INTERNATIONAL DESIGN CONFERENCE - DESIGN 2008
Dubrovnik - Croatia, May 19 - 22, 2008.

PROCEEDING FOR THE CONCEPTUAL DESIGN OF SELF-OPTIMIZING MECHATRONIC SYSTEMS

J. Gausemeier, D. Zimmer, J. Donoth, S. Pook and A. Schmidt

Keywords: mechatronics, self-optimization, conceptual design

1. Introduction

The development of the information and communication technology opens fascinating perspectives for mechanical engineering: mechatronic systems with inherent partial intelligence. We use the term 'self-optimization' for these systems. We relate self-optimization of a technical system to mechanisms, which adapt endogenously the objectives of the system depending on changed environmental influences. This implies the objective-conformal autonomous adaption of the system parameters and, if necessary, the structure and, thus, the adaption of the system-behavior [Frank et al., 2004], [Gausemeier et al., 2005]. The objectives are divided up into external objectives (e.g. user defaults), inherent objectives (e.g. minimization of the abrasion) and internal objectives. The internal objectives represent thereby the active objectives followed by the system. They are created by selection and adaption of the external or inherent objectives or by the generation of new objectives. Therefore, the self-optimization process contains three phase's *analysis of the current situation, determination of the system objectives* and *adaption of the system behavior*. Thus, self-optimization substantially transcends the well-known control- and adaptation-strategies; self-optimization enables mechatronic systems to react independently and flexibly according to changing operation conditions.
The design of such systems is challenging. The already established development methodologies in the sector of classical mechanical engineering [Pahl et al., 2007] and mechatronics, for example the VDI Guideline 2206 "Design methodology for mechatronic systems" [VDI Guideline 2206, 2004], are not of any sufficiency in that context. This is particularly referring to the early phases "planning and clarifying the task" and "conceptual design". This phase results in the principle solution which specifies the basic structure and the fundamental mode of action of the system. The principle solution is described domain-spanning with the help of a set of semi-formal specification techniques. Thus, the specification of the principle solution forms the basis for communication and cooperation of specialists from the domains mechanical engineering, control engineering, digital electronics and software engineering in the course of the further concretization of the system.
This paper presents the proceeding for the conceptual design of self-optimizing systems. In a first step the specification of the principle solution is described, followed by a detailed description of the synthesis during the conceptual design. The conceptual design of a self-optimizing air gap adjustment system of a linear drive serves as an example.
This special drive is in use in the autonomous railway-shuttle of the project "Neue Bahntechnik Paderborn/Railcab". This innovative railway system which is realized in an extensive pilot plant in a reduced scale of 1:2.5. The system's core comprises autonomous shuttles for transporting passengers and goods according to individual demands rather than a fixed timetable. Therefore, they use the existing railway system. The shuttles are driven by a twice fed linear drive [Zimmer et al., 2005] (Figure 1).

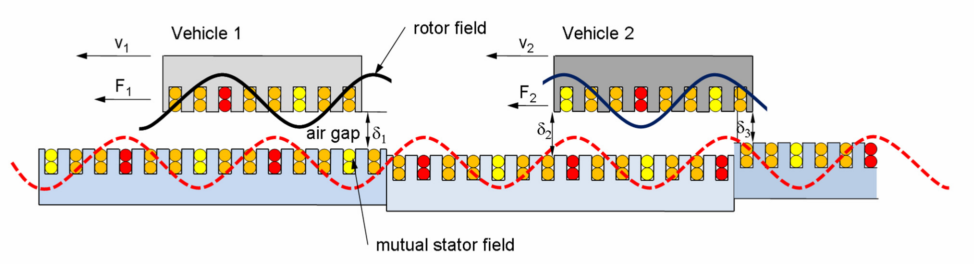


Figure 1. Linear Drive with Varying Air Gap

The stator of the linear drive is set between the rails and the rotor inside the shuttle. The three-phase windings within the stator form a magnetic field. The field undulates forward with the shuttle along the rail. The power transmission takes place across the air gap between the magnetic fields of the stator and the rotor. In addition to the function *generate thrust and brake force* the linear drive realizes the function *transfer energy to the shuttle*.

