

Article

Toward Demystifying the Missing Links in Model-Based Systems Engineering (MBSE)

Azad Khandoker ^{1,*}, Sabine Sint ², Guido Gessl ¹ and Klaus Zeman ¹

¹ Institute of Mechatronic Design and Production, Johannes Kepler University Linz, Altenberger Straße 69, 4040 Linz, Austria; guido.gessl.1@gmail.com (G.G.); klaus.zeman@jku.at (K.Z.)

² Research Unit of Building Physics, TU Wien, Karlsplatz 13, 1040 Vienna, Austria; sabine.sint@tuwien.ac.at

* Correspondence: azadaustria@gmail.com

Abstract

Model-Based Systems Engineering (MBSE) originated in aerospace engineering and has emerged as a promising approach in other fields for designing, analyzing, and managing complex interdisciplinary systems throughout their entire life cycle. While MBSE is applicable to various engineering domains, its applications remain closely tied to software engineering. As software becomes a critical component of physical systems, such as vehicles, appliances, and production plants, bridging the gap between software engineering and other disciplines, such as mechanical, electrical, and civil engineering, becomes essential. Despite its potential, MBSE is still in its early stages when it comes to integrating executable models of physical systems into engineering environments. The purpose of this research is to assess the present capabilities of MBSE by identifying existing missing links, thereby enabling prospective users to make well-informed decisions about its integration into organizational processes. In this analysis, it is important to have a comprehensive view of the complexity of MBSE across different disciplines to obtain an overall picture. In addition to identifying open challenges, we present three critical gaps in the MBSE practice through a comprehensive demonstration case: limited tool interoperability and model integration, modeling language limitations, and dependence on a specialized workforce. Current studies largely view MBSE as the most applicable and effective for the design phase of the system life cycle. Yet, to capture MBSE in its entirety, its principles must be applied throughout the whole system life cycle.

Keywords: model-based systems engineering; MBSE genesis; missing links; system development; MBSE Dissemination Domains; MBSE Challenges Domains



Academic Editor: Jinzhi Lu

Received: 18 December 2025

Revised: 23 January 2026

Accepted: 27 January 2026

Published: 1 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Nowadays, systems, their interconnection, and integration are becoming increasingly complex, especially when physical and digital parts are interconnected in systems, such as Cyber-Physical Systems (CPS), and can no longer be clearly separated from one another [1–3]. The concept of Model-Based Systems Engineering (MBSE) is a promising approach to improve efficiency and effectiveness in system development [4]. It promotes rational understanding, better complexity management, rapid information exchange, transparency, and communication between disciplines and stakeholders, which leads to a solid knowledge base even in the early stages of development. This knowledge base brings long-term benefits throughout the System Life Cycle (SLC). According to an up-to-date understanding of systems engineering, MBSE should be applied throughout the entire SLC.

However, several survey studies show that MBSE application remains largely confined to the early-design phase—particularly requirements and architecture modeling—while later lifecycle phases (e.g., manufacturing, operation, maintenance, and decommissioning) are underrepresented or modeled in isolation [5]. This limitation stems from several critical factors. Although models are robust during the design phase, they often fail to propagate into manufacturing and operations due to a broken “digital thread”. For example, in the aerospace industry, a high-fidelity SysML model used by design engineers is rarely accessible or useful to maintenance technicians on the hangar floor, who still rely on static PDF manuals [6–8]. In addition, there is currently a lack of standardized data exchange between MBSE tools (such as Enterprise Architect, MagicDraw or Cameo) and downstream Enterprise Resource Planning (ERP) or Product Lifecycle Management (PLM) systems. This creates “data silos”, where the early models of the PLC phase become an isolated artifact rather than a living blueprint [8–10]. Keeping models synchronized with the physical system during the “in-service” phase requires significant manual effort. Without automated updates from sensors or operational feedback, the models quickly become obsolete (“as-built” vs. “as-designed” gap), leading stakeholders to abandon them after production starts [7,11]. In aerospace, SysML-based MBSE is routinely used for requirements and system architecture, yet digital continuity to manufacturing (e.g., CAM, PLM integration) is rare—leading to manual re-translation of geometric or tolerance data [12–14]. In the automotive industry, for example, MBSE supports E/E architecture design and functional safety (e.g., ISO 26262 compliance), but rarely extends to service manuals or predictive maintenance. This leads to “design–service gaps” where operational failures cannot be traced back to model assumptions [13].

Although MBSE represents a methodological framework toward model-centric approaches for integrating various engineering disciplines, such as mechanical, electrical, and software engineering, its inadequate application has led to disappointing project outcomes [15]. Many industries have shown interest in implementing MBSE, as highlighted by several surveys [16–23]. However, attempts to transform existing technical approaches to MBSE often occurred without a proper introduction or explanation of the methodology, resulting in insufficient applications. This underscores the necessity for tailoring MBSE practices to specific engineering contexts, as it can be demonstrated by various industry examples. In aerospace, a model-first approach utilizing SysML ensures compliance with strict certification standards like DO-178C/DO-254, facilitating thorough verification along the whole PLC. In contrast, Small to Medium-sized Enterprises (SMEs) in smart agriculture often adopt a code-first strategy, allowing for rapid prototyping in Simulink before integrating key interfaces into SysML; this flexibility speeds up time-to-market. Civil infrastructure projects typically focus on asset lifecycle modeling, where detailed real-time behavior may complicate processes unnecessarily. Moreover, the development of an aircraft navigation system exemplifies how varying requirements necessitate different MBSE models, while advanced driver-assistance systems (ADAS) in automotive engineering must adapt to urban and highway conditions. Similarly, in healthcare, electronic health record (EHR) systems reveal differing requirements between rural and urban clinics.

These cases illustrate that MBSE is not a monolithic methodology, but a flexible framework whose scope, depth, toolchain, and lifecycle coverage must be calibrated to project context-size, domain, regulation, risk, and innovation pace. Hence, we posit “To be applicable, MBSE must be understandable and adaptable to the specific engineering situation”.

This article aims to identify the missing puzzle pieces necessary for establishing a rational understanding of MBSE. To be precise, we investigate potential missing links related to the experience and implementation of MBSE. In doing so, we aim to provide an overview of the requirements related to successful MBSE applications.

The remainder of the paper is as follows. Section 2 provides related work in the area of MBSE challenges. In Section 3, we give a short overview of our applied methodology. Section 4 shows the historical evolution of MBSE that leads to the current state of the art of MBSE. Section 5 presents the missing links toward MBSE from the perspectives of the Challenges and Dissemination Domains. Section 6 shows the challenges and missing links in its application. Finally, in Section 7, we conclude our work and give an outlook to future work.

2. Related Work

This section provides an overview of related work to highlight both the similarities and differences in comparison to our approach. First, we investigate the current state of MBSE implementation across various industries. Second, we examine challenges and limitations identified in multiple scientific studies.

Concerning the industry, we analyzed which sectors use MBSE, which technical disciplines are involved in the application, in which SLC phases MBSE is used, and in which phases the companies achieve significant benefits. Several sources confirm the increasing adoption of MBSE in various industries, especially in aerospace, defense, automotive, information technology, medical, and consulting [12,23–29]. The most crucial engineering disciplines include mechatronics and systems engineering, software engineering, mechanical engineering, and electrical and electronics engineering [12,24,25]. Regarding SLC phases, it can be seen that MBSE brings the most significant benefit in requirements engineering, followed by system architecture development and V&V. MBSE is also used in requirements elicitation and detailed design. The analysis shows that MBSE is used in different industries, technical disciplines, and SLC phases. However, it also highlights a critical gap, which is the need for improved integration along the different system lifecycle phases in MBSE practice.

In the following, we review some scientific studies regarding challenges and limitations related to MBSE.

The publication of Chami and Bruel [12] presents the findings of a survey investigating challenges associated with adopting MBSE in real-world applications. The survey solicited participants' perspectives on these challenges, including their occurrence within the phases of MBSE adoption and the interdependencies among them. This approach aims to improve the understanding and measurement of the challenges associated with MBSE adoption. In comparison, we examine MBSE from three perspectives (cf. Section 5), highlighting its missing links. Furthermore, we illustrate these challenges using the design and development process of an e-bike.

Cordero et al. [30] examine initial efforts to integrate an MBSE approach into the Nanospace open-source framework. They highlight the challenges of knowledge reuse, design convergence, and architecture candidate evaluation. Our paper presents the missing links holistically in the context of MBSE. It is about the overall view of what is missing for a successful implementation of MBSE.

Kim et al. [31] address the challenges of applying MBSE through a human-centered design perspective. Using the Human-Centered Design (HCD) framework, the authors identify key challenges, such as cultural changes, and discuss their impact on the development of tools and infrastructure. They emphasize the need for a deep understanding of existing systems engineering practices, investment in domain-specific ontologies, and the definition of MBSE methods, processes, tools, and infrastructure. Choosing tools without a comprehensive understanding potentially leads to acceptance issues [31]. In contrast to their approach, our paper highlights missing links from historical perspectives to domain-specific viewpoints.

