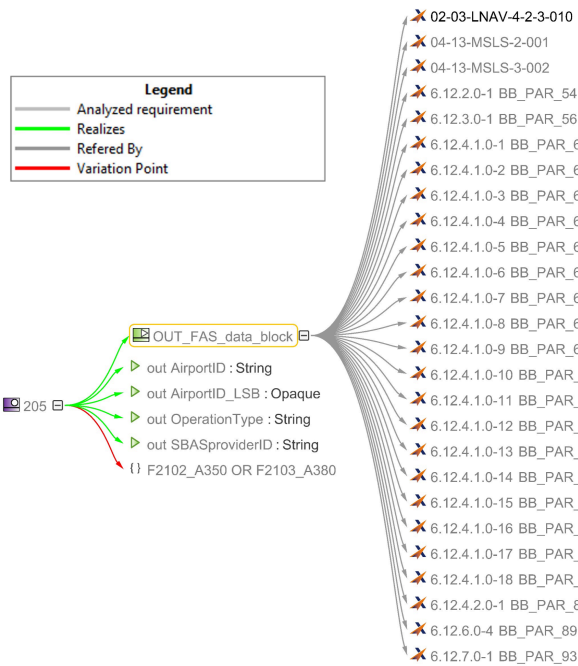


Figure 13: Example of a backward impact analysis (from a technical interface type)

Both analyses are initiated via a customized wizard, wherein the references to SysML and MOFLT are abstracted away from the user, ensuring a streamlined interface. For this purpose, realization and reference attributes are established, employing the MOFLT method, to link model elements between them by extending their profiles.

Document generation. Upon the creation of a new baseline, it is generated and shared with suppliers

after the model undergoes transformation into a 100% model (c.f. 4.c) to uphold confidentiality between parties.

Using *MBSELab@Cameo*, the methodology adopts an ontology-based[6][7][8] approach to maximize agnosticism towards the *MOFLT* framework. Indeed, in our approach, we purposefully separate the functional architecture hierarchy from the DTS hierarchy to minimize disruptions to the existing DTS and to avoid influencing the architecture retro-engineering process based on an unconventional document hierarchy. Within this ontology the DTS abstract elements such requirements, sections, chapters, free text and referenced interface data are defined alongside their attributes (c.f 3) and their relationships. They are subsequently mapped to stereotypes and organized according to the document pattern. Figure 14 displays a combination of ontology elements (green elements) and their corresponding mappings (purple elements). Certain attributes, such as Name and Description, are already established within a higher-level ontology and can be directly utilized in the document pattern.

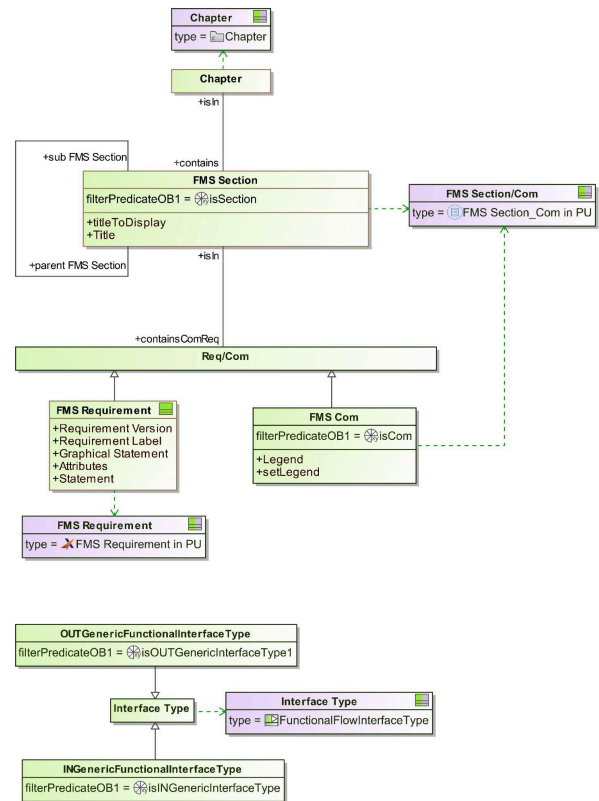


Figure 14: nFMS ontology and mapping diagram

MBSELab@Cameo's enablers facilitate the DTS generation as a henceforth ontology-based