The efficiency of the linear drive depends on the air gap's size. Currently, the air gap of the shuttles is adjusted to 10 mm. This comparative wide air gap provides a huge robustness concerning small variances of the tracks compared to their ideal state, but it implicates a relative low efficiency. Further progression tends to adjust an optimal air gap depending on the state of the track. Therefore, additional actors are supposed to regulate the rotor in the shuttle during runtime so that the air gap becomes optimal.

2. Specification of the Principle Solution of Self-Optimizing Systems

In the following chapter we present the specification technique for the description of the principle solution of a self-optimizing system. It has been developed in the context of the Collaborative Research Centre (CRC) 614. This specification technique includes mechatronic systems; it is based on the work of FRANK [Frank, 2006]. Obviously, the system's complex structure requires a comprehensive description of the principle solution that is to be arranged into different aspects. These aspects are according to figure 2: requirements, environment, system of objectives, application scenarios, functions, active structure, shape and behavior. The last mentioned aspect is a group consisting of different types of behavior, e.g. combinational logic, dynamic behavior of a multi-body system, electromagnetic compatibility etc. The aspects mentioned are represented computer-internally by partial models. There are also relationships between the partial models, leading to a coherent system of partial models that represents the principle solution of a self-optimizing system.

The aspects and the appropriate partial models are to be edited in interrelation, although there already exists a certain order. They are developed in the early design phase *conceptual design*, which is described detailed in chapter 3. In the following the individual partial models are described briefly.

Environment: This model describes the environment of the system to be developed and its imbedding 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. The system's disturbing influences are marked as variable disturbances. Furthermore, the interdependencies between the influences are examined. We regard a consistent amount of co-existing influences as a situation, in which the technical system has to operate successfully. Values of influences can release a state transition of the system; we characterize such values of influences as events. Catalogues with influences and spheres of influence support the development of environmental models.

Application scenarios: Application scenarios are first concretizations of the system. They specify the kind of behavior of the system in a state and a certain situation and how state transition, due to which events, take place. Application scenarios characterize the problem which can be solved for certain cases and also describe the possible solution approximately.

Requirements: This aspect concerns the computer-internal representation of the requirements. The list of requirements establishes the basis for the representation and forms a structured accumulation of all requirements (e.g. size, performance data) to the product which is to be developed. These requirements apply as “levelling staff“ during the entire product development, for which it must be sufficient. They are differentiated into demands and wishes [Pahl et al., 2007]. Each requirement is verbally described and, if possible, clearly defined by attributes and their values. Check lists support the setting up of the requirements list; in addition we refer to [Pahl et al., 2007].

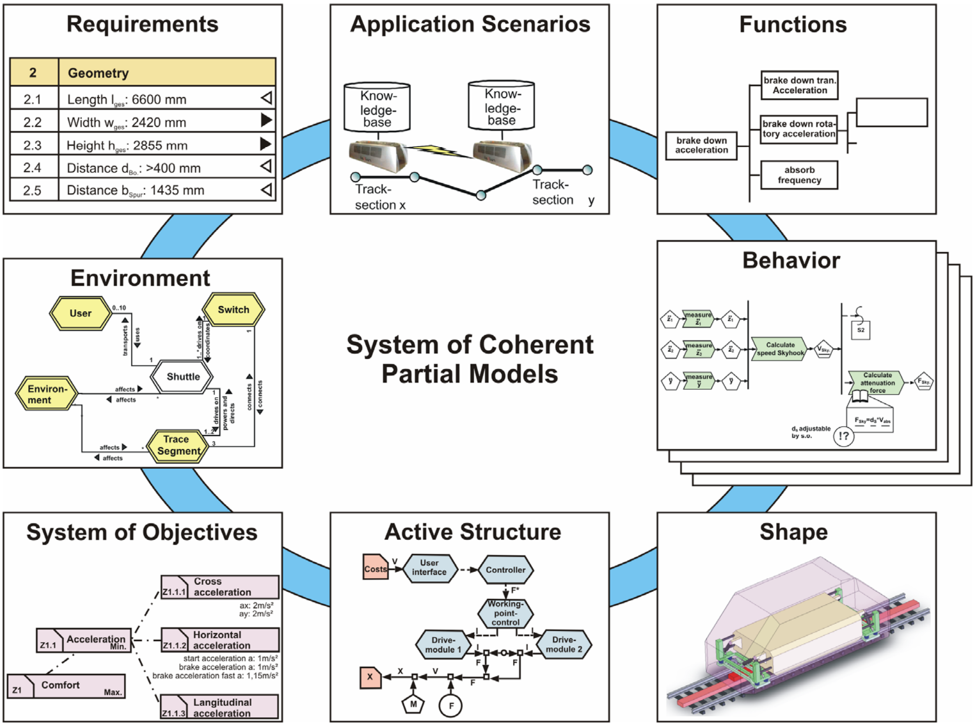


Figure 2. System of coherent partial models for describing the principle solution of a self-optimizing system

