A Collaboration Framework Using Digital Twin for Dynamic Simulation and Requirements Verification Based on MBSE and the MIC Concept.

Imane Bouhali
Systems Engineering Dept
Safran Seats
Issoudun, France.
imane.bouhali@safrangroup.com
0009-0005-9155-4707

Vincent Idasiak
PRISME Laboratory
INSA CVL
Bourges, France
vincent.idasiak@insa-cvl.fr

Jacques Martinez
Systems Engineering Dept
Safran Seats
Issoudun, France
jacques.martinez@safrangroup.com

Faida Mhenni
QUARTZ Laboratory
ISAE-SUPMECA
Saint-Ouen, France
faida.mhenni@isae-supmeca.fr
0000-0002-6990-6164

Jean-Yves Choley
QUARTZ Laboratory
ISAE-SUPMECA
Saint-Ouen, France
Jean-yves.choley@isae-supmeca.fr
0000-0002-9960-6808

Luca Palladino
Systems Engineering Dept
Safran Seats
Issoudun, France
luca.palladino@safrangroup.com

Frederic Kratz
PRISME Laboratory
INSA CVL
Bourges, France
frederic.kratz@insa-cvl.fr
0000-0003-2620-803X

2024 IEEE International Systems Conference (SysCon) | 979-8-3503-5880-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SysCon61195.2024.10553495

***Abstract*—Thanks to their clarity, the use of system models is increasingly widespread in industry, providing engineers with the means of traceability, and reducing ambiguities. Systems engineers use an MBSE (Model-Based System Engineering) approach to describe different views of a system (requirements, operational view, functional view, behavioral view, logical and physical architecture views). The system model is useful to design the system architecture in either a static or a dynamic perspective. The first step in systems engineering is the requirements specification. This is an important step in the development phase, since the system must satisfy the defined requirements (functional and physical), to be able to choose the right architecture solution. Many trade-off methods exist: in some cases, it is sufficient to use parametric diagrams to calculate the system performances based on the functional causality of the system, and to execute simulations in the system modeler tool. However, there are some performances that need a more complex dynamic model requiring, for instance, differential equations, as well as a suitable solver, in a dedicated simulation tool, especially when the system is a complex mechatronic system, that needs multiphysics simulation. Many industries have adopted the digital twin (DT) concept in their activities, because of its fidelity and precision. It is highly recommended to use DT models during the early phases of development in order to verify system requirements, with a view to reducing development costs [18]. It is then necessary to ensure digital thread between system models and DT simulation models. To address these issues, this paper presents a collaboration methodology between simulation engineers and system engineers. The aim is to achieve earlier requirements verification and trade-off decisions. In order to fulfill these objectives, it is necessary to ensure the traceability of models changes and to reduce the probability of miss-understandings. For this reason, the MIC (Model Identity Card) concept is used in the proposed methodology.**

***Index Terms*—Digital Twin, MBSE, Mechatronic, MIC, Requirements, Verification, Simulation.**

I. Introduction

Requirements verification is needed from the very early phases of development, when the system prototype has not yet been built. Physical tests cannot be the first means of requirements verification for two main reasons: first, building and operating the physical prototype to carry out physical tests is a lengthy process, since it is the last step of the V-process, and second, it requires a large number of physical resources that can increase the development costs. This is particularly the case when the requirements are not satisfied, as this causes V-process iterations, and involves manufacturing multiple prototypes.

After identifying the stakeholders' needs, the requirements description and specification are the first steps of the development V-process. Systems engineers used to complete this task using a document-based approach. With the increasing number of requirements of complex mechatronic systems, involving discipline-specific definitions of requirements, this approach proved to be inefficient. It has been replaced by the MBSE (Model-Based System Engineering) approach [6], which not only makes the requirements definition task easier, but also facilitates the system architecture design as well as design changes traceability, thus avoiding development risks [2]. In addition to the MBSE approach, the MBD (Models Based Design) approach focuses on the use of models throughout the development process, providing means of trade-off and decision making analysis [12].

Many tools and methods exist for MBSE modeling. De Saqui-

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

Sannes et al. in [5] proposed a set of keys and decision criteria to help MBSE users (academics and industries) to select the appropriate MBSE languages, tools and methods for their study. For our case study, we used SysML as the modeling language, the CESAM Architecture Grid [11] as the methodology, and Cameo Systems Modeler as the tool.

The requirements verification (RV) issue has been addressed by many researchers who have proposed RV methods using the MBSE approach in order to check requirements compliance. Bruggeman et al. in [4] proposed a requirement verification framework to integrate MBSE in a MDAO (Multidisciplinary Design Analysis and Optimization) process in the aircraft design process to ensure the traceability from requirements to product design. They used an automatic requirements verification by formalizing requirements into patterns and applying compliance means and test cases. The MBSE approach showed its added value in their proposed framework.

