Towards Model-Based Modular Product Families in the Context of Systems Engineering

Christian M. Berschik; Dieter Krause

Hamburg University of Technology, Germany

Abstract: The complexity and interdisciplinarity in the development of products have continued to increase in recent decades. New trends such as digitalisation or the subject of sustainability are leading to new challenges in the development of competitive products. Developing and realising product families has proven to be a promising approach to overcoming these challenges. In addition, many approaches in Systems Engineering can be used to address the changing boundary conditions of development. The trend towards digitally linked approaches in the field of Model-Based Systems Engineering is particularly promising. This contribution aims to look at the two subject areas and to integrate the structural data required for product family development into the understanding of Systems Engineering. The required containers of information are derived and explained using an example of vacuum cleaner robots. Finally, the approach and the advantages of linking product family development with systems engineering aspects are discussed.

Keywords: Systems Engineering (SE), Model-Based Engineering, Modular Product Family, Model-Based Systems Engineering, Variant Management

1 Introduction

In recent decades, the complexity and interdisciplinarity of product development has risen significantly. It can be attributed to the increased competitive pressure from international competitors, who demand a fast response time to changes and a greater variety of products. It can also be explained by new challenges, such as increasing digitalisation and focusing on sustainability. The many external parameters result in more significant effort for structuring the product architectures of products (Mertens et al., 2022). Developing product families has proven to be a promising approach to counteract this. These approaches consider product families from a technical-functional and product-strategic perspective and thus lead to a better lifecycle view (Simpson et al., 2005; Krause and Gebhardt, 2023).

Systems Engineering (SE) is a widely used approach for describing and developing systems. Here, changes are also being induced by varying external conditions. Similar to product development, there is a trend towards more digitally linked approaches. These can be summarised in Model-Based Systems Engineering (MBSE) and go far beyond actual SE (Walden et al., 2015; Berschik et al., 2023).

When strategically handling multiple products within product families with even more internal dependencies compared to single products, model-based approaches have an even bigger potential. However, because of the strong relation to the processes of systems engineering, it is first necessary to analyse how modular product families can be integrated or distinguished from products as understood by the SE. Only then it is feasible to transfer the model-based approach, as it originated from MBSE and SE, to product families and create an approach for model-based modular product families. At first glance, it becomes evident that product family development is hardly considered in the context of SE. Especially when focussing on the modular product family, the development in the context of SE can provide added value for a holistic view and modelling. This paper will take a deeper look into the combination of Product Family Engineering and Systems Engineering.

The following section will classify both topics as a basis for further development and demonstrate the potential of a targeted approach from both perspectives. Therefore, the research background is first presented, then both topics are categorised in relation to each other, and the resulting potential benefits are discussed.

2 Research Background

The following explains and analyses the basic concepts of Systems Engineering and Model-Based Systems Engineering. The topics of product lines and product families are also presented.

2.1 Systems Engineering and Model-Based Systems Engineering

Systems Engineering (SE), as described by the International Council on Systems Engineering (INCOSE), is an interdisciplinary approach to supporting the development of systems. It is based on the system concept, a view of reality in which the integrated links between system parts are considered in the overall context. To this end, SE provides various processes that support the development of systems technically, organisationally and project-related. Technical considerations occur from the requirements recording to the final system validation (Haberfellner et al., 2019; Walden et al., 2015). The processes described are defined in (Walden et al., 2015) and ISO 15288 (International Standard ISO, 2023) and can also be found holistically from a technical perspective in the V-Model (Verein Deutscher Ingenieure, 2021). The SE approach described supports the developer with a multidisciplinary approach. It thus provides a basis for overcoming the challenges arising from ever-growing projects and the associated dependencies within the developed systems. It also enables the development of mechatronic or cyber-physical systems, which pose particular challenges due to their internal structures and links (Walden et al., 2015).

The SE uses document-centred documentation of the development data. A model-based approach was developed to meet the associated challenges, especially in larger and more interdisciplinary projects called MBSE, which stores development data in formally described models that link different perspectives in a consistent system model (Holt and Perry, 2008).

