22nd INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2020

CAMBRIDGE, MASSACHUSETTS, USA, 13 – 15 October, 2020

Model Based Early-Stage Assessment for Modular New Product Development

Jan Küchenhof, Florian Seiler, Dieter Krause

Hamburg University of Technology

Abstract: Higher market dynamics and lower product life cycles lead to an increasing demand in new products which makes changes necessary. Therefore, firms need to comply with technological as well as customer environments and adapt their product portfolio accordingly. This is often costly and associated with high risks as high initial investments may be necessary, especially when following modular product structuring approaches. Deciding on the right interfaces for a modular product architecture comes with high degree of complexity as they strongly affect the products' overall value and market performance. Based upon a product reasoning framework, we propose an adapted calculation scheme for the early-stage assessment during an initial product architecture development. The modeling framework is then implemented in SysML to allow for recursive simulations in order to efficiently assess the decision making in NPD and change environments.

Keywords: Modularization, New Product Development, MBSE, Change Management, Interface Management

1 Introduction

New technologies, changing customer and user needs and global competitors put companies under pressure to innovate. The expansion of the product portfolio, for example through New Product Development (NPD) is an opportunity to meet these challenges. In order to respond to individual requirements and to offer a wide range of product variants with low internal component variety and process complexity, modular product structuring offer a suitable solution (Krause and Gebhardt, 2018). According to Meyer and Utterback, the idea of product families, can serve as a basis for the evolution of firm's core capabilities. Inventive ideas can lay the foundation for initializing a new product development process (Meyer and Utterback, 1992). When initially structuring a product family not only the question how parts of the products offered in the future change but also how the ideal product architecture looks like. Assuming that the core function contains innovation potential through value increase in multiple possible aspects of the products life cycle and its life phases and that it can be beneficial to follow platforming strategies around the core invention and associated know-how as multiple possible products can be designed around. Regarding the whole product structure with its components and modules as clusters of components, there can be advantages and disadvantages for module decisions from different perspectives. Deciding for different modularization can lead to a shift in the product architecture and subsequently influences the value and cost in different design domains. The modeling framework after de Weck, considers different stakeholders as well as the value and cost streams throughout different perspectives (de Weck, 2007). This model will be used for estimating the depth of value for supporting product architecture decisions. The modeling framework is supplemented by new relations for the considered design environment and implemented in SysML with the model based systems engineering (MBSE) modeling software Cameo Systems Modeler (CSM) in order to run simulations with different boundary conditions fitting the considered product structure alternatives and associated stakeholders. In Section 2, concepts of modularization and innovation are introduced and put into the context of the NPD. Since matrix-based software support is established to analyze product structures as it allows the integration of the different necessary data types, a short introduction to MBSE in order to implement, simulate and evaluate a multi-dimensional data framework is presented. Section 3 shows the adaptations made to the subsequent framework according to the displayed necessities. In Section 4 the presented approach is discussed and an outlook is given.

2 State of the Art

Product properties and product design can be influenced most strongly in the first phases of the product life cycle, the effects of design decisions taken, on the other hand, often only become apparent much later (Ehrlenspiel and Meerkamm, 2013). It is also known that product design decisions influence many other important aspects during the product life phases such as production and marketing means (Krause and Gebhardt, 2018). Especially modularization and platform approaches are associated with higher initial investments but may be cost effective in the long run due to long-term complexity reduction.

