MBSE and MDAO for Early Validation of Design Decisions: a Bibliography Survey

Jean-Charles Chaudemar, Pierre De Saqui-Sannes

MBSE and MDAO for Early Validation of Design Decisions: a Bibliography Survey

Jean-Charles Chaudemar
ISAE-SUPAERO, Université de Toulouse, France
jean-charles.chaudemar@isae-supaero.fr

Pierre de Saqui-Sannes
ISAE-SUPAERO, Université de Toulouse, France
pdss@isae-supaero.fr

***Abstract*—Switching from document-centric engineering to Model Based Systems Engineering (MBSE), Systems Engineering (SE) has significantly evolved in terms of standard practices for the design of complex, interdisciplinary systems. MBSE consists in a top-down, model based approach to describe the entire system focusing on different points of view that cover at least structural and behavioral descriptions. Over the past decade, the need to perform an engineering analysis in the early steps of the system's life cycle has opened avenues for joint use of MBSE and Multidisciplinary Design Analysis and Optimization (MDAO). MDAO is fully dedicated to Analysis and Optimization: the model is restricted to a single aspect of the system that is described in details in a formal language that will be the input of the associated computing tool. This paper surveys and categorizes MBSE and MDAO approaches for better understanding of how MBSE and MDAO can be associated in a systems engineering project. Lessons learned from this literature survey will be used in the framework of French project Concorde. One major expected achievement of the project is to design and implement a methodology to populate parts of the MDAO modeling approach directly from the MBSE one, applied to a UAV case study.**

***Index Terms*—Modeling, Multi-Paradigm, MBSE, MDAO.**

I. Introduction

The increasing complexity of aerospace systems has stimulated research work on models that enable early detection of design errors in the life cycle of these systems. Two complementary approaches exist: MBSE (Model Based System Engineering) and MDAO (Multidisciplinary Design Analysis and Optimization).

MBSE is a top-down, model based approach to describe the entire system focusing on different points of view that cover at least structural and behavioral descriptions. This approach is mainly supported by informal and semi-formal languages that often do not directly offer automated reasoning capabilities.

Conversely, MDAO is fully dedicated to Analysis and Optimization. The model is restricted to a single aspect of the system that is described in details in a formal language that will be the input of the associated computing tool.

This paper surveys major MBSE and MDAO approaches and compare them in terms of language, tools, and methods. From a survey of the literature, this paper enumerates the expected benefits of joint use of MBSE and MDAO.

This work has been supported by the Defense Innovation Agency (AID) of the French Ministry of Defense (research project CONCORDE N 2019 65 0090004707501).

This paper is organized as follows. Section II presents an overview of MBSE approaches by focusing on its assets for the description of complex systems. MBSE entails a relationship between methods and tools. Section III addresses MDAO approaches, which are a key enabler for early design decisions and trade-offs. Section IV associates MBSE and MDAO by discussing actual synergies. Section V concludes the paper.

II. Model-Based Systems Engineering

A. Systems Engineering

Systems engineering (SE) supposedly appeared during World War II, when L. D. Miles, J. Leftow, and H. Erlicher, three engineers at General Electric, noticed that many of the substitutes had better or equal performance, often at reduced cost, as compared with legacy components. That led to the foundation of the concept of 'value analysis'. 'Value analysis is a philosophy implemented by the use of specific set of techniques, a body of knowledge, and a group of learned skills. As an organized creative approach aims to identify unnecessary costs, *i.e.*, costs which provide neither quality nor use nor life nor appearance nor customer features' [1]. In the following decades until now, the necessity of value analysis has become a systematic process, even for any socio-technical complex systems. The advent of the computer along with the cybernetics strongly increases the need of stringent design and analysis.

The manifold definitions of SE began to be formalized in the 1960s with NASA's Apollo Program and the US military standards [2], [3]. They all highlight the principles of holistic, interdisciplinary, or interrelationships, together with the concepts of system life-cycle and requirements [4]. International Council on SE (INCOSE) works out the classical definition of SE by consensus: 'An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customers needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem' [5].

The SE discipline is considered as the application of the systems science for the sake of grounded and pragmatic multidisciplinary engineers. In general, the new socio-technical challenges are no longer a matter of solving problems in a silo perspective by accounting for one discipline or one technology. They involve the practice of the systems thinking 'for seeing the wholes, relationships, patterns of change rather that static snapshots' [6], [7]. System engineers need to use systems engineering processes in order to act in a more creative and efficient way. According to Senge [6], this systems thinking requires a 'specific set of tools and techniques'. Those tooled methods and soaring digitization in industry have marked a turning point in the evolution of SE towards the Model-Based Systems Engineering (MBSE). Thus the emerging MBSE approach is about to supplant the classical, document-centric systems engineering approach.

B. MBSE and modeling

The main assets of MBSE reside in its power of abstraction and its graphical representation. On the one hand, an appropriate model abstracts reality and represents some aspects of the real world considered as important by the modeler [8]. According to Laing's classification of models, their degree of abstraction might enable to classify them into a physical, geometric, or mathematical category [9]. The models tackle the complexity of their subject and make it understandable. They also allow to reason in order to anticipate avoidable consequences due to inappropriate actions on the concerned real system.

