

## Utilization of system models in model-based systems engineering: definition, classes and research directions based on a systematic literature review

Fabian Wilking , Dennis Horber , Stefan Goetz  and Sandro Wartzack 

*Engineering Design, Friedrich-Alexander-Universität Erlangen-Nürnberg, Martensstraße 9, 91058 Erlangen, Germany*

### Abstract

The use of system models within model-based systems engineering (MBSE) is essential for improved communication or system documentation. Previous publications have investigated further reuse of these system models, for example, transforming them directly into discipline-specific models for reuse. The authors refer to this as the term “Utilization” of system models. It aims the compensation of modelling efforts and a further integration of linked models within MBSE. Motivated by a lack of common understanding of this term, a systematic literature review of the state of the art is presented. With this systematic overview, a definition and classification system for different use cases and system life cycle stages are created. These are key results to support engineers and researchers in adopting existing or discovering new utilization approaches. This supports the mission of advanced systems engineering and aims the identification of new research directions coming along with SysML v2 and the advanced systems engineering methods.

**Keywords:** Model-based systems engineering, MBSE, Systems engineering, System models, SysML

Received 26 May 2023  
Revised 08 January 2024  
Accepted 26 January 2024

#### Corresponding author

F. Wilking  
wilking@mfk.fau.de

© The Author(s), 2024. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

*Des. Sci.*, vol. 10, e6  
[journals.cambridge.org/dsj](https://journals.cambridge.org/dsj)  
DOI: 10.1017/dsj.2024.3



### 1. Introduction

Descriptive system models have various benefits for the development of technical as well as non-technical systems. In the beginning, these models, for example, with representing dynamics (Forrester 2013), focused on grasping the system’s complexity and existing interdependencies between its own elements and with other systems. Hick, Bajzek & Faustmann (2019) describe the system model as an allocation of central hubs, which are used at different phases in a project (horizontal) at different levels in the system hierarchy (vertical) for the horizontal and vertical distribution of information. With that goal, the system modelling language (SysML) was created. This enables system developers to integrate multiple disciplines and domains, as well as different views and perspectives on a system, into one model (Friedenthal, Moore & Steiner 2015; Albers *et al.* 2019). The main purpose of these models is defined by their use and the improvement of



interdisciplinary communication (Weilkiens 2019). This contrasts with the usually domain-specific models, such as computer-aided design (CAD) models for mechanical engineers and allows to manage system complexity, which is a main area of interest within advanced systems engineering (ASE) (Dumitrescu *et al.* 2021). Fuelled by the increasing complexity of technical products, for example, advanced systems like a Digital Twin and accompanying new business models, this complexity management is a crucial aspect to ensure holistic approaches for the system development of future products on future markets (Albers *et al.* 2022). It requires the usage of advanced engineering methods to handle this complexity and support the system development (Albers *et al.* 2018). This interaction of the three fields advanced systems, systems engineering and advanced engineering defines the field of ASE which creates a frame of actions and a holistic thinking and application of methods.

Within model-based systems engineering (MBSE), the use of system models is an essential aspect to clarify the whole system context and identify possible interdependencies between subsystems and components or with the system environment. Besides the improvement of interdisciplinary communication throughout the system development, it serves for documentation purposes of the system as well. However, several studies investigating the quantifiable benefits of MBSE and system models have revealed that many of the vowed benefits can only be quantified to a limited extent (Campo *et al.* 2023; Henderson *et al.* 2023). Especially the main benefit of system models through communication improvement was identified as an often only presumed benefit in literature that is not yet fully measurable (Henderson & Salado 2021). Along value chains, this quantification aspect gets lost and reveals further obstacles to identify and receive a benefit from using system models (Wiling, Schleich & Wartzack 2020). While system integrators might benefit from the use of system models to manage complexity, suppliers that are forced to comply with the MBSE approach of their customer might not because of the limited complexity of their own products. Especially these forced users of MBSE approaches are seeking for new opportunities to compensate the effort coming along with the approach, for example, creating system models of their products. This reveals that the sole usage of system models is not significant enough to provide the necessary justification for it. Hence, advanced engineering methods are necessary to improve the benefit of system models and increase the feasibility to fully unfold their potentials. Therefore, system models require reuse, combined with these methods, during the product development at different stages, for example, for systems design (Mendieta *et al.* 2017; Mahboob & Husung 2022) or consistent data models (Schwede *et al.* 2019). These two examples show the reuse in terms of recycling of system models, a common and feasible approach for other models such as CAD. Approaches within the last years have shown another type of reusing system models. They focus on how to transform system models into other model artefacts, to enable an authoritative source of truth for other models or derive directly executable code from it, such as shown by Vogel-Heuser *et al.* (2014) or Schumacher & Inkermann (2022). But it is a crucial aspect to consider data availability at specific stages of the system development, for example, by synchronizing models at specific milestones. The system model in this case is not a single source of truth, but supports the model and data management for information that is stored in other models in more detail. Overall, there is a variety of use cases of system model utilization as exemplarily shown by Umeda *et al.* (2022). The term of

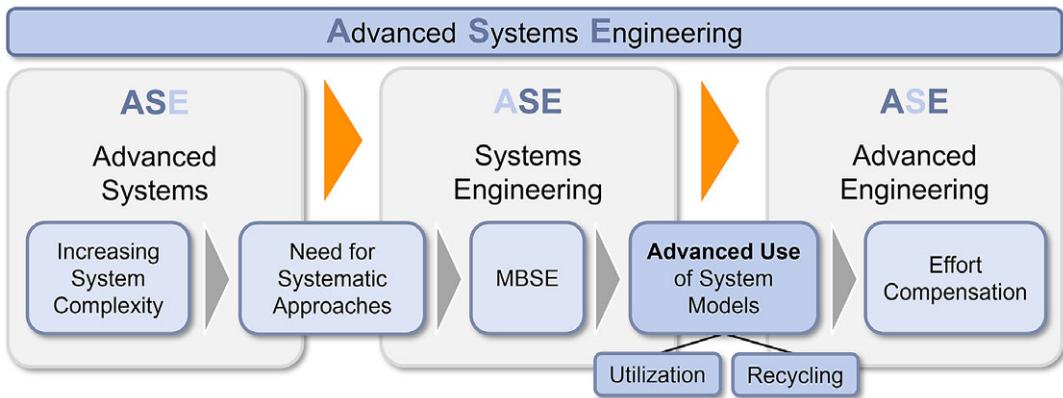
utilization used but not clearly specified by Umeda *et al.* (2022) might therefore address the whole variety of use cases within this context.

Concluding from that, two types of *advanced use* of system models are noticeable. Those describe use forms, such as reuse, that exceed common use scenarios, like documentation and communication purposes. First, the reuse in terms of *recycling* existing content within system models to share information across projects, for example, for the next product generation development. Second, the reuse in terms of system model *utilization*, which involves the vertical as well as horizontal distribution of information and processing of system models, such as the derivation of executable code or development artefact, in order to facilitate advanced engineering methods. The distribution of information is motivated by the aim, to extract information from the system model and support the coping with other challenges along the life cycle or within the system model, for example, by making information available that is stored within the system model but required in a different tool. This initial understanding of utilization and its contextualisation in system model reuse forms the basis for the conducted literature review (see Section 2).

With respect to ASE, the holistic application of MBSE is vital as a systematic approach to handle the complexity of a system. Nevertheless, advanced engineering methods are needed to support its application. With the efforts invested in system modelling, the reuse of those models enables a compensation of such efforts and, therefore, the successful utilization of system models contributes to the fundamentals of ASE, as highlighted in Figure 1. With its mission 2035 described by Albers *et al.* (2019), the community of ASE follows different goals to strive for excellence in engineering and in order to enable the development of successful products. To fulfil this, stakeholders from industry, economy, politics as well as research need to be involved and therefore fields of actions were defined (Albers *et al.* 2019). Especially with regards to those for research, the utilization seems a promising approach in the following defined fields of actions:

- Integrate systems and software engineering by combining different disciplines through model utilization.
- Enabling a cross-cutting activity that can affect the whole life cycle of a system.
- Support data competency of engineering by using a cross-discipline source for information.

Beyond that, utilizing system models promises to be an approach to significantly increase the value of system models and use them throughout the lifecycle of a system. These system model utilizations have a variety of application scenarios that were published over the last years. However, a bibliographical characterization and definition of the term is yet missing. As a foundation for an analysis in this context, the extensive literature review on SysML of Wolny *et al.* (2020) until 2017 can be used. This addresses a research direction identified by them, which concerns the deeper analysis of their results to answer future research questions (Wolny *et al.* 2020). Therefore, the present contribution builds on their results and shifts the focus of the analysis towards the utilization of system models. In this regard, they additionally conclude that future approaches need to focus more on the whole life cycle rather than the implementation phase. Utilizing system models, based on the preliminary definition, is aiming this enlargement towards the whole lifecycle.



**Figure 1.** The three main aspects of ASE regarding Dumitrescu *et al.* (2021) and the contribution of system model reuse.

Coming along with a special interest of software engineering into SysML since 2013 and new visions such as the Digital Twin and the Internet of Things (IoT), recent publications already utilize SysML in different use cases along the life cycle, for example, the operation of Digital Twins or other life cycle stages (Fursin, 2019; Zhang *et al.*, 2021; Wilking *et al.* 2022b).

### 1.1. Aim

Based on the derived initial understanding of system model utilization and taking the general literature study on SysML by Wolny *et al.* (2020) into account, it can be seen that the term of utilization and its characteristic are not considered explicitly within their review. However, the given overview of relevant SysML papers and the derived research directions as key findings of the literature review provide a good insight for further investigations towards the utilization of system models. This complies with their identified research direction which motivates a further investigation of the different approaches and to answer new research questions coming along with them. The creation of a new search string supports the identification of relevant research specified on SysML utilization, even if the favoured term of utilization is not used. In addition, the definition supports a common understanding and could be enriched with a classification system to structure existing research and identify similar approaches to take recourse to existing knowledge. Such a classification system might help to create a foundation for future modelling guidelines and conventions as well as methodical recommendations. While SysML is one language to model systems, other languages, and methods such as ARCADIA (Bajzek *et al.* 2021a) or OPM (Dori 2016) are not considered in this review, as SysML is a widely spread modelling language. It is later discussed how transferable the results are for other languages. The main research questions (RQ) of this contribution are therefore:

- RQ1: Which bibliographical characteristics of a literature database search string enable engineers and researchers to identify relevant literature in the context of system model utilization?

- RQ2: How can the use cases within the relevant papers be provided to support engineers in finding similar approaches to their desired utilization scenario and take recourse to existing knowledge?
- RQ3: Which initial methodical implications can be provided to support engineers and researchers during the process of creating system models that can be utilized within the product life cycle?