MBSE is based on the fundamental use of models. Systems Theory Models played a key role in the initial establishment of model-based approaches. The models utilised discrete time automata or state machines. With the help of these models, the behaviour of systems was described fundamentally for the first time (Wymore, A.W., 1993). These mathematical models are seen as the predecessors of today's visual system models, which can be constructed using modelling languages. INCOSE and the Object Management Group played a crucial role in the formalisation and further development of the understanding of SE and MBSE. They paved the way for the model-centred, interdisciplinary development of systems (INCOSE, 2007).

Graphical modelling languages are often used when modelling system models. One example of this is the System Modelling Language (SysML). Authoring tools apply the modelling languages with which the system can be modelled. In addition, a modelling method is required that specifies the modelling procedure (Delligatti, L., 2014). With the help of these three principles, the elements of the individual system can be created and stored in a Single Source of Truth. The resulting data consistency improves access and simplifies the maintenance of stored data. Viewing different areas and extracts from the overall model from different perspectives is also possible. The modelling of system models is not tied to the individual application. It means that modelling is not limited to the tasks and processes of the SE. Such applications are, therefore, not initially SE and MBSE but a model-based implementation of a specific topic. However, system modelling and data handling can be applied to various challenges today (Berschik et al., 2023).

2.2 Product Families and Product Lines

Products, which can also be referred to as systems depending on the level of detail considered, that a company offers can be categorised into groups according to various criteria. The terms frequently used in the literature are product line and product family. In the following, the different definitions are explained and categorised in relation to each other.

A product line represents a group of products usually considered together on a company-specific basis for organisational reasons (Rupp, 1988). At this level, products are grouped according to strategic markets or, from a business perspective, by operational programme planning. The company's resulting organisational division also affects the division of departments such as development. Product lines bring together products related to each other according to a similar area of application, function, demand, or production context (Krause and Gebhardt, 2023; Meffert et al., 2012). When considering product families, a next lower level must be established at this point, describing the product variants inside of product families within the product lines.

Product families consist of different product variants that utilise a similar area of application and the same technical solutions or technologies (Meyer 1997). The product variants under consideration are a set of products that mainly use the same components, have a very similar design and exhibit the same functional behaviour but differ in at least one technical characteristic (Krause and Gebhardt, 2023; Franke et al., 2002; Simpson et al., 2001). In the best-case scenario, the differing technical properties should influence the customer's decision and thus support the selection of a product variant (Mortensen, 1999). Further synergies between the product variants can be utilised by establishing product families. For example, holistic development approaches can be introduced, thus reducing development time or increasing quantities through the number of identical parts. Overall, this increases the competitiveness of product variants in product families (Krause and Gebhardt, 2023). If the consideration within product families is extended further, a significant increase in efficiency in development and production can be ensured on the one hand by reducing the variance of individual components used in the product family and on the other hand by using modular structures (Krause and Gebhardt, 2023). This increase in efficiency is particularly evident in products characterised by mechanical engineering but also in mechatronic products and products that exhibit a great deal of variance in their design, as monetary aspects also increasingly arise here due to the commitment of capital during procurement and production. The resulting modular product families are characterised by the possibility of combining different types of modules with specific, predefined interfaces. Modular structures are particularly suitable for mapping a wide range of products with a low internal variance of components (Krause and Gebhardt, 2023).

Above the product families and product lines described is the so-called product program. It includes all goods produced in the company and purchased goods that are resold without further processing (Krause and Gebhardt, 2023; Meffert et al., 2012).

As shown in Figure 1, the product program can be divided into various product lines, which in this example are subdivided into professional robots for industrial use, household robots and accessories. Each of the product lines covers a different market and a different strategic orientation. Multiple product families can be categorised within the Household Robots product line, which implements different technical solutions. For example, there is the vacuum cleaner and mopping robot product family. Below the product family are the different product variants on offer, each with at least one other technical feature.