In the systems engineering design process, to define the physical architecture of the system, many technological solutions need to be proposed by systems engineers in order to perform a trade-off analysis after requirements verification of each proposed solution, based on different trade-off metrics [15]. Parnell et al. in [17] showed the interest of using MBSE for decision making purposes by performing a trade-off between alternative systems decisions throughout the whole life cycle based on different evaluation performances. The need for physics models and simulations to carry out the trade-off study is also discussed by Hausse et al. in [16], where the authors used PTC Integrity Modeler SySim as the MBSE tool. This modeler enables the execution of SysML models, making the MBSE tool able to handle both the architecture design, and the trade-off analysis using simulation capabilities, thanks to parametric diagrams. Nowodzienski et al. in [13] showed the interest of integrating simulation in the MBSE approach by leveraging system models with simulation techniques. When the system is complex and needs a suitable compiler, these simulations have to be done with another dedicated simulation tool, so that the simulation can be executed with greater precision. This is particulary so when the system is a mechatronic system that needs the design of discipline-specific models. Other authors therefore proposed SysML extensions so as to be able to export SysML models into dedicated simulation tools such as Simulink or Modelica. OMG (Object Management Group) proposed a SysML extension for Physical Interaction and Signal Flow Simulation (SysPhS) to generate simulation models automatically from SysML [14]. The SysPhS extension requires the design of specific diagrams in SysML that are not necessarily part of the classical tasks of the systems engineer (parametric diagrams and state machines for each solution proposition). Moreover, these diagrams are highly dependent on the simulation intention which changes when the test use case changes.

Recently, the concept of digital twin (DT) has become widely used by many industries [19] in different life cycle phases with different integration levels. Focusing on the development phase, the digital twin offers means of virtual simulation from the beginning of a project until the system manufacturing and operation. It has the same architecture of the physical system since it is its virtual representation, many authors used the MBSE approach to design the DT functional architecture [3] and physical architecture [21]. The DT can be considered as a system since it is composed of components (virtual models) and interaction between its components (co-modeling), and with the physical system (control and sensing). An important aspect to address is then the consistency between the system architecture and its digital twin architecture. Once consistency is ensured, the requirements verification and trade-off process can be executed using the digital twin model, and validated in the system MBSE model.

For this purpose, this paper proposes a collaboration framework between 3 actors: systems engineer, simulation architect and simulation engineer. This methodology enables consistency and digital thread between system MBSE models, DT MBSE models, and DT simulation models. The remainder of the paper is organized as follows: section II will present the model identity card (MIC) concept and its benefit for our approach; the proposed methodology is presented in section III, and a case study to validate the methodology is presented in section IV. The conclusion and future work are discussed in section V.

II. Model Identity Card (MIC)

The concept of Model Identity Card was proposed by Sirin et al. in [20]. Its aim is to define an identity card for a simulation model containing attributes that normalize the simulation model specifications. The attributes are all the necessary information that a simulation model can have (modeling method, modeling tool, model performances,...). Its main benefits are that it reduces ambiguities between simulation model stakeholders and support model specifications.

The MIC classes and attributes are represented in figure 1. For our approach, we proposed new attributes in the MIC; these attributes are detailed in section IV-C.

The authors in [7] created a MIC profile in the Capella tool, in order to transform MBSE models to physics-based simulation models. Their study focused on multiphysics models management, and showed the real interest of the MIC concept application.

III. Proposed Collaboration Framework

The proposed methodology consists of three main actors: systems engineer, simulation architect, and simulation engineer (figure .2). Each one of them has different roles and should provide a list of deliverables which can be of various kinds (models, diagrams or documents).

- Role of systems engineer: He uses the MBSE approach to design the system architectures, and then he provides the simulation architect with the requirements that he wants to verify using simulation capabilities. He also needs to provide the simulation architect with the system architecture, described for instance through the SysML

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

Attributes	Remarques	Type	Example	Main Classes
Generic Name *	Physical componant regroupment	String	Engine	Object Description
Specific Name *	Unique identifier	String	Compressor 7V16	
Granularity Level *	List(System/Sub-system/Componant)	String	Sub-System	
Developer Name *		String	F.Ravet	
Model Version no. *	x.x format	Float	0.1	
Creation Date		Date	14/03/2013	
Documentation	Attached technical report	String		
Image	Attached references image	Image		
Model Dimension	List (0D-3D, mix)	String	1D	Method
Chosen Method	List (Finite Volumes, Finite Elements, Finite Difference, OD...)	String	Finite Difference	
Physical Equations	List (Chemistry, Dynamic behavior of materials, Maxwell, Navier-Stokes, Strength of materials, Electric, Signal, Runge Kutta)	String	Navier-Stokes	
Integrated Solver	List (Controllable Pitch, Fixed Pitch, Without Solver)	String		
Time Step	List (Second, Minute, Mili-second, Hour, Steady state)	String	Second	
Linearity	List (No/Yes)	String	No	
Discontunity	List (Yes, No)	String	Yes	
Name of Compilator	List ()	String	Yes	Usage
Time Computation	List (Elapsed Time / Real Time)	String	Elapsed Time	
Scalability	List (Yes/No)	String	Yes	
Tool Name	List (Amesim, Matlab Simulink, GT-Power, Modelica...)	String	GT-Power	
Tool Version	x.x format	String	7,3	
Hardware Requirements	CPU, OS etc...	String		
Accuracy	Requested/Provided Accuracy	Float	%+-5	Model Quality
Robustness	Requested/Provided Robustness	String	1	
Software (Code) Verification	List (Candidat/Development/Previous/Reference)	String		
Solution (Mathematical) Verification	Level 1(Poor), Level2 (Satisfactory), Level3 (Good), Level4 (Excellent)	String		
Validation	Level 1(Poor), Level2 (Satisfactory), Level3 (Good), Level4 (Excellent)	String		

Fig. 1: Model Identity Card classes and attributes [20]

BDD (Block Definition Diagram) and IBD (Internal Block Definition), in order to ensure consistency between simulation models and system components. Furthermore, the systems engineer has to define trade-off metrics and evaluation performances, because based on that, he will choose the ”best” configuration after trade-off analysis, based on the results report made by the simulation architect.