To answer the derived research questions, this contribution is divided into two main sections. First, a systematic literature review is conducted in Section 2 to identify relevant use cases and scenarios for the utilization of system models. Second, based on this a definition of the term utilization is presented in Section 3 as well as a derived classification system to identify relevant classes of utilization along the life cycle of the system. This is further used to create a basis for guidelines towards the utilization that support engineers who are aiming to utilize system models.

## 1.2. Significance

Yet, there is no common understanding of the term utilization, which comprises a variety of use and reuse scenarios for system models. By contributing to this understanding, the communication of current and future research in this context can be supported. For example, through a classification system to separate the different approaches and to derive further recommendations towards modelling and guidelines. Therefore, the results of a literature review within this area, which includes the analysis of bibliographical characteristics and is enriched with a definition and a classification system for the approaches, can guide further research in this field. This will support the way towards a common understanding and harmonized modelling approaches. With this, the utilization of system models is a valuable contribution towards the ASE portfolio. While it is based on systems engineering principles by using notation compliant models, the utilization of system models rethinks previous modelling use cases and defines new approaches of how to model and how to further use these models. The complex nature of systems, developed by using systems engineering methods, comes along with a variety of benefits for utilizing system models, as later showed in the examples. Nevertheless, advanced systems like Digital Twins are a likely use case for the utilization as they are highly based on models and require a sophisticated integration of multiple disciplines and domains throughout the life cycle (Schleich *et al.* 2019).

## 2. Literature review

The methodical approach of this publication follows the depicted steps in [Figure 2](#). It begins with the literature review of Wolny *et al.* (2020), which offered a first set of relevant SysML papers. Yet, the review was restricted to publications prior to 2017. Therefore, the review was updated by the authors of this contribution to 2022 by adapting the search string of the previous review, receiving relevant papers from 2005 until the end of 2022. With the initial understanding of system model utilization in Section 1 in mind, relevant papers from the study results were identified. This approach generated 48 relevant papers from the enlarged SysML literature review, which foster the initial understanding of the term utilization.

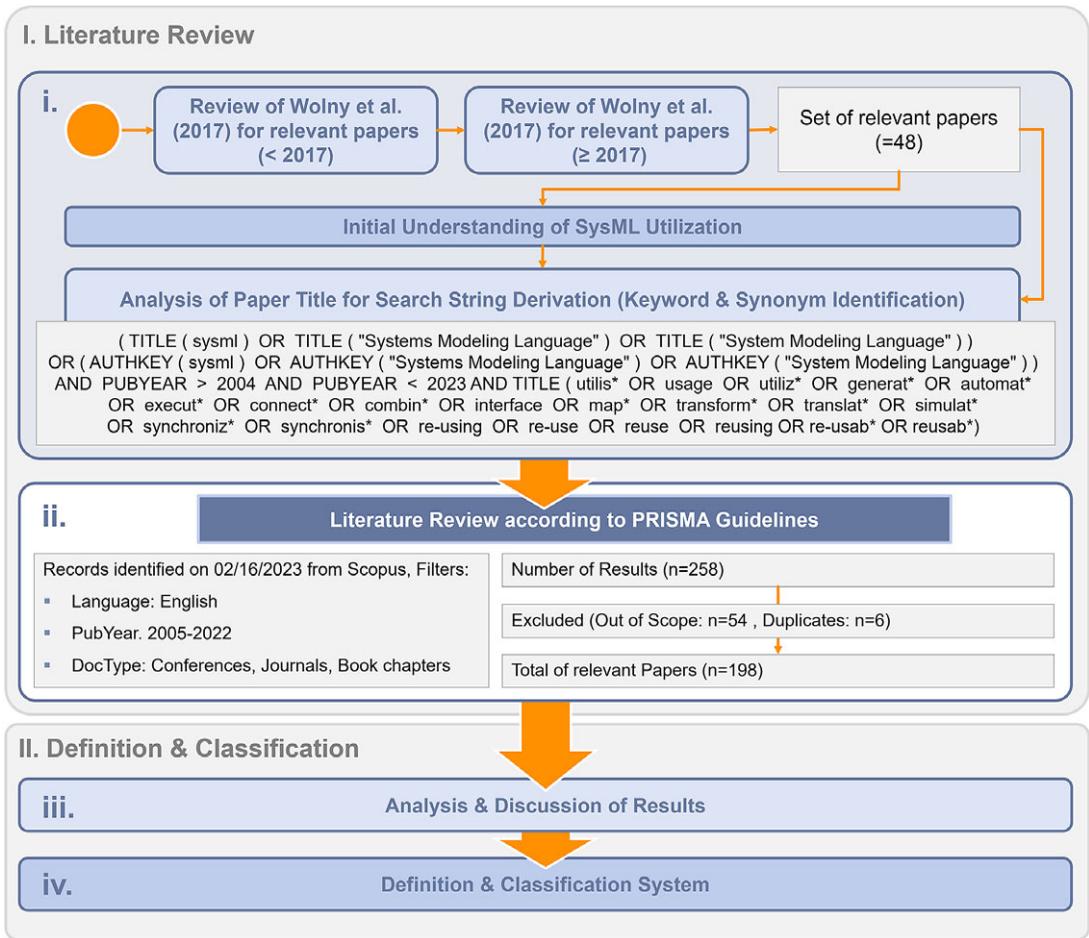


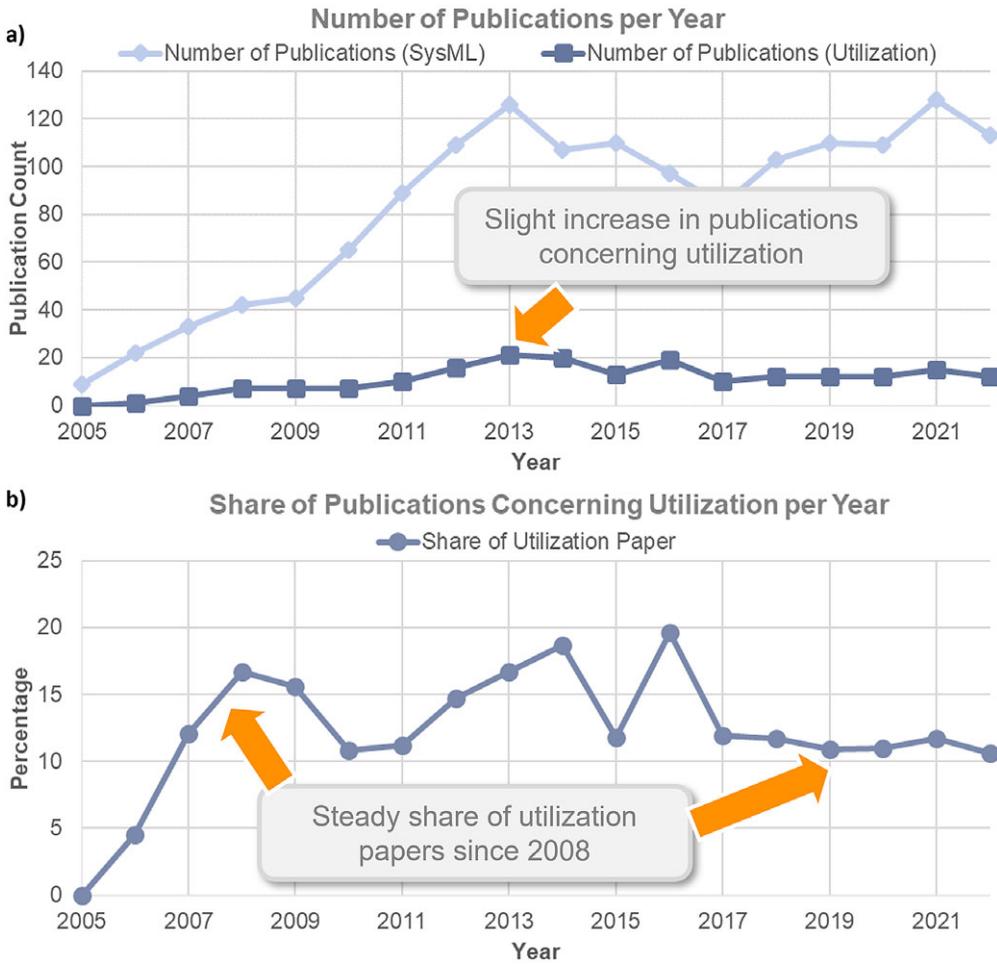
Figure 2. Methodical approach for the systematic literature study on utilization of system models.

To receive the bibliographical characteristics of the utilization, this preliminary set was analysed for relevant search items as well as synonyms of the terms, building up a matrix for verifying the match of the search string with all relevant papers. Resulting from that, the main search string for the literature review on utilization was developed as shown in Figure 2. Similar to Wolny *et al.* (2020) the title was the main search field for SysML papers, specifically searching for “SysML” and written-out versions. Besides, the authors of this contribution decided to add author keywords in combination with relevant synonyms or close terms for the utilization, such as “utilis\*” or “utiliz\*” or connected terms like “generat\*” or “automat\*.” These synonyms were identified by using various English language dictionaries, such as Oxford and Cambridge dictionary. In addition, the identified papers within the previous literature study were taken as source for relevant terms. Thus, the search string combines two aspects. First, the identification of relevant SysML papers by searching for them through title and author keyword. The second part includes all relevant and identified search items, resulting from the initial set of identified papers.

Although stated by the International Council on Systems Engineering (INCOSE) and other publications (Weilkiens 2019), MBSE does not necessarily include the usage of SysML. However, many relevant publications do not specify the occurrence of SysML models within their publications but refer to MBSE models or generic system models (e.g., Karban *et al.* 2016; Horber *et al.* 2021; Zerwas *et al.* 2022). Therefore, a reasonable trade-off is the inclusion of author keywords for “SysML,” as these are actively chosen by the author. Adding MBSE or systems engineering into the search string would lead to a significant increase of the received results without contributing significantly to more relevant papers. With the search string of Figure 2 the literature review was conducted following PRISMA (Page *et al.* 2021). The literature search was conducted on 02/16/2023 and used the scientific database Scopus to cover high-quality and peer-reviewed journal articles, conference papers and book chapters in English language. The publication years were limited to the period up to the year 2022 and after 2005, the latter grounds on the first public release of SysML as described by Wolny *et al.* (2020). The results from Scopus were compared to other databases like IEEE and no additional results were identified. From the resulting 258 contributions, 6 duplicates were removed. After reading the papers’ titles as well as abstracts and comparing the content to the initial understanding of utilization, 54 papers were classified as out of scope and therefore excluded. Out of scope summarizes three different exclusion reasons. First, the majority of the excluded papers describe scenarios in which SysML is used to model a specific system. These papers often contain the part “[...] using SysML” in their titles and therefore are captured with the search string. A second exclusion reason is the use of other modelling languages. These papers use the abstract to introduce another modelling language that is similar to SysML. However, by focusing on SysML only, these papers were excluded from the list of relevant papers. In addition, the term “reuse” is used ambivalently, containing cases which describe the reuse in terms of utilization or recycling of models. With the focus on utilization, all papers that focus the relevant topic of model recycling were not part of this literature review and therefore were excluded. With the included 198 papers, the analysis and discussion of the results is performed in the following and shown in Figure 2 leading to the detailed definition of utilization of SysML system models.