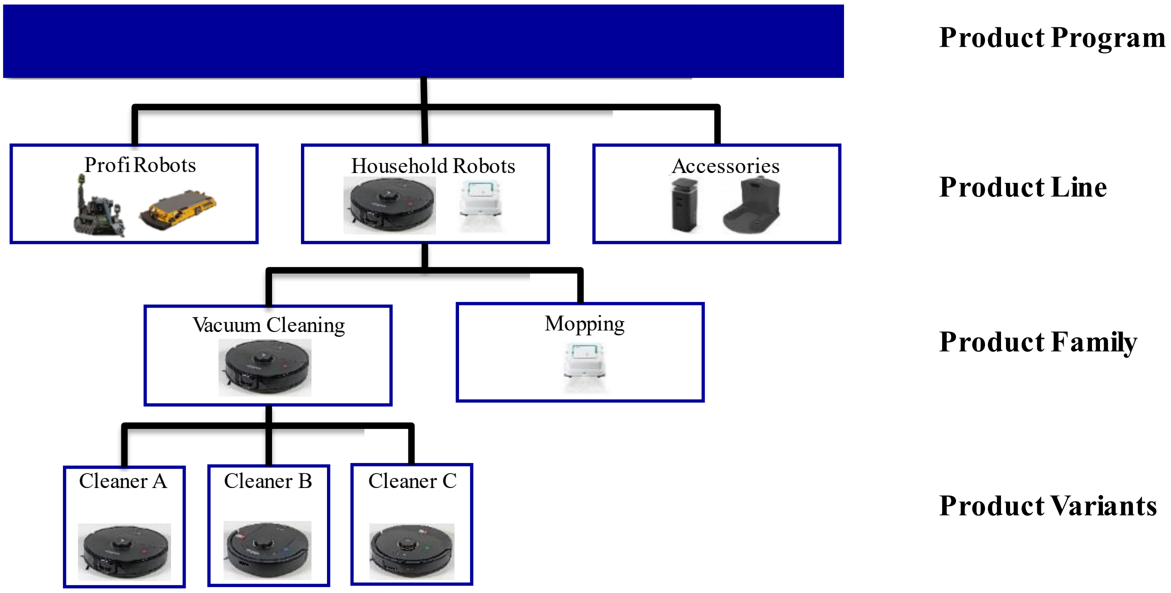

Figure 1. Difference between Product Lines and Product Families using the example of cleaning robots, adapted on (Krause and Gebhardt, 2023)

To summarise, it can be seen that modular product families focus on the structuring and product-oriented development of product variants. Accordingly, topics such as considering variance and, in a broader sense, mapping modules or modular structures are essential building blocks for the holistic description of product families. In the area of SE, this topic is only described to a limited extent. SE has essentially developed from the military and aerospace sectors, in which systems are mainly produced and developed in small quantities. The use of holistic approaches such as SE has many advantages due to the high development cost and the sometimes extreme demands placed on the systems regarding their functionality and application areas (Honour, 2014). In particular, the model-based representation of procedures allows large systems to be documented efficiently. With the integration of SE processes outside the areas described and the associated increase in the number of units and breadth of the range, the topic of product line engineering (PLE) has become established in this context based on the software development approach (International Standard ISO, 2021).

PLE was adopted from software engineering and is sometimes used synonymously for the software product family engineering (Pohl et al., 2005). In the 1970s, Parnas (Parnas, 1976), introduced the approach of reusing software code parts for software programs with similar properties. This clustering was described as a program family and can be seen as the basis for today's Software Product Line Engineering (SPLE). The basic concept of today's product lines was demonstrated with the description of the Feature-Oriented Domain Analysis (FODA) method (Kang et al., 1990). SPLE is generally split into two fields: domain engineering and application engineering. Domain engineering forms the product line infrastructure and maps various assets to describe software. These assets can be reused for the program description and are, therefore, a development for reuse. In application engineering, the software programs are developed specifically for the application. They build on the product line infrastructure and implement the desired customer properties. In doing so, they implement the defined features that can be chosen from and thus constitute the paradigm of development with reuse. In principle, the focus is on the reuse of software parts to reduce costs and time or for quality assurance (Linden et al., 2007; Käköla and Duenas, 2006).

The PLE was derived from this existing approach to building software programs and anchored in SE. As described at the beginning, product line engineering involves a market-oriented categorisation of the various systems on offer, which are composed and defined with the help of features concerning the customer's wishes. With the generic approaches such as the software and systems product line engineering, which uses a feature-based description to generate advantages through a common platform and reusable assets for product families, an expansion towards the engineering of systems was made. (International Standard ISO, 2021). The description of systems is based on the various features and shows the use of domain and application engineering. Nevertheless, the focus is strongly on the features relevant to the customer, that is, the external variance of products and the resulting variation points of the product structures, and less on the technical implementation within the superordinate product family. However, the challenges posed by the internal variance that occurs and the synergies of using sensibly customised modules are not considered here, as this aspect only plays a subordinate role in the development of software code without physical assets or a lower quantity of products produced. If products with larger quantities are considered, an increase in efficiency and a reduction in complexity costs can be achieved by establishing modular product families (Krause and Gebhardt, 2023).

