INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

NETWORKED PRODUCT MODELLING – USE AND INTERACTION OF PRODUCT MODELS AND METHODS DURING ANALYSIS AND SYNTHESIS

Frank Deubzer and Udo Lindemann
Institute of Product Development, Technische Universität München

ABSTRACT
The paper proposes a procedural model for the systematic definition of product architectures by gradual functional decomposition and assigning possible technical solutions to the functions on different hierarchical levels. The framework is designed for the early phases in product development, where current methods can only support a certain state of detail of the desired solution. For that matter, different methods for analysis and synthesis are applied, ranging from graph depiction and working method to rule-based algorithms and analysis methods. A vehicle power train and possible variations thereof are provided as an example. The main success factors for the efficient and reliable conduction of the early phases of product design, based on the discussion of the situation and available methods are the systematic synthesis of valid solutions based on available subsystems, the depiction and management of the complete solution space and the definition of assessment criteria. As existing approaches can only support a certain phase in early product design, the presented framework allows for the achievement of these objectives, compensating the shortcomings of singular methods.

Keywords: design synthesis, functional modeling, decision making, complexity management

1 MOTIVATION – SHORTCOMINGS IN THE EARLY PHASES OF DESIGN
In early phases of product design, meeting overall design targets such as cost and quality is crucially influenced by the chosen product architecture, implemented functions, realized modules, use of technologies etc. Time pressure forces designers to make decisions based on assumptions, although the defined product characteristics at that point of the development process are crucial for the success of the product. The influence onto product properties, such as performance, quality, and cost is vast at this point of product definition. Incorrect decisions might cause immense costs for product changes, generate unsuccessful products due to poor product quality or functions, or simply cause losses due to not tapping the full potential of the product.

1.1 Situation
When starting the product development process, the initial situation is in most cases characterized by the existence of a previous product or product portfolio. Only a small percentage of product design tasks start from scratch. The main design task therefore is normally the product improvement in terms of meeting more demanding design targets and requirements [8]. Given this initial situation, a number of subsystems, variants, modules and components are known and their properties familiar. Important design decisions are often not based on the evaluation of the solution space. The experience of designers who are extrapolating values of properties, following the company culture or concentrating on established technologies and concepts are the main factors influencing the decision making. This behavior has a number of causes. As numerous existing variants and solutions of subsystems cause a large number of possible combinations to form product architectures, the solution space turns out to be very complex. First, subsystems are dependent onto or rejecting each other. Second, the complete solution space – the amount of valid product solutions – is unknown in its entirety. Third, especially the behavior of the chosen product architecture, interaction of functions and dependencies of subsystems are fairly unpredictable [10, 11]. To gain knowledge about system properties, prototypes – virtual or physical models of the product – are set up to analyze the product's behavior. The development, start-up and validation of the required models is consuming both time and money, as the

final properties can only be measured and discussed with the access to a fairly complete physical model of the product. Iterations and setbacks are the result, making the process of product architecture definition, evaluation of solutions, and decision-making expensive in terms of time to market and development costs. To avoid drawbacks and risks during design, innovative technologies and concepts are repressed, and innovations are reduced to incremental improvements, not allowing for the successful implementation of new product architectures. As a result, subsystems and singular components are improved, but the overall revolution in design is missing.

Summing up, depicting the entire solution space and thus supporting the decision making in complex product design is often prohibited due to unpredictable behavior of the system, complex interdependencies of subsystems, and the mere size of the solution space. Incremental innovations of subsystems – enhancements of existing technologies – are the result, lacking the radical character of innovations in product architectures and overall systems.

1.2 Arising questions

To overcome this stalled situation, methodical support during the early design phases must enable an efficient and reliable process, leading to new product architectures and innovative products to accomplish radical improvements. A number of questions remain to be answered before such methodical support can be realized.

As the level of detail evolves during the early phase of design, it has to be clarified on which level of detail the major contribution for the definition of innovative solutions can be generated. Levels of detail contain for example the system integration level, system and subsystem level, component level, technologies, physical effects level, material etc. Different levels of detail allow for different methodical support and the outcome of the method application differs according to the considered product, as the results section will show. Physical simulation models, for example, allow for the detailed analysis of certain materials, whilst the more abstract application of functional models enhances the rudimentary change of a product’s functional behavior. Both methods allow for the generation of innovative solutions and widening of solution space, but on completely different levels of detail – component and architectural level [11].

