

Article

A Model-Based Approach for the Methodical Development and Configuration of Modular Product Families

Michael Hanna *, Lea-Nadine Wöller, Florian M. Dambietz and Dieter Krause 

Institute of Product Development and Mechanical Engineering Design, Hamburg University of Technology, Denickestraße 17, 21073 Hamburg, Germany; lea.woeller@tuhh.de (L.-N.W.); florian.dambietz@tuhh.de (F.M.D.); krause@tuhh.de (D.K.)

* Correspondence: michael.hanna@tuhh.de

Abstract: This paper shows how a methodical development and configuration of modular product family concepts and their effects on economic targets can be implemented in SysML. For this purpose, different sources of inconsistency between different methodical tools are highlighted and the need for research is shown. As a solution approach, a methodical framework is presented, which can be used to implement product development methods for the developing of modular product family modeling by means of Model-Based Systems Engineering (MBSE) in the modeling language SysML. By applying the framework, it is shown on the one hand how a product family of vacuum cleaner robots as a simple example can be modularized in a methodical, model-based manner. On the other hand, a configuration system and an impact model of modular product families are connected with the system model and applied to a product family of laser systems as an industrial use case. This made it clear that the framework can be used to model various methodical topics of product family modeling in a consistent manner, to enable higher-level analyses with the use of MBSE tools. This can reduce errors, decrease effort and increase traceability across different methodical tools.

Keywords: model-based systems engineering; methodical development of modular product families; modularization; platform design in industrial context; configuration



Citation: Hanna, M.; Wöller, L.-N.; Dambietz, F.M.; Krause, D. A Model-Based Approach for the Methodical Development and Configuration of Modular Product Families. *Systems* **2023**, *11*, 449. <https://doi.org/10.3390/systems11090449>

Academic Editors: Ed Pohl and Eric Specking

Received: 7 June 2023

Revised: 16 August 2023

Accepted: 21 August 2023

Published: 31 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Individual customer demands force companies to offer a wide range of products on the market. This high external variety leads to a high internal variety, accompanied by high complexity and high costs for companies. One possibility to meet this challenge is the use of modular product architectures or platform designs, as they offer high external variety with low internal variety. Several methods for the development of modular product architectures have already been developed and applied, such as the *Integrated PKT approach for developing modular product families* or the use of Design Structure Matrices [1,2]. The selection of the most suitable method is decisive for the success of a company, since methods have different effects on the target time, costs, quality and flexibility, for which an impact model of modular product families can provide qualitative support [3]. In addition, a clever configuration management of developed modular product families is becoming more and more important, as it allows product variants to be efficiently configured according to individual customer needs [4].

These different areas of methodical product development use different data and information, which are often stored in documents or only implicitly. However, product development is often document-based, which leads to problems like inconsistencies and contradictions. An examination of documents across all life cycle phases, as is also required for the link of an impact model, is more difficult. Also, links to external data, which are required within a configurator, cannot be guaranteed. All in all, the problem is that there is no support in methodical research for consistently mapping and linking the data that

occurs during the methodical development of modular product families [5]. Model-Based Systems Engineering (MBSE) can be a solution, since it is suitable for modeling a consistent and linked system model based on a formalized approach. However, the use of MBSE has not yet been sufficiently researched within methodological product developing research and, in particular, the modeling of product families and modularization has not yet been implemented, and it is unclear what a modeling approach for product developing methods should look like [5].

The aim of this paper is to show how tools and approaches of the MBSE can be applied in methodical research for developing modular product families through a methodical framework. It is applied exemplarily for one method and one application example and, in the outlook, it is shown how further methodical aspects can be connected. A software-supported implementation of a consistent data model using tools and approaches of the MBSE can help here if all relevant information is used within a central system model. Especially, the use of a model-based approach will be investigated, and the overall implementation using the modeling language SysML for the application to modular product family design will be determined. By considering MBSE in the context of methodical product development, it becomes clear how consistency can be improved and modular design can be supported. In order to implement this, a meta model for product data is developed and implemented through a framework, with which a continuous modeling of the different frames becomes possible.

In Figure 1 the paper structure is visualized.

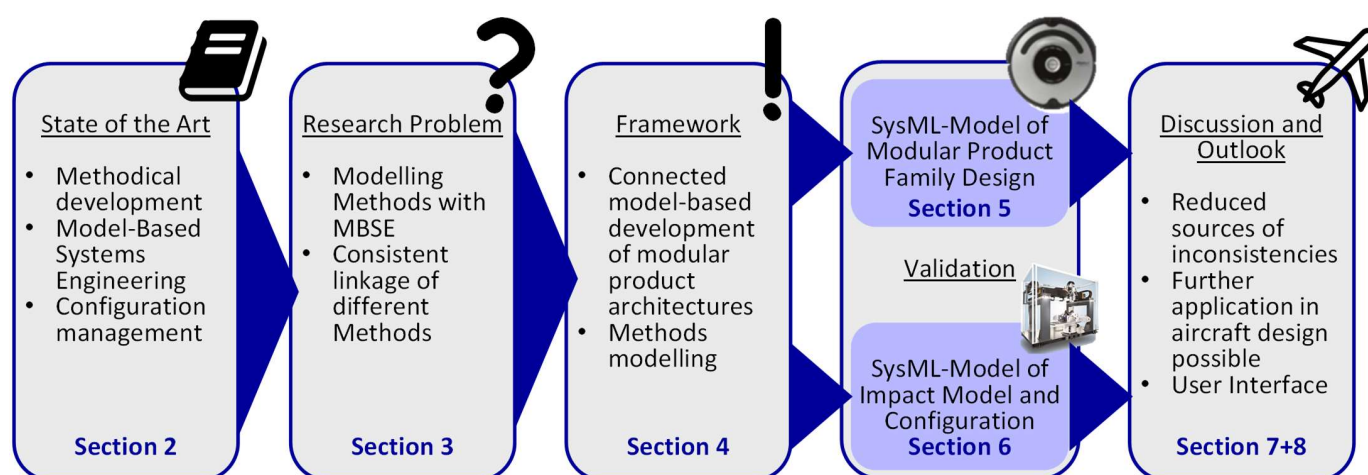


Figure 1. Structure of the paper.

Section 2 explains the state of the art of the relevant topics of methodical development of modular product families, the configuration management and the Model-Based Systems Engineering. In Section 3, the research problem and its approach are analyzed and the research questions are presented. Section 4 shows the research framework for the connected model-based development of modular product architectures to support the methodical product development with MBSE. The resulting product model for the methodical development of modular product families, and its implementation using SysML using the example of a product family of vacuum cleaning robots, is shown in Section 5. Two different application examples are used here to illustrate the use in different industrial sectors. Section 6 shows the extension of the data model by the topics of the impact model and the configurator system, using the example of a laser system. This article concludes with a discussion in Section 7 and the summary in Section 8.

2. State of the Art

In this section, the basics for the methodical development of modular product families, the configuration management, the consistency management and the MBSE are briefly explained. These relevant topics are shown in green in Figure 2.

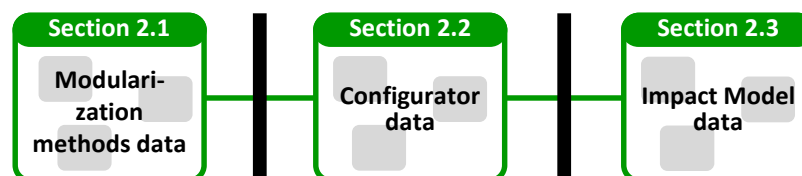


Figure 2. Overview of the addressed methodical areas.

Within the three methodological topic areas (shown in green and explained in detail in Sections 2.1–2.3), the associated unlinked data and documents are indicated in grey. These data are not only unlinked within a methodical subject area, the link to the other areas is also not provided (represented by the black bars). To solve this problem, the basics of MBSE and consistency management are explained in Section 2.4.

2.1. Methodical Development of Modular Product Families

If a high level of variety is desired by the market, this is associated with a growing internal variety, which results in high complexity costs and an immense effort. One solution to this conflict of objectives is modularization or platform design in product development. Modules are understood to be units of observation. Very often, functional units are understood by this. However, modular product architectures hold potential above all when product structuring which is not only based on technical–functional aspects, but also takes into account product-strategic and organizational aspects of different life phases [1]. Modular product architectures are often used to reduce internal variety when external variety is high. In the literature, different methods for the development of modular product families are presented [1]. Min et al. analyze the lowest level of system decomposition for modular system architectures [6]. Design Structure Matrices (DSM) [2] and structural complexity management (MDM) [7] are examples of modularization through technical–functional relationships. According to Stone, module formation is based on the flows between functions [8]. Otto et al. show how to build modularity based on fields. Therefore, the authors work out how the integration of fields supports the designer in product development [9]. Modular Function Deployment [10] is an example of modularization according to product-strategic aspects. The focus here is on the building of modules based on the so-called module drivers. An example of a module driver is the “common unit”. This means that all components, which may not only be contained in one product variant but in several, are grouped together in a module. Thus, this module could be standardized across a product family, which brings complexity-reducing economies of scale. Baylis et al. presented a method for product family platform selection with a balance of commonality and strategic modularity. For this, they use a Pareto front of maximum commonality and strategic modularity [11]. Chen et al. presented a Decision-Based Design approach for complex decision making in product family designs [12].

The Integrated PKT Approach for developing of modular product families combines the two strategies, technical–functional and product-strategic modularization, and contains numerous methodical units to reduce internal variety, while external variety is not compromised [1]. Design for Variety brings a product family closer to the ideal of a variety-oriented product architecture. Such product architecture represents an ideal starting point for the application of modularization strategies. The first step is to analyze the existing external variety of a product family using the Tree of External Variety (TEV). It includes the characteristics of the variant product that are relevant for customers and the product variants offered (see Figure 3, top left). The internal variety can be analyzed using the Product

Family Functional Structure (PFS) and the Module Interface Graph (MIG) for the variety of components and connection sequences (see Figure 3, middle). The components receive their shape from the existing physical product, and are displayed in the color scheme. The variant components are colored grey, while the standard components are colored white. The Variety Allocation Model (VAM) contains this information in a form that shows the relationships between the differentiating characteristics, functions, working principles and components (Figure 3, top right). In the VAM, all variety problems are identified, and solutions for an optimized variety product family are found, which may lead to new product architecture [1].

