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

CONCEPTUAL DESIGN OF MODULARIZED ADVANCED MECHATRONIC SYSTEMS

Jürgen Gausemeier[1], Daniel Steffen[2], Jörg Donoth[1] and Sascha Kahl[1]
(1) Product Engineering, Heinz Nixdorf Institute, University of Paderborn, Germany
(2) UNITY AG, Germany

ABSTRACT

Future mechatronic systems will have inherent partial intelligence. We call these systems "self-optimizing" systems. Their functionality leads to increased complexity of their development. A method to handle complexity is product structuring.

This contribution presents a holistic approach for the domain-spanning conceptual design of mechatronic and self-optimizing systems, especially taking into account product structuring. The aim is the identification of a development-oriented product structure that includes modules, which can be developed in parallel. The approach comprises of a domain-spanning specification technique for the description of the principle solution of mechatronic and self-optimizing systems, a detailed procedure model for their conceptual design and a methodology for structuring such systems. Based on the analysis of the development task, an adequate product structure type is chosen and design rules for its realization are assigned. For the application of the design rules well-known methods like DSM, MIM and their derivatives "Reconfiguration Structure Matrix" and "Aggregation-DSM" are used. The approach is demonstrated by the example of an autonomous railway vehicle.

Keywords: product structuring, product development, domain-spanning conceptual design, principle solution, self-optimization

1 INTRODUCTION

The products of mechanical engineering and related industrial sectors, such as the automobile industry, are often based on the close interaction of mechanics, electronics and software engineering, which is aptly expressed by the term mechatronics. The conceivable development of communication and information technology opens up more and more fascinating perspectives, which move far beyond current standards of mechatronics: mechatronic systems having an inherent partial intelligence. We call these systems "self-optimizing" systems. Self-optimization enables advanced mechatronic systems that have the ability to react autonomously and flexibly on changing operation conditions. The functionality of self-optimizing systems leads to increased complexity of their development and requires an effective cooperation and communication of the developers from different domains during the whole development process. The established design methodologies, i.e. the VDI Guideline 2206 [1], lay the foundation to meet these challenges. To handle complexity these methodologies need to be fundamentally extended and added by domain-spanning methods and tools. This especially applies to the early development phase "conceptual design".

One method to handle complexity is product structuring [2]. The aim is the identification of modules that form logical and functional units, which can be separately developed, tested, maintained and, if necessary, exchanged. Thus the product structure effects the whole product lifecycle. Product structuring is carried out in the early design phase, when the overall construction and behavior of the system are defined. Structuring strategies can be differentiated into shape- and function-oriented strategies. For example, the so called packaging, found in the automobile industry, induces a shape-oriented product structure. An example is a car's "front module" which includes different functions like the radiator, headlights, bumper, etc. In a function-oriented module for the realization of one superior function, i.e. a "driving dynamics control", the distributed elements control unit, sensors, breaks, etc. are combined. Both strategies are not sufficent on their own, but need to be combined. Depending on the product and the general conditions of the development, one of the strategies leads, but is not the only one.

It is clear that for an efficient product structuring, different aspects need to be taken into account and a fundamental understanding of the whole system by all developers' right from the beginning of the development process is essential. This requires an efficient collaboration of the developers especially in the early design phases. Thus a holistic description of the whole system is necessary that regards all engineering domains in an equitable way and supports a methodology, which enables the developers to generate an adequate product structure. For both the principle solution, as the result of the conceptual design phase, as well as the basis for the further "concretization" of the system, represents a significant milestone [3].

This contribution presents a holistic approach for the conceptual design of mechatronic and self-optimizing systems, especially taking into account product structuring. First, the paradigm of self-optimization is explained. Second, the general procedure for the development of mechatronic and self-optimizing systems is explained. The following subchapters show a specification technique for the domain-spanning description of the principle solution [3], a detailed procedure model for the conceptual design [3] and a new developed methodology for the structuring of advanced mechatronic systems [4]. All three elements are aligned to each other and well-matched. Afterwards the whole approach is demonstrated by the example of an autonomous intelligent railway vehicle. The contribution closes with a summary of the essential results.

2 THE PARADIGM OF SELF-OPTIMIZATION

The aim of mechatronics is to optimize the behavior of a technical system. The conceivable development of communication and information technology will enable mechatronic systems with inherent partial intelligence. We call these systems "self-optimizing systems". At this self-optimization connotes the endogenous adaptation of a system's optimization objectives on changing operating conditions and the consequent adaptation of the parameters and, if necessary, of the structure and thereby of the behavior of a technical system [5]. Regarding this, the self-optimization process goes far beyond conventional control and adaptation strategies. Self-optimization enables systems that have inherent "intelligence". They have the ability of acting and also reacting autonomously and flexibly on changing operating conditions. The realization of self-optimizing systems is the aim of the Collaborative Research Center (CRC) 614 "Self-optimizing Systems and Structures in Mechanical Engineering".