3 Challenges of Modelling Modular Product Families

As described in the last section, modular product families can be used to counter better existing external influences that companies experience. The internal variance can be reduced by restructuring the product family, considering the variance with a suitable modularisation. In this way, the complexity of the offered products can be controlled, and hence, a modular product family is established. Modular product families are initially complicated by their structure as there are many dependencies, especially regarding variance (Krause and Gebhardt, 2023). They have many dependencies to consider when developing and maintaining such structures.

Model-based solutions are a viable approach to map the various interrelationships better. In particular, the advantages of a Single Source of Truth and the associated traceability of interrelationships beyond system boundaries can positively affect the development and utilisation of product families. These result from the better overview, the (semi-)formal form of representation and the computer readability made possible by the digital modelling of a system. It makes it easier to build on models with other software solutions and to expand the existing development data over the product life cycle (Laukotka and Krause, 2023). An in-depth review of the literature on model-based approaches shows that SE and the associated MBSE have already pursued many promising approaches for application to product families. For example, Hanna et al. have shown how development data of a modularisation process can be modelled using SysML (Hanna et al., 2023).

In particular, the careful consideration of the structure and the behavioural description for the developed systems occurs when considering product families - compare modelling methods such as OOSEM (Friedenthal, 2014), Alt (Alt, 2012) or CONSENS (Gausemeier, 2012). Similar to model-based approaches to PLE, such as MBPLE (NoMagic, 2023), several systems are developed together. The emphasis lies on the features described and the resulting variation points in the product structure. There is no further technical consideration, such as a behaviour description. Variance is also addressed in many approaches, such as VAMOS (Weilkiens, 2016). However, no distinction is made between internal and external variety. In particular, considering internal, primarily technical variety is very often not considered. Therefore, there is already much preliminary work in SE and MBSE for sub-areas considering modular product families. However, there is still no holistic support for the modelling, especially regarding modularity. For this reason, the first step is to consider how a modular product family can be integrated into the SE notation at a structural level. Accordingly, the research question "To what extent can modular product families be considered or integrated in the Systems Engineering Approach?" arises from the abovementioned background. This realisation can be transferred to a model-based approach based on what is presented in the next section. At this point, the first step towards a model-based implementation is to look at how to integrate modular product families into the notation of the SE. Hence, focusing on the model-based realisation will be part of future publications that build upon the result presented here.

4 Modular Product Families in the Systems Engineering Context

The research question described will be approached in the following. Therefore, the fundamental terms of SE and modular product families will first be linked abstractly and later illustrated using an example of vacuum cleaning robots.

In systems engineering, a system describes the reduced representation of the real-world system to be developed. It consists of system elements that cannot be further subdivided, which structure the overall system. The term system can be used at different levels of consideration. Depending on the level of detail, a system element can be understood as a further (sub)system. Each system element is initially a black box in itself, designed as a component of the overall system during further development. Classically, a system is understood to be a single system. Suppose multiple systems are combined to generate a new result that could not be generated by the individual systems on its own. Additionally, the systems can operate independently of each other. In that case, this is known as a System of Systems (SoS). Accordingly, an SoS forms the environment for integrating other interacting systems. If the focus is placed on a single system or a collection of systems that are to be analysed in more detail, this is often referred to as a System of Interest (SoI). (Walden et al., 2015)

When these fundamental elements of SE are expanded to include relevant structural elements of product family development, it can be seen that the number of systems considered must be increased (see Figure 2). If the focus is on developing Modular Product Families, several systems must be regarded as being within the product family. The product family can be understood here as a SoI. The SoI provides a superordinate description of the elements that make up the various product variants. The product variants are themselves systems that exist within the product family, illustrated by the dashed line in Figure 2. A product family is not an SoS, as the product variants do not interact with each other. They have been designed for similar functionality but differ in at least one technical feature. The modular product family, therefore, contains several product variants, each representing a system in itself, see Equ. 1.

$$ModularProductFamily \mathrel{\widehat{=}} \{System \mid System \mathrel{\widehat{=}} ProductVariant\} \quad (1)$$