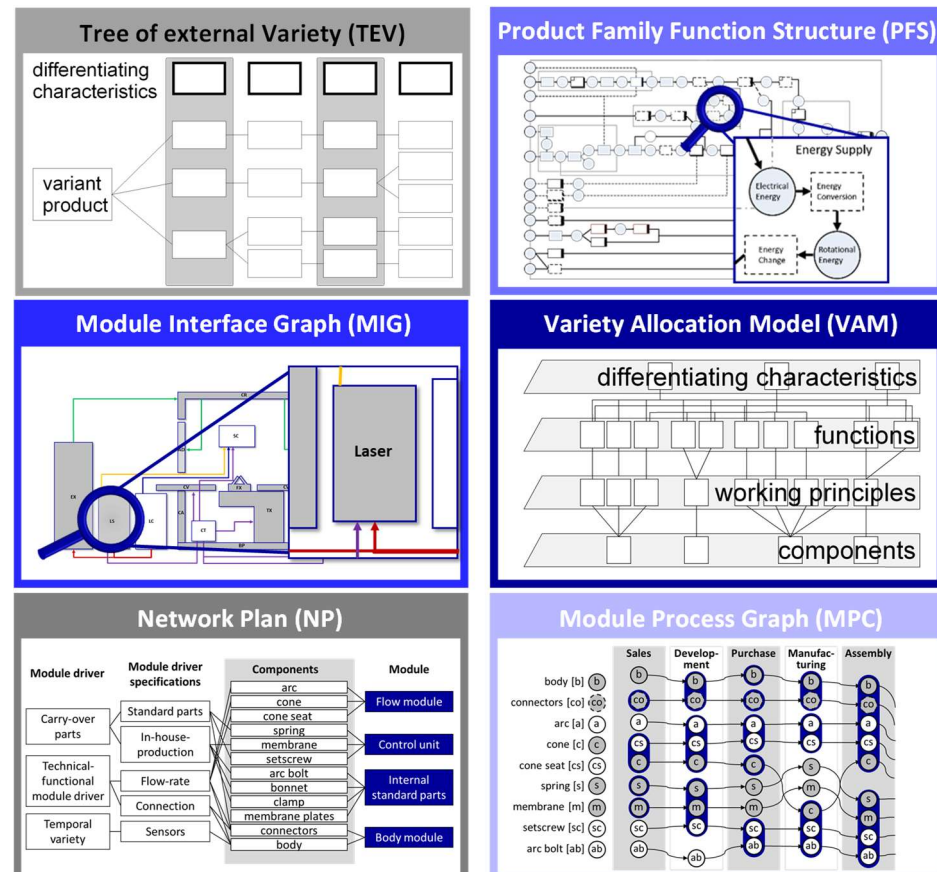


Figure 3. Schematic illustration of Tools of the Integrated PKT Approach for developing modular product families, based on [1,5].

Based on the variant-oriented product architecture through the Design for Variety, the Life Phase Modularization is based on a harmonization of different modularization concepts of individual life phases. For this purpose, a DSM, or the heuristics according to Stone, can be used for the technical–functional modularization, and network plans for all relevant product life phases can be used for the product-strategic modularization on the basis of module drivers (see Figure 3 bottom left). The set of module drivers used here represent a further development of the module drivers according to Erixson. Especially, the characteristics of the individual module drivers, which should explicitly adapt the generic module drivers according to Erixson to the application, play a major role. In the Module Process Chart (MPC) (see Figure 3, bottom right), the different modularizations are combined and harmonized in order to develop alternative concepts that meet the needs of all phases of life [1,10].

The data of the numerous visualizations and analysis tools are continuously developed and supplemented in an iterative process. The data preparation is complex, especially if you consider a whole product family and not only single variants [1].

As already described, there are different methods of modularization in which different data and information are used. In general, development methods can be represented as processes that have an input and an output [13]. Depending on which method is considered, different inputs and resources are needed for the implementation of the method. For example, for the application of Stone's method [8], a functional structure is used as input. Other methods start directly with the product architecture, such as the Life Phase Modularization of the Integrated PKT approach [1].

2.2. Effects of Modular Product Families

With the help of different modularization methods, concepts for the modular design of product families can be generated. Different methods lead to different results, even if the input is the same. To confirm the statement, the same methods were applied for the same input and the differences were shown [14,15]. For the final concept selection, the effects of these concepts represent an important evaluation support. In order to describe the effects of different modular product architectures, the product architectures themselves must first be described.

Modular product architectures can be described according to Salvador [16]. The two main characteristics are common use and combinability. Common use describes the use of a module in several product variants. Combinability allows different product variants to be configured, similar to the modular principle [16].

Hackl has collected the different effects in all life phases of the common use and the combinability of modules in modular product families, which can be found in a broad literature study [3], and then presented them in an *Impact Model of Modular Product Families* (IMF). An example of a connection represented in the impact model is that, in a variant-oriented and modular product architecture, the part numbers can be reduced, whereby the procurement lot sizes increase. By the increase in the procurement lot sizes, the procurement conditions can be improved, which affects the process costs positively [3]. Further effects are listed in the indicated literature. In further research, modularization methods were linked to the impact model of modular product families. Thus, the effects of different modularization methods and their results can be shown. The impact model can be used to compare the effects of different concepts. By showing the economic impacts, a decision for or against a concept is reinforced. The literature-based content of the impact model was validated in various survey-based and interview-based studies, which strengthened this model [1,3,5].

2.3. Configuration Management

In general, configuration management is regarded as the task of using existing (modular) architectures and combining them according to specific underlying requirements. Configuration systems are, therefore, essential for mastering comprehensive product architectures [17]. Liebisch describes the configurator as a tool for optimally matching specific customer requirements with a suitable product variant [18]. Since customers do not purchase customer-relevant features, but decide on variant features based on them, a configurator's task is usually to process all the information between the customer's requirements and the BOM, i.e., to perform a kind of "translation" [19].

Pakkanen et al. shows a brownfield process for modular product family development aimed at product configuration. Therefore, the authors present a method which the aspects of include partitioning logic, a set of modules, interfaces, architecture and configuration knowledge [20].

2.4. Consistency Management and Model-Based Systems Engineering

A system is considered consistent if it does not contain any contradictions [21].

Finkelstein et al. show that overlaps occur when multiple people work on a system [22]. Consistency can be achieved by using clear version management, a common data model and strict access rights [23]. Only if a model respects the formal system, the designer's

preferences and the laws of nature, is it useful [24]. According to Stachkowiak, a model is an abstraction of reality, which has three characteristics [25].

Since product generation development plays an important role in methodological development, version management is also important in this context. It is necessary to identify which version currently represents the correct one [26]. This problem proves to be even more difficult in the further development of modular product families. If a new module is to be added to the existing modular product architecture, compatibility must be checked, and also whether the module already exists. However, consistency testing is more difficult with modular kits than with product architecture without variance, since there are many more dependencies than in the description of the product variants, and it is no longer manageable without software support.

In the context of product development, the models can be divided into those that represent the process and those that represent the products [27].

The MBSE is defined as support for the formalized application and modeling of systems, and is thus part of the Systems Engineering [28].

With MBSE usually being considered as a competitive branch to using PLM systems, the following paragraph examines the differences and commonalities in the use and field of application for these two data management systems in a detailed way. However, it is often problematic that the data formats of the individual subsystems are not always compatible, so that the ontology is not guaranteed [29]. Here, the use of MBSE can help, by using the underlying semantic language SysML. It is based on nine diagrams, five of which are used to create a structure like a block definition diagram (BDD), and four to create a behavior like an activity diagram. The BDD is used to represent the structure of the system, where the system elements are represented as blocks and their connections can be the generalization or directional composition [30].

Until now, a physical item hierarchy was created using the bill of material (BOM), on the basis of which different models from different subsystems were combined. The MBSE now opens up the possibility of efficiently using structural and behavioral information, as well as abstracted links to merge individual models into a coherent meta model [31]. In addition, requirements can be linked to individual product architecture elements, i.e., boundary conditions or links can be provided for components and modules, such as those specified by the MIG in the Integrated PKT approach [1].

Furthermore, MBSE allows the integration of individually adapted, but nevertheless compact, models and methods into the existing data structure. Eigner et al. show that pure PLM systems have strong limitations in terms of openness and free programmability [32].

Therefore, we consider the use of MBSE not as a replacement, but as an extension of already existing PLM systems [4], which is also supported by Sendler and Weilkins, who describe the linking of PLM and SysML as the basis for data consistency, in that not only the geometry and structure of a product are managed, but also its logic and functions, which are the requirements behind this system [33]. Above all, the forward and backward integration of all systems is one of the greatest benefits created by MBSE. On the basis of this object-oriented modeling, logic and function can be mapped in addition to a representation of geometry and structure, as they can be mapped in a classical PLM system, for example. The use of MBSE thus opens up the consistent integration of the ontology of modular product architecture. Thus, validations of the individual models can be carried out one below the other, to ensure a continuous consistency of the data even during versioning or adaptation of the models [4].

Albers et al. show how the model-based systems engineering approach (MBSE) can be used to model the complex real-world problem of vehicle kits. The aim is to investigate the potential in this context, how modular kits and products can be efficiently modeled and, finally, how MBSE can support modular design [26]. Bursac defines a framework for the use of MBSE for the development of building kits in product generation development [34]. Scheerer investigates different modeling methods for the development of modular kits using the example of hybrid powertrains [35]. Stirgwolt shows an integrated approach for

developing modular systems based on SysML ontology [36]. Weilkens describes the modeling of variants using MBSE by variation points, variant elements, variant configurations and variant boundary conditions [37]. Kim et al. show a modeling and evaluation process to analyze the architectural complexity of product families using DSM [38]. Reid et al. show the impact of product design representation on customer judgement [39]. Hvam et al. show a procedure for building product models that contain knowledge and information associated with products in different phases of their life cycles [40]. Gräßler et al. present a methodology to support engineers in developing certification-compliant effect-chain models [41]. For the support of variant management in the automotive industry, an MBSE based approach with SysML is also presented [42].

3. Research Problems and Research Approach

In the methodical development of modular product families to reduce internal variety, data and documents of different variants must be considered in different steps of the methods. At the same time, information available in companies on the effects of modular concepts, as well as an effective configuration system, must be considered, and a comprehensive linkage must be ensured. Since this is only insufficiently possible so far, an analysis of possible inconsistencies will be shown here as a basis for the modeling. Different types of consistency can be considered.

Possible sources of inconsistencies in the methodological development of modular product families in this context are shown in Figure 4.

