17th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2017

ESPOO, FINLAND, SEPTEMBER 11 - 13, 2017

Combination of Matrix-based and Graph-based Modeling for Product and Organizational Structures

Sebastian Schweigert[1], Thomas Luft[2], Sandro Wartzack[2], Udo Lindemann[1]

[1]Technical University of Munich, Chair of Product Development
[2]Friedrich-Alexander-Universität Erlangen-Nürnberg, Chair of Engineering Design

Abstract: As a consequence of volatile markets and agile development processes, products and organizational structures have become increasingly complex. Modeling and analyzing complex products and organizations via matrices and graphs are useful ways to handle this complexity. This paper combines approaches for modeling products and organizational structures with matrices on the one hand. On the other hand, graph-based methods for analyzing the dependencies within as well as between the product and organization domain are described. Combining these domains gives the possibility to improve the overall development process by minimizing iterations and by adjusting the organizational to the product structure.

Keywords: DSM, DMM, MDM, product modeling and analysis, organization modeling and analysis.

1 Introduction

The consideration of a wide variety of customer requirements and the collaboration in teams of diverse engineering disciplines is necessary to develop modern and successful products. These challenges result in complexity in the product development process (PDP). Thus, there is an increasing complexity in the domains, e.g. product, people, tools, data, and process (Herfeld et al., 2006; Eppinger and Browning, 2012; Bartolomei et al., 2011). Hence, it is essential to understand the complex relations between the elements of all domains. Particularly the interaction of the product and organization domain during the PDP represents a major challenge. This is due to the fact that organizational human interaction adds complexity to the PDP, for example in terms of conflicts or a lack of information exchange between different employees (Luft et al., 2013a). Until now, no holistic and useful approach can be found to address this issue and to support developers by mastering the described complexity.

Therefore, the overall aim of this paper is to apply a combined matrix-based and graph-based product model to analyze, visualize, and thus handle complex product and organizational structures. However, filling huge matrices in a manual way is very time-consuming and prone to errors, while modeling, analyzing, and visualizing systems via graphs needs special algorithms. Therefore, also computer-aided tools will be considered and subsequently used to support the matrix-based and graph-based modeling approach.

2 State of the Art

The fundamental basis for the paper is briefly summarized in the following two sections. Therefore, matrix-based methods for modeling are explained in section 2.1 and graph-based methods for analysis are presented in section 2.2.

2.1 Matrix-based modeling methods

The PDP is often a complex system consisting of various subsystems or rather domains (Lindemann et al., 2009; Eppinger and Browning, 2012). To understand this complexity, it is helpful to model the entire system. Therefore, various domains of interest (e.g. product, people) as well as possibly existing relations between the domains` elements can be considered (Browning, 2001). For modeling the correlation of diverse elements of a subsystem, the Design Structure Matrix (DSM) can be used (Steward, 1981). The DSM points out direct dependencies between elements of the same domain. The Domain Mapping Matrix (DMM) is suitable for direct dependencies between two different domains` elements. This matrix can also be considered as a cause-effect-matrix whose rows show elements of one domain (e.g. domain product), while its columns represent the elements of the other domain (e.g. domain organization). To show direct dependencies within a single domain as well as between different domains in one matrix, Maurer developed the Multiple Domain Matrix (MDM) (Lindemann et al., 2009).

Apart from the previously considered direct dependencies, understanding and analyzing indirect dependencies is crucial for the improvement of PDPs. Indirect dependencies result from the relation of two elements of specific domains via one or more elements of another domain. In an MDM, for example, an employee E_1 of the domain “organization” generates a product element (e.g. a product component) and another employee E_2 uses it. Thus, the two employees E_1 and E_2 are indirectly dependent, although they are not directly linked (e.g. by belonging to the same department). Capturing these indirect dependencies in so-called “derived DSMs” is possible through matrix multiplications of various matrices of the MDM (Luft et al., 2013a).

2.2 Graph-based analysis methods

The utilisation via graphs is just a possible visualisation form of matrices (cf. e.g. Keller et al., 2006; Eppinger and Browning, 2012). In comparison to matrices, graphs do not contain any aditional information. However, as humans perceive 70 % of all information visually (Eiselt et al., 2013), graphs can have a major advantage to represent information and to support communication. This is especially important as many decision makers are used to graph-based visualizations (e.g. from workflow charts), while matrices are less common. Accoriding to Kreimeyer (2009), graphs have the following properties. They can have directed (“diagraphs”) or undirected edges. There are also graphs with both types of edges (“mixed graphs”). Graphs can contain weighted edges (“weighted graphs”) and an edge can connect a node to itself (“loop”). Graphs without loops are called “simple graphs”. Graphs can contain edges with multiple edges in-between (“multi graphs”). For complex analyses, graphs can have further properties (Kreimeyer, 2009).