2.1 Modularization and Innovation

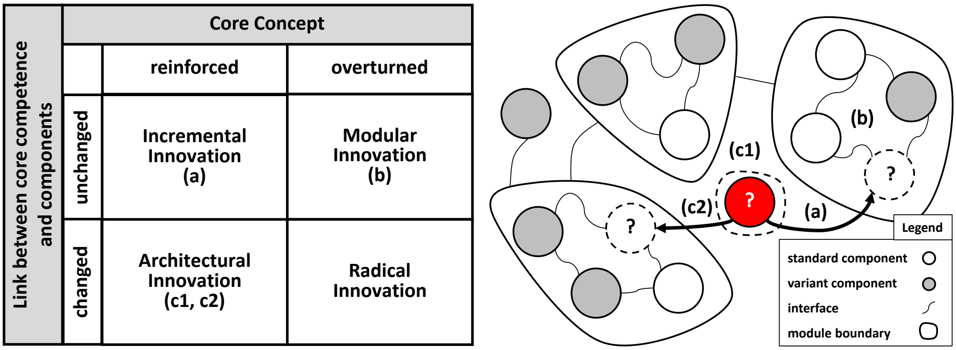
Modularization approaches are mostly carried out on existing product structures where the knowledge-base is high and effects on decisions can be estimated quite accurately. In order to not only accomplish economies of scales by implementing well-known concepts of commonality but also economies of scope and integrate concepts of variety and customization, flexibility and new ways of collaborative interaction, (e.g. with partner, suppliers), modular product structuring as an NPD strategy is becoming increasingly popular (Mikkola, 2006). The aforementioned design paradox gains in importance as early design decision can have high impact on product success on the market and investments for modular platforms are generally higher, why market success and associated business revenue is crucial for firm's viability. The main difference between conventional product design and modular product design is that the product architecture is traditionally the result of design activities, whereas in the latter case it is the input as the interfaces are defined in the beginning (Sanchez, 1996). Following the well-known definition after Ulrich, the product architecture is the scheme in which the products' functions are allocated to physical components. It can be defined as the arrangement of functional elements, the mapping from functional to physical elements and the specification of the interfaces among the components (Ulrich, 1995). Ulrich defines that a modular architecture exhibits a one-to-one mapping from functional elements to physical components and that integral architecture exhibit complex mappings.

In order to obtain a modular product structure, different approaches can be chosen. A generic procedure of modularization is explained in four steps in Krause and Gebhardt (2018). *Step 1* describes the decomposition of the existing product structure (often hierarchical structuring). In *step 2*, the components are analyzed regarding technical-functional and product strategic module drivers. In *step 3*, the actual modularization takes place on the chosen level of analysis by forming modules based on the identified module drivers in the precedent step. The newly modularized product structure is finally implemented in *step 4*. (Krause and Gebhardt, 2018). *The Integrated PKT-Approach for the Development of Modular Product Families* aims to reduce internal product and process variety while maintaining the external product diversity. It consists of two main methods in order to reduce complexity, which can be referred to as *step 1* within the generic procedure. First a Design for Variety (DfV) is performed in order to prepare the components for the subsequent Life-Phase Modularization which considers technical-functional as well as product-strategic aspects (Krause and Gebhardt, 2018). An adapted NPD approach considering future variety in product properties and mapping them to product functions, principle solutions and physical components in an extended design model has been presented in (Küchenhof and Krause, 2019). The used models within the design method have been transferred into a matrix representation and supported by means of graph analysis in Küchenhof et al.. The influence of external changes on the structure of a product family is assessed by calculating the activity and passivity of the components and their centrality values (Küchenhof et al., 2020a). In order to reach data consistency and traceability the used development models to support the tailored DfV are implemented in SysML (Küchenhof et al., 2019). The matrix-based approach to represent the cross-domain model has been extended in order to represent product family generations. The changes induced by the new design objectives are estimated by change propagation paths that are retraced with help of graph analysis software. The occurring changes necessary for developing the second product generation can be comprehended in adjacency matrices in CSM after implementing the data frameworks of both product family generations in SysML with CSM (Küchenhof et al., 2020b).

While the procedure of modularization can be applied to new development projects, major differences are found in *step 1* and *step 4*. Since there are no internal structures for decomposition, these must first be created. Thus, external knowledge must be transferred to internal knowledge, e.g. by benchmarking and analyzing existing products and product structures within the relevant market segments. The implementation in *step 4* is associated with high investments, the choice of process architecture, such as design of product lines and machinery are a high risk within entrepreneurial activities as resources are generally low and mis-investments can become a quick failure of the start-up. Increasing market dynamics require more flexibility in product and process design, which on the one hand increases the already existing uncertainty in the early phase and on the other hand makes the provision of change mechanisms for later adaptations increasingly important. Poorly implemented structures often cause high change costs which can be significantly higher than the implementation costs, in the long run (Zwerink et al., 2007).

