27th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2025
HOBOKEN, NJ, USA, 24 – 26 September, 2025

Formalizing Constraint-Based Configuration of Variant-Rich Products through Dependency Modeling using Multiple-Domain Matrices

Thorsten Schmidt[1], Fionn Winger[2], Daniel Roth[2], Matthias Kreimeyer[2], Frank Mantwill[1]

[1]Helmut Schmidt University, Chair for Machine Elements and Computer Aided Product Design, Hamburg, Germany
[2]University of Stuttgart, Institute for Engineering Design and Industrial Design, Stuttgart, Germany

Abstract: To meet customer-specific requirements, mass-customizable products are available for configuration. The combination of defining features is deterministically specified by underlying constraints, which need to be considered during the whole configuration process. A propositional solver can be used to validate complex dependencies and guarantee consistency. To model these complex dependencies, this paper proposes a matrix-based approach to represent configuration knowledge in associated Function-, Feature-, and Component-Design Structure Matrices (DSMs) connected in an overall Multiple-Domain Matrix (MDM). In the presented approach, a product is first decomposed into its sub-functions using the METUS method. The sub-functions are translated into features and then verified pairwise for logical compatibility against the underlying constraints. A valid and complete configuration triggers the selection of components through part validities, which are then combined into modules and derived into a product structure. Potential optimization of the product structure is highlighted by adapting features and elaborating changes in constraints.

Keywords: Dependency Structure Modelling, Product Architecture Optimization, Complex System Design, Design for Variants, Constraint-Based Configuration

1 Introduction

To meet customer-specific requirements, mass customization approaches are used for individualized products that are developed and manufactured using mass production methods in a configuration process. For this purpose, variance is considered in product development from an external customer perspective and an internal development perspective. Between these two perspectives, the product is characteristically described by features and constraints to ultimately translate functions into assigned components for their realization. The combinatorics results in variant-rich products that are implicitly described by constraints and realized through e.g. modular product architectures, feature family engineering and common part strategies. A prominent example of constraint-based, variant-rich products build upon shared platform and modular manufacturing concepts, and assembly line production is the automotive industry, particularly the premium sector of the German automotive industry. Dealing with variant-induced complexity requires the use of tools, assistances and decision support systems to support the designer, for example summarized in the “Variant Management Toolbox” (Braun & Strattner, 2017). Variant management is therefore considered to be a subcategory of complexity management (Deubzer et al., 2008). Tools and assistants, such as configurators, are used to avoid, reduce and control the induced complexity. For the visual understanding of dependencies given in Boolean algebra, matrices are used to model pairwise dependencies. Since the 1980s, these matrices and adaptations thereof have been named Design Structure Matrices (DSMs).

To understand variant-rich product description, a product is distinguished in functional-, feature-, and component level. The customer buys a validated function, which must first be broken down into sub-functions and elementary functions (decomposition). These functions are realized by interacting components, which are defined in the product architecture. The variant description happens at feature level, where the configuration takes place and the feasibility check is performed. A valid feature-based configuration is resolved by interpreting installation condition in the Bill-of-Material (BOM). Assigning elementary functions to features is conceivable but involves a great amount of manual effort and is prone to errors. Various domain boundaries are crossed, and development must involve interdisciplinary cooperation in the early phase. When modeling in complex system design, complex interrelationships must be taken into account, and architectural changes must be enabled in order to remain competitive and be able to react to changing requirements (Soltan et al., 2019).

The continuous documentation of variant-rich products across interdisciplinary domains is a matter of current research need. Especially the translation of decomposed functions from the customers' perspective to the internal component view of the development in a sophisticated product architecture is incomplete and lacks methodological approaches that covers the consideration of logical constraints along the configuration process. Existing approaches like METUS do not consider feasibility checks of Boolean constraints. Logic solvers used for compatibility do not support visual interpretation of dependencies and therefore lack explainable results. Combined, there is an identified research need for methodological product architecture support while satisfying given constraints and foster product structure optimization across domains in a matrix-based notation. Concluding from the above-mentioned need for research, the research question of this paper is: How can the constraint-based configuration of variant-rich products be modelled in a matrix-based format and what potentials exist for product architecture optimization?

This contribution applies the METUS method to describe the product architecture and the function decomposition, as well as the transfer of the configuration problem of multi-variant products into function-, feature-, and component DSMs and a combined Multi-Domain Matrix (MDM). To model the dependencies and fill the feature DSMs, a novel approach from constraint solving is used to translate Boolean constraints into pairwise dependencies in order to ensure satisfiability. The features are then translated to parts and components and summarized to modules and assemblies and finally to a product variant. This provides a transparent approach to the holistic consideration of the configuration problem from the customer to the finished product by representing the configuration problem in several DSMs. This approach enables consistent and feasible configuration and offers the possibility to optimize the product structure by adding new dependencies in the form of shared functions, features, constraints or component decisions and to consider changes over time with less effort. Bringing experience from designers and documenting them into engineering tools like dependency matrices is part of 'Knowledge-based Engineering' (KBE) (Whitney et al., 1999). This contributes to a standardized notation, transparent, comprehensible, interpretable and machine-readable approach to support product development, which increases data quality and consistency and can be visually extended to graph-based methods and sensitivity analysis of changing dependencies between elements and the addition of new configuration options.

This paper is divided into six chapters. An overview of the state of the art in feature-based product description, the METUS method and dependency structure modelling is followed by the used methodology. Subsequently, a concept for formalization is presented and then discussed against the background of product architecture optimization. Finally, the results are summarized.

2 State of the Art

This chapter describes the state of the art for variant-rich product documentation of feature-based products, the concept of the METUS method for function decomposition and structuring of the product architecture as well as the matrix-based approaches of DSM, MDM and DMM for modelling dependencies in complex system design.