Furthermore, characteristic design targets, requirements, and their matching product properties demand a certain level of detail for the definition of innovative solutions. For example, a cost assumption requires knowledge about the used material and the necessary amount thereof.

Accordingly, the methods for evaluation have to support different levels of detail. The complexity of the solution space has to be managed by the iterative confinement on different levels of detail, keeping the amount of time and cost in conceptual design down.

The arising questions to be answered are: Can different levels of detail be integrated in a comprehensive approach, covering the entire solution finding process? Can different system properties be evaluated on different levels of detail? Which methods are appropriate for different levels of detail?

2 RELATED WORK AND REMAINING CHALLENGES

Goal of the presented research is the analysis of different methods, their assignment to different levels of detail, and the proposition of a procedural model or framework to benefit most from different methods. A combination of chosen methods should allow for a holistic and systematic approach to product architecture definition, enhancing decision-making by depicting the solution space, and enabling the analysis thereof. The following discussion of existing approaches gives an overview on relevant research utilized for the presented approach, not claiming to give a complete overview on the presented field. The approaches and methods were chosen by their capability to cover different levels of detail and different degrees of automated support of the design process.

Figure 1 gives an overview onto the presented methods for analysis and synthesis. A differentiation was made in each section (analysis and synthesis) from top to bottom, according to the possible level of automation of the methods. The listing starts with working methods, executed manually, followed by matrix-based approaches, which allow for partial automation during the analysis process. Last in the list of analysis methods are calculation and simulation methods, allowing for a high level of automation when processing and analyzing different variants of a system model.

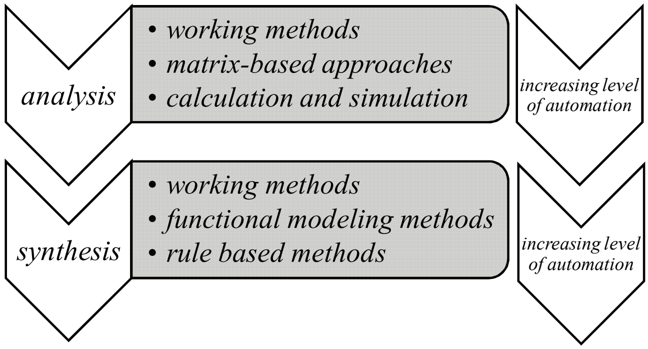

Figure 1. Utilized Methods – Selection

2.1 Analysis of product architectures

Working methods in the given context are methods supporting the product development process by providing guidelines, models and tools for the designers. They provide a variety of approaches for the analysis and assessment of product architectures [8] and are executed manually. Central are the functional models, for example the flow-oriented functional model by Ehrlenspiel [4]. The model represents information-, material- and energy-flows from a functional perspective. The functional model provides a sound basis for design synthesis by formulating functions as neutral to technical solutions as possible. Further functional modeling methods supporting design synthesis are for example the relation-oriented models known of the TIPS methodology or elementary hierarchical models (see [4, 8] for more comprehensive overviews). To assess solutions in the early phases of design, numerous methods, such as FMEA or the use of weighted factors for different solutions are appropriate approaches for the differentiation and manual evaluation of product concepts [8].

Matrix-based approaches for system analysis, the well-known Design Structure Matrix (DSM) [2] method and the evolved Multiple-Domain Matrix (MDM) [6] allow for a thorough analysis of system structures. The representation in matrices and graphs enables the application of generic algorithms and measures to analyze and improve systems from a structural perspective. Main advantage is the applicability onto processes, products or other systems, taking the dependencies between system elements into account. The level of automation of the analysis with matrix methods exceeds the manual execution of working methods, as algorithms are applied, allowing for the systematic analysis of standard structural characteristics and metrics [3, 5].

Finally, the simulation of virtual prototypes marks the most automated level of design analysis, requiring detailed physical models of the product. The advantage is therefore the detailed assessment of solutions, with the downside of requiring large efforts in terms of time and product knowledge to define the models. The presented research aims at the preselecting of product architectures and thus does not consider the simulation of the product, which is considered the next step in the procedural model.

