INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006
Dubrovnik - Croatia, May 15 - 18, 2006.

DESIGN 2006

SPECIFYING THE PRINCIPLE SOLUTION OF TOMORROWS MECHANICAL ENGINEERING PRODUCTS

J. Gausemeier, U. Frank and D. Steffen

Keywords: mechatronics, self-optimization, design methodology, specification techniques

1. From Mechatronics to Self-Optimization

Nowadays, most mechanical engineering products already rely on the close interaction of mechanics, electronics, control engineering and software engineering which is aptly expressed by the term mechatronics. The ambition of mechatronics is to optimize the behavior of a technical system. Sensors collect information about the environment and the system itself. The system utilizes this information to derive optimal reactions. Future mechanical engineering systems will consist of configurations of system elements with inherent partial intelligence. The behavior of the overall system is characterized by the communication and cooperation between these intelligent system elements. From the point of view of information technology we consider these distributed systems to be cooperative agents. This opens up fascinating possibilities for designing tomorrow's mechanical engineering products. The term self-optimization characterizes this perspective [Frank et al., 2004].
Although there are numerous examples for the use and benefit of mechatronics (representative of the many available publications we refer you to [Isermann et al., 2002] and [VDI Guideline 2206, 2004]), the potential benefits of self-optimization as a feature of mechanical engineering systems are only now beginning to be recognized. It is clear that we need imagination to define machines that possess inherent partial intelligence. An additional challenge is the particular characteristic of self-optimizing systems, namely that in the design stage we can no longer anticipate all the system's possible constellations and behaviors because self-optimizing systems also exhibit cognitive abilities and are able to learn.
The intelligent mechanical engineering systems of tomorrow that we are considering are founded on mechatronics. We have therefore taken the hierarchical structuring of complex mechatronic systems suggested by Joachim Lückel and extended it to include the aspect of self-optimization [Lückel et al., 2001]. The basis of this is provided by what are called "mechatronic function modules" (MFMs), consisting of a basic mechanical structure, sensors, actuators and a local information processor containing the controller. A combination of MFMs, coupled by information technology and/or mechanical elements, constitute an autonomous mechatronic system (AMS). Such systems also possess a controller, which deals with higher-level tasks such as monitoring, fault diagnostics and maintenance decisions as well as generating parameters for the local information processing systems of the individual MFMs. Similarly, a number of AMSs constitute what is called a networked mechatronic system (NMS), simply by coupling the associated AMSs via information processing. The controller of a NMS carries out higher-level functions in the same way as that of the AMS. In the context of vehicle technology, a spring and tilt module would be an MFM, the shuttle would be an AMS, and a convoy would be a NMS. On each level the controller is enhanced by the functionality of self-optimization. Thus the previously mentioned system elements (MFM, AMS, NMS) receive an

inherent partial intelligence. The key aspects and the mode of operation of a self-optimizing system are illustrated in Figure 1. The self-optimizing system detects factors that influence the system. The factors may originate in its surroundings (environment, users, etc.) or from the system itself. The self-optimizing system determines its currently active objectives on the basis of the encountered influences. Objectives formulate the behavior that is required of the system, desired, or to be avoided [Frank et al., 2004].

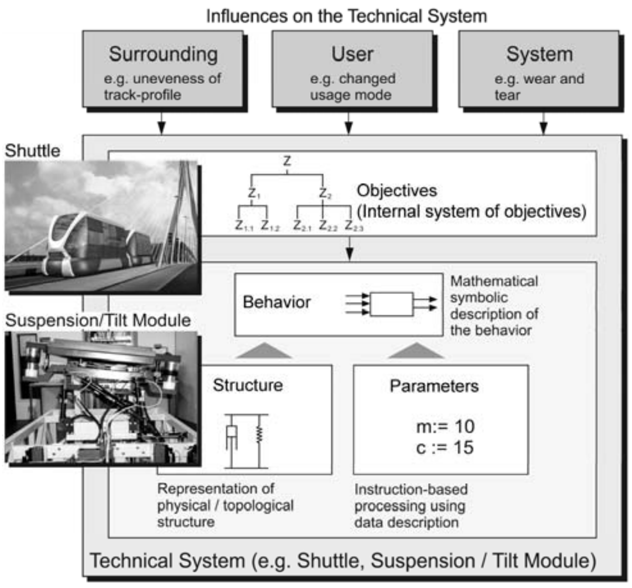
Influences on the Technical System
Surrounding
e.g. uneveness of track-profile
User
e.g. changed usage mode
System
e.g. wear and tear
Shuttle
Z
Z_1
Z_2
Objectives (Internal system of objectives)
$Z_{1.1}$ $Z_{1.2}$ $Z_{2.1}$ $Z_{2.2}$ $Z_{2.3}$
Behavior
Mathematical symbolic description of the behavior
Suspension/Tilt Module
Structure
Parameters
m:= 10
c := 15
Representation of physical / topological structure
Instruction-based processing using data description
Technical System (e.g. Shuttle, Suspension / Tilt Module)

