INTERNATIONAL DESIGN CONFERENCE - DESIGN 2014
Dubrovnik - Croatia, May 19 - 22, 2014.

GENERATING A META-MODEL FOR MODULARIZATION METHODS

J. Hackl, T. Gumpinger and D. Krause

Keywords: product family, modularization, multiple-domain matrix, product structure

1. Introduction

Product architecture has to reflect the needs of the lifecycle phases of a product. Megatrends like globalization and individualization change these needs and new demands on product architecture arise. These have to be addressed to be competitive in a changing market. Producing individualized product variants is a major challenge in satisfying market demands. Modularizing product structures is a typical way to achieve this; several product development methods have been developed and applied in industry. They can be classified by their modularization goals, whether technical-functional or product-strategic. These methods need different information for their application. This paper analyses commonly applied methods for their data acquisition needs and information processed by their domains and relations used. This information is used to synthesis a generic information framework for modularization. Seven methods are analysed using the approach presented in the next section. The steps of each approach are described and domains and relations used are identified. The information acquired is subsequently processed into an overall information framework and evaluated. Therefore, the previously used author-specific domains and relations are generalized at an abstract level. The methods' information schemes are united into a common meta-model.

2. Analysis Model

An abstract form is chosen to compare different modularization methods. In this paper the processed information is of interest, hence the generation and flow of information needs to be represented. Multiple-domain modelling [Maurer 2007] is a generic way to describe the information of complex systems. The general idea of multiple-domain matrices (MDM), represented by domains and relations, is adapted to represent the information needs and flow of the methods. Domain means a kind of entity representing a specific view [Kreimeyer and Lindemann 2011]. Relations are the interactions or links between the domains or within a domain. By collecting the domains and their relations it is possible to get a more abstract view of the methods. This is necessary to find similarities and differences. Kreimeyer and Lindemann [2011] use a similar approach to compare several methodologies in the field of structural process modelling.
The methods are examined in three steps. First, a short description of the method is given, within which the goals and the typical initial situation of the method are described. Thus it is possible to find out if methods with the same goal use a similar approach by connecting the same domains with similar relations. The initial situation might play a role in how a method is set up.
The methods are classified into the underlying scheme of Suh [2005]. Suh created a concept of domains which was used to build an axiomatic design framework: a method for a structured design of systems. According to Suh the design world consists of the four domains customer, functional,

physical and process. He states that many systems of product design can be described using them. Simpson et al. [2006] adds logistics to Suh's domains, summarising the domains into three parts: front-end issues, optimization-based methods and back-end issues. Figure 1 shows the simplified scheme of Simpson, which will be used to group the methods, facilitating the comparison of methods. After classifying the methods in this scheme, individual method steps can be allocated to the area in which they provide solutions and can be specifically chosen.

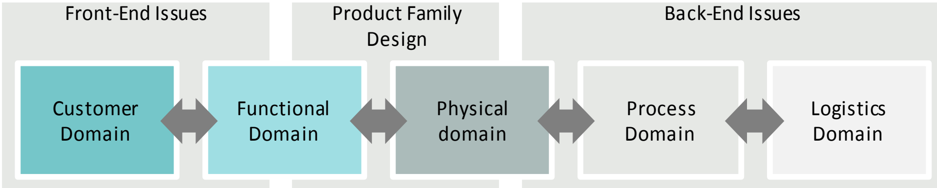

Figure 1. Generic domains from [Simpson et al. 2006] and [Suh 2005]

The following section contains a summary of every method. The information used is transferred into the domain space. For example, components relate to the physical domain and functions to the functional domain. Any attributes (written in *italics*), like costs of components, which specify domain elements are recorded in the step description below each MDM. The relations between domains are located in the fields of the matrices. Characters within the fields enumerate the sequence of the steps. Specific relation attributes are also in the step description. The second classification is in the field of knowledge management and concerns the terms data, information, knowledge, wisdom [Rowley 2007]. The key parts of the methods are found by analysing the steps in which data are acquired, information and knowledge is created and new insights are gained. This results in a content quality of the methods steps. Each method closes with a short summary.

