



HAL
open science

Model-based Systems Engineering methodology for defining multi-physics simulation models

Gregoire Galisson, Moncef Hammadi, Mehdi Gherib, Jean-Yves Choley

► To cite this version:

Gregoire Galisson, Moncef Hammadi, Mehdi Gherib, Jean-Yves Choley. Model-based Systems Engineering methodology for defining multi-physics simulation models. 2022 IEEE International Systems Conference (SysCon), Apr 2022, Montreal (Virtual ON LINE), Canada. pp.1-8, 10.1109/SysCon53536.2022.9773905 . hal-03790807

HAL Id: hal-03790807

<https://hal.science/hal-03790807v1>

Submitted on 8 Jun 2023

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.

Model-based Systems Engineering methodology for defining multi-physics simulation models

Grégoire GALISSON
Quartz, ISAE-Supméca
Saint-Ouen, FRANCE
gregoire.galisson@isae-supmeca.fr

Moncef HAMMADI
Quartz, ISAE-Supméca
Saint-Ouen, FRANCE
moncef.hammadi@isae-supmeca.fr

Mehdi GHERIB
THALES SIX
Gennevilliers, FRANCE
mehdi.gherib@thaligroup.com

Jean-Yves CHOLEY
Quartz, ISAE-Supméca
Saint-Ouen, FRANCE
jean-yves.choley@isae-supmeca.fr

Abstract—Due to the increasing complexity of the systems to be developed, it is necessary to proceed to an ever more precise modeling. The design of products delivered to customer are driven by industrial criteria grouped under the acronym SWaP-C (Size, Weight and Power and Cost). They must be the smallest, lightest, most energy-efficient, least expensive and most efficient as possible, with even shorter lead times and limited budgets. We identified two main problems in the design of complex systems, the need of a clear interface between Systems Engineering and Model Providers, and an adequate consideration of multi-physics and multi-component interactions. This paper suggests a methodology developed to ensure a consistency between the system architecture and models of physics simulation. This methodology creates a bridge between MBSE and physical simulation discipline by implementing an additional diagram/viewpoint on the ARCADIA method. This viewpoint will specify the models, organize model building and give stakeholders a common document on which they can exchange. It also allows being exhaustive on the interfaces and the physics involved. MBSE allows the understanding of customer needs, an exhaustive definition of system functionalities and setting up a physical architecture that serves as a reference for the construction of the physical simulation model architecture.

Index Terms—Multi-physics, MBSE, Systems Engineering, Model Architect, Simulation, Simulation Architect

I. INTRODUCTION

Today, in order to respond to the constraints of an increasingly competitive market and a need for more precise modeling of complex systems, simulation is an essential tool. It allows the fine exploration of equipment in their final environment, the validation of the design against the identified risks and it limits costs, by making the design converging more quickly and by allowing the reduction of physical loops of prototypes.

In the industry, commonly, simulations often involve one physic at a time (thermal, mechanical, electrical or even control command) and for a steady state scenario. However, if the impact of other physics is not considered in simulations, this can bring partial results and unexpected failures for the final product.

In this context, the concept of Digital Twin (DT) is emerging. It is the virtual representation of a real system or equipment responding faithfully to various solicitations, whether functional or environmental. The Digital Twin can integrate the modeling and coupling of the different physics (thermal, mechanical, electromagnetic, vibratory dynamics, etc.) involved in its design and can reproduce the behavior of the real system during transient phases. The DT is a tool to assist in bid, architecture, design, validation and integration phases. Systems developed by today's industry involve simultaneously different domains, physics and disciplines. The dedicated methods of systems engineering show benefices in building our digital twins.

System architects and simulation model providers are both major contributors to the specification of product design. System architects define operational scenarios, decision criteria and produce a system architecture. Model providers are the domain experts that simulate the system in its environment. In [1], G. Sirin et al. highlighted the gap between the roles of system architect and model providers, as there is no clear and standardized definition of the model specification. There is a need of a clear interface between Systems Engineering and Model Providers for the design of complex systems, and an adequate consideration of multi-physics and multi-component interactions.

The purpose of this paper is to suggest a solution for the exhaustive specification of physical simulation models. This solution would specify the models as an output of Systems Engineering. All stakeholders would share in continuity the same information, language and documents.

After this introduction, the paper is organized as follows. First, a presentation of the state of the art of Model Based Systems Engineering and Physical Simulation. Then, the explanation of the methodology and the results are detailed. The paper conclusion deals with perspectives and conclusion on the achieved work.

II. STATE OF THE ART

A. Systems Engineering

The objective of Systems Engineering is to manage and control the design of complex systems. It allows engineers to analyze, develop the architecture, specify and then ensure the continuous link between the needs expressed by the customer and the elements that will build the system. It is the guarantee of the overall consistency from a technical point of view. Implemented more than fifteen years ago in the defense, aeronautics and automotive sectors, Systems Engineering has largely proven its effectiveness [2], [3] and [4].

Among Systems Engineering methods, Model Based Systems Engineering (MBSE) is an approach based on the visual representation (model) of the system to be developed, rather than a text document, whose interpretation usually leads to errors. ARCADIA is an MBSE method that allows a solution to be described in a coherent manner between the different levels of engineering. It was developed by Thales in 2007 [5] and [6]. Fig. 1 present the ARCADIA method which extends from the customer business requirements phase to the definition of the physical system architecture. It allows specialty architects to impose constraints and to verify that they are properly taken into account in the solution by the system architect. It ensure that engineering decisions are consistent, all engineering stakeholders share the same information, the same description of the requirement and the product, in the form of a set of shared models and viewpoints [7]. ARCADIA has an associated tool called Capella.