Figure 1. Aspects of self-optimizing systems

The self-optimizing system is able to adapt the system of objectives autonomously. This 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. Adapting the objectives in this way leads to adaptation of the system behavior. That is achieved by adapting the parameters and where necessary the structure of the system. The term parameter adaptation means adapting a system parameter, for instance changing a control parameter. Structure adaptations affect the arrangement of the system elements and their relationships. Here we distinguish between reconfiguration, which changes the relationships between a fixed set of available elements, and compositional adaptation, in which new elements are integrated into the existing structure or existing elements are removed from it.

We express self-optimization as a series of three actions that are generally carried out repeatedly. This sequence of actions is designated a self-optimization process:

1. **Analysis of the current situation**: Here the current situation includes the state of the system itself and all the observations that have been made about its environment. Such observations may also be made indirectly by communicating with other systems. The current state of the system also includes any records of previously made observations. One essential aspect of this analysis is examining the degree to which the pursued objectives have been fulfilled.
2. **Determination of the system objectives:** The current system objectives may be determined by selection, adaptation or generation. A selection is here understood as choosing one alternative from a fixed discrete finite set of possible objectives, while the adaptation of objectives describes the gradual modification of existing objectives. We speak about generating objectives when new objectives are created independently of the existing ones.
3. **Adaptation of the system behavior**: This is determined by the three aspects: parameters, structure and behavior. The reaction at the end of the self-optimization cycle is effected by adapting the system behavior. The individual adaptation cases may be extremely diverse

depending on which level of the mechatronic system (MFM, AMS, NMS) we are dealing with. The domain in which the adaptation takes place also plays a considerable role.

From a given initial state the self-optimization process passes, on the basis of specific influences, into a new state, i.e. the system undergoes a state transition. We refer to the influences, that trigger a state transition, as events. The self-optimization process defines the activities, that effect this state transition, and thereby describes the system's adaptive behavior.

2. A Design Methodology for Self-Optimizing Systems

A new and powerful paradigm such as self-optimization naturally calls for new development methods and tools. Apart from that, there is also the question whether the approaches and methods of mechanical engineering's design methodology need to be fundamentally extended. This question particularly applies to the initial phases "planning and clarifying the task" and "conceptual design". For these phases it has emerged that the basic structure of the design methodology of mechanical engineering (formulating requirements, defining functions, searching for active principles to fulfill those functions, and so on) also applies to mechatronics and self-optimization, but if we look into this more deeply it becomes clear that this design methodology does need to be expanded. New aspects are, for example, the integrative use of solution patterns and the need to model the environment, application scenarios and the complex system of objectives of a self-optimizing system [Gausemeier et al., 2005].