## 2.1. Statistical analysis of relevant papers

Based on the conducted systematic literature review described in the previous section, 198 paper with reference to utilization were identified. In advance to a detailed analysis and discussion of the contributions in the following section, the overall statistics of the results are analysed first. In order to enable a comparison of the results from the study within this paper, a comparative study on the general use of SysML in scientific papers was performed. In contrast to the study of Wolny *et al.* (2020), the search string was not limited to the paper titles, but also considered author keywords. As a result of this comparative study, 1501 contributions concerning SysML were identified. Reviewing those results regarding their years of publication (Figure 3a), a rising trend between 2005 and 2013 as well as a declining trend between 2014 and



**Figure 3.** Comparison of publications per year (a) resulting from systematic literature review on utilization (198 papers) and general search on SysML (1501 papers) as well as the share of publications per year (b) concerning utilization.

2017 can be observed, which matches the findings of Wolny *et al.* (2020). Since the year 2018, the number of contributions focusing SysML increased again to approximately the level of 2017 in the year 2021. Comparing those numbers to the resulting 198 publications within this systematic literature review, it can be observed that the rising trend of SysML papers in general is not represented in the results of utilization papers. In the years after 2009 the number per year declines, but rises again after 2011. After a drop in papers per year in 2015 and a high in 2016, the number averages at about 12 papers per year (Figure 3a).

In order to emphasize this result, the share of publications of general SysML and utilization paper is shown in Figure 3b. Thus, about 11% of the total SysML papers per year address the utilization within the past five years. This leads to the finding in the context of the conducted literature review, that there is a continuous but not rising interest in this aspect of SysML model reuse.

## 2.2. Clustering and distribution

The 198 papers of the literature review show all facets of the term utilization. With the generic initial understanding and the identification of relevant bibliographical characteristics for the search string, a broad field for the further reuse and the utilization is given. However, each of them describes an individual use case to answer a specific research question. For example, while the specific use case is focused on directly creating a Failure Mode and Effects Analysis (FMEA) from a SysML model (Girard *et al.* 2020), the general aim is to derive a development artefact from a system model. Insights of these derived artefacts or documents can lead to manual feedback towards the system model. For creating these classes, the relevant papers were labelled with their specific use cases. Hence, these individual approaches can be clustered into larger classes (see Table 1, where a short description of each is given), which define the overall aim of the paper assigned to those classes. A detailed description of the classes with examples is given in Section 2.3.

**Table 1.** Definition of classes for the utilization of SysML models and assessed papers

Class	Short description	# Paper in the Supplementary Material	Share of papers (%)
Artefact derivation	Artefact derivation describes an approach to utilize system models for receiving relevant documents for the development, such as FMEA, fault trees or test sequences. The documents are not directly executable	[1–23]	12
Execution	Execution involves cases, in which an executable artefact is generated through the system model, for example, the creation of executable software code	[24–35]	6
Model simulation	Model simulation describes the analysis and simulation of system models, for example, for verification and validation purposes. However, the simulation is mainly based on the logic of the system model	[36–58]	12
Model synchronization	With the goal of decreasing inconsistency throughout the models, system model is utilized as a bidirectional model synchronizer	[59–65]	3
Model transformation	Model transformation describes the unidirectional transformation of a system model or parts of it into other models and tools, such as domain-specific models or simulation models	[66–177]	57
Semantic integration	Cases in which information is extracted partially from system models to enrich or combine them with information outside of the system model are labelled as Semantic integration	[178–196]	10

The analysis of the results of the conducted literature study regarding the aforementioned classifications reveals an uneven distribution of research efforts towards the utilization, see [Table 1](#). Since the resulting classes in section only use selected contributions as examples, [Table 1](#) lists all 196 papers within their respective classes. The complete list of literature is provided in the [Supplementary Material](#). For this literature review, the relevant papers were labelled with specific use cases. Merging these use cases led to the creation of the presented classes.

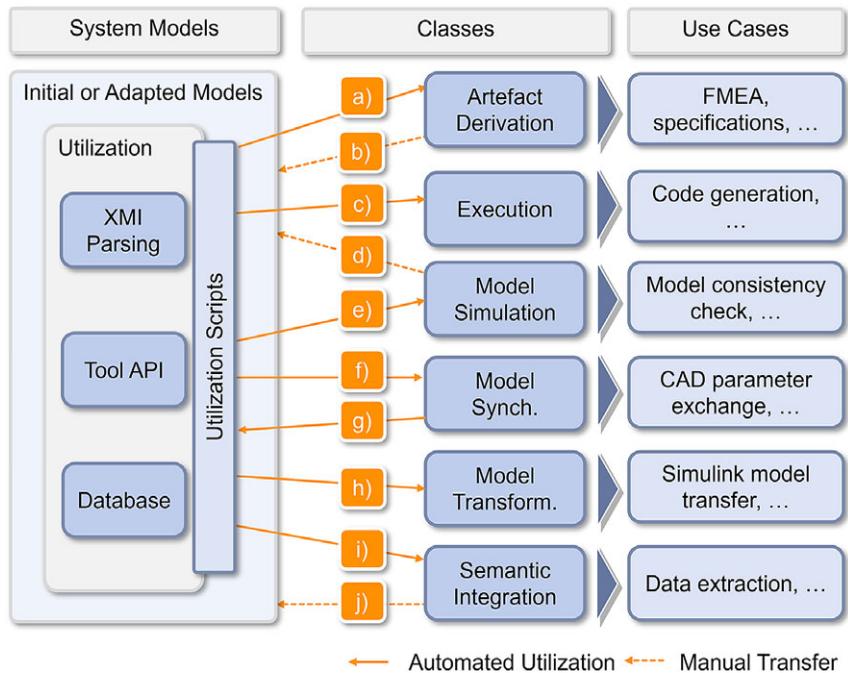
While the majority with 57% of the publications is classified as model transformations, only a fraction of the papers (3%) considers model synchronization, which was simplified as a form of bidirectional model transformation. Interestingly, model transformation could be assumed as a pre-stage for bidirectional synchronization and explain the missing publications in this class as they require preceded works. However, this cannot be observed by analysing the allocation of classes over the years. Generally the term “Single Source of Truth” or “Authoritative Source of Truth” is often mentioned with system models and MBSE approaches (Kruse & Blackburn 2019; Henderson *et al.* 2023). It is considered as one of the main aims for introducing an MBSE approach. Although there is no official definition of this term, it goes along with improved accessibility for information. This could be achieved by synchronizing models bidirectional and defining the system model as the source for relevant information throughout the system development. Nevertheless, the model transformation is the dominant research direction within the utilization of SysML models since 2006.

## 2.3. Detailed class description and interaction with the system model

After labelling all relevant papers, the presented classes in [Table 1](#) were identified as common thread to describe the format of reusing and utilizing system models. In addition to these classes, different approaches for building up the system model can be observed as depicted in [Figure 4](#). While some use the initial system model directly, others are creating new sub-models and packages with the only purpose of utilizing these. The utilization process itself can also be differentiated into three different categories. First, the system model is feeding a database that can then be further used for the different approaches. Second, the tool API is used to connect different tools, for example, the connection between other software, such as Cameo or Matlab Simulink (Chabibi *et al.* 2016). Third, the system model’s XMI structure is utilized to extract the relevant information from the model (Wilking *et al.* 2022a). The approaches are not restricted on the usage of only one class but can be assigned to multiple classes. The individual classes with examples are explained individually in the following sections whilst referring to [Figure 4](#) as indicated by lower case characters.

### 2.3.1. Artefact derivation

SysML is an integrational part of MBSE (Friedenthal *et al.* 2015) defining a contrast to document-based approaches by using models. However, documents are still an essential part of product development and cannot be left out, for example, for documentation purposes. Therefore, a convenient automation ([Figure 4a](#)) is the derivation of documents from existing models. This supports the engineer in



**Figure 4.** Derived classes of utilization with automated and manual tasks and use cases. The assigned characters (a–j) are described individually within the textual class description.

everyday situations and prevents the creation of manual engineering artefacts throughout the product development. A common scenario for the artefact derivation is an automatic creation of Failure Modes and Effects Analyses (Hecht, Dimpfl & Pinchak 2014; Girard *et al.* 2020; Hecht *et al.* 2020) or Fault Trees (Mhenni, Nguyen & Choley 2014). However, artefacts are not limited to static documents but can also be extended to other artefacts such as test cases (Dahlweid, Brauer & Peleska 2015) or Design Structure Matrices (McLellan *et al.* 2009). Insights from these documents can be manually transferred back into the system model (Figure 4b).

### 2.3.2. Execution

SysML is based on UML (Friedenthal *et al.* 2015). While the generation of usable code from UML diagrams is a common approach for over 20 years (Herrington 2003; Ciccozzi, Malavolta & Selic 2019), similar approaches based on SysML models appeared over the last years. These approaches describe a direct generation of executable code from individual SysML diagrams, packages or whole models, for example, to be used as core input for a software tool (Vogel-Heuser *et al.* 2014) or for the definition of the behaviour of a system (Godart *et al.* 2017). However, this category does not involve a semantic integration of information within a SysML diagram to an already existing code frame, but describes the direct execution of a software code generated through the SysML model (Figure 4c). The approach has been identified as helpful to create a single source of truth regarding the software code, as abnormal behaviours of the system as well as failures are easy to identify

(Wilking *et al.* 2022b). In addition, the derivation of executable code saves manual coding effort and the structure of the code is more accessible through the graphical visualization of a SysML diagram.

### **2.3.3. Model simulation**

SysML models are still mainly created by humans and even though their initial aim is the improvement of the overall system understanding, large system models are difficult to overview and check manually. A common use case is the simulation of SysML models, which is often executed in the SysML tool itself (Figure 4e). Typically, model simulations focus on the verification of the model, for example, verifying requirements and designs (Morkevicius & Jankevicius 2015), checking consistency (Bankauskaite & Morkevicius 2018) or event simulation to check the behaviour of a system (Liu *et al.* 2014). Insights from these simulations are likely used to enhance the system model itself (Figure 4d). Unlike the execution, the model simulation does not aim the derivation of executable code outside of the SysML tool.