3. Analysis of methods for a modular product architecture

In this section, seven methods for developing modular product structures are analysed, as described in the previous section. This selection of methods provides initial insights into the information needed for modularization. Krause gives an overview of further methods to be evaluated using this approach [Krause and Ripperda 2013].

Hölttä-Otto – Modular product platform design

Hölttä-Otto [2005] extends existing modularization methods for optimizing product families by considering variants. The approach encompasses three metrics. The first is flexible interface design, which lets the user identify critical interfaces in a product architecture. The main approach lies in evaluating the main flows between functions and their effort when being changed. The second metric is the identification of common modules across the product family, which is done by normalizing all inputs and outputs of product functions and calculating the Euclidian distances between functions. The same is possible for the physical domain, using component requirements and attributes like weight, for example. The Euclidian distances are then drawn in a dendrogram to derive an adequate product structure. The various platform concepts are evaluated with the third metric. Figure 2 shows the MDM that can be developed from Hölttä-Ottos metrics.

	Function	Component	Module (Platform concept)	Strategy
Function	A. Coupled with *Flow* B. *Change effort* D. *Euclidian Distance*		C. Summed up to	
Component		D. *Euclidian Distance*		
Module Platform concept				E. Scored
Strategy				

Figure 2. Abstracted MDM as per the method of Hölttä-Otto [2005]

Table 1. Method steps of Hölttä-Otto [2005]

Step		*Attribute*	Quality of content
A	Building of functional structure	*Flow*	- Information
B	Evaluation of change effort of flow types	*change effort*	- Product knowledge needed
C	Evaluation of modules regarding change effort sum		- Information about quality of the existing modularization
D	Deriving of interfunctional/ inter-component Euclidian distance; alignment in dendrogram	*Euclidian distance*	- Information about coupling between modules and information for module generation
E	Scoring of platform concepts		- Knowledge of company strategy and evaluation criteria needed to gain information on concept quality

Hölttä-Ottos metrics concentrate on the two domains of function and component. They are specifically designed for the field of product family design, relating to Simpson. Her metrics are of a technical-functional nature. Hölttä-Otto communicates the product family's degree of commonality by visualizing the optimal module fragmentation with a dendrogram.

Göpfert – Modular product families

Göpfert's [1998] Metus method aims to harmonize product structure and organisational structure by developing a modular product architecture. This contributes to a reduction in organization efforts in the product development process. The initial situation for Metus deployment is typically an existing product family, where the external variety needed has often led to an increase in internal variety. With an academic background, this method is transferred into a consulting service. Metus organizes components into modules that consider the functional aspect and the organizational aspect. The MDM below (Figure 3) represents this in an abstracted context.

	Function	Component	Module	Person
Function	A. Connected with	B. Fulfilled by		
Component			C. Summed up to	
Module				
Person	D. develops			

Figure 3. Abstracted MDM as per the method of Göpfert [1998]

Table 2. Method steps of Göpfert [1998]

Step		*Attribute*	Quality of content
A	Building of functional structure		- Data collection
B	Mapping of functions to components		- Information about variant components
C	Building of modules		- Data collection
D	Organizational structure		- Information about organizational effort

Göpfert reorganizes components into modules, benefitting the domains that influence this aggregation. It contributes to the functional binding and organizational structure. He enhances his method with the metus rhombus visualization. This visual representation relates two domains and groups them hierarchically.

Jiao – A methodology of developing product family architecture for mass customization

Jiao et al. [1999] generally aim to increase product variety in mass customization while keeping development and production efforts as lean as possible. Keeping manufacturing costs low is another aim mentioned. The initial situation is an existing mass product with increasing external variety, which leads to an increase in internal variety. The method utilizes systematic steps to formulate the product family architecture. Jiao analyses the problem from three different views: functional, physical and technical. He develops a new product structure by going through all of Suh's [2005] domains, from customer requirements to functional requirements, design parameters and components to the final product structure, considering cost and performance.