On the other hand, the visualization and the graphical facet of some models are of upmost importance. In the context of human factors and human interaction, visualization is a powerful means to transfer information [10]. Visualization relies on two main processes, *i.e.*, the presentation (what shape the data will be given) and the perception (how the beholder will grasp the data) [11]. In order to minimize the effort of the human mind to understand a graphical model, Moody gives nine principles, such as 'semiotic clarity', 'visual expressiveness', and 'complexity management' [12].

A survey of the literature indicates that MBSE approaches can advantageously compare with one another by relying on a triptych: (languages, tools, methods). The remainder of this section is structured accordingly.

C. Modeling Languages

Figure 1 categorizes modeling languages into (1) semi-formal and diagrammatic languages, and (2) formal methods.

1) Semi-formal Languages: Many papers address MBSE through an association with SysML. They discuss how SysML's inherits from UML and why this is an advantage or not. They also compare SysML-based methods with Arcadia. It is worth to be noticed that MBSE cannot be restricted to SysML. Other main modeling languages are listed below.

- SDL (*Specification and Description Language*) [13] from the telecommunications community. SDL describes a protocol architecture and protocol machines. SDL is commonly coupled with MSC (Message Sequence Charts) that depict messages exchanges in the form of scenarios.
- VHDL-AMS has been designed with electronic systems in mind [14].
- The Mathworks languages (Matlab, Simulink, Stateflow and Simscape) [15].

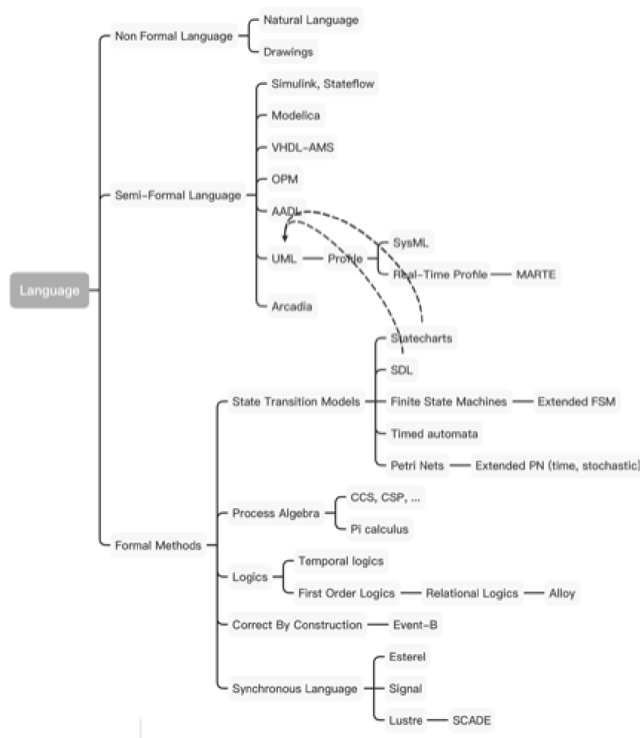


Fig. 1. Modeling languages for MBSE

- Modelica, for mechatronic systems [16].
- SCADE for critical software [17].
- AADL (*Architecture Description Language*) [18].
- OPM (*Object Process Methodology*) [19].

2) Formal Methods: Formal methods have a mathematically defined semantics that make models unambiguous. Many criteria could be applied to compare formal methods. Given the expected benefits of using formal methods include the capacity to analyze models for early detection of design errors in the life cycle of systems, it is possible to distinguish two kinds of formal methods:

1) Formal methods that support a priori validation. Example include models that implement a sate/transition paradigm. Examples include Extended Finite State Machines, Petri Nets [20], timed automata [21], and Labeled Transition Systems [22]. The model is built up and subsequently checked against its expected properties. The latter can be expressed using logic, *e.g.*, temporal logic.
2) Formal methods that support a correct by construction approach. Event-B is a refinement-based formal method for modeling and analysis of complex safety-critical systems [23]. A system is modeled in Event-B as a collection of state variables and events (guarded actions) that act on variables. System properties are specified as invariants on state variables and are formally verified by deductive proof. The key mechanism of the method is refinement, *i.e.*, incremental development from an

