applied sciences

MDPI

Review

Systematic Literature Review of MBSE Tool-Chains

Junda Ma [1], Guoxin Wang [1], Jinzhi Lu [2,*], Hans Vangheluwe [3], Dimitris Kiritsis [2] and Yan Yan [1]

1 School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China; mjd2015@sina.cn (J.M.); wangguoxin@bit.edu.cn (G.W.); yanyan331@bit.edu.cn (Y.Y.)
2 EPFL—École Polytechnique Fédérale de Lausanne, Route Cantonale, 1015 Lausanne, Switzerland; dimitris.kiritsis@epfl.ch
3 Department of Computer Science, University of Antwerp/Flanders Make, Middelheimlaan 1, B-2020 Antwerp, Belgium; hans.vangheluwe@uantwerp.be
* Correspondence: jinzhi.lu@epfl.ch

Abstract: Currently, the fundamental tenets of systems engineering are supported by a model-based approach to minimize risks and avoid design changes in late development stages. The models are used to formalize, analyze, design, optimize, and verify system development and artifacts, helping developers integrate engineering development across domains. Although model-based development is well established in specific domains, such as software, mechanical systems, and electrical systems, its role in integrated development from a system perspective is still a challenge for industry. The model-based systems engineering (MBSE) tool-chain is an emerging technique in the area of systems engineering and is expected to become a next-generation approach for supporting model integration across domains. This article presents a literature review to highlight the usage and state of the art to generally specify the current understanding of MBSE tool-chain concepts. Moreover, the results are used for identifying the usage, advantages, barriers, concerns, and trends of tool-chain development from an MBSE perspective.

Keywords: MBSE; tool-chain; literature review; model; integration

check for updates

Citation: Ma, J.; Wang, G.; Lu, J.; Vangheluwe, H.; Kiritsis, D.; Yan, Y. Systematic Literature Review of MBSE Tool-Chains. *Appl. Sci.* **2022**, *12*, 3431. https://doi.org/10.3390/app12073431

Academic Editor: Antonella Petrillo

Received: 2 March 2022
Accepted: 25 March 2022
Published: 28 March 2022

1. Introduction

Model-based systems engineering (MBSE) has been defined by the International Council on Systems Engineering (INCOSE) as "the formalized application of modeling to support system requirements, design, analysis, verification and validation (V&V) activities beginning in the conceptual design phase and continuing throughout development and later life cycle" [1]. MBSE is expected to formalize system artifacts and development during the entire life-cycle using models. The modeling techniques support structural analysis and design, data flow diagrams, process modeling, CAD, et al. However, such models always have heterogeneous data structures and formats leading to difficulties in the management and integration of domain-specific knowledge and information for the entire system development.

During system development, a tool-chain is proposed as a set of programming tools to perform complex system development tasks or to create a system solution [2]. Specifically, MBSE tool-chains (referring to the integration of modeling, simulation, and design tools) are considered as solutions for implementing multi-domain and interdisciplinary design [3]. In the development of MBSE tool-chains, different techniques and tools are selected to create an open environment to control, manage, and integrate heterogeneous models and data. Compared with others, such as tool-chains for model-based design and software engineering, MBSE tool-chains are relatively new concepts for product development using MBSE. One definition of MBSE tool-chains [3] is proposed as one tool-chain consisting of two or more modeling, simulation, and design tools that, when combined, can support and construct a system engineering workflow with the following features:

1. The tools in the workflow support system formalisms in the activities of product development, such as system requirements, design, analysis, and V&V.
2. The tools in the workflow support the formalisms of system functional and non-functional aspects, such as process management and dependability.
3. The tools are required to support heterogeneous model integration for system-level verification.
4. The tools are required to support data, knowledge, and information exchange, and to execute model transformations within the system development process.
5. While constructing the MBSE tool-chains, a unified ontology is defined to support the entire formalism of systems engineering workflows.
6. The tool-chain enables the integration of existing engineering tools and system development platforms for project management and process management.

The first two features aim to deal with formalisms of system artifacts and system development. Feature 3 is expected to support integrated verification based on simulation across domains, such as co-simulation and Modelica [4–6]. Feature 4 aims to support data sharing among MBSE tools across system life-cycles. Features 5–6 aim to manage traceability and to implement design automation based on system engineering workflow.

When developing MBSE tool-chains, different techniques are selected to create an environment to control, integrate, and manage heterogeneous models and data [3]. However, there is a risk of R&D cost for MBSE tool-chain development when the tool-chains are not developed in a manner that supports the target system development. On the other hand, weak communication between tool-chain users and developers also leads to a poor understanding of tool functionalities and insufficient motivation for technique selections. Moreover, unspecified data integration leads to tool-chains with poor scalability. Thus, it is important for researchers to understand MBSE tool-chain concepts and to capture the requirements of tool-chain development. In this paper, we review the literature to provide clues regarding the required direction for future research.

Because of increasing interest in MBSE, several investigations have been independently proposed by researchers from different organizations. Estefan et al. [7] provided one survey for investigating MBSE methodologies. Bita and Jinzhi provided a questionnaire survey about MBSE for industries, which provides clues to the potential of MBSE [8,9]. Huldt et al. [10] provided a literature review about MBSE to understand MBSE and related concepts. To understand the application of MBSE tools in specific domains, Rashid provided a literature review for embedded systems [11]. These studies focus on MBSE and investigate how MBSE techniques are used in academia and industry.

Moreover, some researchers have investigated specific techniques that are used in tool-chains. Hutchinson provided a survey for tools used in model-driven engineering in the software engineering process [12]. The scientific challenges and benefits of co-simulations were identified from a questionnaire survey and a literature review [13]. Furthermore, model-based tool integration, an important solution supporting tool-chain development, was investigated through literature reviews by Asplund et al. [14]. These researches are a part of MBSE tool-chain techniques. Compared with these contributions, our paper focuses on a survey of MBSE tool-chains to understand their concepts and related technologies. In contrast to previous research studies, we provide a new blueprint from the perspective of systems engineering and tool integration, aiming to provide guidance for MBSE tool-chain development.

The rest of the paper is organized as follows. We discuss the research methodology in Section 2. The results of our survey are illustrated in Section 3. In Section 4, we discuss the features of MBSE tool-chains though the findings. Finally, in Section 5, we present our conclusions.

2. Research Methodology

To implement the literature review, Kitchenham's procedure was used to support the whole search strategy based on seven steps [15]: (1) research questions; (2) search processes;

(3) inclusion criteria; (4) exclusion criteria; (5) quality assessment; (6) data collection; and (7) data analysis.

2.1. Research Questions

The objective of our study was to investigate the current state of the art pertaining to MBSE tool-chains. In particular, peer-reviewed literature was used to characterize and synthesize available contributions with respect to the investigated research topics, research types, research methods, contribution types, and research gaps in the context of MBSE tool-chains. The main research question driving this research and reflecting our objective was as follows (four research questions were posed, each of which addresses a particular aspect of MBSE tool-chains):

1. **RQ 1**: What is the intensity of the research on MBSE tool-chains? This research question was designed to identify the scope of research topics associated with MBSE tool-chains. The investigated views are summarized in Table 1.
2. **RQ 2**: What are the most frequently investigated research areas related to MBSE tool-chains? Furthermore, what application areas have they been used in? This research question was used to analyze how MBSE tool-chains support researchers' work in systems engineering and to capture which domains they are in. It was analyzed by the given information about application domains and functionalities of MBSE tool-chains, as illustrated in Table 1.
3. **RQ 3**: What are the most frequently applied research types and research methods? This question was used to understand which techniques can be used to construct MBSE tool-chains. Four aspects were considered: social perspective, process, information, and technical perspective based on the SPIT framework for tool-chain development [3]:
    - The social perspective refers to an explicit network of stakeholders related to system development as well as environmental constraints and policies regarding MBSE tool-chain development and implementation. We focused on identifying stakeholders, referring to authors and their relevant studies.
    - The process aspect refers to development processes of the target systems developed using MBSE tool-chains. For this survey, we focused on the development phases in which the tool-chains are implemented.
    - The information refers to system artifacts of complex systems. We focused on understanding the life properties from a system engineering perspective, which the MBSE tool-chains aim at. The life properties were selected based on Olivier's definitions [16], including: (1) quality; (2) reliability; (3) safety; (4) flexibility; (5) robustness; (6) durability; (7) scalability; (8) adaptability; (9) usability; (10) interoperability; (11) sustainability; (12) testability; (13) modularity; (14) resilience; (15) extensibility; (16) agility; (17) manufacturability; (18) repairability; and (19) evolvability.
    - The technical perspective refers to the related technologies for constructing MBSE tool-chains. We expected to understand the contents related to: (1) modeling purposes; (2) modeling languages; (3) modeling methods; (4) simulation methods; (5) modeling tools; (6) solutions for integrated verification; (7) tool integration; (8) testing; (9) process modeling; (10) process control and management; (11) model checking; and (12) optimization [17].
4. **RQ 4**: What contributions are provided by research related to MBSE tool-chains? This question was used to clarify the authors' contributions. We summarized the authors' contributions from the papers, including: (1) framework/method/technique; (2) guidelines; (3) lessons learned; (4) models; (5) tools; (6) surveys; and (7) tool-chain architectures. Moreover, metrics accessing MBSE tool-chains were surveyed from the literature reviews, which included: (1) dependency; (2) interoperability; (3) traceability; (4) integrated capability; (5) reusability; (6) design efficiency; and (7) scalability.

Table 1. Basic information about publications.

Id	Title	Cardinality [1]	RQ	Description
1	Publication Year	1:1	**RQ 1**	The publication time.
2	Publisher	1:1	**RQ 1**	The publisher's name, such as IEEE.
3	Publication Type	1:1	**RQ 1**	The types of publications, such as conference or journal.
4	Roles contributed by papers	1:n	**RQ 2**	The stakeholders' roles to which the papers contribute.
5	Development phases contributed by papers	1:1	**RQ 2**	The development phases that papers focus on.
6	Application domain	1:n	**RQ 2**	The domain that papers focus on.
7	Functionalities of MBSE tool-chains	1:n	**RQ 2**	Characteristic purpose of the MBSE tool-chains.