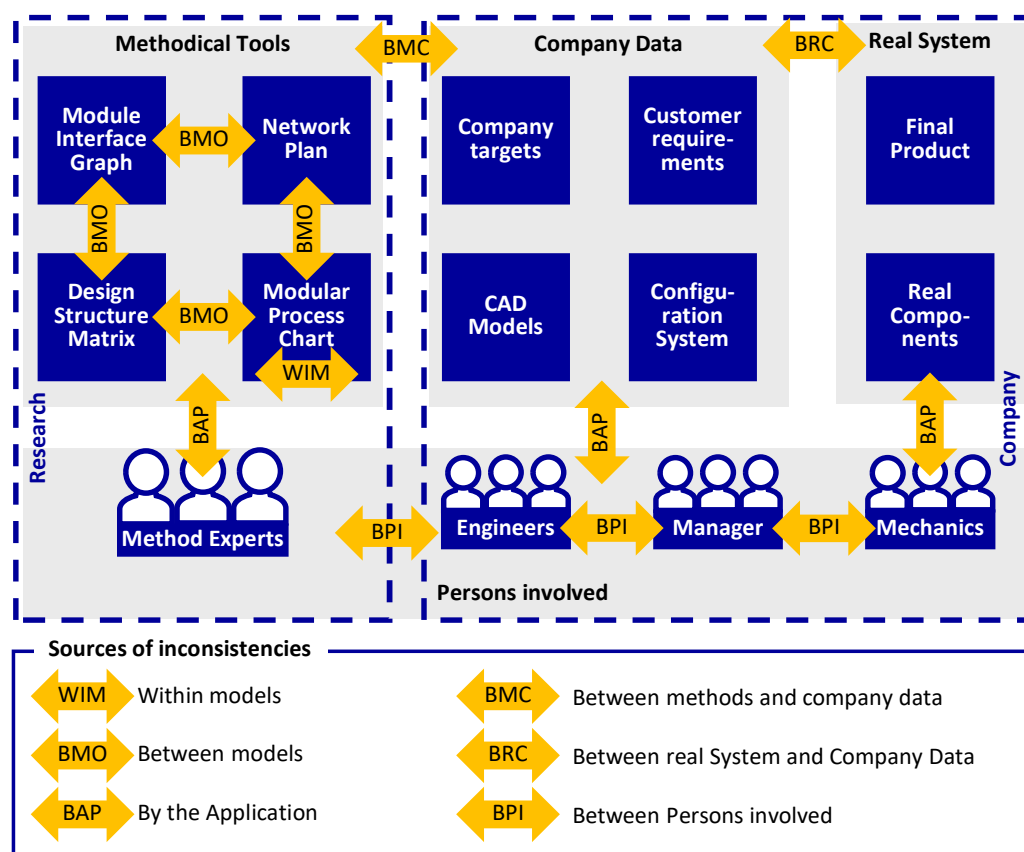


Figure 4. Considered inconsistencies, based on [5].

On the left side, the methodical tools are shown, whereas in the middle, the documents available in the company for the development of real products (right side of Figure 4) are shown. The MIG is shown in the upper left corner, where the components, their position, the type of variance and the flows are contained. In the DSM, modules are formed based on

the couplings between components and thus support a technical–functional modularization. The MPC can be used in combination with the product strategic modularization, in which the components are combined into modules within each life cycle phase from the Network Plans, thus supporting harmonization [1]. Common to these four models is that they all contain components that are used at different points in the methodical procedure. However, there is no adjustment of the component types, which can lead to errors, especially when changes are made. Large product families are difficult to map. In the literature, the limit for the existing document based approach of the Integrated PKT approach is indicated at about 80 components [5].

A configurator system can access the network plan on the one hand and the customer requirements on the other hand, which is why inconsistencies can occur between data on the research side as well as on the company side with linked models. An impact model of modular product families uses, on the one hand, the modularization concepts created by methods and, on the other hand, the effects on time, costs and quality, which is why there should also be a continuous link between these tools.

To illustrate the problem, the application of the document-based Integrated PKT approach is shown here as an example. To determine the number of possible inconsistencies and thus errors, the Gaussian summation formula can be used. For this purpose, the inconsistency index (*IQI*) is introduced here. This tells us how many possible inconsistencies can occur when using this method and, thus, how many errors can occur in the application. The number of the individual elements, *n*, says in how many documents an element occurs.

$$IQI_G = (n - 1) \times \frac{n}{2}$$

So, for example, for the element component, the index can be determined. The component occurs in four documents (MIG; VAM, MPC, network), so the index *IQI* is equal to 6. This number with the real number of components for an application example then results in the real number of inconsistency sources and thus error possibilities. The following formula can be used to calculate the real sources of inconsistency for the application of a method to a product family:

$$IQI_R = Z \times IQI_G$$

where *Z* stands for the number of components actually occurring in the product.

As already described, the consistent management of such complex and diverse data systems and structures is one major task of modern product architecture development. Especially, the management of different data types and the forward and backward consistency is considered the main issue within this area.

One approach to solve these problems is MBSE, since the problems described above can be reduced or even avoided by consistent data storage and consistency checks.

As already described, the concept of MBSE can offer one possible solution to addressing these major challenges of data management and consistency. The literature analysis shows that there are many individual contributions to the topics, but MBSE and modularization are almost not linked at all and, if they are, it is not clear how to move from one to the other and how to link different methodical aspects together. In the use of MBSE, advantages are thus possible, whereby an application to the methodical development of modular product families could solve the challenges described. However, since, as already mentioned above, MBSE is originally intended for the SE, adaptations of the tools of MBSE are necessary here and a methodical approach is needed.

It becomes clear that the relevant research topics of methodical development have not yet been linked, and MBSE was rarely used for the methods for the development of modular product families. Therefore, in this paper, we focus on two specific problems, which are reflected in the following research questions:

1. How can methods such as those used to develop modular product families be modeled and applied in a systematic way in MBSE, so that added value can be achieved?

- How can different methodical aspects be consistently linked with each other in an MBSE model?

To make this possible, the gap between the methodical product development view from the design methods area and the modeling view from the MBSE area should be closed by a feasible framework. In particular, the question of how these different data sets of the three considered areas can be used and linked in order to make efficient use of the corresponding synergies.

4. Research Framework for the Connected Model-Based Development of Modular Products Architecture

The literature analysis in Section 2 has shown that there is no description for how to move from method models to MBSE application. In order to solve the problems described, the associated research procedure is presented here.

Based on the problems analyzed in Section 3, the following requirements for the methodological approach have emerged:

- The procedure should consider and be able to model methods;
- Both static and dynamic models should be modeled, since methods contain both tools and a procedure;
- A link to real product data should be possible;
- A model should be adaptable and extendable with the procedure;
- Further methods and methodical data should be linkable.

For this purpose, a developed framework is presented, in which the steps processed in this paper are shown in Figure 5.

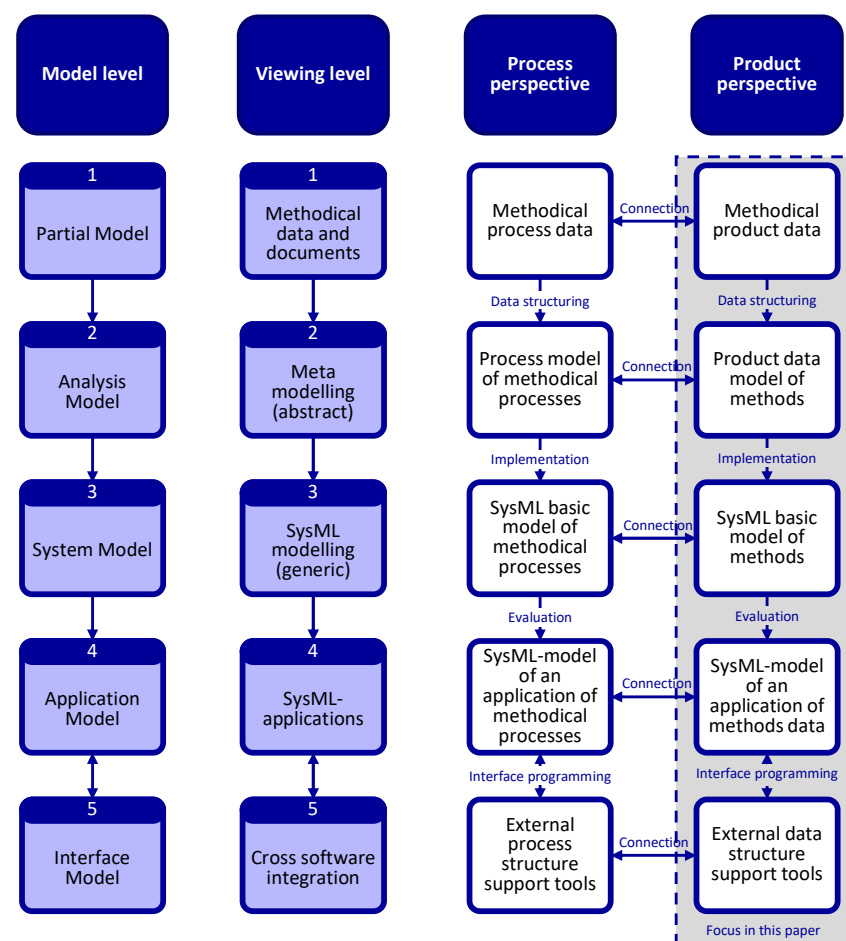


Figure 5. Framework for a connected Model-Based Development of modular Product Architecture.

In Figure 5, different modeling strategies are shown in the rows. The columns show the modeling procedure for the methodical case. The specific coupling or procedure step between individual elements is illustrated with the different arrows. In the general approach, methodical activities and tools are divided into a process and a product view. In this paper, the focus is on product view (highlighted in gray). In the columns, different levels of view for the linked model-based development of modular product family are visible. The first column schematically shows the modeling of a continuous process model, which is not considered in this paper. The process model should be linked to the product data model.

The first level shows the existing methodical data and documents for the product development context. The first step in this approach is the structuring of these methodical data and documents. For this purpose, the available literature is analyzed, correlations are recognized and approaches to the solutions are identified. The insights gained from this analysis help to structure the data. An abstract meta model (level two) is needed, in which the basic data and their interrelationships are analyzed and represented on an abstract level. Next, this meta model can be implemented in SysML, which creates a basic system model in which the individual models and model elements are modeled consistently but also abstractly (level three). For evaluation purposes, this model should then be filled with a use case where the advantages of the SysML model can be used (level four). Further software can be integrated by interface programming, as shown on level five.

The focus of this paper is the consistent modeling, implementation, and evaluation of a product data model for the topics, and methodical development of modular product families, configuration systems and impact models of modular product families. For this purpose, the framework for the product view is detailed for the three areas (see Figure 6).