	Customer	Functional Requirement	Design Paprameter	Component	Module	Variant	Process capabilities - person
Customer	C. *no. planned products*	C. Needs C. Importance					
Functional Requirement		B. Is part of B. *instance*	D. Fulfilled by				
Design Parameter				F. Fulfilled By			
Component				H. *Cost*			
Module				F. Contains	G. *Performance*	I. Included in	
Variant							
Process capabilities - person				F. Develops			

Figure 4. Abstracted MDM as per the method of Jiao et al. [1999]

Table 3. Method steps of Jiao et al. [1999]

Step		*Attribute*	Quality of content
A	Formulation of functional requirements (FR)		- Data acquisition
B	Hierarching FRs down to instances/variables		
C	Customer exploration Most important FR instances across customers Similar FR instances across customers	*No. planned products instance and importance*	- Customer knowledge necessary
D	Mapping of FR to Design Parameters (DP)		- Data acquisition
E	Module generation; clustering FR-DP matrix		- Information about design modules and technical modularity
F	Component determination regarding design modules and process capabilities		
G	Performance of modules	*performance*	- Information about performance gained
H	Cost estimation	*cost*	
I	Final configuration structure after economic evaluation in G. and H.		- Previous information and knowledge needed

Jiao creates a modularization approach which he reconciles for technical performance, costs and process capabilities. He presents a broad method that addresses and assesses all domains except logistics.

Lindemann – Structural Complexity Management

With structural complexity management, Lindemann et al. [2009] offers a systematic approach for managing complex structures in product design. It is not specially made but is applicable to the development of modular product structures. The approach leads from system definition and information acquisition to the deduction of indirect dependencies, structure analysis and to an improved product system. For modularization he finds the key domains components, functions and features, which can be extended to a specific problem.

	Function	Component	Feature
Function	A. Requires		
Component	C. Serves	B. Has spatial dependency to D. Has functional dependency D. Has feature dependency	C. Has influence on
Feature		C. Has influence on	C. Has influence on

Figure 5. Abstracted MDM as per the method of Lindemann et al. [2009]

Table 4. Method steps of Lindemann et al. [2009]

Step		*Attribute*	Quality of content
A	Flow-oriented functional model		- Data acquisition
B	Record geometric dependencies		- Data acquisition
C	Record of direct dependencies		- Data acquisition
D	Deduction of indirect dependencies		- Information generation about indirect dependencies of components
E	Structure analysis by clustering all component related matrices		- Aggregated information for module candidates
F	Generation of product structure depending on relations		

Multiple-domain modelling is a generic approach for structural complexity management. It is closely related to the Design Structure Matrix (DSM) and graph theory. Hence many clustering algorithms are available for modularization. Software tools, like LOOMEO, support the approach and provide a visual representation of the MDMs using node-link diagrams. A powerful possibility is the deduction of indirect dependencies of domains. This gives insights into previously unknown relationships. Multiple case studies use the approach for modularization of products.

Pimmler/ Eppinger – Design Structure Matrix (DSM)

Pimmler and Eppinger [1994] adapt the DSM [Steward 1981] with the aim of improving the product architecture and team structure. It is a technical-functional modularization that clusters the functional or component domain and is carried out to redesign single products.

The method's main idea is the breakdown of a product into components or functions (in the early development phase). These are grouped into modules based on coupling criteria that quantify the strengths between the components. A cluster algorithm swaps rows and columns to regroup the DSM elements.

	Function	Component	Module	Person
Function	B. Interacts with B. *Spatial, enery, information, material*			
Component		B. Interacts with B. *Spatial, enery, information, material*	D. Summed up to	
Module				
Person	E. Assigned to	E. Assigned to		

Figure 6. Abstracted MDM as per the method of Pimmler and Eppinger [1994]

Table 5. Method steps of Pimmler and Eppinger [1994]

Step		*Attribute*	Quality of content
A.	Decomposition of product		- Product knowledge needed
B.	Coupling between components	*spatial...*	- Data acquisition

C. Simplification on main relation		
D. Regrouping of DSM		- Information about sensible modules
E. Derivation of organizational structure		- Information about organizational efforts

The consideration of couplings between domain elements allows a deeper understanding of the technical-functional dependences of the modular product structure. The influence of the clustering algorithm has to be considered. Pimmler and Eppinger use a holistic algorithm that centres the positive quantified elements at the main diagonal of the DSM.

Simpson – Integrated approach to product family design

Simpson et al. [2012] takes several detached methods and integrates them. He develops communal components by translating customer requirements into commonality specifications. The aim is to find a product platform for the analysed product family.