There are many IT-tools that can be used for the modeling of graphs. Luft and Wartzack (2016) give a good overview of available systems. Another system, which is not included in that summary, is Soley Studio. This is a new software to manage and analyze data in an engineering context and it is used for the analyses in this paper. It uses the graph rewrite generator GrGen.NET, which allows specifying typed and attributed multigraphs. This programming language is especially suitable for graph transformations (Helms, 2013; Geiss et al., 2006; Jakumeit et al., 2010).

3 Matrix-based and Graph-based Product Modeling

The matrix based product model (cf. Figure 1) systematically maps all product elements and their corresponding relations (Luft et al., 2013b). Thus, all requirements (REQ), behavioral (B) aspects, function structures (FS), active principles (AS), properties (P) of the overall system (OS) and the subsystems (SS), as well as all characteristics (C) of the individual components (CP) are modelled. It contains DSMs on the diagonal of the matrix (e.g. characteristics-characteristics-matrix) and DMMs (e.g. characteristic-property-matrix) located above or below the diagonal (Krehmer, 2012). As shown in the schematic model in Figure 1, each matrix field is an n*m matrix. It is possible to expand the matrix-based product model flexibly to more subsystem levels, if necessary. Thus, the OS can be structured in subsystems, modules, and submodules.

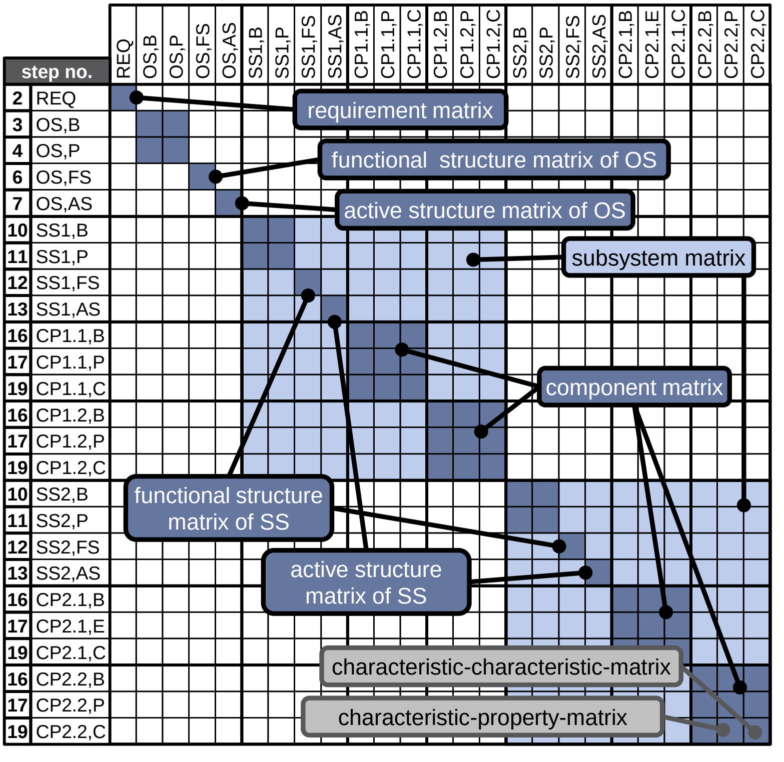

Figure 1: Simplified structure of the matrix-based product model (Luft et al., 2014a)

With the development of new products, the product properties are only realizable by determining certain characteristics (Weber, 2005). Thus, when using the matrix-based product model, it is necessary to create characteristic-property-matrices for all components (cf. Crostack et al., 2015). Afterwards, all necessary characteristics (e.g. diameter, length) need to be determined for several components in characteristic-characteristic-matrices (cf. Luft et al., 2014b). The matrix-based product model relates all various requirements with the product´s characteristics and properties at different levels. Because of the stepwise procedure (see Luft et al., 2013b), all relations between the elements of the product model are modelled systematically. Furthermore, due to more visibility and traceability regarding the effects of characteristic changes, developers can decide more easily which characteristic should be modified to reach certain properties (cf. Luft et al., 2017).