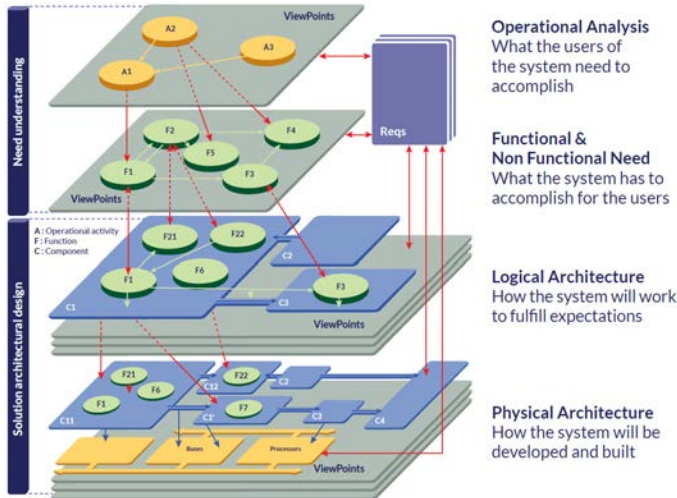


Fig. 1. ARCADIA Methodology - Copyright © Eclipse Foundation

B. Systems Engineering and Simulation

Simulation and model-based systems engineering have proven their effectiveness in the design of complex systems. Many methods have been developed linking these two approaches [8], [9], [10], [11], [12], [13], [14], [15], [4], [16] and [17]. As a result of their research [1] on the gap between the system architect and the model providers, G. Sirin et al. introduced a new actor in the design process called the *Model Architect*. This new role bridges the gap between the

system specification discipline and the physical simulation domain. The *Model Architect* has to express as accurately as possible the needs of the *System Architect*, in order to build a simulation architecture of models. He also has to communicate efficiently with the *Model Providers* (supplier of the simulation model) to obtain the expected model(s). Model Architects have a multidisciplinary view of a product and knowledge of simulation. They also have a comprehensive list of system-level requirements, and how models of different physics and components should interact with each other.

C. Model Identity Card

In addition to the creation of a new role, G. Sirin et al. created a new concept, the Model Identity Card (MIC) [18]. The MIC is a tool used to standardize the specification of a simulation model. It is a meta-model that includes important information about the simulation, such as, simulation assumptions, behavior and interface specification... It is used to help model providers building the simulation model by understanding the model architect expectations [19]. The main benefit from this work is the time and effort reduction in the interface specification and the reduction of misunderstanding errors. A MIC example could contain the following meta-data (see Fig. 2):

Attributes	Remarques	Type	Example	Main Class
Generic Name *	Physical component regroupment	String	Engine	Object Description
Specific Name *	Unique identifier	String	Compressor	
Granularity Level *	List(System/Sub-system/Component)	String	Sub-System	
Developer Name *		String	F.Ravet	
Model Version no. *	x.x format	Float	0.1	
Creation Date		Date		
Documentation	References to technical report	String		
Image	References to an image	Image		
Tool Name*	List (Amesim, Matlab Simulink, GT-Power, Open Modelica...)	String	GT-Power	
Tool Version*	x.x format	Float	7,3	
Operating Systems *	Windows, Linux		Windows	Usage
Name of Compiler *	List (INTEL C/C++, INTEL FORTRAN, Visual C++ 2008 express, Visual C++ 2010 express, Visual Studio C++ 2008, Visual Studio C++ 2010, Visual Basic 6, Visual Basic 5, GCC, LCC, Compileur fortran)	String	Visual C++	
Time Computation*	List (Steady State, Elapsed Time / Real Time)	String	Elapsed Time	
Scalability*	List (Yes/No)	String	Yes	
Hardware Requirements*	CPU, OS etc...	String	Win7, 64bit	
Model Dimension	List (OD-3D, mixte)	String	1D	
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, Milli-second, Hour, Steady state)	String	Second	
Linearity	List (No/Yes)	String	No	Model Quality
Model Behavior	List (Continuous, Discrete, Mixed)	String	Yes	
Validation of models	1-5 (ref NASA quality metrics)	Float		
Verification of models	1-5 (ref NASA quality metrics)	Float		
Input pedigree	1-5 (ref NASA quality metrics)	Float		
Uncertainties	1-5 (ref NASA quality metrics)	Float		
Robustness	1-5 (ref NASA quality metrics)	Float		
History	1-5 (ref NASA quality metrics)	Float		

Fig. 2. MIC Classes and their attributes [9]

Multiple works have already used the Model Architect and MIC concepts [1], [9], [10], [17], [18], [20], [21] and

[22]. Leroux-Beaudout et al. [21] have shown that applying systems engineering principles to Physical Simulation Systems is possible and meaningful. However, those methods does not give a viewpoint of the model architecture.

D. Interactions analysis

Systems developed are composed of several components that react to several physics. This complexity and multidisciplinary aspect make the consideration of interaction a major issue in the system design. Retho et al. have worked on a method [20] to support complex system design via system interaction analysis. This method highlights the importance of taking into account the interactions between components and physics. The Chan matrix [23] is a tool to verify that every interaction between component and physics is considered. It helps the model architect to identify and classify all the interactions between components (see Fig. 3).

