INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.254

Supporting modular product family representations by methodically utilising meta-models

Markus Christian Berschik ✉, Fabian Niklas Laukotka, Marc Züfle and Dieter Krause

Hamburg University of Technology, Germany

✉ markus.berschik@tuhh.de

Abstract

Due to the rising multidisciplinarity and connectivity of products especially modular product families, a sophisticated handling of the information is crucial for reducing complexity during the development. System modelling techniques have evolved to assist engineers with managing information. However, nowadays, it is rarely focusing on modular product families. This paper introduces a meta-model based on an ontology, which improves the creation and management of modular product family and its occurring data. The meta-model is presented using the example of a Passenger Service Unit (PSU).

Keywords: meta-model, product families, model-based systems engineering (MBSE), modelling

1. Introduction

Due to the increasing customer and market requirements for the development of products, additional effort is required in structuring product architectures to meet these requirements (Mertens et al., 2022). The development of product families offers a suitable and validated approach to this challenge to achieve a more efficient implementation of requirements and needs from both a technical and product-strategic perspective (Simpson et al., 2005; Krause and Gebhardt, 2023).

The development of mechatronic and cyber-physical products is increasingly relevant when considering developing such product families (Hehenberger et al., 2016). However, their variety- and collaboration-induced complexity impacts the development itself and the preceding and subsequent phases of the product life cycle (ElMaraghy et al., 2012; Tomiyama et al., 2019).

This increase in complexity means that the management and coping of the relevant data and its dependencies are becoming more difficult. A suitable framework assisting with the representation and data handling might improve this challenge, but likewise, it needs to consider the particular aspects of product families beyond simply mapping variety on a technical level.

This leads to the overarching research question of how the representation of modular product families can be supported during their development.

Based on previous experiences and works, in order to tackle this challenge of modular product families, an implementation applying systems modelling techniques to support the data handling stands to reason. At that, modelling product families need to consider the technical development background and the resulting complexity of the different product variants as well.

It is proposed to incorporate a higher-level model that enables engineers to transfer even more complex products and their structures into the system model. This allows the models to be consistently and semantically unambiguous across development teams and product variants. The modelling of product families carried out this way increases the traceability, compatibility and collaboration capability of development and further product life phases. A possible procedure for establishing a superordinate meta-model for product families is described below.

2. Research background

In the following, the essential meaning and the difference of the concepts "product family" and "product line" are clarified and stated. Additionally, system modelling and the associated meta-modelling are described as the background of this paper.

2.1. Product Families and Product Lines

Several literature sources view the terms product family and product line differently. A product line is a product set that utilises similar application areas, functions and production processes. These are grouped based on operational or organisational aspects (Krause and Gebhardt, 2023). The company-specific aspects considered are based, for example, on production or market considerations. The executing goal is to offer product families within the product lines tailored to the market and demand (Walden et al., 2015).

In software development, the product line is often considered a generic term for developing long-term investments for software-heavy products. The similarities and variability in different software implementations are defined and recorded with the help of product line architectures. In addition, software product lines are often constructed using features and different member variants are defined by characteristics of these features. Hence, this differentiation is based on an external feature view (International Standard ISO, 2021). This contrasts with the product development perspective, which maps the variety of product variants within product families (Rommes and America, 2006). A product family is a set of different product variants of an application area or production process characterised by similar functional principles or technologies (Meyer 1997).

From a technical point of view, the combined product variants are characterised by using many shared components and very similar designs or functions (Krause and Gebhardt, 2023). Product variants have the same basic functionality but fulfil at least one differing property (Franke et al., 2002). The resulting differentiation between the different product variants is represented in the internal variety. The internal variety describes the variety of components, products or processes that occur within the organisation in the product family. In addition, there is external variety, which represents the variety of products on offer, i.e., the product variants offered to the customer. When considering product families, it is important to ensure that the internal variety is as small as possible; otherwise, it implies variety-induced complexity in the whole product life cycle that must be avoided. A product family can be expanded so that the synergy between the modules used within the product family is significantly increased. In the resulting modular product family, individual product variants can be configured by combining individual module variants using standardised interfaces (Krause and Gebhardt, 2023).