	Customer Requirement	Market Segment	Price/ performance tier	Engineering Requirement	Design Variable	Component	Variant	Time
Customer Requirement				B. Characterizes				
Market Segment								A. Served when
Price/ performance tier								
Engineering Requirement						C. Influences D. *Likelihood of redesign*		
Design Variable								
Component						E. Coupled with		
Variant		A. Sold to	A. Fulfills					
Time								

Figure 7. Abstracted MDM as per the method of Simpson et al. [2012]

Table 6. Method steps of Simpson et al. [2012]

Step	*Attribute*	Quality of content
A Mapping of variants to market segments and price/performance tiers		- Information about assigning products to market segments
B Mapping customer requirements (CR) to engineering requirements (ER)		
C Mapping of ER to components		
D Rating ER – component matrix on	*Likelihood of redesign*	- Knowledge of future changes needed, information about candidates for a communal platform outcome
E Deriving a coupling index		- Information about component coupling
F Derivation of a commonality index		- Information about commonality of the product family
G Multi-Objective Optimization		

In the approach shown he uses existing modularization methods that can be exchanged as needed. The approach especially looks at the variety across variants. The framework overcomes the fragmentation of the individual methods used.

Krause – Integrated PKT-approach

The integrated PKT-approach by Krause and Eilmus [2011] aims to reduce internal variety while maintaining external variety levels for the customer. The initial situation is an existing product family that has to be redesigned due to an increase in variety. Krause looks at both technical-functional modularization and product-strategic modularization. The approach consists of two major parts. First, the product family is redesigned to comply with design for variety [Kipp 2012]. The components of the adapted product family are then clustered according to the entire product lifecycle [Blees 2010].

	Customer	Customer relevant propertie	Variant value	Function	Working principle	Component	Module	Variant	Production	Supplier
Customer							H. Drives			
Customer relevant propertie										
Variant Value		A. Specifies						B. Included in		
Function		D. Fulfills		*C. standard, variant, optional*			H. Drives			
Working principle				D. Fulfills	*C. standard, variant, optional*					
Component					D. Fulfills	*C. standard, variant, optional* F. Has information, energy, material flow				
Module						F. Contains G. Contains				
Variant										
Production							H. Drives			
Supplier							H. Drives			

Figure 8. Abstracted MDM as per the method of Krause and Eilmus [2011]

Table 7. Method steps of Krause and Eilmus [2011]

	Step	*Attribute*	Quality of content
A	Specifying customer relevant properties with variant values		- Data collection
B	Mapping of variant values to variants		- Information about variety in the product family
C	Analysing functions, working principles and components on the attributes	*Standard, variant, optional*	- Data collection
D	Allocation of the variant elements of domains into A - C		
E	Generation of solutions for design for variety		
F	Clustering of components according to flow schemes		- Module definition for technical-functional modularization
G	Developing modularizations for every life phase relevant for the company		- Information about conflicts between the life phases
H	Harmonizing the modularizations to a final concept		- Expert knowledge necessary for finding an optimal solution

This approach for developing modular product families is workshop-oriented and therefore uses many visualization tools to communicate the results to the customer. Commonly used methods are utilized for modularization. The step of harmonizing different modularizations of a product family across its lifecycle differs from other methods.

4. Evaluation of the methods and derivation of a meta-model

To evaluate the methods, their information flows need further abstraction. The author-specific domains were previously assigned to the MDM. To unify them, the domains are sorted into the scheme by Suh and Simpson (Figure 1). The general domains in this figure are color-coded. For reference, these colours are reused in each method's MDM (Section 3) to indicate the domain affiliation, e.g. Lindemann et al [2009] describe component and feature domains that both correspond to the physical domain. The relations of the author-specific domains are reordered by this classification scheme.

This sorting creates the possibility of combining the methods into an overall MDM (Figure 9). The appearance of relations between domains is counted within the fields. All methods use a relation within the physical domain and nearly all use one in the functional domain as well. This can be interpreted, as most of the analysed methods approach the modularization problem from the technical-functional point of view [Krause and Ripperda 2013]. Modularization eventually has to affect the physical domain, grouping components into modules.