When designing products one can classify product development projects into new design, adaptive design and variant design projects varying from their degree of novelty. A new product generation is thus consisting of subsystems that are the result of variation in order to carryover, subsystems that are new developed with embodiment variation or principle variation (Albers and Bursac, 2015). Within a new product generation, several new development parts of the subsystems of a product can be distinguished. The new development of a subsystem of a product generation through principle variation, for example by adaptation from products that have similar functions and properties in other contexts, or through the systematic search for alternative solutions. The new development of a subsystem through design variation, in which a known (and usually proven) solution principle is derived from a reference product or from the general state of the art is adopted and the function-determining properties are varied in such a way that an increase in competitiveness, performance and/or the quality of the functional performance is possible. The shape variation is the most frequent product development activity and is also a highly creative and complex process. In addition, subsystems are adapted as transfer variations, that is existing solutions from reference products or from suppliers into new generations of products and according to the requirements of system integration adapted to the interfaces. This activity is described as a transfer variation and the constructive adjustments should be minimized as far as possible.

While different views on innovation exist, Henderson and Clark complement the traditional categorization of innovation in incremental or radical by examining the product structure in terms of their components and their linkages, integrating the role of the product architecture. The different types of innovations are classified along two dimensions as shown in Figure 1 on the left. The horizontal dimension shows the impact of innovations on components and the vertical axis shows its impact on the linkages between them. Incremental innovation introduces small technological changes to the existing product, mostly improving certain components and embedded functions. Innovation that changes the core design concepts of a technology can be understood as modular innovation, the relationships between modules stay the same. The reconfiguration of an established system to link existing components in a new way is described as architectural innovation, that is often triggered by a change in a component. (Henderson and Clark, 1990).



Link between core competence and components		Core Concept	
		reinforced	overturned
	unchanged	Incremental Innovation (a)	Modular Innovation (b)
	changed	Architectural Innovation (c1, c2)	Radical Innovation

Figure 10: Left: A framework for defining innovation after Clark and Henderson (1996); Right: Exemplary modular product structure design decision considering the different types of innovation

An exemplary design decision problem is depicted in Figure 1 on the right on the example of a generic modular product structure. The components are differentiated into standard components (white), that are used commonly along the product family and variant components (grey), that inhibit variant technical properties, induced by differentiating product properties to integrate product variety. The linkages of the components can differ in its relations (e.g. mechanical, energetic) but are abstracted in this picture. The module boundaries are indicated and the core component exhibits inventive technological progress is highlighted as a red module as it is the pivot point for the further design question. The different types of innovation are related to the product structure as the authors understand the concept of structural innovation. The different possible positions for the core component are referred to by the question marks. Path **(a)** describes the concept of an incremental innovation as the core component is improved but the structure of the product remains unchanged as competing products would show the same. If the core concept of the whole module **(b)** can be changed by introducing the core component, one would classify it as a modular innovation following the definition above. Considering the core component as an autonomous module **(c1)** or moving it to the other module **(c2)** would change the product structure as new interfaces are needed. This has a significant impact on the product architecture, as functions are assigned to other modules and other components may have to fulfil other functions that would previously be fulfilled by others. The question where to move the core component is not trivial as business success may be very dependent from that choice. While incremental or modular innovation strategies may be easier to design and to implement, architectural innovation can lead to completely new product designs that may find high interest on the market but may also be rejected by the customers as they overstrained by new using behaviors. The engineering effort may be also higher for reaching technical feasibility for newer concepts, but a major are the subsequent investments of resources and capacities following early phase design decisions. This conflict will be addressed in the following with a simulation model in order to quickly estimate the effects of decisions from different perspectives.