- Role of the simulation architect: His role, discussed in [20],is that of a model designer, who transforms MBSE models to more detailed domain models to reduce the gap between the systems engineer and the simulation engineer. With the advent of the DT concept, his new role is to be be responsible for the Digital Twin (DT) MBSE model management and development of the MICs. A MIC has to be defined for each DT simulation model. The simulation architect will then define a MIC library, which will be delivered to the simulation engineer, depending on the simulation intention, to execute simulations according to the information in the MICs. After that, the simulation architect will prepare a results report and send it back to the systems engineer.
- Role of the simulation engineer: He is responsible for the DT simulation model which is developed in a dedicated simulation tool (Matlab, Dymola, ...). His role is to execute simulations according to the MICs information and simulation intention, and to send the results back to the simulation architect.

The process is detailed in figure 3, which represents an activity diagram of the three actors and the collaboration between

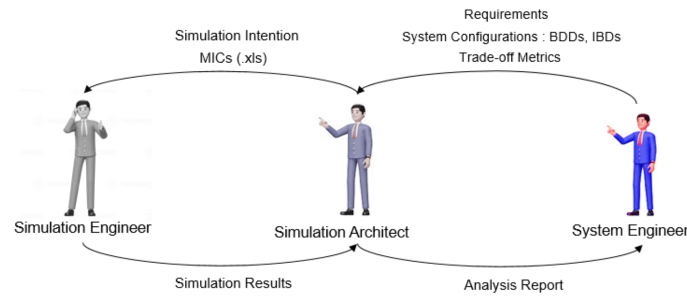


Fig. 2: Collaboration between: system engineer, simulation architect and simulation engineer

them.

IV. Case Study: Aircraft Seat Mechatronic TestBench

A. System Description

To illustrate the described methodology, we used a business class (BC) aircraft seat as a case study. The seat needs to satisfy aircraft requirements, because it will be integrated in the aircraft environment, and mechanical and electrical integration requirements, because it is a mechatronic system. A BC aircraft seat is composed of:

- Mechanical parts: Seatpan, Legrest, Backrest, and different parts of the seat structure that enable seat kinematics.
- Electrical parts: Actuators, Controllers, Sensors, Lights.

B. Digital Twin Architecture

Companies produce a diverse range of products. To avoid constructing a digital twin for each product, as discussed by Grieves et al. in [8], we designed a testbench digital twin, which is a modular structure representing the entire range of company products. This DT is connected in real-time with the testbench physical structure, composed of:

- Generic Seat: Seatpan, Legrest, and Backrest.
- Multikinematics Platform: enables the generic seat to move following different trajectories, in order to represent all the seats kinematics.
- Bench Actuators: actuate the multikinematics platform and the generic seat.
- Seat Sub-systems: the testbench provides the possibility to integrate seat sub-systems into the physical structure, to come closer to the real seat structure with the aim of testing the integration of the seat components in the seat environment.
- Bench Sensors: to instrument the testbench, we implement a set of sensors, in order to obtain behavior feedback data.

The Testbench DT is composed of:

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

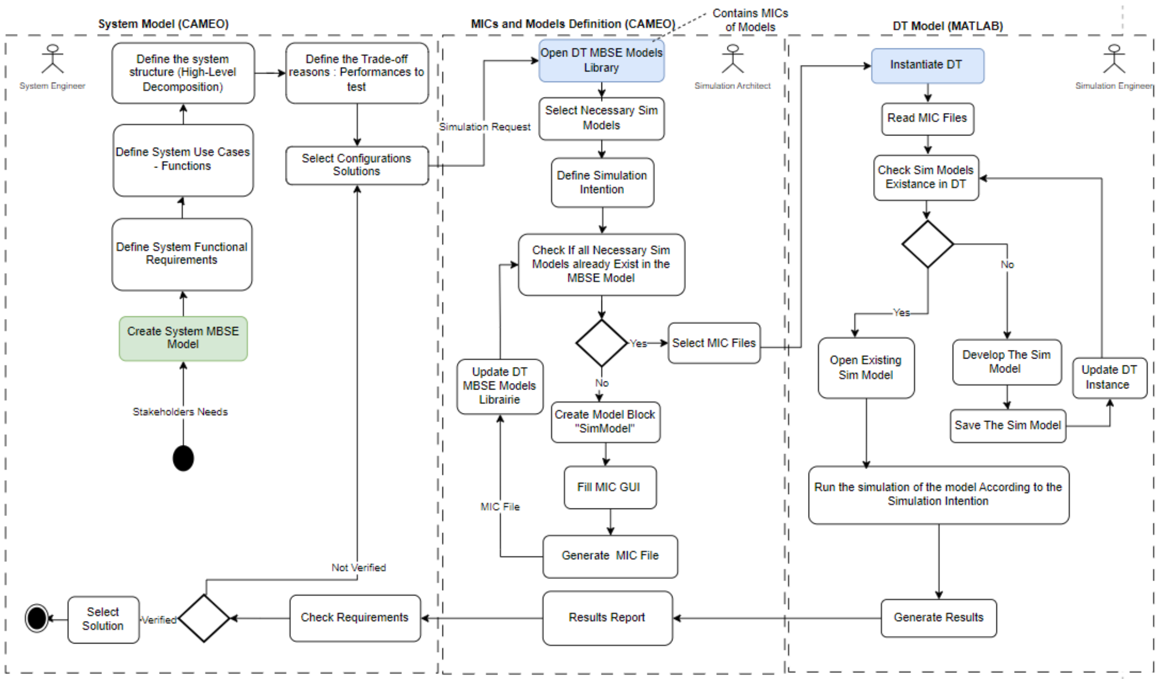


Fig. 3: Main Activities of the three actors for simulation and requirements verification