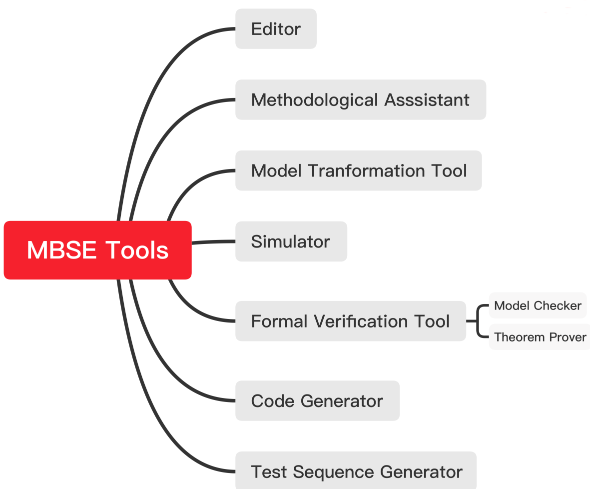


Fig. 2. Modeling tools for MBSE

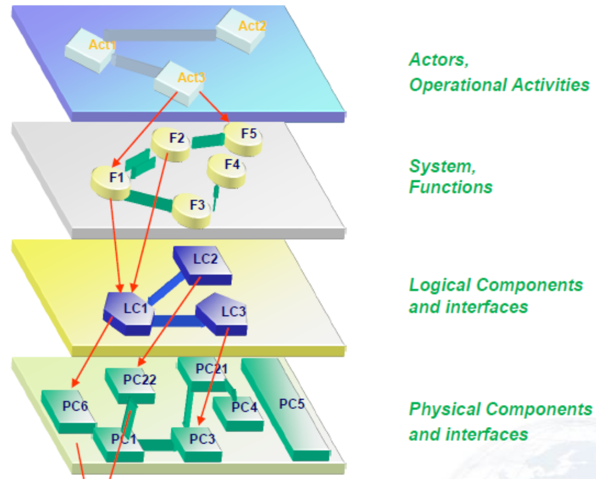


Fig. 3. Arcadia approach in Capella [40]

abstract model to a concrete one, that splits the complex task of formal verification into manageable proofs, allowing to detect the errors as soon as they are introduced along the modeling process. This is the principle of correct modeling by construction. The extensible Rodin platform aids this process with automatic provers and additional modeling features from third-party plug-ins [24]. Models can also be validated by ProB model checker and Animator tool [25].

D. MBSE Tools

In [26] Reilly, Edwars, Peak and Mavris state that 'model-based methods must provide more than just descriptive information'. Models become references for early debugging of design errors in the first steps of the system's life cycle [27], for automated code generation [28], and for automated generation of test sequences [29].

Figure 2 categorizes tools that may be integrated into a MBSE approach.

In [30] Buchiarone, Cabot, Paige, and Pierantonio identify grand challenges in model-driven engineering. Since the early times of MBSE, scalability has been denoted as one of the main challenge. Scalability remains a great challenge when joint use of MBSE and MDAO is at stake. In [30], the authors further advocate for Artificial Intelligence (AI) techniques to cognify model driven techniques, *e.g.*, smart model auto-completion.

E. Methods

The many methods for inclusion in MBSE approaches can be categorized into four groups:

1) Methods compliant with a standard which is a reference for systems engineering. Examples include ANSI EIA632 [31], IEEE 1220 [32], and IEC 15288 [33].
2) Methods compliant with a standard which is a reference for an application domain, such as ARP 4754A [34] for aeronautics [35].
3) Methods developed for specific tools whilst remaining applicable to a broad variety of systems. For instance, [27] associates a method with SysML and free software TTool [36], and discusses application to drones.
4) Methods non initially developed for MBSE can be extended with MBSE features. [37] extends the STPA (*Systems Theoretic Process Analysis*) method with SysML and TTool, in particular to benefit from the model formal verification approaches supported by TTool. Another example of associating a SysML method with another method is discussed in [38] for the Formose project that associates SysML with KAOS.

Next section focuses on Arcadia, a method initially promoted by Thales and supported by Capella tool.

F. Arcadia/Capella

Arcadia describes the detailed reasoning necessary to understand the real customer need, define and share the product architecture among all engineering stakeholders, validate its design and justify it, early on, and ease a master integration, validation and verification [39].

Arcadia intensively relies on functional analysis. It introduces four engineering engineering perspectives (Fig. 3):

- Operational analysis,
- System analysis,
- Logical analysis, and
- Physical analysis.

By doing so, Arcadia promotes a clear distinction between the expression of the need, which is covered by the first two perspectives, and the expression of the solution which is addressed by the last two perspectives.

In [39] Navas, Tannery, Bonnet and Voirin present a tailored version of Arcadia method that distinguishes between System Analysis and Logical Analysis.

- System analysis includes two tasks

1) `Identification of Stakeholders and Interfaces`. It identifies the stakeholders and how the system interacts with them.
2) `Definition of external functions`. It determines that the system ensures the functions. Operational states and stakeholders are both considered to define how the system shall interact with the stakeholder at a given state and derive the external functions.

- Logical analysis defines the requirements over the subsystems and components that will be actually built, commissioned and operated. Logical analysis includes three tasks.

1) `Derivation of external functions into internal functions`. It exhaustively consists in decomposing the system's external functions into internal, less complex functions that together specify how the system's functions are achieved.
2) `Allocation of internal functions`. It allocates the internal functions and associated requirements to the components of the reference architecture.
3) `Consolidation through dynamic scenarios`. It aims at ensuring the exhaustiveness of the design by challenging the design with operational scenarios and checking that the existing functions and required properties specify the expected behavior of the system.