document. Emphasizing PLE, they enable thanks to our ontology the automatic generation of mappings between generic functional interfaces and their specific variants, presented in a comprehensive matrix within the DTS appendix (see section 4.c). These enablers also allow the extraction of model diagrams for visually specific requirements and configuration management data from the *Teamwork Cloud* server. Indeed, in legacy documents, the initial pages are dedicated to cataloging the integrated components within the baseline and providing a brief history of the documents. Utilizing APIs, the baseline numbers, release dates and commit comments are seamlessly retrieved and incorporated into the generated document.

e. Tools

Having a methodology (tailoring of *MBPLE4MOFLT*) and a language (*SysML*) defined, one main challenge was to develop a toolchain able to answer all the needs while being interoperable with already existing tools used for other layers (equipment and multi-systems ones) and being user-friendly to ensure quick and efficient handling by system designers with no specific modeling background. To do so, some of the tools solutions promoted by Airbus Framework have been adopted. All the modeling activities are performed in *Cameo Systems Modeler 21x[20]*, including requirements authoring. Top-level requirements coming from the multi-systems layer and defined in *Rational DOORS* are imported in *Cameo Systems Modeler* thanks to *Cameo DataHub[21]*. Traceability between nFMS requirements and top-level one is performed directly in *Cameo Systems Modeler*. However, as *Rational DOORS* is the official database storing all requirements in Airbus, when a DTS baseline is frozen in *Cameo System Modeler*, requirements and their traceability links towards top-level requirements are exported in *Rational DOORS* through *Cameo DataHub*. Nevertheless, *Cameo Systems Modeler* remains the only source of truth for FMS requirements.

Collaboration between designers is ensured through *Teamwork Cloud*, allowing them to work on common shared models.

For the product line aspect, *SysFM*⁵ plugin is used to define the feature models and all the variation points, and *Pure::Variants* plugin is used for model transformation from a 150% model to a 100% one (for export by supplier for instance). Additionally, the

VPP⁶ plugin has been specifically crafted to propagate variation points from functions to their allocated requirements, while also computing the applicable variants from the variation points set with *SysFM*.

Finally, for cascading the requirements to suppliers (equipment layer), an export is performed directly from *Cameo System Modeler* to *Microsoft Word*.

The overall toolchain is represented in Figure 16.

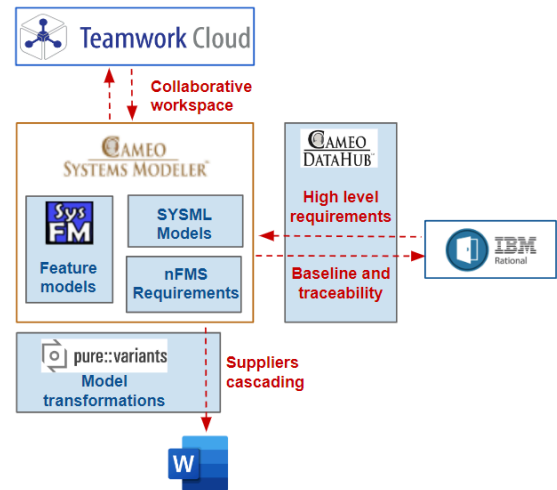


Figure 16: Developed toolchain

f. Customizations

The rationale behind the implemented customizations is to fulfill the nFMS golden rules by extending the *MOFLT* plugin with regards to the desired data model defined in section 3.