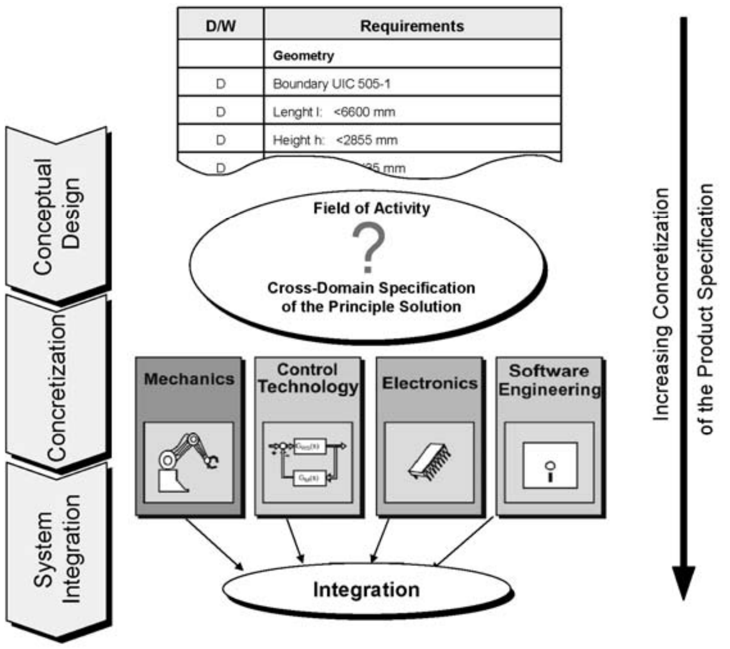

Figure 2. Field of action: Cross-domain specification of the principle solution

As in classic mechanical engineering the principal decisions in the development process of self-optimizing systems are taken during the early phases. The result of these phases is the principle solution. The principle solution is one of the essential milestones. It is developed in a cross-domain and integrative way. Afterwards, the development process is subdivided according to the modules of the developing systems as well as to the domains involved, such as mechanics, electronics, control (self-optimization) and software engineering.

A highly significant aspect is the integrative specification of the principle solution, for which the development methodology makes no adequate provision (Figure 2). This point is particularly significant because a holistic description of the principle solution constitutes the basis for the communication and cooperation between the engineers from different areas of expertise who are engaged in developing a self-optimizing system. Within the principle solution, the fundamental decisions concerning the structure and operation of the system are made. Suitable specification techniques for this purpose have been developed within the SFB 614.

3. Specification Technique for the Principle Solution of Self-optimizing Systems

In order to describe the principle solution of self-optimizing systems we use a set of semi-formal specification techniques. For a complete description we need a variety of views on the self-optimizing system [Frank, 2006]. The developed set of specification techniques allows to describe these views and how they are interlinked. Each view is mapped by computer onto a partial model. As shown in figure 3, the principle solution is made up of the following seven views or partial models: requirements, environment, system of objectives, functions, active structure, shape, application scenarios and the group behavior (cf. [Suh, 1998]). This last is considered a group because there a various types of behavior (e.g. the logic of a circuit, the dynamic behavior of a multi-body system, electromagnetic compatibility).

There are also relationships between the partial models, leading to a integrative system of partial models that represents the principle solution of a self-optimizing system. Previously, in mechatronics, the focus was normally on the system's active structure, but here the system's states and state transitions are in the foreground, i.e. the self-optimization process and its effects on the active structure and the processes taking place within the system (cf. [Buur, 1990]). The partial models are briefly described below.

Requirements: Here we are considering how to represent the requirements in the computer.

Environment: This model describes the system's environment and how it is embedded in that environment. It identifies the relevant areas of influence and possible disturbance variables (e.g. external temperature, mechanical stresses, higher-level systems). We also investigate any interactions or reciprocal effects between individual influences, and the possibility of their concurrent occurrence. A consistent set of concurrent influences constitutes a "situation", in which the technical system has to function.

System of objectives: This is the representation of the external, inherent and internal objectives as already explained, and the links between them.

Application scenarios: Application scenarios offer a way of reducing the complexity of the development task; they focus on a subset of the system that is being developed together, with its environment and the development task for that subset. The application scenarios specify how the system has to behave in a given state and a given situation, or how and on the basis of what influences state transitions should occur.

Functions: Here we are concerned with a hierarchical classification of the operating functions as a way of defining the system's basic functionality. The functions dealt with here may be conventional functions, like those listed in [Pahl et al., 1996], or functions used for self-optimization.

Active structure: Here we depict the system elements that represent solution patterns, together with their characteristics and the interrelationships between the other system elements. Our objective is to map the basic structure of the self-optimizing system together with all the envisaged system configurations. In this manner it is specified which variables can be detected and therefore also on which influences or events the system basically can react with behavior adaptations.

Shape: This model contains information about the rough shape of the elements, positions and arrangements, plus the types of active surfaces and points of action of the self-optimizing system.