- Bench GUI: A graphical user interface to communicate with the DT user, instead of modifying the DT models directly, to facilitate DT use.
- Bench Models: Include all DT multiphysics dynamic models.
- Seat Sub-systems Models : Represent the seat specific models, and constitute a library of models, used to select the appropriate sub-systems for each seat program we want to test and integrate in the testbench DT.
- Real-Time Model: The model used to control testbench actuators in real-time: for this purpose, we used dSpace hardware that integrates the real-time model (configuration and control). This model controls both the DT and the physical testbench actuators.
- Data Analysis Model: Built based on a comparative study between real data collected from the physical testbench, and simulation data calculated from the testbench DT. This model is used to adjust DT models, to be more faithful to reality.

The testbench physical structure is represented in figure 4, as well as its interaction with its DT.

The Testbench architecture is represented in figure 5, where two types of flow appear: the flows in green represent data flows, and the flows in black represents exchange of models (FMU blocks flow and controlLaw flow).

C. MIC Profile Creation

To create MICs for the DT models, a MIC profile was created in Cameo Systems Modeler, and a MIC GUI was elaborated to represent all the testbench DT models and their MIC attributes.

The MIC profile is based on the MIC specification defined in [10]. The attributes were typed based on the defined enumerations. Some attributes were added to focus on certain aspects: since the digital twin is built to represent a modular testbench, it contains then ”variant” blocks (Model 150%), and variability management will be modeled in a specific variability tool such as PureVariant [1]. An attribute named ”Variability” that indicates if the model contains variants or not is then added to the MIC. Moreover, the MIC differentiates between variables that are associated to a port or an interface of the model, and internal variables that evolve with the model behavior evolution, but are not represented by a port. Sometimes, these internal variables need to be observed for prediction analysis or failure-cause identification: for example, the resistance of an electrical motor, that can change because of temperature. We then added to the MIC, an attribute named ”Observer” to indicate what the appropriate observer for this resistance is, based on model variables (Motor Voltage (U) and Motor Current (I)), which can be measured through the model interfaces, without modifying the internal blocks of the model. This new attribute will indicate equation (1) to calculate the internal variable, or the virtual sensor provided by the simulation tool. A new attribute named ”Attached Files” was also added to indicate any files attached to the model (the script files containing the parameter values for example, or the CATIA model needed to export the system catparts that are used in the simulation model). The MIC Profile is represented by the Meta-Model in figure 6. The ”Component” and ”Physical Port” classes are outside the defined MIC profile, and are used to be specialized by other classes in the MIC profile.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