System of objectives: This concerns the representation of the external, inherent and internal objectives and their relationship. 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, prevent mutually or whether they are neutrally to each other. In place of graphs this circumstances can be expressed also by influence matrices.

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. For setting up function hierarchies a catalogue with functions exists, which is based on the work of BIRKHOFER [Birkhofer, 1980] and LANGLOTZ [Langlotz, 2000] and which is extended by functions for describing self-optimization particularly. Functions are realized by solution patterns or/ and their concretizations. A subdivision in sub-functions is to be executed as long as reasonable solution patterns will not be 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 structure of the self-optimizing system including all ahead-meant system configurations. In this manner the values, which can be detected, gets specified and, therefore, to which influences and incidents the system can react with a behavior adaption.

Shape: For first definitions of the system-shape during the conceptual design, this aspect has to be modelled. That concerns, in particular, working surfaces, working spaces, envelope surfaces and supporting structures. The computer aided modelling works out with the help of usual 3D CAD systems.

Behavior: This group of partial models contains different kinds of behavior. Essentially, the system-states have to be modelled with the associated operation processes and the state transitions with the underlying adaption processes. The adaption processes represent the appropiate realisation of the self-optimization process. If several systems are involved in a self-optimization process, the interrelation of these systems has to be represented. Depending on the task of development further kinds of behavior have to be specified, e.g. kinetics, dynamics or the electromagnetic compatibility of the system-components.

- 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. Incidents can be distinctive influences on the system or terminated activities.
- The **partial model behavior – activities** describes the previously mentioned operation sequences, which take place in a system state, as well as the adaption processes, which are typical elements of self-optimization. The processes are essentially modelled with activities. Examples of activities are *detecting control errors*, *reading sensor data* etc.
- 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. Usually, a sequence diagram represents one of several possible system operations.

3. Conceptual Design of Self-Optimizing Systems

The basic proceeding in the *conceptual design* phase is structured in several sub-phases according to figure 3. These sub-phases are described below.

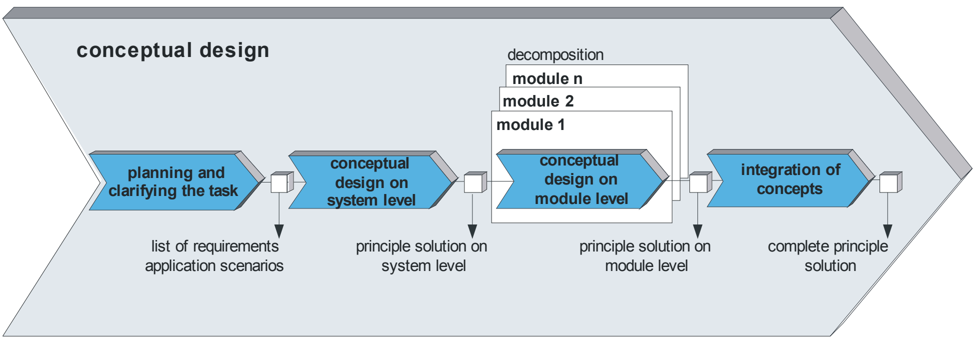


Figure 3. Proceeding during the conceptual design of a self-optimizing system

Planning and clarifying the task: In this phase the task will be identified. Additionally, the resulting requirements for the system, which has to be designed, will be developed. The results are the list of requirements and application scenarios

Conceptual design on system level: Solution variants for each application scenario will be designed based on the previously identified requirements for the system. The best solution variants will be selected and merged into a principle solution on system level. Subsequently, it will be analysed if the principle solution on system level contains inconsistencies and which of these inconsistencies can be solved by self-optimization. For such inconsistencies we define self-optimizing concepts that contain the three steps of the self-optimization process (*analysing the current situation, determining the system of objectives, adapting the system behavior*). The result of this phase is the principle solution on system level.