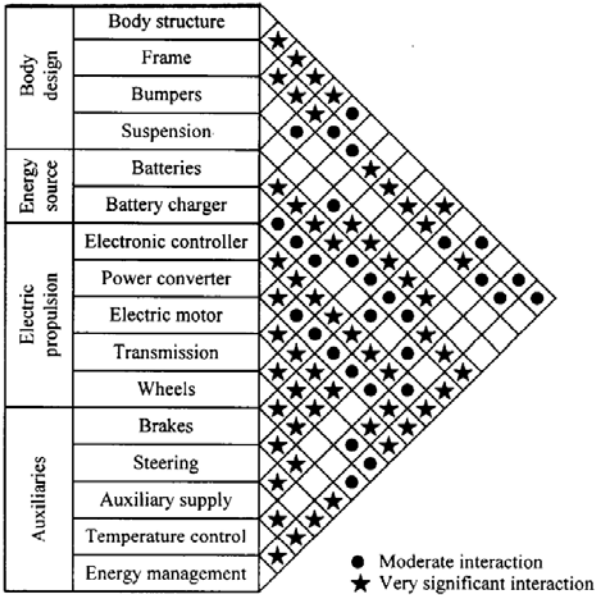


Fig. 3. : Chan Matrix for interactions among EV subsystems [23]

III. PROPOSED METHODOLOGY

Using previous works from literature, we propose a methodology to close the gap between MBSE and physical simulation models in the different phases of the development by tooling the model specification. This methodology uses the MBSE method ARCADIA and its associated tool Capella. The methodology creates a bridge between MBSE and physical simulation discipline by implementing an additional viewpoint on the ARCADIA method. This viewpoint will specify the models, organize model building and give stakeholders a common document on which they can exchange. MBSE allows the understanding of customer needs, an exhaustive definition of system functionalities and setting up a physical architecture that serves as a reference for the construction of the physical simulation model architecture. The first step of this methodology is to follow the ARCADIA method from Operational Analysis to the Physical Architecture for the system to be simulated and evaluated.

A. Operational Analysis

In the ARCADIA Method, the operational analysis answers what the users of system need to accomplish. It focuses on understanding the real customer need, in terms of tasks to be achieved by users. It captures the work that the final user has to perform while using the system by defining the operational capabilities that the system must contribute to (see Fig. 4).

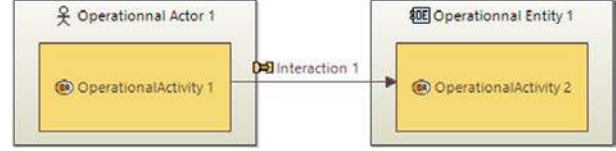


Fig. 4. : Capella diagram Operational Architecture

B. System analysis

The System analysis defines the main functional and non-functional need/expectations for system. It defines what the system has to accomplish for the users. An early system design is developed to check the requirement feasibility (see Fig. 5). The system analysis is meant to show the functional interactions of the system components without identifying the actual components.

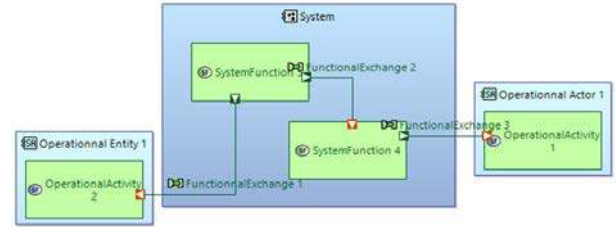


Fig. 5. : Capella diagram System Architecture

C. Logical architecture

A logical architecture is an abstract representation of the system components. Every function of the system can be performed by a corresponding logical component. The logical architecture represents how the system will work to fulfil expectations (see Fig. 6). The logical elements define the system logical architecture that implements the system functional architecture. During the definition of the logical architecture, more logical functions can be identified leading to the creation of more logical elements.

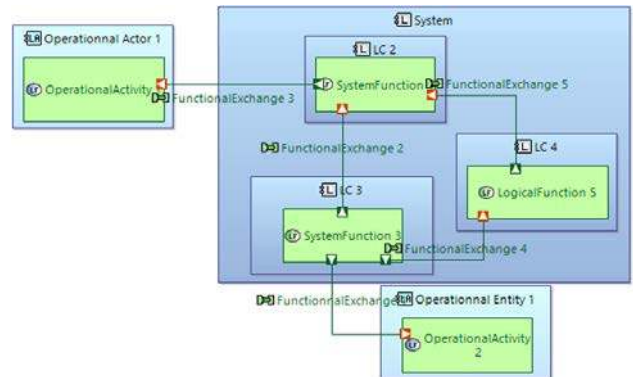


Fig. 6. : Capella diagram Logical Architecture

D. Physical architecture

The purpose of a physical architecture is to develop a technical solution of the logical architecture. In this phase, the logical components are allocated to physical components that will actually perform the internal functions (see Fig. 7). In other words, this architecture shows how the system will be developed and built, a well-defined logical architecture can lead to multiple physical architectures.