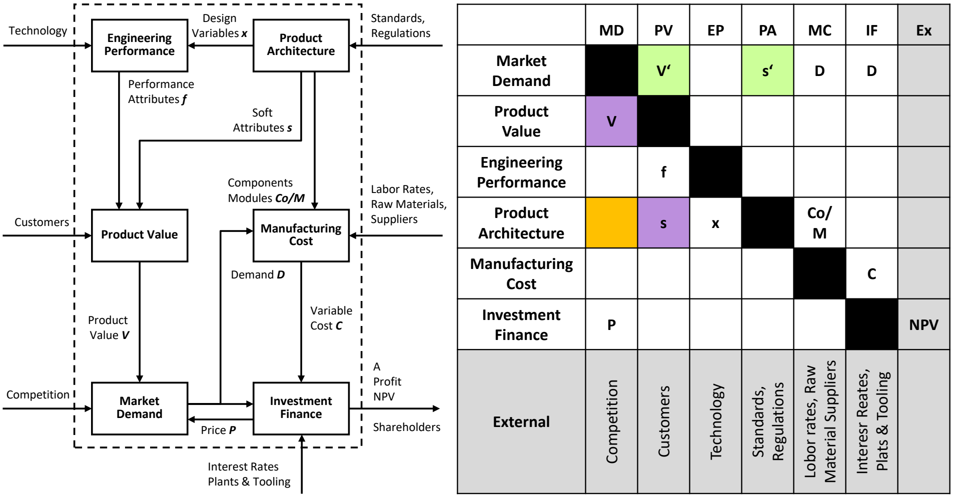
In order to determine the platform extinct of product platforms, which represents the upper and lower boundaries of the totality of product variants within a product family, de Weck proposes a product reasoning framework, pictured in Figure 2 (left) (de Weck, 2007). De Weck takes into account different domains such as Market Demand (MD), Product Value (PV), Engineering Performance (EP), Product Architecture (PA), Manufacturing Cost (MC), Investment Finance (IF) as well as external inputs and outputs. The different domains, portrayed in the block diagram, are connected by certain attributes such as Performance Attributes (f) connecting engineering performance and product value or performance attributes that the product architecture contributes to the product value. This modeling framework sets the basis for the upcoming simulation. The extended framework is then implemented in SysML. The basics are explained briefly in the following.

2.2 Data Handling of Complex Systems with Model Based System Engineering

Existing models, such as CAD or FEM, which are classically manually implemented and maintained as well as updated, are therefore prone to errors and inconsistencies and make changes costly. These problems of inconsistency are particularly common with methods for developing modular product families, as they use a large amount of data that is not consistently linked (Hanna et al., 2018). In the context of consistency management, model theory can help by using a data model. A software-supported implementation of a data model can be made possible by MBSE. A model is an abstraction of reality and has three main characteristics. Representation means that models represent something; reduction means that not all attributes are represented in the model, but only the relevant ones; pragmatism means that models are not uniquely assigned to their originals. A metamodel is itself a model that can be used to describe modeling (Hanna et al., 2020; Holt et al., 2012). Considering the requirements mentioned above, (Hanna et al., 2018) developed a MBSE-data model with the goal of keeping the three levels of consistency high: temporal continuity, hierarchical continuity and consistency. For a model-based support of product development, MBSE was developed using abstract system models (Holt et al., 2012). With MBSE, system elements can be modeled and information can be linked so that information can be stored and used in a networked model. The SysML language was developed for MBSE. The modeling software CSM uses SysML notation as well as diagrams and tables. It is based on nine diagrams, five of which are used to create the structure and four to create the behavior (Partsch et al., 2010). Using SysML, an object-oriented meta model of an underlying structure, such as e.g. a modular product architecture (MPA), can be semantically expressed, enabling the numerical processing of the implemented information. For these specific tasks when it comes to adding relevant MPA data to the underlying MBSE data framework, a data value vector has been developed. This vector holds the relevant information for each component, module or product variant, providing the algebraic basis for further applications, such as e.g. product configuration systems or the analysis of MPAs (Seiler and Krause, 2020).

3 Differences & Implementation

The modeling framework used can also be depicted in matrix view (Figure 2, right) and serve as the adjacency matrix in CSM. The matrix is read as rows serve inputs for each column.



	MD	PV	EP	PA	MC	IF	Ex
Market Demand	■	V‘		s‘	D	D	
Product Value	V	■					
Engineering Performance		f	■				
Product Architecture	🟧	s	x	■	Co/ M		
Manufacturing Cost					■	C	
Investment Finance	P					■	NPV
External	Competition	Customers	Technology	Standards, Regulations	Lobor rates, Raw Material Suppliers	Interesr Reates, Plats & Tooling	

Figure 11: Left: Single Product Reasoning Framework (de Weck, 2007); Right: Matrix representation with recursion on Market Demand

While the unidirectional flow from product architecture to product value and further to the market demand can be comprehended (V, s; purple fields), we see greater potential in integrating a recursive data flow from market demand and product value to the product architecture (V', s'; green fields). In this way, the correlation between market demand in the form of the necessary product diversity, which is increasing, and product architecture (orange field) can be reconciled with changes in both areas.