[1] Means that each paper has only one related option, for instance that each paper has one public year. 1:n means that each paper may have more than one option.

2.2. Search Process

A literature review refers to a research approach that reviews, critiques, and synthesizes the representative literature on a topic in an integrated framework [18]. When implementing the search process of the literature review, two phases were proposed:

1. Initial literature review: the aerospace and aircraft industries were first investigated as two leading domains for MBSE practices. Therefore, they were selected for searching papers for the initial literature review, including databases, research groups, journals, conferences, and online websites. Our previous article surveyed papers from 28 journals and 26 conferences of the American Institute of Aeronautics and Astronautics (AIAA) [3]. The results were analyzed to provide some initial topics for the collection and analysis of the MBSE tool-chains literature.
2. Exploratory literature review: After identifying several topics based on the initial literature review, the strategy when selecting literature was to start with a systematic search through Google Scholar using "MBSE", "model-based systems engineering" in combination with "tool-chain" as keywords. Finally, papers were collected and analyzed for this exploratory literature review.

2.3. Inclusion Criteria and Exclusion Criteria

2.3.1. Inclusion Criteria

Inclusion criteria are properties that the prospective topics must have if MBSE tool-chains are to be included in the literature review. Based on the definition of MBSE tool-chains, we defined inclusion criteria to include formalisms of system artifacts and system development, verification based on simulation across domains, traceability management, and design automation based on system engineering workflow [3].

2.3.2. Exclusion Criteria

Exclusion criteria are those properties that disqualify prospective topics from inclusion in the literature review. Papers that were not mentioned as MBSE tool-chains or did not explicitly use the MBSE tool-chain concept were excluded.

As shown in the PRISMA flow specification in Figure 1, through inclusion criteria and exclusion criteria, 260 papers were selected for the final analysis from Google Scholar.

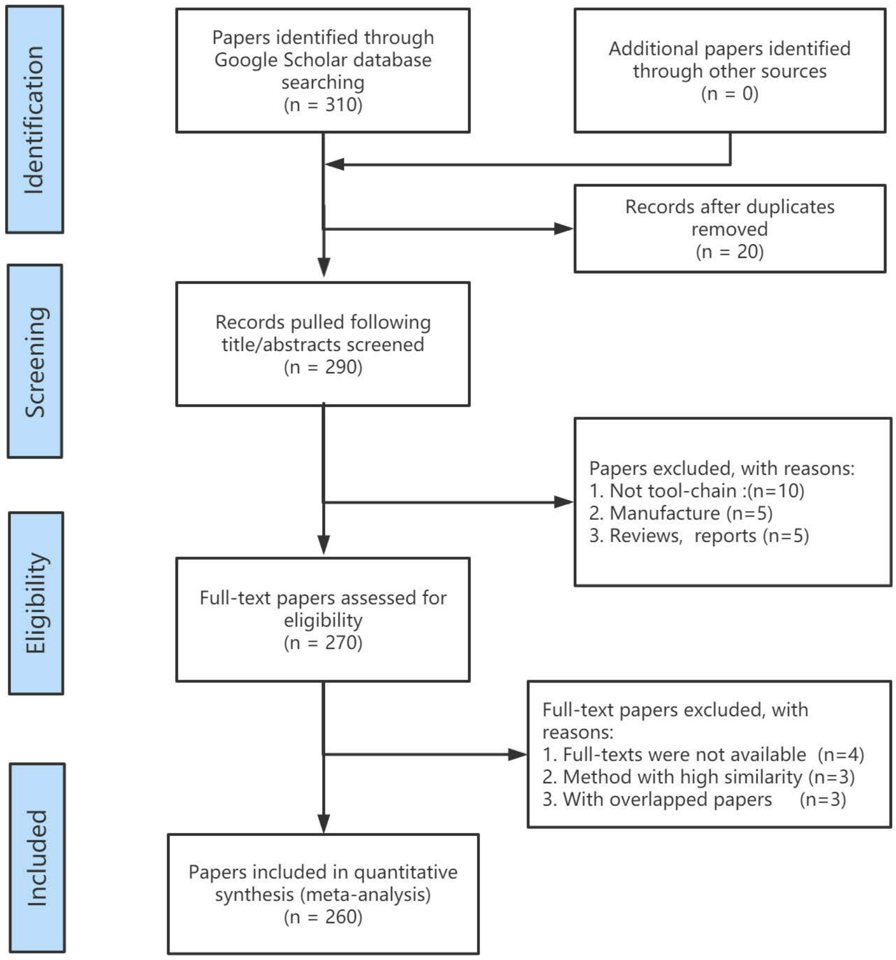

Figure 1. PRISMA flow diagram.

2.4. Quality Assessment

To promote the quality of the findings, we developed quality specifications to understand the significant outcomes of the selected papers. The developed criteria also defined the credibility of each paper as follows:

- Data valuation of an MBSE tool-chain was based on concrete facts and a theoretical perceptive without any vagueness.
- Research was implemented using methods verified by other researchers.
- More comprehensive papers were acquired, if available. We collected papers from 2003 to 2021; approximately 73% of the papers were from 2015 to 2021.
- To ensure the intensity of the research, more than five renowned scientific databases were included.

2.5. Data Collection and Analysis

Data collection and synthesis were performed for the selected papers to obtain the answers to our research questions. In the Findings section (Section 3), the entire process is implemented aiming at identifying the research content in each paper, as summarized in Table 2. In the Discussion section (Section 4), we provide the analysis of the data in order to answer our four research questions.

Table 2. Data collection.

No.	Description	Details
1	Bibliographic information	Title, publishers, publication types, publication year
2	Application areas	Aerospace, aircraft, vehicle, machine, CPS, etc.
3	Stakeholders	System engineer, domain engineer, team leader, etc.
4	Process	R&D project, the whole life-cycle of real product development, pilot project, etc.
5	Life properties	Quality, reliability, safety, flexibility, robustness, durability, scalability, etc.
6	Technical aspect	Modeling purposes, modeling languages, modeling methods, simulation methods, modeling tools, etc.
7	Contributions	Clarifying the contributions of authors with guidelines, models, tools, surveys, etc.

3. Findings

This section summarizes the results of the literature review. All the papers are analyzed using Kitchenham's process to illustrate the results from the four research questions: (1) basic information to answer **RQ 1**; (2) the extent of MBSE tool-chains to answer **RQ 2**; (3) related techniques regarding MBSE tool-chains to answer **RQ 3**; and (4) contributions to answer **RQ 4**.

3.1. Basic Information

RQ 1 concentrates on understanding the main focus of MBSE tool-chains. For this purpose, the literature allows us to understand themes facilitating the continued investigations into the next three research questions. To understand the intensity of basic information, three aspects were considered in order to analyze the scope of the literature reviews: (1) publisher; (2) publication type; and (3) publication year.

In Figure 2A, the publishers of each paper are analyzed, including the American Institute of Aeronautics and Astronautics (AIAA), the Association for Computing Machinery (ACM), Elsevier, HAL, the Institute of Electrical and Electronics Engineers (IEEE), Wiley for the International Council on Systems Engineering (INCOSE) publication, the Society of Automotive Engineers (SAE), Springer, Multidisciplinary Digital Publishing Institute (MDPI), and other publishers (referring to the source on the Web), such as universities that publish technical reports or theses. Most of the papers were from the IEEE (Figure 2A), and approximately 14% of the papers were from Wiley. Moreover, approximately 44% of the papers were published for conferences, and approximately 35% of the papers were from journals (Figure 2B). The publication years, shown in Figure 2C, show the changes in the number of MBSE tool-chain papers from 2003 to 2021.

3.2. Extent of MBSE Tool-Chains

RQ 2 aims to understand the application areas of MBSE tool-chains. Two main aspects were considered: (1) the stakeholders' roles that the papers contribute to, and (2) the development phases that the papers contribute to, which were both estimated by review and analysis. The results in Figure 3 demonstrate that most of the papers were contributed by system engineers and domain engineers, and approximately 58% of the papers contributed to R&D projects.

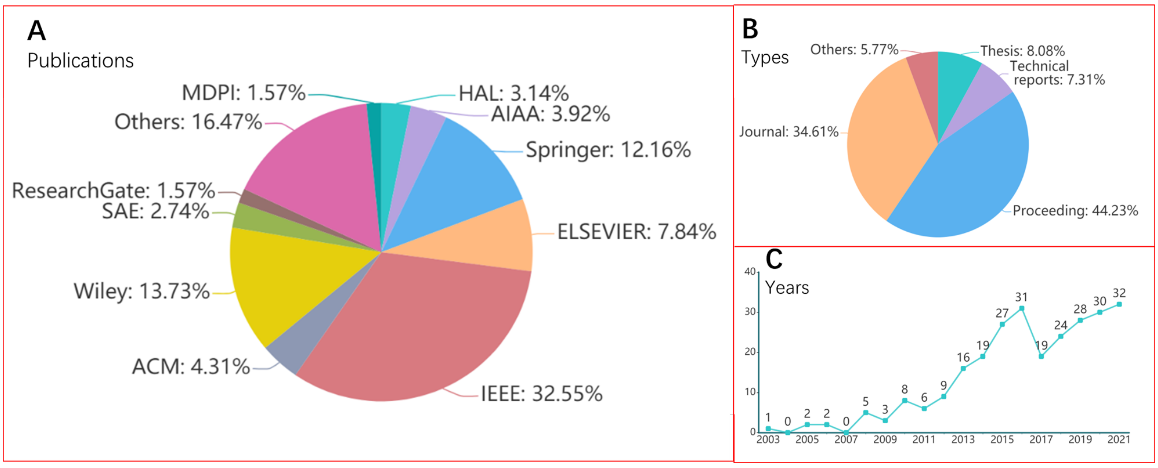

Figure 2. (**A**) Source of papers. (**B**) Publication types. (**C**) Publication years.