### **2.3.4. Model synchronization**

The system model in literature is often considered as the single source of truth (Hick *et al.* 2019). Shown by the class of semantic integration and model transformation, the information stored in the system model can be used to compensate development efforts in other stages of the product creation or along the value chain. However, change scenarios and their handling require switching between the layers in the V-Model (Hick *et al.* 2023). For example, a change of a requirement will lead to an impact propagation analysis, often conducted on the physical layer to analyse the effect on specific subsystems and components. However, this requires a connection between discipline-specific models and a deep connection between the other layers of the V-Model. As shown in the class of model transformation, there are already many approaches, which transform a system model into other models (Figure 4f), but not the other way around. Hence, the class of model synchronization describes any cases in which a bidirectional model transformation is conducted to bring information back into the system model and enrich the single source of truth (Figure 4g). This enables consistency throughout the different models in an MBSE approach. In addition, this class involves any case, in which a system model is created from outside of the modelling tool but was not necessarily transformed from it before, such as shown by Chami, Zoghbi & Bruel (2019) and Dworschak *et al.* (2019).

### **2.3.5. Model transformation**

The successful integration of MBSE is not restricted on the usage of SysML system models but involves all relevant models throughout the development (Bajzek *et al.* 2021b). Hence, a consistent toolchain is crucial for the application of MBSE (Ma *et al.* 2022). However, multidisciplinary systems will require synchronous modelling activities during the development. This requires domain-specific models, which are based on a holistic system model that allows interoperability (Brahmi *et al.* 2021). To enable a consistent transformation of a system model into a domain-specific model, an automated approach (Figure 4h) reduces the risk of biased human interpretation. Model Transformation enables the use of SysML

models or individual diagrams within another modelling tool, for example, by defining first parameters for a CAD tool (Schumacher & Inkermann 2022). Other examples are the transformation into digital signal processing and control simulation such as Simulink (Paredis *et al.* 2010; Palachi, Cohen & Takashi 2013) or even the transformation from a SysML into a UML model. It involves the direct transformation of the whole SysML model or diagram into a model within another tool, for example, for further modelling activities (Mahboob *et al.* 2019), simulations (Barbau, Bock & Dadfarnia 2019) or execution (Kapos *et al.* 2021). It does not involve the return of information into the initial system model.

### 2.3.6. Semantic integration

A whole model transformation is not always necessary. In particular cases, information just needs to be extracted from a system model (Figure 4i), for example, for the integration of design requirements into CAD (Brahmi *et al.* 2021) or product architecture design (Schwede *et al.* 2020). This allows the storage of system information in a model that can be used, for example, in domain or discipline-specific applications. This is significantly relevant for information that is used by multiple applications or is transferred along the value chain (Wilking *et al.* 2020) and is exchanged through several system models. In fact, the extracted information can later be used to build up another model as shown by Kerzhner & Paredis (2011). In addition, the extracted and processed information leads to new insights about the system that can be manually handed back into the system model (Figure 4j). For example, for the OEM or for suppliers, it can be crucial to store the information in a more generic system model rather than in a discipline-specific model. The recipient can read the necessary information in the system model instead of manually searching for it, which often comes along with human errors and misinterpretation. However, this class describes the specific search for information within the model and does not attempt to transform the whole model into another.

### 2.3.7. Other

Few papers describe cases of SysML utilization that are not classifiable in any of the aforementioned classes but for example, describe methodical approaches for the utilization (Fu, Liu & Wang 2021).

## 3. System model utilization

### 3.1. Definition and classification system

The literature review enabled the analysis of bibliographical characteristics for the reuse and utilization of SysML system models. Within the introduction in Section 1, an initial understanding was given, that was based on the subjectively and implicitly usage of the term “utilization”. To enable a common understanding of the term as well as giving a basis for harmonized modelling strategies, a definition is presented in this section. Shortly after the introduction of SysML first attempts were made to further reuse the models. In fact, reuse describes the generic reuse of the models and therefore can also mean that elements of a system model are reused, for example, throughout several product generations (Albers, Bursac & Wintergerst 2015). This reuse is defined as an *advanced use* of the three purposes

for system modelling, that is, improvement of communication, analysis of the technical system and the documentation of it (Friedenthal, Dori & Mordecai 2021).

More specific the utilization of system models describes the reuse for use cases or several application scenarios along the product life cycle of the system by creating machine-readable system models. The authors of this contribution therefore define utilization as a subcategory of reusing system models. Those models are used for automated vertical and horizontal distribution of their included information as well as interfacing domain-specific tools (e.g., requirements management system). Therefore, utilization adds new objectives for existing or adapted system models, whereas recycling of system models does not change the objectives of the modelled information. This definition of the utilization is shown in Figure 5. However, utilization is only one form of reusing system models, as mentioned before.

The utilization of system models in the context of SysML can therefore address the introduced classes of artefact derivation, execution, model simulation, model transformation, semantic integration and/or model synchronization (Figure 5). Although this review was conducted for SysML models only, the classification system is also suitable for other languages and utilization scenarios, such as within Capella and the connection with discipline-specific models. However, other approaches might lead to additional use cases that cannot be grasped by SysML.

In addition to the classes, every single use case of utilization, such as verification and validation, Digital Twins, and so on, and stages within the life cycle according to ISO/IEC/IEEE 15288 (2015) can be assigned to a class. This leads to a three-dimensional classification system for the utilization of system models, as shown in Figure 6. The orange indications within this figure are relevant for the exemplary application for two different scenarios of the proposed classification system within Section 3.2. Interdependencies between the three axes are possible. While classes can be found in any life cycle stage, specific use cases might predefine a class and a

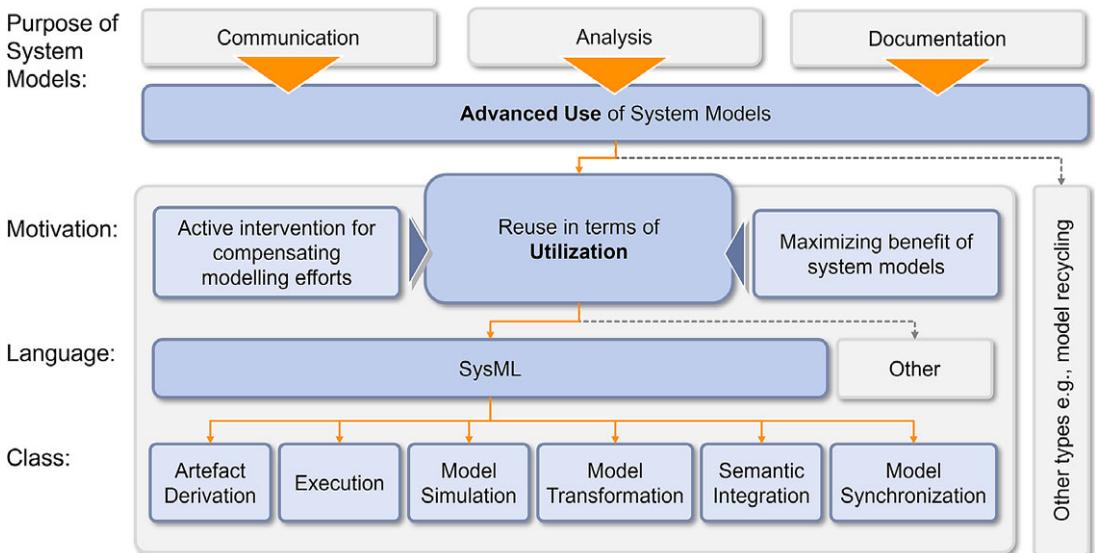
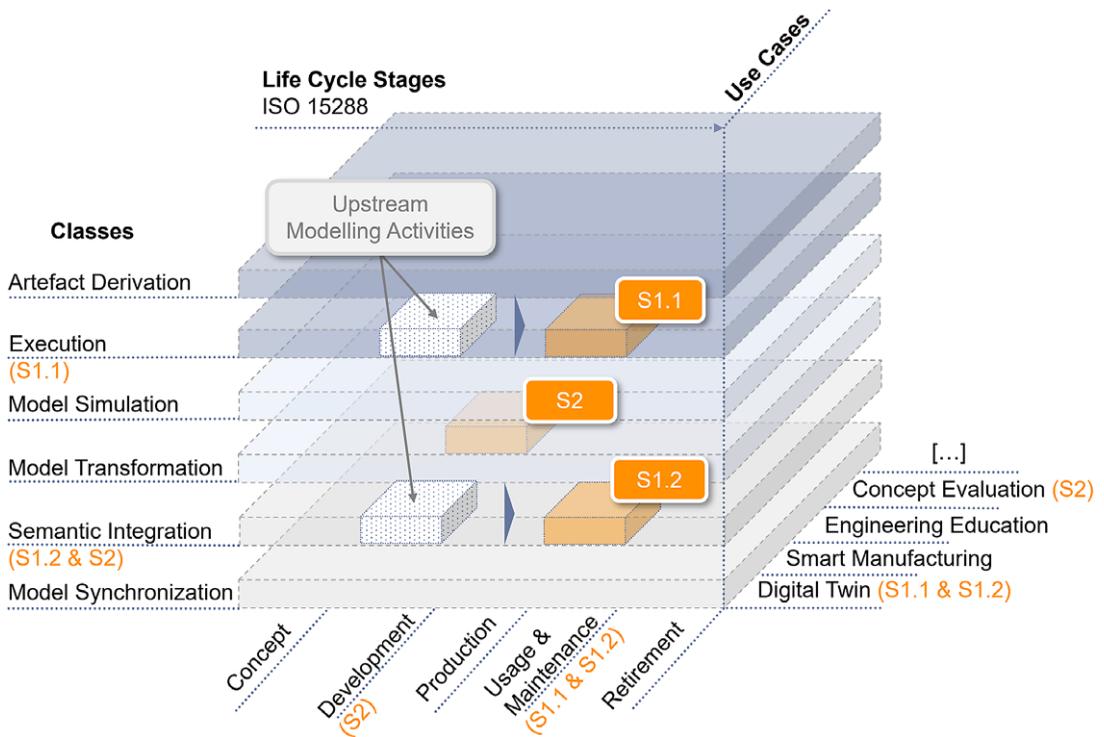


Figure 5. Definition of the reuse in terms of system model utilization.



**Figure 6.** Proposed classification system for the utilization with integration of scenarios S1.1 and S1.2 (see Table 2) with upstream modelling activities and scenario S2 (see Table 3) into the classification system; visual design based on VDI/VDE Society Measurement and Automatic Control (2015).

stage of the life cycle. This does not lead to a limitation of the classification system, but reveals applied and potentially more feasible approaches.