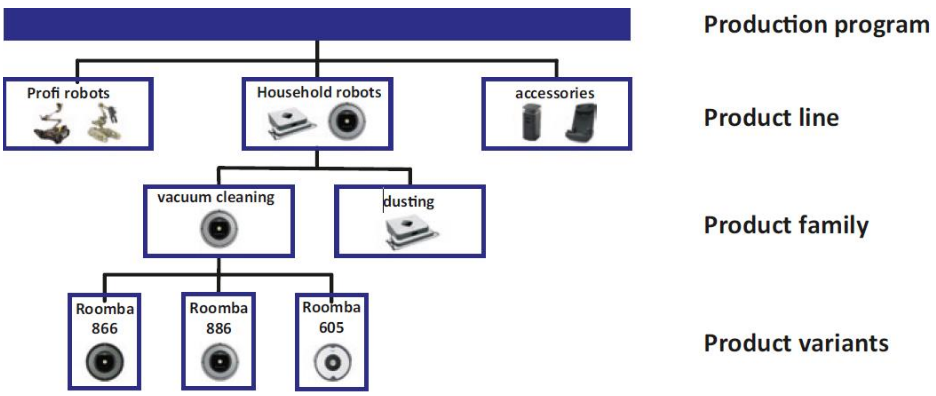

Figure 1. Difference between Product Line and Product Family (Krause and Gebhardt, 2023)

An illustration of the different levels of product lines and product families is given in Figure 1. The exemplary Product Lines of Household and Professional Robots are separated from the Product Families

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

of Vacuum Cleaning and Dusting Robots. Those are subsequently split into different external product variants at the bottom of Figure 1.

2.2. System-Modelling (as originated from MBSE)

The modelling of systems is most prominently being used within the scope of Model-based Systems Engineering (MBSE), where the INCOSE describes it as "the formalised application of modelling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases " (INCOSE, 2007). Specifically targeting the tasks of Systems Engineering (SE), it was established to cope with the challenging handling of information of increasingly larger and more complex products and projects by moving from document-based to model-based documentation. SE goes beyond the task of engineering or developing systems. Instead, it is an interdisciplinary approach that "means to enable the realisation of successful systems" and "focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem […]" (Walden et al., 2015). At that, it provides methodologies and specifically defined tasks for engineering these systems. It comes with much information that needs to be clearly defined and used across different teams and domains.

With the application of system modelling, the information is handled in system models, often in the graphical form of models using the System Modeling Language (SysML) (NoMagic Inc., 2011). Specialised authoring tools allow for the easy creation of models and provide a single source of truth management for the modelled elements (Friedenthal et al., 2015). With multiple views representing different aspects of the system of interest, a clear definition and structure of the system can be created. Depending on the application, different modelling methods guide the user through the modelling process. However, these three pillars of modelling, "modelling language, modelling tool and modelling method," as described by Delligatti (2014), are not bound to the application, to the tasks and processes found in Systems Engineering. While this, then, strictly speaking, is no SE, thus, no *MBSE*, the evolved approaches can be transferred to tasks that face similar challenges. As the challenges regarding the handling of data can be found in many disciplines, today, System Modelling techniques are applied to many different applications and scenarios (Berschik et al., 2023).

It should be noted that the product family itself is rarely considered in modelling. Influenced by software development, various approaches to modelling product lines are summarised in Product Line Engineering. There are multiple methods for extending SysML to include variety, such as VAMOS (Weilkiens, 2016), or special plug-ins, such as MBPLE (NoMagic Inc., 2023) or pure::varients (Pure-Systems, 2023), which were developed for specific tools such as the *CAMEO Systems Modeler* by Dassault Systems. However, there is no superordinate modelling method for modelling Modular Product Families.

2.3. Meta-modelling

When applying system modelling techniques, the closest possible proximity of the models to the real conditions of the respective application facilitates the comprehensibility and usability of these models. Basing the actual models on a definition using so-called meta-models helps by allowing for some framework of available elements and relations. The meta-model provides a basic structure that facilitates the compatibility of the resulting models and ensures their re-usability. A methodical approach for creating this meta-model is to base it on a model-independent ontology based on the structures and definitions found in the real application. Such ontologies define common terminology for a specific scope like a system, perspective, domain or asset. With this definition, they can ensure that the same term is interpreted identically across stakeholders and that there is the same cognitive connection between the term and concept (Lehner, 2021).

Regarding this, meta-models allow the transparent definition of the structure of knowledge within a subject and bolster the communication between involved people or even between people and machines (El-Haji, 2014). The modelling based on such an ontology-based meta-model supports consistent modelling across different people, teams and even projects. This improves the compatibility not only between different models but also between processes that need to work with these models.

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

A meta-model can be used to describe the syntax and semantics of a model. It is, in a manner, a model of a model. It represents and thus specifies the use of a model. Models, in particular meta-models, utilise the so-called closed-world assumption, i.e., everything that is not specified in the model is not explicitly defined and hence not available for further modelling. Furthermore, meta-models are prescriptive, i.e., they prescribe what the modelled world looks like. In contrast, an ontology is descriptive. It uses the open-world assumption and, thus, an under-representation as a stylistic element to describe the world for which it was created (Aßmann et al., 2006).
It explicitly admits that things exist even though they are not defined in the ontology. This coherent creation of the models using methodically created meta-models is endorsed, especially when targeting comprehensive and dynamic systems that need to grow and are handled by different stakeholders.