If this description is continued, it becomes clear that product variants themselves consist of different modules. These modules correspond to the system elements of the product variants in the SE notation. It is important to note that the product variants comprise different realisations of the modules. If this relationship is abstracted, the result is that product variants consist of a set of modules that can be considered system elements of the overall product variant, see Equ.2 and the dashed line in Figure 2.

$$ProductVariant \mathrel{\widehat{=}} \{Module \mid Module \mathrel{\widehat{=}} SystemElement\} \quad (2)$$

This abstract description can be applied to other levels by shifting the level of detail and, thus, the SoI. In this way, the module can be viewed as a system in a recursive representation. Similar to Equ. 2, a module consists of various components representing the module's system elements. The components can exhibit variance across the product family, which is propagated via the modules into the product families.

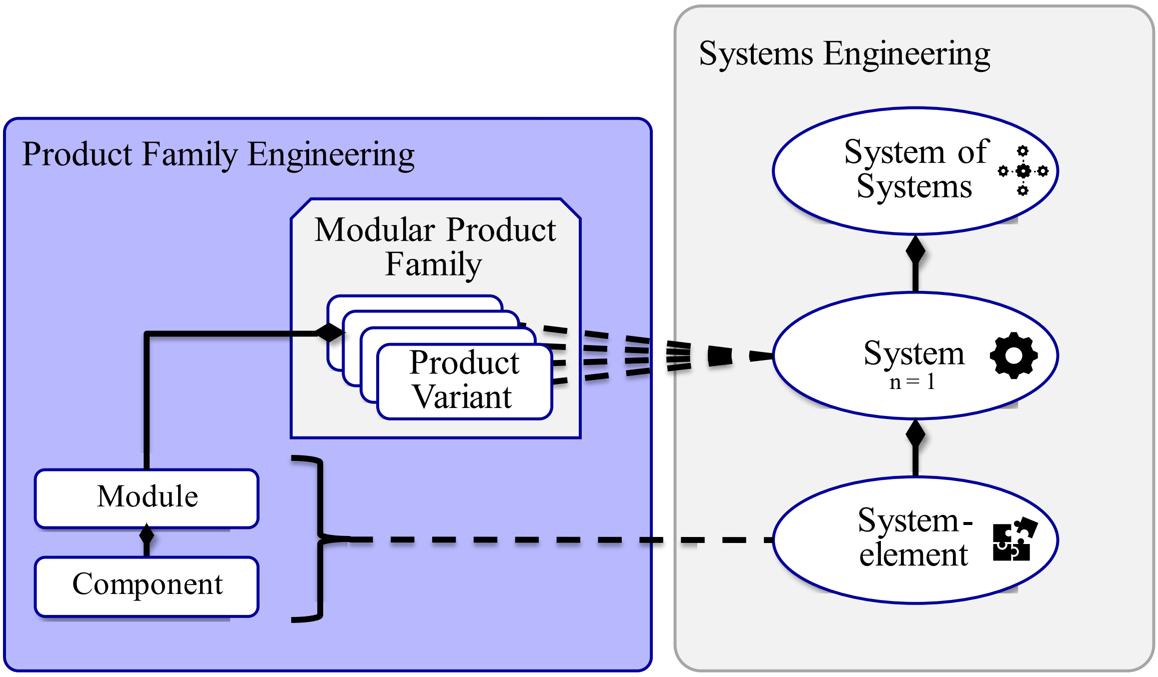

Figure 2. Representation of Modular Product Families as SoI in Correspondence to SE

Therefore, both the product family and the modules have characteristics that represent the actual physical realisation of the system. In the case of product families, these are the individual different product variants. In the case of modules, the variant characteristics are often referred to as module variants or module realisations. Combining product variants in a modular product family is a strategic decision that offers overarching added value in developing and maintaining individual product variants. These levels are very relevant for strategic consideration and offer a combination that makes comprehensive development possible in the first place. However, these layers are not provided for in classic SE. Therefore, the product family and a module are constructs that cannot be categorised in the classic description of systems or SoS but offer added value, especially when considering the variance and, thus, the architecture design through the holistic superordinated view.