Since the product architecture is set up with regard to the external offer variety in our case, identified customer relevant, differentiating product properties directly affect the internal component variety. Offered product variants increase the sales offer and leverage revenue from platform derivatives, thus enabling long term cost efficacies. The increasing market dynamics make it necessary to respond more frequently with improved components, modules or even making structural changes and enabling new product solutions. The bidirectional data flow can be seen in the matrix as the integrated entries are highlighted. The simulation can now show the influence of changing market demand onto different product structure alternatives. As already described above, this results in analyzing and cross-calculating large data amounts as well as different data types and structures. We therefore propose a software-based approach that takes these requirements into account. As consistency is also a highly important requirement, the use of MBSE provides one suitable solution for these issues. As during the early stages of MPA development, cost and time consumption are crucial, especially when entering new market areas, relying on a resilient simulation proposes various advantages, such as simulating various alternative market-to-product scenarios with little impact on the actual resources as well as a failure-resistant study design (Hermann et al., 2013).

The for the exemplary implementation of this framework used software environment is CSM, offering the possibility to model any desired scenario, as long as it can be mathematically expressed as a parametric state diagram. The following figure 3 shows the modelled framework. There are mainly four different parts of which this parametric simulation diagram consists. On the one hand (upper left corner) the market input scenario and on the other hand (upper right corner) the product costs based upon the individual companies internal parametric. The lower right corner contains the input from various early-stage developed MPA alternatives. These three main input parts are then cross-calculated in the overall lover constraint part according to the mathematical basis provided by de Weck resulting in the overall value and thus the performance of each MPA alternative. Since different scenarios are to be analyzed, all relevant input parameters are implemented as changeable parameters. These are the main input values for the calculation of the demand's elasticity (k), consisting of the average competitive demand (acD) and the average competitive price (acP). Together with k, the product price (P) and the perceived product value (V) are combined according to de Weck to mathematically express the final market demand. On the other hand, the product costs consist of production costs (pC), engineering performance costs (epC), supply chain costs (scpC) and assembly costs (aC). These are all individually adaptable as the input scenarios change. In order to implement the various MPA alternative structures and therefore the alternative module variants, the data structure systematic developed by (Hanna et al., 2018) is used.

This root data structure allows for the conjoint implementation of customer relevant properties as well as their corresponding modules with their individual degree of variety and the adjacent components. As these data are also embedded within the MBSE environment, the consistency as well as the addressability by the simulation are enabled. These three input factors are then used to derive the individual scenario's impact on the MPA alternatives' performance in the customer-producer environment. By comparing the simulations output results, a quantitative and reproduceable supporting factor for the choosing about which further development path to consider is given. One major factor when analyzing the MPA's performance is described as "Configuration depth" (Seiler et al., 2019). This factor has been developed in terms of the presented contribution and expresses the normalized degree of customer requirement fulfilment by the accordingly determined product variant and is described by the following equation 1.

$$E_{MPA} = \frac{\sum \; g_i \; * M \begin{pmatrix} KWV_1 \\ KWV_2 \\ \dots \end{pmatrix}}{n_{Module}}$$

With a full configuration depth of 100%, the determined product variant is able to meet all initially defined customer requirements. As this is a gradual factor, the resulting values are strongly dependent from the number of customer requirements and their possible degree of fulfilment.