Conceptual design on module level: The principle solution on module level describes the overall system. In order to assess the technical and economical feasibility of the chosen solution it is necessary to examine the solution in a more detailed way. Therefore, the system will be unitized and for each module principle solutions will be developed as well. The proceeding is consistent with the *conceptual design on system level*, starting with *planning and clarifying the task*. This phase's solution will be represented by the principle solution on module level.

Integration of concepts: The module's principle solutions will be integrated into the overall system. To fulfil this purpose the solutions will be examined for inconsistencies and, again, it will be tested which of the inconsistencies can be solved by self-optimization. Finally, the chosen solution will be evaluated on a technical-economical base. The solution of this phase is the principle solution of the overall system. It acts as a starting point for the following concretization which takes place simultaneously in the separate domains (mechanics, electrical/electronic engineering, control engineering and software engineering).

In the following the two phases *planning and clarifying the task* as well as *conceptual design on system level* are described in detail on the basis of an example mentioned in chapter 1 – a self-optimizing air gap adjustment system of a linear drive of an autonomous acting railway-shuttle. Due to the fact that the *conceptual design on module level* takes place similarly to the *conceptual design on system level* and the *concept integration* already was described above sufficiently, there will not be any further explanation at that point.

In the phase "**planning and clarifying the task**" (fig. 4) the first step will be the abstraction (analyzing the task), and further the core's identification of the development-task.

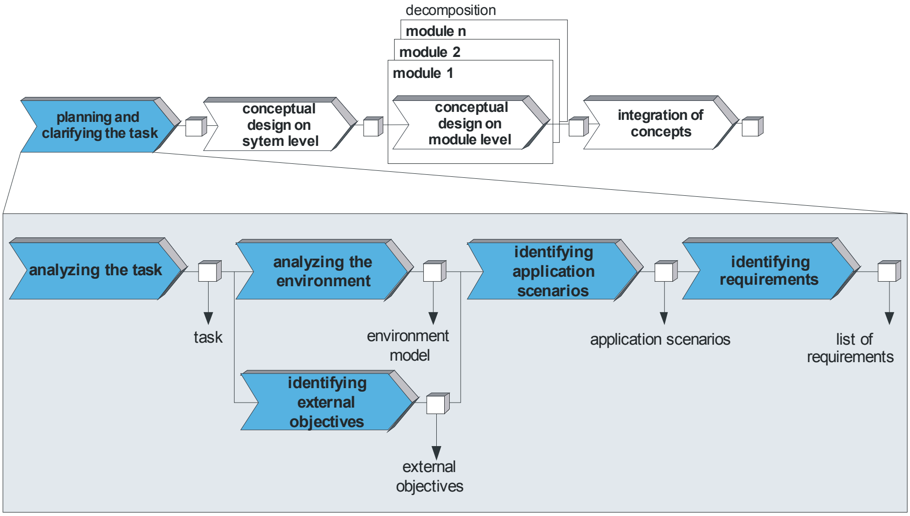
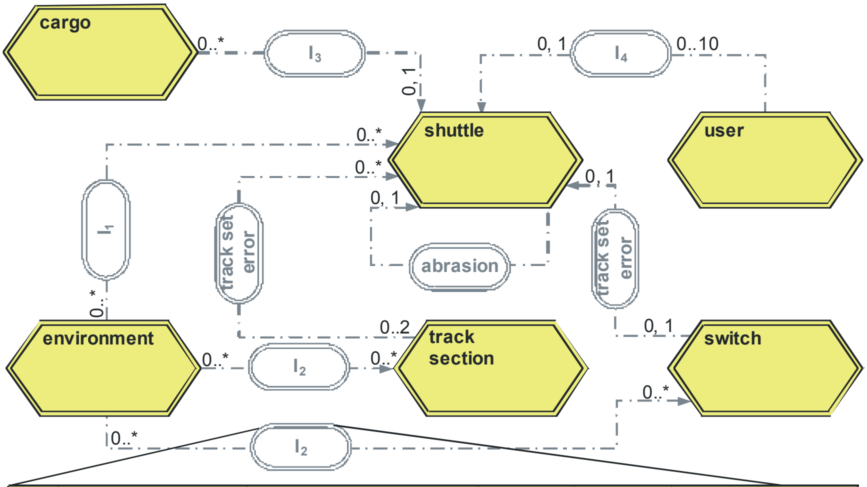