According to previous analysis of Luft & Wartzack (2016), the commercial software LOOMEO turned out to work well with the modeling, analyzing, and visualizing of product models. It is focused on structural complexity management and combines matrices with force-directed graphs and diagrams. Thus, it is possible to represent and manipulate system elements and identify complex relations. Hence, developers can create a matrix-based product model (see Figure 2).

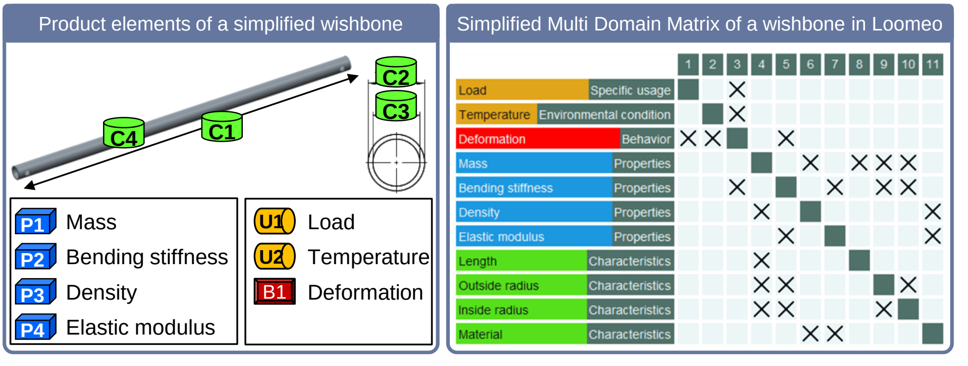

Figure 2: Matrix-based representation of the product model (Luft et al., 2014a)

4 Matrix-based and Graph-based Organization Modeling

Matrix-based and graph-based modeling is also useful for elements of the organization domain. Especially inter-departmental collaboration can form complex systems that have to be handled. A previous paper proposed an approach for the collaboration of design and simulation departments (Schweigert et al., 2016). This paper extends this approach and combines it with the product domain. The overall goal is to manage complexity and thereby to identify the right measures to improve the situation of a specific company. The approach consists of the following steps (cf. Figure 3). The first step is the data collection via workshops, checklists or from existing sources (e.g. data bases, software tools). In the next step, the situation is described by transferring the collected data into a system graph. Afterwards, the situation analysis follows through semi-automated identification of critical areas with graph transformations as proposed in Heckel (2006) and Kissel (2014), and algorithms, including the derivation of characteristic numbers. “Critical” in this context is used for missing relations (cf. section 5). The fourth step is the linkage of identified barriers and appropriate measures via the characteristic numbers, including aspects from communication science. The last step is the implementation by presenting the barriers and according appropriate measures in workshops and expert interviews, implementing measures, and changing organizational structures and IT landscapes.

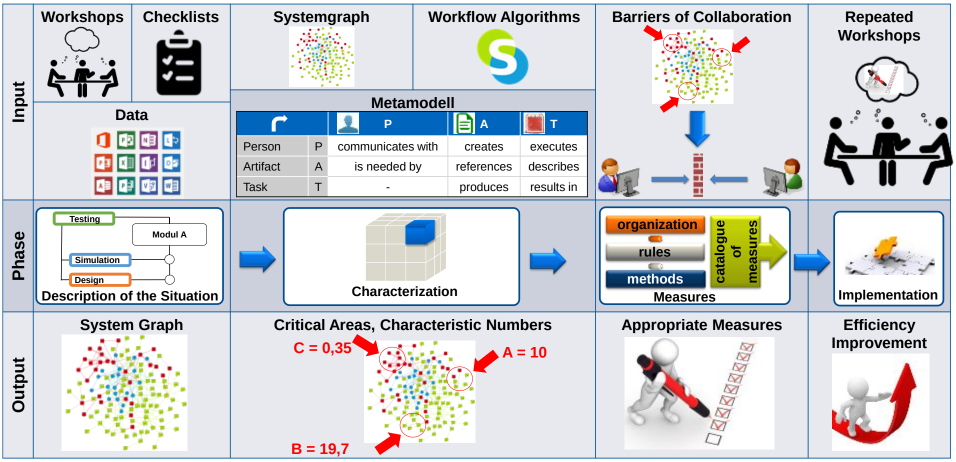

Figure 3. Overview of the approach for organizational modeling and analysis

5 Combination of Product and Organizational Modeling

In the case study, real data from the development of a bicycle was used to model components and requirements. As this is a concept paper, the domain "person" and the connections between persons and the other domains were added artificially. The metamodel in Figure 4, which consists of the three domains "person", "component" and "requirement" and six relation types, is used to model and analyze this system.