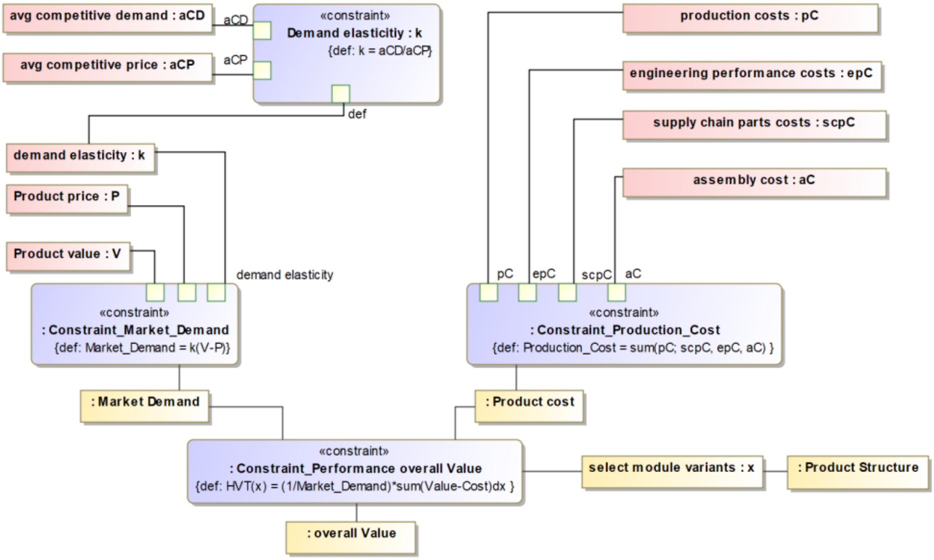

avg competitive demand : aCD
aCD
avg competitive price : aCP
aCP
«constraint»
Demand elasticitiy : k
{def: k = aCD/aCP}
def
demand elasticity : k
Product price : P
Product value : V
demand elasticity
«constraint»
: Constraint_Market_Demand
{def: Market_Demand = k(V-P)}
: Market Demand
production costs : pC
engineering performance costs : epC
supply chain parts costs : scpC
assembly cost : aC
pC
epC
scpC
aC
«constraint»
: Constraint_Production_Cost
{def: Production_Cost = sum(pC; scpC, epC, aC) }
: Product cost
«constraint»
: Constraint_Performance overall Value
{def: HVT(x) = (1/Market_Demand)*sum(Value-Cost)dx }
select module variants : x
: Product Structure
: overall Value

Figure 12: MBSE-parametric simulation for interconnected overall value simulation in CSM

The weighting is generally done by using a linear language-to-number transverse scale, as customers tend to implicitly express the relevance of a customer-relevant property in three stages: not important, intermediately important, important. These statements are then transformed into a linear number scale reaching from 1 to 3 with 3 expressing the highest degree of importance. This configuration depth factor therefore is one major part of the proposed simulation's output, which is then returned as being included into the overall value result.

4 Discussion & Outlook

With the previously described MBSE simulation as a basis for the early-stage development of modular product architectures, a cost efficient and target-oriented approach can be realized. Especially the consideration of the variance and modular granularity is one key influencing factor resulting from the simulation when implementing a real product example. Furthermore, a major improvement to the original framework by de Weck is the bidirectional effect analysis enabled by the consistent simulation. As the modelled effects and parameters are not solid-state values but interdependent data sets influencing directly or indirectly both market and company perspective, a clear differentiation and effect analysis can be performed. With the recursive design of the modelled simulation, clear visualizations of the market demanding for product change can be seen. Secondly, with the knowledge of these change demands and the simulated change processes to the product itself, the market's reaction due to an altered fulfilment set of customer-relevant properties changes as well. This interdependency is only possible using a bilateral, central root data model with dynamic constraint parameters. Furthermore, when implementing the underlying product structure alternatives, the critical paths and most impacting connections between modules and customer-relevant properties can be identified, offering a supporting method for identifying the negative effect drivers within the product architecture and to identify tendencies about the market and company environments behaviour.

On the basis of the proposed simulation, it is now necessary to analyse structural product architecture alternatives using a real product example, to estimate effects initially and to demonstrate their effectiveness after implementation. As an additional added value, a comprehensive market environment simulation is considered useful in order to be able to depict the increasing market dynamics cost-efficiently and realistically on the one hand, and to be able to quickly draw conclusions about the role of the product architecture, in particular its components and interfaces. As a result, the innovation cycle in NPD processes can be supported efficiently and holistically, and the knowledge about the effect characteristics contributes to risk minimization which is especially important for modularization projects with high investment efforts or start up structures with typically low resources. To support new value creation, digital shares can be further examined in terms of both products and services. This requires the development and structuring of service offers and the design of suitable interfaces.

References

Albers, A., Bursac, N., 2015. Produktgenerationsentwicklung – Bedeutung und Herausforderungen aus einer entwicklungsmethodischen Perspektive. SSP 2015, Stuttgart, Germany.

De Weck, O.L., 2007. Determining Product Platform Extent. Product Platform and Product Family Design. New York: Springer.

Ehrlenspiel, K., Meerkamm, H., 2013. Integrierte Produktentwicklung: Denkabläufe, Methodeneinsatz. München: Hanser.

