Interface modeling for complex systems design: an MBSE and PLM system integration perspective

Yana Brovar[1][0000-0001-9923-9772], Arkadii Kazanskii[1][0000-0002-3930-1244], Betania Tapia[1][0000-0003-3357-9592] and Clement Fortin[1][0000-0002-9761-3107]

[1] Skolkovo Institute of Science and Technology
`yana.brovar@skoltech.ru`

Abstract. A continuous digital thread is not yet fully supported by the product lifecycle tools. A particular discontinuity appears between Model-Based Systems Engineering (MBSE) and Product Lifecycle Management (PLM) solutions and methodologies. Seamless data exchange is an essential element to achieve system components integration, that would support positive emergence and prevent negative ones. A significant integration challenge lies at the heart of MBSE and PLM solutions, since the former facilitates system conceptual design, while the latter focuses on detailed design. However, these two domains are missing common language elements and concepts. We foresee that this could be facilitated through better interface management definition, as one of the most critical integration processes which contains information about both behavior and form, thus, links conceptual and detailed design. In our paper, we propose to use a Design Structure Matrix as a supporting tool to achieve a continuous digital thread, necessary for Digital Engineering solutions. To verify this approach, an aircraft bleed-air system was used as a use-case. We demonstrate that this method can effectively enrich the interface management and control processes through the analysis of metaphors for interface representation proposed by the conceptual design language and product mock-up.

Keywords: PLM, MBSE, Digital Engineering, integration, interfaces, DSM.

1 Introduction

During the product development process all systems are divided into subsystems that interact with one another. When the system is functioning and its components interact through interfaces, there is of emergence appearing, which refers to the functions of the product. Emergence is an important aspect during the design process, as it can lead to both desirable properties, for example, a new product with greater functionality and negative ones which can lead to accidents [1]. Thus, through the analysis of interfaces, we can support desirable functions and prevent negative ones. To support proper interface design, we need to ensure smooth exchange of interface design data between each phase of the product development process. However, the product lifecycle currently is not supported by a continuous digital thread. A particular discontinuity appears between the conceptual design phase and the following stages of the design process. Thus, to accomplish upstream and downstream data flow throughout the product development process, Model-Based Systems Engineering (MBSE) and Product Lifecycle Management (PLM) tools need to be seamlessly integrated to implement Digital Engineering solutions. We thus propose to better represent explicit interfaces throughout the product lifecycle as a possible path to a better integration of these solutions.

1.1 Conceptual design and detailed design

At the beginning of product design, the basic functionality and characteristics need to be defined. This part of the design phase is called the conceptual design, where multiple different product concepts are studied according to the product requirements [2]. Frequently, models are used to support the conceptual design process, and document the product concepts. Models of the product concept which encompass the composition of its architecture, as well as the product's interaction with the environment are at the core of Model-Based Systems Engineering. These models at the conceptual stage can be described with the help of various modeling and domain-specific languages, depending on the specific needs and context of the system being developed. In practice the Systems Modeling Language (SysML) [3] is the most used in industry. It is a general-purpose graphical modeling language with natural language descriptions for defining, analyzing, developing, and validating complex systems. SysML consists of nine diagrams, each focusing on a specific modeling aspect. For instance, the Requirements diagram captures requirements hierarchies and requirements derivation, and “satisfy” and “verify” relationships allow a modeler to relate a requirement to model elements that satisfy or verify the requirements, bridging requirements management tools and the system models. In turn, Product Lifecycle Management solutions are powerful tools for modeling complex products with an emphasis on spatial representations which are also coupled to powerful analysis tools, but lack explicit representations of product functions, which leads to discontinuities of the upstream and downstream data flow with conceptual design. Thus, to develop a continuous digital thread through these phases, MBSE and PLM methods and tools, that are currently applied for conceptual and detailed design phases respectively, need to be seamlessly integrated.