		P	C	R
Person	P	Communicates with	Responsible for	Responsible for
Component	C		Connected with	Fulfils
Requirement	R			Connected with

Figure 4. Metamodel and system graph of the case study

In the first step according to the methodology in Figure 3, the collected data is transferred into a graph as depicted in Figure 4 on the right. The second step "characterization" for the analysis of a system is oriented on the Goal-Question-Metric (GQM) approach as described by Koziolek (2008), which was developed by Basili and Weiss (1984) and later extended by Basili et al. (1994). It always states a question that is formulated to reach a certain goal in the process of understanding and characterizing a system. Derived from this question, a metric is formulated that is able to answer the question in a quantitative way.

Figure 4 shows the whole system graph of the case study on the right hand side. Through the coupling of product modeling and organizational modeling, communication needs arise (for similar approaches see Chucholowski et al., 2016 and Sosa et al., 2004). The communication needs are a result of the connections between components and requirements respectively and the fact that there is always a person that is responsible for the components and the requirements. After identifying these communication needs, the system can be improved as specific communication channels between persons are made mandatory or responsibilities are manipulated. When doing this in regard to communication needs arising from connected requirements, this also influences the communication needs in regard to components. This interdependency has to be taken into account when implementing measures according to the approach described below.

The algorithm depicted in Figure 5 is used to analyze the system of this case study. To reach the goal of an improved communication, the question is asked: “Do persons communicate if they are connected via components or requirements?” In order to answer that question, the number of ideal relation chains (left side in Figure 5) and critical relation chains (right side in Figure 5) is determined. Ideal and critical relations are defined as follow:

- If two persons that are connected indirectly via two components or requirements are connected directly by a “communicates with” relation, this is called “ideal relation chain”.
- If no connection exists between two persons that are connected indirectly either via two components or two requirements, a so called “critical relation chain” is identified.

The ratio between ideal relation chains (IR) and critical relation chains (CR) is called the communication ratio metric (CRM, cf. Equation 1). The higher the communication ration metric in a system, the more people do not communicate with each other, even though they should as they have an interface through a requirement or a component. As a result, the higher the CRM, the higher is the probability of redundant iterations or duplications due to a lack of communication.

$$CRM = CR/(IR+CR) \qquad (1)$$

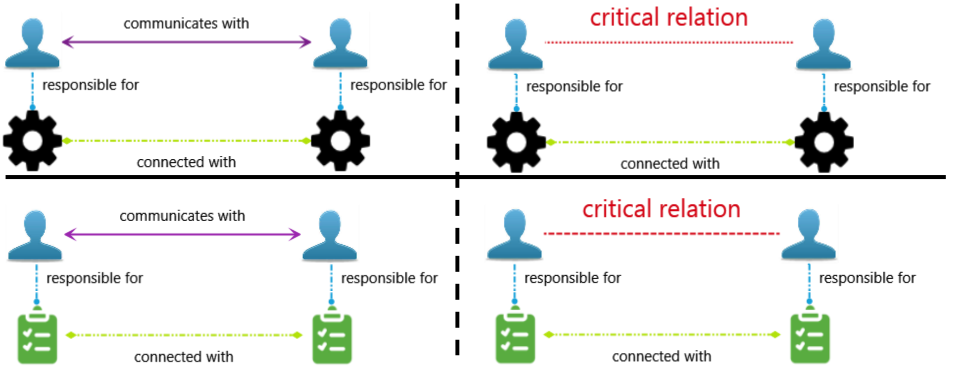

Figure 5. Ideal relations (left) and critical relations (right) between persons

Applied to the case study, the metrics in Figure 6 (left) are calculated and visualized in Figure 7 (left). To improve collaboration, it is decided to separate responsibilities within the development team more clearly. As the person "Fisher" is part of three critical relation chains regarding to requirements and is also responsible for many components, his requirement responsibilities are transferred to person "Doe". As a result, "Fisher" is only responsible for components, while "Doe" is only responsible for requirements. To analyze, whether this really has a positive effect on communication needs, the ratios metrics are calculated again (see Figure 6, right) and visualized (see Figure 7, right).