G. MBSE: acceptance and feedback

SE may suffer a lack of complete acceptance in terms of scientific and technical fields. SE is sometimes perceived as a fuzzy mixing between 'soft' sciences such as sociology, or science of interrelationships, and 'hard' sciences such as mathematics, or physics. Even though SE relies on scientific principles of complex systems based on a few laws (*e.g.*, complexity theory, chaos theory, nonlinear dynamics) [41]. SE discipline knows a weak breakthrough in engineering training programs. For a few students, SE methodology is derogatorily labelled as merely 'common sense' in a practical way.

Another pitfall for SE application is still a lack of metrics for a shared Return On Investment (ROI) in industry. In [5], INCOSE has demonstrated in two studies SE's effectiveness and ROI perspective. However, in a number of companies, due to the corporate culture, change management might require a greater endeavour and much time.

Practically, MBSE aims at fixing these drawbacks. Its reusability, its relationship and connection with models related to the system's physics - particularly Multidisciplinary Design Analysis and Optimization (MDAO) models - let MBSE build up its strengths.

III. MULTIDISCIPLINARY DESIGN ANALYSIS AND OPTIMIZATION

This section addresses MDAO with system early-design in mind. At this point, it is essential to distinguish between design models and analysis models. A design model describes the system that it intends to represent, whereas an analysis model aims to demonstrate properties or outcomes of the intended system [42]. Accordingly, analysis models are often dynamical models based on mathematical theories.

MDAO is a branch of applied mathematics in the domain of optimization mainly. Multidisciplinary optimization aims to formulate optimization equations and algorithms for the sake of complex system design. Further, MDAO takes leverage of high computing performance for simulation and analysis in order to become an emerging discipline. According to [43], its key strengths stem from its ability:

- to integrate high fidelity simulation tools;
- to process a huge number of variables and constraints;
- to have at its disposal a bunch of efficient optimizers;
- to take into account model uncertainties.

Fig. 4 depicts an adaptive MDAO framework around aircraft (A/C) design capability. This framework implements several open source platforms, such as openMDAO [44] and SMT [45] in the WhatsOpt design environment [46] developed by French laboratory ONERA.

The Design Structure Matrix (DSM), or dependency structure matrix, is used to depict the dependency of one component/activity/process on another one. The matrix can represent a variety of interactions and interfaces, such as time-based relations, spatial connections, signals, data, and material flows [47]. The simplest DSM is the binary DSM which shows a coupling with a 0/1 or an X equalling the 1. In this case, the X represents a direct dependency between 2 elements in the sense of: *'Element A depends on Element B'*. The main value of DSM pertains to the implementation of analysis techniques such as partitioning or clustering in order to organize a system or to show its modularity.

Similarly, the N^2 coupling matrix is a systems engineering approach used in the analysis of interfaces for logical architectures [5]. Thus, Fig. 5 represents two alternatives of logical architectures from the same functional architecture. Furthermore, let consider three features of an architecture:

- the coupling ratio R
$$R = \frac{number\_of\_interactions}{max\_of\_possible\_interactions}$$
or equal to 1 if no interaction;
- the coupling quality of a component i, called CQ_i
$$CQ_i = R_{in\_i} - R_{ext\_i}$$
where R_{in} stands for internal couplings and R_{ext} external couplings;
- as a result, the breakdown quality of an architecture, $BQ_a = avaerage(CQ_i)$ for all components i.

Thereby, the higher the BQ_a is, the more modular is the targeted architecture.

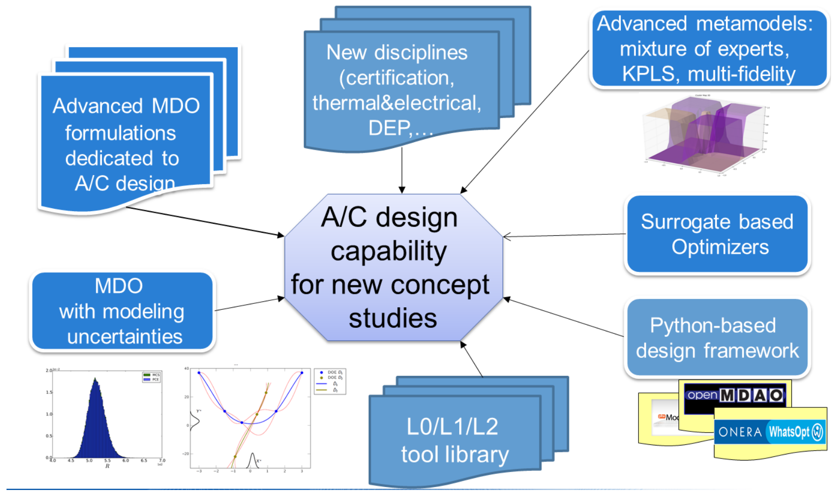
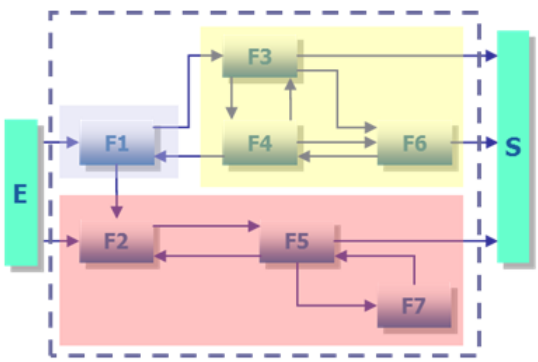
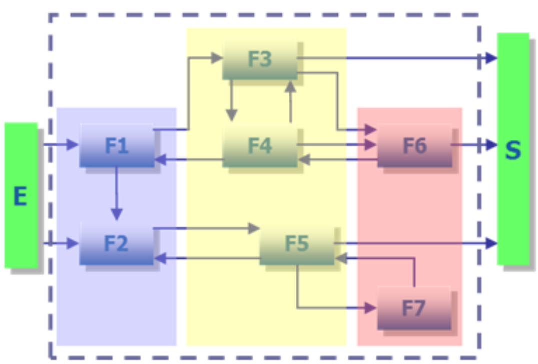