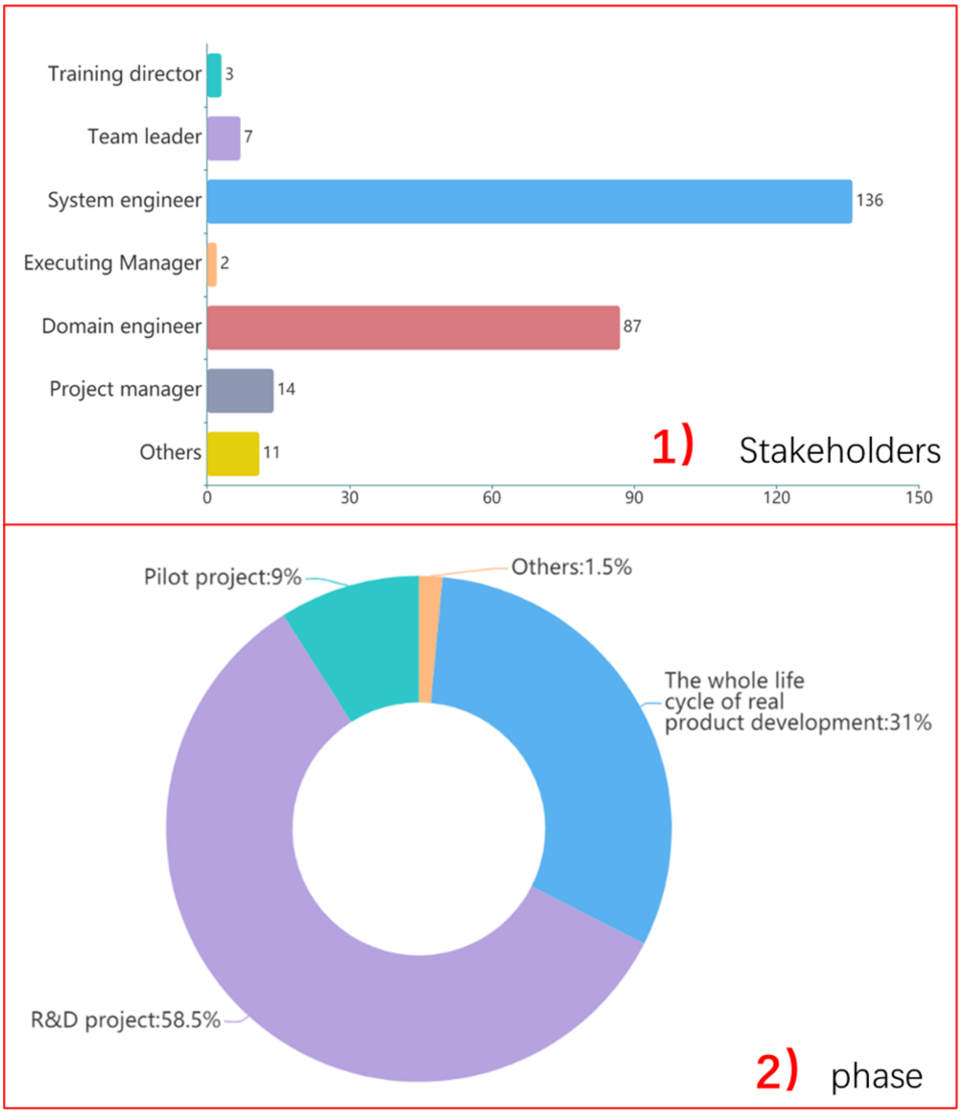

Figure 3. (**1**) The roles that the papers contributed to; (**2**) the development phases that the papers contributed to.

Based on the results shown in Figure 4, aerospace and vehicles were the two domains to which the papers contributed most. Moreover, approximately 47 papers were not covered by the given domains. Figure 5 introduces the functionalities of MBSE tool-chains. In our terminology, system design and development based on systems engineering are defined collectively as “system design”, which is the area that most of the developed tool-chains support.

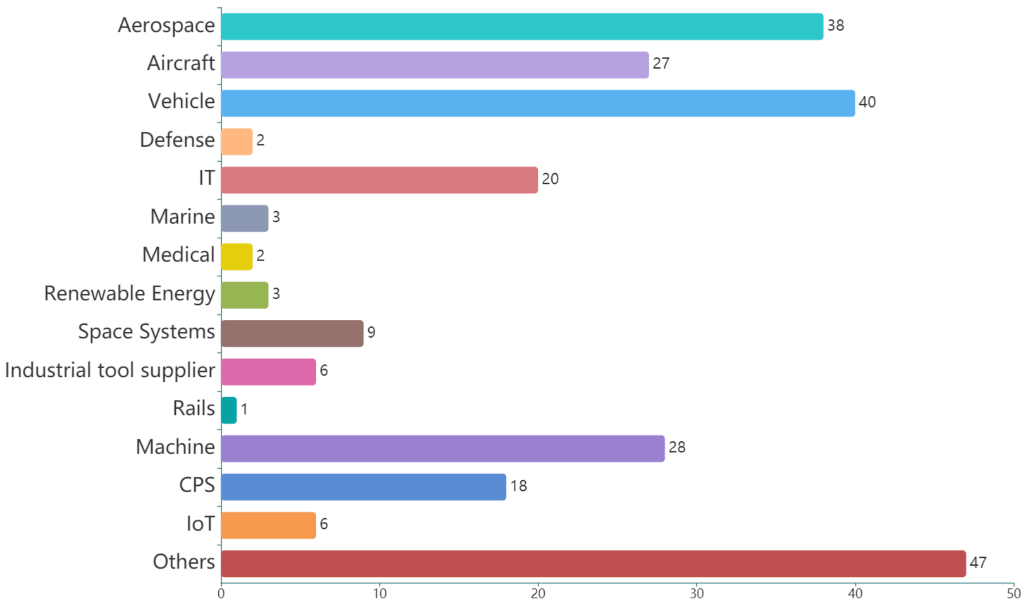

Figure 4. Application areas of MBSE tool-chains.

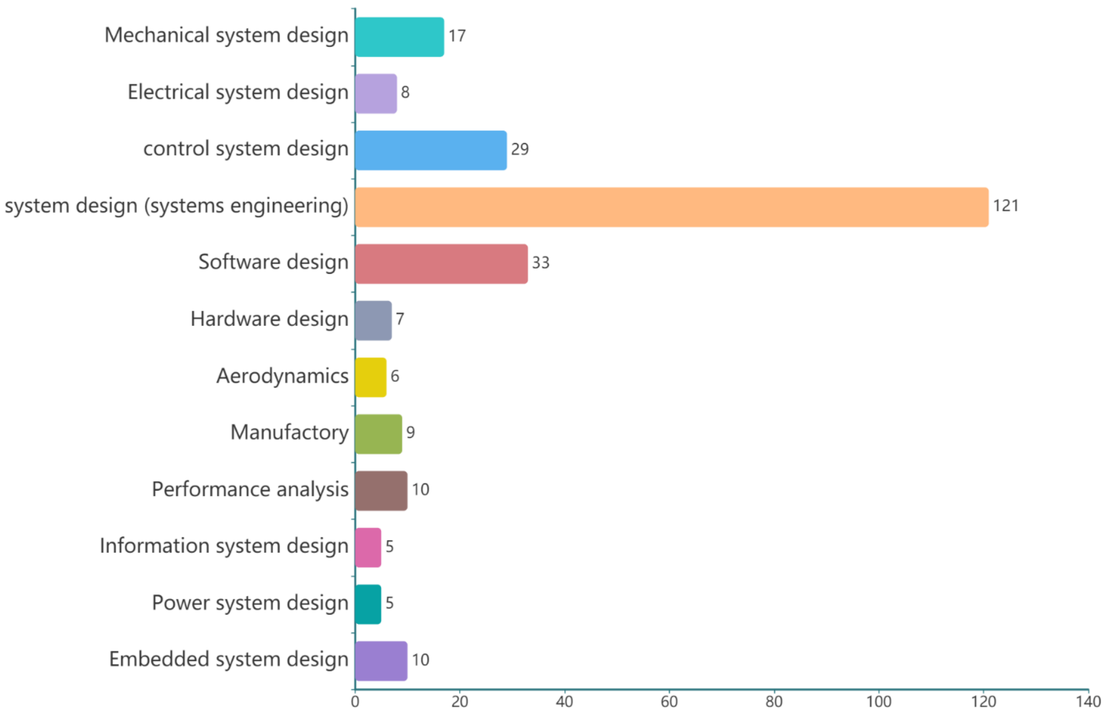

Figure 5. Functionalities of MBSE tool-chains.

3.3. Techniques of MBSE Tool-Chains

RQ 3 aims to identify the techniques of MBSE tool-chains. Life-cycle properties are defined as systems properties that often manifest themselves after a system has been put to initial use during the life-cycle [16]. To investigate which life-cycle properties were captured by the papers, each life-cycle property was searched as a keyword. The results in Figure 6 demonstrate that both *safety* and *reliability* were the most frequently mentioned life-cycle properties.