2.2 Synthesis of product architectures

As for the analysis of product architectures, working methods for the synthesis exist, allowing the design of products systematically. The methods can be divided into methods for solution finding and creativity (e.g. TIPS, synectics) and methods for the systematic composition of subsystems or partial solutions such as the morphological matrix or the compatibility matrix [8]. Problems arise when large solution spaces exist, which is why the combination with more automated methods is recently being conducted [7].

The functional modeling methods, also mentioned amongst the analysis methods, build a profound basis for the product architecture synthesis, as is shown by the well-known TIPS-methodology or the Contact & Channel Model [13]. Due to the solution-neutral description, functional models are in general considered as an important step for product synthesis [10].

Another perspective on design synthesis is provided by rule-based or grammar-based approaches such as expert systems, FBS (Function Behavior Structure) systems or design grammars (see [5, 12] for example). Also using the functional perspective as basis, these approaches require rules for the coupling of functions as well as the coupling with parts fulfilling the formalized functions. Rule-based methods require a large amount of information as a starting point, with the results reflecting the information input. Most important benefit of the mentioned approaches is the automation of the synthesis process and the definition of possible solutions, requiring a profound set of rules.

2.3 Remaining Challenges

The discussion of existing approaches has shown that promising methods for design synthesis and analysis exist. The methods are usually addressing a certain level of detail for analysis and synthesis, lacking a comprehensive approach for the design of complex systems, supporting the gradual confinement of the solution space.

The main success factors for an efficient and reliable methodical support of the early phases of product design, based on the discussion of the situation and available methods are: the systematic synthesis of valid solutions based on available subsystems, the depiction and management of the complete solution space and the definition of assessment criteria. The assessment criteria have to match the available level of detail and thus may not be applicable at all early phases of the design process. As the overview of different methods has shown, only the combination of different approaches allows for the achievement of these objectives, compensating the shortcomings of singular methods.

Design Theory provides a variety of methodologies addressing challenges also addressed by the presented approach. Certain mindsets, such as the differentiation of functional and physical space of Axiomatic Design [14] for example, are integral parts of the presented approach. Different “spaces” or “domains” are also considered by the Theory of Domains [15] and provide a similar point of view as does the recursive part of the presented framework. Nevertheless, the presented approach is rather defined to purposefully combine the iterative and recursive application of methods on differing levels of detail than providing a completely revised view onto engineering design. Focus is the evaluation of methods according to different available levels of system properties and their iterative and recursive implementation.

3 A FRAMEWORK FOR GRADUAL FUNCTIONAL DECOMPOSITION

Given the requirements for a comprehensive framework, the existing methods were combined to a framework (Figure 2) for the definition of product architectures in the early phases of design, based on the existence of subsystems and elements derived from an existing product portfolio. The framework allows for the decomposition of a product’s functions and the composition of product architectures. In the following paragraphs, the framework will be briefly described, while section 4 will provide an example of the procedure’s outcome.

The first step consists of the definition of a starting level of detail. Therefore, existing product architectures of the type to be designed are analyzed, and different functional models derived thereof.

The functional models are divided into the class of system functions and the class of user functions to allow for a broader spectrum of assessment criteria (Figure 3). Thus, the architecture’s properties can be evaluated as well as their customer benefit. As a flow-oriented model is used for the system function model, the user functions are modeled in hierarchical order.

Given the functional description of the system, the solution space can be outlined by defining the functions necessary for a fully functional product architecture. Only architectures fulfilling the defined necessary functions are valid. For that matter, system elements and subsystems can be assigned to the different functions, defining the solution space. Valid solutions can be generated by establishing a consistency matrix of system elements (see the example in section 4 for details).

The fulfillment of user functions serves as assessment criterion in the next step of the framework. The system functions can be evaluated by the quality of assigned system elements whose properties can be defined according to the available level of detail. As an outcome of this step, a number of preselected architectures which are rated high according to the assessment criteria are defined.

The last step consists of the analysis of the architectures’ properties, again according to the available level of detail. Properties can be compared and weighted, enabling the identification of promising solutions according to the given requirements.

As a result, the promising solutions can now be evaluated by further criteria on a more detailed level. For this process of detailing, the number of valid solutions has been reduced to a reasonable number of solutions in the previous steps. Functional modeling can be detailed if necessary and the solution space generated according to that level of detail.