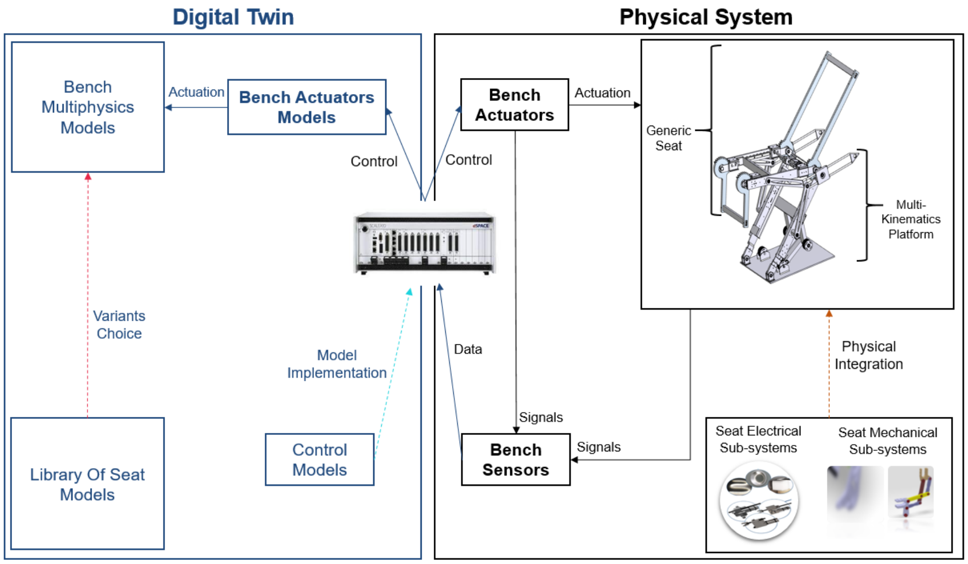


Fig. 4: Mechatronic TestBench Structure Interaction with its DT

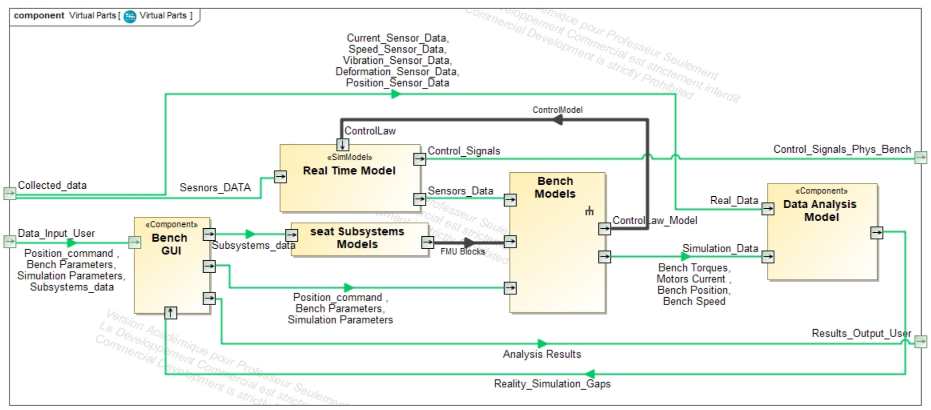


Fig. 5: DT High Architecture

$$R = U/I \tag{1}$$

D. Requirements verification: DT Simulation

To illustrate our approach, we defined a simulation scenario to verify the requirements of the business class aircraft seat, defined by systems engineers. The process was organized in 3 steps:

Step 1: Requirements and simulation intention definition
After modeling the functional and physical architecture of the seat, the systems engineer has to perform a trade-off between different architecture configurations. Therefore, he defines a BDD for each solution (figure 8a), where physical (referenced) components (figure 8b) are associated to the logical components using the ”Generalization” link. The chosen solution has to satisfy the linear position, angular speed and control performance requirements of the linear movement of the seatpan. The use case scenario (figure 9) is the movement from the TTL (Taxi, Take-Off, and Landing) position (Seated Position) to the BED position. The BED position is achieved when the seatpan position (x) reaches 0.4747m, the angle between seatpan and backrest (tetha1) is 0 deg, and the angle between seatpan and legrest (tetha2) is 0 deg. The requirements are represented in the requirement diagram (figure 7). The systems engineer then needs to verify these requirements for each defined configuration, by requesting a simulation from the simulation architect.

Step 2: MICs identification and models selection
The simulation architect will then select the DT models in the DT MBSE model according to the seat architecture. The component name will correspond to one or more simulation models, depending on the model attributes. He has to select the MICs according to the required simulation request. For instance, for the component ”DC Rotational Motor P/N 12678”, the corresponding simulation model will appear as ”DC Rotational Motor P/N 12678 **Model**”. Its behavior can be modeled by more than one model. These models have