Albers and Zingel [32] discuss in their study the challenges of MBSE-related languages, particularly SysML, and the difficulty in understanding SysML syntax and semantics in MBSE-related applications, especially in software and mechanical engineering. They note that understanding basic terms, such as “function” and “behavior”, is still very inconsistent among systems engineers in academia and industry, but tends to converge. As a result, there is an opportunity to harmonize and formalize the terms, especially within modeling languages, such as SysML. Compared to this study, we propose a holistic approach where different challenges throughout the PLC are identified.

Pinquié et al. [33] discuss the current practice and challenges with a survey, observing that models are not used as they should, leading to model-based design review becoming a real challenge during the product development initial phase. Other identified challenges include the modeling approach, model review, and interoperability problems. Notably, their paper does not discuss common challenges, such as model management and tool integration. In contrast, our paper seeks to fill these gaps by addressing the entire product lifecycle and identifying the challenges impeding the full potential of MBSE in real-world problem-solving scenarios. We demonstrate these challenges and their implications, providing a comprehensive understanding of the obstacles faced when applying MBSE.

D’Ambrosio and Soremekun [13] highlight several critical points regarding the challenges and opportunities in SE within the automotive industry, emphasizing the need to address these in MBSE-based approaches. One of the key challenges they discuss is the increasing complexity of high-performance computing platforms, where managing numerous Electronic Control Units (ECUs) and extensive software code becomes difficult. Another important challenge is the application of MBSE, which, while showing promise in its early stages, requires further demonstration of its benefits to validate its effectiveness. Additionally, the authors address the need for a comprehensive enterprise-wide acceptance strategy and the development of effective methodologies. D’Ambrosio and Soremekun [13] also outline several specific applications of MBSE, particularly highlighting feature modeling, system safety case development, and testing procedures. Compared to this approach, our paper takes a unique approach to identifying MBSE-related challenges, where the importance of understanding the field of MBSE by tracing its historical evolution is emphasized, and discipline-specific perspectives are addressed.

It is crucial to note that none of the compared works delve into the genesis or historical schools of thought related to MBSE, which is fundamental for grasping the essence of MBSE. Understanding the historical evolution of MBSE provides valuable insights into its foundations and evolution, which is essential for applying MBSE effectively to real-life problem-solving scenarios. Furthermore, our approach of investigating the challenges from the Dissemination Domain’s specific views allows us to identify and address challenges specific to each field. This approach enables us to address all key stakeholders who must work together to overcome the challenges of MBSE. Our research aims to holistically identify and demonstrate the missing links to contribute to a more comprehensive understanding of MBSE, its practical implications, and eventually, to advancing and disseminating MBSE across disciplines and application domains.

3. Methodology

As an emerging technology, MBSE is entangled with diverse challenges, presenting a contrast where specific challenges can be mastered while others remain subtle. Consequently, not all promising options may be thoroughly explored when using MBSE, leading to unsuccessful attempts, especially by inexperienced users. Such failures permeate the engineering domains and might brand MBSE as a technology prone to shortcomings [34,35]. In light of this evolutionary perspective, our objective is to formulate a methodology ensur-

ing comprehensive exploration of critical avenues, providing individuals with a thorough understanding of MBSE capabilities. Subsequently, individuals can make informed decisions about whether integrating MBSE would contribute value to their engineering projects, thus enabling conscious and deliberate utilization. In formulating our methodology, we have drawn upon three theoretical frameworks: the “MBSE Maturity Model” [36], the “MBSE knowledge Graph” [37], and the “MBSE logical framework” [38]. These conceptual frameworks guide our exploration of three facets, namely MBSE Genesis, MBSE Domains (more precisely: MBSE Challenges Domains and MBSE Dissemination Domains), and MBSE Applications, which are identified as pivotal components encompassing missing links within the MBSE landscape.

Building on those historical and theoretical perspectives, we developed a methodology, presented graphically in Figure 1. To illustrate this methodology, we employed the metaphor of a house. Viewing the house from the outside provides an external perspective akin to observing the outer structure of a building. However, a comprehensive understanding requires an exploration of the interior, just as one cannot determine the suitability of a house for its inhabitants without looking at its internals. From an architectural point of view, external and internal observations are essential for the architecture and construction of the house. In alignment with this concept, the challenges faced by MBSE are divided into two categories: external challenges and embodied challenges, where embodied challenges cover missing links directly in the process and implementation of MBSE, and external challenges represent a broader external perspective. Developing and applying the MBSE landscape involves systematically reducing missing links—traversing from MBSE Genesis, through MBSE Domains, to MBSE Applications. The MBSE Genesis focuses on understanding its historical development, while MBSE Domains’ Missing Links highlight related domains and their associated challenges. The MBSE Applications’ Missing Links reveal challenges related to implementation. It is worth noting that these missing links not only fall under either external or embodied challenges but span both areas. This is also due to the holistic understanding required for the MBSE Applications.

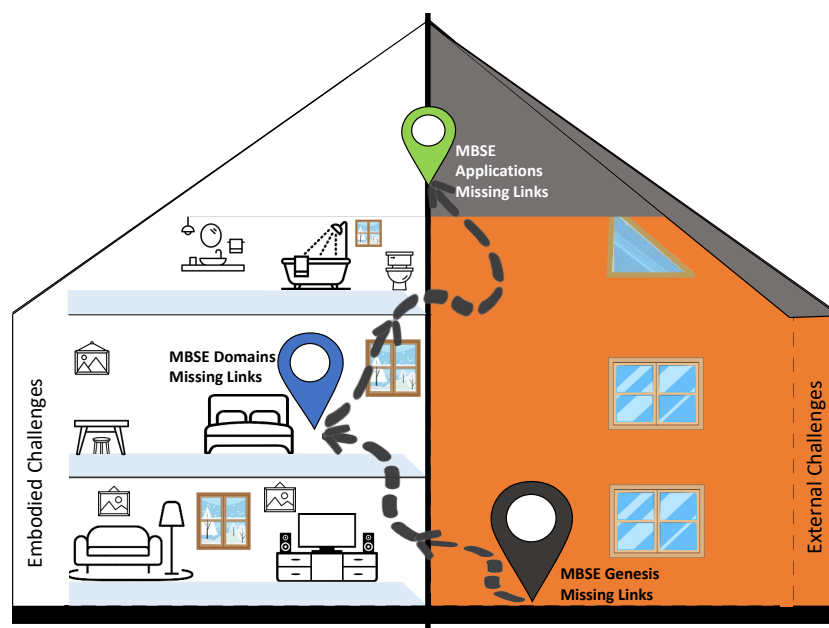


Figure 1. Methodology regarding navigation of the MBSE missing links.

The following sections provide a detailed description of each part of the methodology.

4. MBSE Genesis

This section synthesizes the conceptual and methodological evolution underlying MBSE by identifying the principal schools of thought that contribute to its formulation. Rather than providing a comprehensive historical account, the section clarifies how different system-related paradigms progressively have evolved in formalization and practical applicability, ultimately enabling MBSE. This perspective is essential for fostering a coherent and rational understanding of MBSE and for recognizing the fundamental stages that have led to its current state. Figure 2 presents this synthesis as a “road metaphor” structured by the degree of formalization rather than by exhaustive historical chronology, supporting the interpretation of MBSE application processes and bridging domain-specific knowledge gaps.

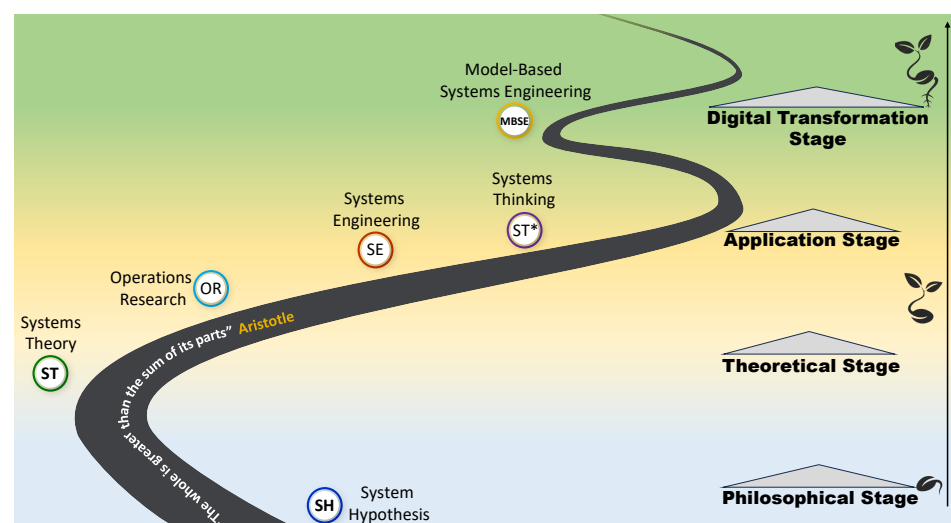


Figure 2. Evolution of foundational schools of thought towards MBSE.

The conceptual foundations underlying MBSE can be traced to the Aristotelian concept of wholeness, which stated that all things have parts and that the whole is distinct from the sum of its parts [39–41]. We adopt this notion as the philosophical foundation and consider it as the “seed” from which the field of MBSE has emerged. Aristotle also emphasized the connections and unity between elements within a system. This philosophical foundation has influenced the development of artificial systems for centuries. Accordingly, this foundational school of thought is designated as the System Hypothesis (SH), representing the philosophical stage of the system concept that underpins later formal developments and serves as a driving force that ultimately contributed to the emergence of MBSE.