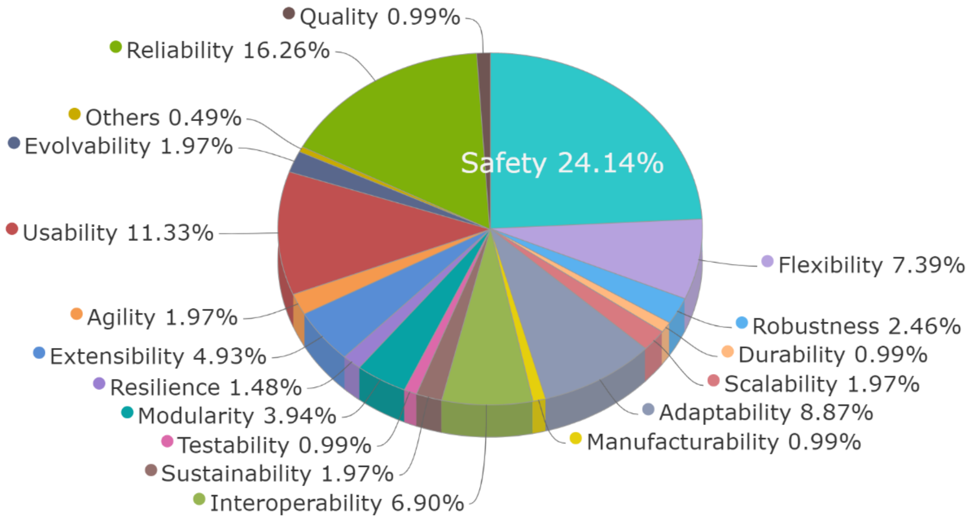

Figure 6. Life-cycle properties related to the studies.

The studies were surveyed to understand the purpose of the models, as perceived by each author. Ultimately, when reviewing such papers, multiple purposes emerge; however, the results demonstrate that *understanding problems at an abstract level* is the most important modeling purpose, as shown in Figure 7.

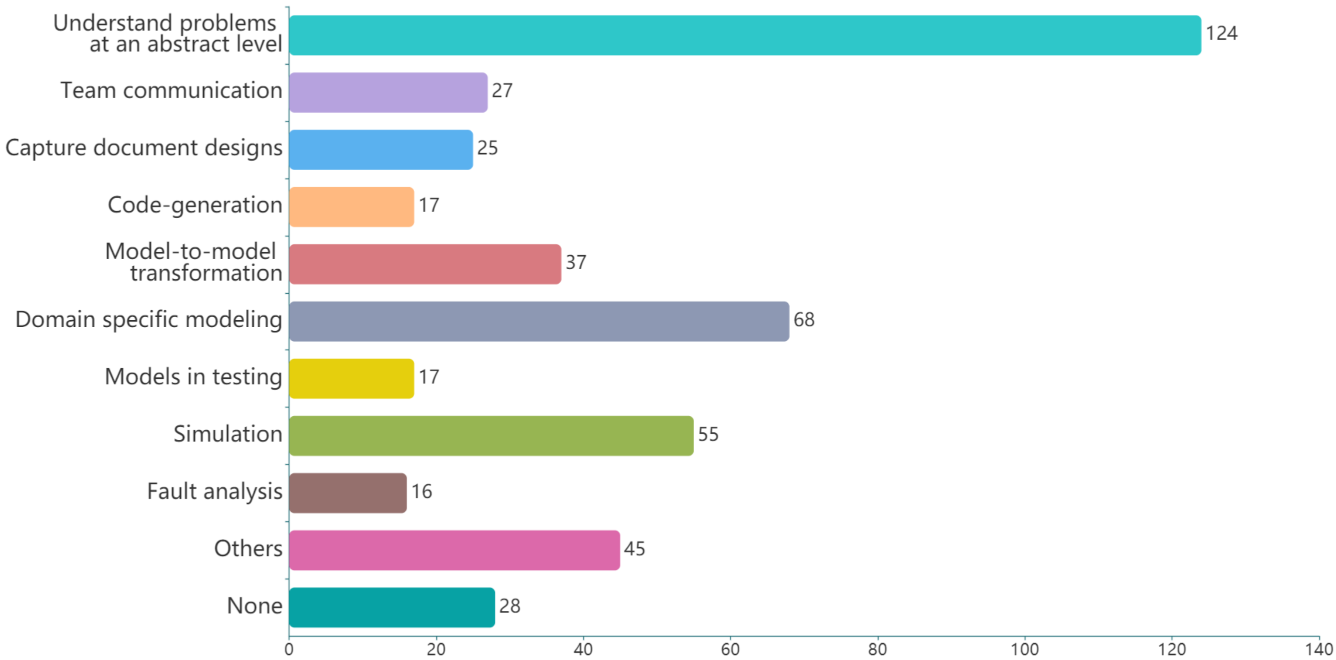

Figure 7. Modeling purposes in the studies.

Each study was analyzed to investigate which perspectives of system artifacts are formalized, including requirements, function, behavior, logic, physical architecture, interface definition, and verification & validation. Moreover, three types of formalisms were considered: (1) models; (2) mathematics; and (3) text. The results shown in Figure 8 demonstrate that system artifacts were formalized as specific models rather than text or mathematics. The requirements were the most frequently mentioned perspective formalized using models.

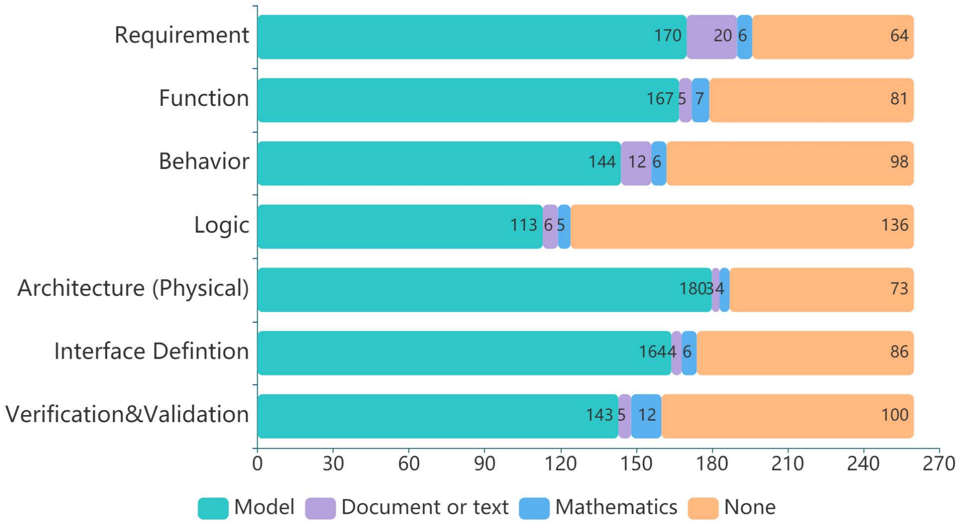

Figure 8. Formalisms mentioned in the studies.

Modeling and programming languages were also investigated in these papers. From the results in Figure 9, SysML was found to be the most widely used language in the tool-chains proposed in the studies. In addition to modeling languages, modeling methods (i.e., how to model) were investigated. Figure 10 shows that meta-modeling was the most widely used approach. Apart from modeling methods, simulation methods were also investigated. Among simulation methods, Figure 11 shows that the numerical simulation method was more general than other methods.

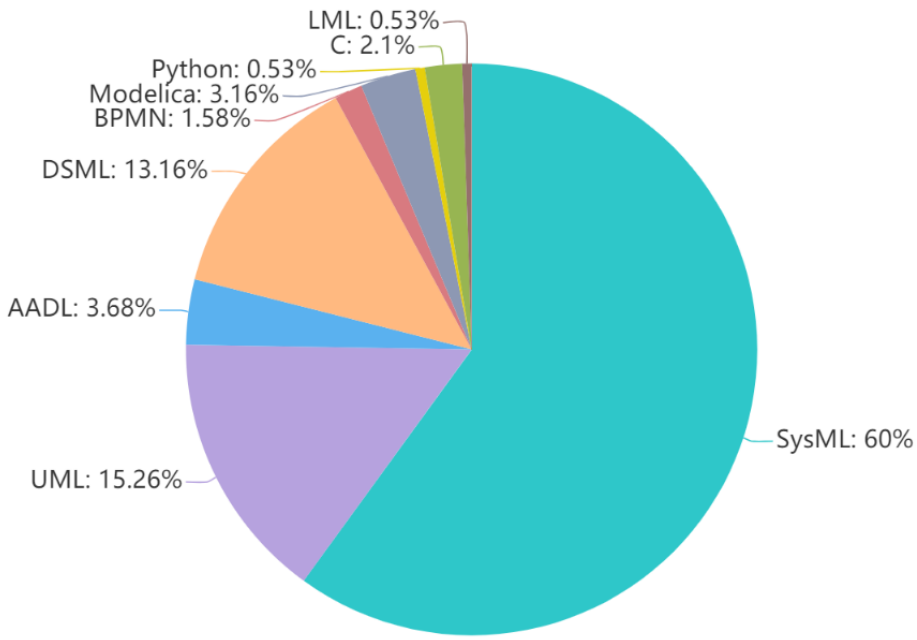

Figure 9. Languages used in the studies.

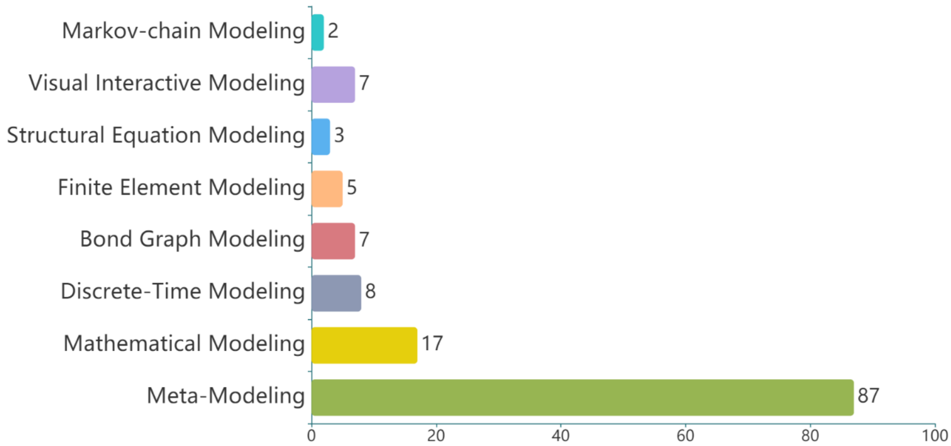

Figure 10. Modeling methods used in the studies.