Before improvement	After improvement
No. of relation chains via components: 19 Ideal: 17 Critical: 2	No. of relation chains via components: 14 Ideal: 13 Critical: 1
Communication needs ➢ Smith-------King ➢ Doe-------Miller	Communication needs ➢ King-------Smith
No. of relation chains via requirements: 10 Ideal: 3 Critical: 7	No. of relation chains via requirements: 9 Ideal: 4 Critical: 5
Communication needs ➢ Anderson ⇆ Fischer ➢ Snider ⇆ Cook ➢ Farmer ⇆ Anderson ➢ Doe ⇆ Snider ➢ Fischer ⇆ Snider ➢ Anderson ⇆ Snider ➢ Fischer ⇆ Cook	Communication needs ➢ Anderson ⇆ Snider ➢ Anderson ⇆ Farmer ➢ Doe ⇆ Snider ➢ Cook ⇆ Snider ➢ Farmer ⇆ Cook
Communication Ratios Metrics (CRM) ▪ CRM_Components: 2/19 = 11 % ▪ CRM_Requirements: 7/10 = 70 %	Communication Ratios Metrics (CRM) ▪ CRM_Components: 1/14 = 7 % ▪ CRM_Requirements: 5/9 = 56 %

Figure 6. Results before and after the separation of responsibilities

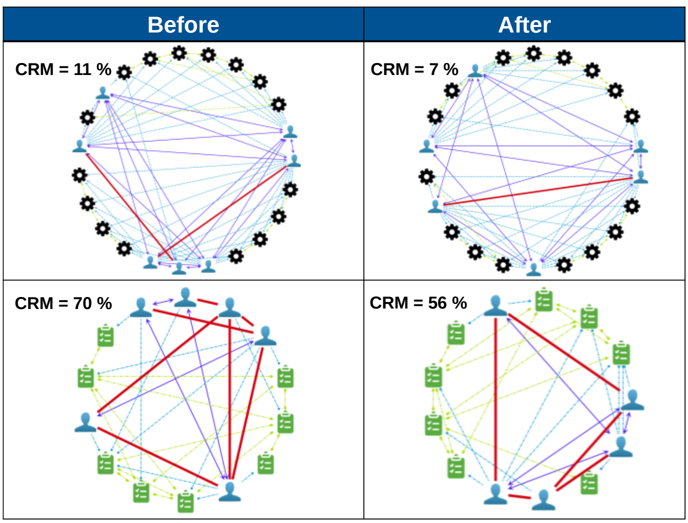

Figure 7. Communication ratio metrics before and after the separation of responsibilities

Thus, in this case, the communication ratio metric for both perspectives can be decreased. As the team that deals with requirements becomes smaller, the number of total relation chains decreases naturally. Within these chains, the number of communication needs and therefore the communication ratio metric regarding requirements decreases as well. While this was to be expected as the separation of responsibilities was aiming at the requirements team, it is interesting to see that the communication ratio metric for the component view decreases as well (11 % to 7 %).

These insights can now be used to test different team compositions and identify the most suitable allocation of tasks among the team members. To do so, only little changes have to be made in the underling matrices of the graph model while the algorithms can just be executed on the new data. The results are then represented in a graph view what makes it easy to discuss them with responsible managers.

6 Conclusion and Outlook

The software tool Soley Studio proves to be useful in tackling complexity problems as shown in this case study. The different visualization views support developers in restructuring and improving the arrangements of the product architecture and the organization simultaneously. Through graph transformations and the calculation of characteristic numbers like the communication ratio metric, change propagations and effects of certain measures can be analyzed.

From a practitioner's point of view, our approach is a system modeling and analyzing tool that is easy to adjust to the specific situation of a company. New team composition scenarios can be evaluated quickly and numerically to form a basis for decision-making. Furthermore, the combination of graphical representation and matrix-based documentation of both product data and organizational structures allows a holistic view that supports the managing of complex systems going from requirements over team composition as far as to the component level.

The scientific contribution of this approach lies in a metric for measuring the communication needs between different disciplines within and across design teams. Also, the approach proved promising when not only looking for the avoidance of communication needs but also for testing different scenarios to reach an overall optimal result, taking into account different views on the system like requirements management and product architecture.