The key aspects and the mode of operation of a self-optimizing system are depicted in Figure 1. Using the influences as a basis, the self-optimizing system determines the internal objectives that have to be pursued actively. These internal objectives are based on external ones, whereas those are set from the outside, e.g. by the user or other systems, and also on inherent objectives that reflect the design purpose of the system. Inherent objectives of a driving module can be for example: saving of the driving functions and a high efficiency. If we below talk about objectives, we refer to the internal ones, because those are part of the optimization. Low energy demand, high travelling comfort and low noise emission are examples of internal objectives. The adaptation of objectives means, for instance, that the relative weighting of the objectives is modified, new objectives are added or existing objectives are discarded and no longer pursued. The adaptation of the objectives leads to an adaptation of the system's behavior. Altogether self-optimization takes place as a process that consists of the three following actions, called the **Self-Optimization Process**:

1. **Analyzing the current situation:** The current situation includes the current state of the system including all observations of the environment that have been made. Observations can also be made indirectly by communication with other systems. Furthermore, a system's state contains possible previous observations that have been recorded. One basic aspect of this first step is the analysis of the fulfillment of the objectives.
2. **Determining the system's objectives:** The system's objectives can be generated by choice, adjustment and generation. By choice we understand the selection of one alternative out of predetermined, discrete, finite quantity of possible objectives; whereas the adjustment of objectives means the gradual modification of existing objectives, respectively of their relative weighting. We talk about generation, if new objectives are being created that are independent from the existing ones.
3. **Adapting the system's behavior:** The changed system of objectives demands an adaptation of the behavior of the system. As mentioned above, this can be realized by adapting the parameters and, if required, by adapting the structure of the system. This action finally closes the loop of the

self-optimization by adapting the system’s behavior.

The self-optimizing process leads, according to changing influences, to a new state. Thus a state transition takes place.

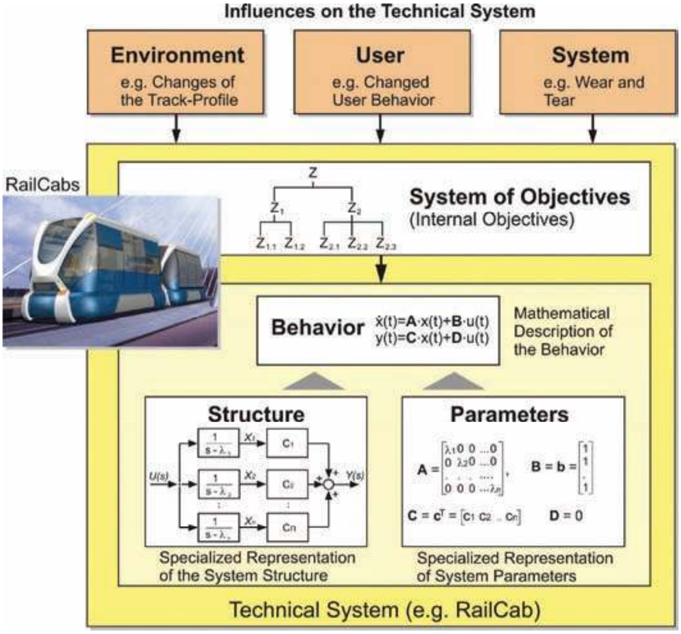

Figure 1. Aspects of a self-optimizing system – influences on the system result in an adaptation of the objectives and an according adaptation of the system’s behavior

3 DEVELOPMENT OF ADVANCED MECHATRONIC SYSTEMS

The development of mechatronic and self-optimizing systems is still a challenge. The established design methodologies, i.e. the engineering design by PAHL/ BEITZ [6] or the VDI Guideline 2206 [1], lay the foundation to meet these challenge. Nevertheless these methodologies need to be fundamentally extended and added by domain-spanning methods and tools to handle the complexity of the development. This especially applies to the early development phase “conceptual design”.

On the highest degree of abstraction, the development process of mechatronic and self-optimizing systems can be subdivided into the domain-spanning conceptual design and the domain-specific “concretization” (Figure 2). Within the conceptual design, the basic structure and the operation mode of the system are defined. Thus the conceptual design has to include the decomposition of the system into modules. This decomposition has to result in a development-oriented product structure, which integrates the two basic and mostly contradictious views of shape- and function-oriented structure. All results of the conceptual design are specified in the so-called “principle solution”. How to specify the principle solution has not been fixed for the field of mechatronics and self-optimizing systems by now. Within the CRC 614, a set of specification techniques in order to describe the principle solution of advanced mechatronic systems has been developed. By using this specification technique, the system that is to be developed will be described in a holistic, domain-spanning way. The description of the principle solution provides all relevant information for the structuring of the system and forms the basis for the communication and cooperation of the developers from different domains. Based upon the principle solution the subsequent domain-specific “concretization” is planned and realized. The term “concretization” describes the domain-specific design of a technical system, based on the principle solution. The aim of the concretization is the complete description of the system by using the construction structure and the component structure. In so doing, all defined modules are developed in parallel, and each module is developed in parallel in the participating domains (Figure 2).