1.2 Context of MBSE and PLM integration

To achieve effective integration of data between the conceptual and detailed design phases, MBSE and PLM should be properly integrated. Major PLM applications vendors have made very serious efforts to extend their solutions to the early and late phases of product development. However, in practice less than 30% of large-scale Digital Engineering projects which require the full integration of MBSE and PLM, are successful [6], as different product lifecycle phases need models with different purposes, which are not well integrated. Thus, to solve this issue, MBSE and PLM should be considered as complementary approaches rather than trying to reach a single solution for such diverse purposes. The need for a smooth data exchange between MBSE and PLM is a subject that is described a lot in the literature. For instance, a general future outlook on this problem [7] highlighted the need for MBSE and PLM integration around Digital Twins and their various applications. Unfortunately, the integration of MBSE with PLM solutions is still very limited in practice. The fundamental issue associated with the integration of MBSE and PLM is due to the fundamental essence of systems, which needs both explicit representations of time and space [8] to completely represent the system structural connections (that corresponds to form) and flow interconnections (that corresponds to behavior) throughout the product life cycle.

2 Literature Review

According to the NASA Systems Engineering Handbook [9] one of the processes applied to the early part of system design as well as to later stages is the Interface Management process. Interface management helps to ensure that different subsystems or components within a system that need to interact with one another, are compliant and work together effectively to ensure proper product behavior.

2.1 Interface management

Parslov et al. [10] noted that interface management is a key product development activity for proper system integration. To store the key information during the Interface Management procedure, Interface Control Documents (ICDs) are used. ICDs record all interface information (such as drawings, diagrams, tables, and textual material) developed for a project. ICDs provide specifics and explain the interface or interfaces between subsystems. One of such standards regarding the ICDs, is the Space engineering Interface management normative (ECSS-E-ST-10-24C [11]). It states that the purpose of the ICDs is to define the design of the interfaces ensuring compatibility among involved interface ends by documenting form, fit, and function. However, this normative document contains minimal information on the development and realization of the ICDs. It just indicates what the ICD should contain. For example, ICDs could contain definitions of terms, descriptions of interfaces, product tree, etc. However, this standard does not include a clear description about the process and methodology in general.

2.2 Interfaces and interface relationships

As provided in the Expanded Guide for NASA Systems Engineering [12], one key aspect of interface management is the identification and definition of interfaces between different subsystems or components. This involves identifying the types of interactions that will take place between these components, as well as the specific requirements and constraints that must be considered to ensure proper communication and operation. System interfaces hold essential information about the interaction of system components with each other and with the environment. The essence of modeling, according to Zeigler [13], “lies in establishing relations between pairs of system description”. According to Prof. Crawley’s definition [1]: interfaces should be represented through form and function. There are two general types of relationships: functional and formal [1]. Functional relationships emphasize the dynamic nature of interactions; in turn, formal relationships denote relationships that are stable for some period of time. To support interaction modeling, several taxonomies were proposed. In current design practice, the most widely-spread taxonomy of interactive systems modeling was proposed by Pimmler and Eppinger [14], where they consider four important types of interaction. The first type of interaction defines the need for adjacency or orientation between two components (structural connection). The other three types are associated with an exchange (flow) of material (or matter), energy, and information. In turn, Crawley et al. extended the four-dimensional taxonomy with an introduction of a 'spatial' type of interface relationship, that “capture absolute or relative location or orientation” [1].

2.3 Metaphors for interfaces and interface relationships

Approaches that denote one kind of entity in place of another to suggest a likeness or analogy between them are defined as metaphors [15]. Metaphors are used to make abstract things more tangible, in order to get the proper level of understanding for the design team members [16]. A study by Jonathan H.G Hey and Alice M. Agonino [17] elaborated on the importance of metaphor use for design itself. They analyzed a variety of engineering design books that led them to discover various perspectives on design that need different metaphors for core design concepts which they defined as ideas, problems, and solutions. Metaphors are also very important for interface design. Referring again to the Expanded Guide for NASA Systems Engineering [14], the interface management procedure involves well defined documentation of the interfaces and analysis of interface compatibility. To realize this, a proper approach for interface modeling is needed. This could be done through formulating metaphors to achieve proper representations of interfaces to help design team members from different disciplines to collaborate and effectively highlight system interfaces. For instance, one the most obvious interface representation metaphor is arrow, which can metaphorically represent pipes, wires etc.