Figure 3 illustrates the abstract representation of modular product families using the example of robotic vacuum cleaners. Simple geometric shapes were used to represent variant or standard module realisations instead of detailed example images to simplify the visual representation. The product program, which consists of three different product lines designed for various markets, is shown above. Product line B consists of several product families. One product family of robot vacuum cleaners is demonstrated and analysed in more detail here. The product family consists of several product variants that are build of modules. The higher-level modules represent the essential components of the product variant. In this example, the modules shown for the control module and the brush module are considered standard modules across the product variants, as they use the same realisation everywhere (see ○ and ◊ in Figure 3).

The housing module, on the other hand, is used to different extents in different realisations, as can be seen from the different shades of grey of the ▲visually in the Figure 3. Although the housing module occurs in both product variants, it appears in different realisations and is therefore considered as a variant. Overall, the various realisations can be summarised for the development of the M1 module. By using different realisations of the modules, different product variants can be mapped in the modular product family; this is shown with the two product variants 1 in light blue and product variant 2 in dark blue. The product variants can be based on standard module realisations (e.g., M2 and M3) and variant module realisations (e.g., M1 and M4).

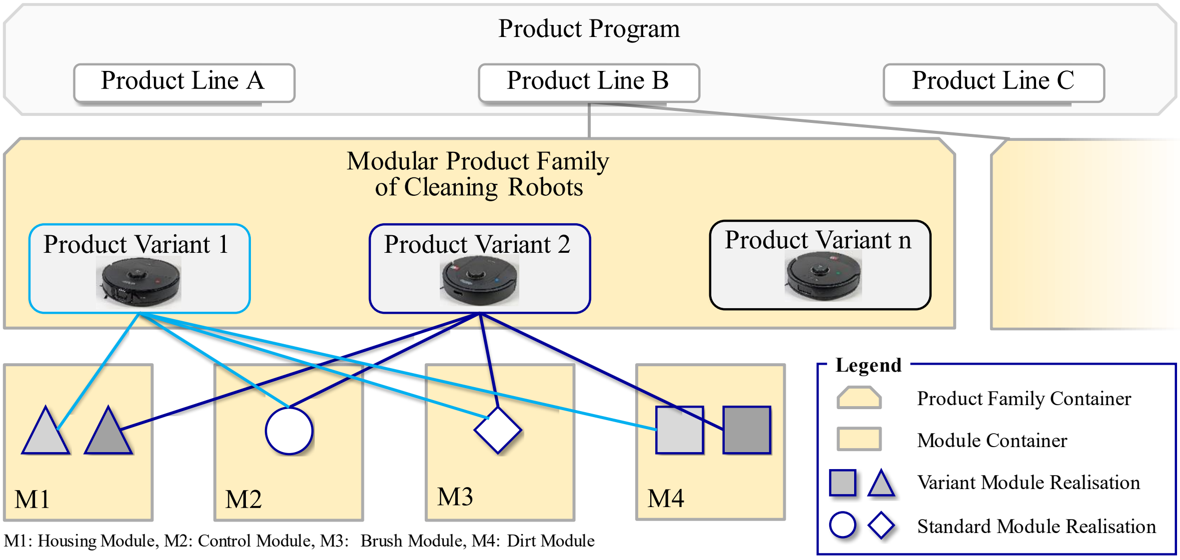

Figure 3. Hierarchical Representation of Modular Product Families with different Containers using the example of Cleaning Robots

As described above, the yellowish areas shown in Figure 3 combine several product variants or module realisations. In the following, these summaries are to be understood as Containers, which are not classic elements of the SE. Instead, these are strategic groupings that improve the cohesion of the product variants during development and further use at the product family level. This grouping is very important for mapping variance at the module level, particularly when considering modular product families. It can be seen that the interactions between the individual systems at their respective product family or module level within the yellow-coloured containers are very high compared to the links between systems that are not considered within product families or modules. It makes it possible to consider module realisations that exist in parallel together and thus potentially be rearranged to reduce the variance within the modules and, therefore, within the product family. Using the modules as an example, the characteristics of modularity, such as the commonality or combinability of the modules, can be taken into account with the help of suitable modularisation. The resulting structured representation can restore the lack of clarity due to the many dependencies and thus be used in particular for analysing internal variety. Due to the unique nature of the modular product family structure, the modules' design has a significant influence on possible synergy effects between the product variants. The systematic approach of the SE combined with the introduction of containers for product families and modules, in which systems and system elements are summarised, can create added value. Without this broadening of the perspective compared to current approaches to SE, this would not be possible in this form without further work. It leads to identifying relevant areas where a stronger focus can be placed on development.