For an application of the classification system and finally the derivation of connected methodical modelling approaches, further considerations have to be stated. Since machine-readability and interoperability between SysML and other models are highly depending on a tool support, many of the described scenarios are using the XMI structure of SysML models (Handley *et al.* 2021). This requires awareness in the modelling process of the system model. As humans are able to interpret text fields on a diagram and might make the correct association with nearby diagram objects, the machine-readable context is gone and information is lost in the visualization of the model, but not in the model itself. This is not necessarily caused by a false application of SysML, but rather by the given interpretation spaces coming along with this language. The further utilization of system models therefore requires modelling recommendations and guidelines or simplified APIs that enable the utilization described in this definition. Universal methodical approaches are not yet existing, since the applications are use case specific and might require different methods. Therefore, a thorough classification of the approach, for example, through the presented classification system, might help to develop certain guidelines and methods to enable a harmonized approach for the utilization. However, the allocation within the classification system must not be limited to a distinct point but can be enlarged to multiple sections, for

example, for relevancy along different life cycle stages. Nevertheless, a separation of the class is recommended, as these significantly influence the purpose of the utilization. Yet, multiple classes can be combined to a connected application scenario that involves different classes but their purposes are dissociated from each other. The recommended classification system therefore offers a separation of the described classes in layers, as they are strictly separated. In addition, the context of the life cycle stage is given based on ISO/IEC/IEEE 15288 (2015). Models have different purposes regarding their usage within the life cycle of a system. Nevertheless, almost every class of the utilization can be discovered in every stage of the life cycle regarding the analysed paper, which can be seen in the [Supplementary Material](#). Furthermore, the separation of use cases for the utilization supports the definition of a goal for the approach that might be achieved in a later stage of the life cycle but must be conducted in a previous one, for example, through the creation of models for the operation phase that are created during the development. This shift of the required effort can be useful to create the right modelling approach in the right context of the product creation. The given examples for use cases, such as the Digital Twin, show this shift. While Digital Twins might benefit from the utilization of the SysML model during operation, the required modelling and utilization effort will happen during its development and concept phase as well as the twinning during the production and instantiation (Stark *et al.* 2020). But the assignment to a use case can be shaped by different implications and can have intersections with other use cases. The shown use cases in [Figure 6](#) are examples for published work, such as a system model-driven student project for engineering education (Wilking *et al.* 2023). While some use cases reveal technical implications for methodical recommendations, others can simply provide similar approaches to the designer. Used within product development, the classification of the system model towards utilization reveals insights in the application scenario as well as required modelling activities. This is necessary to develop further methodical approaches for the specific utilization scenario.

## 3.2. Application of the proposed classification system for utilization

For a further description of the definition of the term and the integrated classification system, as well as first recommendations towards guidelines for a methodical approach, two scenarios are given which describe different use cases, classes, and stages of the product life cycle. The variety of use cases for the utilization of system models is huge. However, the two use cases can give first valuable feedback on the requirements for modelling guidelines or methodical approaches towards the utilization. [Figure 6](#) shows their integration into the classification system, where they are indicated in orange colour.

### 3.2.1. S1: SysML 4 Digital Twins

Wilking *et al.* (2022b) describe an approach to utilize SysML models that are visualizing the behaviour of a Digital Twin system. Since models are a significant part of a Digital Twin (Schleich *et al.* 2017), MBSE is a promising approach for the design of Digital Twins (Wang *et al.* 2021). System models add a special value to the development and operation of Digital Twins by defining the general behaviour of

**Table 2.** Scenario description of “SysML 4 Digital Twins”

Scenario	Description
Goal:	The behaviour of the Digital Twin is modelled within SysML to visualize the complex behaviour regarding received input data and simulation results. These models are derived into directly executable code to feed a Digital Twin software prototype
Classification: (see <a href="#">Figure 6</a> )	Class: Execution and Semantic Integration; Life Cycle Stage: (Development), Operation and Maintenance; Use Case: Digital Twin
Benefit:	The direct derivation of the executable code saves coding effort and enables a significantly easier error identification through the visualization within the SysML diagram

the physical and digital counterparts as well as connecting relevant information and other models that are used within the Digital Twin concept. Examples for this are the integration of the system into its environment or the inclusion of model decay regarding the physical counterpart throughout its lifecycle into the behaviour of the system. In this scenario, a system model was created as a central model that visualizes the interaction of multiple relevant models for the Digital Twin as well as the detected and stored operational data from the physical counterpart. Packages of that model were directly derived and used as executable code to base the actual behaviour of the Digital Twin completely on the SysML model, see [Table 2](#). In addition, a machine learning algorithm as well as relevant information for its execution was implemented into the system model to directly connect physical elements of the model with relevant elements for machine learning (Wilking *et al.* 2022a). This enabled parallel activities to model and execute the behaviour of the Digital Twin, for example, by conducting a precise but time-consuming simulation and parallel run a machine learning script to gather less detailed but therefore quick recommendations towards the current state of the system and required actions, for example, maintenance intervals. This scenario shows a case in which the benefit of the utilization is achieved within a later stage of the life cycle, but the effort has to be invested already during the development. This is a significant shift of effort and benefit and requires a thorough consideration within the modelling approach, as existing approaches do not yet involve the utilization and the preparation of utilized system models. In addition, the whole use case is taking advantage of two classes, showing that a combination is possible and for some cases reasonable.

### 3.2.2. S2: Formalization of integrated variation management and its use in robustness evaluation

A second scenario is introduced by Horber *et al.* (2022a), which proposed an approach to formalize engineering activities in the context of integrated variation management. Motivated by the variety of different methods, tools as well as data in variation management and a lack of their integration into a consistent model, the approach uses activity, requirement and block diagrams (SysML) to model the contents of integrated variation management. Its use enables the consistent reuse

**Table 3.** Scenario description for formalization of integrated variation management

Scenario	Description
Goal:	Engineering activities are formalized within integrated variation management and integration of relevant data, tools and methods into a system model. Information from a system model to support robustness evaluation of product concepts is retrieved
Classification: (see <a href="#">Figure 6</a> )	Class: Semantic Integration; Life Cycle Stage: Development; Use Case: Concept Evaluation
Benefit:	The formalized approaches of integrated variation management enable a consistent reuse of generated data (e.g., key characteristics) throughout the concept and development phase and propagation of changes from later stages of development into early concept decisions and vice versa

of model elements in connected approaches within the variation management domain, see [Table 3](#). As applied by Horber *et al.* (2022a) to early stage variation management, which focuses on the development of robust product concepts, the created model can be utilized for further approaches, for example, the robustness evaluation (Horber *et al.* 2022b). The approach uses qualitative sketches of product concepts, combines them with embodiment-function-relations and transfers the qualitative model into a corresponding SysML model. The contents of the model can then be retrieved for robustness evaluation and build the basis for further development of the selected concept. Other model elements, such as key characteristics, are thereby relevant for even later stages like parameter design or even production.

### 3.3. Methodical implications resulting from the classification system

The two described scenarios give a detailed insight into the utilization of SysML system models as an equal purpose for the creation of system models. While the life cycle stages and classes are finite, the use cases allow various application scenarios for the utilization. However, it does not seem that the use cases affect the modelling approach significantly. In contrast, classes and stages of the life cycle have a significant influence on the approach. While the classes are naturally describing separate approaches and aims of the utilization, life cycle stages strongly influence the approach by determining the context of the creation and the use for a not yet or already existing system with significant differences for the modelling.

However, the scenarios have shown that a methodical approach towards harmonizing the modelling procedure is crucial. This is especially the case where the utilization is conducted along a value chain and where harmonized modelling strategies between stakeholders are necessary. Currently, some system modelling tools support interfaces with other tools and by that enable a model transformation for a further utilization. Thus, many of the other defined classes are not yet sufficiently integrated into systems model toolchains, as this is not only relevant

for the system modelling tool, but also for discipline and domain-specific tools that integrate the system model.

Many use cases are utilizing the XMI structure of the modelling language to further utilize the models. The XML files contain the whole model as well as the visualization. This leads to immense and confusing model sizes. Hence, individual packages within the model must be created, which leads to scenarios in which a model is just created for its further utilization but not used again. Thus, the overall benefit of the approach is diminished. A possible solution for this could be the introduction of a utilization view which enables the reduction of a model towards necessary aspects for the utilization without affecting the model itself. Views are a common tool in SysML to enable different degrees of detail on a model as well as different perspectives depending on the discipline and domain (Alt 2012; Albers *et al.* 2019). With this, it can be ensured that the model itself can still be used for other purposes, while the utilized version is reduced. However, this reduction cannot be generalized. In fact, while a model transformation reasonably should transfer the visual aspects of a diagram, other utilization classes might not use the diagram at all.

The need for such a methodical approach, that integrates the aimed utilization into the methodical model creation, and its effect on the benefit was shown by Fu *et al.* (2021) for the reuse of system models. Mendieta *et al.* (2017) already discussed the tool-specific obstacles coming along with XML and SysML model recycling. Nevertheless, the presented classification system offers a first basic structure for modelling guidelines and recommendations. Especially with SysML v2, which is no longer based on UML and offers new potentials, for example, through changing the restriction of the XML exchange format by an accessible API.

The different classes distinguish fundamentally different approaches for the further reuse of SysML models with different aims and necessary efforts. While model transformations might aim to transfer individual diagrams into domain-specific models, such as shown in the transfer of electrical system designs between the Siemens tools Capital Systems Capture and System Modelling Workbench (Hick *et al.* 2023), others will require the collection of information from the whole model. Especially the transfer of the visual layout of a diagram into domain-specific tools will cause tremendous effort to develop certain modelling guidelines to ensure a transfer without information loss and, most importantly, a benefit with the domain or discipline-specific model. Furthermore, a synchronization requires a harmonized modelling approach throughout the whole model. This shows that the creation of the methodical approach depends on the aim and the corresponding class that was assessed for the utilization of the system model.

This basis of considerations for the methodical approach is enriched by viewing the life cycle stages. Engineers who want to utilize system models will be confronted with several challenges regarding the life cycle of the system. The model maturity significantly differs throughout the development until the instantiation of the system. Therefore, the actual point in time for making use of and benefit from the utilized model might differ from the point in which the system model is created. This leads to two aspects coming along with a life cycle related consideration for the utilization. First, the life cycle stage itself in which the model is utilized. A utilized system model for a concept phase will be based on a low-matured model foundation. Furthermore, it has different aims. Models, which are being used in later

stages, will be confronted with a different level of detail regarding the system model or the connected domain-specific models. The second important aspect is considering the difference between utilization and creation of the model. This is a crucial consideration for models that decay over time and do not represent, for example, the behaviour of a system throughout the period of use. This decay needs to be considered. Approaches must integrate this dynamic to represent the stage of a system, in which the system model is utilized and not only the state in which the model is created. Summing up, methodical approaches have to consider these two differentiations for the utilization.

An additional aspect for considerations towards methodical approaches is the differentiation of use cases within the utilization. For example, utilization approaches for the Digital Twin use case are mainly focused on the operation phase of the system predefined by the use case that is describing a predominantly maintenance- and usage-oriented concept. In fact, specific approaches, such as deriving specified documents, for example, a FMEA can help to define a very specific guideline for the utilization, such as seen in several publications (Girard *et al.* 2020; Hecht *et al.* 2020). Some of the specific approaches can be clustered together with the classes of the utilization. However, as SysML application throughout enterprises and sectors can be affected by the usage of own profiles and adaptations. Specific guidelines for these might be an inspiring example. But they do not necessarily provide a generic approach for supporting enterprises in the attempt of utilizing their system models.