2.1 Constraint-based and feature-based product description with high variance

Variant-rich products are uniquely described on the basis of their characteristic features and are defined by the combination of features values that are restricted by additional constraints (Eigner & Zagel, 2007). Features, components and functions need to be separated in different perspectives on the same product (Kesper, 2012). The continuous variant modelling starts from the external customer view from features, feature values and dependencies to the internal view of the product structure and the components used to realize the product variants documented in a BOM (Braun & Strattner, 2017). Due to the management through features, this is referred to as development according to the principle of 'feature engineering'. Features are realized through a common product architecture and reused parts and modules. They are designed according to the feature family model in product portfolios. In this process, a customer selects features from a catalogue of options, often supported by a configurator. Development is based on the principle of variant design, in which synergy effects are realized by exchanging modules and reusing other modules on the basis of a modular architecture. The tailoring of modules to create an expandable modular product architecture is described in Lee et al. (2025). The definition and control of variance in variant-rich product portfolios is the subject of configuration management. Product portfolios often consist of product family lines, which are developed according to the principle of 'product line engineering'. In addition to the 'right' number of product variants, the right product structure is also significant in configuration management (Eigner & Zagel, 2007). Constraints are part of the design knowledge and can be represented as well as captured in DSMs (Germani et al., 2006). Configurators are used to guide a customer to their desired product. The whole point of using a configurator is hereby to find a compatible set of features from a customer view and translate them to a matching set of required parts and modules in a BOM (Helo, 2006). Therefore, underlying design knowledge and technical constraints need to be considered at all times - for example by using a DSM to model dependencies. Table 1 exemplary illustrates two features with two feature values each given by their three-digit code (so called "primary number") and a short description.

Table 1. Excerpt of feature catalog for the feature-based product description according to von Eisenhart Rothe (2002)

Feature Catalogue		
Feature	Feature Value	Description
TRW (Ger.: "*Trennwand*")		Partition wall
	3CA	Without partition
	3CX	Mesh partition
LBH (Ger.: "*Ladeboden Heck*")		Loading floor
	3GA	Without loading floor
	3GN	Variable loading floor

Dependencies between features values are formulated as commandments and prohibitions, as shown by an exemplary dependency between the partition wall and the loading floor in Equation 1.

Constraint as commandment: "Mesh partition" forces "Variable loading floor": $3CX\ Z\rightarrow 3GN$ *(1)*

According to the feature family concept, a prohibition can be transformed into a commandment with the same logical statement and vice versa. The statement from Equation 1 is formulated as a prohibition in Equation 2.

Constraint as prohibition: "Mesh partition" prohibits "Without loading floor": $3CX\ V\rightarrow 3GA$ *(2)*

Both statements are formulated as logical expressions using Boolean operators AND, OR, NOT (!) shown in Equation 3.

Equation (1) and (2) in Conjunctive Normal Form: 3CX AND (3GN OR 3GA) AND (!3GN OR !3GA) *(3)*

With the help of a propositional logic solver, the logical expression can be interpreted and returned as true or false. Equation 4 shows the two logical statements for a satisfiable and unsatisfiable configuration in a pairwise comparison.

Propositional logic solver (SAT): $SAT(3CX, 3GA) = false;$ $SAT(3CX, 3GN) = true;$ *(4)*

Each valid combination of feature values corresponds to a configurable product variant. To manage the required parts, components and derived modules and assemblies, the BOM uses part validities (Ger.: "*Teilegültigkeiten*" - TeGü) or part selection rules (Ger.: "*Teileauswahlregeln*" - TAR). Table 2 shows an excerpt from a constraint-based complex Bill-of-Material (cBOM) for controlling the loading floor depending on the configuration.

Table 2. Excerpt of a complex BOM according to von Eisenhart Rothe (2002)

Part Number	Description	Part Validity	Quantity
1J0.343.132.A	Load compartment trim left	3CA+1XO+3GA+7B2+K8D/K8M	1
1J0.343.132.B	Load compartment trim left	3CA+1X0+3GN+7B2+K8D/K8M	1
1J0.343.132.J	Load compartment trim left	3CX+1CX+3GA+7B0+K8D	1

2.2 Product structure and function decomposition using METUS

A product architecture can be summarized as functional- and product structure. Product architectures include both functional and physical terms and define the scheme by which parts, components and assemblies implement the function of a product (Ulrich & Eppinger, 2004). Product architecture is represented, for example, by the METUS rhombus (see Beitz et al., 2013, pp. 257). The Methodical Support for System Creation (Ger.: "*METhodische Unterstützung zur Systembildung*" - METUS) enables the visualization of functional dependencies and product structure in the form of a rhombus as shown in Figure 1. METUS was originally developed by Daimler AG in cooperation with ID-Consult GmbH and is now marketed by pwc as a software tool for continuous support of the configuration process. Starting from an overall function on the left, a functional decomposition is performed on sub-functions and elementary functions, which are matched to components in the center of the rhombus, which in turn are combined into modules and assemblies and structured into an end product on the right. The METUS rhombus supports the visualization of variant drivers (Kesper, 2012). Since only one product variant can be mapped at a time, METUS is useful for product architecture but not suitable for mapping products with many variants. In addition, matching functions to components involves manual effort. METUS is one of the methods used in the development of modular product architectures. Modularity of product platforms is one of the key characteristics of competitive architectures (Germani & Mandorli, 2004). The variety of a product is defined as 'the range of product models the firm can produce within a particular time period in response to market demand' (Ulrich & Eppinger, 2004, pp. 190) and is realized by modularized components and standardized interfaces. The generation of products from functional decomposition is further addressed in Al Handawi et al. (2024) and Müller et al. (2019).

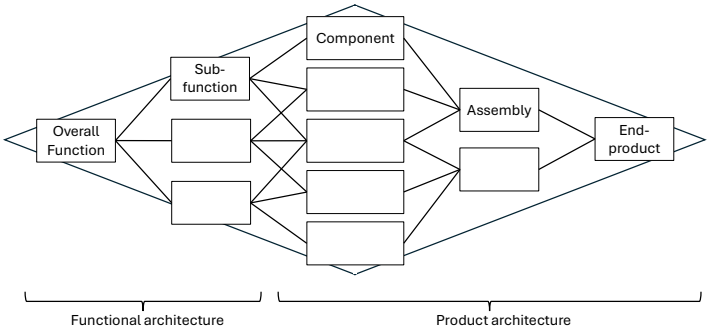

Figure 1. METUS rhombus developed by Daimler AG, ID-Consult, pwc.