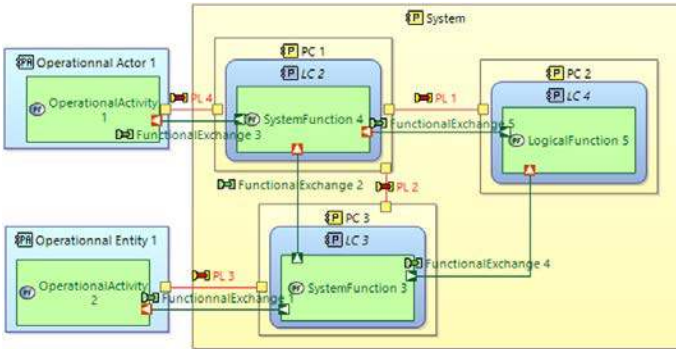


Fig. 7. : Capella diagram Physical Architecture

E. Physical Simulation Model Architecture [PSMA]

The physical architecture identified with the ARCADIA method needs to be evaluated and validated with simulation. To help building the simulation model of the system, an additional viewpoint is introduced in the design process, the Physical Simulation Model Architecture (see Fig. 8). This viewpoint shows how the global model of the system is built and makes a transition from MBSE to the physical simulation domain. The model architect builds this viewpoint. It is the interface between MBSE and the physical simulation domain, it specifies how the models should be built.

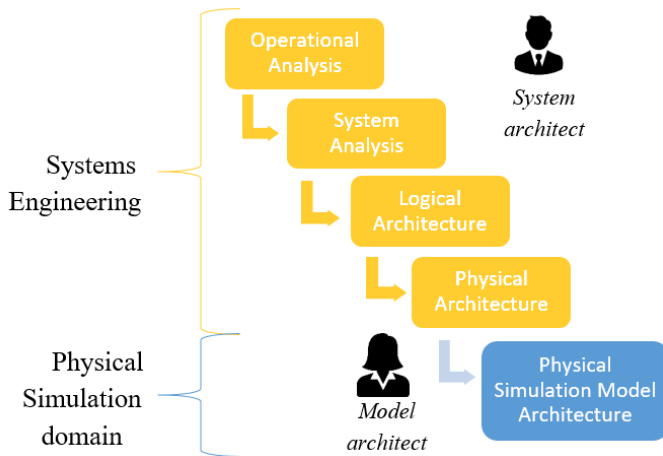


Fig. 8. : Introduction of a new layer and a new role the Physical Simulation Model Architecture and the Model Architect

Currently, this diagram does not exist on ARCADIA and Capella. These tools are only designed for Systems Engineering, they do not allow the transition from MBSE to the physical simulation domain.

F. Transition from Systems Engineering to Physical Simulation domain

The simulation model virtually represents the system behavior in the real world. It needs to act and respond as closely as possible to what the system would do in real life. The Physical Simulation Model Architecture is directly derived from MBSE (see Fig. 9). It contains all the relevant information for the specification of the physical simulation models. Some of this information is transferred from MBSE to the physical simulation models, for example, Physical flux and architecture; Interactions between components; Requirements... The physical architecture from MBSE allow the model architect to specify simulation models in the real-world environment in an exhaustive and precise way.

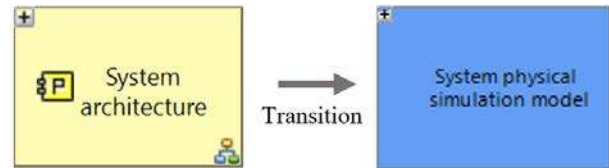


Fig. 9. : Transition from System Engineering to System Physical simulation model

G. Transition of system physical components to simulation components

The System Simulation model is composed of several components like the real system (see Fig. 10). Each physical component from physical architecture is transferred to a simulation model, a Component simulation model. There is an associated simulation model for each physical component.

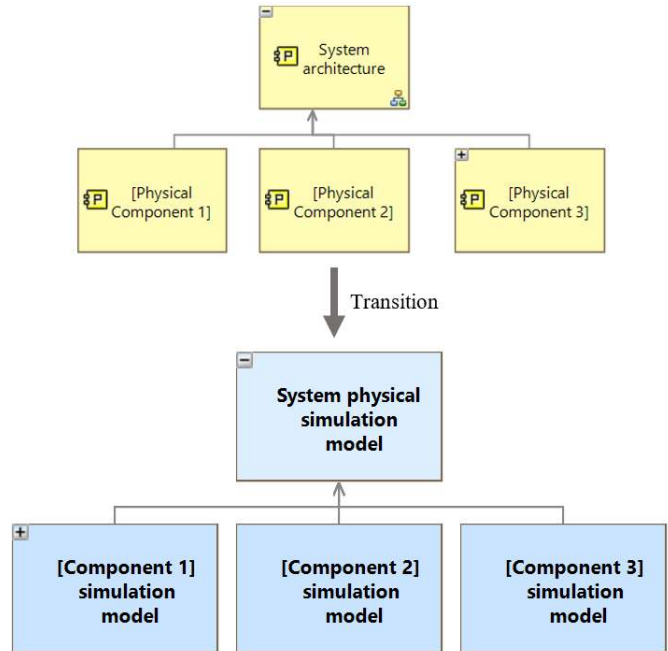


Fig. 10. : Breakdown of the model into component simulation models

H. Breakdown of components into Physical domain simulation models

The system can be submitted to various multiphysical constraints. Every component simulation model is broken down into several simulation models according to the domains (thermal, mechanical, electrical...) that the model architect has identified to be simulated and whose requirements have to be validated (see Fig. 11).