The basis for this is the product data model of modularization methods, which is presented in Section 5. Its implementation by means of the MBSE is carried out in a SysML basic model, whereas one exemplary use case for a product family of vacuum cleaner robots is shown in Section 5. The meta model of product data for the configuration system and the impact model are linked to the meta model of the modularization methods, and are described in Section 6. Their SysML Basic Models and their application for a product family of laser systems are also described in Section 6. Further product data can be linked in the future (see Figure 6, right).

The first step is the analysis of the existing literature in order to identify the corresponding modularization methods and their data correlations (colored in green). This is necessary for data structuring, and can be represented with an abstract data model on the meta model level. Here, the Integrated PKT approach (explained in Section 2) is used. On this basis, the modularization methods can be implemented using SysML-based software like the Cameo Systems Modeler (Cameo Systems Modeler from No Magic, Version 19.0) with SysML version 1.6. To understand the methods, the vacuum cleaning robot can be used as a simple explanation example. The resulting findings help to support the discussion and form the basis for iterative further development of the model. The first step of the analysis is especially relevant by comparing the model-based and the PowerPoint-based implementation of the methods. Furthermore, the abstract data model of the second step can be supplemented by the findings of the SysML-based implementation and the improved analysis.

The transfer of SysML models into industrial practice is difficult. Nevertheless, researchers agree that with the support of MBSE, problems in companies, especially challenges posed by industry 4.0, can be solved. [43]. The integration of SysML models into industry has already been studied by other researchers [44]. Among other things, design thinking seems to make sense as an option for knowledge transfer and thus enable a user-centered MBSE [45].

In Sections 5 and 6, the approach of the framework is validated on the basis of two application examples.

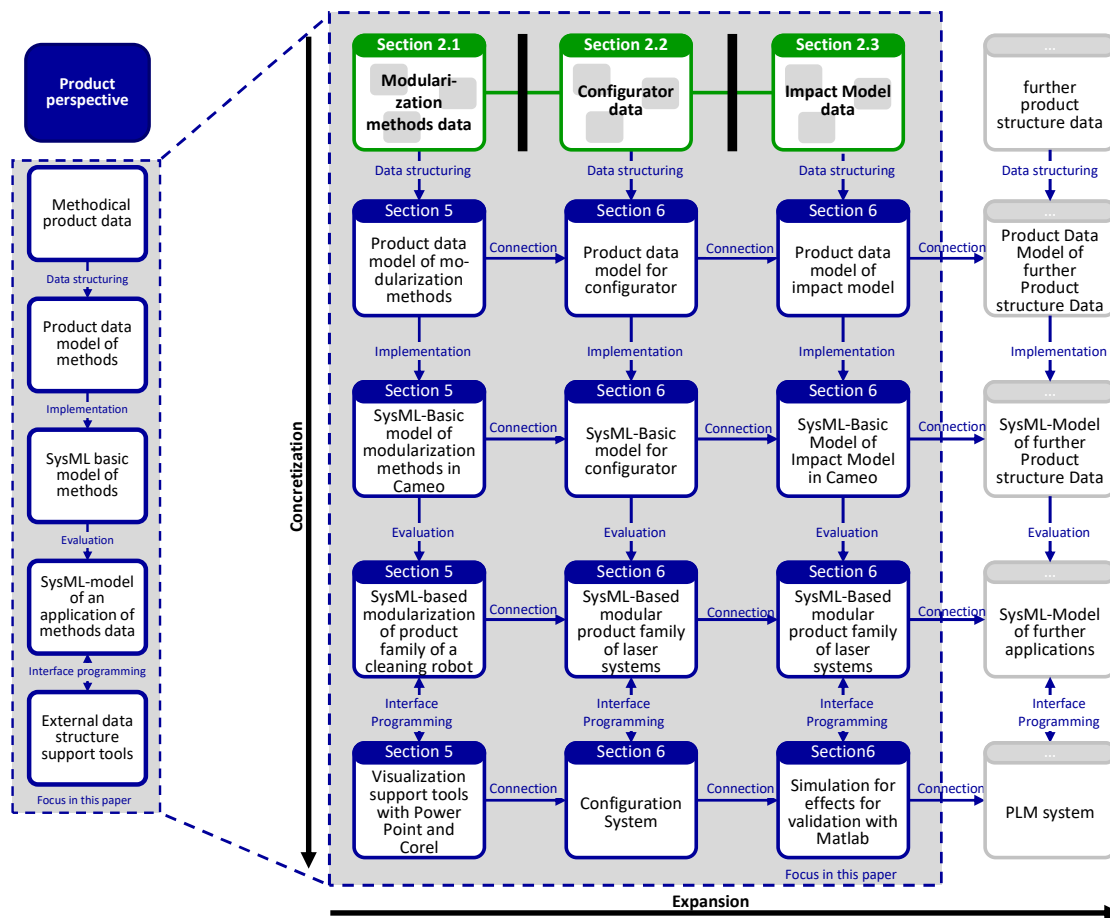


Figure 6. Detailing and proceeding of the framework for consideration of different methodical areas.

5. A Product Model for the Methodical Development of Modular Product Families, and Its Implementation Using MBSE

Within the context of design methods research, validation can be carried out with regard to applicability, usability and usefulness, according to Blessing and Chakrabati [46]. Pedersen et al. [47] and Olewnik et al. [48] also show how validation can look in the field of methodological product development. In this paper, the focus is on validation in terms of applicability. For this purpose, the framework was validated using the method Integrated PKT Approach and two product examples in Sections 5 and 6.

This section shows the meta model of product data for the development method integrated PKT approach as a result of the data structuring and its implementation with SysML.

5.1. Analyzing and Structuring the Data of the Methodical Product Development of Modular Product Families in a Data Model

The analysis of the data correlations motivated in principle that a data model can be used to minimize sources of inconsistency. A consistent data model should be designed to use the data correlations as comprehensively as possible and to consider all aspects relevant for the development of modular product families. Different types of consistency should be taken into account to enable the most effective model-based support of methodical product development. Especially, inconsistencies between single models as well as within a model should be addressed. Due to the often-iterative nature of methodical product development, data modeling must also ensure that changes are as easy and error-free as possible. Redundant information should be identified and eliminated, and networking between different data should be possible. The deposit of a data model is important when the data connections become so extensive that they are difficult to handle. This also includes

the fact that data from different domains are present in the data contexts. An example of this is the use of data models to investigate the effects of modularization of product families. Also, here already different data are used like, for example, product architecture data, components and modules, and also functions. The possibility of interface modeling can be of particular importance here if external models and systems are to be linked, as is necessary for a configurator system.

Here, the methodical development of modular product families of the Integrated PKT approach is used as a basis. The focus here is on the two methodical units, Design for Variety and Life Phases Modularization. Their sub models were examined to determine what types of data and links are available and the different hierarchical levels. The links are not exactly defined in the data model, in order to remain as solution-neutral as possible when implementing the modeling software. Subsequently, the data structure is created, in which the relationships and elements are linked and mapped. Figure 7 shows schematically the data structuring of the analyzed methodical product development data for a product data model. Within the framework of modularization, components can be formed from the assemblies and parts by decompensation, which form the basis of the methodical procedures. These are used in the VAM and MIG in the context of Design for Variety, whereas they are used in DSM, network and MPC for life-phase modularization. The overlapping areas of these sub models are thus the components, since they each exist only once. Changing an element in one partial model results in the same element being changed in another partial model.

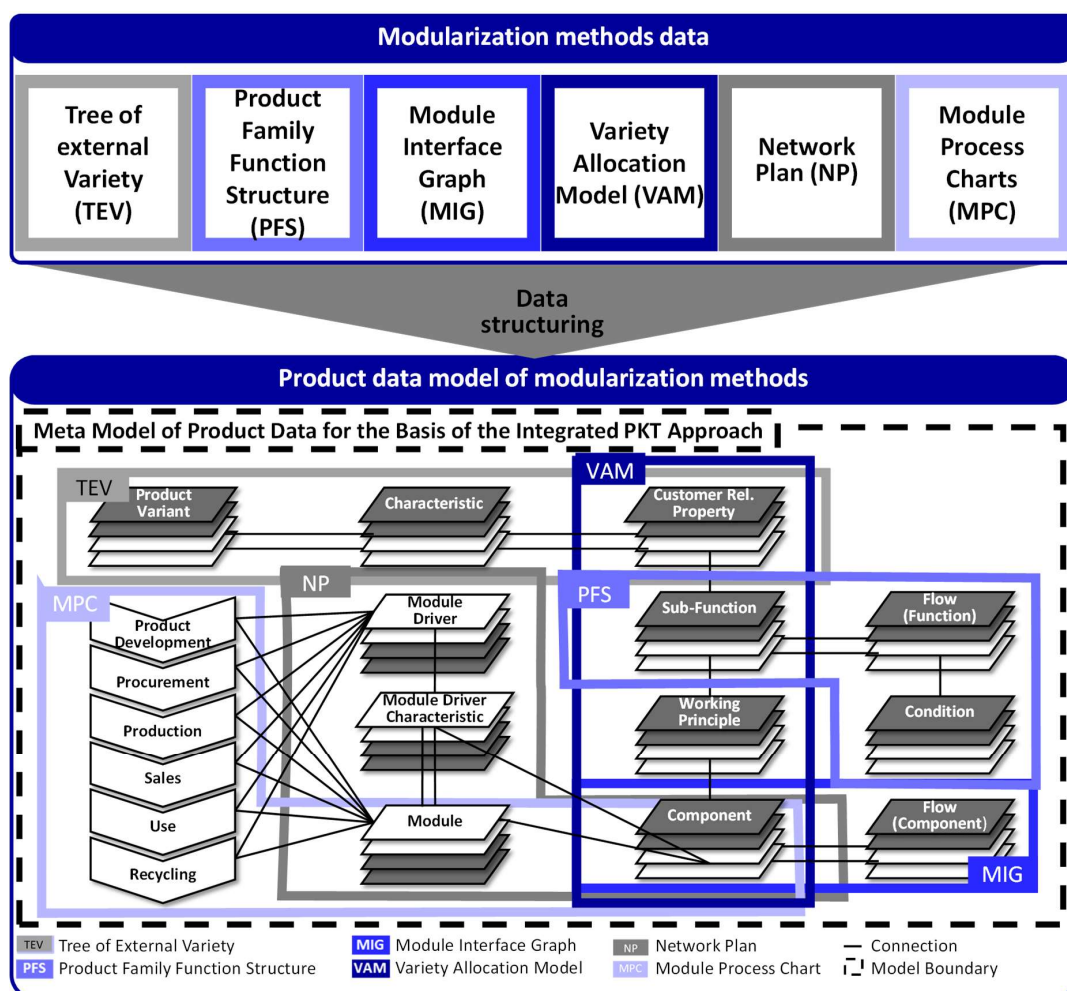


Figure 7. Data structuring of the analyzed data for modularization methods in an abstract data model, based on [5].