In the following, the specification technique for the description of the principle solution of advanced mechatronic systems is introduced. Then the procedure model of the conceptual design of such systems, with the product structuring being the operative word, is presented. At the end of this chapter the methodology for the structuring is described.

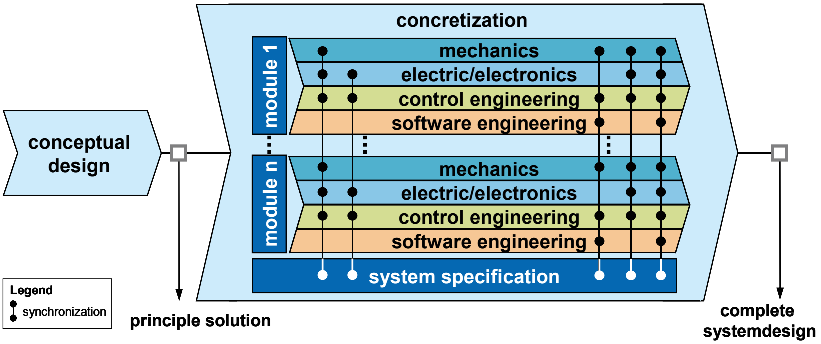

Figure 2. Basic structure of the development process

3.1 Domain-Spanning Specification of the Principle Solution

Already at the beginning of the work for the development of a holistic specification technique for the description of the principle solution of mechatronic and self-optimizing systems, it became apparent that such a description needs to be divided into aspects. The specification technique, developed within the CRC 614, is based on the research of FRANK, GAUSEMEIER and KALLMEYER [3]. The following aspects need to be taken into account: requirements, environment, application scenarios, functions, active structure, system of objectives, shape and behavior (Figure 3). The mentioned aspects are captured in the principle solution and described by partial models. The partial models are intertwined and form a coherent system. The relations are modelled between the constructs of the relating partial models and amount to a coherent system. In the following, the individual partial models are described briefly. [3]

Requirements: This partial model describes the requirements. They are represented by a list of requirements. This list forms a structured accumulation of all requirements (e.g. size, performance data, maximum costs, number of variants, etc.) of the product that is to be developed. These requirements apply as “levelling staff“ during the entire product development, for which it must be sufficient.

Environment: This model describes the environment of the system and its embedding into the environment. Relevant spheres of influence (e.g. weather, mechanical loads, superior systems) and influences (e.g. radiant heat, wind force, information) are recognized. Furthermore, the interdependencies between the influences are examined. A consistent amount of co-existing influences is regarded as a situation in which the system has to operate successfully.

Application scenarios: Application scenarios are first refinements of the system. They specify the behavior of the system in a certain. Application scenarios characterize the problem which can be solved for certain cases and also describe the possible solution approximately.

Functions: This concerns a hierarchical classification of functionality. A function is the general and intended relationship between input and output values with the objective of performing a task. A subdivision into sub-functions is to be executed as long as reasonable solution patterns are not found.

Active structure: This aspect describes the system elements, as well as their attributes and the relations of the system elements to each other. The objective is the illustration of the fundamental construction of the system including all system configurations. In this manner the values, which can be detected, gets specified and to which influences and incidents the system can react with a behavior adaptation.

System of objectives: This is the representation of the external, inherent and internal objectives and their relations. The objectives are represented hierarchically as a tree. The hierarchical relations are specified by logical relations with declaration of the hierarchical criterion *“is sub-objective of...”*. Graphs are used for the modelling of objectives if the influence of the objectives among themselves has to be expressed, i.e. whether the objectives support each other, mutually exclude each other or whether they are neutrally to each other.

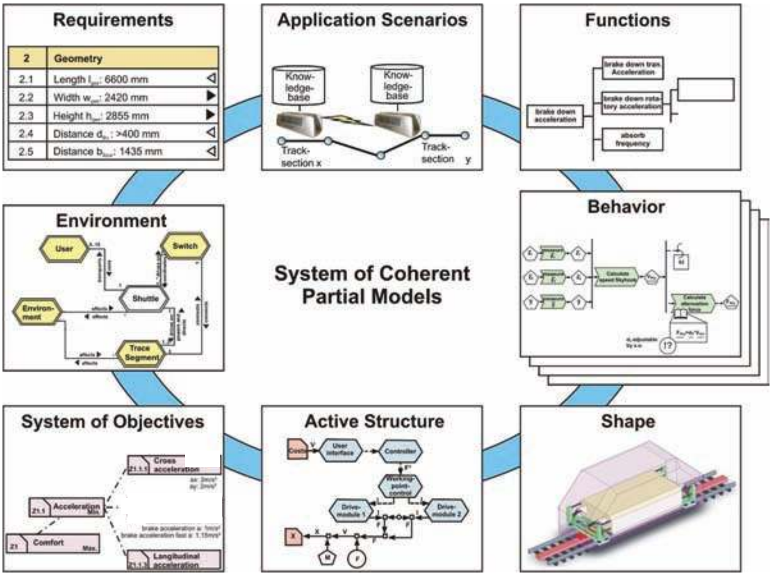

Figure 3. Partial models for the domain-spanning description of the principle solution of self-optimizing mechatronic systems

Shape: For first definitions of the system-shape, this aspect has to be modelled during the conceptual design. It concerns, in particular, working surfaces, working spaces, envelope surfaces and supporting structures. The final results are a rough construction structure and shape model.