Henderson, M.H., Clark, K.B.,1990. Architectural Innovation: The Reconfiguraiton of Existing Product Technologies and the Failure of Established Firms. Adm. Sci. Quarterly, 36, 9-30.

Hanna, M.; Schwede, L.-N., Krause, D., 2018. Model-Based Consistency for Design for Variety and Modularization, 20th Int. DSM Conference, DSM 2018, Trieste, Italy. https://doi.org/10.35199/dsm2019.11

Hanna, M, Schwenke, J., Krause, D., 2020. Inconsistency Management For Product Families With Many Variants Through a Model-Based Approach in Modular Lightweight Design. International Design Conference - DESIGN 2020, Dubrovnik, Croatia. https://doi.org/10.1017/dsi.2019.367

Holt, J.; Perry, S.; Brownsword, M., 2012. Model-based requirements engineering. IEEE, The Institution of Engineering and Technology, London. https://doi.org/10.1109/JSYST.2014.2312051

Hermann, M.O., Michler, J., Schönthaler, F., 2013. Wo Kundenwünsche auf technische und wirtschaftliche Notwendigkeiten treffen. Business News 03/2013.

Holt, J., Perry, S., 2014. Sysml for systems engineering: A model-based approach (2. ed.). Professional applications of computing series: Vol. 10. London: Inst. of Eng. and Tech..

Krause, D., Gebhardt, N., 2018. Methodische Entwicklung modularer Produktfamilien: Hohe Produktvielfalt beherrschbar entwickeln. Berlin: Springer Verlag. https://doi.org/10.1007/978-3-662-53040-5

Küchenhof, J., Krause, D., 2019. Entwicklung eines Produktarchitekturmodells zur Ableitung modularer Produktstrukturen. 30th Symposium DFX. Hamburg, Germany. https://doi.org/10.35199/dfx2019.3

Küchenhof, J., Schwede, L.-N., Hanna, M., Krause, D., 2019. From Visualizations to Matrices – Methodical support for New Development of Modular Product Families. 21st Int. DSM Conference, Monterey, USA. https://doi.org/10.35199/dsm2019.11

Küchenhof, J., Tabel, C., Krause, D., 2020a. Assessing the Influence of Generational Variety on Product Family Structures. CIRP Design Conference, Skukuza South-Africa.

Küchenhof, J., Schwede, L.-N., Krause, D., 2020b. Planning & Tracking the Changes – Matrix Mapping of Modular Product Family Generations. NordDesign 2020, Lyngby, Denmark.

Meyer, M.H., Utterback, J.M., 1992. The Product Family and the Dynamics of Core Capability. Massachusetts Institute of Technology, USA.

Mikkola, J.H., 2006. Capturing the Degree of Modularity Embedded in Product Architectures. Journal of Product Innovation Management 23, pp. 128-146.

Partsch, H., 2010. Requirements-Engineering systematisch, Springer-Verlag, Berlin. https://doi.org/10.1007/978-3-642-05358-0_1

Sanchez, R., 1996. Strategic product creation: Managing new interactions of technology, markets, and organizations. European Management Journal, 14 (2), 121–138. https://doi.org/10.1016/0263-2373(95)00056-9

Seiler, F., Schwede, L.-N., Krause, D., 2019. MBSE-basierte Produktkonfiguratoren zur Analyse der Modularisierung. EEE Conference, Vol. 2, Dresden.

Seiler, F.; Krause, D.: „A Multi-Dimensional Configuration Algorithm for Modular Product Architectures. International Design Conference - DESIGN 2020, Dubrovnik, Croatia, 2020.

Ulrich, K., 1995. The role of product architecture in the manufacturing firm. RP 24, pp. 419-440.

Zwerink, R., Wouters, M., Hissel, P., Kerssens-van Drongelen, I., 2007. Cost management and cross-functional communication through product architectures. R&D Management, 37: 49-64. doi:10.1111/j.1467-9310.2007.00458.x

Contact: Jan Küchenhof, Institute of Product Development and Mechanical Engineering Design/ Hamburg University of Technology, Denickestraße 17, 21073, Hamburg, Germany, +49-40-42878-2176, , jan.kuechenhof@tuhh.de, https://www.tuhh.de/pkt/