As the field matured, it entered the theoretical stage, which represents the gradual “sprouting” of systems concepts through increasing abstraction and formalization. This evolution leads to another fundamental component within the MBSE schools of thought, namely Systems Theory (ST), which accentuates the holistic nature of complex systems. The inception of ST can be traced back to the theoretical biology of von Bertalanffy [42] and was later formalized as General Systems Theory (GST). GST examines the concept of wholeness in a generalized and abstract manner. Moreover, Shannon’s information theory [43] and Wiener’s cybernetics [44] contribute significantly to this domain by addressing uncertainty, communication, and control in both biological organisms and machines—marking a transition from philosophical insight to formal theoretical abstraction essential for artificial systems [45]. Operations Research (OR), following ST, as an analytical precursor, supports decision-making, optimization, and resource allocation, aligning with the foundational principles of MBSE.

The next phase of the progression is the application stage, representing the “growing plant” where system concepts are operationalized into engineering practices. To address the escalating complexity of artificial systems, Systems Engineering (SE) evolved into an independent discipline, particularly suited to manage uncertainty in large-scale technical systems [46]. The initial methodological application of SE took place at Bell Labs in the 1940s [47,48]. Systems Thinking (ST*) represents the cognitive and methodological mindset required to apply system concepts in practice and serves as a conceptual bridge between theoretical understanding and practical engineering implementation [49].

Building on the impacts and implementations of the previous stages, we have now reached the digital transformation stage, where digital recording and consistency are used to improve data analysis and to automate processes. In this context, MBSE evolved. The term Model-Based Systems Engineering (MBSE) was coined by Professor Wayne Wymore in 1993, and it delves into the theoretical underpinnings of SE through the use of explicit models and formal representations [50]. This approach received further impetus through initiatives led by the International Council on Systems Engineering (INCOSE) [50,51]. Currently, INCOSE defines MBSE as “the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” [52] (p. 15).

The staged representation in Figure 2 does not imply that MBSE is solely derived from ST, ST*, and SE, but rather delineates their interconnected contributions based on increasing levels of formalization and practical applicability. While ST provides the abstract foundation for understanding holistic system behavior, SE operationalizes these ideas into structured engineering processes—particularly in managing complexity and uncertainty in large-scale technical systems. ST* serves as the cognitive and methodological bridge between the two, enabling practitioners to interpret system interdependencies and to apply holistic reasoning within engineering contexts. Although ST primarily constitutes a mindset, its integration into SE is essential, as systematic implementation inherently requires structured thinking. Similarly, OR, with its focus on decision-making, optimization, and resource allocation, aligns closely with the analytical underpinnings of SE and MBSE. As noted by Wymore [50], OR’s problem-solving orientation complements SE’s lifecycle-oriented approach, making it an integral facet of the broader systems engineering framework. Based on the preceding discussions, Figure 3 is proposed as a consolidated depiction of essential elements within MBSE necessary for comprehending and applying it in practical contexts.

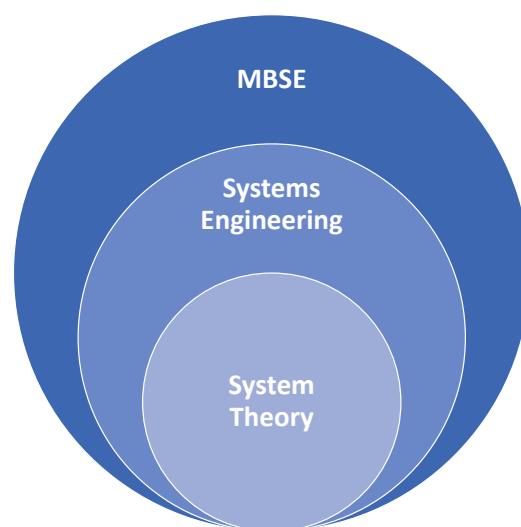


Figure 3. The concentrated presentation of MBSE in terms of application understanding.

5. MBSE Domains Specific Missing Links

In this section, we outline the main challenges and missing links encountered by moving towards MBSE. These challenges are based on conference discussions with academics [53], interviews with tool developers [38], and consultations with practitioners from various industries [12,31–33]. While we aim to offer a comprehensive overview, we recognize that our coverage may not be exhaustive. The challenges presented have been identified as prevalent across multiple contexts. To categorize these challenges, we distinguish between three different MBSE Domains: applications, tool developers, and academics [38]. Each MBSE Challenges Domain refers to a specific sector of activities involving stakeholders that contribute to and interact with MBSE methodologies and tools. Each of the three key MBSE Domains (Figure 4) plays a distinct but interconnected role in advancing and applying MBSE. Understanding these Domains and their interactions is essential for realizing the full MBSE potential.

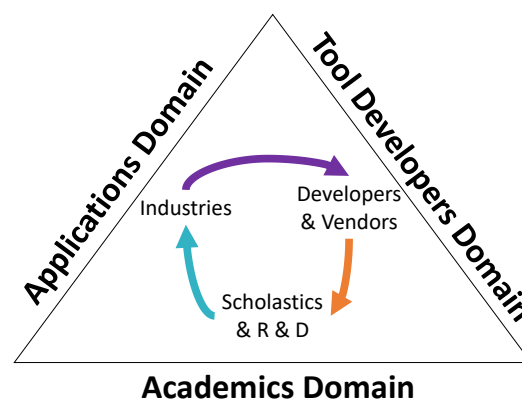


Figure 4. MBSE Challenges Domains based on [38].

5.1. Challenges of the Applications Domain

The Applications Domain, which includes industries that are applying and implementing MBSE to support the Product Life Cycle (PLC) [38], has to overcome the following shortcomings.

5.1.1. Unified Implementation Process

The Applications Domain encompasses various industries, departments, projects, and tasks. However, the main challenge lies in creating a transparent implementation process addressing fundamental questions like application location, requirements, methods, and benefits. This process should be comprehensive and step-by-step, supplementing all types of system development. Different implementation processes for various industries or products can lead to confusion and conflicts during the system integration phase of interdisciplinary products. For instance, language and tool selection may differ if one subsystem is developed using a PLC approach and others using a System Development Life Cycle (SDLC) approach. Therefore, a unified, however, maximum flexible and adaptable implementation process is the ultimate goal. Providing such a clear workflow is a significant challenge.

Probable Solutions: An MBSE-centric implementation process should define common phases, artifacts, and decision gates, while enabling configurable tailoring across industries and project types. This process framework requires clear, step-by-step guidance for integrating diverse lifecycle models, supported by reference workflows and best-practice patterns to ensure consistency and to reduce ambiguity. Agile approaches can enhance flexibility and responsiveness through iterative cycles and embedded feedback loops, fostering

continuous improvement [54]. Finally, targeted process automation streamlines repetitive tasks, freeing teams to focus on strategic and creative problem-solving.

5.1.2. Available Workforce

Implementing new technologies in industries often faces challenges due to a scarcity of trained workforces. Building a trained workforce is time-consuming and involves risk factors like cost-benefit issues, trainer hiring, team motivation, and convincing authorities. Part of the responsibility for building a workforce for new technology applications can be taken on by academia. The challenge, however, is to create a culture where all essential factors and risks are considered.

Probable Solutions: It is important to establish collaborative programs between industry and academia to systematically train and certify a workforce in new technologies, ensuring alignment with real-world industrial needs. Additionally, fostering a culture of continuous learning, risk awareness, and institutional support can help organizations effectively overcome challenges related to costs, motivation, and acceptance. To accelerate workforce development, targeted MBSE certification courses covering essential topics can be offered to quickly train and prepare new professionals.

5.1.3. Unified Languages and Methods

Languages are crucial for the exchange of information between humans and machines because they provide a standardized vocabulary and rules (grammar) to communicate without confusion. The languages capture the concepts and relationships between them, regardless of the notation that users want to use to manipulate models of the languages. For instance, UML (<https://www.uml.org/>) and SysML (<https://sysml.org>) are widely used languages. In the context of MBSE, however, it must be noted that these concepts cannot model the entire PLC. For example, SysML does not cover all systems engineering processes, such as design synthesis and analysis, or verification and validation of all relevant properties of a proposed solution (e.g., basic design or detailed design). It also cannot express the aesthetic appearance of a system, which is a non-functional requirement for many products. In addition, Micouin [55] argues that SysML requirements diagrams do not fully fulfill the promise of MBSE because they are a new representation of text-based traditional requirements. Other languages, such as Business Process Model and Notation (BPMN (<https://www.omg.org/bpmn/>)) or Lifecycle Modeling Language (LML (<https://www.lifecyclemodeling.org/specification>)), can support further MBSE use cases. The challenge is to ensure consistency, standardization, and exchange between these languages and to provide methods for manipulation. However, language models could reduce new users' learning curve, facilitate tools' use, and support interoperability as the concepts follow clear definitions.