Behavior: The aspect behavior consists of a whole group because there are different kinds of behavior, e.g. the logic behavior, the dynamic behavior of multi-body systems, the cooperative behavior of system components etc. Essentially, the system-states have to be modelled with the associated operation processes and the state transitions with the underlying adaptation processes. The adaptation processes represent the appropiate realisation of the self-optimization process.

- The **partial model behavior – states** illustrates the states and the state transitions of a system. All intended and to be considered system states and state transitions have to be described as well as the state transition releasing incidents.
- The **partial model behavior – activities** describes the operation sequences, which take place in a system state, as well as the adaptation processes, which are typical elements of self-optimization. The processes are essentially modelled with activities.
- The **partial model behavior – sequence** represents the interaction between several system elements. The activities executed during the interaction of the system elements and the information exchanged between them is modelled in chronological order.

3.2 Procedure Model for the Conceptual Design

As already mentioned, within the conceptual design phase the basic construction and the operation mode of the system are defined as well as the system is decomposed into modules. The basic procedure in the conceptual design phase is divided into four sub-phases, which are now explained in detail. [3]

Planning and clarifying the task

This sub-phase identifies the design task and the resulting requirements on the system is worked out in here. At first the task is analyzed in detail. At this the predefined basic conditions for the product, the product program, and the product development are taken into account. This is followed by an analysis of the operational environment which investigates the most important boundary conditions and influences on the system. The external objectives emerge next to disturbances. Beyond that, consistent combinations of influences, so-called situations, are generated. By the combination of characteristic situations with a first discretion of the system's behavior, application scenarios occur. With this information in hand, it is possible to identify an adequate product structure type for the system and design rules, which guide the developers to realize this product structure type. The design rules can be of implicit or explicit nature. The methodology for product structuring is described in detail in the following subchapter 3.3. The results of this sub-phase are the list of requirements, the environment model, the aspired product structure type and the assigned design rules as well as the application scenarios.

Conceptual design on the system's level

Based on previously determined requirements of the system, solution variants are developed for each application scenario. The main functions are derived from the requirements and set into a function hierarchy. The function hierarchy needs to be modified according to the specific application scenarios, e.g. irrelevant functions are removed and specific sub-functions are added. Then there is a search for solution patterns in order to realize the documented functions of the function hierarchy, which will be inserted into a morphologic box. In many times, there are already existing, well-established solutions which we call solution elements. If there are such solution elements, they will be chosen instead of the abstract solution patterns.

The consistent bunches of solution patterns form the basis for the development of the active structure. In this step, the refinement of the solution patterns to system elements takes place as well. Based on the active structure, an initial construction structure can be developed because there are primal details on the shape within the system elements. In addition, the system's behavior is roughly modeled in this step. Basically, this concerns the activities, states and state transitions of the system as well as the communication and cooperation with other systems and subsystems. The analysis of the system's behavior produces an imagination of the optimizing processes, running within the system. The external, inherent and internal objectives can be defined. During the described activities the developers apply the implicit design rules.

The solutions for the application scenarios need to be combined. It is important that workable configurations are created which make a reconfiguration of the system possible. Keeping this information in mind, it is identified if there is a containing potential of self-optimization at all. There is a potential for self-optimization if the changing influences on the system require modifications of the pursued objectives and the system needs to adjust its behavior. If there is potential for self-optimization, the function hierarchy needs to be complemented by self-optimizing functions. In particular solution patterns of self-optimization are applied to enable self-optimizing behavior [3]. The resulting changes and extensions of system structure and system behavior need to be included appropriately.

The best solution for each application scenario is chosen and these solutions are consolidated to a principle solution on the system's level. Afterwards, an analysis takes place which looks for contradictions within the principle solution of the system and which contradictions might be solved by self-optimization. Self-optimizing concepts for such contradictions are defined, which contain the three basic steps of self-optimization. The principle solution of a self-optimizing system on the system's level is the result of this phase.

Conceptual design on the module's level

The principle solution on the system's level describes the whole system. It is necessary to have a closer look at the solution, in order to give a statement on the technical and economical realization of the principle solution. For that purpose, the system is decomposed into modules and a principle solution for each single module is developed. The division is based on aspired product structure type and the application of the explicit design rules. Extreme views on the system are generated and weighted against one another (see also sub-chapter 3.3). The development of a principle solution for each single module corresponds to the "conceptual design on the system's level", starting out with "planning and clarifying the task". This phase results in principle solutions on the module's level.

Integration of the concept

The module's principle solutions will be integrated into a detailed principle solution of the whole system. Again there is an analysis in order to find contradictions within the principle solutions of the modules and it is checked if these contradictions can be solved by self-optimization. Concluding, a technical-economical evaluation of the solution takes place. The result of this phase is a principle solution of the whole system that serves as a starting point for the subsequent concretization.