3. Methodology

As explained in the research background, the term product family is rarely used in the modelling context. On the one hand, this can be attributed to the strong influence of the term product line engineering from software development; on the other hand, the topic of variety is regarded as a fundamental aspect and is therefore mainly considered. Due to the often-missing consideration of the internal variety within the conglomerate of product variants of the product family, variance in different technical solutions is not taken into account appropriately. Yet, this is very important for further consideration and the associated processing and complexity reduction within the product family. As a result, many development artefacts remain unrelated to the model and cannot be consistently mapped.
As shown in Figure 2, the proposed approach is divided into four major steps. In step 1, the database on which the description is based should be collected. To do this, the data to be analysed must be defined. This can be done based on existing development data, which is prepared in such a way that in step 2, a visualisation of the required data and its connections is created in an ontology. The ontology is used as an interface to raise the data used to describe the product family to a more abstract level. With the help of this description, the development team can discuss whether all relevant data has been mapped and linked together. By using an ontology, everything that could be relevant can initially be included without a descriptive limit. This descriptive model can then be used in step 3 to create an initial version of a prescriptive meta-model. To do this, the syntax and semantics of a modelling language such as SysML are used to abstract the scope further.

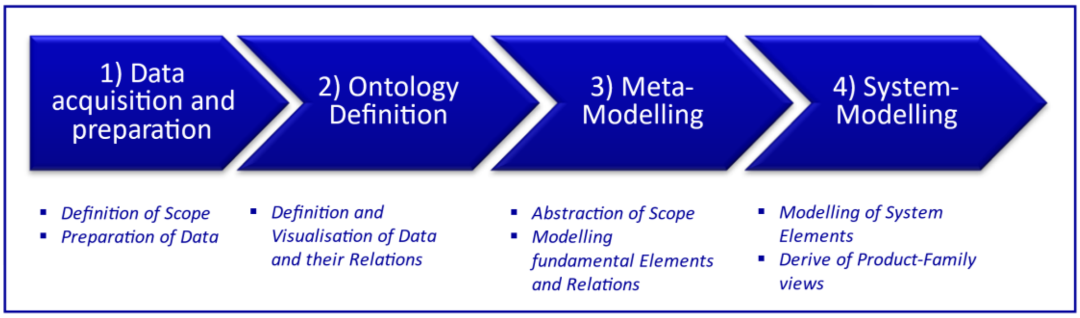

Figure 2. Approach to establish a modelled Product Family

All fundamental data elements and their links can now be modelled in the meta-model in a prescriptive manner. The models based on this thus have a group of modelling elements that can be used to implement the description of holistic product families. Afterwards, step 4 of the system modelling can begin, and the system can be modelled based on the meta-model. Through the standardised definition of the data elements to be used, relevant views for product families can be derived based on the holistic meta-model. When analysing product families, the RFLP approach can be used as a basis. Thus, the various views that can occur during product development can be organised. (Eigner et al., 2014; Gräßler et al., 2018). However, other product family-relevant data must be categorised within the approach. These are, for example, the various elements that can be classified at the physical level of the approach. Not only parts and assemblies are introduced there, but also components; in other words, a further level of clustering is strategically selected for further consideration. These components are combined in

modules. If the different module variants are combined, variants are created that are offered to the customer, also described in (Berschik et al., 2022b; Krause and Gebhardt, 2023). A possible simplified concatenation of different data elements for modelling product families is given in Figure 3. It provides an excerpt of different data used in the methodical development of product families and displays them in the different clusters regarding requirements, function, logical constructs and physical structures. These four layers are essential for product family design because they consider and include the occurring variety and, thus, allow for strategic and holistic management of the product family.
In the next section, the provided approach is applied to a realistic application example of a Passenger Service Unit Product Family.

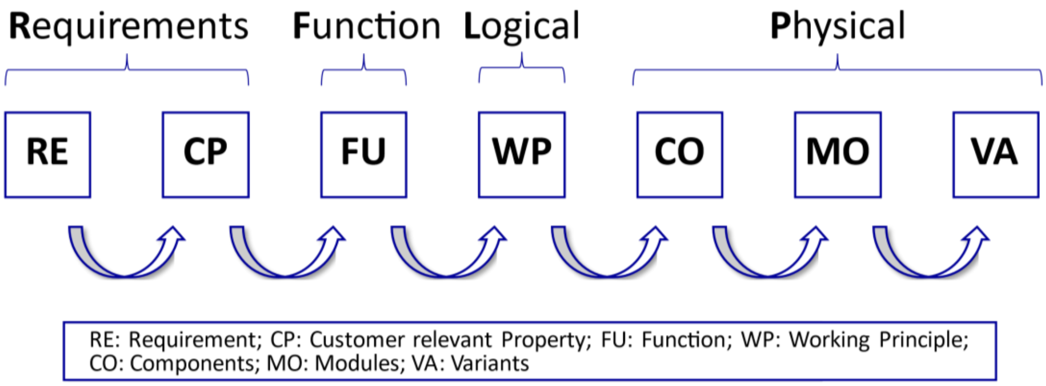

Figure 3. Possible simplified concatenation of data elements for the description of Product Families