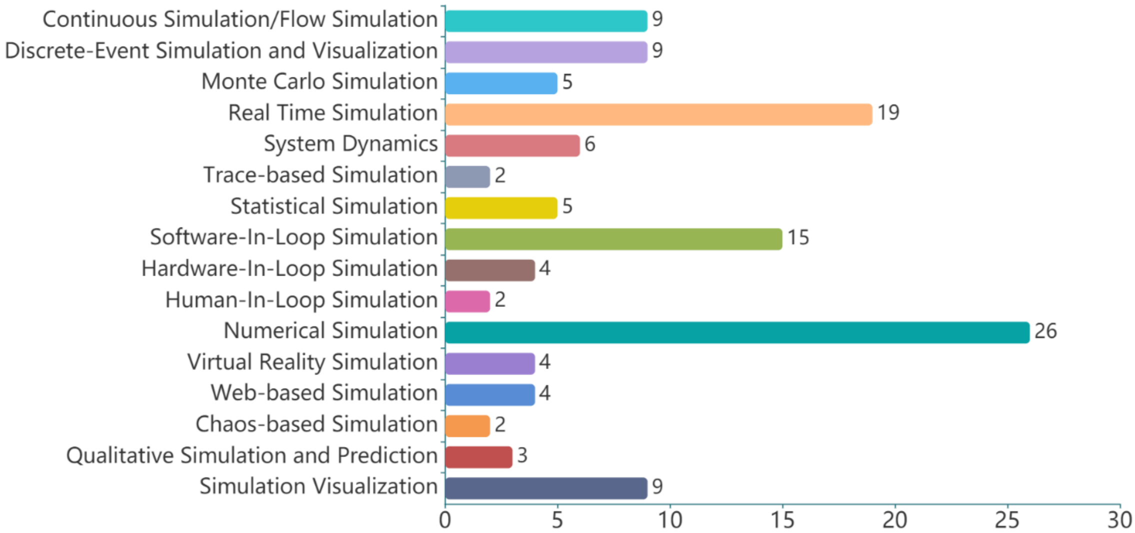

Figure 11. Simulation methods used in the studies.

Modeling tools were investigated to understand how the authors developed and implemented models in MBSE tool-chains. In Figure 12, MATLAB and Simulink are shown as the most frequently used tools. In addition, MBSE tools were classified from the perspective of how to support life-cycle activities. As shown in Figure 13, system modeling was the most frequent life-cycle activity support. In order to understand how to support integrated verification, we investigated the integrated simulations and their solutions in these papers. Figure 14 shows the integrated simulation based on commercial tools, Modelica, and commercial tools for co-simulation; these were found to be the three main solutions used by the tool-chains. We found that more than half of the papers did not support integrated simulation.

Tool integration is an important technical aspect for developing MBSE tool-chains. We investigated five types of tool integration: (1) data integration, referring to data sharing and the relevant complexity management among data in different tools; (2) control integration, providing tool operations to notify and activate other tools; (3) platform integration, providing a virtual operation environment for integrating tools; (4) process integration, referring to integration of process management, development operation, and data configuration; and (5) presentation integration, providing a unified and formal representation of user interfaces [19]. As shown in Figure 15, data integration was the most frequently used for tool integration. Moreover, tool integration solutions were analyzed to identify which techniques were widely used in these studies. From the results, most of the tools were found to be integrated using proprietary exchange rule data exchange based on standards, as shown in Figure 16.

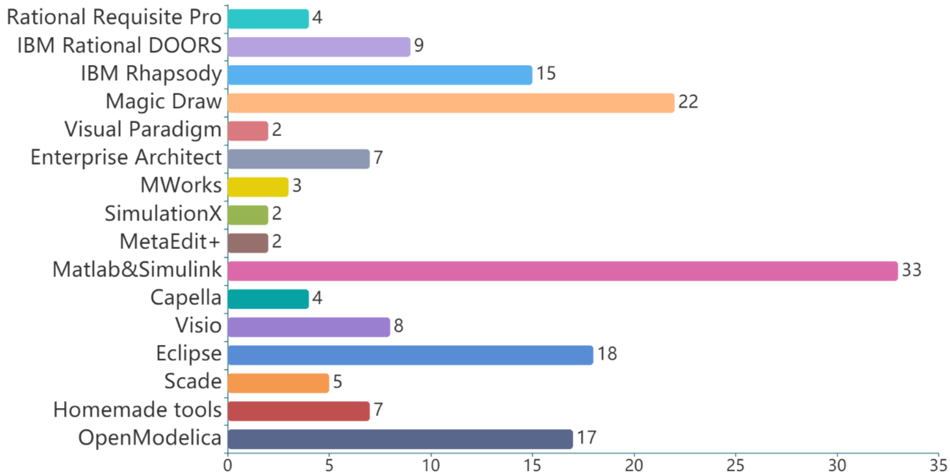

Figure 12. Modeling tools in the studies.

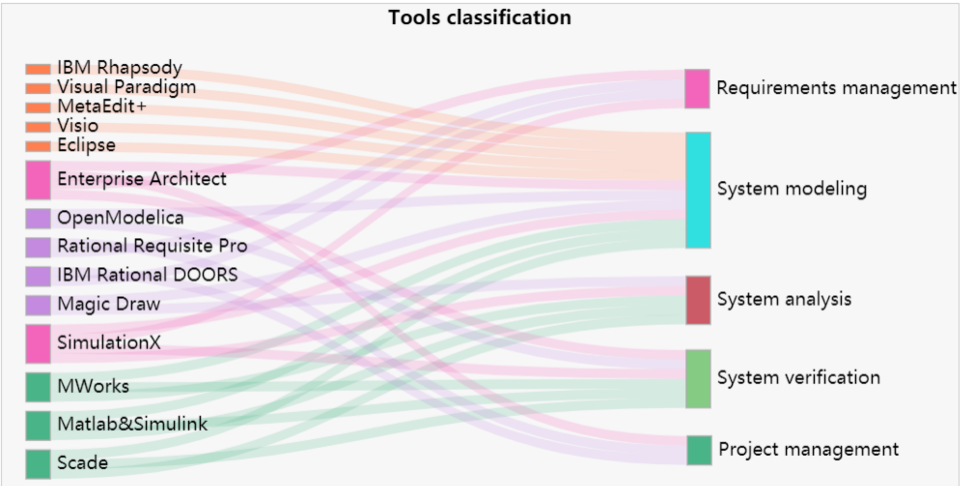

Figure 13. Tools classification.

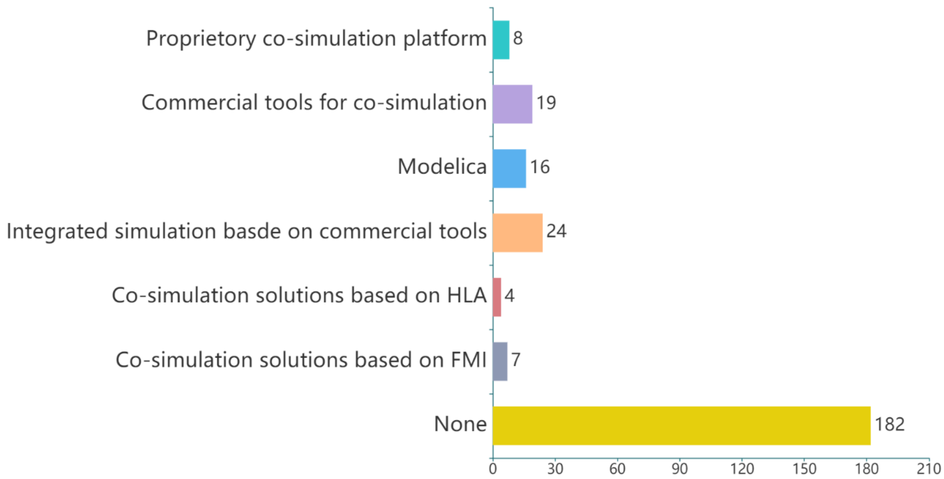

Figure 14. Integrated simulation used in the studies.

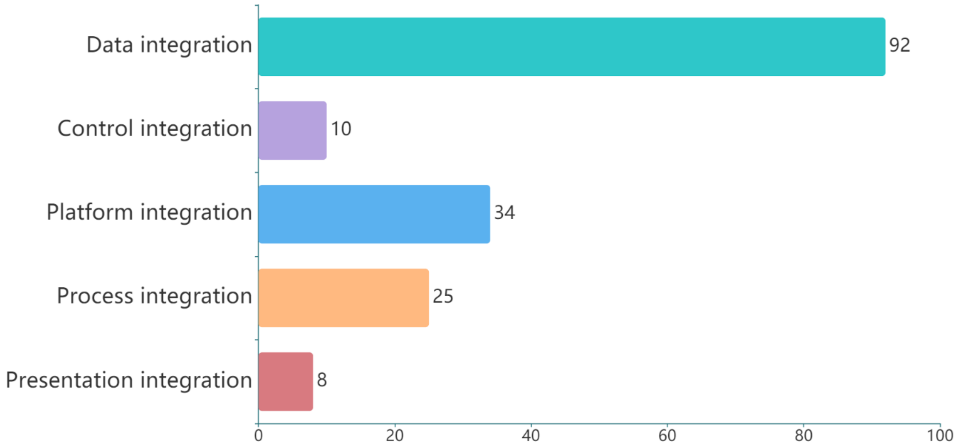

Figure 15. Tool integration in MBSE tool-chains.