Behavior: Actually this stands for different types of behavior. It will always be necessary to model the system states with the associated operative processes and the state transitions with the underlying adaptive processes. The adaptive processes bring out the concrete implementation of the self-optimizing process. Depending on the design task we will need to specify furthermore different types of behavior, such as the system's kinematics, dynamics or cooperation behavior.

The partial models are created in the process phases "planning and clarifying the task" and "conceptual design" [Gausemeier et al., 2004]. They are put in a concrete form alternating, but following a certain sequence. At the end of the phase "conceptual design" the partial models are as concrete as needed for describing the principle solution.

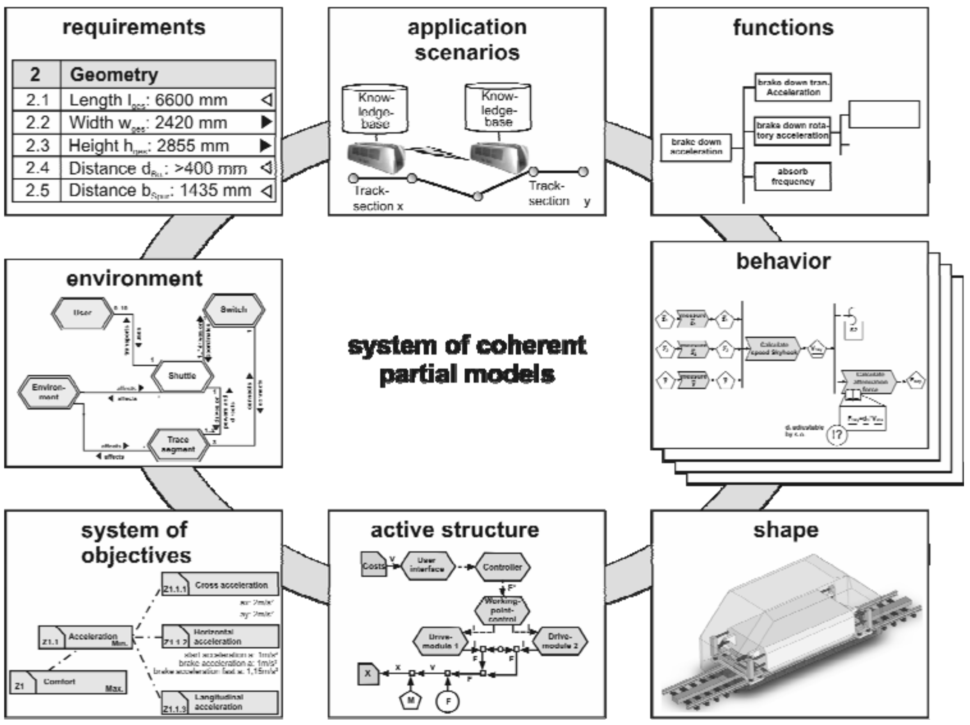

Figure 3. Integrative system of partial models for describing the principle solution of a self-optimizing system

4. Specifying the Self-Optimization Process and its Effects on the System

The core of the system's self-optimization is the self-optimization process and its effects on the system. The partial models *behavior-states* und *behaviour-activities* out of the group behavior and the *active structure* are particularly important in this context. Figure 4 shows which of the correlations between these partial models are relevant here. For the sake of clarity the partial models have been simplified and are shown at only one level of hierarchy in each case.

The states and the events that trigger a state transition are shown in the partial model *behavior-states*. The system behaves differently depending on its state and this is expressed by different behavior models, in particularly by operative processes. Each operative process is carried out by a particular system configuration. So each of the system states in the partial model *behavior-states* is correlated with one of the operative processes from the partial model *behavior-activities* that is active in that state and a system configuration from the *active structure* that is also active in that state. The example in figure 4 shows the operative processes and system configurations that belong to the states S5 and S6. In each case the operative processes and system configurations are shown as so called a logical group and are colored in the same shade of grey as the associated state. System elements and activities shown on a black field are active in all states.

A state transition is specified by the system's initial state, the event that triggers the state transition, and the system's final state. It is effected by an adaptive process, such as a self-optimization process, and the system elements that carry out this adaptive process. The initial and final states are assigned to the corresponding system configurations and operative processes. The corresponding adaptive process from the partial model *behavior–activities* and the system elements from the active structure, that perform the adaptive process, are assigned to the state transition. Figure 4 shows a state transition from S5 to S6 that is triggered by the event E7. In the initial state, S5, the system configuration shown in dark grey and the operative process shown in dark grey are both active.