The next steps are the integration of additional domains, like the knowledge and the process domain into the graphs and matrices. In the future, further investigations and evaluations to ensure the practicality of the tool are useful and therefore planned within product development use cases at the chairs as well as with industrial partners. This also includes the comparison with findings from case studies in literature like Sosa et al., 2004 and the implementation of further algorithms to transfer metrics from network analysis into the domains described herein. In combination with findings from parallel research projects like expert interviews in the field of interdisciplinary design and the resulting barriers and measures, new metrics can be developed and integrated into the approach.

Acknowledgment

The authors of the FAU Erlangen-Nürnberg thank the German Research Foundation (DFG) for supporting the research project with the title "Product-oriented process management – Computer-aided modeling as well as graph-based analysis and visualization of the matrix-based product description".

Furthermore, the authors thank Mousaab Askour for his support in the development of the algorithms.

References

Bartolomei, J.E., Hastings, D.E., de Neufville, R., Rhodes, D.H., 2011. Engineering Systems Multiple-Domain Matrix: An Organizing Framework for Modeling Large-Scale Complex Systems. Systems Engineering, 15, (1), 41-61.

Basili, V.R., Caldiera, G., Rombach, H.D., 1994. The Goal Question Metric Approach. In: Encyclopedia of Software Engineering - 2 Volume Set, pp. 528-532.

Basili, V.R., Weiss, D.M., 1984. A Methodology for Collecting Valid Software Engineering Data. IEEE Transactions on Software Engineering 1, 728–738.

Browning, T.R., 2001. Applying the design structure matrix to system decomposition and integration problems. IEEE Transactions on Engineering Management 48, 292-306.

Browning, T.R., Eppinger, S.D., 2002. Modeling Impacts of Process Architecture on Cost and Schedule Risk in Product Development. IEEE Transaction on Engineering Management 49, 428-442.

Chucholowski, N., Kriegler, J., Hollauer, C., Kattner, N., Becerril, L., Weidmann, D., Lindemann,. U., 2016. Systematic Partitioning in Mechatronic Product Development by Modeling Structural Dependencies, in: Monteiro de Carvalho, M., Eppinger, S.D., Maurer, M., Becerril, L. (Eds.), Proc. of the 18th Internat. DSM Conference. Sao Paulo, pp. 57-66.

Crostack, A., Luft, T., Binz, H., Wartzack, S., Roth, D., 2015. Konzept eines erweiterten Produktbeschreibungsansatzes basierend auf dem Vergleich des Characteristics-Properties-Modelling und der matrixbasierten Produktbeschreibung. In: Binz, H., Bertsche, B., Bauer, W. Roth, D. (Eds.), Stuttgarter Symposium für Produktentwicklung, Stuttgart: Fraunhofer, pp. 1-10.

Eiselt, T., Zickner, H., Schuldt, J., Gröger, S., Weidlich, D., 2013. Interdisciplinary Product Development-Virtual Reality Application in FMEA. Industrial and Systems Engineering Review, 1 (1), 59-67.

Eppinger, S., Browning, T., 2012. Design Structure Matrix Methods and Applications. MIT Press, Cambridge.

Geiß, R., Batz, G. V., Grund, D., Hack, S., Szalkowski, A., 2006. GrGen: A Fast SPO-Based Graph Rewriting Tool, in: Corradini, A., Ehrig, H., Montanari, U., Ribeiro, L., Rozenberg, G. (Eds.), Internat. Conf. on Graph Transformations, Natal, Springer. pp. 383-397.

Heckel, R., 2006. Graph Transformation in a Nutshell. Electronic Notes in Theoretical Computer Science. 148, 1, pp. 187-198.

Helms, B., 2013. Object-Oriented Graph Grammars for Computational Design Synthesis. PhD Thesis, Technical University of Munich, Munich.

Herfeld, U., Kreimeyer, M., Deubzer, F., Frank, T., Lindemann, U., Knaust, U., 2006. Verknüpfung von Komponenten und Funktionen zur Integration von Konstruktion und Simulation in der Karosserieentwicklung. VDI-Berichte, 1967, 259-276.

Jakumeit, E., Buchwald, S., Kroll, M., 2010. GrGen.NET. International Journal on Software Tools for Technology Transfer 12, 3-4, 263-271.