The product structuring is performed in three steps: '1) decomposition of the system into elements, 2) documentation of the interactions between the elements, and 3) clustering the elements into architectural and team chunks.' (Pimmler & Eppinger, 1994). Decomposition is a process of breaking down an element into smaller subelements and in described in general product design in Rosenthal et al. (2024). Decomposition for use in DSMs is described in Browning (2001). For clustering, there are numerous contributions in the literature on algorithms for clustering elements to derive modules and subassemblies. A genetic algorithm for clustering is presented in Yu et al. (2003) and a stochastic hill-climbing approach by Borjesson & Hölttä-Otto (2012). More algorithms include satisfying multiple design constraints (Sinha et al., 2019) or multidimensional scaling (Qiao, 2017), clustering for decomposition of rectangular dependency matrices (Li, 2011) and based on axiomatic design and design structure matrices for modularization (Cheng et al., 2012) or to measure the strength of overlapping processes in Concurrent Engineering (Yang et al., 2014).

2.3 Matrix-based Dependency Structure Modelling and Mapping approaches (DSM, DMM, MDM, XDSM, …)

The first idea to model interconnections of complex systems and system components in Engineering Design through binary matrices dates back to the 1970s (Warfield, 1973). Since then, square matrices for describing dependencies have been referred to as Design Structure Matrices (DSMs), also referred to as Dependency Structure Modelling (Steward, 1981; Eppinger and Browning, 2012; Malmqvist, 2002). In addition to (a) the Dependency Structure Matrix (DSM), there are also the methods of (b) Molecular Diagrams (MD), and (c) Visibility-Dependency (VD) signature diagrams for describing product architectures (Sharman & Yassine, 2004). A DSM is a rectangular matrix that captures the dependency relationships between design functions and parameters. DSMs are used for modelling complexity, for example in concurrent engineering projects with complexity through iteration and overlapping tasks (Eppinger, 1991; Yassine & Braha, 2003).

DSMs are used in variant management for numerous purposes, including: designing modular products (Yu et al., 2003), modelling interdependencies in matrix-based approaches from configurable product-level all the way to system-level configuration like supply chains, construction, infrastructure and factory design (Kristianto et al., 2015), designing matrix based product variety with DSMs and case studies of using a quantified DSM to help designers creating product family variant design solutions in concurrent engineering (Luh et al., 2009). DSMs are also used for process modelling (Browning et al., 2006). The matrix notation describes possible system element combinations including features, BOM, and variables representing constraints (Helo, 2006). All directed and undirected as well as weighted and unweighted graphs can be written as a DSM (in Graph Theory: adjacency matrices). A review of numerous extensions and adaptations from different domain applications of DSMs can be found in Browning (2016). An overview of the types of DSMs and MDMs can be found in Malmqvist (2002). Figure 2 shows an example DSM and the corresponding directed and unweighted graph.

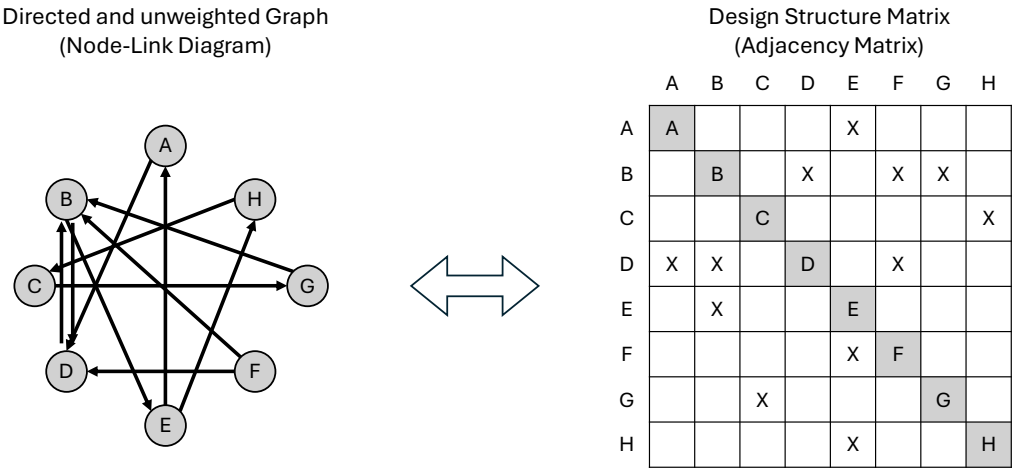

Figure 2. Dependency Structure Modelling of a directed, unweighted graph in a DSM and vice versa (Browning, 2012).

Domain mapping matrices (DMMs) are used to model complex system design across domains and complement DSMs (Danilovic & Browning, 2004). When comparing DSM and DMM approaches, DMM analysis offers several benefits, including (1) capturing the dynamics of product development, (2) showing traceability of constraints across domains, (3) providing transparency between domains, (4) synchronizing decisions across domains, (5) cross-verifying domain models, (6) integrating a domain with the rest of a project or program, and (7) improving decision making among engineers and managers by providing a basis for communication and learning across domains (Danilovic & Browning, 2007).

Modelling more than one domain is part of Multiple-Domain Matrices (MDMs) to link between components, functions, organizations or features (Maurer & Lindemann, 2008; Maurer & Lindemann, 2007). Combining different MDMs of different product design domains into one integrated coherent matrix representation is part of Domain Mapping Matrix

(DMM) (Eichinger et al., 2006; Kreimeyer et al., 2009). Further research proposes sophisticated approaches like the High-Definition Design Structure Matrix (HDDSM) (Tilstra et al., 2012) or a new diagram for visualizing process flows in Extended Design Structure Matrix (XDSM) (Lambe & Martins, 2012). Figure 3 illustrates the modeling of three different domains in a joint MDM schematically and the terminology used for a better understanding. Using multiple DSMs for product variants could benefit from the development of a "logic DSM" that represents relationships among components as well as features or functions while elucidating the importance to good architecture of design rules (Baldwin and Clark, 2000).

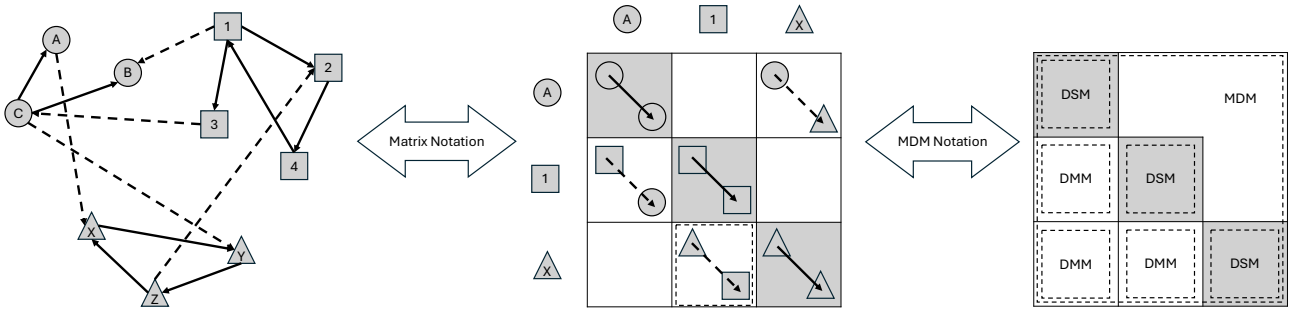

Figure 3. Modelling Multi Domain dependencies in a MDM (Maurer & Lindemann, 2007).