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

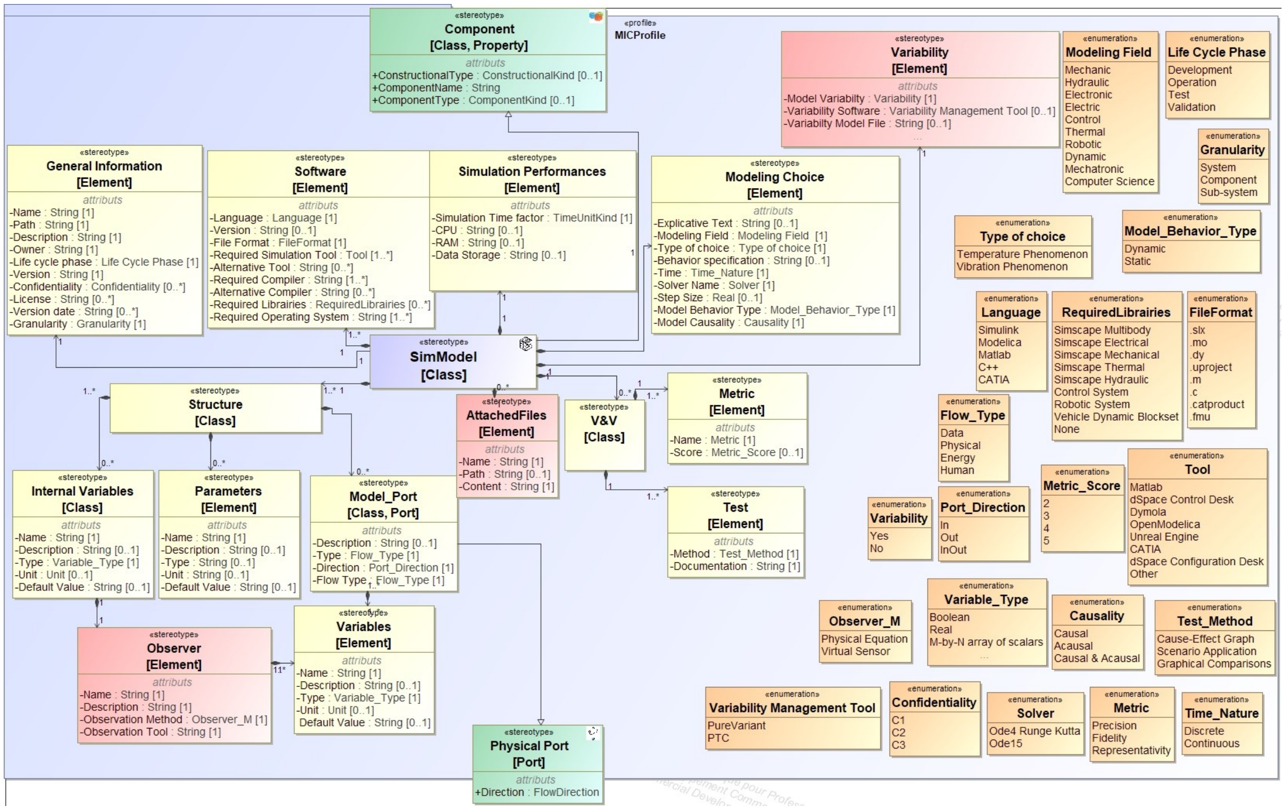


Fig. 6: MIC Profile in Cameo Systems Modeler

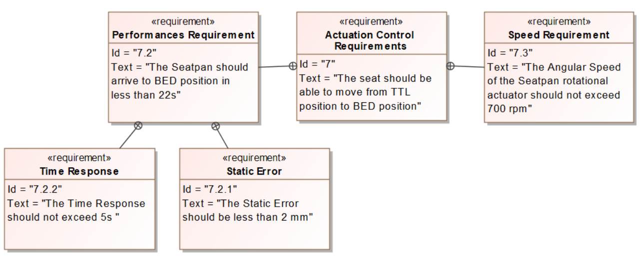
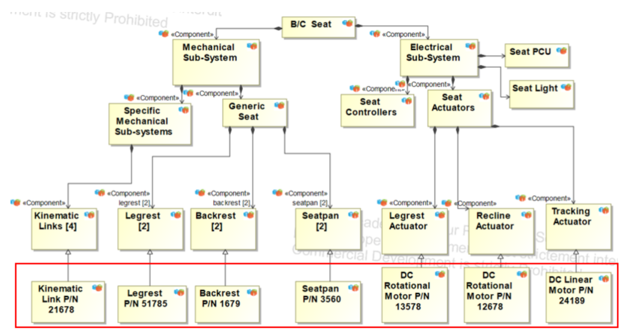
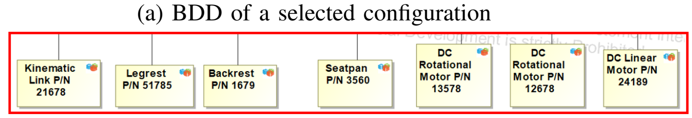


Fig. 7: Actuator Control Physical Requirements



(a) BDD of a selected configuration



(b) Zoom in the referenced components of the selected configuration

Fig. 8: Seat Configuration Physical Decomposition

different attributes, e.g. the electrical model of a motor can be built using different tools/libraries, with different methods, as depicted in the MIC table (fig.10). The selection of the model will depend on: first, the model parameters available at the moment of the simulation request, and second, the technical and operational constraints that the model may involve. If the desired model does not exist in the DT model, the simulation architect will create a block with the stereotype *"SimModel"* (created in the MIC profile). This model will automatically appear in the MIC table created in Cameo Systems Modeler, thanks to the diagram customization option used to generalize the MIC table columns, that represent the MIC attributes. Then, the simulation architect needs to fill the MIC table with model MIC attributes, and update the MBSE DT model. A Part of the MIC table is represented in figure 10, where the required models are selected. Some of the motor models are FMU (Functional Mock-Up [9]) blocks, received from the actuator suppliers. These FMU models can be integrated in the DT model represented in the Simulink environment.

The MIC table contains all the necessary information that the simulation engineer may need in order to execute the simulation.

Step 3: Simulation Execution

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

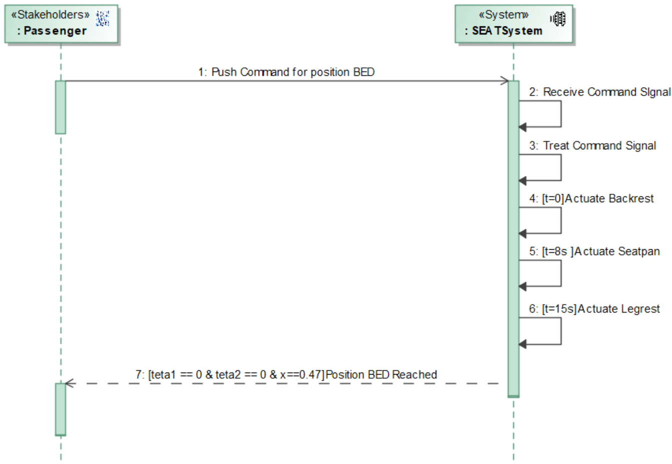


Fig. 9: Use Case Scenario of simulation

For each architecture configuration, after receiving the MICs information, the simulation engineer who is responsible for the testbench DT developed in Matlab/Simulink, instantiates the DT to make the necessary modifications on the model instance. Then, he selects the models according to the MIC table, knowing that the DT model, powered by variability management in Simulink, contains a library of seat models. The simulation engineer can manage and choose the variants according to the desired seat architecture. If all the models exist in the DT models library, he can execute the simulation after linking all the selected models according to the seat functional causality. Otherwise, he will need to develop the missing models with respect to the MIC specification, defined by the simulation architect, and update the DT simulation model. The simulation plots in figure 11 show the results of one configuration, using the referenced actuator characteristics (DC Linear Motor P/N 24189), integrated into the seat environment. The plots show that the seatpan reaches the BED position in around 21s with a response time less than 5s, and a static error almost equal to 0. For this solution, the results satisfy the defined requirements (fig. 7). After executing simulations for each configuration, with different solutions, the simulation engineer sends the simulation plots back to the simulation architect, who makes a results report that groups all the solutions. The systems engineer can use this report to decide on the right architecture to be selected.