Figure 4. Conceptual design phase "planning and clarifying the task"

Subsequently, an environment analysis will be accomplished, in which the most important edge conditions and influences on the system are determined. In fig. 5 a cut-out from the specification of the environment of a shuttle is illustrated. The users and the cargo affect e.g. by their weight the handling of the shuttles. Values of the environment such as wind, snow, ice and leaves influences the handling of shuttles and the state of the track sections as well as switches.

Track set errors in the track sections have likewise influence on the handling of shuttles such as abrasion of the shuttles themselves. In the shown example the quantity of the influences I2 is concretised by an influence table. Apart from the variable disturbances the external objectives of the system becomes apparent here. Further consistent combinations of influences are established, which are called situations.



No.	factor of influence (short description)	attribute	attribute value	range of tolerance	frequency of appearance	purpose: supporting, disturbing, neutral
1	terrain	inclination	> 5 %	0,30 %	medium	disturbing
2	wind	velocity	> 50 km/h	2 km/h	often	disturbing
3	rain	liters per minute	1 l/min	0,1 l/min	medium	disturbing
4	snow	depth of snow	20 cm	4 cm	rare	disturbing
5	ice	adhesion	<< 1	3	rare	disturbing
6	flood	level above track	> 10 cm	2 cm	rare	disturbing
7	leaves	thickness on rail	> 8 mm	2 mm	rare	disturbing

legend: amount of system elements; variable disturbance; x..y multiplicity; disturbance relation

Figure 5. Model of environment of a shuttle (cut-out)

As a result of the combination of characteristic situations with system states, application scenarios arise which describe a cut-out of the total functionality of the system to be developed. Thus, the drive of the shuttle is realized with the help of a doubly fed linear drive [Zimmer et al., 2005], whereby its efficiency is dependening strongly on the size of the air gap between stator and rotor. The objective of the development is to adapt the air gap as a function of the condition of the track optimally and, thus, to maximise the efficiency. Therefore, the shuttle ask the track-section-control (the "knowledge-supporter") for information about the environment before the transit occurs. The track-section-control transmits available data to the shuttle which uses the data to adjust the air gap [Dangelmaier et al., 2007]. After the transit, the shuttle becomes the "knowledge-provider" and transmits the current data back to the track-section-control of the accordant section. After all application scenarios has been identified, the results of the phase *planning and clarifying the task* are documented in form of demands and wishes in a requirement list [Pahl et al., 2007].
In the phase "**conceptual design on system level**" (fig. 6) the main functions are developed from the requirements and represented in a function hierarchy initially.

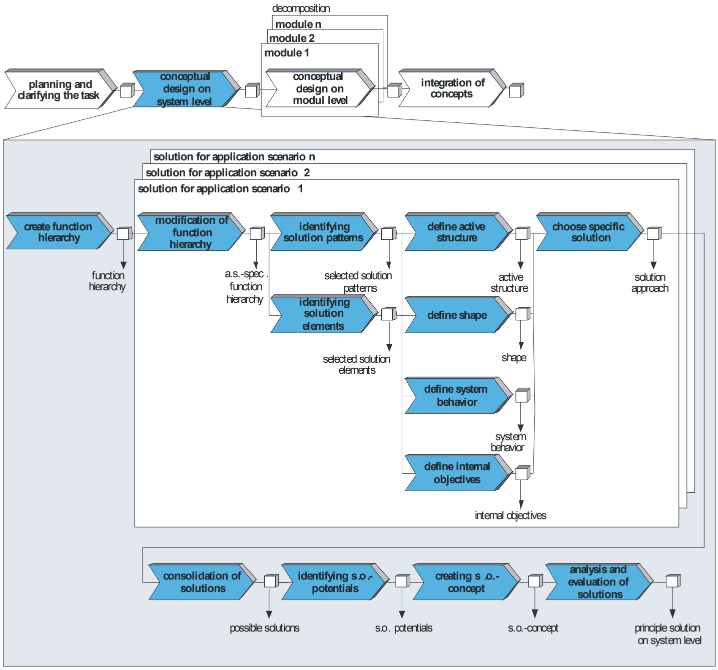


Figure 6. Conceptual design phase "conceptual design on system level"

Subsequently, for each application scenario a solution will be developed. So the function hierarchy will be modified according to the respective application scenario, i.e. irrelevant functions are masked out and specific sub-functions are supplemented. In fig. 7 a cut-out from the function hierarchy of a shuttle is represented focusing on the air gap optimization.