Keller, R., Eckert, C.M., Clarkson, P.J., 2006. Matrices or node-link diagrams: which visual representation is better for visualizing connectivity models? Information Visualization 5, (1), 62-76.

Kissel, M.P., 2014. Mustererkennung in komplexen Produktportfolios. PhD Thesis, Technical University of Munich, Munich.

Koziolek, H., 2008. Goal, Question, Metric, in: Eusgeld, I., Freiling, F.C., Reussner, R. (Eds.), Dependability Metrics. Springer, Heidelberg, pp. 39-42.

Kreimeyer, M., 2009. A Structural Measurement System for Engineering Design Processes. PhD Thesis, Verlag Dr. Hut, Munich.

Luft, T., Bochmann, J., Wartzack, S., 2013a. Enhancing the Flow of Information in the PLM by Using Numerical DSMs – An Industrial Case Study, in: Bernard, A., Rivest, L., Dutta, D. (Eds.), Product Lifecycle Management for Society. IFIP Advances in Information and Communication Technology, 409. Springer, Berlin, pp. 90-99.

Luft, T., Krehmer, H., Wartzack, S., 2013b. An advanced procedure model for property-based product development, in: Lindemann, U., Srinivasan V., Kim, Y.S., Lee, S., Clarkson, J., Cascini, G. (Eds.), Proc. of the 19th Int. Conf. on Engineering Design, Seoul, pp. 187-196.

Luft, T., Ewringmann, N., Wartzack, S., 2014a. Application and validation of the matrix-based product description in a case study by using the software Loomeo, in: Moroni, G., Tolio, T. (Eds.), Proc. of the 24th CIRP Design Conf., pp. 479-484.

Luft, T., Klein, F., Wartzack, S., 2014b. Weight and cost reduction of the thermal management for traction batteries by using the DSM methodology. in: Marjanovic, D., Storga, M., Pavkovic, N., Bojcetic, N. (Eds.), Proc. of the 13th Internat. Design Conf., Dubrovnik, pp. 293-302.

Luft, T., Schuh, P., Wartzack, S., 2017. A Software Tool for the Graph-Based Visualization of Complex Relations Between Product Properties and Characteristics. in: Chakrabarti, A., Chakrabarti, D. (Eds.), Research into Design for Communities, Volume 1 Proceedings of ICoRD 2017. Singapore: Springer Nature, pp. 745-756.

Luft, T., Wartzack, S., 2016. Application, evaluation and future research potential of the matrix and graph based product model, in: Marjanovic, D., Storga, M., Pavkovic, N., Bojcetic, N., Skec, S. (Eds.), Proc. of the 14th Internat. Design Conf., Dubrovnik, pp. 351-360.

Lindemann, U., Maurer, M., Braun, T., 2009. Structural Complexity Management - An Approach for the Field of Product Design. Springer, Berlin.

Reitmeier, J., Luft, T., Wartzack, S., Paetzold, K., 2015. Consideration of uncertainties in the product-oriented development process, in: Schabacker, M., Gericke, K., Szélig, N., Vajna, S. (Eds.), Modelling and Management of Engineering Processes. Berlin: Springer, pp. 137-149.

Schweigert, S., d'Albert, H., Lindemann, U. 2016. Enhancement of Collaboration and Communication between Design and Simulation Departments by Methods of Requirements Engineering, in: Marjanovic, D., Storga, M., Pavkovic, N., Bojcetic, N., Skec, S. (Eds.), Proc. of the 14th Internat. Design Conf., Dubrovnik, pp. 351-360.

Sosa, M.E., Eppinger, S.D., Rowles, C.M., 2004. The Misalignment of Product Architecture and Organizational Structure in Complex Product Development. Management Science, 50 (12), 1674-1689.

Steward, D.V., 1981. Design structure system: A method for managing the design of complex systems. IEEE Transaction on Engineering Management 28, 71-74.

Weber, C., 2005. CPM/PDD - An extended theoretical approach to modelling products and product development processes, in: Bley, H., Jansen, H., Krause, F.-L., Shpitalni, M. (Eds.), Proc. of 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes, Fraunhofer Stuttgart, pp. 159-179.

Contact: S. Schweigert, Technical University of Munich, Chair of Product Development, Boltzmannstrasse 15, 85748 Garching, Germany, Phone: +49 89 289 15126, Fax: +49 89 289 15144, sebastian.schweigert@pe.mw.tum.de, www.pe.mw.tum.de