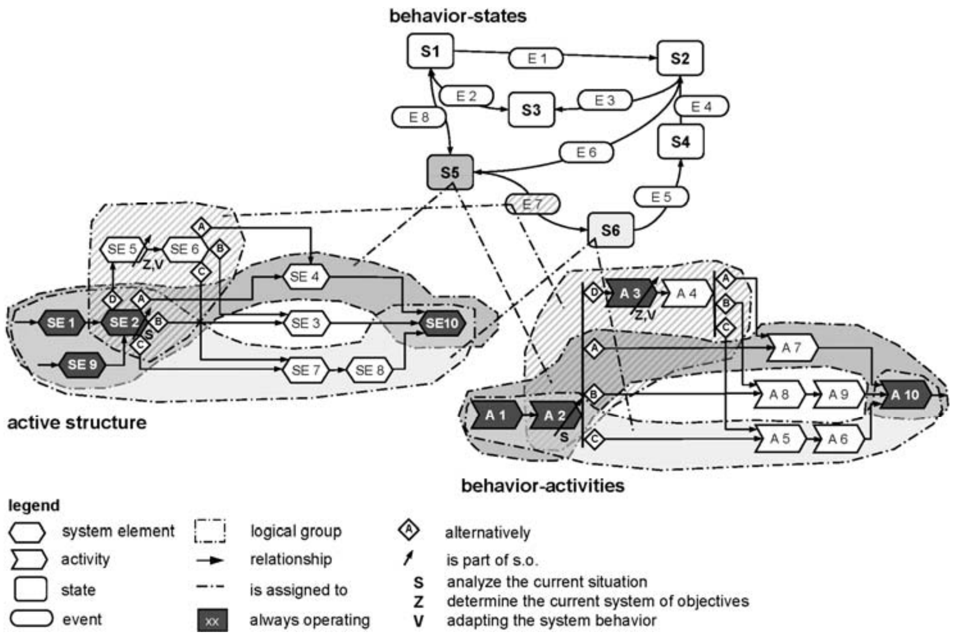

Figure 4. Core of the system's self-optimization (simplified)

When the event E7 occurs, the system generates a new system of objectives and modifies its behavior by performing a reconfiguration. The system elements responsible for the state transition and the activities of the self-optimization process that effect it are shown as a logical group (indicated by the cross-hatching) and are assigned to the event E7. As soon as the new system configuration becomes active, the system is in state S6. The system configuration that is active in S6 and the active operative process are both marked light grey.

This makes it clear that self-optimization processes effect a state transition. It also shows which events give rise to a self-optimization process, and which system elements carry out which activities to effect it. Which system objectives are active in which point of time, and which rules (e.g. configuration rules) the system applies during the self-optimization process, are specified in the currently active active structure and the corresponding operative or adaptive process.

5. Sample Application

The partial models and also the associated specification techniques have been validated on an entire railway vehicle (shuttle), which serves as a demonstrator in the SFB 614. In the following only an extract of the principle solution of the vehicle`s spring/tilt module is described, the partial models "shape" and "behavior activities". The spring/tilt module is shown in picture 5. On the left the rough shape of the elements of the spring/tilt module, their positions and arrangements are demonstrated as the result of the phase "conceptual design". On the right the realized module on the test site is shown.

Figure 6 shows an extract from the partial model "behavior-activities" namely the application-specific adaptive process for determining a suitable level of damping for the spring/tilt module. The activities, their input and output quantities, logical and temporal relationships, restrictions and interdependencies are depicted using the developed techniques for specifying adaptive and operative processes for self-optimizing systems. In this example, logical groups are used to structure the process steps corresponding to "analyze the current situation", "determine objective" and "behavior adaptation".

Only a subset of the possible behavior adaptations are shown here. Both the process steps and the logical groups are specified according to their real-time capability and how they are controlled. It is also possible to describe the methods on which they are based and, in the case of decisions, on what basis they were reached.