Modelling product variants and ensuring a valid configuration using a DSM has already been approached by Kristianto et al. (2015), Baldwin & Clark (2000), Eppinger & Browning (2012) and Helo (2006). The modelling of variant-rich products in a matrix enables the following advantages: 1. the influence of additional values is made transparent at every level, 2. a product structure designed for variants can be determined, 3. optimization of the product structure can be achieved taking commercial information into consideration (Eigner & Zagel, 2007). Germani et al. (2006) proposed a multi-level DSM instead of a DMM to model configuration and to satisfy market and product requirements on the first level and functional specifications on a second level translated into features. For modelling variant-rich products using DSMs, there is the K&V matrix to represent and map customer and technical views in an MDM developed at ETH Zurich (Bongulielmi et al., 2001). K&V matrix represents an understandable and comprehensible MDM approach for mapping configuration knowledge regarding variant products and usable, for example, in product configurators (Bongulielmi et al., 2002). 'K' - configuration (Ger.: "*Konfiguration*") matrix represents a functional view with customer relevant properties vs technical modules, 'V' - compatibility (Ger.: "*Verträglichkeit*") matrix represents compatibility of combination of customer relevant properties and the combination of modules in a technical view (Bongulielmi et al., 2002). Deubzer et al. (2008) also describe the interdependencies of variant management in a different DSM of a common DMM. In the modelling and structural analysis, a component view, a functional view and a commonality in system design view are modelled (Deubzer et al., 2008). The advantage of describing variant-rich products in matrix notation is that they can be analyzed and optimized using the numerous possibilities of graph theory (Braun & Deubzer, 2007). By using an MDM, links between domains are possible. Advantages of MDM include 1. matrix notation enables intuitive representation, 2. transparency, 3. efficient representation of configuration constraints, 4. cluster analysis offers optimization potential, 5. impact analysis for modification/deletion of features and components, 6. identification of modules and carry over parts (Braun & Deubzer, 2007). Eppinger and Browning introduce 3D DSM (2012, pp. 76) and Delta MDM (2012, pp. 250) to model product variants in matrix notations.

Clustering of the DSM (provided by DSM software vendors like Lattix, Loomeo or the Cambridge Advanced Modeller) visualizes clusters of similar elements like functions, features, and components in the BOM and should therefore be considered to be handled together or assembled together as a sub-configuration. A comprehensive introduction into clustering algorithms for DSMs is provided by Borjesson & Hölttä-Otto (2012). Clustering in DSMs is not to be confused with clustering of variant-rich configurations demonstrated in Mehlstäubl et al. (2023) and Schmidt et al. (2025).

3 Methodology

To optimize the product architecture, a combined approach of the METUS representation of the product architecture with a matrix-based representation of the dependencies in translating the customer language into the component world is selected, thus linking two existing approaches. Firstly, the overall function of a product is conceptually decomposed into sub-functions and elementary functions. Subsequently, the sub-functions are checked for compatibility using a function-DSM and translated into features of the multi-variant product description (function-feature-DMM). In the next step, the satisfiability check is performed in the feature-DSM, using the existing constraints to fill the feature-DSM and allowing a definite interpretation of the Boolean constraints. With a complete and consistent configuration, parts and components are triggered, which are controlled via part validities (TeGÜs and TARs) and are required to fulfil the respective functions (feature-component-DMM). Finally, the components are checked for compatibility in the component-DSM (buildability

check/collision check/interfaces etc.). Lastly, the required components are combined into assemblies, units and modules to form an overall product and constitute a single end product. The various combination possibilities throughout the process allow all end product variants to be modelled. The three DSMs and three DMMs are combined into one overall MDM. Figure 4 summarizes the approach systematically.

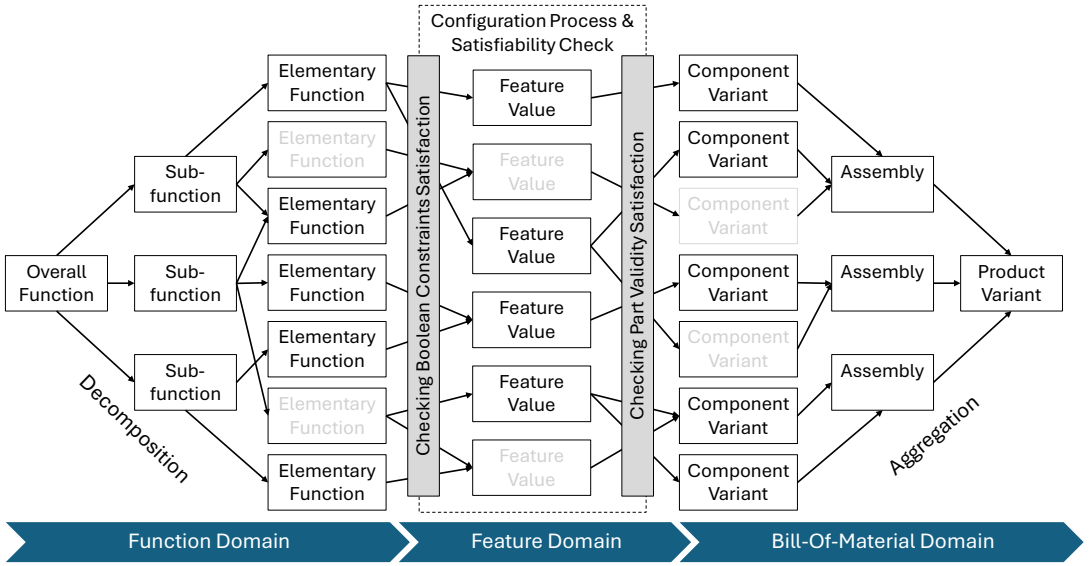

Figure 4. Overview of the proposed method modelling variant-rich products across domains.