Probable Solutions: MBSE-focused languages, such as SysML v2 (<https://www.omg.org/sysml.org/SysML-2.htm>), aim to provide comprehensive coverage of the PLC while supporting clear definitions for concepts, relationships, semantics and interoperability. The textual and graphical syntax not only increases user-friendliness but also promotes automation possibilities. The SysML v2 specifications were officially adopted by the OMG in July 2025, and a standard API and associated set of services for navigating, querying, and updating the model are intended to enable interoperability with other tools and software applications throughout the system development lifecycle [56–58]. As a result, SysML v2 may hold the potential to address this missing link. However, there is currently no empirical evidence that proves this claim, and further research is needed to validate the actual benefits and effectiveness of its implementation in practice. Moreover, a modeling language alone cannot resolve all gaps in MBSE adoption, rather the method or implementation

approach, and supporting tools play equally critical roles. In the meantime, integrating standardized exchange protocols and APIs between modeling languages and tools (e.g., CAD, BPMN, and LML) can enhance consistency, facilitate tool adoption, and reduce the learning curve for new users.

5.2. Challenges of the Tool Developers Domain

The Tool Developers Domain incorporates companies that develop MBSE-related tools in cooperation with developers and vendors [38]. Below, we describe some of the challenges that arise during realization.

5.2.1. Factual/Interdisciplinary Requirements

“Correct” requirements are crucial for a successful product. However, MBSE tool developers often fail to meet user expectations due to a lack of explicit, factual requirements and their application. Kautz et al. [15] stated that MBSE, akin to numerous other new and innovative technologies, has made multiple promises, not all of which have been fulfilled. This leads to a mismatch between supply and demand for MBSE and poses an open challenge. It could be counteracted with interdisciplinary domain research, such as mechanical–electrical software.

Probable Solutions: Tool-developer organizations should engage directly with real-world projects to understand stakeholders’ actual pain points and derive tool requirements from those insights.

5.2.2. Unified Tool Exchange

The success of a new technology relies on its consistency, and MBSE depends on various tools to support the PLC. However, using different tools in different phases of various PLCs probably creates inconsistencies in information exchange, disrupting the potential for a consistent flow of information, which is central to the concept of the digital thread for seamless integration of data across all phases, minimizing information silos and promoting better collaboration [59]. Due to these inconsistencies, the lack of exchange between the tools, and the issue that tools cannot serve all phases of the PLC, models become “static” (document-centered SE) again instead of being used dynamically across all PLC phases. The challenge is, therefore, to unify the exchange among tools so that information can be shared across different databases depending on requirements. This issue is intertwined with the challenge of uniform languages and methods, as only together can redundancies and inconsistencies be avoided. It should be noted that creating a uniform tool that covers the entire PLC is not feasible, as the tools have grown over decades and fulfilled specific tasks. Thus, the unified exchange must enable the integration and automation of various tools that a user can perceive as one uniform working environment. In addition, this integration of tools should also include design, analysis, evaluation, and decision methods, such as QFD, CAD, FEM, CFD, FMEA, decision matrix, and regression analysis.

Probable Solutions: Tool-developer organizations should support bidirectional import/export of models and provide standardized APIs to enable seamless toolchain integration, ensuring interoperability across modeling platforms, disciplines, and lifecycle phases. Implementations of SysML v2 may offer a possible solution in this case, although the tool adaptations are not yet publicly available but at least are currently under development [60].

5.2.3. Unified Standards

Standards can create a framework to automate tasks for methods and tools [61]. Currently, most MBSE tools support standardized formats, such as Extensible Markup Language (XML) or XML Metadata Interchange (XMI). However, files generated by one tool are often only partially readable or not readable by another tool. This is because the formats

allow for variations, and each program has specific peculiarities in creating and reading files. Therefore, the challenge lies in specifying a standardized global exchange format to support all MBSE tools and to create a common platform for processing the PLC. In this respect, INCOSE could be crucial in introducing appropriately unified standards. Such standards should ensure tool interoperability and guide the entire MBSE implementation process. In the area of CAD tools, the interoperability of 3D and 2D geometry models, parts, and design data between different CAD tools is being driven forward by, for instance, the STEP standard [62–64]. Similarly, the MBSE ecosystem requires standards that specify specific file formats and ensure file exchange and interoperability within the MBSE ecosystem.

Probable Solutions: A vendor-neutral exchange standard should be introduced that ensures consistent model semantics and automated tool integration across the MBSE ecosystem [65]. This must be supported by a common interoperability platform that links heterogeneous tools through standardized APIs.

5.3. Challenges of the Academics Domain

The Academics Domain represents academic and research institutions, such as universities, research institutes, and training organizations [38]. Concerning MBSE, this Domain also faces some challenges and shortcomings.

5.3.1. Educational Courses

Education is directly related to workforce training, which is an essential link in the scope of the application. However, very few MBSE courses are offered by universities and educational institutions [53]. Most are short or certificate courses instead of bachelor's or master's programs. Good training is essential to take advantage of the opportunities offered by MBSE. However, delivering and implementing these courses is still an open challenge (in terms of both the budget and well-trained teaching staff).

Probable Solutions: It is essential to integrate MBSE into formal engineering curricula through collaborative programmes involving universities, industry partners, and standardization bodies to ensure sustainable funding and relevant course content. At the same time, institutions should invest in training educators and developing modular, open-access teaching resources to build long-term instructional capacity. Initiatives, such as those undertaken by SparxSystems Europe (e.g., <https://etraining.sparxsystems.eu/>) in training and consultancy, could serve as a model for other tool vendors to strengthen MBSE-related education.

5.3.2. Hands-On Training

In addition to basic training, proper education requires a wealth of examples related to courses following INCOSE's MBSE definition, but they are not freely available. In most cases, only SysML, UML, or Simulink-based models are available, which do not provide a holistic view of MBSE, including all PLC phases. Examples of tool integration, ease of use, reusability, traceability, cybersecurity, and data management are rare or not publicly available [38]. Expanding the range of examples across all phases of the PLC and making them freely available is an open issue and essential for training on MBSE.

Probable Solutions: It is necessary to develop open-access MBSE example repositories that cover all phases of the PLC, created collaboratively by universities, industry partners, and standardization bodies. Additionally, tool vendors should provide publicly accessible reference models demonstrating integration, traceability, cybersecurity, and data management to support comprehensive hands-on training.

5.3.3. Interdisciplinary Research

Interdisciplinary research is crucial to the success of MBSE, as it combines the strengths of different disciplines to develop better systems. However, this collaboration is still an open challenge and is often not practiced. On the one hand, interdisciplinary developments require time to understand the “language” of others. On the other hand, people usually are not used to thinking outside their own box (e.g., internal research, which is more straightforward in calculating budgets). As a result, knowledge and requirements are not exchanged, and, for example, relevant knowledge from mechanical engineering, electrical engineering, or mechatronics is not passed on to software development. Consequently, the expected goal is difficult or impossible to achieve. Thus, introducing interdisciplinary research in academia is increasingly needed to achieve a technological leap and to fulfill the need for systems architects and systems engineers who are able to take the duty of system architecting during design, integration, verification, and validation of the overall system (product).

Probable Solutions: It is important to establish structured interdisciplinary research programs that bring together mechanical, electrical, software, and systems engineers through joint projects, shared laboratories, and interdisciplinary domain curricula. Additionally, universities and industry should incentivize cross-disciplinary collaboration by allocating dedicated funding, recognition, and time resources to foster sustained knowledge exchange and to cultivate skilled systems architects.

The advancement of MBSE is intrinsically linked to the collaboration and integration of its constituent disciplines and domains (Figure 4). The core components and methodologies within one domain are often interdependent with those in other domains, emphasizing that isolated improvements within a single domain are insufficient for advancing the MBSE field as a whole. Therefore, a coordinated and participatory approach across all MBSE Domains (i.e., Tool Developers, Applications, Academics) is essential to ensure balanced and holistic progress. This highlights the need for a systemic alliance, where the strengths and innovations of each domain contribute synergistically to the overall MBSE framework. Building on the identified shortcomings, Figure 5 illustrates critical issues within these challenges that impact the different MBSE Domains. For example, inadequate training and resistance to change not only produce an underqualified workforce but also hinder the adoption of new technologies and collaborative practices. The absence of standardized tool interfaces and data exchange protocols further complicates system integration, resulting in inefficient processes. Additionally, financial constraints—such as unclear investment strategies and insufficient cost-benefit analyses—undermine organizational willingness to pursue necessary transformations. To ensure successful MBSE implementation, organizations must proactively address these interrelated challenges, as their ripple effects can significantly undermine transformation efforts.

5.4. Synopsis

Interdisciplinary research serves as the cornerstone for MBSE, enabling the fusion of diverse perspectives and methodologies to effectively address complex systems engineering challenges. This integration can be further strengthened through structured educational programs and hands-on training initiatives, which not only disseminate theoretical knowledge but also provide practical experience in applying MBSE principles across various contexts. Such initiatives cultivate a workforce equipped with the necessary skills to navigate and innovate within the dynamic landscape of MBSE.

However, the active involvement of tool developers is essential for the effective advancement of MBSE. Consistent implementation processes are necessary to achieve a holistic application, ensuring that all aspects of the system are aligned. Furthermore, unified

languages and cross-disciplinary methods facilitate seamless tool integration, promote interoperability, and maintain consistency across models. These elements collectively support the efficient exchange of information within the MBSE ecosystem.