4. Exemplary case study

An aircraft's cabin design is one of the main differentiators regarding an airline's competitiveness. The airline is, therefore, forced to guarantee its customers ever-better prices, greater reliability and better comfort. Digital services are also playing an increasingly important role in order to satisfy those demands (Berschik et al. 2022a).

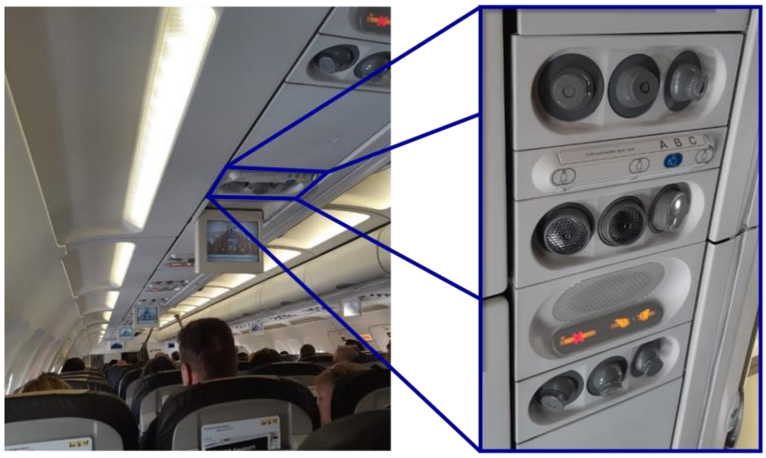

Figure 4. Passenger Service Unit built in the aircraft cabin

One of those interfaces between the cabin and the passenger is the so-called Passenger Service Unit (PSU). The PSU is installed above the seat rows underneath the so-called Overhead Stowage Compartment (OHSC) - the hand baggage stowage in the cabin - in the PSU channel. Various service functions such as a reading light, ventilation and other display options are implemented on the PSU. In addition, safety-relevant components, such as the emergency oxygen supply via oxygen masks, are installed in it. The oxygen masks drop down from the PSU in the event of a pressure drop. The PSU is highly individualised due to its interface with the passenger but offers a clear, functionally oriented modularised structure.

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

Figure 4 shows the PSU's installation situation in the aircraft on the left. On the right, an enlargement of the PSU is shown. It is clear to see that different functions are strictly differentiated in the design. For example, reading light and air ventilation are spatially separated. However, depending on customer needs, the PSU can also be offered in a different configuration. The resulting variants of the PSU are summarised in a product family and will be considered below as a case study.

4.1. Data acquisition and preparation

The product families are analysed based on existing parts lists, configuration tables and engineering bills of materials (EBOM). The relationships between the various components can be taken from the parts lists. The composition can also be further deepened using Aircraft Configuration Guides and verified using extracts from EBOMs. Furthermore, development procedures and standards for developing and certifying aircraft components were consulted and included in the development procedures analysed. To better visualise the PSU, Figure 5 shows a Module Interface Graph (MIG) of a simplified PSU with the components and flows used.

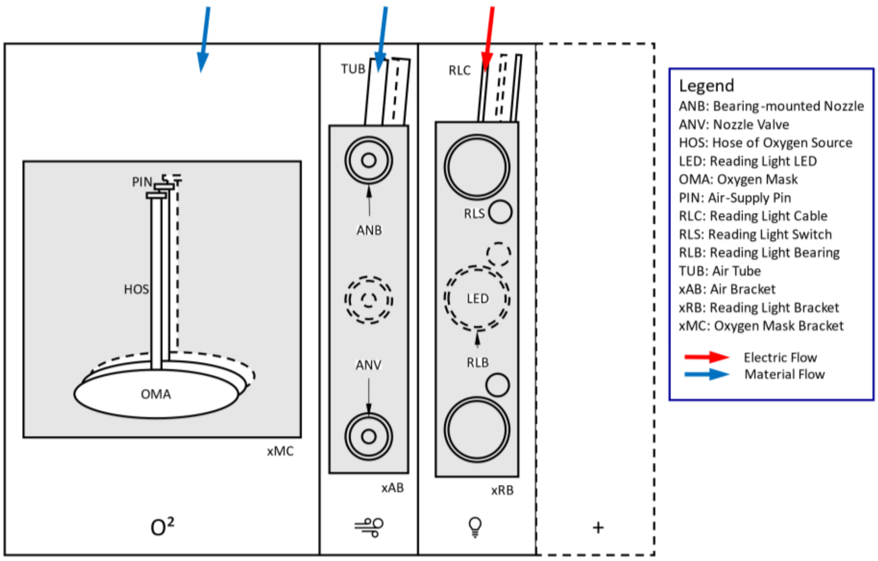

Figure 5. Module Interface Graph of simplified Passenger Service Unit Product Family

