TOWARDS THE DESIGN OF COGNITIVE FUNCTIONS IN SELF-OPTIMIZING SYSTEMS EXEMPLIFIED BY A HYBRID ENERGY STORAGE SYSTEM

Roman Dumitrescu[1], Jürgen Gausemeier[1], Christoph Romaus[2]

[1]Heinz Nixdorf Institute, University of Paderborn, 33102 Paderborn, Germany
[2]Power Electronics and Electric Drives, University of Paderborn, 33098 Paderborn, Germany

Abstract: Machines are omnipresent. They produce, they transport. Machines facilitate work and assist. The increasing penetration of mechanical engineering by information technology enables considerable benefits. This circumstance is expressed by the term mechatronics, which means the close interaction of mechanics, electronics, control engineering and software engineering to improve the behavior of a technical system. The conceivable development of information technology will enable mechatronic systems with partial intelligence. We refer to this by using the term "self-optimization". Self-optimizing systems react autonomously and flexibly on changing environmental conditions. They have to learn and optimize their behavior during operation. Hence, the design of such systems is an interdisciplinary task. Mechanical, electrical, control and software engineers are involved as well as experts from mathematical optimization and artificial intelligence. Furthermore, self-optimizing systems adopt functions, which come with the territory of cognitive systems and are known as cognitive functions. In order to design self-optimizing systems, we have to consider aspects of the paradigm of cognition, too. Even though in the last years more and more theories of modeling cognitive behavior in technical systems were developed and published, an applicable support of the system developer, especially in the early stages of the development process, is missing. Already the identification of self-optimization and appropriate systems functions is a challenge for the system designer. This contribution presents an approach to design cognitive functions in mechatronic systems and provides a subsystem of an innovative railway technology as a concrete example to demonstrate how the development of future mechatronic systems can profit from such an approach. This subsystem is a hybrid energy storage system (HES), consisting of a NiMH-battery and a double layer capacitor (DLC) and managed by a self-optimization operating strategy.

Keywords: Mechatronics, Self-Optimization, Cognitive Functions, Energy Management, Operating Strategy.

I 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 aim of mechatronics is to improve the behavior of technical systems by using sensors to obtain information about the system environment and the system itself. The processing of this information enables the system to react optimally to its current situation. 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. Therefor we use the term "self-optimization". Self-optimization enables mechanical engineering systems that have the ability to react autonomously and flexibly on changing operation conditions. Self-optimization takes place as a process of the three actions "analyzing the current situation", "determining the system's objectives" and "adapting the system's behavior". We refer to this series of actions as the "Self-Optimization Process".

During the self-optimization process functions for self-optimization such as "to share knowledge", "to coordinate behavior" or "to learn from experience" have to be implemented. Those functions are also typical for cognitive systems and are known as cognitive functions. It is easy to see that a technical system, that realizes cognitive functions to use information, for instance, about its surroundings and itself in order to adjust its own behavior, is distinctly more viable and robust in operation than conventional reactive controlled systems.

The design of such systems, however, is a challenge. Established design methodologies of conventional mechanical engineering [1] and also methodologies of mechatronics [2] are no longer adequate to that task. Within the Collaborative Research Centre (CRC) 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" at the University of Paderborn, a new design methodology is developed. It facilitates engineers from different domains to describe systems in a comprehensive domain-spanning way and enables the reuse of proven solutions in the form of solution patterns [3]. We distinguish between solution patterns, which relay on physical effects, and patterns, which serve data processing. However, there are no solution patterns, which consider the paradigm of self-optimization or cognitive functions. Thus "Active Patterns for Self-Optimization" (AP_{SO}) were devised to indicate potential solutions for self-optimization information processing [4], [5].

This paper is organized as follows: In section 2 we will present the general concept of self-optimization and point out the connections to the paradigm of cognition. We will then focus on the development of self-optimizing systems, providing methods and techniques to rise to the challenge. According to the methods presented in section 3, we will demonstrate how they should be used by given a brief overview of the development of an innovative energy storage system. Eventually we will give a short summary of our contribution.

II SELF-OPTIMIZING SYSTEMS AND COGNITIVE FUNCTIONS

A. Self-Optimization

The key aspects and the mode of operation of a self-optimizing system are illustrated in Figure 1.