Each data element (such as the components) is created only once and can be used from the centrally stored location in multiple partial models. All data elements can appear in different versions during the application of the product development method. In order to represent this, different layers have been selected in the data model [5].

5.2. Implemented Basic Model for Methodical Modularization in SysML

For the methodical product development of modular product families, an implementation of the basic model in SysML is shown here based on the data model for the Integrated PKT approach. The basic model in Figure 8 is intended to serve as a template for the model-based implementation of a concrete use case.

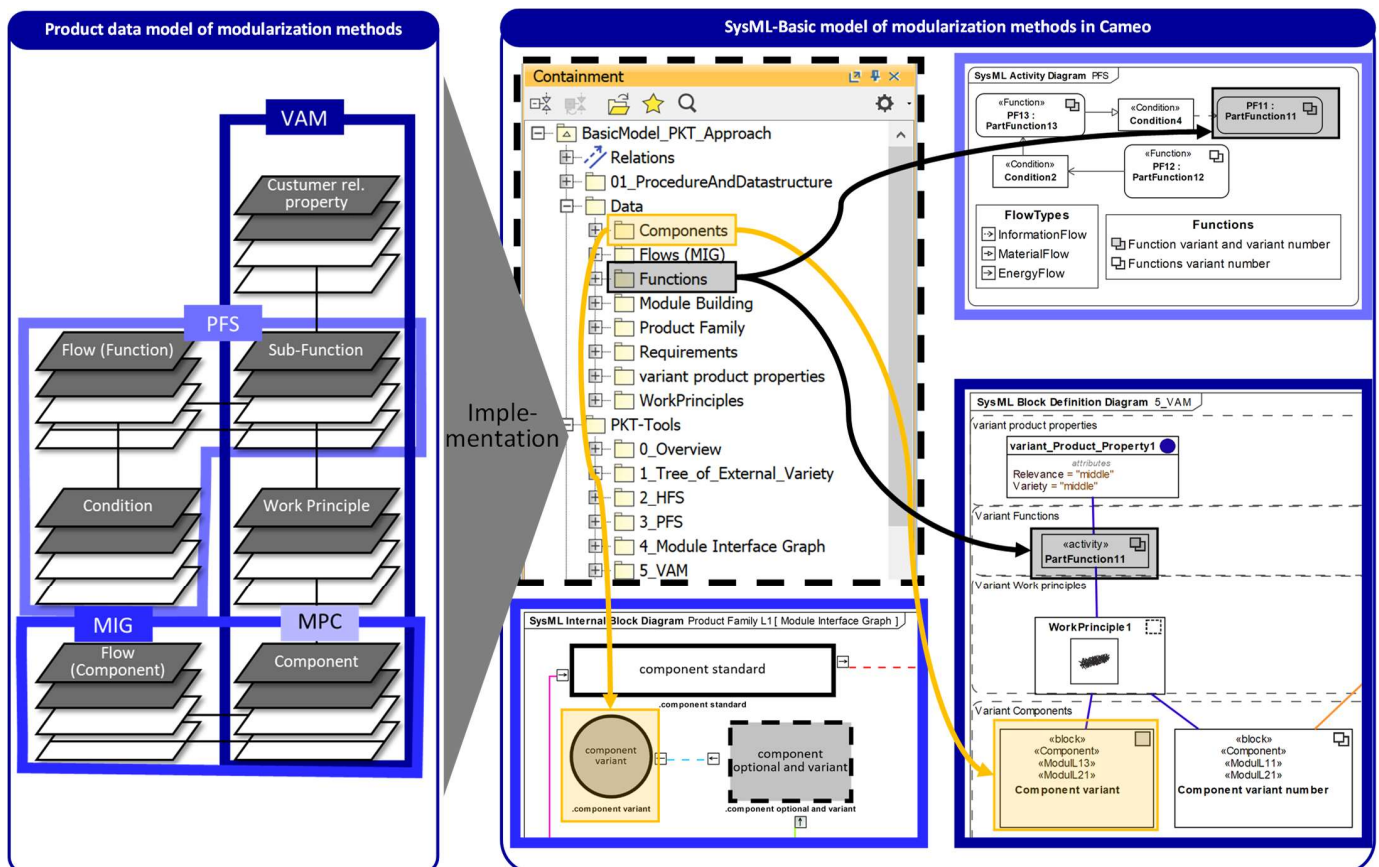


Figure 8. Section of the meta model of product data (left) and its implementation as basic model in SysML (right) for the Design for Variety.

Figure 7 shows a section of tools for variant-oriented product design (VAM, PFS and MIG) of the Integrated PKT approach for developing modular product families. On the right side of Figure 8, the containment tree of the SysML model is shown, on its top level below the data all used model elements are found. These include the components, which are stored centrally here and are used in the different models (MIG, VAM, PFS) (marked orange) or the functions (marked black). This ensures consistency, since changes to a component or function are automatically changed in the models in which they are used. Furthermore, the variance reference is highlighted in color. In addition to the data describing a product architecture (functions and components), other data required for methodical product development (such as requirements and customer-relevant characteristics) are also stored here [5]. For the MIG, an internal block diagram is used, since it can represent different kinds of flows between the elements (in these case, components). The flows can be assigned by the flowports of the individual elements, and the flows can be visualized thereby. Furthermore, the VAM was modeled. Since different data are linked together here, a block

definition diagram is also useful here. In the PFS, the different data types, function and state are connected with flows. In SysML, there is the activity diagram, where the definition of the activities resembles those of the flows. Therefore, an activity diagram is selected here, where the functions are defined as stereotypes and the states are also shown. Since different data are linked together in the VAM, a block definition diagram is also useful here. The tools for life-phase modularization can also be modeled with block definition diagram for the network plans and the MPC.

5.3. Using the Basic Model to Develop a Vacuum Cleaner Robot Product Family

To fill this basic model in SysML with data from a use case and to develop a modular product family, a product family of vacuum cleaner robots was used. The product family consists of four variants and includes a total of 29 components.

Figure 9 shows a part of the implementation using the Cameo System Modeler.

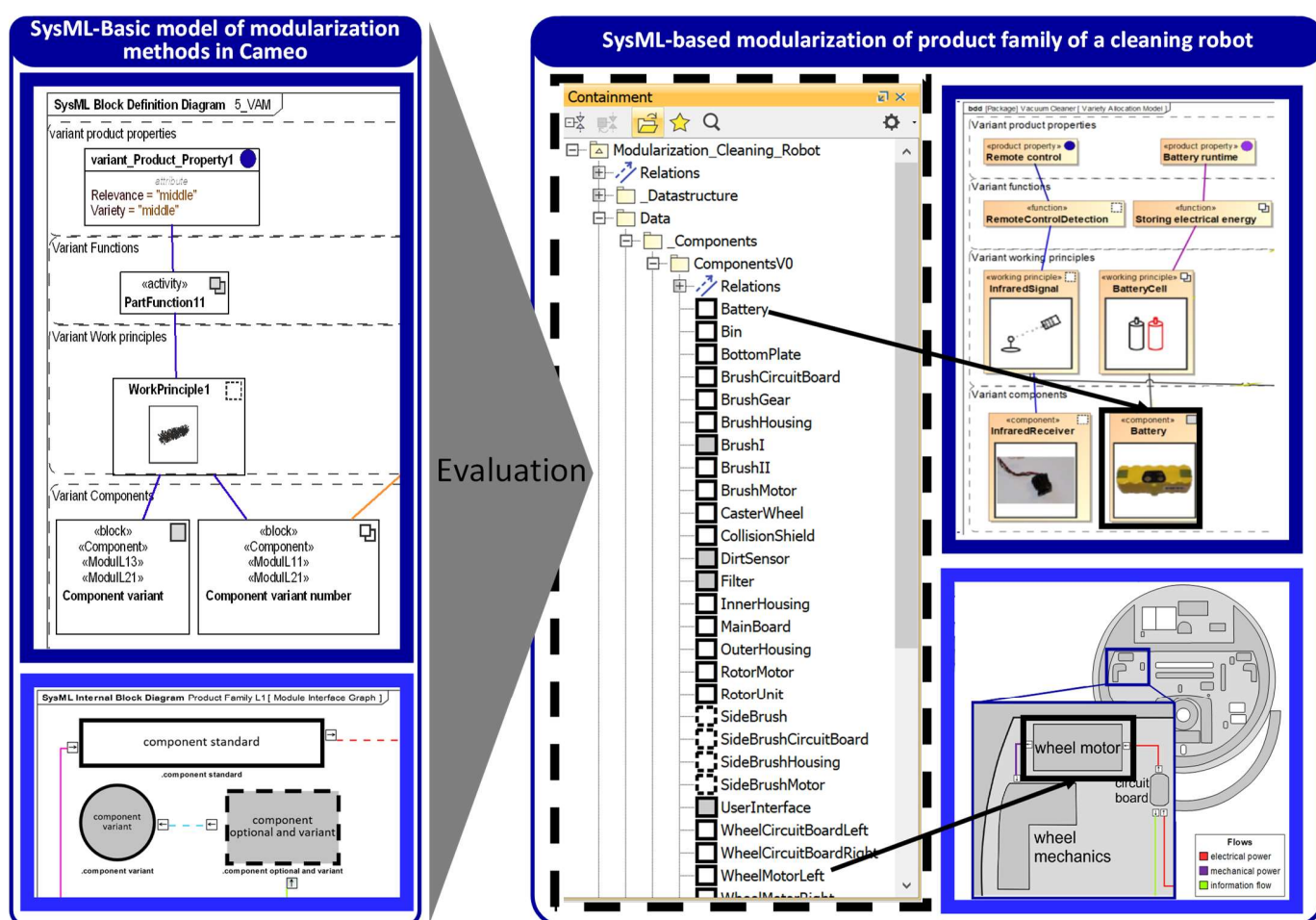


Figure 9. Section of the model-based modular concept for a vacuum cleaner robot product family, based on [5].

The Figure in the middle shows the section of the containment tree that shows a part of the components. Thus, the component wheel motor, which is stored centrally there, is used in the section of the MIG on the right. Simple graphics have been added to the individual components to visualize the rough shape, and the colored flows are connected to the components via ports. The variant components are colored gray, such as the wheel motor. The flows are defined by different stereotypes.