It should be noted that a PSU can consist of different functional areas. Depending on the design, these can, for example, provide the oxygen masks, the ventilation or the reading light. There are also so-called dividers, which provide areas without functionality. These dividers are used to change the distance between the modules to cope with the layout of the aircraft cabin. Different areas, which are also grouped as modules, have predefined interfaces to the installation channel, such as a standardised connector for the power transfer of the reading light. The modules exist in different module variants due to the attachment of different components, which can be combined into different modules.

4.2. Ontology definition

Once the development data and the existing documents have been analysed, an ontology is constructed from them. To do this, the data used was abstracted and linked to the data recorded for the development with the help of the development processes used.

Various existing ontologies, such as the ACRE-Ontology (Holt et al., 2012), were used as the basis for the development data. Based on this representation, further data elements can be linked to the earlier-mentioned RFLP clustering. In addition, the data relevant to mapping product families is supplemented. Figure 6 shows an excerpt from the product family ontology created for the P-level.

The linking of the components that are organised within modules can be seen. Different modules form a variant, summarising different variants in a product family. Other data elements are not shown in this excerpt, e.g., development-related data such as working principles or functions, which are also relevant to the ontology.

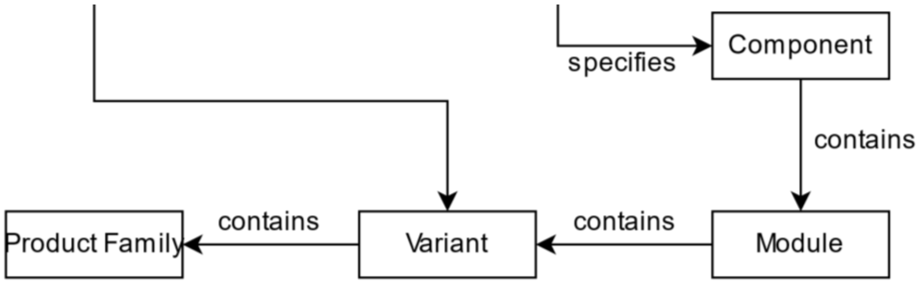

Figure 6. Excerpt of established ontology for Product Families

Meta-modelling

The existing data can be transferred to the modelling environment based on the ontology. SysML is chosen as the modelling language for these purposes. The SysML profile mechanism is used to implement a meta-model for modelling product families. *Cameo Systems Modeller* by *Dassault Systems* is used as a modelling tool. It extends the existing SysML stereotypes to include the context of product family-relevant stereotypes. The modelling is applied to the introduced PSU. Modelling aims to provide easy access to modelling and the reuse of models that describe product families.
To define the meta-model, the elements represented in the ontology are modelled in SysML as ontology elements, including their dependencies. These elements are stored in the *Ontology - Definitions* package. In the next step, all mapped ontology elements are linked to elements of the SysML. If elements already exist within SysML, they are connected to the *OntologyElement* with a representation link. New elements are added using custom stereotypes and linked to the respective *OntologyElements* as well, as shown by the red dotted line in Figure 7. For example, this can be the *Component* stereotype, which is referred to as a *Component* in the scope of the ontology. This generic description provides the base for numerous implementations of product families and can be reused or tailored accordingly.