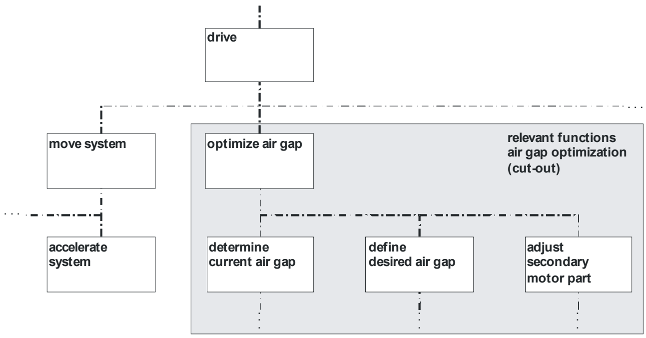


Figure 7. Function hierarchy of a shuttle with focus on the optimization of the air gap (cut-out)

Subsequently, solution patterns for the realisation of the functions documented in the function hierarchy has to be found and recorded into a morphological matrix. Frequently, there exist already realized and proven solutions, which we call ‘solution elements’. If these are applicable, they will be selected instead of the abstract solution patterns. For example, this can be a appropiate electro-mechanical actuator for the adaption of the air gap. Thereby, the search for solution patterns is supported by a solution pattern catalogue. For the determination of the suggestive combinations of solution patterns from the morphological matrix we use the consistency analysis [Koecklering, 2004].
The high-consistent bundles of solution patterns establish the basis for the development of the active structure. In this step the concretization of the solution patterns for system elements takes place. Since

the system elements contain also first approximate data concerning the shape, an initial construction structure can be developed on the basis of the active structure. In fig. 8 a cut-out from the active structure of a shuttle is represented, which concentrates on the relevant system elements and their relations for the air gap optimization. The active structure contains system elements such as primary motor part, secondary motor part, electromechanical actuators, frequency converter, global control, etc. and their relations.
To improve the system's clarity it is structured by the support of logical groups. There are, e.g. system elements which adjust the air gap summarized by a logical group. System elements, in which the process of the objective determination that is relevant for self-optimization, takes place, are characterized by a diagonal arrow (see figure below: global control, electromechanical actuators).

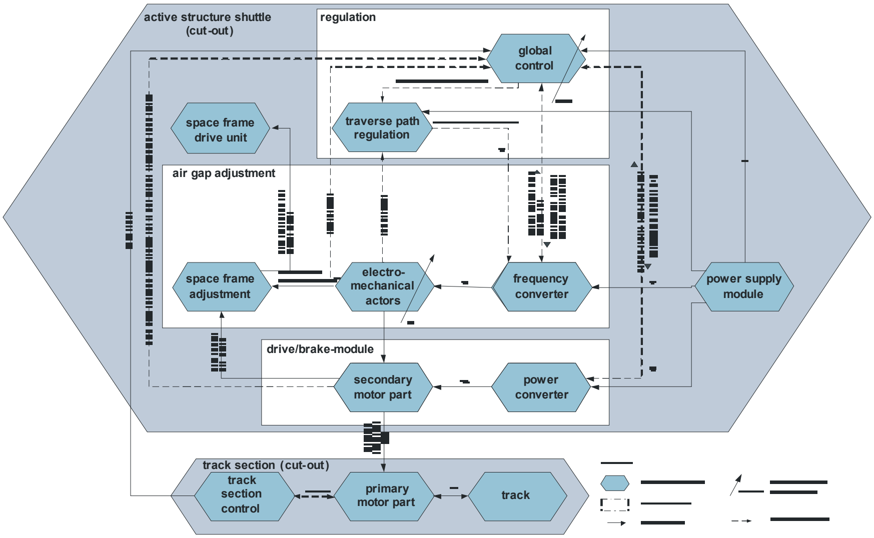


Figure 8. Cut-out from the active structure of a shuttle with focus on the air gap optimization