In both the feature-DSM and the component-DSM, the solver is used to consider the underlying constraints. In the feature DSM, a distinction is made between the three possibilities when comparing features in pairs: must be combined (Z), can not be combined ( ) and can be combined (X). Derived from the logical formulation and the use of a propositional solver described in Chapter 2.1, the following checks can be used in the pairwise comparison of feature values A and B exemplary (Equation 5-7).

Can (X): $SAT(A,B) = true;$ (5)

Can not ( ): $SAT(A,B) = false;$ (6)

Must (Z): $SAT(A,B)\ AND\ !SAT(A,!B)\ AND\ !SAT(!A,B) = true;$ (7)

The proposed methodology enables the configuration process to be represented consistently across all levels of product, organization and product architecture using neighboring domains and to translate customer requirements into a fully configurable product using the methodical approach of the DSM while taking into account the complex dependencies of the variant-rich product description. This requires little manual effort due to the use of a solver and the constraints involved in the process.

The breakdown into a feature-DSM, a component-DSM and a product-variant-DSM and a matrix-based notation is also addressed in Kesper (2012). The use of a solver to model logical dependencies has not been addressed in the literature examined and is therefore a novel contribution.

4 Results

The result of this work consists of two parts. On the one hand, the method presented in the previous chapter for formalizing constraint-based configuration of variant-rich products in a multi-domain matrix-based modelling approach and on the other hand, an excerpt of a generic feature-based and variant-rich product from the automotive industry that serves as a use case. Figure 5 shows the method and the configuration path of a customer conceptually.

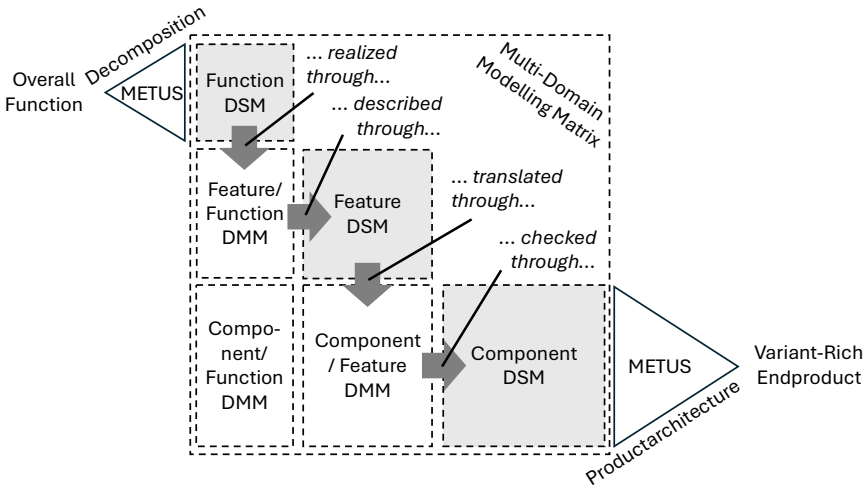

Figure 5. Conceptual approach of combining METUS and MDM

The generic product consists of the four main functions ‘Store Energy’, ‘Convert Energy’, ‘Change Torgue and RPM’ and ‘Heating’, which are decomposed into sub-functions and elementary-functions according to METUS and whose dependencies in terms of possible combinations are shown in the function-DSM (see Figure 6). The sub-functions are then mapped to 12 feature values out of 4 features, which are used to explicitly manage the variance. In the feature DSM, the configuration process proceeds by pairwise testing of the feature values according to equations (5)-(7), where "Z" denotes a command and "X" denotes a constraint. The feature-DSM is filled, and the part validities are interpreted in the feature-, and component-DMM using an interface to a propositional SAT solver. Finally, an assignment to the required components takes place through the interpretation of the part validities, which is controlled via the selection of the respective feature values and is then summarized into assemblies and modules.

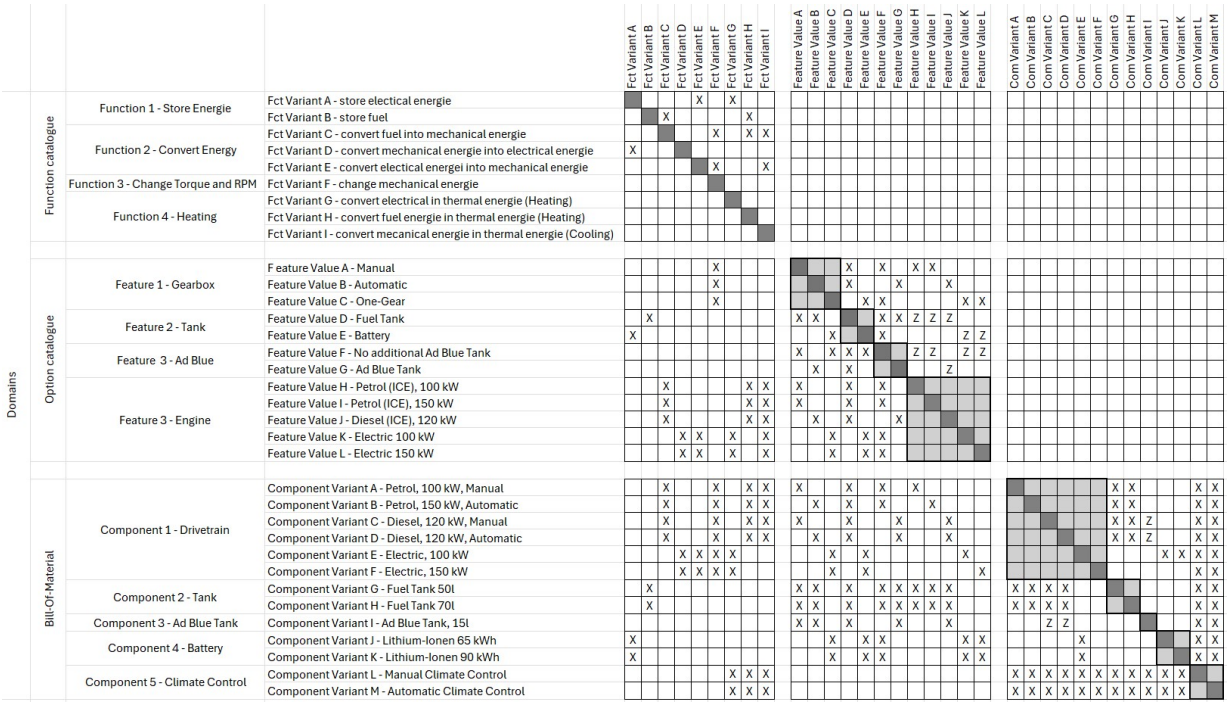

Figure 6. Exemplary excerpt of a generic product described in the proposed MDM.