Here, pictures of the vacuum cleaner robot were added to the individual elements and colored flows (such as electrical power, mechanical power and information flow)

are connected to the components. The flows were defined by different stereotypes. For example, the electrical power between the wheel motor and circuit board is shown in red, whereas the mechanical power between wheel motor and wheel mechanics is visualized in purple on the right. Ports are used to illustrate the flow direction and the position of the components. Furthermore, it is possible to insert legends in Cameo. In the lower right corner, you can see the legends for the rivers with their color scheme and the components, including the visualization scheme for the component type [5]. The Figure on the right shows a section of the VAM, where, for example, the components stored centrally in the structure tree are used and are linked to the variants of active components, variant functions and the variants of customer-relevant properties, which are also stored centrally.

As shown, MBSE software like Cameo Systems Modeler is suitable to support the data correctness of methodical product development. Since visualizations can only be created in a limited way using SysML, images are integrated for the visualization, which are created outside the MBSE software. For the visual effect for the MIG, the individual components are drawn in Corel or PPT and then integrated into the data model, which can be understood as the fifth level of the framework.

6. Extension of the Data Model with Data of the Configurator and the Impact Model and SysML Application for Configuration and Its Effects

In this section, it is shown how the developed model for methods for the development of modular product families can be supplemented by means of an impact model of modular product families and a configurator system. In addition, it shows the external connection of customer data to the data model for the use of the configurator system for the configuration of product variants for laser systems. The modularization characteristics and their effects are likewise used by an external connection to the business goals of the laser equipment company for the use by means of the data model.

6.1. Connection of the Elements of the Impact Model and Configuration System to the Existing Data Model

As a basis for the extension, the data model of the methodical product development from Figure 7 is used. Their connection can be seen in Figure 10.

The relevant section of the data model is shown in the middle. The impact model ties in with the data of the life phase modularization (orange connections). In it, the characteristics of modularization are linked to the properties of modularization via the effects with economic targets. Thereby, the characteristics of the modularization are linked to the modules and components. The effects, in turn, are linked to the module drivers [5]. The effects are linked to the life phases, in the Figure Sales, Purchase, Development, Production, Product and After Sales.

The configuration system also uses the data of the Life Phases Modularization of the Integrated PKT Approach, which is made clear by the black connection. The data of the components, modules, and their module driver characteristics are relevant here. In addition, the customer-relevant properties must be taken into account. Since a configuration system is classically the interface to the customer, the sales life cycle phase must be considered as the most important one. On basis of the above, a data model on basis of an adapted sales network diagram (CND = Configuration Network Diagram (CND)) appears to be meaningful (see Section 2).

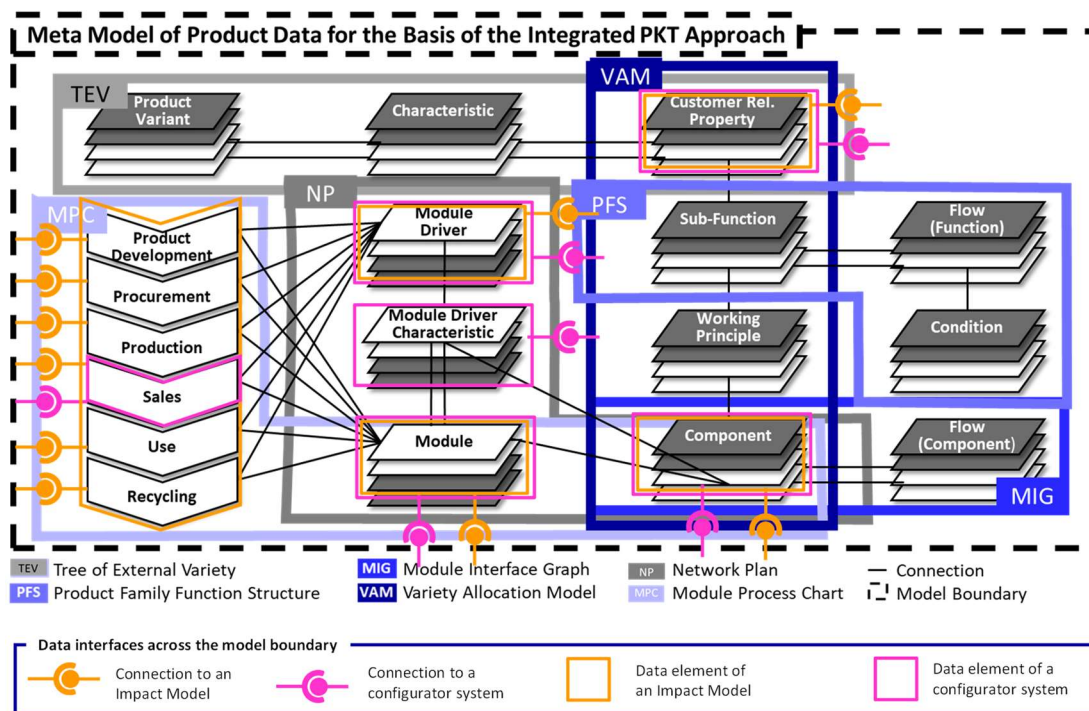


Figure 10. Expansion of the data model of the modularization methods to include additional data interfaces.

6.2. Generic Data Modeling for Configuration Systems

The following section describes the use of MBSE as backend for configuration systems of modular product families. In this context, it should be particularly emphasized that MBSE is not used as an individual method with a specific model, but is used as a dynamic data basis via appropriately developed interfaces.

In order to build on the existing research results in the area of database and interfaces for an MBSE-based configuration system, the MBSE environment is first described as a configuration backend.

Modular, dynamic product configurators are described as absolutely necessary for the generation and maintenance of increasingly comprehensive, multi-variant product architectures [49]. Thus, explicit customer requirements can be optimally mapped to an available product variant. However, it should be noted that it is usually not possible to cover all customer requirements [18]. With customers of complex product systems usually not perceiving their requirements as customer-relevant properties, but as specific needs, and therefore expressing these requirements from the customer perspective, a configurator generally has the task of processing all the information between customer requirements and the bill of materials into the company perspective, which is a kind of “translation”. Two interacting systems are used: the front-end, which is the user interface, and the back-end, which provides the product architecture to the customer.

The MBSE now opens up the possibility of efficiently using structural and behavioral information, as well as extracted links, in order to combine individual models into a coherent meta model [4]. In addition, requirements can be linked to individual product architecture elements, i.e., boundary conditions or links can be provided for components and modules, such as those specified by the MIG in the Integrated PKT approach [5]. In order to link defined modules with specific customer-relevant properties, the database must be capable of recording requirement links and customer-relevant properties as qualitative data and their quantitative characteristics in a structured and processable manner. The necessity of the use of MBSE becomes even clearer when considering the most important requirements for configuration systems: on the basis of a complete information system, it

is important to enable consistent product configuration and the possibility of plausibility checks. This can only be illustrated by using a suitable modeling language, as provided by the use of a suitable MBSE tool [4]. This means that not only are the geometry and structure of a product managed, but also its logic and function. On the basis of these defined requirements, it is necessary to select a suitable data structure that can simultaneously represent customer-relevant properties, modules, components and their individual links.

The configuration algorithm for this was developed generically in such a way that, by accessing exported linkage matrices from the MBSE environment, these dependencies are derived and applied dynamically during the configuration process. The configuration-relevant data, such as material costs of the individual components, their delivery time or the respective design effort, are assigned to the corresponding components after the required module variants have been determined. Since this data corresponds to the individual parameters of the components, which, when added together, result in a total value when a module or product variant is put together, this data is attributed in the form of a configuration value vector (CVV). This allocation, as well as the output of the relevant data via the interface, is performed by using a recursively applied parameter diagram in the MBSE environment. This so-called RollupPattern enables a fast, efficient, consistent and generic data assignment and data extraction [4].

Figure 11 shows the generic structure of a Network Plan with the link to the customer relevant properties and the product model for individually configurable laser welding systems in SysML.

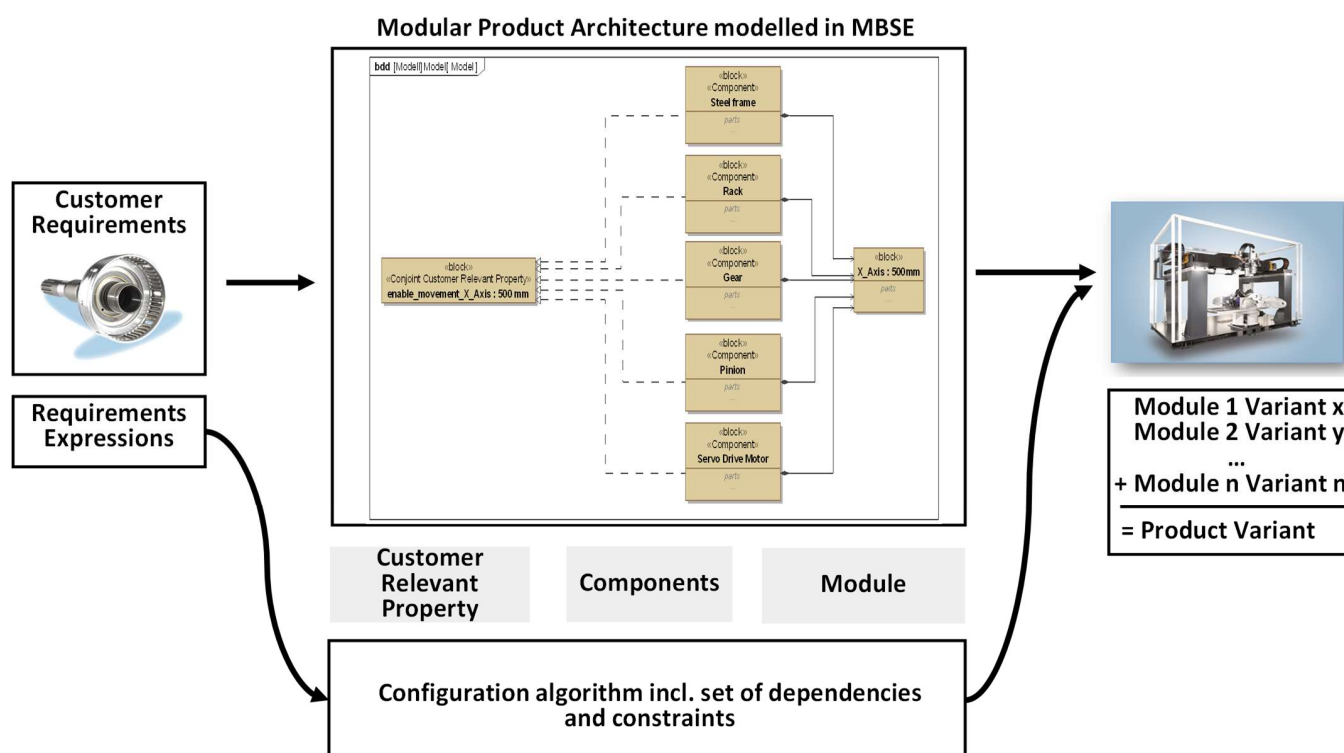


Figure 11. The translation process between customer requirements and corresponding product variant based on the MBSE modelled configuration network diagram, based on [4].