Fig. 4. MDAO framework for A/C design [43]



	F1	F2	F3	F4	F5	F6	F7	S
F1	0	1	1	0	0	0	0	0
F2	0	0	0	0	1	0	0	0
F3	0	0	0	1	0	1	0	1
F4	1	0	1	0	0	1	0	0
F5	0	1	0	0	0	0	1	1
F6	0	0	0	1	0	0	0	1
F7	0	0	0	0	1	0	0	0
E	1	1	0	0	0	0	0	0

(a) First alternative



	F1	F2	F3	F4	F5	F6	F7	S
F1	0	1	1	0	0	0	0	0
F2	0	0	0	0	1	0	0	0
F3	0	0	0	1	0	1	0	1
F4	1	0	1	0	0	1	0	0
F5	0	1	0	0	0	0	1	1
F6	0	0	0	1	0	0	0	1
F7	0	0	0	0	1	0	0	0
E	1	1	0	0	0	0	0	0

(b) Second alternative

Fig. 5. Two alternatives (a) and (b) of a functional architecture

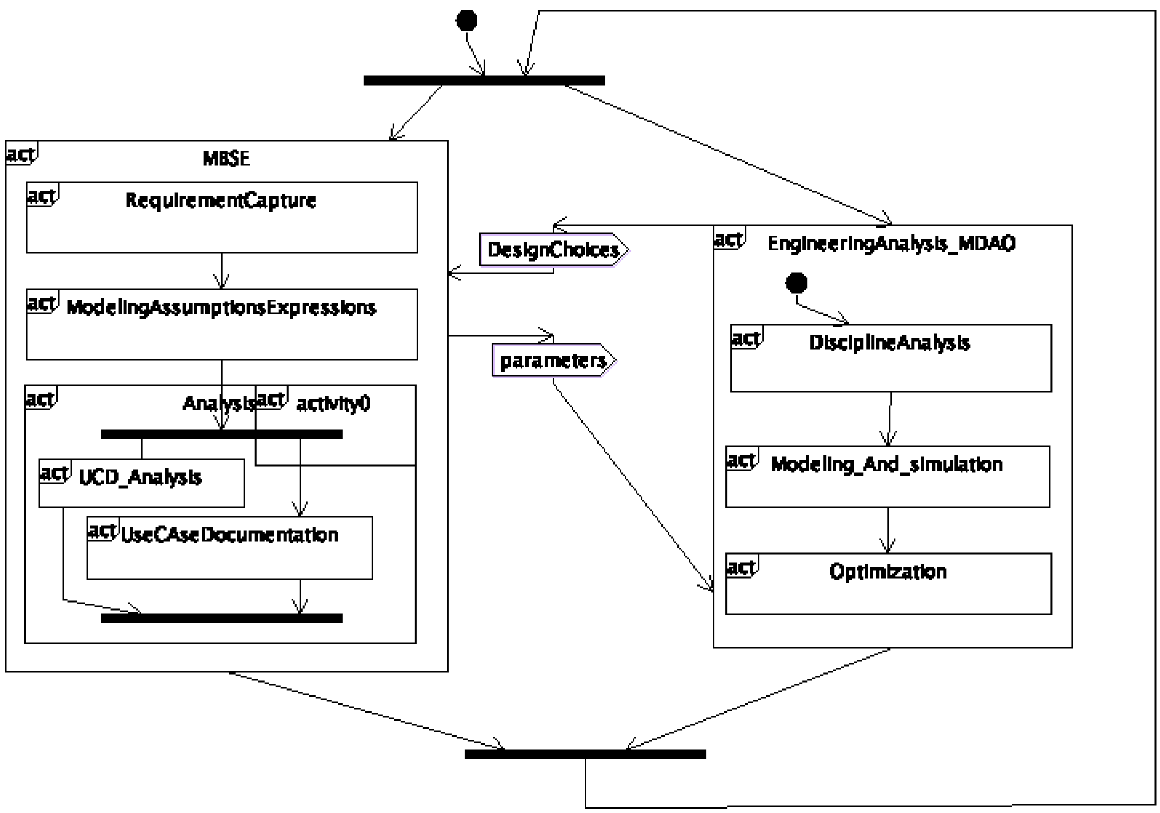


Fig. 6. MBSE and MDAO

IV. MBSE AND MDAO

Obviously, a unique universal modeling tool is unthinkable and unfruitful in a context of collaborative engineering. Models interoperability is of the utmost importance as long as performance and efficiency are concerned. Therefore, the coupling of both MBSE approach and MDAO methodology arouse a great interest in the early design phase of complex cutting-edge systems. In the literature, the integration of MBSE and MDAO is widely explored in a lot of strategic domains, such as aeronautics and defense. Tools have been developed and used in many industrial projects. Two approaches have been identified:

- A centralized approach with 'strong' coupling. This approach considers MBSE models as a focal point since it encompasses all requirements and design processes of the intended system. The other models such as MDAO or MBSA (Model-Based Safety Analysis) have point-to-point connection with MBSE models so as to exchange, *e.g.*, variables or parameters. In [48], the trade-off analysis and the design optimization are based and driven by SysML models extended with homemade plug-ins. For instance, the meta-model proposed by [48] describes a SysML framework including decision points for redundancy and system variability aspects, along with a multi-domain optimization context, using Constraint Satisfaction Problem (CSP) solver or Pareto frontier optimality.
- A centralized approach with 'weak' coupling. In this approach, the interoperability and the integration of various models enable cross-domain views from manifold disciplines [49]. The linking between the models relies on an ontology defining the data exchanges through a syntactic and semantic interoperability [50], [51]. Moreover, current commercial software such as ModelCenter facilitates the reuse of design practices suitable for multiple technical disciplines by automating interoperability between models through parametric diagrams in SysML [52], [53].

The method depicted by figure 6 associates MBSE with MDAO, which are therefore launched in parallel. Relying on the method [27] associated with SysML and free software TTool [36], figure 6 starts MBSE with requirement capture and use case driven analysis. MDAO, which is started in parallel, is made up of three steps: Discipline Analysis, Modeling and Simulation, and Optimization. MDAO models will be physical models (differential equations) and behavioral models (dynamics associated with scenarios).