By combining METUS and MDM, the constraint-based configuration problem is translated into a matrix-based approach and the consistent documentation of variant-rich products from the customer view to the technical view of the component variant (Ger.: "*Komponentenvariante*" - KV) is documented while accounting for the underlying set of constraints. This ensures the validation of the functions (Ger.: "*Funktionsrealisierungsfreigabe*" - FuRe), the consistency of the configuration (satisfying the Boolean constraints) and the feasibility of the BOM (TeGü and TAR) along with a high level of data quality. This promotes a standardized language for technical development as well as for customers and builds on established and proven feature-based variant descriptions.

5 Discussion

One challenge of the product architecture of high-variant products is to create constraints for the individual variants and their specific components and to adapt these to new framework conditions. In the configuration case, these constraints, described by features and feature values in Boolean algebra, trigger the correct components for the specific variant.

Due to new variants and changes in the product portfolio, the constraints for the components must be updated manually and constantly monitored. Due to the notation in Boolean algebra, a constraint can contain negated feature values, which means that no error occurs in the monitoring for a new feature value within a feature, as the constraint adds a component despite the new value. This can lead to errors in production, as the incorrectly added component only becomes apparent there. An exclusively positive notation of constraints makes it easier to monitor the components with the constraints when changes are made to the portfolio.

Using the MDM, these constraints can be derived directly in the feature component matrix in positive notation. Furthermore, the structure of the constraints is always the same and the features and feature values used to describe the constraints are always sorted in a generic order, which makes it much easier to understand a constraint. This positive notation and the always identical structure of the constraints would make the data more transparent and uniform and can improve overall data quality.

For this reason, the proposed approach of this paper supports the continuous, cross-departmental, cross-domain and interdisciplinary documentation in modern product development of variant-rich products through traceability of assignments and transparent development decisions. A matrix-based documentation of functional relationships, constraints and part validities enables strategic product planning in the early phase of the development process by controlling the variance. The configuration path shown along the DSMs and DMMs comprehensively represents the configuration problem of high-variant products along the function decomposition, through the feature check to the BOM and the product architecture and thus provides the possibility of optimization and sensitivity analysis in a standardized language.

Additionally, this type of dependency modelling enables not only machine-readable, but also human-readable and understandable interpretation for designers and can be used and, if necessary, extended for the traceability of combinatorics in portfolio maintenance. Clustering algorithms of the various DSMs and DMMs make it possible to identify similar functions, combine feature values into features or jointly model the combination of components into assemblies and modules and derive packaging proposals from this, for example. These resulting packaging proposals are new constraints which can be derived directly from the matrix.

6 Conclusion

In conclusion, this article addresses the modelling of dependencies in constraint-based configuration by mapping function, feature and component relationships in a combined MDM. In this process, a function to be fulfilled is first decomposed according to the METUS method, translated into components using the matrix of features and then structured in the form of a product architecture. In this way, a standardized language between customer and designer is improved using the feature description and, simultaneously, the various constraints of the configuration are consistently considered.

The research question on mapping constraint-based configuration of variant-rich products can thus be answered by modelling an MDM and has been demonstrated partially for a generic product. With the support of a solver, the work of filling in the DSMs and thus the consideration of the constraints can be automated, supporting the designer in his value-adding work. The clustering possibilities in the respective DSMs and the subsequent structuring options of METUS have identified potential for product architecture optimization. While the feature-, and component-DSMs are filled by interpreting the constraints and part validities, filling the function-DSM and DMMs involves considerable manual effort. In particular, since the preceding function decomposition has a significant influence on the result but is inconclusive today. In addition, the application on a complete product is still pending along with an evaluation of the assistance for designers and product engineers. Additionally, chronological changes as well as further neighboring domains like physically contacting parts for geometry checking and connection techniques will be considered in future research in order to not only understand but also be able to successfully model and eventually influence the complex interrelationships in variant-rich products.

Modelling variant-rich products in a matrix notation enables designers to not only validate satisfiability against constraints on feature-level and part validities on BOM-level but also to use sensitivity analysis to simulate changes and derive new constraints directly as logical term from the matrix in machine- and human-readable formats.

References

Al Handawi, K., Brahma, A., Wynn, D.C., Kokkolaras, M., Isaksson, O. 2024. Design Space Exploration and Evaluation Using Margin-Based Trade-Offs. ASME. Journal of Mechanical Design, 146(6):061701. https://doi.org/10.1115/1.4063966

Cheng, Q., Zhang, G., Gu, P., Shao, X.A., 2012. Product Module Identification Approach Based on Axiomatic Design and Design Structure Matrix. Concurrent Engineering. 20(3), pp. 185-194. https://doi.org/10.1177/1063293X12453350

Baldwin, C., Clark, K., 2000. Design Rules - The Power of Modularity. Vol. 1. Cambridge, MA, USA: MIT Press Cambridge. https://doi.org/10.7551/mitpress/2366.001.0001

Bongulielmi, L., Henseler, P., Puls, C., Meier, M., 2001. The K- & V-Matrix Method – An Approach in Analysis and Description of Variant Products. In: Proceedings of the 13th International Conference on Engineering Design (ICED 01), Glasgow. Bury St. Edmunds: IMechE, 2001, pp. 571-578.

Bongulielmi, L., Henseler, P., Puls, C., Meier, M., 2002. The K-&V-Matrix-Method in Comparison with Matrix-Based Methods supporting Modular Product Family Architectures. NordDesign 2002 - Visions and Values in Engineering Design, 14-16 August, Norwegian University of Science and Technology, NTNU Trondheim, Norway.