## 4. Limitations

The results of this systematic literature are analysed aiming towards the identification of relevant literature in the context of utilization. As a limitation, the derived search string only shows use cases within the language of SysML as shown in [Figure 5](#). As a widely spread language for modelling systems, this limitation is reasonable but can cause the lack of use cases connected to other languages. An example for this is Capella with its ARCADIA method (Voirin *et al.* 2016) and a possible PLM integration for system models. From a methodical point of view, other languages might result in other implications when it comes to modelling processes. But the general aim of system modelling and the utilization is similar across the languages and therefore not language neutral.

The literature review was conducted along with the PRISMA guidelines, but the results are limited due to the used search string. To ensure reliable results, the initial understanding of utilization was derived from the literature review of Wolny *et al.* (2020) and the search was conducted in a broader way by taking author keywords besides titles into account. Hence, it cannot be finally excluded whether all papers were identified but the methodical approach reduces this risk.

In addition, the presented classification system is an open one. While the classes and the life cycle stages are fixed, the differentiation between use cases is not yet limited and standardized, as more and more papers are published, which add new application scenarios for the utilization of system models. However, at one point these use cases must be clustered to enrich the classification system and differentiate more precise towards specific modelling guidelines and recommendations for methodical approaches.

## 5. Discussion and conclusions

The high number of publications, which are focussing the utilization of SysML models, shows that the models and their purpose have reached a next step of their evolution. The model transformation is the dominant research direction for the utilization, immediately compensating effort by transforming existing system models into discipline or domain-specific models. As discussed, this could be a first step towards an authoritative source of truth within MBSE and particularly in system modelling. But it reveals that there are still lots of undiscovered potentials within this research field, such as data availability. However, model synchronization is not necessarily needed at any point throughout development but is required at specific milestones to achieve consistency for the next stage of the development process. Especially for complex systems, this consistency is difficult to achieve along the used models. Change scenarios reveal missing model links in case of missing synchronization, thus preventing traceability. These two classes, model transformation and synchronization, require a significant effort for a methodical approach, modelling guidelines and a sufficient tool integration, considering the different aspects of them (Khandoker *et al.* 2022; Saqui-Sannes *et al.* 2022). This methodical approach is highly relevant to integrate the utilization into existing modelling activities and achieve the holistic aim of compensating effort. The other classes are promising towards a quicker integration and directly benefiting from applying the underlying use cases of the classes, for example, through document derivation.

Enterprises might naturally benefit from using system models, while non-classical sectors that recently just began integrating MBSE methods and models into their development approaches seek for more benefits. Their products might be on the edge of feasible complexity for integrating MBSE. Justifying systems engineering and MBSE through quantifiable performance indicators is still an ongoing research topic (Honour 2013; Henderson *et al.* 2023). Compared to current approaches, the utilization of system models offers a quantifiable comparability, for example, through saved lines of code or reduced modelling time.

With this contribution, by giving a literature-based definition and classes, combined with scenarios, the diverse field of SysML utilization has been clustered into different research directions. The definition offers a basis for further research, such as building up modelling strategies and introducing the utilization of system models into methodical approaches. The definition, in combination with the bibliographical characteristics of the term, will help other researchers to identify the intersection of their research with the utilization. It also reveals that a generic recommendation for a methodical approach to conduct this utilization is not feasible, but can be characterized through the presented classification system. This demands further research on specific approaches and the realization of sophisticated utilizations of system models. Especially the realization of the six presented classes includes further potential for modelling activities, use cases and software integration. Based on the results, the defined research questions of this contribution can be answered as follows:

- RQ1: The developed search string depicted in Figure 2 shows the bibliographical characteristic of the term utilization. Engineers and researchers are enabled to use this string to identify relevant research in this field and conduct further analysis towards new research questions. It was revealed that a further adaption

of the search string, for example, by splitting reuse into “use”, leads to a significant enlargement of the results, which, however, does not increase the amount of relevant paper. This is because most approaches simply describe the use of SysML, which does not refer to utilization (see [Figure 5](#)). With this contribution and the [Supplementary Material](#), a collection of relevant material for engineers, who are trying to implement the utilization of system models into their work, is presented. In addition, with the presented classes and their interaction with the system model, depicted in [Figure 4](#), similar approaches can be identified and be extracted from the [Supplementary Material](#) to reduce time-consuming research.

- RQ2: Derived from the literature review, a holistic definition of the term “Utilization” was proposed and summarized in [Figure 5](#). Based on that, a possible classification system was provided in [Figure 6](#) to support future attempts for specific methodical approaches that are designed to address a specific class and use case of the utilization. By that, researcher and practitioners can take recourse to existing knowledge in the form of similar approaches when conceptualizing system models for utilization.
- RQ3: Along with general methodical implications in Section 3.3, two scenarios were mapped to the given definition and classification system to show scenario-specific implications. Nevertheless, further integration of utilization requires tool support and the current possibilities can be limited by the tool capabilities. In addition, other scenarios might lead to more and different methodical implications. Generally, with this contribution the following three initial methodical implications for engineers and further research were identified:
  - Specific modelling rules or guidelines are needed, especially for taking advantage of the full possible utilization, for example, to harmonize the created system models.
  - While it is crucial to aim the utilization of existing system models, the isolation of these might be useful to narrow them purposefully. This requires the introduction of a new view that aims the utilization of these models and combine the initial system model with the adapted version. For example, this new view can be an additional layer for diagrams of a system model, specifying their reuse in terms of utilization.
  - Model maturity can have a significant impact on the possibilities and implementation of the system model utilization. An example for this is the availability of information at specific milestones of a project and how this information can be provided for specific use cases of utilization. This must be considered for the creation of these models and could be a future research topic to enhance system modelling languages to further meet this requirement.

In summation, a system-model-centred approach is a promising direction for future system development (Bajaj, Zwemer & Cole 2016). The utilization of system models will be the key for further realization of this idea. SysML v2 shows a promising approach for this further integration (Bajaj, Friedenthal & Seidewitz 2022), for example, through the accessible API. The definition and classification system were shown for SysML models, but it can be stated that the definition and classification are not limited to this modelling language, as bridges might exist between languages (Badache & Roques 2018). Therefore, principles of the

definition as well as the classification are directly translatable towards the other languages or potential enrichments of the languages such as shown with SysML v2.

The upcoming advanced systems require the further integration of systems engineering into the development approaches of enterprises. Advanced engineering methods, such as from digital engineering, can help to enable development strategies for this future of advanced systems. The utilization of system models promises to be a useful approach and addition to the pool of Advanced Engineering methods. It enables the compensation of system modelling effort and a partial evaluation of it. Furthermore, it integrates disciplines into the systems engineering, which are three of the main goals of ASE (Albers *et al.* 2018).

## Supplementary material

The supplementary material for this article can be found at <https://doi.org/10.1017/dsj.2024.3>.

## Acknowledgements

This project is supported by the Federal Ministry for Economic Affairs and Climate Action (BMWK) on the basis of a decision by the German Bundestag. It is part of the IGF Project 22467 BG (FVA 889 II Digital Twin) in collaboration with the Forschungsvereinigung Antriebstechnik (FVA) e.V.

Supported by:



Federal Ministry  
for Economic Affairs  
and Climate Action

on the basis of a decision  
by the German Bundestag

## References

- Albers, A., Bursac, N., Scherer, H., Birk, C., Powelske, J. & Muschik, S. 2019 Model-based systems engineering in modular design. *Design Science* 5, e17; doi:[10.1017/dsj.2019.15](https://doi.org/10.1017/dsj.2019.15).
- Albers, A., Bursac, N. & Wintergerst, E. 2015 Product generation development - importance and challenges from a design research perspective. In *Proceedings of the International Conference on Mechanical Engineering (ME 2015), Vienna, Austria, March 15–17* (ed. N. E. Mastorakis & C. W. Solomon To), pp. 16–21. KIT.
- Albers, A., Dumitrescu, R., Gausemeier, J., Lindow, K., Riedel, O. & Stark, R. 2022 *Strategie Advanced Systems Engineering - Leitinitiative zur Zukunft des Engineering und Innovationsstandorts Deutschland*. National Academy of Science and Engineering.
- Albers, A., Dumitrescu, R., Gausemeier, J., Riedel, O. & Stark, R. 2018 *Advanced Systems Engineering – Eine Leitlinie zur Stärkung der Innovationskraft (acatech Kooperation)*. Acatech – Deutsche Akademie der Technikwissenschaften, Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Fraunhofer-Institut für Entwurfstechnik

- Mechatronik (IEM), Fraunhofer-Institut für Produktionsanlagen und Konstruktionstechnik (IPK), Institut für Produktentwicklung (IPEK) am Karlsruher Institut für Technologie (KIT).
- Alt, O.** 2012 *Modellbasierte Systementwicklung mit SysML*, Hanser-eLibrary. Hanser, Carl.
- Badache, N. & Roques, P.** 2018 Capella to SysML bridge: a tooling methodology for MBSE interoperability. In *9th European Congress Embedded Real Time Software and Systems*. HAL.
- Bajaj, M., Friedenthal, S. & Seidewitz, E.** 2022 Systems modeling language (SysML v2) support for digital engineering. *Insight* **25** (1), 19–24; doi:[10.1002/inst.12367](https://doi.org/10.1002/inst.12367).
- Bajaj, M., Zwemer, D. & Cole, B.** 2016 Architecture to geometry - integrating system models with mechanical design. *AIAA Space and Astronautics Forum and Exposition, SPACE 2016*, 1–19; doi:[10.2514/6.2016-5470](https://doi.org/10.2514/6.2016-5470).
- Bajzek, M., Faustmann, C., Krems, D., Kranabittl, P. & Hick, H.** 2021a A concept for a digital thread based on the connection of system models and specific models. *INCOSE International Symposium* **31** (1), 1035–1048; doi:[10.1002/j.2334-5837.2021.00885.x](https://doi.org/10.1002/j.2334-5837.2021.00885.x).
- Bajzek, M., Fritz, J., Hick, H., Maletz, M., Faustmann, C. & Stieglbauer, G.** 2021b Model based systems engineering concepts. In *Systems Engineering for Automotive Powertrain Development* (ed. H. Hick, K. Küpper & H. Sorger). Springer Nature.
- Bankauskaite, J. & Morkevicius, A.** 2018 An approach: SysML-based automated completeness evaluation of the system requirements specification.
- Barbau, R., Bock, C. & Dadfarnia, M.** 2019 Translator from extended SysML to physical interaction and signal flow simulation platforms. *Journal of Research of the National Institute of Standards and Technology* **124**, 1–3; doi:[10.6028/jres.124.017](https://doi.org/10.6028/jres.124.017).
- Brahmi, R., Hammadi, M., Aifaoui, N. & Choley, J.-Y.** 2021 Interoperability of CAD models and SysML specifications for the automated checking of design requirements. *Procedia CIRP* **100**, 259–264; doi:[10.1016/j.procir.2021.05.064](https://doi.org/10.1016/j.procir.2021.05.064).
- Campo, K. X., Teper, T., Eaton, C. E., Shipman, A. M., Bhatia, G. & Mesmer, B.** 2023 Model-based systems engineering: evaluating perceived value, metrics, and evidence through literature. *Systems Engineering* **26** (1), 104–129; doi:[10.1002/sys.21644](https://doi.org/10.1002/sys.21644).
- Chabibi, B., Douche, A., Anwar, A. & Nassar, M.** 2016 Integrating SysML with simulation environments (simulink) by model transformation approach. In *2016 IEEE 25th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)*, 13.06.2016–15.06.2016, Paris, France, pp. 148–150. IEEE; doi:[10.1109/WETICE.2016.39](https://doi.org/10.1109/WETICE.2016.39).
- Chami, M., Zoghbi, C. & Bruel, J.-M.** 2019 A first step towards AI for MBSE: generating a part of SysML models from text using AI. In *INCOSE Artificial Intelligence for Systems Engineering: 2019 Conference Proceedings*, 1st Ed. INCOSE.
- Ciccozzi, F., Malavolta, I. & Selic, B.** 2019 Execution of UML models: A systematic review of research and practice. *Software & Systems Modeling* **18** (3), 2313–2360; doi:[10.1007/s10270-018-0675-4](https://doi.org/10.1007/s10270-018-0675-4).
- Dahlweid, M., Brauer, J. & Peleska, J.** 2015 Model-based testing: automatic generation of test cases, test data and test procedures from SysML models. In *SAE Technical Paper Series, Sep. 22, 2015, SAE International 400 Commonwealth Drive, Warrendale, PA*. SAE International; doi:[10.4271/2015-01-2553](https://doi.org/10.4271/2015-01-2553).
- Dori, D.** 2016 *Model-Based Systems Engineering with OPM and SysML*. Springer.
- Dumitrescu, R., Albers, A., Riedel, O., Stark, R. & Gausemeier, J.** 2021 *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft: Ein Beitrag zum Advanced Systems Engineering*. Universität Paderborn.