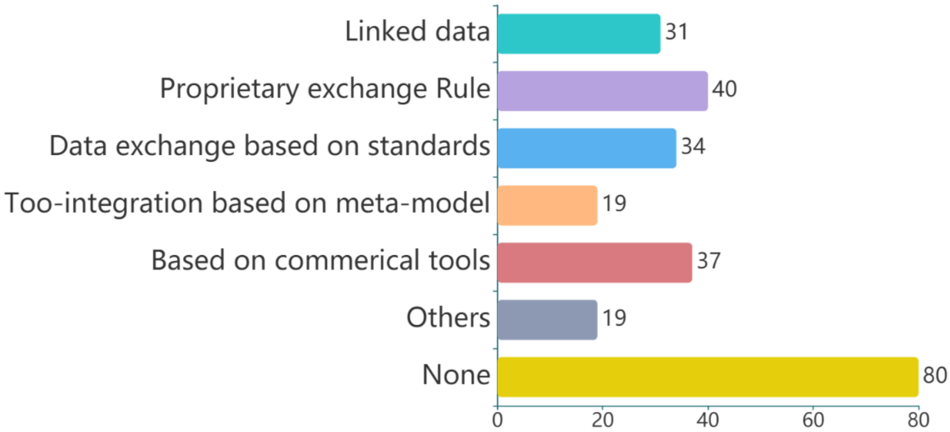

Figure 16. Survey on tool integration solutions.

Tool-chain assessment is always a significant challenge when developing tool-chains [20]. Thus, in this paper, metrics were investigated for assessing MBSE tool-chains and were proposed based on standardized terminologies [21]:

- Integrated capability, means the degree of combining software components, hardware components, or both into an overall system to constitute a tool-chain.
- Interoperability, means the degree of two or more systems or components to exchange information and use the information that has been exchanged in tool-chains.
- Traceability, means the degree to which a relationship can be established between technical recourse, system information, system development processes, and social networks.
- Dependency, means the capability of disposing of the determinant of technical recourse, system information, system development processes, and social networks.
- Reusability, means the capability of reusing technical resources (e.g., models, data, tools) by other modules or work products.
- Design efficiency, means the degree to which the system developers implement their design jobs with minimum consumption of resources by developed tool-chains.
- Scalability, means the degree to which the tool-chain handles a growing amount of engineering work or its potential to be enlarged to accommodate such growth.

As shown in Figure 17, according to the evaluation, integrated capability was most frequently mentioned among these papers.

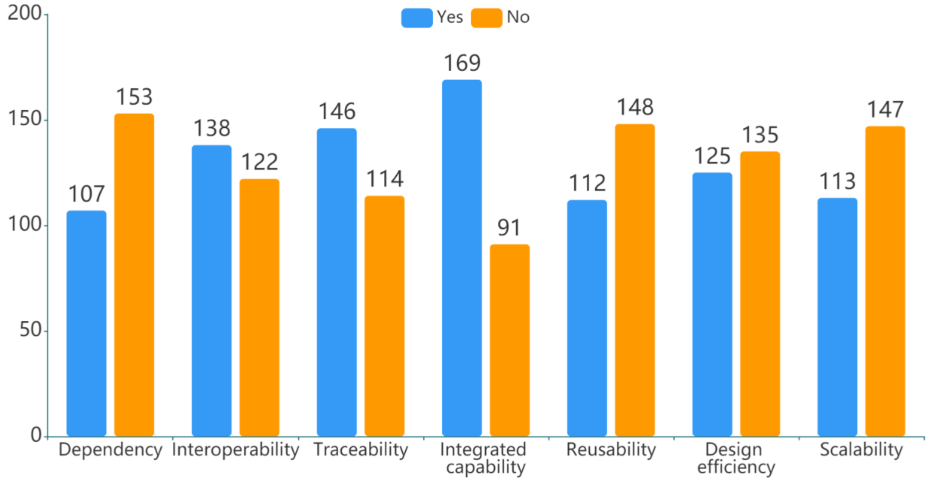

Figure 17. Measurements of MBSE tool-chains in the studies.

3.4. Contributions

To answer **RQ 4**, the contributions of each paper were analyzed based on the defined items: (1) framework/method/technique; (2) guidelines; (3) lessons learned; (4) models; (5) tools; (6) surveys; and (7) tool-chain architectures. The results, as shown in Figure 18, demonstrate that the most frequent contribution of the papers was *framework/method/technique*.

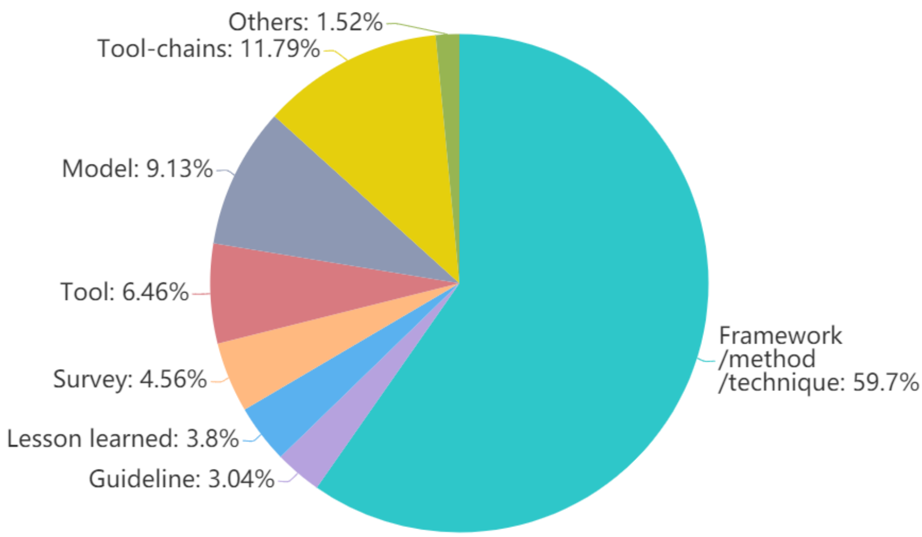

Figure 18. Contributions of studies.

4. Discussion

4.1. Overview of Findings

By analyzing the results of the literature review, we clarify the answers to the four research questions **RQ 1–RQ 4**. The outcome of **RQ 1** shows the intensity of MBSE tool-chains through the publisher, publication type, and publication year. In order to answer **RQ 2**, four perspectives are analyzed, including the background of stakeholders, development phases that MBSE tool-chains contribute to, application areas, and tool-chain functionalities to clarify how the tool-chains support system engineering. As for **RQ 3**, 13 questions are used to analyze MBSE technologies for tool-chain development. With regard to the last question, each paper is evaluated to capture the contributions for MBSE tool-chains.

4.2. Detailed Analysis from Findings

4.2.1. Basic Analysis

From *Basic information* in Section 3.1, we found that more than half of the published papers in the total papers about MBSE tool-chains were published by the publishers from IEEE, INCOSE publication in Wiley, and Springer. Since 2003, the number of papers has increased every year, and the scale of tool-chains research has continued to expand. Especially in the field of aerospace and space system, the research enthusiasm for MBSE tool-chains is higher. The number of MBSE tool-chains papers has grown steadily over the past 5 years, and the total number has reached the sum of the number of papers in the previous 14 years.

4.2.2. Extent Analysis of MBSE Tool-Chains

Regarding to the application areas, as shown in Figure 2, system engineers and domain engineers have greater opportunities to engage in tool-chains. Moreover, Figure 3 shows that researchers pay more attention to the stage of R&D in the product life-cycle. From the perspective of functionalities, we summarized 12 functionalities developed in MBSE tool-chains based on application scenarios in Figure 5. Among them, system design was embodied in 121 papers, which means that the research on tool-chains focused on the overall design of the system and that there was little research on the design of professional fields such as mechanics, electronics, and control.

4.2.3. Techniques Analysis of MBSE Tool-Chains

Life-Cycle Properties

To fully identify the techniques for MBSE tool-chain development, a series of the most frequently mentioned life-cycle properties are summarized in Figure 6. Among the 16 investigated properties, *safety* and *reliability* were the most frequently mentioned. These two properties were specifically supported by an OMG group, which consisted of both industry and academia and defined a new standardized UML profile to support safety and reliability analysis [22]:

- Safety refers to the application of safety engineering in MBSE tool-chains, identifying potential dangers in the system through risk model definition and simulation analysis for taking effective control measures to minimize the dangers. For example, Timo and Seppo extended the scope of UML AP to also cover the development and design of safety systems. The work was targeted to the requirement concepts of the profile but also to documenting the results of risk and hazard analysis, which aided in the discovery of the information when the developers needed it [23]. Lena et al. presented a functional safety analysis approach in the process of system modeling, and the analysis of the resulting model was performed through a stochastic Bayesian model. This approach strove to both bypass the necessity for costly hardware testing and integrate the functional safety analysis into an intuitive component development process [24].
- Reliability is defined as the ability of a functional unit to perform a required function under given conditions for a given time interval [25]. For example, Zhao Huang et al. explored the integration of failure mode and effects analysis (FMEA) and MBSE by generating FMEA models from MBSE models to support system reliability design [26]. Furthermore, David et al. generated FMEA models from structural and behavioral diagrams in SysML models and used the AltaRica language to compute the reliability indicators [27]. Moreover, the reliability of the code generator has been identified as a key concern in automatic code generation. In order to investigate the model-based design approach for control system software development, Matias Soini et al. constructed the traceability between Simulink models and the generated program code by developing reliable code generators [28]. The result implies that *safety* and *reliability* are two important life-cycle properties in aerospace and vehicles, because most of the papers we collected were from these two domains, as shown in Figure 4.