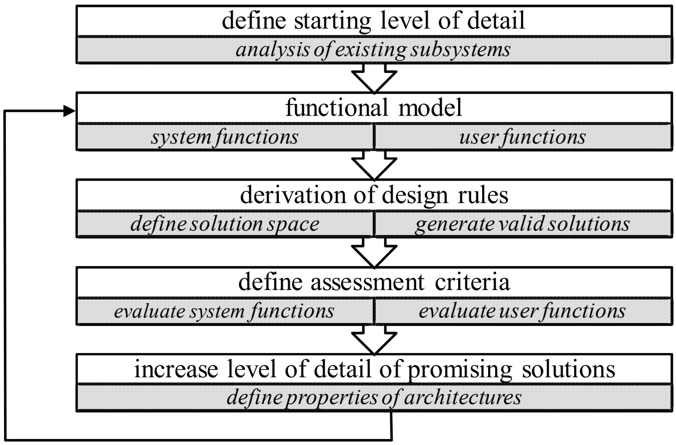

Figure 2. Framework for Gradual Functional Decomposition

4 APPLICATION – VEHICLE POWER TRAIN

To give an example for the application of the approach, a vehicle power train was chosen. Given the fact that the actual environmental and economic situation requires innovative solutions for the mobility of individuals, the need for radical innovations is apparent. In the following, the system is depicted from the point of view of the energy flow, leaving other measures such as cost or package aside.

4.1 Starting Level of Detail

The analysis of existing architectures lead to a hierarchical description of the system as is depictured in figure 3. Core functions of the system are the storage of energy, the conversion and the use of energy. As a starting level of detail, this level provides a generic view onto the system's functions.
User functions are defined as functions actively used, chosen and experienced by the user. Examples are the ability to drive with electrical or conventional power or the ability to boost by doing both.

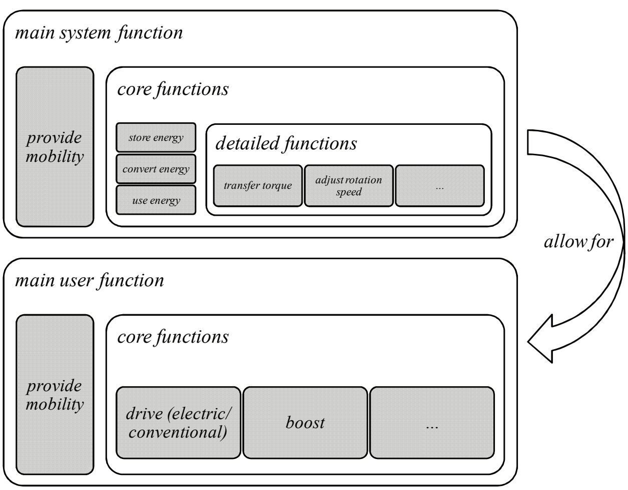

Figure 3. Functional classes and hierarchies

4.2 Functional Modelling

In the second step, the functional models of the system are derived from the analysis. The highest level of abstraction is given in figure 4, which is a general model consisting of the three core functions, depicting all known vehicle power trains. The modeling method in the style of the flow-oriented functional model [acc. to 4] was adapted to allow for the depiction of different energy states in one model, and is thus not following the semantics of the method exactly. Nevertheless, the definition of new architectures requires more detail to be able to generate solutions.

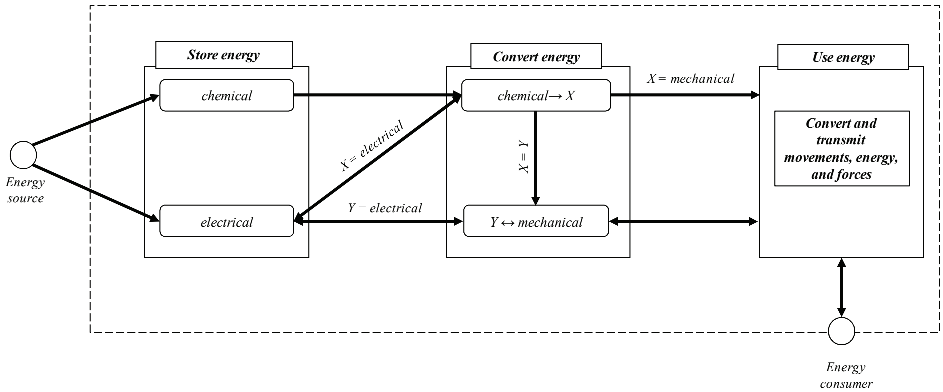

Figure 4. System functions – example: vehicle power train (abstract level)