- Dworschak, F., Zirngibl, C., Schleich, B. & Wartzack, S.** 2019 Konzept für den MBSE-Einsatz zur automatisierten Individualisierung von komplexen Produkten. In *DFX 2019: Proceedings of the 30th Symposium Design for X*, 18–19 September 2019, Jesteburg, Germany. The Design Society; doi:[10.35199/dfx2019.24](https://doi.org/10.35199/dfx2019.24).
- Forrester, J. W.** 2013 *Industrial Dynamics*. Martino Publishing.
- Friedenthal, S., Dori, D. & Mordecai, Y.** 2021 SEBoK - Why model?, online document (downloadable on October 18th, 2023) [https://sebokwiki.org/w/index.php?title=Why\\_Model%3F&oldid=67848](https://sebokwiki.org/w/index.php?title=Why_Model%3F&oldid=67848).
- Friedenthal, S., Moore, A. & Steiner, R.** 2015 *A Practical Guide to SysML: The Systems Modeling Language*, 3rd ed. Elsevier; Morgan Kaufmann.
- Fu, C., Liu, J. & Wang, S.** 2021 Building SysML model graph to support the system model reuse. *IEEE Access* **9**, 132374–132389; doi:[10.1109/ACCESS.2021.3115165](https://doi.org/10.1109/ACCESS.2021.3115165).
- Fursin, G.** 2019 SysML'19 demo: customizable and reusable Collective Knowledge pipelines to automate and reproduce machine learning experiments. <http://arxiv.org/pdf/1904.00324v1>.
- Girard, G., Baeriswyl, I., Hendriks, J. J., Scherwey, R., Müller, C., Hönic, P. & Lunde, R.** 2020 Model based safety analysis using SysML with automatic generation of FTA and FMEA Artifacts. In *Proceedings of the 30th European Safety and Reliability Conference and 15th Probabilistic Safety Assessment and Management Conference, 01.11.2020–05.11.2020* (ed. P. Baraldi, F. Di Maio & E. Zio), pp. 5051–5058. Research Publishing Services; doi:[10.3850/978-981-14-8593-0\\_4941-cd](https://doi.org/10.3850/978-981-14-8593-0_4941-cd).
- Godart, P., Gross, J., Mukherjee, R. & Ubellacker, W.** 2017 Generating real-time robotics control software from SysML. In *2017 IEEE Aerospace Conference, 04.03.2017–11.03.2017, Big Sky, MT*, pp. 1–11. IEEE; doi:[10.1109/AERO.2017.7943610](https://doi.org/10.1109/AERO.2017.7943610).
- Handley, H.A.H., Khallouli, W., Huang, J., Edmonson, W. & Kibret, N.** 2021 Maintaining the consistency of SysML model exports to XML metadata interchange (XMI). In *2021 IEEE International Systems Conference (SysCon), 15.04.2021–15.05.2021, Vancouver, BC, Canada*, pp. 1–8. IEEE; doi:[10.1109/SysCon48628.2021.9447105](https://doi.org/10.1109/SysCon48628.2021.9447105).
- Hecht, M., Chuidian, A., Tanaka, T. & Raymond, R.** 2020 Automated generation of FMEAs using SysML for reliability, safety, and cybersecurity. In *2020 Annual Reliability and Maintainability Symposium (RAMS), 27.01.2020–30.01.2020, Palm Springs, CA, USA*, pp. 1–7. IEEE; doi:[10.1109/RAMS48030.2020.9153708](https://doi.org/10.1109/RAMS48030.2020.9153708).
- Hecht, M., Dimpfl, E. & Pinchak, J.** 2014 Automated generation of failure modes and effects analysis from SysML models In *2014 IEEE International Symposium on Software Reliability Engineering Workshops, 03.11.2014–06.11.2014, Naples, Italy*, pp. 62–65. IEEE; doi:[10.1109/ISSREW.2014.117](https://doi.org/10.1109/ISSREW.2014.117).
- Henderson, K., McDermott, T., van Aken, E. & Salado, A.** 2023 Towards developing metrics to evaluate digital engineering. *Systems Engineering* **26** (1), 3–31; doi:[10.1002/sys.21640](https://doi.org/10.1002/sys.21640).
- Henderson, K. & Salado, A.** 2021 Value and benefits of model-based systems engineering (MBSE): evidence from the literature. *Systems Engineering* **24** (1), 51–66; doi:[10.1002/sys.21566](https://doi.org/10.1002/sys.21566).
- Herrington, J.** 2003 *Code Generation in Action*. Manning Publications Co.
- Hick, H., Bajzek, M. & Faustmann, C.** 2019 Definition of a system model for model-based development. *Applied Sciences* **1** (9), 1074; doi:[10.1007/s42452-019-1069-0](https://doi.org/10.1007/s42452-019-1069-0).
- Hick, H., Sanladerer, S., Trautner, J., Ryan, K., Piguët, J., Wilking, F., Horber, D., Faustmann, C., Kranabittl, P., Kolleger, S., Bajzek, M., Schleich, B. & Wartzack, S.** 2023 Combining system models and CAD for change scenario management. *INCOSE International Symposium* **33**(1), 117–132; doi:[10.1002/iis2.13012](https://doi.org/10.1002/iis2.13012).

- Honour, E.C.** 2013 *Systems Engineering Return on Investment*. Dissertation, Defence and Systems Institute, University of South Australia.
- Horber, D., Goetz, S., Schleich, B. & Wartzack, S.** 2022a A model-based approach for integrated variation management In *Volume 2B: Advanced Manufacturing*, 30.10.2022–03.11.2022. American Society of Mechanical Engineers; doi:[10.1115/IMECE2022-90956](https://doi.org/10.1115/IMECE2022-90956).
- Horber, D., Li, J., Grauberger, P., Schleich, B., Matthiesen, S. & Wartzack, S.** 2022b A model-based approach for early robustness evaluation – combination of contact and channel approach with tolerance graphs in SysML. In *DS 119: Proceedings of the 33rd Symposium Design for X (DFX2022)*, 22 and 23 September. The Design Society; doi: [10.35199/dfx2022.18](https://doi.org/10.35199/dfx2022.18).
- Horber, D., Wilking, F., Schleich, B. & Wartzack, S.** 2021 Durchgängige Systemmodellierung - integration von Zielen in technische Systemmodelle im Kontext des Anforderungsmanagements. In *Stuttgarter Symposium für Produktentwicklung SSP 2021*, 20.05.2021 (ed. H. Binz, B. Bertsche, D. Spath & D. Roth). Universität Stuttgart.
- ISO/IEC/IEEE 15288** 2015 *Systems and Software Engineering - System Life Cycle Processes*, ISO, IEC, IEEE.
- Kapos, G.-D., Tsadimas, A., Kotronis, C., Dalakas, V., Nikolaidou, M. & Anagnostopoulos, D.** 2021 A declarative approach for transforming SysML models to executable simulation models. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51** (6), 3330–3345; doi:[10.1109/TSMC.2019.2922153](https://doi.org/10.1109/TSMC.2019.2922153).
- Karban, R., Dekens, F. G., Herzig, S., Elaasar, M. & Jankevičius, N.** 2016 Creating system engineering products with executable models in a model-based engineering environment. In *Modeling, Systems Engineering, and Project Management for Astronomy VI, Sunday 26 June 2016* (ed. G. Z. Angeli & P. Dierickx). SPIE; doi:[10.1117/12.2232785](https://doi.org/10.1117/12.2232785).
- Kerzhner, A. & Paredis, C.** 2011 Combining SysML and model transformations to support systems engineering analysis. In *338 kB / Electronic Communications of the EASST*, Vol. 42. Multi-Paradigm Modeling; doi:[10.14279/tuj.eceasst.42.613.630](https://doi.org/10.14279/tuj.eceasst.42.613.630).
- Khandoker, A., Sint, S., Gessl, G., Zeman, K., Jungreitmayr, F., Wahl, H., Wenigwieser, A. & Kretschmer, R.** 2022 Towards a logical framework for ideal MBSE tool selection based on discipline specific requirements. *Journal of Systems and Software* **189**, 111306; doi:[10.1016/j.jss.2022.111306](https://doi.org/10.1016/j.jss.2022.111306).
- Kruse, B. & Blackburn, M.** 2019 Collaborating with OpenMBEE as an authoritative source of truth environment. *Procedia Computer Science* **153**, 277–284; doi:[10.1016/j.procs.2019.05.080](https://doi.org/10.1016/j.procs.2019.05.080).
- Liu, Y., Irudayaraj, P., Zhou, F., Jiao, R.J. & Goodman, J.N.** 2014 SysML-based model driven discrete-event simulation. In *Moving integrated product development to service clouds in the global economy: Proceedings of the 21st ISPE Inc. International Conference on Concurrent Engineering, September 8–11, 2014; CE2014, Advances in Transdisciplinary Engineering*, pp. 617–626. IOS Press.
- Ma, J., Wang, G., Lu, J., Vangheluwe, H., Kiritsis, D. & Yan, Y.** 2022 Systematic literature review of MBSE tool-chains. *Applied Sciences* **12** (7), 3431; doi:[10.3390/app12073431](https://doi.org/10.3390/app12073431).
- Mahboob, A. & Husung, S.** 2022 A modelling method for describing and facilitating the reuse of SysML models during design process. *Proceedings of the Design Society* **2**, 1925–1934; doi:[10.1017/pds.2022.195](https://doi.org/10.1017/pds.2022.195).
- Mahboob, A., Husung, S., Weber, C., Liebal, A. & Krömker, H.** 2019 The reuse of SysML behaviour models for creating product use cases in virtual reality. *Proceedings of the Design Society: International Conference on Engineering Design* **1** (1), 2021–2030; doi: [10.1017/dsi.2019.208](https://doi.org/10.1017/dsi.2019.208).