V. Conclusion and Future Work

This paper has discussed the requirements verification and trade-off issue, that leads to simulation needs in the early development phases. The proposal put forward here to establish a collaboration framework between systems engineer and simulation engineer through the role of the simulation architect, and the MIC concept. The goal is to reduce missunderstanding between the different actors, and to ensure the traceability of changes, in order to support trade-off decision making. The simulation is ensured using modular testbench DT models, that contain a library of multi-physics models of BC aircraft seats.

The paper presents only the simulation results of one configuration. However, the simulation has been executed for all defined configurations in order to be able to choose the solution which respects the defined requirements, and to evaluate the contribution of our proposed methodology.

We can see that the consistency between system architecture models and DT models, ensured by the collaboration methodology developed, has made the trade-off process much clearer and easier, by proposing unambiguous flows exchange between the test request, the models selection, and the simulation execution.

In future work, the variability management needs to be established in a dedicated tool (PureVariant). Moreover, the testbench physical system is being manufactured so that DT models can be adjusted using real-time synchronization with the testbench physical system. Furthermore, the models selection in the MIC table during step 2 of the process is a task that needs to be automatized, especially when the system contains many components.

The simulation was executed with the aim of verifying the seatpan actuator performances, derived from the whole system performance, through the functional causality and physics transformation. For certain complex systems, however, it is not straightforward to derive component performances from the global performance of the system. Thus, a more detailed methodology needs to be established in future work to address this issue.

References

[1] Beuche, D. (2019, September). Industrial variant management with pure variants. In Proceedings of the 23rd international systems and software product line conference-volume b (pp. 37-39).

[2] Boggero, L., Ciampa, P. D., and Nagel, B. (2022). An MBSE architectural framework for the agile definition of complex system architectures. In AIAA Aviation 2022 Forum (p. 3720).

[3] Bouhali, I., Martinez, J., Idasiak, V., Choley, J-Y.,Mhenni, F., Palladino, L., and Kratz, F. (2023). Early Validation Methodology Integrating Digital Twins In Mechatronics Systems Development Process. IEEE AISM23 Mecatronics.

[4] Bruggeman, A. L., van Manen, B., van der Laan, T., van den Berg, T., and La Rocca, G. (2022). An MBSE-Based Requirement Verification Framework to support the MDAO Process. In AIAA AVIATION 2022 Forum (p. 3722).

[5] De Saqui-Sannes, P., Vingerhoeds, R. A., Garion, C., and Thirioux, X. (2022). A taxonomy of MBSE approaches by languages, tools and methods. IEEE Access, 10, 120936-120950.

[6] Friedenthal, S.,Griego, R.,Sampson, M.(2007, June). INCOSE model based systems engineering (MBSE) initiative. In INCOSE 2007 symposium (Vol. 11).

[7] Galisson, G., Hammadi, M., Gherib, M., and Choley, J. Y. (2022, April). Model-based Systems Engineering methodology for defining multi-physics simulation models. In 2022 IEEE International Systems Conference (SysCon) (pp. 1-8).

[8] Grieves, M., Vickers, J. (2017). Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems. In: Kahlen, J., Flumerfelt, S., Alves, A. (eds) Transdisciplinary Perspectives on Complex Systems. Springer, Cham. https://doi.org/10.1007/978-3-319-38756-7-4

[9] https://fmi-standard.org/

[10] IRT SystemX (2020). Model Identity Card (MIC): Towards a standardization of the specification and description of simulation models. https://mic.irt-systemx.fr

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.

Simulation Models MICs: **Modeling Field:** Mechanic Hydraulic Electronic Electric Control Thermal Robotic Dynamic Mechatronic

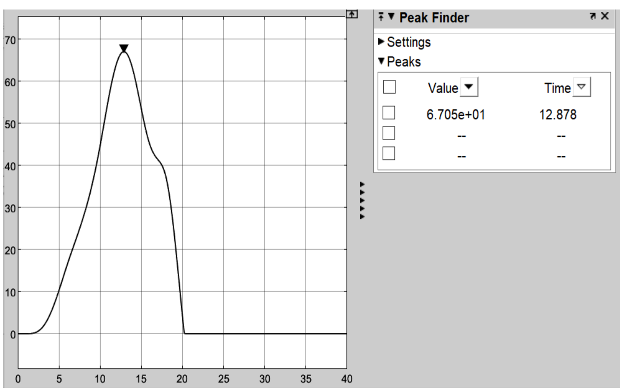
#	Nom	«General Information» Name	«General Information» Life cycle phase	«General Information» Owner	«General Information» Version	«General Information» Confidentiality	«General Information» Granularity
1	Backrest P/N 1679 Model	1679B.slx	Operation	Imane Bouhali	V0	C2	Component
2	DC Linear Motor P/N 24189 Model	24189DC.fmu	Operation	Imane Bouhali	V1	C3	Component
3	DC Rotational Motor P/N 12678 Model F1	12678DCF1.slx	Operation	Imane Bouhali	V1	C2	Component
4	DC Rotational Motor P/N 12678 Model F2	12678DCF2.slx	Operation	Imane Bouhali	V0	C2	Component
5	DC Rotational Motor P/N 13578 Model	13578DC.fmu	Operation	Imane Bouhali	V1	C3	Component
6	Kinematic Link P/N 21678 Model	21678J.slx	Operation	Imane Bouhali	V0	C2	Component
7	Legrest P/N 51785 Model	51785L.slx	Operation	Imane Bouhali	V0	C2	Component
8	Seatpan P/N 3560 Model	3560S.slx	Operation	Imane Bouhali	V0	C2	Component