Technical Aspects

As shown in Figures 7–16, we analyzed the technical aspects of the MBSE tool-chains to support system development, including modeling purposes, modeling languages, modeling methods, etc. Figure 7 demonstrates that *understanding problems at an abstract level* is the most common purpose of tool-chains. As shown in Figure 8, the formalisms of requirement, function, behavior, logic, physical architecture, interface definition, and verification & validation are used to understand the system problems at the abstract level. As the most frequently researched aspect, the requirement not only needs to tell system engineers what the system needs to do, but also needs to improve the traceability of design problems through verification [29,30]. Figure 9 shows that SysML [31] is widely used in terms of modeling languages. As for modeling methods, due to the ability of abstract system architectures, meta-modeling is widely used in MBSE, as shown in Figure 10. Simulation is an important part of system design; therefore, simulation methods are widely used in MBSE tool-chains. Only about half of all the papers included simulation in their tool-chain, as shown in Figure 11. From the perspective of modeling tools in Figure 12, MATLAB and Simulink were the most widely used tools in tool-chains. As a visual simulation tool, Simulink provides a graphical editor, a customizable module library, and a solver which can perform dynamic system modeling and simulation. As seen in Figure 14, a limited number of articles mentioned integrated verification; *Modelica*, *the integrated simulation based on commercial tools*, and *commercial tools for co-simulation* were the main approaches to support integrated simulation.

During tool-chain development, tool integration plays an important role in sharing information. Figure 15 shows that data integration was the most frequently used for tool integration, which refers to automated data transformation to manage data consistency in order to improve the execution efficiency and accuracy of tool-chains. For example, failure models were generated through unified data models to achieve analysis automatically [32]. Lankhorst et al. outlined an integrated language which serves as underlying data for other existing modeling languages. Furthermore, they presented a workbench for enterprise architecture based on the integrated language that supports both the integration of models in existing modeling languages and the integration of existing modeling tools [33]. In addition to tool integration, tool integration techniques were also investigated. As shown in Figure 16, we found that the total number of *proprietary exchange rules* was slightly higher than the total amount of *data exchange based on standards*. However, from 2018 to 2021, 20 papers referred to tool integration based on *data exchange based on standards*, while only 5 papers were related to the *proprietary exchange rule*. Thus, the result indicates that more standards were used when researchers developed tool-chains over the last four years. For example, Hu transformed simulation data, models, and tool operations to Open Services for Lifecycle Collaboration (OSLC) services based on unified standard, which are accessed through identified URLs by other tools [34].

4.2.4. Assessment of MBSE Tool-Chains

Figure 17 evaluates the metrics of the MBSE tool-chains. The results show that the metrics of *integrated capability*, *interoperability*, and *traceability* are mentioned in more than 130 papers. The integration capability is the tool-chain's function for managing the tools' interconnections, interfaces, relationships and dependencies, and the processes of requirement management, architecture design, interface definition, and verification & validation. Integration capability increases the automation of system development; for example, mathematical simulation tools and system modeling tools were integrated to support automatic verification in control systems [35]. Moreover, integration capability improves design efficiency, as, for instance, in a model-driven embedded system design flow integrating the SysML and SystemC-UML profiles to realize hardware–software co-design [36]. In addition, integration capability is also important for system analysis; for example, a novel approach was presented to address functional safety requirement verification, in which the authors integrated functional tests as full-fledged components into a model-based architecture using OpenModelica [24].

Interoperability emphasizes the model and information interaction in an MBSE tool-chain, which is a key impact factor when describing data in a unified way. For example, Shani and Franke et al. proposed a unified ontology approach for transforming product-related data and aggregated the ontologies of different data sources into a holistic product service system [37]. Interoperability is also an important attribute for developing middleware. For example, when connecting the data of system design to operational robotics, good interoperability facilitates model-driven development in robotics to promote the development efficiency based on a tool-chain [38].

Traceability refers to the ability through which an element is associated with other related elements, such as verification models that are traced to the related requirements in MBSE tool-chains. Traceability can be used to implement project and process management and to support automation; for instance, Pavalkis et al. provided the integrated platform for the life-cycle process management of MBSE and PLM data with traceability management [39]. Moreover, traceability enables the management of requirements; for example, a recursive hierarchy was used in a requirements management system to make the system traceable from decision-making to its success criteria [40].

4.2.5. Contributions

To identify the current research trends regarding MBSE tool-chains, we defined seven types of contributions to distinguish the selected papers.

- *Framework/method/technique* indicates that the contribution of the article is about the framework of the tool-chain, the methods used by the tool-chain, and the technology involved in the tool-chains.
- *Guideline* indicates that the contribution of the papers is to provide guidance for building the MBSE tool-chains.
- *Lesson learned* emphasizes the experience summary of the tool-chain construction process.
- *Model* indicates that the contribution is about the modeling.
- *Tool* indicates that the contribution of the paper is about the use of the tool itself.
- *Survey* indicates that the contribution of the paper is about a survey.
- *Tool-chain architecture* indicates that the contribution of the paper is about tool-chain architecture.

From the results, we found that more than half of the papers focused on the *framework/method/ technique* of MBSE tool-chains, which indicates that the research on MBSE tool-chain is at the stage of concept design. The relatively fewer contributions to tools and models illustrate that tool-chains are not widely used in the industry. This is consistent with the results in Figure 3, which show that most of the papers were proposed in R&D projects. In addition, a framework development for tool integration is very important to construct a tool-chain for different use cases, because a good framework provides more flexibility for tool integration from the perspective of **RQ 3**, such as a model-driven framework [41], a CPS open standardized platform [42], and a framework to support collaborative virtual prototyping of a system-of-systems [43]. As a lesson learned from these studies, a good integration framework is required for software designers before building MBSE tool-chains.

4.3. Answers to the Questions

To combine Sections 3 and 4, we use Table 3 to answer the four questions mentioned in Section 2.

4.4. Limitations

Although we completely followed Kitchenham's procedure and strictly observed the developed review protocol, there are still certain limitations to be addressed:

- We used the appropriate keywords "MBSE" or "model-based systems engineering" or "tool-chain", and thoroughly scanned the search results. However, we missed some

papers which, with keywords such as "model driven", "PDM", and "Integration", are also related to MBSE tool-chains.
- The paper collection and data analysis was completed in October 2020, so that any papers after that date did not appear in our research, which may have resulted in the data results not being up-to-date.
- In order to answer the questions, we needed to analyze the paper content, because it was not obvious how to capture some of them. For instance, a large number of papers belong to two application areas of tool-chains, but we only considered one. This may have caused data deviations, but may not have affected the overall analysis of the data.

Table 3. Answer to the questions.

RQ	Content	Result
RQ1	What is the intensity of the research on MBSE tool-chains?	1. Research articles are typically published in IEEE, Wiley, and Springer, as shown in Figure 2. 2. Since 2012, researchers have shown increasing interest in tool-chains, as shown in Figure 2.
RQ2	What are the most frequently investigated research areas related to MBSE tool-chains?	1. System engineers contribute the most to tool-chain development. Most of the papers focus on R&D projects, as shown in Figure 3. 2. The tool-chains are most widely used in the domains of aerospace and vehicles, as shown in Figure 4. 3. System design is the field which is the most supported via developed tool-chains, as shown in Figure 5.
RQ3	What are the most frequently applied research types and research methods? (Mainly to identify the technology of the MBSE tool-chains.)	1. The life-cycle properties of *safety* and *reliability* are of interest to researchers, as shown in Figure 6. 2. *Understanding problems at an abstract level* is the most important modeling purpose, as shown in Figure 7. 3. The model formalism is mostly mentioned, among which the requirement model is the most frequently mentioned perspective, as shown in Figure 8. 4. SysML is the most widely used modeling language in the context of tool-chains, as shown in Figure 9. 5. Meta-modeling is the most widely used modeling method, as shown in Figure 10. 6. Numerical simulation is the most widely used simulation method, as shown in Figure 11. 7. MATLAB and Simulink are the most frequently used tools, as shown in Figure 12. 8. *Integrated simulation based on commercial tools*, *Modelica*, and *commercial tools for co-simulation* are the three main solutions employed by the tool-chains for supporting integrated verification, as shown in Figure 14. 9. Data integration is the most frequent process for tool integration, as shown in Figure 15. Data exchange based on standards has been used for integration in most tools in recent years, as shown in Figure 16. 10. Integrated capability is most frequently considered tool-chain assessment metric, as shown in Figure 17.
RQ4	What contributions are provided by research related to MBSE tool-chains?	1. The item of the *framework/method/technique* is the most frequent contribution, as shown in Figure 18.

5. Conclusions

Model-based systems engineering is a model-based approach to capture, analyze, share, integrate, and manage not only requirements, but also development processes, products, verification & validation, and production across the whole life-cycle. The MBSE tool-chain is a solution aiming to support integrated co-design and data management for the entire life-cycle. In this study, we analyzed the current state of the art of MBSE tool-chains from intensity analysis, application areas analysis, technique analysis, and research contributions.

The literature review has provided a wealth of insights into the current state of MBSE tool-chains. From the results, we put forward the following suggestions for reference in the process of tool-chain research in the future:

- The related technologies of SysML are the most mature. In the field of system design, tools such as requirements management, system modeling, and simulation can be integrated based on the SysML language specification.
- Data exchange based on standards has become an important solution for tool integration. In the future, we should pay attention to the unified formulation of standards.
- We suggest that, with the steady development of tool-chain research, it is necessary not only to concentrate on research in the R&D phase, but also to focus on the whole life-cycle of real product development.
- When assessing tool-chains, we expect integrated capability, interoperability, and traceability to become more effective indicators.

This literature review is expected to be used as a stepping stone in designing a strategy for MBSE tool-chain development and as guidance for future actions.