In addition, the relations count from the functional domain to the physical domain is distinctive. This reflects that mapping of functions to components or modules is a basic concept of modularization (function binding in [Salvador 2007]). The domains surrounding the functional and physical centre domains are less connected. The customer domain has a big influence on modularization methods. This is represented by the active sum (5), the sum of values in each row, which indicates the overall influence a domain has on other domains [Lindemann et al. 2009]. The passive sum, sum of values in each column, shows how influenced a domain is. Figure 9 (right) illustrates the analysed methods' domains with an influence portfolio, mapping active to passive sum. Again, functional domain and physical domain are highly interactive.

	Customer Domain	Functional Domain	Physical Domain	Process Domain	Logistics Domain	Active sum
Customer Domain	1	1	2	1	0	5
Functional Domain	1	6	4	0	0	11
Physical Domain	1	2	7	1	0	11
Process Domain	0	2	2	0	0	4
Logistics Domain	0	0	1	0	0	1
Passive sum	3	11	16	2	0	

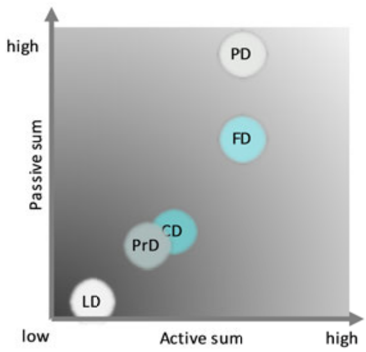

Figure 9. Relationships among domains across all analysed methods; influence portfolio

Relations with the same meaning across the methods' MDMs can be unified. The matrix above is extended with these relationships. This gives an information framework for modularization of product families. Further detailing of the domains and relations with attributes results in a meta-model (Figure 10). Unified relations with a high occurrence are marked in **bold**. Relations are important because they enable aspects of the modularization to be put in context. The context gives new insights and enables matching with the product structure. For example, Pimmler/Eppinger optimized the team structure by relating the process domain (persons) with the physical domain (components). Methods try to create domain-specific benefits by relating and harmonizing structures of or within a domain.

The coupling elements of the functional domain (by assigning relations with flow-based attributes) occur several times within the analysed methods. Especially in early design phases, but without a physical structure, this abstraction is used to cluster functions into modules. When modularization is applied to existing product families the couplings of the physical domains are analysed. Some methods specify the couplings with attributes (e.g. flow types). Most of the analysed methods are only partly product-strategic. Hence the derived meta-model preferably represents technical-functional modularization.

	Customer Domain	Functional Domain	Physical Domain	Process Domain	Logistics Domain
Customer Domain	*planned products*	Needs Importance	Characterizes Drives	Served when	
Functional Domain	Specifies Fulfills	**Coupled with** *Change effort* *Euclidian Distance* Is part of *Instance* ***Spatial, enery, information, materia*** *standard, variant, optional* Requires	**Fulfilled by** Summed up to Included in Drives		
Physical Domain	Fulfills Sold to	Fulfills Serves	*Euclidian Distance* Fulfilled By *Cost* **Contains** *Performance* **Included in** *Spatial, enery, information, material* Influences *Likelihood of redesign* **Coupled with** *standard, variant, optional* Drives	Scored	
Process Domain		Develops Assigned to	Develops Assigned to		
Logistics Domain			Drives		

Figure 10. Meta-model showing relations and attributes (*italics*) used across the methods

Methods that pursue the same aim were analysed by comparing their steps and required information flows. Göpfert and Pimmler/Eppinger aim for the alignment of product structure and team structure. Pimmler and Eppinger analysed couplings of functions or components within a domain and clustered them with an algorithm. Göpfert connects functions and components, and aligned the subsequent functional structure with the component structure to derive modules across domains. He matches the organizational structure across domains to functional and physical elements, where Pimmler/ Eppinger derive the team structure from the modularization.

Jiao and Krause aim to maintain external variety levels while reducing internal variety. Both modularize from the technical-functional and the product-strategic view. Jiao builds a functional structure from functional requirements, leading to instances of the function.