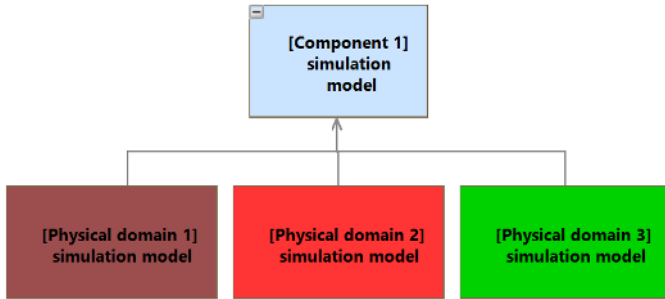


Fig. 11. : Breakdown of component simulation model into Physical domain simulation models

I. Definition of the simulation models using the Model Identity Card

At this stage, the physical simulation model architecture is not done. The simulation models that needs to be built are identified. However, the interactions between components, models and environments are not apparent, as well as what they need to simulate... For each simulation model identified, a Model Identity Card is added to describe and characterize the simulation model that needs to be created (see Fig. 12). The first MIC that needs to be created is the MIC at the system physical simulation model level. The model architect specifies the purpose of the system simulation model. The MIC gives information about what is simulated, on which software, which physics are modeled, what are the assumptions, how is it interfaced, evaluated and validated...

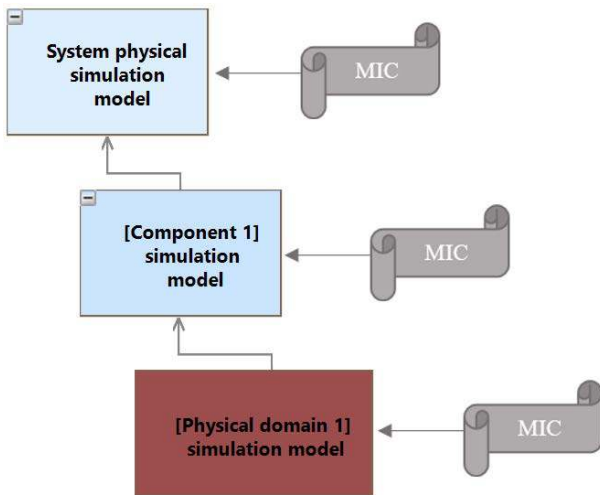


Fig. 12. : MIC associated to each simulation model

The MIC and the simulation layer concept are not yet integrated in Capella. However, thanks to the DSL (Domain

Specific Language) add-on, it is possible to create a profile that will work as a MIC and associate it to each simulation model that needs to be specified (see Fig. 13). Those profiles are linked to the simulations models on the Physical Simulation Model Architecture viewpoint.

General information	Integration	Modeling choice
Name	Tool name	Explicative text
Description	Tool version	Model dimension
Owner	Alternative tool	Modelling field
Life cycle state	Language name	Type of choice
Version	File format name	Behavior specification
Version date	Compiler name	Time scale [s]
Access path	Alternative compiler	
Confidentiality	Numerical compatibility with oth	

Fig. 13. : MIC profile

The physical architecture from MBSE shows the system structure, how the components interact with each other and with the environment. Following this observation, this information can help building the simulation model architecture and the MICs.

J. Building of the physical simulation model architecture

The physical simulation model architecture contains all the interactions between components, environments and physical domains. Viewpoints from Model Based Systems Engineering give most of the exchanges, but not all of them. The Chan matrix [23] can be used to verify that every interaction between components and physics is taken into account.

The physical simulation model architecture is built thanks to the physical architecture from MBSE, the MICs and the Chan Matrix (see Fig. 14). All the interactions and connection between models, components and the environment are modeled.

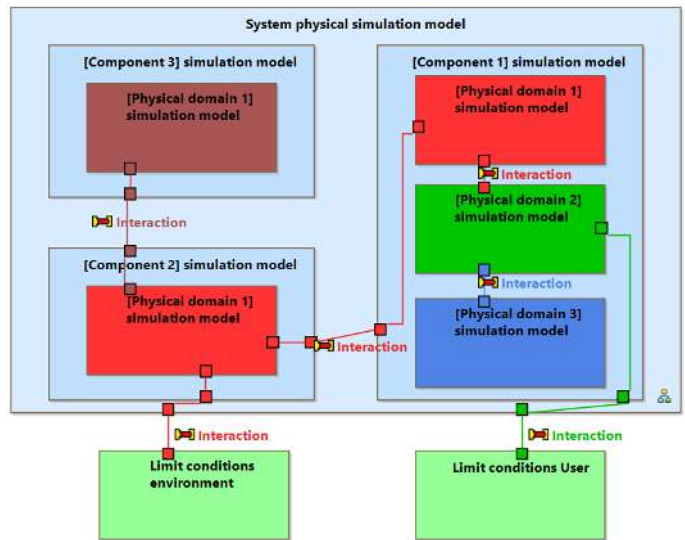


Fig. 14. : Physical Simulation Model Architecture of a system

K. Modelisation of the system using MIC and Capella model information

Once the PSMA is created, the model providers can build the simulation models from the MICs and the PSMA, on the identified software.