Author Contributions: Conceptualization, J.M.; methodology, J.M. and J.L.; analysis, J.M.; writing—original draft preparation, J.M. and J.L.; writing—review and editing, G.W., H.V., D.K., and Y.Y.; supervision, J.L. and Y.Y.; funding acquisition, G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This study is supported by the CN National Key Research and Development Plan (Grant No. 2020YFB1708100) and the CN pre-study common technology (50923010101).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Haskins, C.; Forsberg, K.; Krueger, M.; Walden, D.; Hamelin, D. Systems engineering handbook. *INCOSE* **2006**, *9*, 185.
2. Wiki. Hello. 2005. Available online: https://en.wikipedia.org/wiki/Toolchain (accessed on 2 November 2021).
3. Lu, J.; Chen, D.J.; Gürdür, D.; Törngren, M. An Investigation of Functionalities of Future Tool-chain for Aerospace Industry. In *INCOSE International Symposium*; Wiley Online Library: Hoboken, NJ, USA, 2017; Volume 27, pp. 1408–1422.
4. Binder, C.; Fischinger, M.; Neureiter, C.; Lastro, G.; Polanec, K.; Gross, J.A. Towards a Tool-Based Approach for Dynamically Generating Co-Simulation Scenarios based on complex Smart Grid System Architectures. In Proceedings of the 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2–4 June 2020; pp. 199–204.
5. König, C.; Mengist, A.; Gamble, C.; Höll, J.; Lausdahl, K.; Bokhove, T.; Brosse, E.; Möller, O.; Pop, A. Traceability in the Model-Based Design of Cyber-Physical Systems. In Proceedings of the American Modelica Conference 2020, Boulder, CO, USA, 23–25 March 2020.
6. Lu, J.; Chen, D.; Wang, G.; Kiritsis, D.; Törngren, M. Model-based systems engineering tool-chain for automated parameter value selection. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *52*, 2333–2347. [CrossRef]
7. Estefan, J.A. Survey of Model-Based Systems Engineering (MBSE) Methodologies. *Environment* **2008**, *25*, 1–12.
8. Motamedian, B. MBSE applicability Analysis. *Int. J. Sci. Eng. Res.* **2013**, *4*, 7.
9. Lu, J.; Wen, Y.; Liu, Q.; Gürdür, D.; Törngren, M. MBSE Applicability Analysis in Chinese Industry. *INCOSE Int. Symp.* **2018**, *28*, 1037–1051. [CrossRef]
10. Huldt, T.; Stenius, I. State-of-practice survey of model-based systems engineering. *Syst. Eng.* **2019**, *22*, 134–145. [CrossRef]
11. Rashid, M.; Anwar, M.W.; Khan, A.M. Toward the tools selection in model based system engineering for embedded systems—A systematic literature review. *J. Syst. Softw.* **2015**, *106*, 150–163. [CrossRef]
12. Hutchinson, J.; Rouncefield, M.; Whittle, J. Model-driven engineering practices in industry. In Proceeding of the 33rd International Conference on Software Engineering, Honolulu, HI, USA, 21–28 May 2011; ACM Press: New York, NY, USA, 2011; p. 633. [CrossRef]
13. Gomes, C.; Thule, C.; Broman, D.; Larsen, P.G.; Vangheluwe, H. Co-simulation: State of the art. *arXiv* **2017**, arXiv:1702.00686.
14. Asplund, F.; Biehl, M.; El-Khoury, J.; Törngren, M. Tool integration beyond Wasserman. In Proceedings of the International Conference on Advanced Information Systems Engineering, London, UK, 20–24 June 2011; Springer: Berlin/Heidelberg, Germany, 2011; pp. 270–281.
15. Kitchenham, B.; Pretorius, R.; Budgen, D.; Pearl Brereton, O.; Turner, M.; Niazi, M.; Linkman, S. Systematic literature reviews in software engineering—A tertiary study. *Inf. Softw. Technol.* **2010**, *52*, 792–805. [CrossRef]

16. De Weck, O.L.; Ross, A.M.; Rhodes, D.H. Investigating Relationships and Semantic Sets amongst System Lifecycle Properties (Ilities). In Proceedings of the Third International Engineering Systems Symposium CESUN, Delft, The Netherlands, 18–20 June 2012; pp. 18–20.
17. Lu, J. *Research Survey on Model-Based Systems Engineering Tool-Chain*; Technical Report; KTH: Stockholm, Sweden, 2019.
18. Torraco, R.J. Writing Integrative Literature Reviews: Guidelines and Examples. *Hum. Resour. Dev. Rev.* **2005**, *4*, 356–367. [CrossRef]
19. Lu, J.; Gürdür, D.; Chen, D.J.; Wang, J.; Törngren, M. Empirical-evolution of frameworks supporting co-simulation tool-chain development. In Proceedings of the World Conference on Information Systems and Technologies, Naples, Italy, 27–29 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 813–828.
20. Lu, J.; Wang, G.; Tao, X.; Wang, J.; Törngren, M. A domain-specific modeling approach supporting tool-chain development with Bayesian network models. *Integr. Comput.-Aided Eng.* **2020**, *27*, 153–171. [CrossRef]
21. Geraci, A.; Katki, F.; McMonegal, L.; Meyer, B.; Porteous, H. *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries*; Technical Report; IEEE Press: Piscataway, NJ, USA, 1991; [CrossRef]
22. Biggs, G.; Juknevicius, T.; Armonas, A.; Post, K. Integrating Safety and Reliability Analysis into MBSE: Overview of the new proposed OMG standard. *INCOSE Int. Symp.* **2018**, *28*, 1322–1336. [CrossRef]
23. Vepsäläinen, T.; Kuikka, S. Simulation-based development of safety related interlocks. In *Simulation and Modeling Methodologies, Technologies and Applications*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 165–182.
24. Rogovchenko-Buffoni, L.; Tundis, A.; Hossain, M.Z.; Nyberg, M.; Fritzson, P. An integrated toolchain for model based functional safety analysis. *J. Comput. Sci.* **2014**, *5*, 408–414. [CrossRef]
25. Joglar, F. Reliability, availability, and maintainability. In *SFPE Handbook of Fire Protection Engineering*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 2875–2940.
26. Huang, Z.; Swalgen, S.; Davidz, H.; Murray, J. MBSE-assisted FMEA approach—Challenges and opportunities. In Proceedings of the 2017 Annual Reliability and Maintainability Symposium (RAMS), Orlando, FL, USA, 23–26 January 2017; pp. 1–8.
27. David, P.; Idasiak, V.; Kratz, F. Reliability study of complex physical systems using SysML. *Reliab. Eng. Syst. Saf.* **2010**, *95*, 431–450. [CrossRef]
28. Soini, M. Modeling and Simulation Practices in Control System Software Development. Master's Thesis, Tampere University of Technology, Tampere, Finland, 2014.
29. Chen, R.; Chen, C.H.; Liu, Y.; Ye, X. Ontology-based requirement verification for complex systems. *Adv. Eng. Inform.* **2020**, *46*, 101148. [CrossRef]
30. Willich, F.; Wolff, C.; Sutorma, A.; Jahn, U.; Stampa, M. Model-based Systems Engineering of an Active, Oleo-Pneumatic Damper for a CS-23 General Aviation Aircraft Landing Gear. In Proceedings of the 2021 IEEE European Technology and Engineering Management Summit (E-TEMS), Dortmund, Germany, 18–20 March 2021; pp. 166–172.
31. Bilic, D. Managing Variability in SysML Models of Automotive Systems. Ph.D. Thesis, Mälardalen University, Westeros, Sweden, 2020.
32. Myron, H.; Elisabeth, N.; Aaron, C.; Pinchak, J. Automated generation of failure modes and effects analysis for a medical device. In Proceedings of the IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), Gaithersburg, MD, USA, 2–5 November 2015; pp. 29–32.
33. Lankhorst, M.M. Enterprise architecture modelling—The issue of integration. *Adv. Eng. Informatics* **2004**, *18*, 205–216. [CrossRef]
34. Hu, Z.; Lu, J.; Chen, J.; Zheng, X.; Kyritsis, D.; Zhang, H. A complexity analysis approach for model-based system engineering. In Proceedings of the 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, 2–4 June 2020; pp. 000501–000506.
35. Vepsäläinen, T.; Kuikka, S. Integrating model-in-the-loop simulations to model-driven development in industrial control. *Simulation* **2014**, *90*, 1295–1311. [CrossRef]
36. Riccobene, E.; Scandurra, P. Integrating the SysML and the SystemC-UML profiles in a model-driven embedded system design flow. *Des. Autom. Embed. Syst.* **2012**, *16*, 53–91. [CrossRef]
37. Shani, U.; Franke, M.; Hribernik, K.A.; Thoben, K.D. Ontology mediation to rule them all: Managing the plurality in product service systems. In Proceedings of the 11th Annual IEEE International Systems Conference, Montreal, QC, Canada, 24–27 April 2017; pp. 1–7. [CrossRef]
38. Adam, K.; Holldobler, K.; Rumpe, B.; Wortmann, A. Engineering Robotics Software Architectures with Exchangeable Model Transformations. In Proceedings of the 2017 First IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, 10–12 April 2017; pp. 172–179. [CrossRef]
39. Pavalkis, S. Towards Industrial Integration of MBSE into PLM for Mission-Critical Systems. *INCOSE Int. Symp.* **2016**, *26*, 2462–2477. [CrossRef]
40. Jackson, M.; Wilkerson, M. MBSE-driven visualization of requirements allocation and traceability. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–17. [CrossRef]
41. Lu, J.; Chen, D.; Jinzhi, L.; Lu, J.; Chen, D.; Törngren, M.; Loiret, F.; Martin, T.; Lu, J.; Chen, D.; et al. A Model-driven and Tool-integration Framework for Whole Vehicle Co-simulation Environments. In Proceedings of the 8th European Congress on Embedded Real Time Software and Systems, Toulouse, France, 27–29 January 2016.

42. Baras, J.S.; Austin, M.A.; Baras, S. *Development of a Framework for CPS Open Standards and Platforms*; Technical Report; The Institute for Systems Research, University of Maryland: College Park, MD, USA, 2014.
43. Schneider, J.P.; Champeau, J.; Lagadec, L.; Senn, E. Role Framework to Support Collaborative Virtual Prototyping of System of Systems. In Proceedings of the 2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, Larnaca, Cyprus, 15–17 June 2015; pp. 144–149. [CrossRef]