2.4 DSM

Another widely-used approach that proposes interface metaphor is the Design Structure Matrix [18]. It is a useful tool for representing the interactions between different elements, as it is an object-relationship model that sees value in both components and relationships. The notation utilized in DSMs is both clear and intuitive, and it offers as well the benefit of modularity and customization to create Multiple-Domain Matrices [19], enabling adaptation and extension according to the specific design phase or product context. DSM supports adaption to the domain-specific cases, as demonstrated in [20] with the formulation of new types of interface relationships for telecommunication systems. The metaphor to represent the interface consists of the designator (for example, *S* for structural connection, *Sp* for spatial orientation, *E* for energy flow, *M* for material flow, and *I* for information flow), and representation of the direction of the flow depending on the matrix convention. For instance, interactions below the diagonal indicate feed-forward interactions and above the diagonal interactions indicate feedback. Feedback and feed-forward are especially important for time or decisions-based sequences.

3 Methodology

Current tools implemented at the conceptual design phase are more focused on behavioral representation, which means that in terms of interfaces relationships, commonly available, only functional interfaces, such as material, energy and informational flow are modeled, ignoring spatial interposition of components and physical connections. However, these interface relationships are critical to ensure proper flows within the systems. Such lack of attention to these types of interface relationships leads to the gap between MBSE and PLM integration. MBSE tools are more focused on flow representations, while PLM lacks specific flow representation, but can represent spatial and structural interface relationships. To overcome this gap, we propose proper metaphors for all interface relationships types both at the conceptual and the detailed design phases. At the conceptual design phase, we foresee DSM as a tool that supports the interface modeling as well as the analysis of system integration. To support interface relationships representations at the detailed design stage, we propose to use 3D representations as interface metaphors. We foresee such representations as concrete implementation tools to increase the clarity of products and systems to support the reduction of negative emergence. Thus, DSM and 3D metaphors of interface relationships allow a transition between the early phases of the product development process and CAD modeling and analysis.

4 Bleed-air system use-case

To demonstrate the proposed method, an aircraft bleed air system within the aircraft pylon was chosen as a use-case (see Fig. 1).

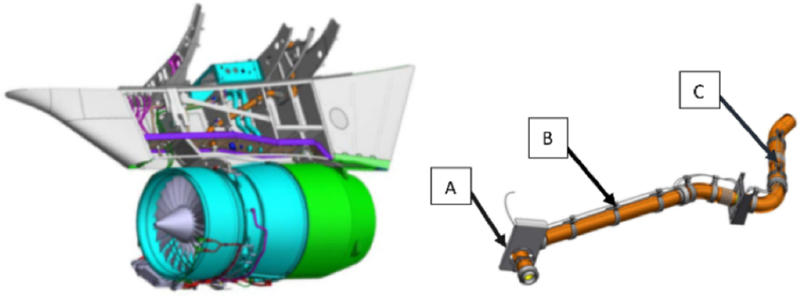


Fig. 1. Aircraft pylon (left hand-side) and Bleed-air system (right hand-side) example

The bleed-air system main role is to transfer a pressurized airflow from the engine compressor to the aircraft fuselage for the heating and cooling of the cabin. One of the most critical emergent properties of the bleed air system is connected to the potential risk of a gas leak, which could lead to a fire. In order to reduce potential risks, leak detection wires are attached to the bleed air system and are used to monitor the temperature at the bleed air pipe wall. Thus, bleed air systems must be designed carefully and must meet a large number of requirements coming from stringent certification rules to prevent the negative emergence of the aircraft system [21].