IV. APPLICATION OF THE METHODOLOGY

A. Operational Analysis

The case study is a heating fan. The operational actors are a user who wants to warm up and a fresh air cabin (see Fig. 15).



Fig. 15. Results of Operational Architecture diagram

B. System Analysis

This analysis identifies the main functions of the systems (see Fig. 16), which are getting the setpoint given by the user, compare it to the temperature of the cabin, create heat and diffuse it.

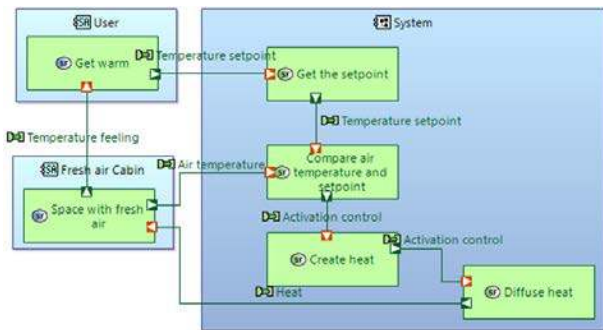


Fig. 16. Results of System Architecture diagram

C. Logical Architecture

The Logical Architecture shows the allocation of functions to logical components (see Fig. 17). Each function is allocated to a logical component, for example, the function get the set point is achieved by a selector switch.

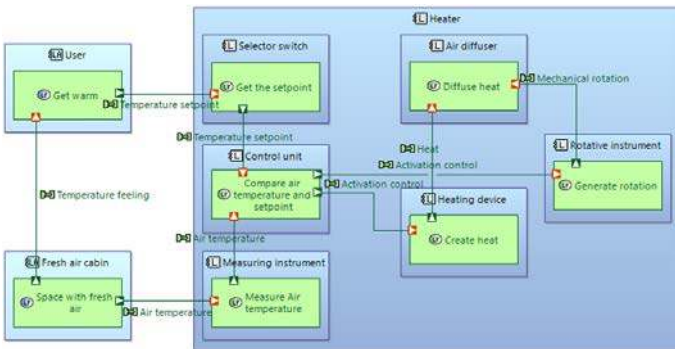


Fig. 17. Results of Logical Architecture diagram

D. Physical Architecture

The Physical Architecture is the allocation of logical components and functions to physical components. Generation of physical links between components (see Fig. 18). For example, the logical component selector switch is identified as a potentiometer in the physical architecture and it is connected to a processor via an analogic cable.

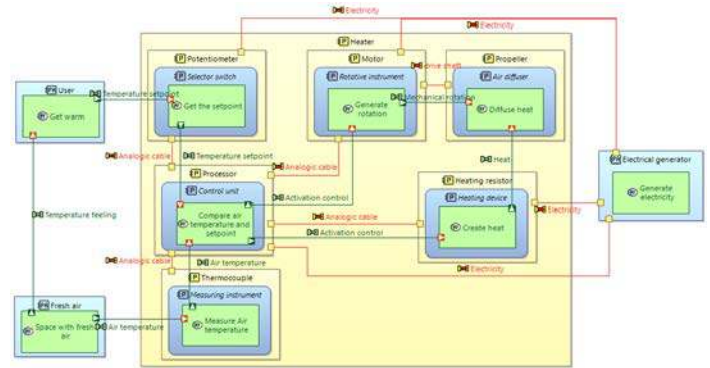


Fig. 18. Results of Physical Architecture diagram

E. Interactions analysis with Chan matrix

The interaction analysis with the Chan Matrix gives all the interactions between components of the system (see Fig. 19). The interaction analysis can also be done to all physics domains identified relevant to be modeled, so that all interactions between them are considered. It gives also the number of interactions for each component or physics.

Propeller	4		X					
Heating resistor	2			X				
Motor	2				X			
Thermocouple	2		X				X	
Processor	4		X					X
Potentiometer	2					X		
User	3		X					
Fresh air	3							

Fig. 19. Chan Matrix results for the interaction analysis of components

F. Breakdown of the System Simulation Model into Components and Physical Domain Simulation models

The Simulation Model Architecture is directly derived from MBSE. Components from the MBSE Physical Architecture are translated into simulation model components, with all the relevant information for the specification of the physical simulation models. Components are broken down into several Physical Domain Simulation models for each Physical domain identified as relevant for the simulation (see Fig. 20).

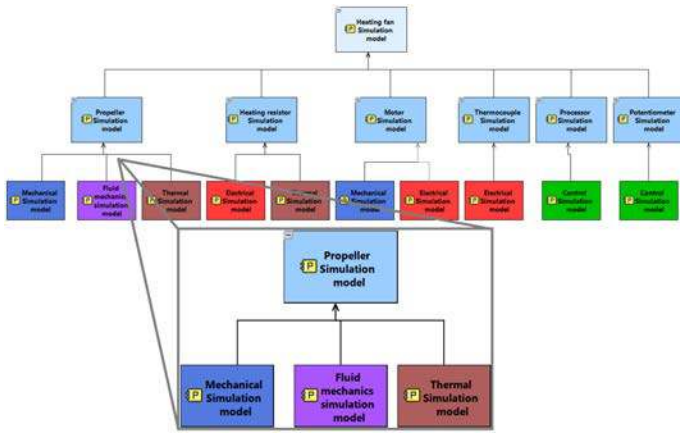


Fig. 20. Results breakdown and translation of physical components

G. Model Identity Card for a simulation model

The model identity cards have to be the most exhaustive as possible in order to consider all the useful information for the model provider. Fig. 21 shows an example of what a MIC might look like and the information it provides.