3.3 Structuring of Advanced Mechatronic Systems

By extracting the activities for product structuring from the procedure model of the conceptual design, as presented in the previous sub-chapter 3.2, the methodology for the structuring of advanced mechatronic systems comprises of five essential steps. Below the specifics and fundamentals of these steps are explained in detail. [4]

Analysis of the development task

The first step's aim is to get an idea of the aspired product structure. Basing on the current development task the predefined basic conditions are analyzed and the degrees of freedom for product structuring are investigated. For this analysis a scheme has been developed. It takes requirements of the product (system size, installation space, weight, performance data, recycling, quality, availability, expandability, reconfigurability), the product program (width of product program, planned product generations, differentiation, variance of costs), and the product development (expense of the development of the product structure, depth of development, time of delivery) into account. The characteristics of the current development task are compared to the characteristics of nine basic development tasks. Specific product structure types are assigned to these basic development tasks that have been tried and trusted. Examples of successfully realized products, basing on these structure types, are additionally assigned to the basic development tasks. They act as orientation aid or "light-house" during the further development comparable to the "ideal concept" by ALTSCHULLER [7]. Thus the focus (shape- or function-oriented) for product structuring of the current development task is chosen (Figure 4). Because of a par for par transfer of the product structure is mostly not possible, design rules have been defined and selectively assigned to the basic development tasks. These design rules guide the developers during the development process and support them to make design decisions appropriate to the aspired product structure type. By this means the whole approach can be applied to new and until now unremedied development tasks, too. This makes the approach more flexible than other approaches. All in all there are 27 design rules that can be assigned to eight categories: performance, recycling, quality, extensibility, standardization, costs, development, and production. The design rules are applied for the development of the active structure, the shape, the information processing, etc. They can be applied in an implicit and explicit way. For one thing an implicit application of design rules takes place, every time a decision according to the product structure is made. For another thing there are development steps, which deal especially with product structuring. The design rules, applied within these steps, are called explicit design rules.

Consequently the results of the first step are the basic conditions of the development task, an aspired product structure as well as an amount of assigned design rules.

Investigation of the Current Product Concept

During the conceptual design phase the different partial models of the principle solution are analysed against the backdrop of product structuring. For the structuring different methods are applied that need different information. The relevant information for the application of these methods are extracted from the partial models within this step.

Product Structuring – Creation of Extreme Views

Respective to the procedure model, presented in sub-section 3.2, there is an implicit application of the design rules during the whole conceptual design as of the conceptual design on the system's level, and an explicit one before the beginning of the conceptual design on the module's level. For the explicit application, e.g. the Design Structure Matrix (DSM) by EPPINGER ET. AL. [8] is used. It enables the analysis of the connections of the system elements. The relevant information of the system elements' connections are mainly extracted form the partial models "active structure" and "shape". The weighting of the different relation aspects (material, energy, and information flows as well as spatial interdependencies) are determined by the aspired product structure and the assigned design rules. For the structuring two extreme views on the system are created. One focused on a shape-oriented structure and one focused on a function-oriented structure. Afterwards the weighting in-between these two views is varied. As a result an application-specific compromise is developed.

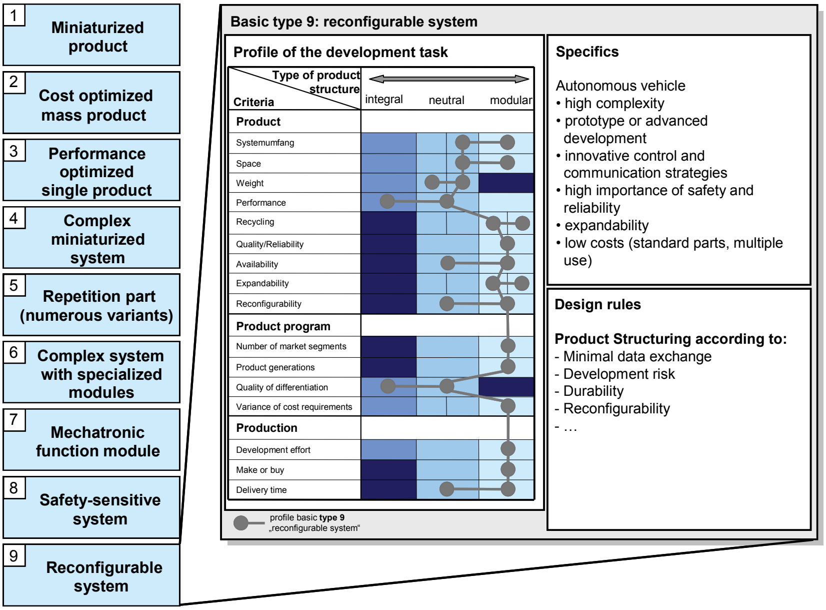

Figure 4. left: basic development tasks; right: description of the characteristics of basic development task "reconfigurable system" with assigned design rules