Krause specifies the customer domain (customer relevant properties) by relating the functional domain (variant values). In application, customer requirements and functional requirements often correspond; hence both methods need similar information. The step of module definition differs between the methods. Jiao clusters the functional requirements with design parameters and evaluates technical performance and cost of the modules. Krause uses design for variety methods to form components, which he modularizes for each product life phase and then harmonizes across the phases. That difference may have its cause in the workshop-based approach of Krause, which is designed for use in the corporate context, while Jiao's approach is of an analytical nature.

5. Conclusion and Outlook

In this paper an information framework was derived by extracting domains, relations and their attributes from different modularization methods. The methods' information flow was traced on an abstract level using an MDM. This revealed the underlying scheme of each method, which enables comparison of different methods in a concise way. By combining method schemes a common meta-model can be extracted. Its evaluation indicated the core domains for modularization: physical and functional domains. Keeping in mind that most of the considered methods are technical-functional in nature, this study should be extended to include more modularization methods. Available case studies on the methods can be incorporated to sharpen their profile. A comparison of modularization theory and practical application using this approach could bring further insights.

Different sets of methods were developed, approaching modularization from different perspectives. Many methods and tools exist in isolation from each other [Simpson et al. 2012]. This framework fosters the combined use of methods by being transparent about their information needs. This is of special use in methodical toolkits like the integrated PKT-approach [Simpson 2014]. Information interfaces between method units for case-specific adaption of methodical tool chains can be identified. An information database for a methods toolkit can be implemented. The unifying nature of this approach contributes to consolidation of modularization methods.

References

Blees, C., "Eine Methode zur Entwicklung modularer Produktfamilien", Hamburg, 2010.

Göpfert, J., "Modulare Produktentwicklung", Dt. Univ.-Verl., Wiesbaden, 1998.

Hölttä-Otto, K., "Modular product platform design", Helsinki University of Technology, Espoo, 2005.

Jiao, J., Tseng, M. M., "A methodology of developing product family architecture for mass customization", 1999.

Kipp, T., "Methodische Unterstützung der variantengerechten Produktgestaltung", TuTech-Verlag, Hamburg, 2012.

Krause, D., Eilmus, S., "A Methodical Approach for Developing Modular Product Families", ICED 11, København, Culley, S., Design Society, Great Britain, 2011, pp. 299–308.

Krause, D., Ripperda, S., "An Assesment of methodical approaches to support the development of modular product families", Proceedings of ICED13 volume 9, Design Society, [S.l.], 2013.

Kreimeyer, M., Lindemann, U., "Complexity metrics in engineering design", Springer, New York, 2011.

Lindemann, U., Maurer, M. S., Braun, T., "Structural complexity management", Springer, Berlin, Heidelberg, 2009.

Maurer, M. S., "Structural awareness in complex product design", Dr. Hut, München, 2007.

Pimmler, T. U., Eppinger, S. D., "Integration analysis of product decomposition", ASME Conference on Design Theory and Methodology, 1994, pp. 343–351.

Rowley, J., "The wisdom hierarchy: representations of the DIKW hierarchy", Journal of Information Science. No.2, 2007, pp. 163–180.

Salvador, F., "Toward a Product System Modularity Construct: Literature Review and Reconceptualization", IEEE Transactions on Engineering Management, No 2, 2007, pp. 219–240.

Simpson, T. W., "Advances in product family and product platform design", Springer, New York, Heidelberg, Dordrecht, London, 2014.

Simpson, T. W., Bobuk, A., Slingerland, L. A., Brennan, S., Logan, D., Reichard, K., "From user requirements to commonality specifications: an integrated approach to product family design", Research in Engineering Design. No.2, 2012, pp. 141–153.

Simpson, T. W., Siddique, Z., Jiao, J., "Product platform and product family design", Springer, New York, London, 2006.

Steward, D. V., "The Design Structure System: A Method for Managing the Design of Complex Systems", IEEE Transactions on Engineering Management, No 3, 1981, pp. 71–74.

Suh, N. P., "Complexity", Oxford University Press, New York, 2005.

Dipl.-Ing. Jennifer Hackl, Scientific Assistant
Hamburg University of Technology, Institute of Product Development and Mechanical Engineering Design
Denickestr. 17 (L), 21073 Hamburg, Germany
Telephone: +49 40 42878 4466
Telefax: +49 40 42878 2296
Email: jennifer.hackl@tuhh.de
URL: http://www.tuhh.de/pkt