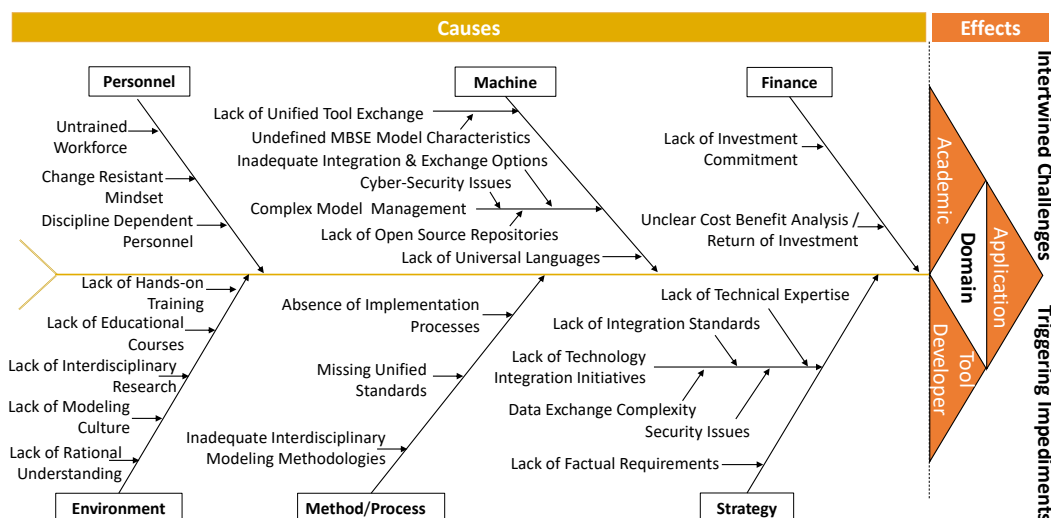


Figure 5. Overview of the critical issues of MBSE in light of the fishbone diagram from Ishikawa and Loftus [66].

6. MBSE Applications Missing Links

In this section, we demonstrate open challenges of applying MBSE to the PLC. One crucial part of the PLC is the system design, which is explained in more detail below. The MBSE approach to domain-specific implementation emphasizes the development of a well-structured systems architecture, serving as a comprehensive blueprint for the system. It also underscores the effective management of models to ensure consistency, facilitate traceability, and to support seamless integration across tools and processes.

As a demonstration example, we create an electric bicycle (e-bike) architecture in a linear fashion using the RFLP approach (Requirements–Functional–Logical–Physical) [67]. We know that this process covers many different aspects and includes various application tools. Our example, therefore, only shows an overview and is created with Enterprise Architect (EA) Version 16 (<https://www.sparxsystems.de/>) and SolidWorks 2022 (<https://www.solidworks.com/>). In the process, we use SysML, CAD volume modeling (tools), and the Finite Element Method (FEM) to realize our network of models and to show a snippet of their individual perspectives, in detail for the load-bearing capacity (marked in orange in the figures), to make the example easy to understand. The entire use case with detailed pictures and sources is available under [68]. It is essential to note that RFLP represents only one of many potential approaches and is not a requirement for the application of MBSE. While it has been selected as an option in this context, alternative methodologies, such as the V-Model or agile approaches, are equally viable. In this demonstration case, although it is relatively simple, we can already observe missing links in applying MBSE and the importance of enabling consistent data exchange. Thus, it necessitates addressing connectivity fragments within the toolchain, both horizontally across the PLC stages and vertically across various functional layers of the e-bike system.

We did not conduct an experimental study [69–71] with the usual metrics and independent comparison groups that would allow for an analysis of the employed method and tools in terms of various KPIs, such as validation time, user satisfaction, error rate, and model consistency. These additional analyses would provide valuable insights, but they fall beyond the scope of this paper and remain of interest for future research.

6.1. E-Bike Demonstration Case

As a starting point for the system architecture, we identify the various user requirements, such as aesthetics, operation modes, lightweight design, and ease of use. While such user needs are not explicit parts of the RFLP approach, they support how to form the technical requirements. These technical requirements are essential for the system design and Verification and Validation (V&V), since they serve as a guideline for synthesizing and evaluating the solution. The technical requirements for the e-bike include aspects of security, performance, production costs, usability, and structure. Each requirement can, in turn, consist of several more detailed requirements. Figure 6 shows an excerpt of the technical requirements, where carry weight (baggage and passengers) is a part of the usability requirement. The requirement specifies a weight (more precisely mass) of up to 150 kg, which means the e-bike's frame must be able to carry a load of 1.5 kN.

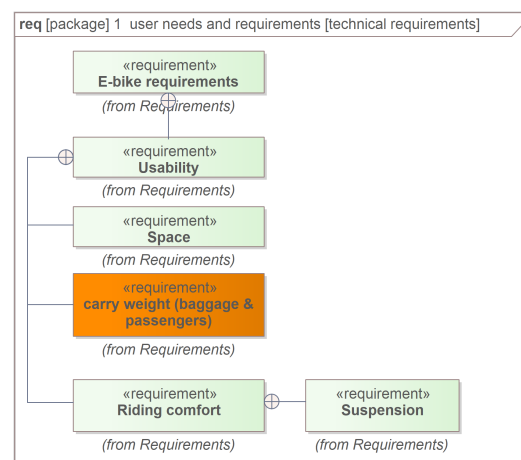


Figure 6. Excerpt of the e-bike's technical requirements (orange rectangle: requirement for the load-bearing capacity).

The structure for the requirements is defined in the requirements architecture to assemble the specifications and to ensure that all requirements meet the business objective. Such architecture can be created in various ways, e.g., through verification, system integration, and functional and non-functional requirements separation. The aim is to keep the model as simple as possible, but to consider all requirements.

The next step is to create the functional architecture. This architecture should identify the purpose of the system, the flow of materials, energy, and information between the individual functions and their relationships. For the e-bike, the functional architecture includes functions, such as “decelerate”, “ride bike”, “accelerate”, etc. For instance, the function “ride bike” includes weight support, as shown before as a technical requirement.

The logical architecture is built once the requirements and functional design have been created. This architecture aims to clarify the system components responsible for deploying functions and to provide a comprehensive overview of the behavior and functionality of the system. It is an extended version of the functional architecture and allows a better understanding of how systems operate during service. Figure 7 enhances the functional architecture of the e-bike by structures, such as wheels and powertrain.

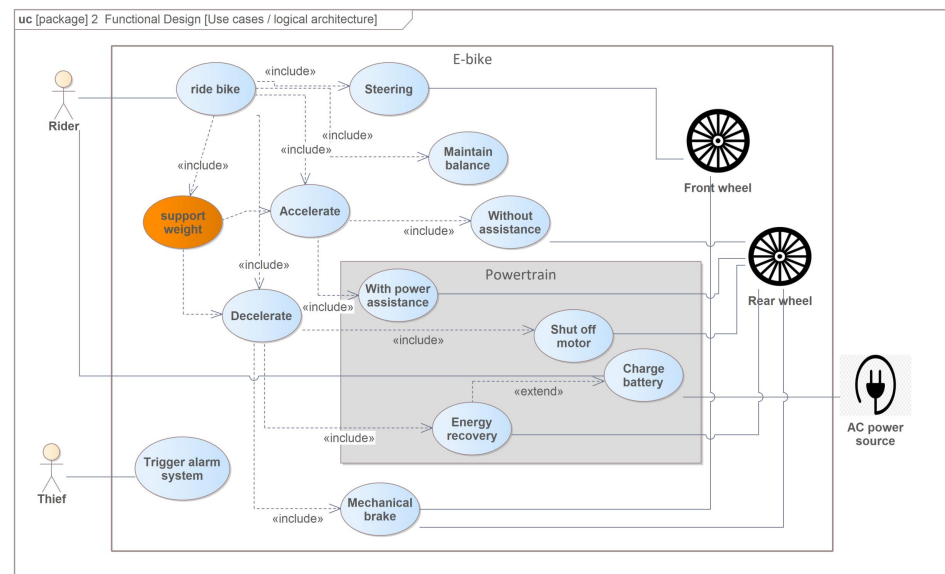


Figure 7. Overview of the e-bike's logical architecture (orange oval: capability for the load-bearing capacity).

Based on the previous parts, the next step is the physical architecture, which outlines the placement of the logical components in the physical environment of the system (system elements and physical interfaces) and illustrates how the system architecture relates from one step to another to ensure a clear understanding of the system structure and to fulfill the system requirements [72]. Figure 8 shows an excerpt of the architecture, in particular, the frame of the e-bike with its relationships, since this part has to fulfill the load-bearing capacity.

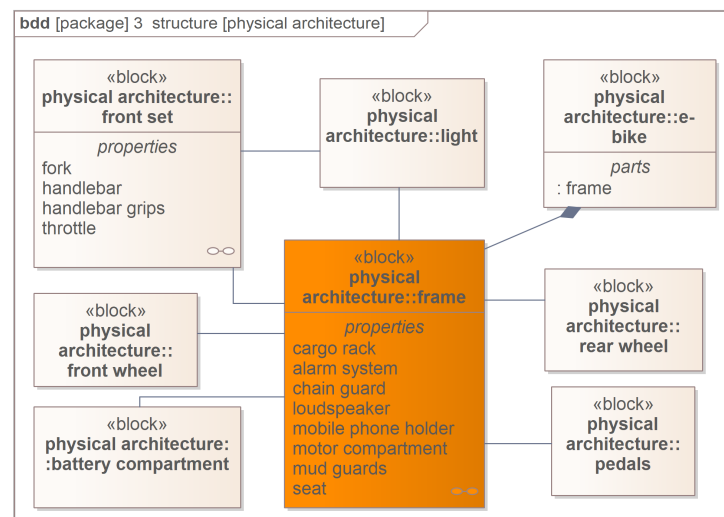


Figure 8. Excerpt of the e-bike's physical architecture (orange rectangle: relevant block for the load-bearing capacity).