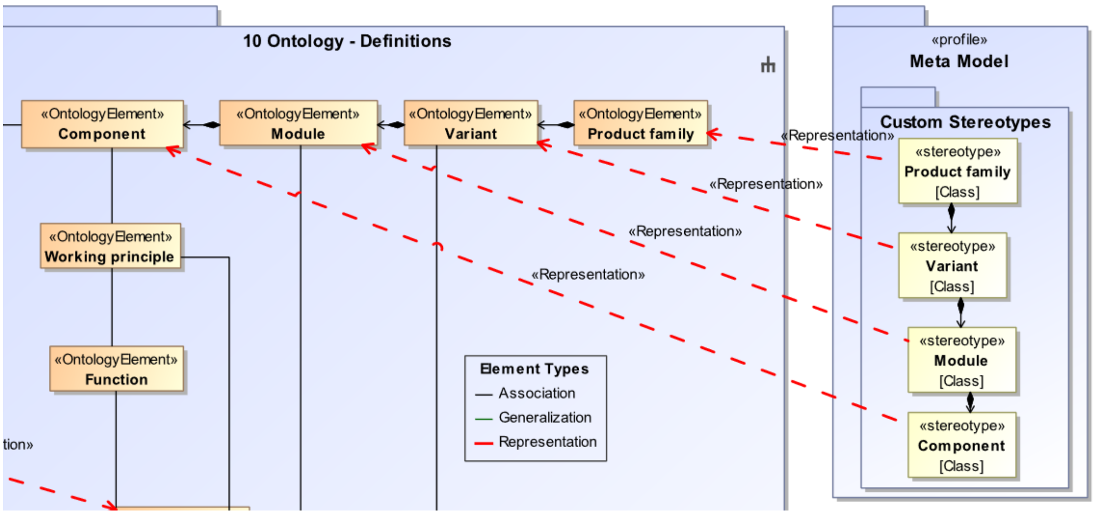

Figure 7. Excerpt of established Meta-Model with the representation of resulting stereotypes in SysML from ontology elements

If some elements do not exist in the native SysML, these can be implemented in SysML using customised stereotypes. The implementation of custom stereotypes is modelled in Figure 7 using the *OntologyElements* already shown. As an example, it can be seen that a module consists of different components and is also applied in different variants. With the help of the red representation link, each ontology element is assigned a stereotype in the *Custom Stereotypes* package. These stereotypes are combined in the *Meta-Model Profile*. Thanks to this abstract description, it is also subsequently possible to clearly trace which elements have which links in the model. This results in a framework that can be used to model further application examples.

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

4.3. System modelling

Based on the meta-model, the model can now be established using a modelling method. A range of modelling methods are available, such as SysMOD (Weilkiens, 2008) or OOSEM (Friedenthal et al.,1998), from which one can be selected to establish the basic information in the model. Creating a custom modelling method is feasible as well. The method is tailored using the meta-model in such a way that it is suitable for modelling modular product families. Hence, the product families of PSU can be modelled using the existing data. In the following, the basic modelling of the product family will not be discussed further. However, a possible further view, which can be used when dealing with modular product families, will be generated based on the underlying system model of the product family. The possible product family view discussed below is that of a so-called product catalogue. Traditionally, offer catalogues are prepared as printed books for the sales department. This catalogue will be generated based on the system model and used to select variants in this implementation. The basis for this is the implementation built on the meta-model, as this is the only way to ensure that the data is fully linked with each other in the previously defined manner.

By holistically linking the development data from the requirements to the single variant in the product family, it is possible to access a single variant via a Meta-Chain through a targeted selection of customer-relevant properties. The linking of data built up in the meta-model, see Figure 3, enables implicit access via the development data modelled in the different layers. The Meta-Chain is a tool to provide such an implicit connection via the modelled relations and connections between different data elements. This has the advantage that if development artefacts are changed, this directly affects the Meta-Chain and, hence, the selection. As no parallel model has to be created for the selection of variants, on the one hand, double work can be saved, and on the other hand, the single source of truth is maintained.

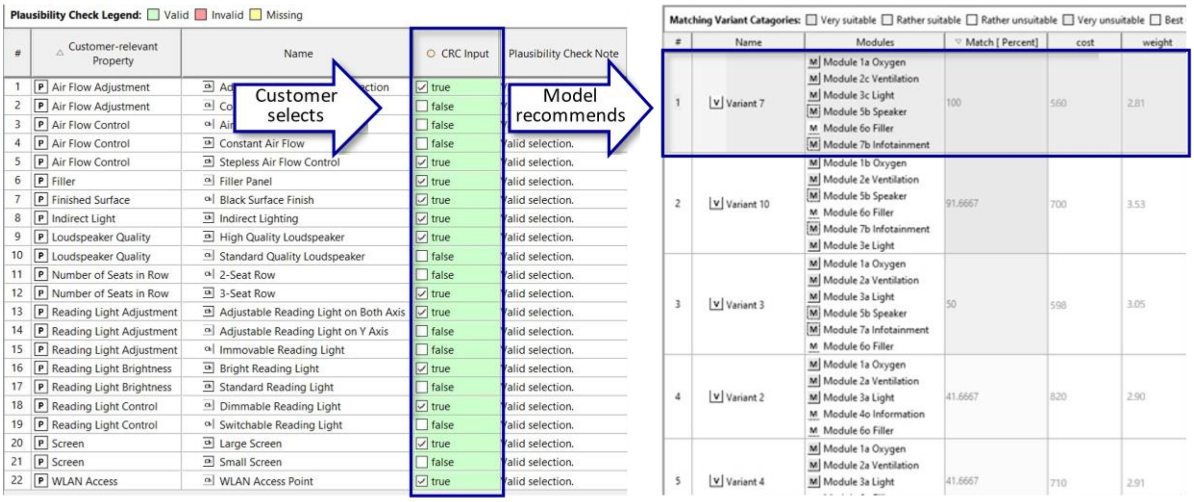

Figure 8. Representation of an offer-catalog in two diagrams, building on the holistic system model of the Product Family