6.3. Data Model of the Impact Model of Modular Product Families

The impact model of modular product families is first analyzed, and then a basic data model is developed based on this to show the relationships between the individual model elements [48]. This data model is the basis for the modeling of the impact model of modular product families in SysML.

6.4. Configurability and Its Effects in SysML

The impact model of modular product families is a static model. Nevertheless, it has been found in the past that some effects of the impact model of modular product families are more likely to occur under certain company boundary conditions than others. For example, a common parts strategy for a mass producer can have a significant impact on procurement costs [3]. This, and other information, can be stored in the model-based impact model of modular product families. This creates a database on the effects of modular product architectures which can be adapted and is therefore no longer static [5].

Based on the basic model for the impact model of modular product families, its data connections were modeled in Cameo using SysML. By modeling the data relationships in SysML, different views of the relationships are made possible. Additional information can be stored, such as applications of the generic Impact model of modular product families to specific use cases.

To illustrate the dynamics of the model, it was adapted to the previously presented application example: configuration of laser systems. In order to ensure configurability, the modularization property combinability is of great importance. This is made possible by the characteristics of interface standardization and decoupling. Combinability is an input variable in the impact model of modular product families. In addition to the assumption that combinability plays a decisive role in the selected application example, validations were also carried out in the form of surveys with a manufacturer of laser systems in special machine construction [4]. Developers were asked which effects of the generic impact model of modular product families result from configurability in the company. This knowledge was stored in the SysML model in order to strengthen the informative value of the model in the future.

All the effects that can be achieved through combinability can now be shown in a relation map (Figure 12). The Figure shows, so to speak, an excerpt with an impact chain of the filtered impact model.

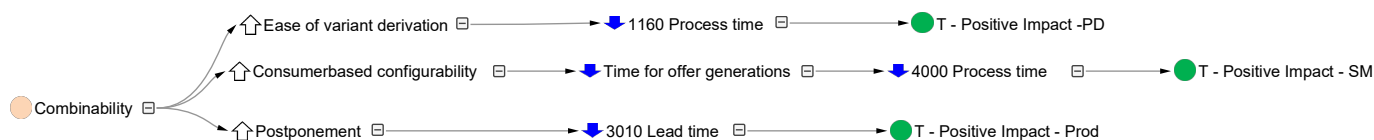


Figure 12. Excerpt with an impact chain of the filtered impact model of modular product families.

7. Discussion

The comparison of the modeled methodical support, the impact model and the configurator system show that an MBSE-based implementation is helpful for a networked, methodical and consistent consideration of modular product families. The three research fields of modularization methods, effects of modular product architecture concepts and configurator systems can thus be related to each other, and their dependencies can be identified and utilized. Due to the systematic approach, starting with the analysis of data for product structuring, meta data modeling, SysML implementation and the application for modular product families, all essential data can be considered comprehensively. Thereby, the validation regarding applicability has been achieved. In particular, consistent data modeling reduces errors and keeps the effort for changes low.

In this way, the identified sources of inconsistencies could be reduced. With regard to the components, six sources of inconsistencies were identified at the generic level in the previous document-based use of the Integrated PKT approach (see Section 3). For the product family of vacuum cleaner robots with its 29 components, this therefore means a reduction of 174 inconsistency sources occurring in the real application, and thus 174 possible errors according to the formula.

$$IQI_R = Z \times IQI_G = 29 \times 6 = 174$$

where Z stands for the number of components actually occurring in the product.

Especially in methodical product development, many steps are performed agilely in workshops, where a quick and easy change can be supported by the model-based approach. In addition to the data relevant for the design of modular product architecture for product structuring and functional analysis, product-strategic data such as costs or lead time can also be managed, used and changed. The large amount of data, as mentioned in the application examples, could be managed and processed in a comprehensible way within one model. The software support enables the further digitalization of the product development and simplifies the methodical product development in the future. Automation can also be implemented through software support. A link to external software can also be made possible by means of SysML-based software. By the calculations and analyses made possible thereby, the model for methodical development of modular product architectures can be further improved via a standardized interface, and thus methodical product development can be supported even more. The same applies to the use of additional tools and diagrams within the SysML modeling language, which can be used to add correlations that were previously not available. Thus, the method user can be provided with additional support tools and the results can be optimized. Several additional dependencies are visible due to the consistent and cross-model modeling. In addition to the easier application of methods, the data model also facilitates the transfer of methods. The sustainability of modular product families can thus also be supported.

However, it was found that the implementation effort can be high and the procedure time-consuming. If the goals are not sufficiently considered in advance, there is still the danger that too much modeling will be performed and the work will be inefficient. Furthermore, the comparison of the SysML modeled tools using PowerPoint-based tools shows that the visualization possibilities are limited. With the framework presented, all these activities can be performed within a networked approach. The various steps and areas can be edited in both the vertical and horizontal planes, with their links visible. An extension with further method modules can also be easily implemented and built on the existing modeling and is supported by the approach of the framework.

8. Summary and Outlook

In this paper, it is shown how modularization methods, configuration systems and the Impact Model of Modular Product Families could be combined in such a way that a comprehensive and consistent modeling became possible. Therefore, the development and configuration of modular product family concepts and their effects and linkages can be implemented based on a methodical framework. It was used to implement methods of product family modeling by means of Model-Based Systems Engineering (MBSE) in the modeling language SysML. For this purpose, the state of the art was shown in Section 2 and, based on that, different sources of inconsistency between different methodical tools were highlighted and, based on a co-citation analysis, the need for research was shown. The application of the methodical framework was presented, as was its use for supporting the developing of a modular product family of vacuum cleaner robots. Also, a configuration system and an effect model of modular product families were linked by the MBSE and applied to a product family of laser systems. This made it clear that the framework can be used to model various methodical topics of product family modeling in a consistent manner. This reduces errors, decreases effort and increases traceability across different methodical tools. In the future, the other missing elements of the framework should be modeled. Especially, the process model with the product data model should be supplemented to improve the temporal consistency. Existing SysML models can be integrated more easily in the future, to provide access to the corresponding data. In particular, the modeling of a user interface is planned, which will make the method application more consistent and valid with the existing industrial data. In addition to the two use cases shown, application in the aviation industry is also planned, where approaches of MBSE have also already begun to

be tried out. Since this paper has shown validation in terms of applicability, future research would still need to verify the usability and usefulness.

The complexity cost analysis [1] and key performance indices should also be linked to the product architecture data, since it is not only important for methodical product development, but also supports the estimation of the effects of modular product architecture concepts and shows the effective selection of configurations. The application of the framework to other methodical aspects such as modular lightweight design can also support consistent SysML modeling in the future [50]. This could further improve cross-method data management and expand existing interdisciplinary SysML-based approaches [51].

At the product data level, the connection to a digital twin should be further researched and implemented [52]. In this context, the application for aircraft cabin development is particularly interesting, as the different challenges of variant management and data management can also occur there, and initial approaches to the use of MBSE have already been applied [53].

Author Contributions: Conceptualization, M.H.; methodology, M.H.; software, M.H., L.-N.W. and F.M.D.; validation, M.H., L.-N.W. and F.M.D.; formal analysis, M.H., L.-N.W. and F.M.D.; investigation, M.H., L.-N.W. and F.M.D.; resources, M.H., L.-N.W. and F.M.D.; data curation, M.H., L.-N.W. and F.M.D.; writing—original draft preparation, M.H., L.-N.W. and F.M.D.; writing—review and editing, M.H., L.-N.W., F.M.D. and D.K.; visualization, M.H., L.-N.W. and F.M.D.; supervision, D.K. All authors have read and agreed to the published version of the manuscript.

Funding: Publishing fees supported by Funding Programme Open Access Publishing of Hamburg University of Technology (TUHH). This research was funded by the Bundesministerium für Wirtschaft und Energie (Federal Ministry for Economic Affairs and Energy) as part of the Federal Aeronautical Research Programme LuFo VI-1, grant number 20K1902C.

Data Availability Statement: Not applicable.

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

References

1. Krause, D.; Gebhardt, N. *Methodical Development of Modular Product Families*; Springer: Berlin, Germany, 2023; ISBN 978-3-662-65679-2. [CrossRef]
2. Pimmler, T.; Eppinger, S. Integration analysis of product decompositions. In Proceedings of the 6th Design Theory and Methodology Conference, New York, NY, USA, 11–14 September 1994; pp. 343–351.
3. Hackl, J. *Wirkmodell der Eigenschaften Modularer Produktstrukturen*; Springer: Berlin, Germany, 2022.
4. Dambietz, F.M. *Performance Simulation of Modular Product Architectures by Model-Based Configuration*; Springer: Berlin, Germany, 2022. [CrossRef]
5. Krause, D.; Schwede, L.N.; Dambietz, F.M.; Hanna, M. Model-Based Systems Engineering: Discovering Potentials for Methodical Modular Product Development. In *Design Methodology for Future Products*; Krause, D., Heyden, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2022; pp. 265–285. [CrossRef]
6. Min, G.; Suh, E.S.; Hölttä-Otto, K. System Architecture, Level of Decomposition, and Structural Complexity: Analysis and Observations, *ASME. J. Mech. Des.* **2016**, *138*, 021102. [CrossRef]
7. Lindemann, U.; Maurer, M.; Braun, T. *Structural Complexity Management: An Approach for The Field of Product Design*; Springer: Berlin, Germany, 2009.
8. Stone, R.B. Towards a Theory of Modular Design. Ph.D. Thesis, University of Texas, Austin, TX, USA, 1997.
9. Otto, K.; Hölttä-Otto, K.; Sanaei, R.; Wood, K.L. Incorporating Field Effects Into Functional Product-System Architecting Methods, *ASME. J. Mech. Des.* **2020**, *142*, 041402. [CrossRef]
10. Erixon, G. Modular Function Deployment: A Method for Product Modularization. Ph.D. Dissertation, The Royal Institute of Technology, Stockholm, Sweden, 1998.
11. Baylis, K.; Zhang, G.; McAdams, D.A. Product family platform selection using a Pareto front of maximum commonality and strategic modularity. *Res. Eng. Des.* **2018**, *29*, 547–563. [CrossRef]
12. Chen, W.; Hoyle, C.; Wassenaar, H.J. A Decision-Based Design Approach to Product Family Design. In *Decision-Based Design*; Springer: London, UK, 2013. [CrossRef]
13. Birkhofer, H.; Klobardanz, H.; Berger, B.; Sauer, T. Cleaning up design methods-describing methods completely and standardized. In Proceedings of the DESIGN 2002 7th International Design Conference, Dubrovnik, Croatia, 14–17 May 2002.
14. Gupta, S.; Kremer, G. Analyzing Three Modularizing Methodologies From Assembly and Variety Viewpoints. In Proceedings of the Industrial Engineering Research Conference, Vancouver, Canada, 17–20 May 2008; pp. 1660–1665.