1 Model	
General information	
Name	Propeller Thermal Simulation model
Description	Thermal Simulation model of the propeller
Owner	Grégoire GALISSON
Life cycle state	to be validated
Version	2.0
Version date	22/10/2021
Access path	\Mymodels\Heating_fan\Prop_Therm_SM
Confidentiality	Internal
Integration	
Tool name	Amesim
Tool version	Simscenter Amesim 2020.2
Alternative tool	Simscape
Language name	C
File format name	.ame
Compiler name	GNU GCC (64bit)
Alternative compiler	Intel C++ / Microsoft Visual C++
Numerical compatibility	Matlab / Excel
Modeling choice	
Explicative text	The propeller is considered as a convectiv...
Model dimension	1D
Modelling field	Thermal
Type of choice	Model convection neglect conduction and ...
Behavior specification	none
Time scale [s]	100 s
Model type	Discrete transient model
Solver name	fixed timestep
Step size [ms]	10 ms
Embedded	true
Verification and validation	
Method	
Documentation	

Fig. 21. MIC Example of the Propeller Thermal model

MICs gives an exhaustive specification of the model with lots of information. For instance, tool integration, modeling field, dimension, numerical variable at ports, physical links...

H. Physical Simulation Model Architecture

The physical simulation model architecture is built with the information from MBSE Physical Architecture, MICs and the Chan matrix (see Fig. 22). Ports and interactions between components, environment and physical domains are modeled.

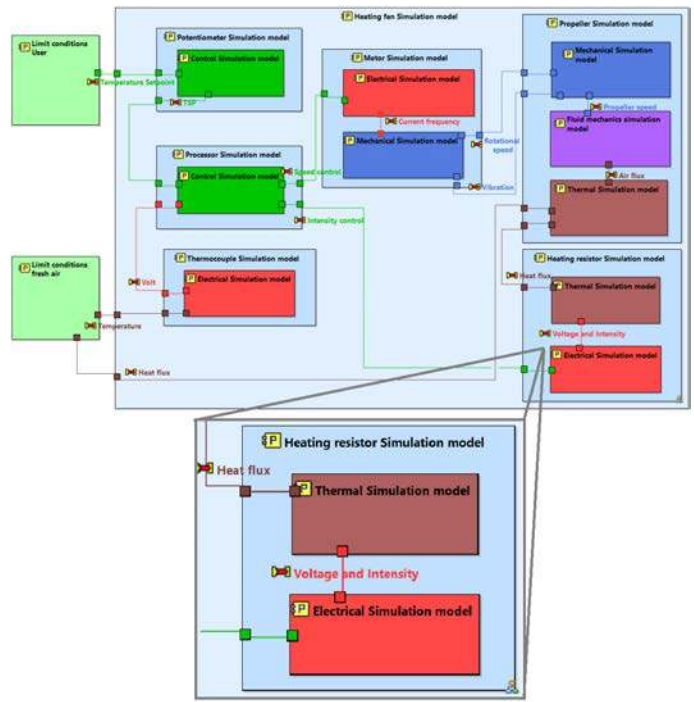


Fig. 22. Example Physical Simulation Model Architecture diagram

I. Model on a given software - Simcenter Amesim

The simulation models are built with the information for the Physical Simulation Model Architecture and the Model Identity Cards. Each component of our system is modeled with the different physics, all interactions between components and physics are modeled, the same number as in the PSMA viewpoint (see Fig. 23). The simulation models and the viewpoint Physical Simulation Model Architecture share the same information.

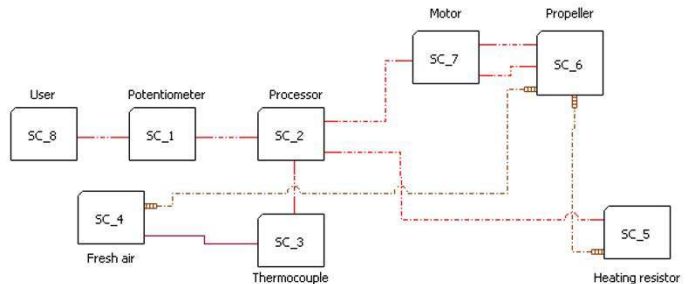


Fig. 23. Simcenter Amesim modeling of the heating fan

V. CONCLUSION

In a highly competitive market, systems and their design are becoming increasingly complex. The consideration of interactions between physics and the interfacing between Model Based Systems Engineering and the physical simulation domain are major issues. The proposed methodology of this paper closes the gap between MBSE and physical simulation models in the different phases of the development by tooling the model specification. Our methodology allows a better specification of complex simulation models by implementing an additional viewpoint, the Physical Simulation Model Architecture. The PSMA is a tool for model providers to build the right simulation models. Results have shown that this method can help building the right models by considering all information from MBSE and the interactions between components and physics. The contribution of this method is to close the gap between MBSE and physical simulation discipline by adding an additional viewpoint for simulation models specification. Future work would be to develop an automated way to translate information from Model Based System Engineering to physical simulation domain.