If the number of elements under consideration increases, it is to be expected that the need for model-based support will grow. The advantages of increased traceability and a Single Source of Truth described above are particularly important for larger product families with a high number of modules. The number of links within the product families or modules and between the individual product variants or module realisations should generate the expected added value based on the concept presented here, as already seen in various MBSE approaches. For this purpose, the hierarchical structure shown here can be used. However, it must be discussed further for a model-based implementation. Particular consideration must be given to whether and to what extent implicit modelling of the product variants makes sense for modular product families with a large number of product variants. Developing a modular kit consisting of the modules and their different module realisations could be a way to model the product variants only implicitly and thus reduce the overall modelling effort.

This example shows that the SE can be expanded to include the structural elements of a modular product family. Additional strategic levels must be introduced, further subdividing the system's structural view. The resulting structure supports the mapping of the relationships and can serve as a basis for analysing and documenting the internal variety. Considering the external variety of systems is already widespread with the existing procedures, such as the PLE using feature models. However, the link to internal variety within the system elements offers further potential for research. The link shown is the basis for the container-wide consideration of internal variety within a modular product family, which would not be possible without these additional levels of consideration. It also strengthens the cross-system view, as individual systems and multiple product variants are being considered. Due to the number of links and dependencies and the resulting complexity, the advantages of a transfer to model-based documentation can be utilised. It not only strengthens traceability but also the clarity and reusability of subsystems. This leads ultimately to a reduction in the time needed and, hence, to a potential decrease in costs per sold product.

5 Summary and Outlook

In this contribution, the topics of SE and MBSE, as well as product families and product lines, were initially addressed and explained in the research background. Based on the findings, it was established that there are many synergies between SE and product family development. In the context of SE, the PFE utilises mergers of systems respectively system elements to cluster these elements in containers. The containers are organised from a strategic perspective and can be used, for example, to examine and further improve the communal use of the individual modules. These clustered containers can then be developed self-contained with regard to their defined interfaces. The hierarchic categorisation described is the basis for a further transfer towards model-based documentation of modular product families. Nevertheless, an in-depth consideration of possible modelling approaches needs to be discussed. The advantages of model-based documentation are that it can be used to build on the many approaches within the SE and to integrate the added value from considering containers to represent modular product families. Due to the reduction in internal variance made possible by introducing and developing modular product families, complexity costs can be reduced and time saved during further development.

Further, a model-based implementation of modular product families should be considered based on the categorisation presented. The procedure required for model-based development plays a central role here. The behavioural description should extend the shown structural view to address the existing internal variety also on the behaviour level. Different modelling approaches for realising product variants building on a modular kit must be considered to minimise the modelling effort. The further abstraction of the system relationships using various abstraction models, such as an Ontology or a Metamodel, should be considered for an additional foundation, as already described in (Berschik et al., 2024). These models can be used as a basis for documenting larger system contexts and provide a definition framework for the data and dependencies under consideration, resulting in more stringent overall models even beyond the system boundaries of individual systems.

References

Alt, O., 2012. Modellbasierte Systementwicklung mit SysML. Hanser, München.

Berschik, M.C., Schumacher, T., Laukotka, F.N., Krause, D., Inkermann, D., 2023. MBSE within the engineering design community – an exploratory study, Proceedings of the Design Society, 3, pp. 2595-2604.

Berschik, M.C., Laukotka, F.N., Züfle, M., Krause, D., 2024. Supporting Modular Product Family Representations by methodically utilising Meta-Models, Proceedings of the Design Society: DESIGN Conference, Dubrovnik, Croatia.

Delligatti, L., 2014. SysML distilled: A brief guide to the systems modeling language, Addison-Wesley, New York.

Franke, H.-J., Hesselbach, J., Huch, B., Firchau, N. L., 2002. Variantenmanagement in der Einzel- und Kleinserienfertigung – Mit 33 Tabellen, Hanser-Verlag, München/Wien.

Friedenthal, S., 2014. A Practical Guide to SysML. The Systems Modeling Language. Elsevier Science, Waltham.

Gausemeier, J., 2012. Produkte und Produktionssysteme integrativ konzipieren: Modellbildung und Analyse in der frühen Phase der Produktentstehung. Hanser.

Haberfellner, R., Weck, O. de, Fricke, E., Vössner, S., 2019. Systems Engineering, Springer International Publishing, Cham.