In these profile extensions, derived properties have been utilized for the calculation of requirements' stakeholders and applicabilities with respect to variability. These derived properties along with wizards defined in 4.d stand for opaque behaviors scripted in Groovy being well-documented in existing APIs, and with superior performance. Regarding impact analyses, efforts were made to base them as much as possible on *SysML* and *UML[16]* to avoid potential modifications to the existing opaque behaviors within the *MOFLT* framework.

Organizing these profile extensions, opaque behaviors, and wizards as project usage allows for their potential application in other models. Additionally, conducting tests when a new framework or *Cameo Systems Modeler* version is released prior to migrating the nFMS model ensures seamless integration and functionality.

⁵ Airbus SysML profile for feature modeling

⁶ Airbus Variation Propagation Plugin

5. Results assessment and future work

During this proof of concept, all DTS requirements were migrated from *Rational Doors* to *Cameo Systems Modeler*, totaling exactly 19626 requirements without encountering any significant delays in tool usage or project access. This successful migration underscores the scalability of the approach. However, only 0.5% of these requirements were retro-engineered within a functional and technical architecture to conduct impact analysis, though these examples were considered representative of the DTS content. This raises two potential paths forward for the proof of concept: selectively retro-engineering DTS components likely to evolve in future increments or exploring generative AI to address a broader scope. By leveraging *MBSE* for enhanced requirement-based engineering and fostering collaborative work, we anticipate a significant reduction of at least 15% in design lead-time, as we shift focus away from document formatting towards engineering digital assets.

Industrialization of this proof of concept is scheduled for 2025, with over 30 FMS designers expected to be involved. Additionally, exporting generated ICDs to the official interface management tool (Albatros) holds promise for further streamlining design processes and reducing lead times.

However, significant challenges remain, particularly in improving the performance of *MBSELab@Cameo* for DTS generation towards suppliers, which currently falls below industrial standards (1 hour for ~100 requirements). Backup solutions are being currently analyzed to mitigate this risk.

Moreover, a first limitation of the developed toolchain concerns a part of the specification composed of mathematical requirements coming from the aircraft performance equations. The capability of writing mathematical equations in *Cameo Systems Modeler* is limited. Being also implemented in Matlab Simulink by the flight physics team, a bridge between the *Simulink* models and the requirements in *Cameo Systems Modeler* might be considered in a future work.

Furthermore, beyond technical concerns, governance issues have surfaced, including the maintenance of customizations across *MOFLT* plugin releases. These considerations will require careful attention to ensure the long-term success of the project. Finally, it is imperative to develop a customized training program for the nFMS design team.

Conclusion

In this paper, we present a novel approach utilizing *MBSE* and *MBPLE* methodologies to enhance the maintenance of large legacy systems without disrupting procurement processes with suppliers. Our approach emphasizes interoperability with existing processes and official tools while minimizing the training burden on designers.

To accomplish this goal, we introduce a method for linking unaltered textual requirements with a retro-engineered architecture, facilitating impact analysis through a meticulously defined representation of interface parameters. This method, implemented as a semi-automated tool, serves as a bridge between PLE and MBPLE, fostering enhanced collaborative work among stakeholders. This customized and tailored solution is now prepared for full deployment on the current use case.

References

- [1] Ferrogallini, M 2019. 'Modelling and Simulation @Airbus a fundamental digital transformation axis across product, manufacturing and support in service'. *Presentation at MODELS Conference 2019* – Industry Day 18 September 2019 | Munich - Germany.
- [2] Henrique Madeira R, Henrique de Sousa Pinto D, A 2023. 'Variability on System Architecture using Airbus MBPLE for MOFLT Framework'. *INCOSE International Symposium*, vol.33
- [3] Ducamp, C, Bouffaron, F, Ernadote, D, Wirtz, J and Darbin, A 2022. 'MBSE approach for complex industrial organization program'. *INCOSE International Symposium*, vol. 32, no. 1, pp. 839-856.
- [4] Forlingieri, M, Weilkiens, T 2022. 'Two Variant Modelling Methods for MBPLE at Airbus'. *INCOSE International Symposium*, vol. 32, no. 1, pp. 1097-1113.
- [5] Airbus, ZEROe - Towards the world's first zero-emission commercial aircraft, viewed 11 August 2023, <<https://www.airbus.com/en/innovation/zero-emission/hydrogen/zeroe>>.
- [6] D. Ernadote, "Ontology-Based Pattern for System Engineering," 2017 ACM/IEEE 20th International Conference on Model Driven Engineering Languages and Systems (MODELS), Austin, TX, USA, 2017, pp. 248-258, doi: 10.1109/MODELS.2017.4.
- [7] D. Ernadote, "An ontology mindset for system engineering," 2015 IEEE International Symposium