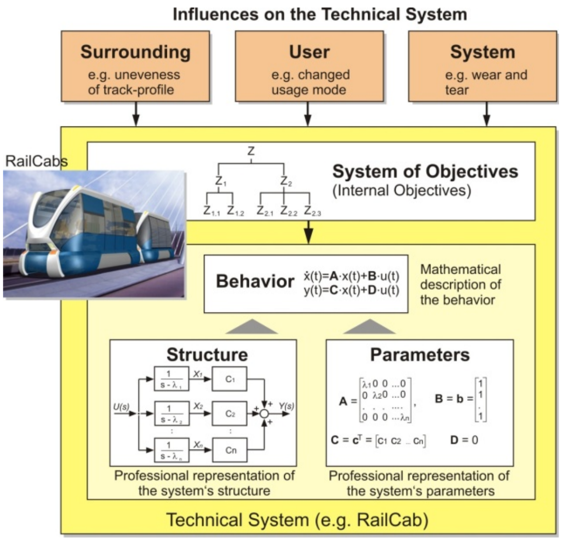

Figure 1: Aspects of a self-optimizing system

The self-optimizing system determines its currently active objective on the basis of the encountered influences. This means, for instance, new objectives are added or existing objectives are discarded and no longer pursued during operation. Adapting the objectives in this way leads to an adjustment of the system behavior. This is achieved by adapting the parameters or, if necessary, the structure of the system (e.g. switching between different controller types). 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 thus their relationships.

The Self-Optimization within the system takes place as a series of the three following actions to which we refer as **Self-Optimization Process** [6]:

1. **Analyzing the current situation:** The current situation includes the current state of the system as well as all observations of the environment that have been carried out.

2. **Determining the system's behavior:** The current system's objectives can be extracted from selection, adaptation and generation.

3. **Adapting the system's behavior:** The changed system of objectives demands an adaptation of the behavior of the system. This can be realized by adapting the parameters and/or by adapting the structure of the system.

The self-optimization process leads, according to changing influences, to a new state. Thus, a state transition takes place. The self-optimization process describes the system's intelligent behavior.

The necessity to divide the architecture of information processing of complex systems into several hierarchical levels is also well-grounded in cognitive science. Whereas technical systems basically only have one reactive action-level (the mechatronic control loop), research in cognitive science tries to prove that complex systems possess not only a reactive behavior, but also can modify the coupling between detecting data and carrying out an action. This modification can be considered as learning. For the mechatronic control loop, which is the motor loop in Figure 2, means this that the direct data processing between sensors and actuators needs to be extended. Conventional control strategies are not sufficient enough. Furthermore, cognitive information processing must not replace the direct and reactive coupling, but has to co-exist with it [7]. To meet these requirements, the concept of the **Operator-Controller-Module (OCM)** was developed (Fig. 2) [6].

From an information point of view the OCM is conform to a software agent and composed of following three levels:

- **Controller:** On this level the access through the technical systems takes place. Since this control loop is an active chain that obtains measurement signals, determines adjustment signals and outputs them, it is called “motor loop”.

- **Reflective Operator:** This level monitors and directs the controller by provided “controller configurations”. Those can be combinations of control units, switch elements or signal flows. The exchange between controller and reflective operator takes place in the “reflective loop”.
- **Cognitive Operator:** At the highest level of the OCM, the system can launch a variety of methods and algorithms to use information about the environment and itself to improve its own behavior. In order to realize cognitive functions, at least a knowledge base must be implemented. The other entities of the cognitive operator depend on the design requirements. The information flow between cognitive and reflective operator is called “cognitive loop”.

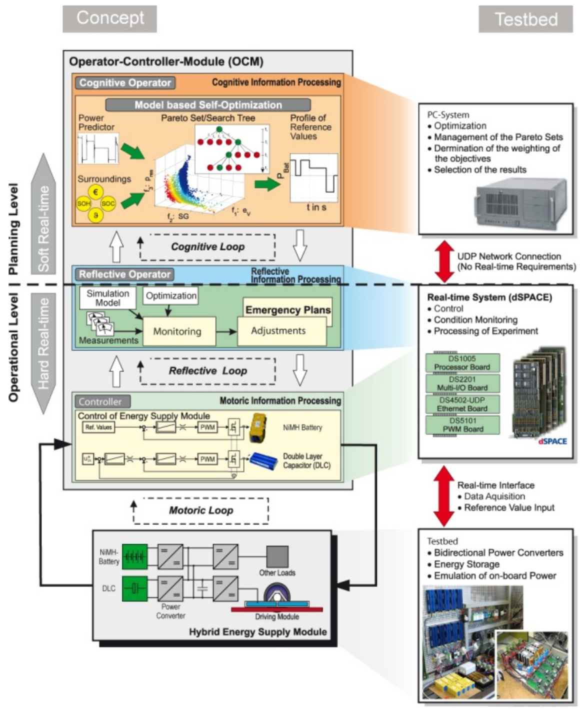

Figure 2: Concept and realization of the Operator-Controller-Module for the hybrid energy storage system (cp. ch. IV)

B. *Cognitive Functions*

Self-optimizing systems additionally to conventional mechatronic systems perform functions such as “to communicate”, “to share knowledge”, “to extract information”, “to determine objectives”, or “to change controller structures”. Several combinations of these functions constitute self-optimization processes. Many of those functions for running self-optimization come with the territory of cognitive systems, therefore known as cognitive functions [7], [8].