A possible interface for the model is shown on the left in Figure 8. The various customer-relevant properties and their characteristics for the PSU are shown here. Exemplarily different kinds of Air Flows can be selected. Checkboxes can be selected and deselected to choose specific attributes of the product variants. In the background, the input is checked with the help of a plausibility check to see whether, for example, all properties have been selected or no property has been selected twice. If there is an error, the colour of the error is marked in red, and if it is a warning, it is marked in yellow.

In the figure, all entries are green; hence, the selection is complete. Due to the traceability within the model, the selection determines which variant comes closest to the customers' needs. The above-described Meta-Chain determines how many modules were implicitly selected for the different variants via the selected characteristic of the provided customer-relevant properties. This results in the relevance ranking, which is shown on the right-hand side of Figure 8. Here, *Variant 7* of the PSU should be chosen according to the selected product properties. Additionally, all included modules of the variant are

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

displayed. Beyond the relevance ranking, further information, such as the selected variant's costs or weight, can be displayed. The explained product family view is completely built on the modelled system model. Only the two diagrams are additionally modelled. Therefore, the fundamental workload to establish such a view is minimised.

5. Conclusion and outlook

In Systems Engineering, the term Product Line Engineering is often used. This term is frequently associated with the mapping of variety. However, considering variety or variants is only a fraction of a product line's scope and should be linked to the term product family. Especially when developing different variants from a technical perspective, the term product family (engineering) is more appropriate. Product lines can be used more as a superordinate term for the organisational or management perspective. For example, to represent the consideration of different market segments. Accordingly, the contribution presented here introduces a methodical approach to creating a meta-model for modelling modular product families. Based on this, the product family system model can be used to create various additional views that profit from the holistic system model. One of these perspectives was presented based on the exemplary build-up system model. The methodical approach for creating a product family system model based on the introduced meta-model makes it possible to involve all stakeholders in creating this specific model. In particular, using ontologies to define the syntax in a meta-model supports establishing the underlying system model. It provides a tool for discussing the relationships between development artefacts more abstractly. The use case presented has shown that the use of a meta-model offers added value in the systematic preparation of a system model. It is, therefore, an assistance for modelling and leads to a reduction in later modelling time. By using the meta-model and the resulting implied syntax of the various development artefacts, holistic mapping and modelling are easier to implement. The resulting models are comparable and compatible as they use the same scope of data. This modelled data is more easily visible to all stakeholders involved via the meta-model and is, therefore, very useful as a basis for ongoing or new development. This means that models of different product variants can be merged into one holistic product family. Herby, the strategic management of modular product family is improved using the RFLP layers and the Meta-Chain.

In addition, further work has to be done on mapping the product architectures of entire product families to provide additional views of the system model. Usually viewed from a technical perspective, the modules can be enriched with further information, particularly product strategy aspects. Parametric descriptions can also enable recursive calculations in the early design phase. Furthermore, the models can be further developed to validate and verify requirements inside the system model.

Acknowledgements

Parts of this work are part of the research projects InDiCaT (Intelligent Digital Cabin Twin, contract number 20D1902C) and VerDiKa (Vernetzte Digitale Kabine, contract number 20D1923E), both funded by the Federal Ministry for Economic Affairs and Energy as part of the Federal Aeronautical Research Programme LuFo VI-1. The authors would like to thank all involved parties for their funding and support.

References

Aßmann, U., Zschaler, S. and Wagner, G. (2006), "Ontologies, Meta-models, and the Model-Driven Paradigm", in Calero, C. (Ed.), Ontologies for software engineering and software technology: With 46 tables, Springer Verlag, Berlin, Germany, pp. 249–273.

Berschik, M. C.; Blecken, M.; Kumawat, H.; Rath, J.-E.; Krause, D.; God, R.; Schüppstuhl, T. (2022), "A Holistic Aircraft Cabin Meta-Model As An Approach Towards An Interconnected Digitised Cabin Lifecycle", 33rd Congress of the International Council of The Aeronautical Sciences (ICAS 2022), Stockholm, Sweden.

Berschik, M. C.; Laukotka, F.; Hanna, M.; Schwan, L.; Krause, D. (2022), "Digitale Innovationen in der Flugzeugkabinen-Entwicklung", Tagungsband 71. Deutscher Luft- und Raumfahrtkongress (DLRK 2022), Dresden, Germany.