REFERENCES

- [1] G. Sirin, F. Retho, B. Yannou, M. Callot, P. Dessante, and E. Landel, "Multidisciplinary simulation model development: Early inconsistency detection during the design stage," *Advances in Engineering Software*, 2017.
- [2] C. Haskins, K. Forsberg, M. Krueger, D. Walden, and D. Hamelin, "Systems engineering handbook," in *INCOSE*, vol. 9, 2006, pp. 13–16.
- [3] L. E. Hart. (2015, Consulted the 16/06/2021) Introduction to model-based system engineering (mbse) and sysml. [Online]. Available: <https://www.incose.org/docs/default-source/delaware-valley/mbse-overview-incose-30-july-2015.pdf>
- [4] C. De Tenorio, *Methods for collaborative conceptual design of aircraft power architectures*. Georgia Institute of Technology, 2010.
- [5] I. Eclipse Foundation. (Consulted the 12/07/2021) Open source solution for model-based systems engineering. [Online]. Available: <https://www.eclipse.org/capella/>
- [6] ——. (Consulted the 14/06/2021) Capella. [Online]. Available: <https://wiki.eclipse.org/Capella>
- [7] P. Roques, "Mbse with the arcadia method and the capella tool," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, 2016.
- [8] A. Tolk, C. G. Glazner, and R. Pitsko, *Simulation-Based Systems Engineering*. Cham: Springer International Publishing, 2017, ch. 4, pp. 75–102.
- [9] G. Sirin, "Ingénierie des systèmes basés sur les modèles (mbse) appliquée au processus de conception de simulation complexe: vers une ontologie de la modélisation et la simulation pour favoriser l'échange des connaissances en entreprise étendue," Ph.D. dissertation, Ecole Centrale Paris, 2015.
- [10] F. Retho, "Méthodologie collaborative d'aide à la construction de produits virtuels pour la conception d'aéronefs à propulsion électrique." Ph.D. dissertation, CentraleSupélec, 2015.
- [11] K. Petter, "A model-based methodology for integrated preliminary sizing and analysis of aircraft power system architectures," Ph.D. dissertation, Université de Toulouse, 2008.
- [12] H. Neema, J. Gohl, Z. Lattmann, J. Sztipanovits, G. Karsai, S. Neema, T. Bapty, J. Batteh, H. Tummescheit, and C. Sureshkumar, "Model-based integration platform for fmi co-simulation and heterogeneous simulations of cyber-physical systems," in *Proceedings of the 10th International Modelica Conference; Lund; Sweden, 03 2014*, pp. 235–245.
- [13] F. Mhenni, J.-Y. Choley, O. Penas, R. Plateaux, and M. Hammadi, "A sysml-based methodology for mechatronic systems architectural design," *Advanced Engineering Informatics*, vol. 28, no. 3, pp. 218–231, 2014.
- [14] V. Galtier, S. Vialle, C. Dad, J.-P. Tavella, J.-P. Lam-Yee-Mui, and G. Plessis, "Fmi-based distributed multi-simulation with daccosim," in *Symposium on Theory of Modeling and Simulation - TMS'15*, Alexandria, VA, United States, Apr. 2015, pp. 804–811. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01155707>
- [15] L. Diogo Couto, S. Basagiannis, E. H. Ridouane, E. Zavaglio, P. Antonante, H. Saada, and S. Falleni, "Lessons learned using fmi co-simulation for model-based design of cyber physical systems," in *Leveraging Applications of Formal Methods, Verification and Validation. Distributed Systems*, T. Margaria and B. Steffen, Eds. Cham: Springer International Publishing, 2018, pp. 488–503.
- [16] A. D'Ambrogio and U. Durak, "Setting systems and simulation life cycle processes side by side," in *2016 IEEE International Symposium on Systems Engineering (ISSE)*. IEEE, 2016, pp. 1–7.
- [17] R. Barbedienne, O. Penas, M. Yagoubi, E. Duceau, and E. Landel, "Model of intention for the model specification verification during the simulation architecture design process," in *ISEE 2018 - IEEE International Systems Engineering Symposium*. Rome, France: IEEE, Oct. 2018, pp. 1–8. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02093003>
- [18] G. Sirin, C. Paredis, B. Yannou, E. Coatanéa, and E. Landel, "A model identity card to support engineering analysis model (eam) development process in a collaborative multidisciplinary design environment," *IEEE Systems Journal*, vol. 9, no. 4, pp. 1151–1162, 2015.
- [19] SystemX, *Model Identity Card (MIC): List of information characterizing a simulation model*, 08/12/2020.
- [20] F. Retho, H. Smaoui, J.-C. Vannier, and P. Dessante, "A model-based method to support complex system design via systems interactions analysis," in *Posters Workshop at CSD&M*, 2013, pp. 115–126.
- [21] R. Leroux-Beaudout, "Méthodologie de conception de systèmes de simulations en entreprise étendue, basée sur l'ingénierie système dirigée par les modèles," Ph.D. dissertation, Université Paul Sabatier-Toulouse III, 2020.
- [22] R. Leroux, M. Pantel, I. Ober, and J.-M. Bruel, "Model-based systems engineering for systems simulation," in *International Symposium on Leveraging Applications of Formal Methods*. Springer, 2018, pp. 429–448.
- [23] C. Chan, "The state of the art of electric and hybrid vehicles," *Proceedings of the IEEE*, vol. 90, no. 2, pp. 247–275, 2002.