Borjesson, F., Hölttä-Otto, K., 2012. Improved Clustering Algorithm for Design Structure Matrix. Proceedings of the ASME 2012 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 3: 38th Design Automation Conference. Chicago, Illinois, USA. pp. 921-930. https://doi.org/10.1115/DETC2012-70076

Braun, T., Strattner, M., 2017. Variant Management Toolbox. 21st International Conference on Engineering Design, ICED17, In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 3: Product, Services and Systems Design, Vancouver, Canada. pp. 409-418, ISBN: 978-1-904670-91-9

Braun, T., Deubzer, F., 2007. New Variant Management using Multiple-Domain Mapping. 9th International Design Structure Matrix Conference, DSM'07, pp. 363-365.

Browning, T.R., 2001. Applying the Design Structure Matrix to System Decomposition and Integration Problems: A Review and New Directions. IEEE Transactions on Engineering Management, Vol.48, No.3, pp. 292-306. https://doi.org/10.1109/17.946528

Browning, T.R., 2016. Design Structure Matrix Extensions and Innovations: A Survey and New Opportunities. In: IEEE Transactions on Engineering Management, Vol. 63, No. 1, pp. 27-52. https://doi.org/10.1109/TEM.2015.2491283

Browning, T.R., Fricke, E., Negele, H., 2006. Key Concepts in Modeling Product Development Processes. Systems Engineering. The Journal of The International Council on Systems Engineering. Vol. 9, No. 2, pp. 104-128. https://doi.org/10.1002/sys.20047

Danilovic, M., Browning, T.R., 2004. A Formal Approach for Domain Mapping Matrices (DMM) to Complement Design Structure Matrices (DSM). Proceedings of the 6th Cambridge DSM Workshop, Cambridge, UK, 2004.

Danilovic, M., Browning, T.R., 2007. Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management, Vol. 25, No. 3, pp. 300-314. https://doi.org/10.1016/j.ijproman.2006.11.003

Deubzer, F., Braun, T., Maurer, M., Lindemann, U., 2008. Applying the multiple Domain Mapping Approach to Variant Management. Proceedings DESIGN 2008, The 10th International Design Conference, Dubrovnik, Croatia, International Design Conference - Design 2008, pp. 335-342.

Eichinger, M., Maurer, M., Lindemann, U., 2006. Using Multiple Design Structure Matrices. Proceedings DESIGN 2006, The 9th International Design Conference, Dubrovnik, Croatia, International Design Conference - Design 2006, pp. 229-236.

Eigner, M., Zagel, M., 2007. Product Structures Designed for Variants. 9th International Design Structure Matrix Conference, DSM '07, 2007. DSM 2007: Proceedings of the 9th International DSM Conference, Munich, Germany, pp. 249-251.

Eppinger, S. D., Browning, T. R., 2012. Design Structure Matrix Methods and Applications. Cambridge, MA, USA: MIT Press, 2012. https://doi.org/10.7551/mitpress/8896.001.0001

Eppinger, S.D., 1991. Model-based Approaches to Managing Concurrent Engineering. Journal of Engineering Design, 2(4), pp. 283–290. https://doi.org/10.1080/09544829108901686

Feldhusen, J., Grote, K.-H., Pahl, G., Beitz, W. (Hg.), 2013. Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung. 8., vollst. überarb. Aufl. Berlin: Springer Vieweg.

Germani, M., Mengoni, M., Raffaeli, R., 2006. Design Structure Matrix used as Knowledge Capture Method for Product Configuration. Proceedings DESIGN 2006, The 9th International Design Conference, Dubrovnik, Croatia, pp. 253-262.

Germani, M., Mandorli, F., 2004. Self-configuring components approach to product variant development. In: Artificial Intelligence for Engineering Design (AIEDAM), special issue on Product Platform Development for Mass Customisation, Analysis and Manufacturing. Vol. 18, No. 1, Cambridge University Press, pp. 41-54. https://doi.org/10.1017/S0890060404040041

Helo, P.T., 2006. Product configuration analysis with design structure matrix. Ind. Manage. Data Syst., Vol. 106, pp. 997–1011. https://doi.org/10.1108/02635570610688896

Kesper, H., 2012. Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden. Doctoral Thesis. Munich.

Kreimeyer, M., Braun, S., Gürtler, M., Lindemann, U., 2008. Relating Two Domains via a Third: An Approach to Overcome Ambiguous Attributions Using Multiple Domain Matrices. Proceedings of the ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 1: 34th Design Automation Conference. Brooklyn, New York, USA. pp. 297-306. https://doi.org/10.1115/DETC2008-49249

Kristianto, Y., Helo, P.T., Jiao, R.J., 2015. A system level product configurator for engineer-to-order supply chains. Computers in Industry, Vol. 72, pp. 82-91. https://doi.org/10.1016/j.compind.2015.04.004

Lambe, A.B., Martins, J.R.R.A., 2012. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. Struct Multidisc Optim, Vol. 46, pp. 273–284. https://doi.org/10.1007/s00158-012-0763-y

Lattix Products. 2025. [Online] URL: https://www.lattix.com/products/, last accessed on July 21, 2025

Lee, J., Lim, J., Hong, Y.S., Kang, C., 2025. Variant mode and effects analysis for effective product family expansion under modular architecture. Journal of Engineering Design, pp. 1–27. https://doi.org/10.1080/09544828.2025.2473982

Li, S., 2011. A matrix-based clustering approach for the decomposition of design problems. Res Eng Design 22, Vol. 22, pp. 263–278. https://doi.org/10.1007/s00163-011-0111-z

Loomeo – Smart Business Analytics. 2025. [Online] URL: https://www.loomeo.com/, last accessed on July 21, 2025

Luh, D.-B., Ko, Y.-T., Ma, C.-H., 2009. A structural matrix-based modelling for designing product variety. Journal of Engineering Design, 22(1), pp. 1–29. https://doi.org/10.1080/09544820902877591

Malmqvist, J., 2002. A Classification of Matrix-based Methods for Product Modeling. In: DS 30: Proceedings of DESIGN 2002, The 7th International Design Conference, Dubrovnik, pp. 203-210.

Maurer, M., Lindemann, U., 2008. The application of the Multiple-Domain Matrix: Considering multiple domains and dependency types in complex product design. IEEE International Conference on Systems, Man and Cybernetics, Singapore, pp. 2487-2493. https://doi.org/10.1109/ICSMC.2008.4811669

Maurer, M., Lindemann, U., 2007. Structural Awareness in Complex Product Design - The Multiple-Domain Matrix. 9th International Design Structure Matrix Conference, DSM'07, pp. 87-97.