To verify if the frame sustains 1.5 kN, the definition of the block frame (Figure 8) alone as part of the physical architecture, as it can be modeled in Enterprise Architect, is unsatisfactory. Instead, a tool is needed to model the frame with its defined properties (as in that architecture), and a specific load (1.5 kN) can be applied to analyze the load-bearing capacity of the frame. Figure 9 shows the e-bike's frame drawn and tested in SolidWorks. For this carry weight test, the material selection plays a crucial role, as it may conflict with production cost requirements.

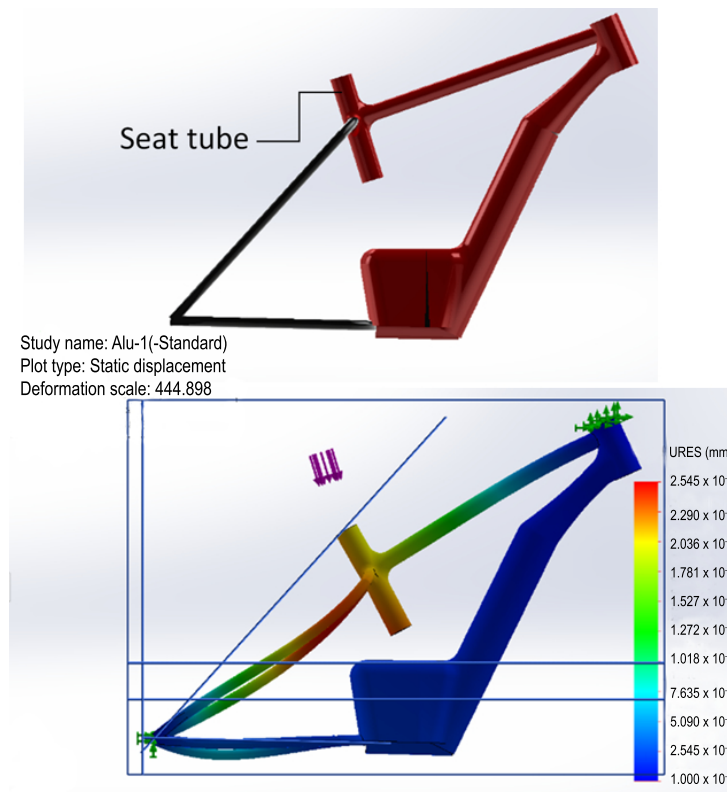


Figure 9. E-bike frame simulation in SolidWorks (purple arrows represent the load applied; green arrows represent the clampings, URES: magnitude of the displacement vector).

6.2. Discussion

The e-bike demonstration case was selected due to its interdisciplinary nature. It integrates mechanical (frame and drivetrain), electrical (motor and battery), embedded software (control logic), and human-system interaction (user interface, safety). Its moderate complexity enables a clear illustration of MBSE workflows.

Even this simple case highlights various challenges:

- **Tool interoperability gap:** Although the frame's mass property (150 kg $\rightarrow$ 1.5 kN load) is defined within the SysML block as text, transferring these data to SolidWorks required manual intervention (Figures 8 and 9). Consequently, any modification in the SysML model necessitates a manual update in the CAD environment, introducing a significant risk of data inconsistency. This case underscores the unified tool exchange challenge (Section 5.2).
- **Language limitation:** The aesthetic requirement, such as “sleek, modern appearance,” could not be formally modeled in SysML—it remained a text note, breaking traceability to styling CAD surfaces. This reflects the unified languages and methods gap (Section 5.1). Although language limitations are a long-standing challenge, this issue remains unresolved and is currently a factor that should not be underestimated in the digital transformation.
- **Workforce dependency:** The FEM validation required a specialist in structural mechanics—outside the core MBSE modeling team—highlighting the available workforce challenge (Section 5.1). There is a growing trend toward specialists sharing the workload, while exchange and collaboration between different disciplines are becoming increasingly challenging. The use of diverse technologies within a single discipline or across multiple disciplines may enhance problem-solving in one area but

intensifies dependencies in another one. This underscores the importance of education and training.

- Seamless integration throughout the PLC: A critical gap persists in the seamless orchestration of data across PLC phases and diverse engineering disciplines. To effectively operationalize sustainability, data frameworks must evolve from static repositories into dynamic, semantically rich models that explicitly represent upcycling, recycling, and even broader circular economy pathways. This structural transformation is essential to bridge the gap between early-stage conceptual design and tangible end-of-life management—ensuring that sustainability is not merely an afterthought, but an integrated, traceable dimension throughout the system's lifecycle.

The application of MBSE often relies on a diverse set of tools, yet integration across these tool landscapes remains a significant challenge. In our shown use case, we utilize only two distinct tools for developing and validating the system architecture (see Figure 10). Although properties are defined in the architecture and used in the simulation, the transfer is completed manually. Thus, inconsistencies may arise if several changes occur over time (the property is not automatically changed in both places). The challenge is that there is no standardized way of automatically exchanging data. While both tools offer interfaces, developing communication and integration requires an implementation effort that should not be underestimated. In this context, research activities try to close this gap. For instance, Demuth et al. [73] present the DesignSpace, an innovative engineering infrastructure, that enables integrating and managing different development artifacts. The DesignSpace promotes distributed collaboration and provides services, such as consistency checks, traceability, and transformation of artifacts. The goal is to enable seamless interaction between teams and tools through improved (meta-)modeling and engineering history documentation.

However, even if interoperability has been established, whether through APIs or other solutions, it can be undermined by updates and enhancements. In addition, proprietary developments usually only work for specific cases that are not freely available for users of various tools, which complicates the exchange again. Thus, offering and maintaining interoperability involves considerable effort. Integrating more tools, such as circuit design, fluid dynamics, 1D simulation, and project management, can be even more complex. But integrating multiple tools is essential to achieve consistency and interconnected digital models over time, where information is not redundant but is retained and forwarded as needed.

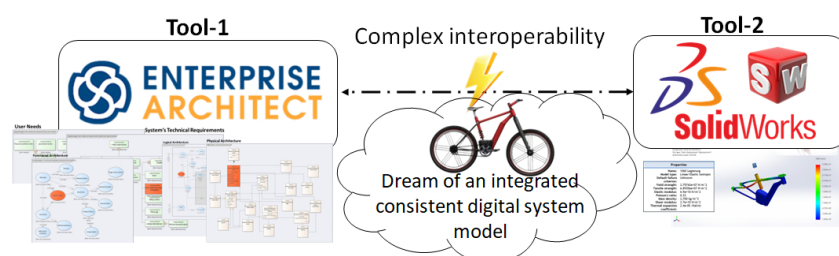


Figure 10. Model and Tool interoperability as an open challenge.

Neither a single tool nor a single language can comprehensively address the entire PLC. Moreover, the previously defined three MBSE Domains are so closely interlinked that even minor changes in one Domain can have a ripple effect on others, analogously to the butterfly effect in chaos theory [74]. Therefore, establishing unified standards is important to promote interoperability and consistency across models and tools. Such standards could also catalyze new interdisciplinary research, driving further exploration to address emerging requirements of different disciplines and MBSE Domains.

7. Conclusions and Outlook

This study contributes to a more nuanced understanding of MBSE by critically examining current gaps—thereby enabling practitioners to evaluate the adoption of MBSE based on evidence rather than on prevailing hypes. We conclude that the different domains and disciplines face various challenges when applying MBSE, including the absence of a unified process framework, insufficient workforce preparedness, and a lack of standardized modeling languages and methodologies. These shortcomings hinder holistic system development and complicate interdisciplinary domain integration. Notably, academic institutions face difficulties in cultivating MBSE-competent professionals, highlighting the urgent need for curriculum enhancements and experiential learning opportunities. Furthermore, existing modeling tools exhibit limited capability to support end-to-end modeling along the full PLC; effective application often necessitates tool interoperability or hybrid toolchains. Therefore, a synchronized orchestration among languages, methods, and tools is essential to enable truly integrated and sustainable MBSE.

Future research should also explore the integration of emerging technologies, such as AI, IoT, blockchain, and quantum computing, within MBSE to strengthen its capabilities and to improve model maturity. In particular, studies should investigate how these technologies can automate modeling tasks, enhance semantic consistency, and support real-time system validation. Further research is also needed to evaluate the impact on lifecycle efficiency, system reliability, and the scalability of MBSE practices across diverse disciplines and domains.

In conclusion, the future development of MBSE research should address current challenges while anticipating and adapting to evolving technological landscapes and industry needs.