For self-optimizing systems especially one aspect needs to be taken into account: self-optimizing systems have the ability to reconfigure. Hence autonomous modules with disjoint functions and homogenous interfaces have to be identified. For this purpose the "Aggregation-DSM" and the "Reconfiguration Structure Matrix" (RSM) have been developed. Both build upon the DSM and use application scenarios as input. For each application scenario a separate DSM is set up. Afterwards the different DSMs are superposed in two ways. First, the aggregation of all DSM is generated. That means all connections within the system are once listed. The resulting "Aggregation-DSM" shows all possible connections within the systems and allows formulating an adequate structure. Second, the frequency of the connections is taken into account by summing up the interrelations over all application scenarios. The resulting RSM allows identifying those system elements, which are only activated in a few application scenarios. They could be integrated in independent additional modules. Those system elements, which are active in all application scenarios, are integrated in basic modules.
The result of this step forms the basis for the application of further methods for the integration of the rest oft the relation aspects in the next step.

Product Structuring – Refinement and integration of further information

For the further refinement of the product structure the Module Indication Matrix by ERIXON [9] and its extension by BLACKENFELD [10] are used. They form the third mainstay of the supporting methods of the methodology for structuring of advanced mechatonic systems (Figure 5). The MIM allows to take into account the properties of the system elements and to summarize them according to matching aspects. Input information are for example the realized functions, the used material, or maintenance intervals of a system element. The relevance ot the aspects again results from the aspired product structure and the assigned design rules. The result of this phase is the development-oriented product structure. It integrates the two basic and mostly contradictious views of a shape- and function-oriented product structure. The process of concretization is planned based on this structure. The product structure needs to mirror both aspects and their relations, because of both aspects are relevant equivalently for mechatronic and self-optimizing systems.

The resulting development-oriented product structure is described by two hierarchical trees. On one axis the shape-oriented structure is described. The other axis shows the function-oriented structure. The assignment of a function to a system element and vice versa is described by relations.

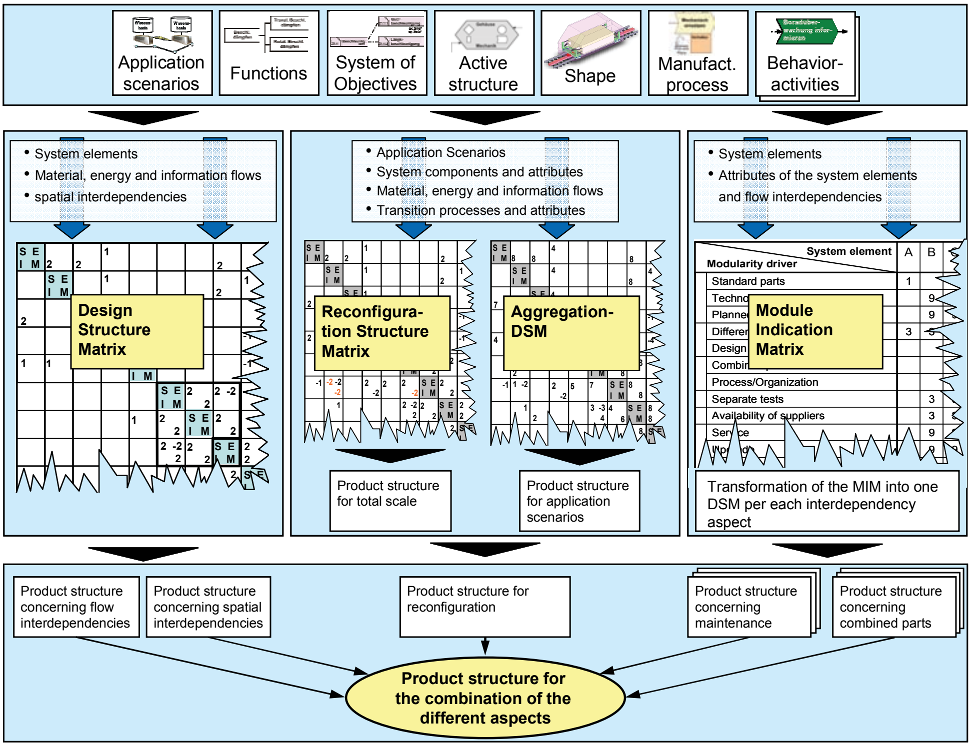

Figure 5. Interaction of the partial models with DSM, RSM, Aggregation-DSM, and MIM

Valuation

Finally the developed product structure is evaluated against the backdrop of the current development task. Revisions of the product concept are initiated that support a consequent realization of the aspired product structure, e.g. modifications of interfaces. Afterwards the parallel concretization of the modules begins. The validation of the product structure always depends on the current development task, technical criteria, economical criteria, and as the case may be available product plattforms. A general validation rule can not be defined.