The expected benefits of adding MBSA to MBSE include the capacity to identify requirements that definitely cannot be met, as well as an assistance in making architecture choices. Let us remark that combining multi-domain approaches rises coherence problems [54]

V. Conclusions and Future work

The field of SE as a solution for the increasing complexity and the interdisciplinary challenges provides methods and tools based on models such as MBSE and MDAO. Indeed, the rise of Information Technology (IT) systems facilitates the use of models throughout all the phases of a system development. Also, the early design phase requires to pay a particular attention due to design error costs.

So far, MBSE and MDAO have mostly been addressed separately since the two communities have few links in terms of goals and methods. However, a promising trend is to encourage collaborative engineering and research. The authors of this abstract contribute to bridge the gap between the two communities in the framework of French Concorde project.

Among the weaknesses of today's manual approach one may identify:

1) A lack of confidence in the fact that the MDAO model is a correct representation of the MBSE model part it is supposed to cover;
2) Formal languages that support MDAO are often very verbose by nature and the models take some non negligible amount of time to be written and validated while part of the information is obviously already available in the MBSE description;
3) Integration of the results of the MDAO process in the MBSE model is rarely achieved. Reasons are twofold: lack of expressiveness of MBSE languages, and high time cost of the manual rewriting of the results with little gain expected from having this information available in the MBSE model.

One major expected achievement of the project is to design and implement a methodology to populate parts of the MDAO model directly from the MBSE one. Aligned with Agile4.0 European project, our challenges are to facilitate the transition from MBSE viewpoints to MDAO design system [55]. However, our focus is less on formalizing and modeling a MDAO design system than on formalizing a global framework for the design of complex critical systems, integrating other approaches such as safety analysis method (Model-Based Safety Assessment approach). Our project will have an impact on the three previous points: improving the confidence in the fact that the two models are addressing the same system, alleviating the effort to produce the MDAO model and hence helping expert to focus on parts of the MDAO model where human expertise is needed, and making explicit the links between the concepts of the MBSE model and the MDAO model opening the way to capture MDAO results in the MBSE model in a more systematic way. Generic modeling patterns - for which the transcription between the MBSE models and MDAO models is formally described - will be the privileged approach followed.

References

[1] Lawrence D. Miles. *Techniques of Value Analysis and Engineering*. Miles Value Foundation, 3rd edn. edition, 1989.

[2] US Air Force. *System Safety Engineering for the Development of Air Force Ballistic Missiles*, 1962.

[3] MIL-STD-882A. *System Safety Program Requirements*, 1969.