- McLellan, J. M., Maier, J. R. A., Fadel, G. M. & Mocko, G. M. 2009 *Generating Design Structure Matrices and Domain Mapping Matrices using SysML*. Paper presented at DSM Conference, 12.–13.10.2009, Greenville, SC, USA.
- Mendieta, R., La Vara, J. L., Llorens, J. & Álvarez-Rodríguez, J. M. 2017 Towards effective SysML model reuse. In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development, 19.02.2017–21.02.2017, Porto, Portugal*, pp. 536–541. SCITEPRESS – Science and Technology Publications; doi: [10.5220/0006267605360541](https://doi.org/10.5220/0006267605360541).
- Mhenni, F., Nguyen, N. & Choley, J.-Y. 2014 Automatic fault tree generation from SysML system models. In *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 08.07.2014–11.07.2014*, pp. 715–720. IEEE; doi:[10.1109/AIM.2014.6878163](https://doi.org/10.1109/AIM.2014.6878163).
- Morkevicius, A. & Jankevicius, N. 2015 An approach: SysML-based automated requirements verification. In *2015 IEEE International Symposium on Systems Engineering (ISSE), 28.09.2015–30.09.2015*, pp. 92–97. IEEE; doi:[10.1109/SysEng.2015.7302739](https://doi.org/10.1109/SysEng.2015.7302739).
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., McGuinness, L. A., Stewart, L. A., Thomas, J., Tricco, A. C., Welch, V. A., Whiting, P. & Moher, D. 2021 The PRISMA 2020 statement: an updated guideline for reporting systematic reviews. *BMJ* **372**, 71; doi:[10.1136/bmj.n71](https://doi.org/10.1136/bmj.n71).
- Palachi, E., Cohen, C. & Takashi, S. 2013 Simulation of cyber physical models using SysML and numerical solvers. In *2013 IEEE International Systems Conference (SysCon), 15.04.2013–18.04.2013*, pp. 671–675. IEEE; doi:[10.1109/SysCon.2013.6549954](https://doi.org/10.1109/SysCon.2013.6549954).
- Paredis, C. J., Bernard, Y., Burkhart, R. M., Koning, H.-P., Friedenthal, S., Fritzson, P., Rouquette, N. F. & Schamai, W. 2010 An overview of the SysML-Modelica transformation specification. *INCOSE International Symposium* **20** (1), 709–722; doi: [10.1002/j.2334-5837.2010.tb01099.x](https://doi.org/10.1002/j.2334-5837.2010.tb01099.x).
- Saqui-Sannes, P. D., Vingerhoeds, R. A., Garion, C. & Thirioux, X. 2022 A taxonomy of MBSE approaches by languages, tools and methods. *IEEE Access* **10**, 120936–120950; doi:[10.1109/ACCESS.2022.3222387](https://doi.org/10.1109/ACCESS.2022.3222387).
- Schleich, B., Anwer, N., Mathieu, L. & Wartzack, S. 2017 Shaping the digital twin for design and production engineering. *CIRP Annals* **66** (1), 141–144; doi:[10.1016/j.cirp.2017.04.040](https://doi.org/10.1016/j.cirp.2017.04.040).
- Schleich, B., Dittrich, M.-A., Clausmeyer, T., Damgrave, R., Erkoyuncu, J. A., Haefner, B., Lange, J. D., Plakhotnik, D., Scheidel, W. & Wuest, T. 2019 Shifting value stream patterns along the product lifecycle with digital twins. *Procedia CIRP* **86**, 3–11; doi: [10.1016/j.procir.2020.01.049](https://doi.org/10.1016/j.procir.2020.01.049).
- Schumacher, T. & Inkermann, D. 2022 Heterogeneous models to support interdisciplinary engineering – mapping model elements of SysML and CAD. *Procedia CIRP* **109**, 653–658; doi:[10.1016/j.procir.2022.05.309](https://doi.org/10.1016/j.procir.2022.05.309).
- Schwede, L.-N., Hanna, M., Wortmann, N. & Krause, D. 2019 Consistent modelling of the impact model of modular product structures with linking boundary conditions in SysML. *Proceedings of the Design Society: International Conference on Engineering Design* **1** (1), 3601–3610; doi: [10.1017/dsi.2019.367](https://doi.org/10.1017/dsi.2019.367).
- Schwede, L.-N., Winter, M., Lödding, H. & Krause, D. 2020 Darstellung des Zusammenhangs von Produktarchitektur- und Produktionssystemgestaltung in SysML. In *Proceedings of the 31st Symposium Design for X (DFX2020), 16–17 December 2020*, pp. 41–50. The Design Society; doi:[10.35199/dfx2020.5](https://doi.org/10.35199/dfx2020.5).

- Stark, R., Anderl, R., Thoben, K.-D. & Wartzack, S.** 2020 WiGeP-Positionspapier: “Digitaler Zwilling”. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* **115**, 47–50; doi:[10.3139/104.112311](https://doi.org/10.3139/104.112311).
- Umeda, H., Takatsuki, S., Kobayashi, T., Ueda, Y., Wada, A., Komatsu, Y., Ishihama, N. & Iwata, T.** 2022 System modeling process with SysML assuming the use of models. In *2022 IEEE International Symposium on Systems Engineering (ISSE)*, 24.10.2022–26.10.2022, pp. 1–8. IEEE; doi:[10.1109/ISSE54508.2022.10005553](https://doi.org/10.1109/ISSE54508.2022.10005553).
- VDI/VDE Society Measurement and Automatic Control (GMA) 2015 Status Report: Reference Architecture Model Industrie 4.0 (RAMI 4.0)**, online document (downloadable on February 9th, 2023) [https://www.zvei.org/fileadmin/user\\_upload/Presse\\_und\\_Medien/Publikationen/2016/januar/GMA\\_Status\\_Report\\_Reference\\_Architecture\\_Model\\_Industrie\\_4.0\\_RAMI\\_4.0/\\_GMA-Status-Report-RAMI-40-July-2015.pdf](https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report_Reference_Architecture_Model_Industrie_4.0_RAMI_4.0/_GMA-Status-Report-RAMI-40-July-2015.pdf).
- Vogel-Heuser, B., Schuetz, D., Frank, T. & Legat, C.** 2014 Model-driven engineering of manufacturing automation software projects – A SysML-based approach. *Mechatronics* **24** (7), 883–897; doi:[10.1016/j.mechatronics.2014.05.003](https://doi.org/10.1016/j.mechatronics.2014.05.003).
- Voirin, J.-L., Bonnet, S., Normand, V. & Exertier, D.** 2016 Model-driven IVV management with Arcadia and Capella. In *Complex Systems Design & Management: Proceedings of the Sixth International Conference on Complex Systems Design & Management, CSD&M 2015* (ed. G. Auvray, J.-C. Bocquet, E. Bonjour and D. Krob), pp. 83–94. Springer.
- Wang, Y., Steinbach, T., Klein, J. & Anderl, R.** 2021 Integration of model based system engineering into the digital twin concept. *Procedia CIRP* **100**, 19–24; doi:[10.1016/j.procir.2021.05.003](https://doi.org/10.1016/j.procir.2021.05.003).
- Weikiens, T.** 2019 *Systems engineering mit SysML/UML: Anforderungen, Analyse, Architektur*, 3. Überarbeitete und aktualisierte Auflage, korrigierter Nachdruck, dpunkt.verlag.
- Wilkig, F., Behringer, M., Fett, M., Goetz, S., Kirchner, E. & Wartzack, S.** 2023 Concept of a modular and system model driven digital twin for engineering education. In *2023 18th Annual System of Systems Engineering Conference (SoSe)*, 14.06.2023–16.06.2023, pp. 1–7. IEEE; doi:[10.1109/SoSE59841.2023.10178517](https://doi.org/10.1109/SoSE59841.2023.10178517).
- Wilkig, F., Sauer, C., Schleich, B. & Wartzack, S.** 2022a Integrating machine learning in digital twins by utilizing SysML system models. In *2022 17th Annual System of Systems Engineering Conference (SOSE)*, 07.06.2022–11.06.2022, pp. 297–302. IEEE; doi:[10.1109/SOSE55472.2022.9812700](https://doi.org/10.1109/SOSE55472.2022.9812700).
- Wilkig, F., Sauer, C., Schleich, B. & Wartzack, S.** 2022b SysML 4 digital twins – utilization of system models for the design and operation of digital twins. *Proceedings of the Design Society 2*, 1815–1824; doi:[10.1017/pds.2022.184](https://doi.org/10.1017/pds.2022.184).
- Wilkig, F., Schleich, B. & Wartzack, S.** 2020 MBSE along the value chain – an approach for the compensation of additional effort. In *2020 IEEE 15th International Conference of System of Systems Engineering (SoSE)*, pp. 61–66. IEEE; doi:[10.1109/SoSE50414.2020.9130497](https://doi.org/10.1109/SoSE50414.2020.9130497).
- Wolny, S., Mazak, A., Carpella, C., Geist, V. & Wimmer, M.** 2020 Thirteen years of SysML: a systematic mapping study. *Software and Systems Modeling* **19** (1), 111–169; doi:[10.1007/s10270-019-00735-y](https://doi.org/10.1007/s10270-019-00735-y).
- Zhang, Y., Hoepfner, G., Berroth, J., Pasch, G. & Jacobs, G.** 2021 Towards Holistic System Models Including Domain-Specific Simulation Models Based on SysML. *Systems* **9** (4), 76; doi:[10.3390/systems9040076](https://doi.org/10.3390/systems9040076).
- Zerwas, T., Jacobs, G., Kowalski, J., Husung, S., Gerhard, D., Rumpe, B., Zeman, K., Vafaie, S., König, F. & Höpfner, G.** 2022 Model signatures for the integration of simulation models into system models. *Systems* **10** (6), 199; doi:[10.3390/systems10060199](https://doi.org/10.3390/systems10060199).