on Systems Engineering (ISSE), Rome, Italy, 2015, pp. 454-460, doi: 10.1109/SysEng.2015.7302797.

[8] Ernadote, Dominique. (2020). MB2SE: A Theoretical Foundation for Systems Engineering Habilitation à Diriger des Recherches. 10.13140/RG.2.2.12357.06886.

[9] Miguel Ángel Mozas Pajares, Carlos Murciano Díaz, Ismael Lafoz Pastor, Carlos Fernández de La Hoz. ICD Management (ICDM) tool for embedded systems on aircrafts. ERTS2 2010, Embedded Real Time Software & Systems, May 2010, Toulouse, France. Hal-02267704

[10] Aeronautical Radio Inc., ARINC specification 429-ALL: Mark 33 Digital Information Transfer System (DITS) Parts 1, 2, 3, 2001.

[11] Aeronautical Radio Inc., ARINC specification 664P7, Aircraft Data Network, Part 7, Avionics Full Duplex Switched Ethernet (AFDX) Network, 2005.

[12] International Council on Systems Engineering (INCOSE) 2007, *Systems Engineering Vision 2020*.

[13] International Council on Systems Engineering (INCOSE) 2014, *Systems Engineering Vision 2025*.

[14] Voirin, J.L., 2017. *Model-based System and Architecture Engineering with the Arcadia Method*. Elsevier.

[15] OMG 2018, *OMG Systems Modelling Language (OMG SysML), Version 1.6*, <<https://www.omg.org/spec/SysML/1.6/PDF>>.

[16] OMG 2016, *OMG® Unified Modeling Language® (OMG UML®), Version 2.5.1*, <<https://www.omg.org/spec/UML/2.5.1/PDF>>.

[17] ISO/IEC 2015, *Software and systems engineering-Reference model for product line engineering and management*, (ISO 26550: 2015).

[18] ISO/IEC 2021, *Software and systems engineering — Methods and tools for the feature-based approach to software and systems product line engineering*, (ISO 26580:2021).

[19] Pure Systems. 2023. pure::variants User Manual. Retrieved from [\[https://www.pure-systems.com/pv-update/additions/doc/latest/pv-user-manual.pdf\]](https://www.pure-systems.com/pv-update/additions/doc/latest/pv-user-manual.pdf)

[20] Dassault Systèmes. (2021). Cameo Systems Modeler 2021x Refresh2 User Manual. Retrieved from [\[https://docs.nomagic.com/display/CSM2021x/2021x+Version+News\]](https://docs.nomagic.com/display/CSM2021x/2021x+Version+News)

[21] Dassault Systèmes. (2021). Cameo DataHub 2021x Refresh2 User Guide. Retrieved from [\[https://docs.nomagic.com/display/CDH2021x/User+Guide\]](https://docs.nomagic.com/display/CDH2021x/User+Guide)

[22] IBM. (n.d.). Rational DOORS and Rational DOORS Web Access 9.6. Retrieved from [\[https://www.ibm.com/support/pages/rational-doors-and-rational-doors-web-access-96\]](https://www.ibm.com/support/pages/rational-doors-and-rational-doors-web-access-96)