[4] Ana Luísa Ramos, José Vasconcelos Ferreira, and Jaume Barceló. Model-based systems engineering: An emerging approach for modern systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):101–111, 2011.

[5] Cecilia Haskins, Kevin Forsberg, and Michael Krueger. *Systems Engineering Handbook - A Guide for System Life Cycle Processes and Activities*. Wiley, 2007.

[6] Peter Senge. The fifth discipline: The art and practice of the learning organization. *Doubleday: New York, NY, USA*, 1991.

[7] Sigal Koral Kordova, Moti Frank, and Anat Nissel Miller. Systems thinking educationseeing the forest through the trees. *Systems*, 6(3):29, 2018.

[8] Jonathan H Klein. The abstraction of reality for games and simulations. *Journal of the Operational Research Society*, 36(8):671–678, 1985.

[9] Gordon J Laing. The classification of models: A proposal. *Interdisciplinary Science Reviews*, 6(4):355–363, 1981.

[10] Jacques Bertin. Semiology of graphics: diagrams, networks, maps. Technical report, ESRI, 1983.

[11] Jonathan C Roberts. Display models - ways to classify visual representations. *International Journal of Computer Integrated Design and Construction*, 2(4):241–250, 2000.

[12] Daniel Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on software engineering*, 35(6):756–779, 2009.

[13] ITU-T-Z.100. Specification and description language, http://handle.itu.int/11.1002/1000/14048. *Retrieved October 31, 2020*, 2019.

[14] IEEE. Vhdl standard. https://fr.mathworks.com/. *Retrieved October 31, 2020*, 2008.

[15] Matworks. Mathworks-languages, https://fr.mathworks.com/. *Retrieved October 31, 2020*, 2020.

[16] Modelica Association. https://www.modelica.org/. *Retrieved October 31, 2020*, 2008.

[17] Thierry Le Sergent, Franois-Xavier Dormoy, and Alain Le Guennec. Benefits of model based system engineering for avionics systems. *ERTS 2016, Toulouse, France*, 2016.

[18] H. Mkaouar, B. Zalila, Jrme Hugues, and M Jmaiel. A formal approach to aadl model-based software engineering. *International Journal on Software Tools for Technology Transfer*, 22(2), 2020.

[19] L. Li, N. L. Soskin, A. Jbara, M. Karpel, and D. Dori. Model-based systems engineering for aircraft design with dynamic landing constraints using object-process methodology. *IEEE Access*, 7, 2019.

[20] TINA. Time petri net analyzer, http://projects.laas.fr/tina//. *Retrieved October 31, 2020*, 2020.

[21] UPPAAL. http://www.uppaal.org/. *Retrieved October 31, 2020*, 2020.

[22] CADP. Constructions and analysis of distributed processes, https://cadp.inria.fr/. *Retrieved October 31, 2020*, 2020.

[23] Jean-Raymond Abrial. *Modeling in Event-B: system and software engineering*. Cambridge University Press, 2010.

[24] Jean-Charles Chaudemar, Vitaly Savicks, Michael Butler, and John Colley. Co-simulation of event-b and ptolemy ii models via fmi. In *ERTS 2014, Embedded real time software and systems*, 2014.

[25] Michael Leuschel and Michael Butler. ProB: an automated analysis toolset for the B method. *International Journal on Software Tools for Technology Transfer*, 10(2):185–203, 2008.

[26] Kevin A. Reilley, Stephen J. Edwards, Russel S. Peak, and Dimitri. N. Mavris. Methodologies for modeling and simulation in model-based systems engineering tools. *IAAA Space Forum, Long Beach, CA, USA*, 19:111169, 2016.

[27] Ludovic Apvrille, Pierre de Saqui-Sannes, and Rob A. Vingerhoeds. An educational case study of using sysml and ttool for unmanned aerial vehicles design. *IEEE Journal on Miniaturization for Air and Space Systems*, 1(2):117..129, 2020.

[28] Jean-Louis Colaço, Bruno Pagano, Cédric Pasteur, and Marc Pouzet. Scade 6: From a kahn semantics to a kahn implementation for multicore. In Hiren Patel, Tom J. Kazmierski, and Sebastian Steinhorst, editors, *2018 Forum on Specification & Design Languages, FDL 2018, Garching, Germany, September 10-12, 2018*, pages 5–16. IEEE, 2018.

[29] Rachida Dssouli, Ahmed Khoumsi, Mounia Elqortobi, and Jamal Bentahar. Testing the control-flow, data-flow, and time aspects of communication systems: A survey. *Advances in Computers*, 107, 2017.

[30] A. Buchiarone, J. Cabot, Paige R.F., and Pierantonio A. Grand challenges in model-driven engineering: an analysis of the state of the research. *Software and Systems Modeling*, 9:5–13, 2020.

[31] ANSI. Eia632 - process for engineering a system. *GEIA, Arlington, VA, USA*, 2003.

[32] IEEE 1220-2005. *IEEE Standard for Application and Management of the Systems Engineering Process*, 2005.