There are many appellations and classifications of cognitive functions within the field of cognitive science [9]. STRUBE, for instance, distinguishes “to observe”, “to recognize”, “to map”, “to memorize”, “to think,” “to solve problem”, “to control motor” and “to use language” [10]. This classification serves only as a very abstract, first order of cognitive function. An explicit classification of cognitive functions regarding technical functions is still missing.

Whereas self-optimizing systems feature only a certain number and combination of these functions to adapt ideally to their changing environment, cognitive systems are provided with the whole variety of cognitive functions in order to achieve their objectives even under circumstances which were ignored during their system design. Hence, cognition can be characterized as the ability that enables not only autonomous and adapting, but also more reliable, effective and viable systems regarding their purpose [11]. As a consequence we have to consider aspects of the paradigm of cognition in order to design self-optimizing systems.

As it is easy to see, we can range the mentioned cognitive functions into the three phases of our self-optimization process. Table 1 shows a classification of cognitive functions for self-optimizing systems on a rather abstract level. This classification can be the starting point for a further detailed classification in order to use those functions for the design of technical systems.

Actions of self-optimization	Cognitive functions
Analyzing the situation	“to observe”, “to recognize”, “to map”
Determining the system’s behavior	“to memorize”, “to think,” “to solve problem”
Adapting the system’s behavior	“to control motor”, “to use language”

Table 1: Classification of cognitive functions regarding self-optimization

III DEVELOPMENT OF COGNITIVE FUNCTIONS

Existing methodologies for the development of technical systems need to be fundamentally extended and added by domain-spanning methods and tools to handle the complexity of the development, in particular during the early stages of the development. There are three basic elements of our new methodology: a procedure model for the complete development process, the domain-spanning specification technique for the holistic description of the system to develop and a new type of solution pattern for the effective reuse of our methods by thirds.

A. Design Process

On the highest degree of abstraction, the development process of self-optimizing mechatronic systems can be subdivided into two serial processes: the domain-spanning conceptual design and the domain-specific "concretization" (see Fig. 3). Within the conceptual design, the basic structure and the operation modes of the system are defined. It starts with planning and clarifying the development task. An interdisciplinary team, consisting of specialists of different domains, creates the so-called principle solution of the system to develop. It describes not only the main physical characteristics of the system, but also the logical operating characteristics. In order to handle the complexity a decomposition of the system into modules will take place during the conceptual phase.

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 domain-spanning principle solution. The aim of the concretization is the complete description of the system by using the construction structure and the component structure. Thus, all modules are developed in parallel in all participating domains. An overall system model, which is based on the principle solution, guarantees the effective and correct integration to complete the system design.

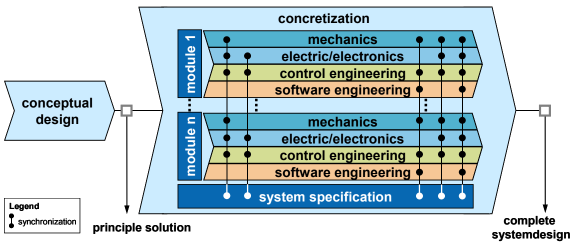

Figure 3: Development of self-optimizing mechatronic systems [12]

B. Specification Technique

Within the conceptual design phase, we use a semi-formal, domain-spanning specification technique to describe the principle solution of a self-optimizing system. The principle solution represents a significant milestone since it is the result of the conceptual design phase. The specification technique was developed within the CRC 614 and is based on the research of FRANK, GAUSEMEIER and KALLMEYER [13], [14]. At the beginning of 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 following aspects need to be taken into account: requirements, environment, application scenarios, functions, active structure, system of objectives, shape and behavior (see Fig. 4).

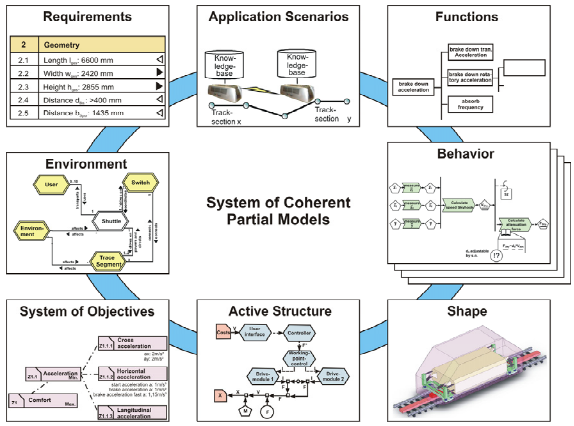

Figure 4: Coherent system of partial models

The mentioned aspects are captured in the principle solution and described by partial models. The partial models are intertwined and form a coherent system model. The relations are modeled between the constructs of the relating partial models and amount to a coherent system. By using this specification technique, the system that is to be developed will be described in a holistic, domain-spanning way. The specification 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.