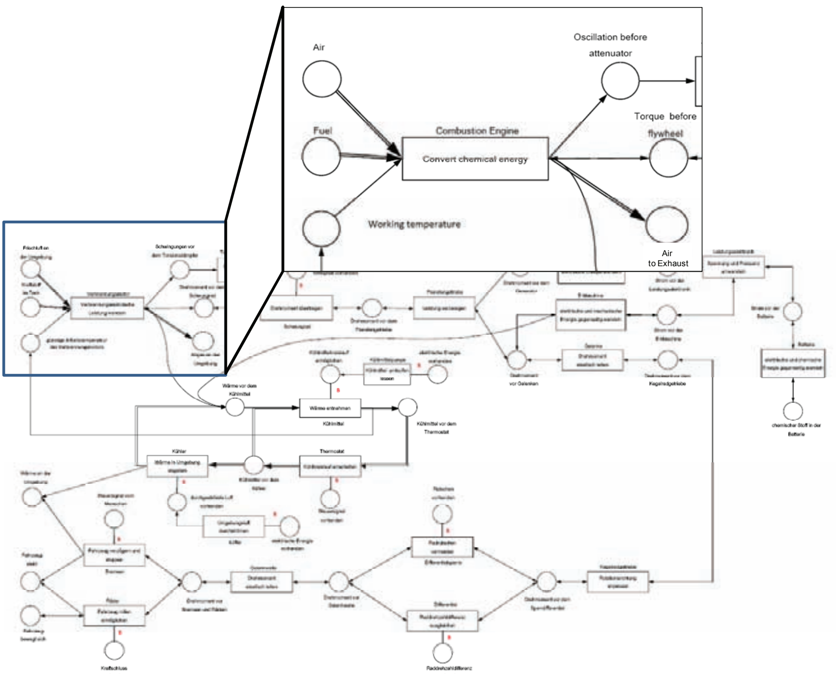

Figure 5. System functions – example: vehicle power train (detailed level)

Figure 5 shows one example of a more detailed flow-oriented functional structure [4], depicting the conventional power train with internal combustion engine in a systematic view. On that level, particular functions can be identified and possible technical solutions – system elements – can be assigned to the functions (such as the flywheel or the combustion engine in the given example), which can be replaced by other technical solutions fulfilling the required functions. Different user functions can be identified as paths within the model of the system's functions, as it depicts the energy flow.

4.3 Derivation of Design Rules

As a next step, the solution space has to be defined, thus enabling the definition of product architectures. System elements are assigned to the functions, and their dependencies amongst each other are generated based on the functional connections by the system given in table 1. Given these combination rules, the existing system elements can be combined within a consistency matrix, wherein the designer is able to configure valid solutions for the system architecture manually. The composition of the matrix is conducted according to the analysis of the functional models and the assignment of physical parts fulfilling certain functions. By the derivation of that information, physical components are interrelated by dependency types 4 or 5 if their combination is existing within the functional model, or by 1 to 3 if their combination is not existent in the solution space provided by the functional models.

Table 1. Interdependencies of system architecture elements

Dependency Type	***Description***
1	invalid (contradictory) combination
2	combination possible, but not beneficial
3	neutral combination (neither beneficial nor contradictory)
4	beneficial combination (supporting each other)
5	necessary combination (to fulfill core functions)

Figure 6 gives a rough idea of the consistency matrix of physical parts, depicting the manual definition of one possible solution by arrows within the matrix. The generation of the complete solution space by manually depicting all valid solutions is time consuming and not particularly beneficial. The matrix can be used to support the designer to depict and compose architectures starting with one element and being influenced by the possibilities given in the matrix, similar to a morphological chart.