(a) First part of the MIC table

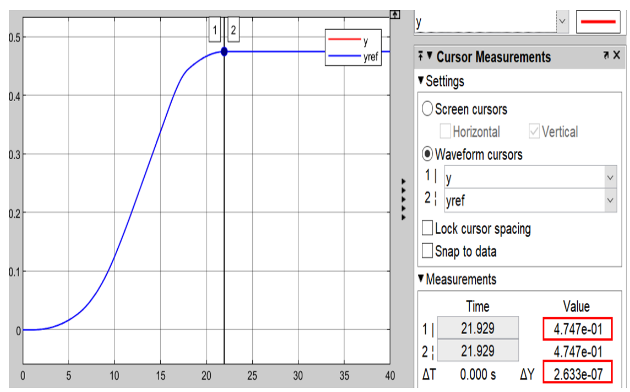
#	«Software» Language	«Software» Version	«Software» Required Simulation Tool	«Software» Required Librairies	«Software» File Format	«Modeling Choice» Modeling Field
1	Simulink	R2022b	Matlab	Simscape Multibody	.slx	Mechanic
2	Simulink	R2022b	Matlab	Simscape Electrical	.slx	Electric
3	Simulink	R2022b	Matlab	Simscape Electrical	.slx	Electric
4	Simulink	R2022b	Matlab	None	.slx	Electric
5	Simulink	R2022b	Matlab	Simscape Electrical	.slx	Electric
6	Simulink	R2022b	Matlab	Simscape Multibody	.slx	Mechanic
7	Simulink	R2022b	Matlab	Simscape Multibody	.slx	Mechanic
8	Simulink	R2022b	Matlab	Simscape Multibody	.slx	Mechanic

(b) Second part of the MIC table

Fig. 10: Extract from the MIC Table created in Cameo Systems Modeler



(a) Actuator Angular Speed



(b) Seatpan Linear Position

Fig. 11: Simulation Results: Performances Verification

[11] Krob, D. (2022). Model-based Systems Architecting: Using CESAM to Architect Complex Systems. John Wiley and Sons.

[12] MathWorks, 2021. How Engineering Teams adopt Model-based Design. Technical report. Retrieved from: uk.mathworks.com/content/dam/mathworks/white-paper/how-engineering-teams-adopt-model-based-design.p

[13] Nowodzienski, P. and Navas, J. (2023), From Model-based to Model and Simulation-based Systems Architectures—Achieving Quality Engineering through Descriptive and Analytical Models. INSIGHT, 26: 40-50. https://doi.org/10.1002/inst.12428

[14] Object Management Group (2018) SysML Extension for Physical Interaction and Signal Flow Simulation, version 1.0.

[15] Parnell, G.S. (Ed.). (2016). Trade-off analytics: creating and exploring the system tradespace. John Wiley and Sons.

[16] Parnell, G.S, Shallcross, N., Specking, E.A, and Phillips, M. (2021). MBSE Enabled Trade-Off Analyses. INCOSE International Symposium. 31. 1406-1416. 10.1002/j.2334-5837.2021.00909.x.

[17] Parnell, G.S., Shallcross, N.J., Specking, E.A., Pohl, E.A., Phillips, M. (2022). Role of Decision Analysis in MBSE. In: Madni, A.M., Augustine, N., Sievers, M. (eds) Handbook of Model-Based Systems Engineering. Springer, Cham. https://doi.org/10.1007/978-3-030-27486-3-14-1

[18] Ritter, C. and Rhoades, M. (2023), Incorporating Digital Twins In Early Research and Development of Megaprojects To Reduce Cost and Schedule Risk. INSIGHT, 26: 57-65. https://doi.org/10.1002/inst.12457

[19] Singh, M., Srivastava, R., Fuenmayor, E., Kuts, V., Qiao, Y., Murray, N., and Devine, D. (2022). Applications of digital twin across industries: A review. Applied Sciences, 12(11), 5727.

[20] Sirin, G., Paredis, C. J., Yannou, B., Coatanéa, E., and Landel, E. (2015). A model identity card to support simulation model development process in a collaborative multidisciplinary design environment. IEEE Systems Journal, 9(4), 1151-1162.

[21] Zhang, X., Wu, B., Zhang, X., Duan, J., Wan, C., and Hu, Y. (2023). An effective MBSE approach for constructing industrial robot digital twin system. Robotics and Computer-Integrated Manufacturing, 80, 102455.

Authorized licensed use limited to: Tsinghua University. Downloaded on February 24,2025 at 08:51:05 UTC from IEEE Xplore. Restrictions apply.