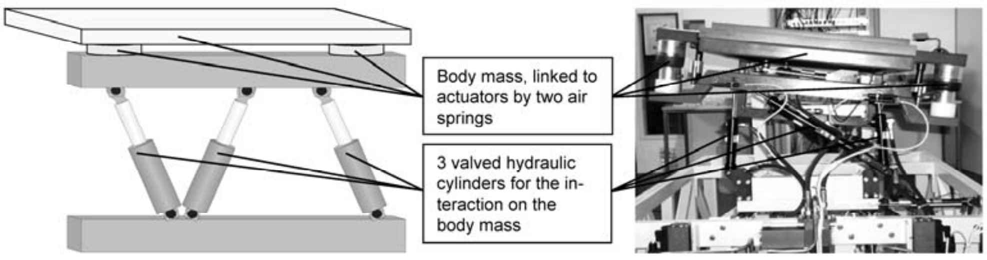
Body mass, linked to actuators by two air springs
3 valved hydraulic cylinders for the interaction on the body mass

Figure 5. Spring/tilt module, a) principle solution (partial model “shape” of the principle solution), b) the real module on the test site (MLaP, Prof. Lückel)

In this example the situation analysis determines the user’s objectives, interrogates the power requirements, analyses the level of fulfillment of the current system of objectives, and accesses previous expertise. While determining the objectives these quantities are used to specify a new system of objectives.

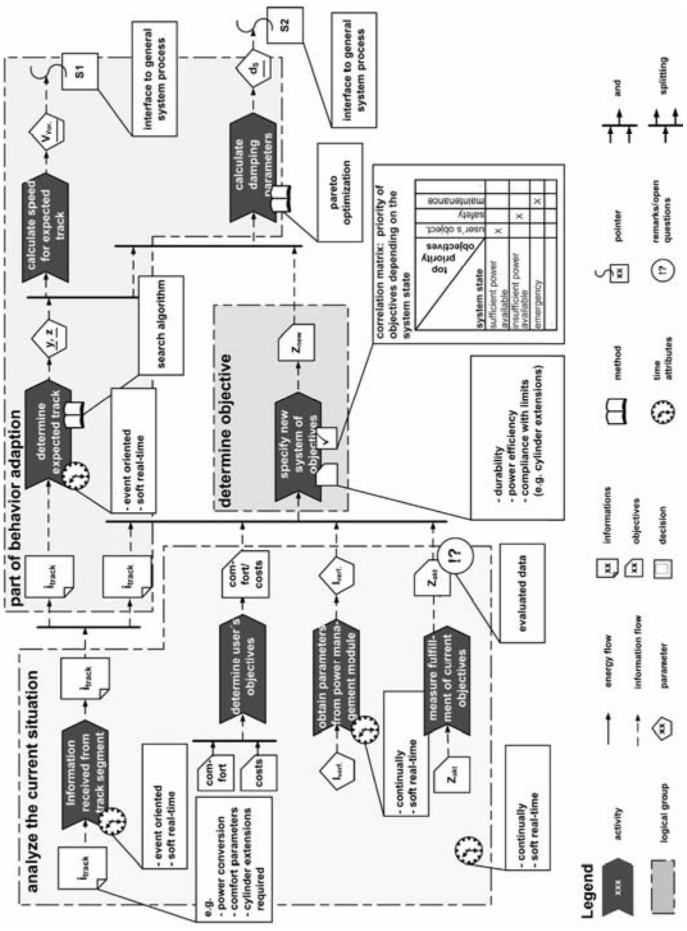
analyze the current situation
part of behavior adaption
determine objective
Information received from track segment
determine user´s objectives
obtain parameters from power management module
measure fulfillment of current objectives
determine expected track
calculate speed for expected track
specify new system of objectives
calculate damping parameters
search algorithm
pareto optimization
interface to general system process
S1
S2
event oriented
soft real-time
continually
soft real-time
com-fort
costs
com-fort/ costs
evaluated data
e.g.
power conversion
comfort parameters
cylinder extensions required
durability
power efficiency
compliance with limits (e.g. cylinder extensions)
correlation matrix: priority of objectives depending on the system state
top priority objectives
system state
user´s object.
safety
maintenance
sufficient power available
insufficient power available
emergency
Legend
activity
logical group
energy flow
information flow
parameter
informations
objectives
decision
method
time attributes
pointer
remarks/open questions
and
splitting

Figure 6. Cut-out from a self-optimization process of the spring/tilt module

The next steps calculate the expected track trajectory and the appropriate level of damping to be used as parameters for a behavior adaptation, and feed this information into the operative process of the spring/tilt module. The actual behavior adaptation then takes place there [Gausemeier et al.; 2004].