Berschik, M.C., Schumacher, T., Laukotka, F.N., Inkermann, D. and Krause, D. (2023), "MBSE within the engineering design community - an exploratory study", 24th Int. Conf. on Eng. Des., Bordeaux, France.

Delligatti, L. (2014), "SysML distilled: A brief guide to the systems modeling language", Addison-Wesley, Upper Saddle River, NJ, United States.

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press

Eigner, M., Dickopf, T., Apostolov, H., Schaefer, P., Faißt, K.-G. and Keßler, A. (2014), "System Lifecycle Management: Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering", in Fukuda, S., Bernard, A., Gurumoorthy, B. and Bouras, A. (Eds.), Product Lifecycle Management for a Global Market, IFIP Advances in Information and Communication Technology, Vol. 442, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 287–300.

El-Haji, M. (2014), "Ontologie-basierte Definition von Anforderungen an Validierungswerkzeuge in der Fahrzeugtechnik", Dissertation, Karlsruher Schriftenreihe Fahrzeugsystemtechnik, Band 48, KIT Scientific Publishing, Karlsruhe, Germany.

ElMaraghy, W., ElMaraghy, H., Tomiyama, T. and Monostori, L. (2012), "Complexity in engineering design and manufacturing", CIRP Annals, Vol. 61, pp. 793–814.

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

Friedenthal. S., Izumi, L. and Meilich, A. (1998), "Object Oriented Systems Engineering", In: Process Integration for 2000 and Beyond: Systems Engineering and Software Symposium. New Orleans, Lockheed Martin Corporation.

Friedenthal, S., Moore, A. and Steiner, R. (2015), "A Practical Guide to SysML: The System Modeling Language", Third Edition, Elsevier/MK, Burlington, MA, United States.

Graessler, I., Hentze, J. and Bruckmann, T. (2018), "V-MODELS FOR INTERDISCIPLINARY SYSTEMS ENGINEERING", in Proceedings of the DESIGN 2018 15th International Design Conference, May, 21-24, 2018, Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia; The Design Society, Glasgow, UK, pp. 747–756.

Hehenberger, P., Vogel-Heuser, B., Bradley, D., Eynard, B., Tomiyama, T. and Achiche, S. (2016), "Design, modelling, simulation and integration of cyber physical systems: Methods and applications", Computers in Industry, Vol. 82, pp. 273–289.

Holt, J., Perry, S.A., Brownsword, M. (2012), "Model-Based Requirements Engineering", The Institution of Engineering and Technology, London.

INCOSE (2007), "INCOSE Systems Engineering Vision 2020", available 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.

Krause, D. and Gebhardt, N. (2023), "Methodical Development of Modular Product Families", Springer, Berlin.

Lehner, F. (2021), Wissensmanagement: Grundlagen, Methoden und technische Unterstützung, 7. Edition, Hanser, München.

Mertens, K.G., Rennpferdt, C., Greve, E., Krause, D. and Meyer, M. (2022), "Reviewing the intellectual structure of product modularisation: Toward a common view and future research agenda", Journal of Product Innovation Management, Vol. 40, pp. 86-119.

Meyer, M (1997), "Revitalise your product lines through continuous platform renewal", in Research Technology Management, Vol. 40, pp. 17–28.

NoMagic Inc. (2011), "Magic Draw - Model-based Systems Engineering", Online-Brochure (accessed 21 September 2022).

NoMagic Inc. (2023), "Product Line Engineering", available at: https://docs.nomagic.com/display/PLE2021xR2/Product+Line+Engineering (accessed 21 November 2023).

Pure-Systems (2023), "pure::variants", available at: https://www.pure-systems.com/de/purevariants (accessed 21 November 2023)

Rommes, E., America, P. (2006), "A Scenario-Based Method for Software Product Line Architecting", in Käköla, T and Duenas, J. C. (Ed.), Software Product Lines", Springer Verlag, Berlin, pp. 3-52.

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

Tomiyama, T., Lutters, E., Stark, R. and Abramovici, M. (2019), "Development capabilities for smart products", CIRP Annals, Vol. 68, pp. 727–750.

Walden, D.D., Roedler, G.J., Forsberg, K., Hamelin, R.D. and Shortell, T.M. (2015), Systems engineering handbook: A guide for system life cycle processes and activities INCOSE-TP-2003-002-04, 2015, 4th Edition, Wiley, New York, NY, United States.

Weilkiens, T. (2008), "Systems Engineering with SysML / UML", Morgan Kaufmann OMG Press.

Weilkiens, T. (2016), "Variant modeling with SysML", MBSE4U, Fredesdorf.

https://doi.org/10.1017/pds.2024.254 Published online by Cambridge University Press