4 APPLICATION EXAMPLE

The described approach has been validated by the example of one of the CRC's demonstrators. It is an innovative railway system called "Neue Bahntechnik Paderborn/RailCab" (http://www-nbp.uni-paderborn.de). The system is prototypically realized on a test track at a scale of 1:2.5. Autonomous vehicles (RailCabs) that supply transport for both passengers and cargo, establish the core of the system (Figure 6). They drive on demand and not by schedule. The RailCabs act in a pro-active way, e.g. in order to reduce the required energy by forming convoys. The actuation is realized by a contact-free dual-feed electromagnetic linear drive [5]. The stator of the linear drive is situated between the track and the rotor within the RailCab. The dual feed allows variable adjustment of the vehicle's magnetic field. Consequently, several RailCabs can be operated on the same stator section with different velocities. With an active tracking module, based on an independent axle chassis with loose wheels, the choice of direction by passing over a switch takes place vehicle-sided. An active spring technology with an additional tilt technology results in a high travelling comfort. The RailCab's basic technology is placed in the plain-built undercarriage on which the chassis for passengers or cargo will be set upon.


passenger shuttle

comfort-version

cargo shuttle

convoy formation

local traffic-version

Figure 6. Shuttles of the project „Neue Bahntechnik Paderborn/RailCab“

The development of the prototype focuses on the validation of the applied technologies and of the new developed information technological processes (self-optimization). At first the development task is analysed. Design and efficiency of the prototype are less important. In return, the drive and the active spring technology have to be accessible and modifiable in later performed test cases. For the validation of new processes in the field of information technology additional properties are important: autonomy of the included modules and system elements, learning ability, high control performance, and high safety requirements. The prototype has to be capable to be updated. In respect of a later serial production, the mechanical components have to be multiple usable and reusable. Altogether the development task has the characteristics of basic development task nine “Reconfigurable System” (compare Figure 4). Thus the main important design rules for product structuring are: function fulfilment, minimal data exchange, ability of testing and validation, durability, reconfigurability, user aspects, independence during further development, and development risk.

Some of this design rules are implicitly applied during the conceptual design phase “specification on the system’s level”. The result is a first principle solution specified with the specification technique presented in sub-chapter 3.1. At this stage the active structure of the RailCab consists of about 150 system elements. Subsequently the explicit application of the design rules takes place at the beginning of the conceptual design phase “conceptual design on the module’s level”. For this information flows (representing functional dependencies) and spatial dependencies are taken into account. Additionally the multiple usability of system elements is relevant. Two product structures are generated and co-ordinated by the usage of DSM. One for the information flows and one for the spatial dependencies. Figure 7 illustrates the results. The figure contrasts the two structures with the aid of the RailCab’s active structure. On the one hand two driving modules (front and rear) result from a spatial point of view. They consist of one drive and break module and one axle including a tracking module as well as a spring and tilt module. This modularization enables a symmetric and integral structure of the RailCab and a plain-build installation space. On the other hand an actuation module, a guidance module and an active suspension result from an information technological point of view. This structure meets the requirements of data exchange.

The initial product structure is refined by taking into account additional aspects. RSM and Aggregation-DSM are used to refine the spring and tilt module. The MIM is used to analyse the aspects reusability and extensibility. By this means the product structure is refined and optimized to the development task during the conceptual design on the modules level. The resulting development-oriented product structure of the RailCab is illustrated in Figure 8.