Furthermore, in this phase the system performance is approximately modelled. This concerns essentially the activities, states and state transitions of the system, as well as communication and cooperation with other systems or sub-systems. Figure 9 illustrates the partial model *behavior – activities* for the application scenario *optimizing efficiency*. In the situation analysis the track data, the shuttle parameters (force/efficiency/security) and also information of other modules like the energy management are queried continuously. In addition, the current objectives are continuously checked concerning their degree of performance. The situation analysis takes place in soft real-time. During the achievement of objectives a new system of objectives is established from shuttle parameters, parameters of other modules and the degree of performance of the current objectives.
During the adjustment of the behavior the development of the air gap is determined in the beginning. According to the air-gap development and in association with the new system of objectives suitable parameters and variants of controllers for the self-optimizing air gap adjustment are chosen. The parameters respectively controller variants serve not only as optimized results for the shuttle, but they are also sent to the track-section-control for the re-use by following shuttles. The analysis of the system behavior provides a appropiate view of the self-optimization process so that the external, inherent and internal objectives can be finally defined.
The developed solutions for the application scenarios have to be merged. Thereby, it must be respected, that operational solutions are generated which enables a reconfiguration of the system at all. Subsequently, it will be identified whether the system contains self-optimization potential or not. Self-optimization potential is present, if changing influences on the system requires adaption of the system-objectives and accordingly the adaption of the system behavior.