Mehlstäubl, J., Pfeiffer, C., Kraul, R., Braun, F., Paetzold-Byhain, K., 2023. Methodical Approach to Cluster Configurations of Product Variants of Complex Product Portfolios. In: Proceedings of the International Conference on Engineering Design (ICED23), Bordeaux, France. Vol. 3, pp. 2645-2654. https://doi.org/10.1017/pds.2023.265

METUS Software. 2014. [Online] URL: http://www.id-consult.com/, last accessed on July 21, 2025

METUS Software. 2025. [Online] URL: https://store.pwc.de/en/products/metus/, last accessed on July 21, 2025

Müller, J.R., Isaksson, O., Landahl, J., Raja, V., Panarotto, M., Levandowski, C., Raudberget, D. 2019. Enhanced function-means modeling supporting design space exploration. Artificial Intelligence for Engineering Design (AIEDAM), Analysis and Manufacturing. 33(4), pp. 502-516. https://doi.org/10.1017/S0890060419000271

Pimmler, T.U., Eppinger, S.D., 1994. Integration Analysis of Product Decompositions. Proceedings of the ASME 1994 Design Technical Conferences collocated with the ASME 1994 International Computers in Engineering Conference and Exhibition and the ASME 1994 8th Annual Database Symposium. 6th International Conference on Design Theory and Methodology. Minneapolis, Minnesota, USA. pp. 343-351. https://doi.org/10.1115/DETC1994-0034

Qiao, L., Efatmaneshnik, M., Ryan, M., Shoval, S., 2017. Product modular analysis with design structure matrix using a hybrid approach based on MDS and clustering. Journal of Engineering Design, 28(6), pp. 433–456. https://doi.org/10.1080/09544828.2017.1325858

Rosenthal, P., Demke, N., Mantwill, F., Niggemann, O., 2024. Plan-Based Derivation of General Functional Structures in Product Design. 2024 IEEE 7th International Conference on Industrial Cyber-Physical Systems (ICPS), St. Louis, MO, USA, 2024, pp. 1-8. https://doi.org/10.1109/ICPS59941.2024.10640039

Schmidt, T., Köhler, M., Winger, F., Roth, D., Kreimeyer, M., Mantwill, F., 2025. High-Dimensional Clustering of Configurations Towards a Customer Segment Analysis in Automotive Industry. Stuttgarter Sympossium für Produktentwicklung 2025.

Sharman, D.M., Yassine, A.A., 2004. Characterizing Complex Product Architectures. Systems Engineering. The Journal of The International Council on Systems Engineering. Vol. 7, No. 1, pp. 35-60. https://doi.org/10.1002/sys.10056

Sinha, K., Han, S.-Y., Suh, E.S., 2019. Design structure matrix-based modularization approach for complex systems with multiple design constraints. Systems Engineering. The Journal of The International Council on Systems Engineering. Vol. 23, No. 2, pp. 211–220. https://doi.org/10.1002/sys.21518

Soltan, A., Addouche, S.-A., Zolghadri, M., Barkallah, M., Haddar, M., 2019. System Engineering for dependency analysis - a Bayesian approach: application to obsolescence study. 29th CIRP Design 2019 (CIRP Design 2019), Procedia CIRP 84, pp. 774-782. https://doi.org/10.1016/j.procir.2019.04.253

Steward, D.V., 1981. The Design Structure System: A Method for Managing the Design of Complex Systems. In: IEEE Transactions on Engineering Management, Vol. EM-28, No. 3, pp. 71-74. https://doi.org/10.1109/TEM.1981.6448589

Tilstra, A.H., Seepersad, C.C., Wood, K.L., 2012. A High-Definition Design Structure Matrix (HDDSM) for the quantitative assessment of product architecture. Journal of Engineering Design, 23(10–11), pp. 767–789. https://doi.org/10.1080/09544828.2012.706748

Ulrich, K.T., Eppinger, S.D., 2004. Product Design and Development. (USA: McGraw-Hill Companies Inc) 2004, The McGraw-Hill Companies Inc., NewYork.

Von Eisenhart Rothe, M., 2002. Konzeption und Einführung eines IT-gestützten Produkt-Konfigurationsmanagements für die technische Information in der Automobilentwicklung und -herstellung. Aachen: Shaker, Innovationen der Fabrikplanung und -organisation; Bd. 7; Hrsg. Univ.-Prof. Dr.-Ing. Uwe Bracht, Clausthal, Techn. Univ., Diss., 2002; ISBN: 3-8322-0419-9.

Warfield, J.N., 1973. Binary Matrices in System Modeling. In: IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-3, No. 5, pp. 441-449. https://doi.org/10.1109/TSMC.1973.4309270

Whitney, D.E., Dong, Q., Judson, J., Mascoli, G., 1999. Introducing Knowledge-Based Engineering Into an Interconnected Product Development Process. Proceedings of the ASME 1999 Design Engineering Technical Conferences. Vol. 3: 11th International Conference on Design Theory and Methodology. Las Vegas, Nevada, USA. pp. 351-360. https://doi.org/10.1115/DETC99/DTM-8741

Yang, Q., Yao, T., Lu, T., Zhang, B., 2014. An Overlapping-Based Design Structure Matrix for Measuring Interaction Strength and Clustering Analysis in Product Development Project. In: IEEE Transactions on Engineering Management, Vol. 61, No. 1, pp. 159-170. https://doi.org/10.1109/TEM.2013.2267779

Yassine, A.A., Braha, D., 2003. Complex Concurrent Engineering and the Design Structure Matrix Method. Concurrent Engineering. 11(3), pp. 165-176. https://doi.org/10.1177/106329303034503

Yu, T.-L., Yassine, A.A., Goldberg, D.E., 2003. A Genetic Algorithm for Developing Modular Product Architectures. Proceedings of the ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 3b: 15th International Conference on Design Theory and Methodology. Chicago, Illinois, USA. pp. 515-524. https://doi.org/10.1115/DETC2003/DTM-48647

Contact: T. Schmidt, Helmut Schmidt University, Department of for Machine Elements and Computer Aided Product Design, Holstenhofweg 85, 22043, Hamburg, Germany, +49 40 6541-3794, thorsten.schmidt@hsu-hh.de, www.hsu-hh.de/mrp