4.1 Conceptual design

The conceptual design was conducted in the MagicDraw [22] software that utilizes the SysML modeling language containing numerous diagrams which define the system from different perspectives. For the purpose of the paper, we discuss the Requirements diagram and the Internal Block Diagram, as they offer specific metaphors for the representation of the interfaces. The requirements diagram proposes a combination of a graphical modeling and natural language approaches to represent interfaces; it has a specific entity called “interfaceRequirement” to emphasize some constraints dedicated to the system interface through the formulation of the sentence using a natural language. For instance, in Fig. 2 the main function of the bleed-air system is represented as a performance requirement that is strongly interconnected with positive emergence of the climate control function that could be reached through the proper integration of the main compressor, the bleed-air system and the airframe.

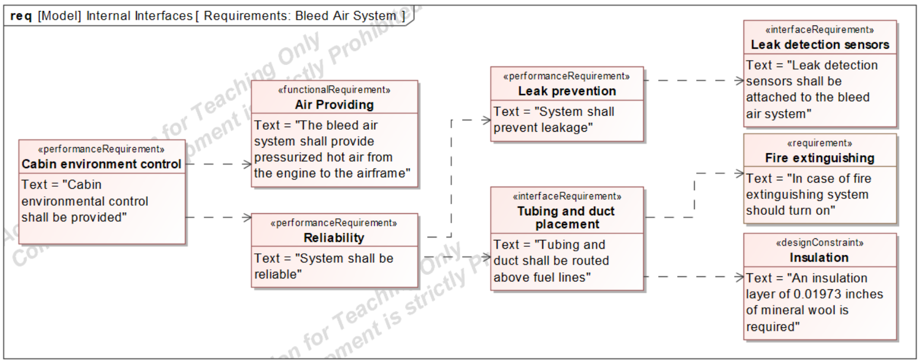


Fig. 2. Requirement diagram for the system of interest

Spatial interfaces are defined through a natural language in the interface requirement of the duct placement, that states that the bleed-air system duct shall be routed above the fuel lines, to avoid a potential fire in the pylon. Thus, in order to prevent a negative emergence of a fire, a structural interface between the bleed-air system and the leak detection sensors is needed to track the thermal flow (energy flow). These interfaces are extremely important, as the error with their design could lead to a high possibility of a system failure. However, the verbal form of the interface representation could lead to a discrepancy, and it is a challenge to manage and analyze textual representations.

The next step of interface design during the conceptual design phase of the Bleed Air System is represented through the SysML Internal Block Diagram. A critical point of the system is demonstrated in a separate diagram for the best understanding of the interfaces between subsystems. As could be seen in Fig. 3, numerous interfaces are represented as several flow modeling constructs. Metaphors for these interfaces are represented as colored lines with directions of flow between subsystems and port identifiers. Also, a connection to the requirement on the leak detection sensors was provided as a rationale for the information flow from “Leak Detection Lines block to Fire Extinguishing System block”. In actual SysML representations, only information, energy and material flows are present. The flow metaphors are listed in the legend of Fig. 3. However, it would be necessary to include structural and spatial interface representations in order to connect all of the requirements to the 3D detailed model of the system.