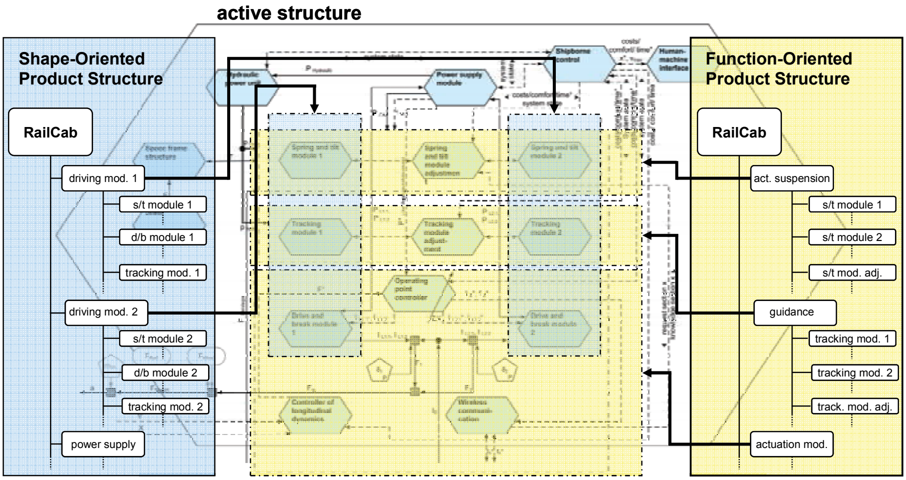

Figure 7. Comparison of initial shape- and function-oriented product structure

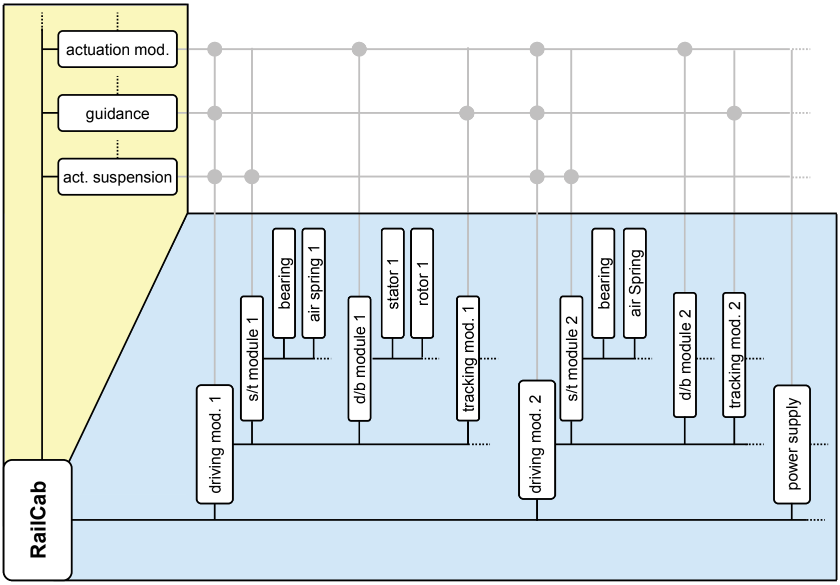

Figure 8. Product structure of the RailCab

5 CONCLUSION

Product structuring is an important step in the development process for modern mechatronic and self-optimizing systems. It helps to reduce the complexity and to increase the reliability of a system, but it also requires additional effort. A success factor is an adequate integration in the development process, by using established specification techniques, methods and tools. The presented approach shows, how this could be realized for tomorrow's mechanical engineering systems with a high amount of information technology. The approach comprises of a domain-spanning specification technique for the description of the principle solution of mechatronic and self-optimizing systems, a detailed procedure model for the conceptual design as well as a methodology for the structuring of such systems. All three elements are aligned to each other and well-matched. The additional effort for product structuring during the conceptual design is profitable, compared to the costs of typically sub-optimal interfaces and high synchronization efforts during further development.

ACKNOWLEDGEMENT

This contribution was developed and published in the course of the Collaborative Research Center 614 “Self-Optimizing Concepts and Structures in Mechanical Engineering” funded by the German Research Foundation (DFG) under grant number SFB 614.

REFERENCES

[1] Verein Deutscher Ingenieure (VDI). *VDI-Guideline 2206 – Design Methodology for Mechatronic Systems*, 2004 (Beuth Verlag, Berlin).
[2] Lindemann U. and Maurer M. Entwicklung und Strukturplanung individualisierter Produkte. In Lindemann U., Reichwald R. and Zäh M. (Ed.) *Individualisierte Produkte – Komplexität beherrschen in Entwicklung und Produktion*, 2006 (Springer Verlag, Berlin).
[3] Gausemeier J., Frank U., Donoth J. and Kahl S. Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme des Maschinenbaus – Teil 1/2. *Konstruktion*, Vol. 7/8 and 9, July/August and September 2008, pp. 59-66/ pp. 91-108 (Springer-VDI-Verlag, Düsseldorf).
[4] Steffen D. *Ein Verfahren zur Produktstrukturierung für fortgeschrittene mechatronische Systeme*. Dissertation, Faculty of Mechanical Engineering, University of Paderborn, 2006 (HNI-Verlagsschriftenreihe, Vol. 207, Paderborn).
[5] Adelt P., Donoth J., Gausemeier J., Geisler J., Henkler S., Kahl S., Klöpper B., Krupp A., Münch E., Oberthür S., Paiz C., Podlogar H., Porrmann M., Radkowski R., Romaus C., Schmidt A., Schulz B., Tscheuschner T., Vöcking H., Witkowski U., Witting K. and Znamenshchykov O. *Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte*. Gausemeier J., Rammig F. J. and Schäfer W. (Ed.), 2008 (HNI-Verlagsschriftenreihe, Vol. 234, Paderborn).
[6] Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H.: *Engineering Design – A Systematic Approach*. ed. 3, 2007 (Springer Verlag, London).
[7] Gimpel B., Herb R. and Herb T. *Ideen finden, Produkte entwickeln mit TRIZ. 2000* (Hanser Verlag, München).
[8] Eppinger S., Whitney D., Smith R. and Gebala D. A Model-Based Method for Organizing Tasks in Product Development. *Reserarch in Engineering Design*, Vol. 6, No. 1, Marc, 1994 (Springer Verlag, London).
[9] Erixon, G. *Modular Function Deployment – A Method for Product Modularization*. Dissertation, Royal Institute of Technology, 1998 (KTH, Stockholm).
[10] Blackenfeld M. *On the Development of Modular Mechatronic Products*. Licentiate Thesis, Royal Institute of Technology, 1999 (KTH, Stockholm).

Contact: Prof. Dr.-Ing. Jürgen Gausemeier
Heinz Nixdorf Institute, University of Paderborn
Fürstenallee 11
33102 Paderborn
Germany
Tel: Int +49 5251 60-6267
Fax: Int +49 5251 60-6268
Email: Juergen.Gausemeier(at)hni.uni-paderborn.de