15. Hölttä-Otto, K.; Chiriac, N.A.; Lysy, D.; Suh, E.S. Comparative Analysis of Coupling Modularity Metrics. *J. Eng. Des.* **2012**, *23*, 787–803. [\[CrossRef\]](#)
16. Salvador, F. Toward a Product System Modularity Construct. Literature Review and Reconceptualization, IEEE Transactions on Engineering Management. *IEEE Xplore* **2007**, *54*, 219–240. [\[CrossRef\]](#)
17. Plietz, M. Patterns in der Produktkonfiguration. In Proceedings of the Tagungsband zum 14 Interuniversitären Doktorandenseminar Wirtschaftsinformatik, Chemnitz, Germany, 14 July 2011.
18. Liebisch, M. Aspektorientierte Datenhaltung in Produktkonfiguratoren—Anforderungen, Konzepte und Realisierung. Ph.D. Dissertation, Friedrich-Schiller-Universität, Jena, Germany, 2014.
19. Kortmann, D.; Klink, H.; Wüpping, J. Strategien zur profitablen Variantenkonfiguration. *Int. J. Interoperability Bus. Inf. Syst.* **2009**, *9*, 57–60.
20. Pakkanen, J.; Juuti, T.; Lehtonen, T. Brownfield Process: A method for modular product family development aiming for product configuration. *Des. Stud.* **2016**, *45*, 210–241. [\[CrossRef\]](#)
21. Herzig, S.; Qamar, A.; Reichwein, A.; Paredis, C. A Conceptual Framework for Consistency Management in Model-Based Systems Engineering. In Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE, Washington, DC, USA, 29–31 August 2011.
22. Finkelstein, A.; Kramer, J.; Nuseibeh, B.; Finkelstein, L.; Goedicke, M. Viewpoints: A Framework for Integrating Multiple Perspectives in System Development. *Int. J. Softw. Eng. Knowl. Eng.* **1992**, *2*, 31–57. [\[CrossRef\]](#)
23. Easterbrook, S.; Finkelstein, A.; Kramer, J.; Nuseibeh, B. Coordinating Distributed ViewPoints: The anatomy of a consistency check. *Concurr. Eng. Res. Appl.* **1994**, *2*, 209–222. [\[CrossRef\]](#)
24. Quamar, A.; Paradeis, C. Dependency Modelling And Modell management in Mechatronic Design. In Proceedings of the ASME 2012 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE, Chicago, IL, USA, 12–15 August 2012.
25. Stachowiak, H. *Allgemeine Modelltheorie*; Springer: New York, NY, USA, 1973.
26. Albers, A.; Bursac, N.; Scherer, H.; Birk, C.; Powelske, J.; Muschik, S. Model-based systems engineering in modular design. *Des. Sci. J.* **2019**, *5*, e17. [\[CrossRef\]](#)
27. Eckert, C.M.; Wynn, D.C.; Maier, J.F.; Albers, A.; Bursac, N.; Xin Chen, H.L.; Clarkson, P.J.; Gericke, K.; Gladysz, B.; Shapiro, D. On the integration of product and process models in engineering design. *Des. Sci.* **2017**, *3*, e3. [\[CrossRef\]](#)
28. INCOSE. INCOSE Systems Engineering Vision 2020. Available online: https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf (accessed on 19 July 2023).
29. Kaufmann, U.; Pfenning, M. *Was Die Produkt-von der Softwareentwicklung Lernen Kann-Durchgängige Integration Disziplinspezifischer Modelle Durch den EINSATZ von Modellierungssprachen*; Tag des Systems Engineerings (TdsE): Bremen, Germany, 2014. [\[CrossRef\]](#)
30. Holt, J.; Perry, S.; Brownsword, M. *Model-Based Requirements Engineering*; The Institution of Engineering and Technology: London, UK, 2012.
31. Mueggo, C.; Pfenning, M. *Die Rolle von MBSE und PLM im Industrial Internet*; Tag des Systems Engineering (TdSE): Ulm, Germany, 2015. [\[CrossRef\]](#)
32. Eigner, M.; Gilz, T.; Zafirov, R. Proposal for functional product description as part of a PLM solution in interdisciplinary product development. In Proceedings of the DESIGN Conference (DESIGN 2012), Dubrovnik, Croatia, 21–24 May 2012.
33. Sendler, U.; Weikiens, T. *Modellbasierte Systementwicklung—Was Sie Schon Immer Über MBSE, PLM und Industrie 4.0 Wissen Sollten*; Tag des Systems Engineering (TdSE2013): Stuttgart, Germany, 2013.
34. Bursac, N. *Model Based Systems Engineering zur Unterstützung der Baukastenentwicklung im Kontext der Frühen Phase der Produktgenerationsentwicklung*; IPEK: Garbsen, Germany, 2016.
35. Scherer, H. *Model Based Methods for the Modeling of the System of Objectives and the Correlation between Form and Function to support the Series Development of Modular Systems Using the Example of Hybrid Powertrains*; IPEK: Garbsen, Germany, 2016.
36. Stirgwolt, B. A Model-Based Systems Engineering Approach for Developing Modular System Architectures. *J. Eng. Des.* **2022**, *33*, 95–119. [\[CrossRef\]](#)
37. Weikins, T. *Systems Engineering with SysML/UML: Modeling, Analysis, Design*; Morgan Kaufmann: Burlington, MA, USA, 2008.
38. Kim, G.; Kwon, Y.; Suh, E.S.; Ahn, J. Analysis of Architectural Complexity for Product Family and Platform, ASME. *J. Mech. Des.* **2016**, *138*, 071401. [\[CrossRef\]](#)
39. Reid, T.N.; MacDonald, E.F.; Du, P. Impact of Product Design Representation on Customer Judgment, ASME. *J. Mech. Des.* **2013**, *135*, 091008. [\[CrossRef\]](#)
40. Hvam, L.; Riis, J.; Malis, M.; Hansen, B. A Procedure for Building Product Models. In *Moving into Mass Customization*; Rautenstrauch, C., Seelmann-Eggebert, R., Turowski, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2002. [\[CrossRef\]](#)
41. Gräßler, I.; Wiechel, D.; Koch, A.-S.; Sturm, T.; Markfelder, T. Methodology for Certification-Compliant Effect-Chain Modeling. *Systems* **2023**, *11*, 154. [\[CrossRef\]](#)
42. Bilic, D.; Brosse, E.; Sadovykh, A.; Truscan, D.; Bruneliere, H.; Ryssel, U. An Integrated Model-based Tool Chain for Managing Variability in Complex System Design. Models and Evolution Workshop (ME 2019). In Proceedings of the Co-Located with the IEEE/ACM 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS 2019), Munich, Germany, 15–20 September 2019. [\[CrossRef\]](#)

43. Wolny, S.; Mazak, A.; Carpella, C.; Geist, V.; Wimmer, M. Thirteen years of SysML: A systematic mapping study. *Softw. Syst. Model.* **2020**, *19*, 111–169. [[CrossRef](#)]
44. Vogelsang, A.; Amorim, T.; Pudlitz, F.; Gersing, P.; Philipps, J. Should I Stay or Should I Go? On Forces that Drive and Prevent MBSE Adoption in the Embedded Systems Industry. In *Product-Focused Software Process Improvement; PROFES 2017. Lecture Notes in Computer Science*; Felderer, M., Méndez Fernández, D., Turhan, B., Kalinowski, M., Sarro, F., Winkler, D., Eds.; Springer: Cham, Switzerland, 2017; Volume 10611. [[CrossRef](#)]
45. Manoury, M.; Horländer, T.; Zimmermann, T. Potentials of Design Thinking for knowledge transfer of Model-Based Systems Engineering. In *Proceedings of the 2022 IEEE International Systems Conference (SysCon)*, Montreal, QC, Canada, 25–28 April 2022; pp. 1–8. [[CrossRef](#)]
46. Blessing, L.; Chakrabarti, A. *DRM, a Design Research Methodology*; Springer: Cham, Switzerland, 2009. [[CrossRef](#)]
47. Pedersen, K.; Emblemavag, J.; Bailey, R.; Allen, J.K.; Mistree, F. The “Validation Square”—Validating Design Methods. In *Proceedings of the DETC '00, 2000 ASME Design Engineering Technical Conferences*, Baltimore, MD, USA, 10–13 September 2000.
48. Olewnik, A.T.; Lewis, K. On Validating Engineering Design Decision Support Tools. *Concurr. Eng.* **2005**, *13*, 111–122. [[CrossRef](#)]
49. Hermann, M.O.; Michler, J.; Schöthaler, F. Wo Kundenwünsche auf technische und wirtschaftliche Notwendigkeiten treffen. *Bus. News* **2013**, *3*, 2013.
50. Hanna, M.; Schwenke, J.; Schwan, L.; Krause, D. Methodical Approach for the Model-Based Development of Aircraft Cabin Product Families under Consideration of Lightweight and Cost-Based Design. In *Proceedings of the Design Society: DESIGN Conference*, Dubrovnik, Croatia, 23–26 May 2022; pp. 435–444. [[CrossRef](#)]
51. Kernschmidt, K. Interdisciplinary Structural Modeling of Mechatronic Production Systems Using SysML4Mechatronics. Ph.D. Dissertation, Technische Universität München, München, Germany, 2018.
52. Loaiza, J.H.; Cloutier, R.J.; Lippert, K. Proposing a Small-Scale Digital Twin Implementation Framework for Manufacturing from a Systems Perspective. *Systems* **2023**, *11*, 41. [[CrossRef](#)]
53. Fuchs, M.; Beckert, F.; Biedermann, J.; Nagel, B. A collaborative knowledge-based method for the interactive development of cabin systems in virtual reality. *Artic. Comput. Ind.* **2022**, *136*, 103590. [[CrossRef](#)]

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