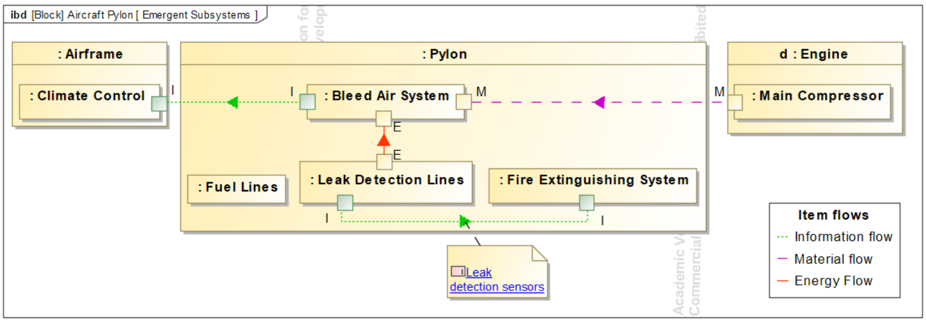


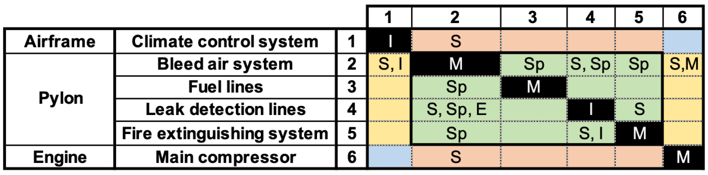
Fig. 3. Internal Block Diagram for the critical elements

4.2 Detailed design

A solid model representation of the pylon and bleed air system are shown in Fig. 1 (right-hand side). The bleed air system is divided into three segments as shown in Fig. 1. These three segments represent the main components of the bleed air system. Each component is connected through main interfaces. There are many secondary interfaces in the bleed air system which include bellows to consider the expansion of the pipe due to large temperature differences and also connectors which ensures the sealing of the insulated pipe segments. The letters in the box represent the segments of the bleed air system. Segment A has a structural component, which connects it to the firewall. Segment B includes the middle section with bellows at both ends. Segment C includes structural components which connect the system to the aircraft fuselage. The leak detection lines cover all three segments of the bleed air system. This is a critical subsystem, especially as it prevents negative emergence (leakage) that can occur.

4.3 DSM as integration tool

As discussed above, functional interfaces at the conceptual design stage are represented through SysML notation with verbal description in the Requirement Diagram and complemented with Internal Block Diagrams, which allows them to represent flows that exist between system components. In turn, structural interfaces are represented through assembly models in 3D models at a detailed design stage. And there is almost no interface data exchange between these two phases. At the same time, spatial interfaces are not explicitly represented within any of these tools, as they are usually just noted in the Requirement Diagram or some specific positioning of components within a Product Mock-up. Thus, none of these metaphors cover Crawley's definition [15] of an interface, as the conceptual phase is focused mostly on the function representation, and detailed design is focused mainly on the representation of the form. We foresee DSM as a proper tool for the interface representation, as it contains system components (as form) and types of interaction at the intersection of columns and rows (as function). Furthermore, the DSM-based interfaces management model is an appropriate tool to reduce complexity, as it permits to simplify the visualization of the system on one page and supports quantitative analysis by adding values at the intersection of corresponding columns and rows.



			1	2	3	4	5	6
Airframe	**Climate control system**	**1**	I	S				
Pylon	**Bleed air system**	**2**	S, I	M	Sp	S, Sp	Sp	S,M
	Fuel lines	**3**		Sp	M			
	Leak detection lines	**4**		S, Sp, E		I	S	
	Fire extinguishing system	**5**		Sp		S, I	M	
Engine	**Main compressor**	**6**		S				M

Fig. 4. Design Structure Matrix of interface relationships for the pylon bleed-air system