Author Contributions: Conceptualization, A.K.; methodology, A.K.; resources, G.G.; data curation, A.K. and S.S.; writing—original draft preparation, A.K. and S.S.; writing—review and editing, G.G. and K.Z.; visualization, A.K. and S.S.; supervision, K.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been partly supported by the COMET-K2 Center of the Linz Center of Mechatronics (LCM), funded by the Austrian federal government and the federal state of Upper Austria. In addition, the paper is supported by Johannes Kepler Open Access Publishing Fund and the federal state of Upper Austria.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The entire use case with detailed pictures and sources is available under Khandoker et al. [68].

Acknowledgments: During the preparation of this manuscript the author(s) used ChatGPT-5 and Google Gemini 3 in order to occasionally improve the readability of paragraphs. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lee, E.A. Cyber-physical Systems—Are Computing Foundations Adequate? In Proceedings of the Position Paper for Nsf Workshop on Cyber-Physical Systems: Research Motivation, Techniques and Roadmap, Austin, TX, USA, 16–17 October 2006.
2. Lee, E.A. Cyber Physical Systems: Design Challenges. In *Proceedings of the 11th IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC 2008)*, Orlando, FL, USA, 5–7 May 2008; IEEE Computer Society: New York, NY, USA, 2008; pp. 363–369.

3. Hermann, M.; Pentek, T.; Otto, B. *Design Principles for Industrie 4.0 Scenarios: A Literature Review*; Working Paper; TU Dortmund: Dortmund, Germany, 2015.
4. Ramos, A.L.; Ferreira, J.V.; Barceló, J. Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Trans. Syst. Man, Cybern. Part C (Appl. Rev.)* **2012**, *42*, 101–111. [\[CrossRef\]](#)
5. Akundi, A.; Lopez, V. A Review on Application of Model Based Systems Engineering to Manufacturing and Production Engineering Systems. *Procedia Comput. Sci.* **2021**, *185*, 101–108. [\[CrossRef\]](#)
6. Singh, V.; Willcox, K.E. Decision-Making Under Uncertainty for a Digital Thread-Enabled Design Process. *J. Mech. Des.* **2021**, *143*, 091707. [\[CrossRef\]](#)
7. Harrison, N.; Whitfield, R.; Odukoya, K.; Powell, A. Model-based Engineering and the Digital Thread. In *Proceedings of the International Naval Engineering Conference (INEC)*; Institute of Marine Engineering, Science and Technology (IMarEST): London, UK, 2022; pp. 1–10. [\[CrossRef\]](#)
8. Xu, Z.; Yang, S. Research and Application of the Data Model System in MBSE. *IEEE Access* **2025**, *13*, 195167–195186. [\[CrossRef\]](#)
9. Failla, L.; Rossoni, M.; Colombo, G. Ontology-Based Product Lifecycle Management: Insights from a Proof-of-Concept Implementation. *IEEE Access* **2025**, *13*, 179295–179314. [\[CrossRef\]](#)
10. Sapel, P.; Gannouni, A.; Abdelrazeq, A.; Hopmann, C.; Schmitt, R.H. Purposeful decision-support with Digital Shadows: A model catalog architecture and its implementation for production scheduling in the injection molding domain. *Int. J. Prod. Res.* **2025**, 1–25. [\[CrossRef\]](#)
11. Wilking, F.; Horber, D.; Goetz, S.; Wartzack, S. Utilization of system models in model-based systems engineering: Definition, classes and research directions based on a systematic literature review. *Des. Sci.* **2024**, *10*, 1–28. [\[CrossRef\]](#)
12. Chami, M.; Bruel, J.M. A Survey on MBSE Adoption Challenges. In *Proceedings of the Systems Engineering Conference of the Europe, Middle-East and Africa (EMEA) Sector of INCOSE (EMEASEC 2018)*, Berlin, Germany, 5–7 November 2018.
13. D'Ambrosio, J.; Soremekun, G. Systems engineering challenges and MBSE opportunities for automotive system design. In *Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2017, Banff, AB, Canada, 5–8 October 2017*; IEEE: New York, NY, USA, 2017; pp. 2075–2080.
14. Zhang, Q.; Liu, J.; Chen, X. A Literature Review of the Digital Thread: Definition, Key Technologies, and Applications. *Systems* **2024**, *12*, 70. [\[CrossRef\]](#)
15. Kautz, O.; Roth, A.; Rumpe, B. Achievements, Failures, and the Future of Model-Based Software Engineering. In *The Essence of Software Engineering*; Gruhn, V., Striemer, R., Eds.; Springer: Berlin/Heidelberg, Germany, 2018; pp. 221–236.
16. Carroll, E.R.; Malins, R.J. *Sandia Report Systematic Literature Review: How Is ModelBased Systems Engineering Justified?* Technical Report SAND2016-2607; Sandia National Laboratories: Albuquerque, NM, USA, 2016.
17. Cloutier, R. Migrating from a Waterfall Systems Engineering Approach to an Object Oriented Approach—Lessons Learned. In *Proceedings of the ICSE and INCOSE Region II Conference*, Las Vegas, NV, USA, 15–18 September 2004.
18. Daly, H.C.; Grogan, P.T. Toward a Reference Architecture for Digital and Model-Based Engineering Information Systems. In *Proceedings of the Recent Trends and Advances in Model Based Systems Engineering*; Springer International Publishing: Cham, Switherland, 2022; pp. 3–13.
19. Henderson, K.; McDermott, T.A.; Salado, A. MBSE adoption experiences in organizations: Lessons learned. *Syst. Eng.* **2024**, *27*, 214–239. [\[CrossRef\]](#)
20. Mousavi, B.A.; Azzouz, R.; Heavey, C.; Ehm, H. A Survey of Model-Based System Engineering Methods to Analyse Complex Supply Chains: A Case Study in Semiconductor Supply Chain. *IFAC-PapersOnLine* **2019**, *52*, 1254–1259. [\[CrossRef\]](#)
21. Husung, S.; Gerhard, D.; Jacobs, G.; Kowalski, J.; Rumpe, B.; Zeman, K.; Zerwas, T. Model Signatures for Design and Usage of Simulation-Capable Model Networks in MBSE. In *Proceedings of the Product Lifecycle Management. PLM in Transition Times: The Place of Humans and Transformative Technologies*; Springer: Cham, Switherland, 2023; pp. 155–164.
22. Gräßler, I.; Oleff, C. *Systems Engineering: Verstehen und Industriell Umsetzen*; Springer: Berlin/Heidelberg, Germany, 2022.
23. Lu, J.; Wen, Y.; Gürdür, D.; Liu, Q.; Törngren, M. MBSE Applicability Analysis in Chinese Industry. *INCOSE Int. Symp.* **2018**, *28*, 1037–1051. [\[CrossRef\]](#)
24. Bone, M.; Cloutier, R. The Current State of Model Based Systems Engineering: Results from the OMG™ SysML Request for Information 2009. In *Proceedings of the 8th Conference on Systems Engineering Research*, Hoboken, NJ, USA, 17–19 March 2010.
25. Motamedian, B. MBSE Applicability Analysis. *Int. J. Sci. Eng. Res.* **2013**, *4*, 7.
26. Cloutier, R.; Obiako, I. *Model-Based Systems Engineering Adoption Trends 2009–2018*; Technical Report; SEBoK: Changwon, Republic of Korea, 2018.
27. Huld, T.; Stenius, I. State-of-practice survey of model-based systems engineering. *Syst. Eng.* **2019**, *22*, 134–145. [\[CrossRef\]](#)
28. McDermott, T.; Hutchison, N.; Clifford, M.; Van Aken, E.; Salado, A.; Henderson, K. *Benchmarking the Benefits and Current Maturity of Model-Based Systems Engineering across the Enterprise: Results of the MBSE Maturity Survey*; Technical Report; Number: SERC-2020-SR-001; Stevens Institute of Technology: Hoboken, NJ, USA, 2020.