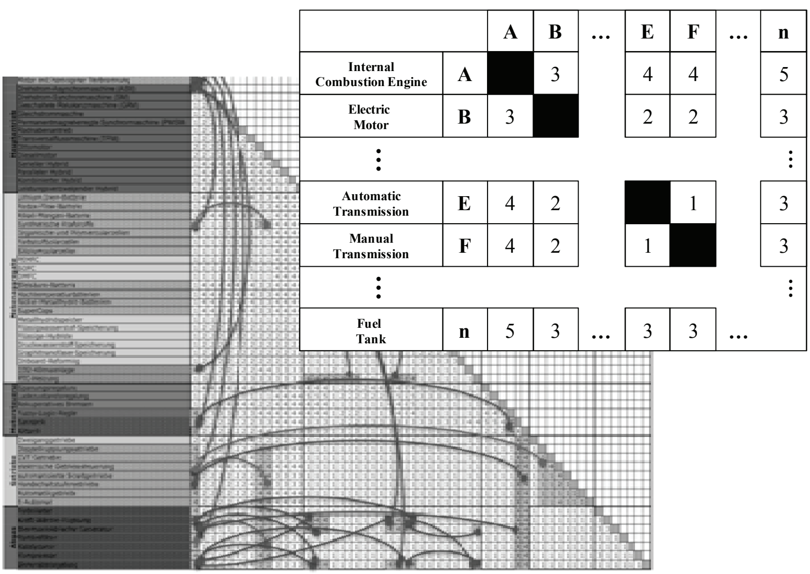

Figure 6. Definition of solution space – consistency matrix

To allow for a more complete depiction of the solution space, algorithms were defined on the level of physical parts, based on the models established so far. As a starting point, the “energy consumer” (see figure 4) was picked, and a path to the energy source chosen. Different combinations of physical elements on that path fulfill this requirement, giving a large number of possible solutions. The differentiation of interrelations of the consistency matrix enables the navigation through the solution space. The use of certain elements requires another (dependency type 5) while others are prohibited (dependency type 1) or optional (dependency type 3), for example.
By randomized repetition of the rule-set, the algorithm is able to generate valid solutions within the solution space. Eliminating the repeating solutions, the test-runs show a result of roughly 200 different architectures of the system. Given these system architectures, the following challenge is to assess the existing solutions. A graph depiction was chosen to evaluate the quality of the architectures manually and makes them accessible for DSM analysis for example.

4.4 Definition of Assessment Criteria

The assessment criteria applied at that stage are for example the number of fulfilled user functions or the number of system elements used within one solution (figure 7). Being the first iteration of the procedural model, these results can only point the direction which to follow. Further analysis is conducted by analyzing the paths of the different graph representations, pointing out for example the degree of efficiency or weight of different solutions.

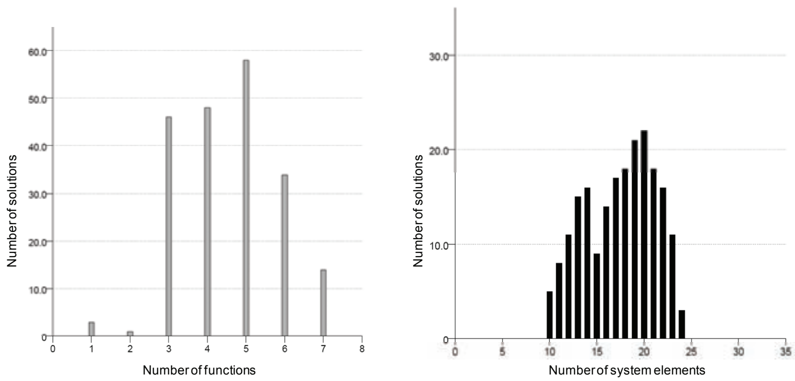

Figure 7. Assessment of solutions: fulfilled user functions and number of system elements within different architectures

4.5 Increase Level of Detail of Promising Solutions

To increase the level of detail, the necessary information, such as degree of efficiency or weight was added to the system elements, allowing for further analysis and higher level of detail of the functional models. As a result of this process, the number of solutions could be reduced, so that the following calculations and simulations to define the estimated behavior in terms of energy consumption and vehicle dynamics can be conducted.
The outcome on that level of detail requires detailed information about the scenarios (traffic, environment etc.) in which the vehicle is used, so that a much more thorough modeling and simulation effort was conducted not being part of this publication.