The higher interconnectivity between the bleed air system with the airframe and the engine requires accurate management of interfaces to prevent system failures. The bleed air system functional relationships include the flow of energy (*E*), material (*M*) and information (*I*) and represent the dynamic aspects of the system. All of these types of interactions are inherently unidirectional, in that they have a specific flow direction and must be read from column to row in the DSM. The off-diagonal element of the matrix is marked with *S* when two given components have a structural relationship between each other. The matrix element is marked with *Sp* when the relationship between two components is Spatial. Both structural and spatial relationships are bi-directional. For example, fuel lines have a critical requirement to a spatial relationship to the bleed air system to prevent an overheat. A matrix element is marked with *I*, *M* or *E* when there is an Informational, Material or Energy flow, respectively. Informational flow represents signals from the bleed air system to the leak detection lines that need to check whether there are any leaks from the ducts of the bleed air system to prevent overheating and fire. The flow of energy in this DSM represents the thermal exchange between the bleed-air system and the leak detection sensors, data from which is converted to information about the bleed-air system state. Material flows represent fluid flows that are contained within the fuel lines. So, the DSM contains information that is relevant for the concept development as it contains the information about functional interactions (flows), and at the same time it represents the formal interfaces through the structural and spatial interfaces in the detail design. Thus, the DSM is a link between function to form and vice versa and is thus a powerful integration tool within the Digital Engineering chain.

5 Discussion and Conclusion

Fig. 5 shows the proposed interface data flow thread through the product development process with the identification of the tools applied for each phase (green rectangles) of the product development process, already existing metaphor (black font), and metaphors that we propose to add (blue font) to fulfill the gaps between MBSE and PLM models. The process begins with an iterative process of requirements and system structure identification. Interface requirements are commonly represented in a SysML Requirement Diagram through a natural language metaphor using the verbs “shall’ or “must”. In turn, at the stage of the system structure identification, the system interactions are indicated on the Internal Block Diagram through the SysML ontology elements, such as connectors, ports, item flow etc. However, in this type of metaphor, there is no representation of structural and spatial interfaces which are missing in SysML. This creates an important gap in the interface management chain, which results in a broken digital thread that leads to inefficiencies and errors.

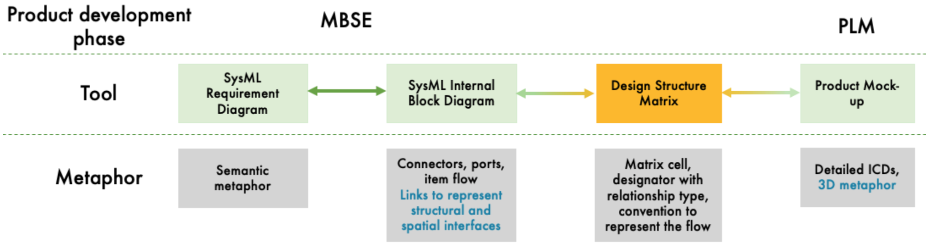


Fig. 5. Representation of interface metaphors interoperability and continuity

Also, interfaces are only implicitly modeled in PLM tools. Currently, there are almost no specific methods to explicitly represent interfaces in the Product Mock-up other than in the 2D drawings of the ICDs. To solve these problems within the digital thread, we propose to complement solid modeling ontologies with three-dimensional metaphors for representing interfaces of various types. In order to ensure a proper level of continuity and interoperability between the early stages of the design process with PLM solutions that could support Digital Engineering solutions, we foresee the Design Structure Matrix (orange rectangle in Fig. 5) as one appearance of mixed interface metaphors, which could integrate information about all types of interfaces in an easy to use and clear representation. Moreover, metaphors offered by the DSM-based approach could be supplemented or replaced with crosses or dots, numbers or the color identification at the intersection of the corresponding row and column. As future work, from a conceptual design perspective there exists a strong need to properly define structural and spatial interface representations, in order to provide a complete set of the interface information and better integrate upstream and downstream processes to ensure a continuous digital thread which will help to prevent negative emergence. As well, it is planned to extend the DSM analysis with the addition of quantitative characteristics to add quantifications to evaluate and analyze both positive and negative emergence. We also emphasize the need for proper 3D interface metaphors to be developed for each interface type as defined in a DSM, in order to increase clarity and properly support the representation of product functionality in a 3D model and analysis. As a way for the validation we plan to provide comparison of proposed approach with the existing methods implemented in industry, for instance, "RFLP" (Requirements engineering, Functional design, Logical design and Physical 3D CAD design).