6. Summary

The active paradigm of self-optimization opens up new perspectives for developing mechanical engineering products: machines with inherent partial intelligence. The basis for this is mechatronics. Previous work on mechatronics has already shown very clearly how vital it is to specify principle solutions in a way that can be understood by experts from any domain, because the principle solution is the starting point for the parallel design and development activities in the participating domains: mechanics, electronics, control engineering and software engineering. This applies all the more to self-optimizing systems. In the early phases of developing this type of systems there are many more aspects to be modeled than in classic mechanical engineering or mechatronics. The methodology introduced in this contribution for the early development phases of a self-optimizing system has been successfully validated with a complex sample – the spring and tilt module of a rail vehicle. This also illustrates how tomorrow's machines could be developed and how adequate specification techniques could be employed.

Acknowledgments

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

References

Buur, J.: A Theoretical Approach to Mechatronics Design. Dissertation, Institute for Engineering Design, Technical University of Denmark, 1990

Frank, U.; Giese, H.; Klein, F.; Oberschelp, O.; Schmidt, A.; Schulz, B.; Vöcking, H.; Witting, K.; Gausemeier, J. (Hrsg.): Selbstoptimierende Systeme des Maschinenbaus - Definitionen und Konzepte. HNI-Verlagsschriftenreihe Band 155, Paderborn, 2004

Frank, U.: Spezifikationstechnik zur Beschreibung der Prinziplösung selbstoptimierender Systeme. Dissertation, Universität Paderborn, HNI-Verlagsschriftenreihe Band 175, Paderborn, 2006

Gausemeier, J.; Frank, U.; Schmidt, A.; Vöcking, H.: Domänenübergreifende Spezifikation der Prinziplösung selbstoptimierender Systeme. In: 2. Gemeinsames Kolloquium Konstruktionstechnik der TU Dresden, Universität Rostock, Otto-von-Guericke-Universität Magdeburg. Schloss Pillnitz, 23.-24. September 2004

Gausemeier, J.; Frank, U.; Redenius, A.; Steffen, D.: Development of Self-Optimizing Systems. In: Proceedings of Mechatronics & Robotics 2004 (MechRob 2004 (IEEE)). 13.-15. September 2004, Sascha Eysoldt Verlag, Aachen, 2004

Gausemeier, J.; Frank, U.; Steffen, D.: Intelligent Systems, Self-optimizing Concepts and Structures. In: Dachtchenko, O. (Ed.): Reconfigurable Manufacturing Systems. Berlin, Springer-Verlag, 2005

Isermann, R.; Breuer, B.; Hartnagel, H.L. (Hrsg.): Mechatronische Systeme für den Maschinenbau - Ergebnisse aus dem Sonderforschungsbereich 241 „Integrierte mechanisch elektronische Systeme für den Maschinenbau (IMES)". WILEY-VCH Verlag GmbH, Weinheim, 2002

Lückel, J.; Hestermeyer, T.; Liu-Henke, X.: Generalization of the Cascade Principle in View of a Structured Form of Mechatronic Systems, IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2001), Villa Olmo; Como, Italy

Pahl, G.; Beitz, W.: Engineering Design - A Systematic Approach, Springer Verlag, Berlin, 1996

Suh, N .P.: Axiomatic Design Theory for Systems. In: Research in Engineering Design, Springer-Verlag, London Limited, No. 10, 1998

Verein Deutscher Ingenieure (VDI): Design methodology for mechatronic systems. VDI Guideline 2206, Beuth Verlag, Berlin, 2004

Prof. Dr.-Ing. Jürgen Gausemeier
Heinz Nixdorf Institute, University of Paderborn, Fürstenallee 11
D-33102 Paderborn
Tel.: ++49-5251-606267
Fax.: ++49-5251-606268
Email: juergen.gausemeier@hni.upb.de