5 CONCLUSIONS

The presented approach is meant to support the efficient and reliable conduction of the early phases of product design, based on the discussion of the design situation and available methods. The systematic synthesis of valid solutions based on available subsystems is supported, the depiction and management of the complete solution space and the definition of assessment criteria is part of the framework. The iterative and recursive character of the framework makes a more thorough analysis possible and allows

for a gradual reduction of the solution space. The combination of different approaches allows for the achievement of these objectives, compensating the shortcomings of singular methods.
Future work includes the systematic definition of a set of assessment criteria which can be analyzed utilizing the presented methods, showing the framework's possibilities. The framework as such has to be evaluated on other products, proving the generic character of the applied methods. Furthermore, products tend to be part of a product family and thus variant management and modularization approaches need to be considered when assessing solutions. Further application will show whether the framework gives practical support and can contribute to a more efficient design process in the early phases of design.

REFERENCES

[1] Ulrich, K.T., Eppinger, S.: Product Design and Development. Irwin McGraw-Hill, 2008

[2] Browning, T.: Applying the Design Structure Matrix to System Decomposition and Integration Problems. IEEE Transactions on Engineering Management 48 (2001) 3, pp 292-306.

[3] Gorbea, C., Spielmannleitner, T., Lindemann, U., Fricke, E.: Analysis of Hybrid Vehicle Architectures using Multiple Domain Matrices. Proceedings of the 10th International DSM Conference. Stockholm, 2008.

[4] Ehrlenspiel, K.: Integrierte Produktentwicklung. München: Carl-Hanser, 2003.

[5] Wyatt, D., Wynn, D., Clarkson, J.: Synthesis of product architectures using a DSM/DMM-based approach. Proceedings of the 10th International DSM Conference. Stockholm, 2008.

[6] Lindemann, U., maurer, M., Braun, T.: Structural Complexity Management: An Approach for the Field of Product Design. Berlin: Springer, 2008.

[7] Hellenbrand, D., Lindemann, U.: Using the DSM to support the selection of product concepts. Proceedings of the 10th International DSM Conference. Stockholm, 2008.

[8] Lindemann, U.: Methodische Entwicklung technischer Produkte. Berlin: Springer, 2005.

[9] Maurer, M.: Structural Awareness in Complex Product Design. München: Dr. Hut, 2007.

[10] Pahl G., Beitz, W., Feldhusen J. and Grote K.-H.: Engineering Design – A Systematic Approach. Berlin: Springer, 2006.

[11] Ponn, J., Lindemann, U.: Konzeptentwicklung und Gestaltung technischer Produkte. Berlin: Springer, 2007.

[12] Yasushi, U., Tetsuo, T.: FBS Modeling: Modeling Scheme of Function for Conceptual Design. In: Proceedings of the 9th International Workshop on Qualitative Reasoning, 1995.

[13] Albers, A., Alink, T., Matthiesen, S., Thau, S.: Support of system analysis and improvement in industrial design through the contact & channel model. In: Proceedings of the 10th International Design Conference, DESIGN 2008. Dubrovnik, 2008.

[14] Suh, N. P.: Axiomatic Design. Oxford University Press, 2001.

[15] Andreasen, M. M., Duffy, A., Mortensen, N. H.: Relation Types in Machine Systems. In: Proceedings of the WDK Workshop on Product Structuring, Delft, The Netherlands, 1995.

Contact:
Frank Deubzer
Technische Universität München
Institute of Product Development
Boltzmannstr. 15
85748, Garching b. München
Germany

Phone: +49 289 151 55
Fax: +49 289 151 44
E-mail Address: frank.deubzer@pe.mw.tum.de
URL: www.pe.mw.tum.de

Frank Deubzer studied mechanical engineering in Munich, Germany and England. He is researcher at the Institute of Product Development, Prof. Lindemann, at the Technische Universität München. His research focuses on Complexity Management, Product Architecture Synthesis and Evaluation and Variant Management.

Udo Lindemann is a full professor at the Technische Universität München, Germany, and has been the head of the Institute of Product Development since 1995, having published several books and papers on engineering design. He is committed in multiple institutions, among others as President of the Design Society and as an active member of the German Academy of Science and Engineering.