References

1. Crawley E., Cameron B., Selva D.: System Architecture: Strategy and Product Development for Complex Systems, Boston: Prentice Hall Press (2016).
2. Horvath, I. (2000) 'Conceptual design - inside and outside', 2nd Seminar and Workshop EDIPro, (January 2000), p. 10.

3. Object Management Group: OMG Systems Modeling Language: The official OMG SysML site. http://www.omgsysml.org/. OMG. See Object Management Group, (2008).
4. ISO: Industrial automation systems and integration – JT file format specification for 3D visualization, ISO 14306:2012 (2012), www.iso.org
5. ISO 10303-242:2014, Industrial automation systems and integration – Product data representation and exchange – Part 242: Application protocol: Managed model-based 3D engineering
6. McKinsey&Company: Unlocking Success in Digital Transformations, Organization, Oct 2018, 14 pages
7. D. Gerhard, S. S. Cordero, R. A. Vingerhoeds, B. Sullivan, M. Rossi, Y. Brovar, et al., "MBSE-PLM integration: Initiatives and future outlook", Proc. 19th IFIP Int. Conf. Product Lifecycle, pp. 1-10, Jul. 2022.
8. Menshenin, Y., Knoll, D., Brovar, Y. and Fortin C.: Analysis of MBSE/PLM Integration: From Conceptual Design to Detailed Design, Proceedings of the IFIP 17th International Conference on Product Lifecycle (2020).
9. NASA (2016). Systems Engineering Handbook.
10. Jakob Filippson Parslov. Defining Interactions and Interfaces in Engineering Design. Technical University of Denmark, 2016.
11. Space Engineering Interface Management, ECSS-E-ST-10-24C, 2015.
12. Steven R Hirshorn. Expanded guidance for nasa systems engineering. volume 1: Systems engineering practices. Tech. rep. 2016
13. Zeigler B., Muzy A. and Yilmaz L. (2009), “Artificial Intelligence in Modeling and Simulation”, In Meyers R. (eds) Encyclopedia of Complexity and Systems Science. Springer, New York, NY. https://doi.org/10.1007/978-0-387-30440-3_24
14. Pimmler, T.U., Eppinger, S.D.: Integration analysis of product decompositions, in: ASME Design Theory and Methodology Conference (1994).
15. https://www.merriam-webster.com/dictionary/metaphor#:~:text=%3A%20a%20figure%20of%20speech%20in,as%20in%20drowning%20in%20money) (accessed: 13 February 2023)
16. Erickson, Thomas D. "Working with interface metaphors." Readings in Human–Computer Interaction. Morgan Kaufmann, 1995. 147-151.
17. Hey, Jonathan H., and Alice M. Agogino. "Metaphors In Conceptual Design." 10 June 2016.
18. Steward, D.V.: The design structure system: A method for managing the design of complex systems. In IEEE transactions on Engineering Management (3), 71-74 (1981).
19. Lindemann U., Maurer M., Braun T. Structural complexity management: an approach for the field of product design. – Springer Science & Business Media, 2008.
20. Meskoob B., Brovar Y., Menshenin Y., Blanchard F., Fortin C., Interface Management for Telecommunicayion System Design through OPM and DSM-based Approaches, in: The 17th Annual IEEE International Systems Conference (2023)
21. Brovar Y., Menshenin Y., Fortin C. Study Of System Interfaces Through The Notion Of Complementarity. Proceedings of the Design Society; 1:2751-2760. doi:10.1017/pds.2021.536 (2021).
22. https://www.magicdraw.com/ (accessed: 13 February 2023)

Acknowledgments

The authors acknowledge the contributions of Eagle Star Aviation that was staffed by students from different Quebec universities affiliated with the “Comité sectorial de Main-d’oeuvre en Aérospatiale” (CAMAQ) and their industrial partners.