29. Akundi, A.; Ankobiah, W.; Mondragon, O.; Luna, S. Perceptions and the extent of Model-Based Systems Engineering (MBSE) use—An industry survey. In *Proceedings of the IEEE International Systems Conference (SysCon)*; IEEE: New York, NY, USA, 2022; pp. 1–7.
30. Cordero, S.S.; Gateau, T.; Vingerhoeds, R.A. MBSE Challenges in the Concurrent Preliminary Design of CubeSats: Nanospace Study. In *Proceedings of the IEEE International Systems Conference, SysCon 2022, Montreal, QC, Canada, 25–28 April 2022*; IEEE: New York, NY, USA, 2022; pp. 1–8.
31. Kim, S.Y.; Wagner, D.; Jimenez, A. Challenges in applying model-based systems engineering: Human-centered design perspective. In *Proceedings of the INCOSE Human Systems Integration Conference (HSI 2019)*, Biarritz, France, 11–13 September 2019.
32. Albers, A.; Zingel, C. Challenges of Model-Based Systems Engineering: A Study towards Unified Term Understanding and the State of Usage of SysML. In *Proceedings of the Smart Product Engineering; Lecture Notes in Production Engineering*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 83–92.
33. Pinquie, R.; Romero, V.; Noel, F. Survey of Model-Based Design Reviews: Practices & Challenges? In *Proceedings of the DESIGN2022 17th International Design Conference*, Cavtat, Croatia, 23–26 May 2022.
34. Blott, J.; Buchholz, C. Challenges of implementing MBSE in industry: A tool vendor experience. In *Proceedings of the Engineering for a Changing World: Proceedings: 60th ISC, Ilmenau Scientific Colloquium, Technische Universität Ilmenau, 4–8 September 2023*; Sattler, K.U., Ed.; TU Ilmenau: Ilmenau, Germany, 2023; p. 17.
35. Vogelsang, A.; Amorim, T.; Pudlitz, F.; Gersing, P.; Philipps, J. Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry. In *Proceedings of the Product-Focused Software Process Improvement—18th International Conference, PROFES 2017, Innsbruck, Austria, 29 November–1 December 2017*; Felderer, M., Fernández, D.M., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2017; Volume 10611, pp. 182–198.
36. Cornu, C.; Chapurlat, V.; Quiot, J.M.; Irigoin, F. A maturity model for the deployment of Systems Engineering processes. In *Proceedings of the IEEE International Systems Conference (SysCon)*; IEEE: New York, NY, USA, 2012; pp. 1–6.
37. Yang, P.; Lu, J.; Feng, L.; Wu, S.; Wang, G.; Kiritsis, D. A Knowledge Management Approach Supporting Model-Based Systems Engineering. In *Proceedings of the Trends and Applications in Information Systems and Technologies—Volume 2, WorldCIST 2021, Terceira Island, Azores, Portugal, 30 March–2 April 2021*; Advances in Intelligent Systems and Computing; Springer: Cham, Switzerland, 2021; Volume 1366, pp. 581–590.
38. Khandoker, A.; Sint, S.; Gessl, G.; Zeman, K.; Jungreitmayr, F.; Wahl, H.; Wenigwieser, A.; Kretschmer, R. Towards a logical framework for ideal MBSE tool selection based on discipline specific requirements. *J. Syst. Softw.* **2022**, *189*, 111306. [CrossRef]
39. History.com Editors. Aristotle. 2023. Available online: <https://www.history.com/topics/ancient-history/aristotle> (accessed on 30 January 2024).
40. Cordon, C.P. System Theories: An Overview of Various System Theories and Its Application in Healthcare. *Am. J. Syst. Sci.* **2013**, *2*, 13–22.
41. Furterer, S. *Systems Engineering: Holistic Life Cycle Architecture Modeling and Design with Real-world Applications*; CRC Press: Boca Raton, FL, USA, 2021.
42. von Bertalanffy, L. *Kritische Theorie der Formbildung, Abhandlungen zur Theoretischen Biologie*; Gebrüder Borntraeger: Berlin, Germany, 1928.
43. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
44. Wiener, N. *Cybernetics: Second Edition: Or the Control and Communication in the Animal and the Machine*, 2nd ed.; Martino Fine Books: Eastford, CT, USA, 2013; reprint.
45. Seising, R. Cybernetics, system(s) theory, information theory and Fuzzy Sets and Systems in the 1950s and 1960s. *Inf. Sci.* **2010**, *180*, 4459–4476. [CrossRef]
46. Gorod, A.; Sauser, B.J.; Boardman, J.T. System-of-Systems Engineering Management: A Review of Modern History and a Path Forward. *IEEE Syst. J.* **2008**, *2*, 484–499. [CrossRef]
47. Haskins, C. A historical perspective of MBSE with a view to the future. *INCOSE Int. Symp.* **2011**, *21*, 493–509. [CrossRef]
48. Honour, E.C. A historical perspective on systems engineering. *Syst. Eng.* **2018**, *21*, 148–151. [CrossRef]
49. CFI Team. Systems Thinking. 2015. Available online: <https://corporatefinanceinstitute.com/resources/management/systems-thinking/> (accessed on 30 January 2024).
50. Wymore, A.W. *Model-Based Systems Engineering*; CRC Press: Boca Raton, FL, USA, 1993.
51. Dickerson, C.E.; Mavris, D.N. A Brief History of Models and Model Based Systems Engineering and the Case for Relational Orientation. *IEEE Syst. J.* **2013**, *7*, 581–592. [CrossRef]
52. INCOSE. *Systems Engineering Vision 2020*; Technical Report INCOSE-TP-2004-004-02; INCOSE: San Diego, CA, USA, 2007.
53. Khandoker, A.; Sint, S.; Wimmer, M.; Zeman, K. An Interdisciplinary Course on Model-Based Systems Engineering. In *Proceedings of the ACM/IEEE International Conference on Model Driven Engineering Languages and Systems, MODELS 2023 Companion, Västerås, Sweden, 1–6 October 2023*; IEEE: New York, NY, USA, 2023; pp. 102–109.

54. Huss, M.; Herber, D.R.; Borky, J.M. An Agile Model-Based Software Engineering Approach Illustrated through the Development of a Health Technology System. *Software* **2023**, *2*, 234–257. [\[CrossRef\]](#)
55. Micouin, P. MBSE, What is Wrong with SysML—First Issue. Technical Report, hal-02070455v2. 2019. Available online: <https://hal.science/hal-02070455v2> (accessed on 1 June 2025).
56. Ahlbrecht, A.; Lukić, B.; Zaeske, W.; Durak, U. Exploring SysML v2 for Model-Based Engineering of Safety-Critical Avionics Systems. In *Proceedings of the 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*; IEEE: New York, NY, USA, 2024; pp. 1–8. [\[CrossRef\]](#)
57. Epp, J.; Robert, T.; Ruch, O.; Olechowski, A. Towards SysML v2 as a Variability Modeling Language. In *Proceedings of the 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*; IEEE: New York, NY, USA, 2023; pp. 251–256. [\[CrossRef\]](#)
58. Gray, J.; Rumpe, B. Reflections on the standardization of SysML 2. *Softw. Syst. Model.* **2021**, *20*, 287–289. [\[CrossRef\]](#)
59. Eigner, M. *System Lifecycle Management*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2021.
60. Boelsen, K.; May, M.; Jacobs, G.; Mennicken, M.; Moers, F.; Zerwas, T.; Höpfner, G. SysML v2 based modelling guidelines for mechanical system elements. *Forsch. Ingenieurwesen* **2025**, *89*, 11. [\[CrossRef\]](#)
61. Lerner, M.; Slovin, M. Methods, tools and standards (software engineering). In *Proceedings of the [1991 Proceedings] Fourth Software Engineering Standards Application Workshop*; IEEE: New York, NY, USA, 1991; pp. 143–146.
62. Pratt, M.J. Introduction to ISO 10303—the STEP Standard for Product Data Exchange. *J. Comput. Inf. Sci. Eng.* **2001**, *1*, 102–103. [\[CrossRef\]](#)
63. Pratt, M.J. A New ISO 10303 (STEP) Resource for Modeling Parameterization and Constraints. *J. Comput. Inf. Sci. Eng.* **2004**, *4*, 339–351. [\[CrossRef\]](#)
64. Lanza, R. Improving and Implementing the STEP ISO 10303 Standard for Design, Analysis and Structural Test Data Correlation. Ph.D. Thesis, Norwegian University of Science and Technology, Trondheim, Norway, 2020.
65. Cibrián, E.; Olivert-Iserte, J.; Díez-Fenoy, C.; Mendieta, R.; Llorens, J.; María Álvarez Rodríguez, J. Ensuring Semantic Consistency in SysML v2 Models Through Metamodel-Driven Validation. *IEEE Access* **2025**, *13*, 121444–121457. [\[CrossRef\]](#)
66. Ishikawa, K.; Loftus, J.H. *Introduction to Quality Control*; 3A Corporation: Tokyo, Japan, 1990.
67. Kleiner, S.; Kramer, C. Model Based Design with Systems Engineering Based on RFLP Using V6. In *Proceedings of the Smart Product Engineering*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 93–102.
68. Khandoker, A.; Sint, S.; Gessl, G.; Zeman, K. Dataset of “Toward Demystifying the Missing Links in Model Based Systems Engineering (MBSE)”. 2026. Available online: <https://researchdata.tuwien.ac.at/records/npj9z-j2c96> (accessed on 26 January 2026).
69. Panchal, J.H.; Szajnfarder, Z. Experiments in systems engineering and design research. *Syst. Eng.* **2017**, *20*, 529–541. [\[CrossRef\]](#)
70. Elm, J.P.; Goldenson, D.R. *The Business Case for Systems Engineering Study: Results of the Systems Engineering Effectiveness Survey*; Special Report CMU/SEI-2012-SR-009; Carnegie Mellon University, Software Engineering Institute: Pittsburgh, PA, USA, 2012.
71. Runeson, P.; Höst, M. Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **2009**, *14*, 131–164. [\[CrossRef\]](#)
72. ISO/IEC/IEEE 24748-1:2018; Systems and Software Engineering Life Cycle Management Part 1: Guidelines for Life Cycle Management. International Organization for Standardization: Geneva, Switzerland, 2018.
73. Demuth, A.; Riedl-Ehrenleitner, M.; Nöhrer, A.; Hehenberger, P.; Zeman, K.; Egyed, A. DesignSpace: An infrastructure for multi-user/multi-tool engineering. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing, Salamanca, Spain, 13–17 April 2015*; Wainwright, R.L., Corchado, J.M., Bechini, A., Hong, J., Eds.; ACM: New York, NY, USA, 2015; pp. 1486–1491.
74. Gleick, J. *Chaos: Making a New Science*; Penguin: New York, NY, USA, 1988.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.