Hanna, M., Wöller, L.-N., Dambietz, F. M. and Krause, D., 2023. A Model-Based Approach for the Methodical Development and Configuration of Modular Product Families. Systems, Vol. 11, pp. 449.

Holt, J., Perry, S., 2008. SysML for Systems Engineering, Institution of Engineering and Technology, Stevenage.

Honour, E., 2010. 11.4.2 Systems Engineering Return on Investment. INCOSE International Symposium, No. 20, pp. 1422-1439.

INCOSE Technical Operations, 2007. Systems Engineering Vision 2020, International Council on Systems Engineering, INCOSE-TP-2004-004-02 , Washington. Available online at https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf.

ISO/EC/IEEE, 2021. International Standard 26580, Software and systems engineering - Methods and tools for the feature-based approach to software and systems product line engineering, 35.080 Software.

ISO/EC/IEEE, 2023. International Standard 15288, Systems and software engineering - System life cycle processes, 35.080 Software.

Kang, K., Cohen, S., Hess, J., Novak, W., Peterson, S., 1990. Feature-oriented domain analysis (FODA) feasibility study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University.

Käköla, T., Duenas, J. C., 2006. Software Product Lines - Research Issues in Engineering and Management. Springer, Heidelberg.

Krause, D., Gebhardt, N., 2023. Methodical Development of Modular Product Families - Developing High Product Diversity in a Manageable Way, Springer Berlin, Heidelberg.

Laukotka, F.N., Krause, D., 2023. Supporting Digital Twins for the Retrofit in Aviation by a Model-Driven Data Handling, Systems 2023, No. 11, 142.

Linden, F., Schmid, K., Rommes, E., 2007. Software Product Lines in Action - The Best Industrial Practice in Product Line Engineering. Springer, Heidelberg.

Meffert, H., Burmann, C., Kirchgeorg M., 2015. Marketing: Grundlagen marktorientierter Unternehmensführung Konzepte - Instrumente – Praxisbeispiele, Springer, Wiesbaden.

Mertens, K.G., Rennpferdt, C., Greve, E., Krause, D., Meyer, M., 2022. Reviewing the intellectual structure of product modularisation: Toward a common view and future research agenda, Journal of Product Innovation Management, pp. 1-34.

Meyer, M., 1997. Revitalize your product lines through continuous platform renewal, Research Technology Management, Vol. 40, No. 2, pp. 17-28.

Mortensen, N.H., 1999. Design modelling in a designer's workbench–contribution to a design language. Dissertation, Department of Control and Engineering Design, Technical University of Denmark, Kgs. Lyngby.

NoMagic Inc., 2023. Product Line Engineering. Available at: https://docs.nomagic.com/display/PLE2021xR2/Product+Line+Engineering (accessed 28 January 2024).

Parnas, D., 1976. On the design and development of program families. IEEE Transactions on Software Engineering, 2(1) pp. 1–9.

Pohl, K., Böckle, G., van der Linden, F.J., 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer, Heidelberg.

Rupp, M., 1988. Produkt-, Markt-Strategien- Handbuch zur marktsicheren Produkt- und Sortimentsplanung in Klein- und Mittelunternehmungen der Investitionsgüterindustrie. Verlag Industrielle Organisation, Zürich.

Simpson, T.W., Maier, J.R., Mistree, F., 2001. Product platform design: method and application, Research in Engineering Design, Vol. 13, pp. 2–22.

Simpson, T. W., Siddique, S., Jiao, J., 2005. Product Platform and Product Family Design: Methods and Applications, Springer, New York.

Verein Deutscher Ingenieure e.V., 2021. VDI-VDE Guideline 2206, Development of mechatronic and cyber-physical systems, 03.100.40, Beuth Verlag GmbH, Berlin.

Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D., Shortell, T.M. (Eds.), 2015. Systems engineering handbook. A guide for system life cycle processes and activities, INCOSE-TP-2003-002-04, International Council on Systems Engineering. 4. edition., Wiley, New York.

Weilkiens, T., 2016. Variant modeling with SysML, MBSE4U, Fredesdorf.

Wymore, A.W., 1993. Model-Based Systems Engineering, CRC Press.

Contact: Berschik M. Christian, Hamburg University of Technology, Germany, markus.berschik@tuhh.de