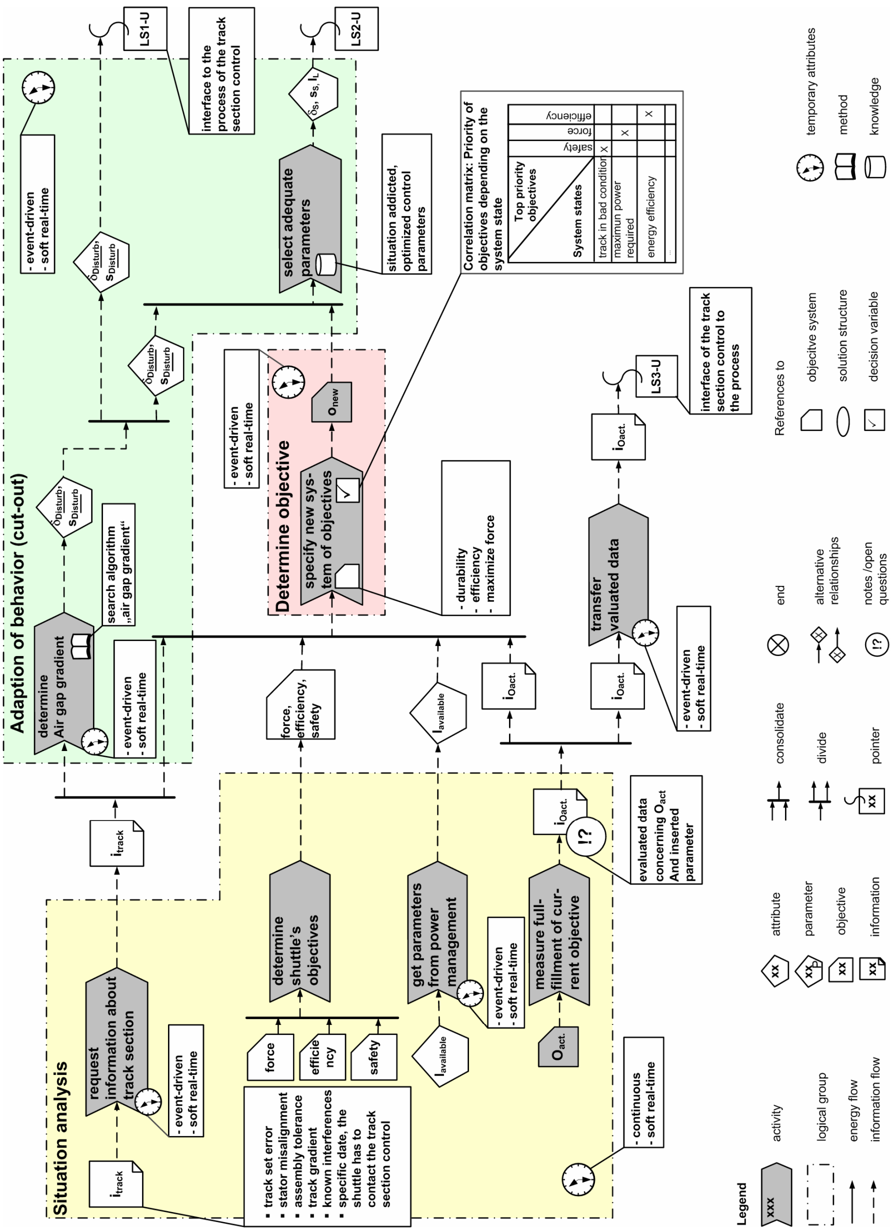


Figure 9. Behavior – Activities for the application scenario "optimize efficiency" (cut-out)

If self-optimization suits, the function hierarchy has to be supplemented around functions of self-optimization. During the development of the self-optimization concept a series of tasks is handled: The self-optimization processes are defined, their consistency is analysed and framework conditions, in which the self-optimization should operate, are determined. The resulting modifications and enhancements for system-structure and -behavior have to be included into the partial models.

The presented procedure is not to be regarded as a stringent sequence of process steps, but it is characterised by a series of iterations which are not depicted in the figures. Order and quantity of iterations depend on the object that has to be modelled, on organisational framework requirements, on individual procedures of developers as well as on the usage of methods. While the process steps are executed, the contents of the principle solution are worked out and specified successively, until a completely specified solution is available at the end of the conceptual design.

4. Summary

The concept of self-optimizing systems is affected by the intensive cooperation of developers from different domains. In the specific domains different specification techniques have been established for describing a system. The communication between experts of different domains is interfered by the usage of different vocabulary and symbols. The principle solution serves as a domain-spanning appreciation about the fundamental structure and the mode of action. It is developed by the help of a set of domain-spanning specification techniques and it serves as a basis for the concretization in advanced phases of construction. Thereby, the presented procedure for the conceptual design of self-optimizing systems serves as a guideline for the development of the principle solution.

Acknowledgement

This contribution was developed in the course of the Collaborative Research Centre 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

Birkhofer, H., "Analyse und Synthese der Funktionen technischer Produkte", Dissertation, TU Braunschweig, 1980

Dangelmaier, W., Gausemeier, J., Frank, U., Kloepper, B., Schmidt, A., Zimmer, D., "Using acitve patterns for the conceptual design of self-optimizing systems examplified by an air gap adjustment system", ASME 2007 - International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Las Vegas, 2007

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

Frank, U., "Spezifikationstechnik zur Beschreibung der Prinziploesung selbst-optimierender Systeme". Dissertation, Fakultaet für Maschinenbau,Universitaet Paderborn, HNI-Verlagsschriftenreihe, Band 175, Paderborn, 2006

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

Koeckerling, M., "Methodische Entwicklung und Optimierung der Wirkstruktur mechatronischer Systeme", HNI-Verlagsschriftenreihe Band 143, Paderborn, 2004

Langlotz, G., "Ein Beitrag zur Funktionsstrukturentwicklung innovativer Produkte", Dissertation, Shaker Verlag, ISBN 3-8265-7246-7, Universitaet Karlsruhe, 2000

Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., "Engineering Design – A Systematic Approach", Springer Verlag, Berlin, 3rd English Edition, 2007

Verein Deutscher Ingenieure (VDI), "VDI-Guideline 2206 – Design Methodology for Mechatronic Systems", Beuth-Verlag, Berlin, 2004

Zimmer, D., Boecker, J., Schmidt, A., Schulz, B., "Elektromagnetische Direktantriebe im Vergleich", in: Antriebstechnik, Ausgabe 2/2005, Vereinigte Fachverlage GmbH, Mainz, 2005

Prof. Dr.-Ing. Juergen Gausemeier
Heinz Nixdorf Institute, University of Paderborn, Fuerstenallee 11,
D-33102 Paderborn
Tel.: ++49-5251-606267
Fax.: ++49-5251-606268
Email: juergen.gausemeier@hni.upb.de
URL: http://wwwhni.uni-paderborn.de/en/rip/