[33] ISO. *IEC 15288 - https://www.iso.org/fr/standard/63711.html*, 2003.

[34] SAE. *ARP4754A: Guidelines for Development of Civil Aircraft and Systems*, 2010.

[35] S. Zhu, J. Tang, Jean-Marie Gauthier, and Raphal Faudou. A formal approach using sysml for capturing functional requirements in avionics domain. *Chinese Journal of Aeronautics*, 32(12), 2019.

[36] TTool. https://ttool.telecom-paris.fr/. *Retrieved October 31, 2020*, 2020.

[37] Fellipe Rey de Suza, Juliana de Melo Bezerra, Celso Hirata, Pierre de Saqui-Sannes, and Ludovic Apvrille. Combining stpa with sysml modeling. In *The 14th annual IEEE International Systems Conference (SysCon 2020)*, Montréal, Qc, Canada, 2020.

[38] Steve Jeffrey Tueno Fotso, Régine Laleau, Hector Ruiz Barradas, Marc Frappier, and Amel Mammar. A formal requirements modeling approach: application to rail communication. In *ICSOFT 2019: 14th International Conference on Software Technologies*, Proceedings of the 14th International Conference on Software Technologies, pages 170–177, Prague, Czech Republic, July 2019. Scitepress.

[39] Juan Navas, Philippe Tannery, Stphane Bonnet, and Jean-Luc Voirin. Bridging the gap between model-based systems engineering methodologies and their effective practice a case study on nuclear power plants systems engineering. *INCOSE*, 21(1), 2018.

[40] EclipseCon-PRFC. Hands-on systems modeling with arcadia / capella. http://www.clarity-se.org/wp-content/uploads/2015/05/PRFC_Capella-WS_2016.pdf, 2016.

[41] Sarah A Sheard and Ali Mostashari. Principles of complex systems for systems engineering. *Systems Engineering*, 12(4):295–311, 2009.

[42] Tao Yue, Lionel C Briand, and Yvan Labiche. A systematic review of transformation approaches between user requirements and analysis models. *Requirements engineering*, 16(2):75–99, 2011.

[43] Nathalie Bartoli. *Optimisation adaptative basée sur les métamodèles*. PhD thesis, Université Toulouse III, 2019.

[44] Justin Gray, Kenneth T Moore, Tristan A Hearn, and Bret A Naylor. Standard platform for benchmarking multidisciplinary design analysis and optimization architectures. *AIAA journal*, 51(10):2380–2394, 2013.

[45] University of Michigan, ONERA, ISAE-SUPAERO, and NASA. Surrogate model toolbox. https://github.com/SMTorg/SMT/.

[46] Rémi Lafage, Sebastien Defoort, and Thierry Lefebvre. Whatsopt: a web application for multidisciplinary design analysis and optimization. In *AIAA Aviation 2019 Forum*, page 2990, 2019.

[47] J Bartolomei, M Cokus, J Dahlgren, R De Neufville, D Maldonado, and J Wilds. Analysis and applications of design structure matrix, domain mapping matrix, and engineering system matrix frameworks. *Massachusetts Institue of Technology*, 2007.

[48] Patrick Leserf, Pierre de Saqui-Sannes, and Jérôme Hugues. Trade-off analysis for sysml models using decision points and csps. *Software and Systems Modeling*, 18(6):3265–3281, 2019.

[49] Mark Blackburn, Dinesh Verma, Robin Dillon-Merrill, Roger Blake, Mary Bone, Brian Chell, Rick Dove, John Dzielski, Paul Grogan, Steven Hoffenson, et al. Transforming systems engineering through model-centric engineering. Technical report, Stevens Institute of Technology Hoboken United States, 2018.

[50] Andreas Tolk. What comes after the semantic web - pads implications for the dynamic web. In *20th Workshop on Principles of Advanced and Distributed Simulation (PADS'06)*, pages 55–55. IEEE, 2006.

[51] Amogh Kulkarni, Daniel Balasubramanian, Gabor Karsai, and Peter Denno. An analytical framework for smart manufacturing. In *MATEC Web of Conferences*, volume 249, page 03010. EDP Sciences, 2018.

[52] Wayne Hurwitz, Shane Donovan, Jose Camberos, and Brian German. A systems engineering approach to the application of multidisciplinary design, analysis and optimization (mdao) for efficient supersonic air-vehicle exploration (esave). In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, page 5491, 2012.

[53] Nicolas Albarelloet and Hiongman Kim. Applying a mbse approach to multidisciplinary design processes integration (in french). *Génie Logiciel*, (108), 2014.

[54] S. Missaoui, F. Mhenni, J.Y. Choley, and N. Nguyen. Verification and validation of the consistency between multi-domain system models. *2018 Annual IEEE International Systems Conference (SysCon), Vancouver, BC, Canada*, pages 1–7, 2018.

[55] P. D. Ciampa, B. Nagel, and G. L. Rocca. A mbse approach to mdao systems for the development of complex products. In *AIAA AVIATION 2020 FORUM*, 2020.