C. Solution Patterns

One leading point of the design methodology is the reuse of proven solution in form of patterns. Generally a pattern describes a recurring problem in our surrounding and the core of a solution to that problem [3]. The solution core is specified in a "solution pattern" that defines the characteristics of the system's elements that is to be developed and the interactions between those elements. We distinguish between solution patterns, which relay on physical effects, and patterns, which serve information processing. However, there are no solution patterns, which consider the paradigm of self-optimization not to mention cognitive functions. For this purpose "Active Patterns for Self-Optimization" (AP_{SO}) were devised. According to a domain-spanning description an AP_{SO} requires several aspects which can be modeled with the presented specification technique (see Fig. 5).

The aspect *functions* lists and describes the functions for self-optimization, which are implemented by the AP_{SO}. This aspect is essential to modify and complete the present system's functions. We think it is a suitable and elegant way to integrate cognitive functions in the system design.

The *principle concept* characterizes the underlying ideas behind the AP_{SO}. It enables developers to acquire an intuitive understanding of the AP_{SO} without any further formal details.

The aspect *structure* describes which system elements are fundamentally necessary in order to implement the AP_{SO} and how those system elements are interrelated.

The aspect *behavior* describes the self-optimization process. Thus, we need to model the autonomous, cognitive behavior that initiates, supports and/or effects state transitions. The behavior aspect is split in two sub aspects. According to the corresponding partial model these sub aspects are behavior – state and behavior – activity. The aspect behavior – state models possible sequences of states and state transitions realized by the system's functions. The aspect behavior – activity aspect describes the activities that are performed by system elements during a certain point within the self-optimization process.

Methods serve to implement the self-optimization processes, in particular to adapt objectives and behaviors (as a result of adaptations to parameters and possibly also structures). The aspect methods is a set or catalogue of methods which can be used to implement the self-optimization process during later on. Examples of such methods are fuzzy neural rule switching for adapting internal objectives, and case-based planning for experience-based behavior adaptation.

The aspect *application scenario* describes successful established applications of AP_{SO}. An application scenario encompasses the same aspects as the AP_{SO}. The aspects of application scenario are instances or concretization of the aspects of the corresponding AP_{SO}.

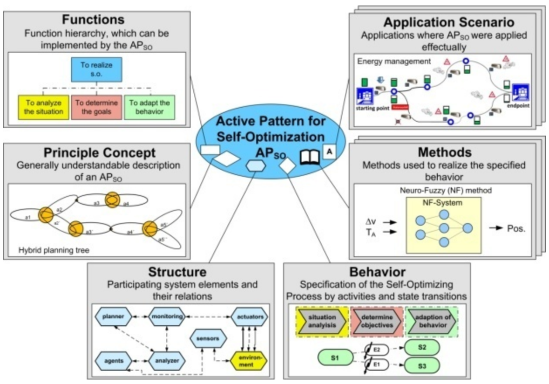

Figure 5: Aspects of an active pattern for self-optimization

The digital representation of an aspect of an AP_{SO} in called a partial model. This means, that only if all aspects are implemented the respective partial models build the whole model of an AP_{SO}. The main challenge for the use of AP_{SO} is on the one hand to find methods and respective system elements for the big variety of cognitive functions, and on the other hand the fusion of the partial models of the AP_{SO} and the partial models of the current system design. The successful integration of eligible AP_{SO} results in an early specification of the information processing, which facilitates not only the work in the following concretization phase, but also the consistency of the system design.

Once a self-optimization process has been specified and implemented successfully, the information needs to be documented in order to enable the reuse in different applications as well. This is necessary not only if the developer could not refer to an existing AP_{SO} (in this case a new AP_{SO} has to be described), but also if an AP_{SO} was implemented (in that case at least a new application scenario has to be described). Apparently it is necessary to use a database, which stores the information in such a manner, that developers are able to recognize appropriate AP_{SO}. Therefore we developed a user-friendly knowledge base for the systematic management of AP_{SO}, called "Active Pattern Knowledge Base". The graphical user interface of the Active Pattern Knowledge Base is illustrated in Figure 6.

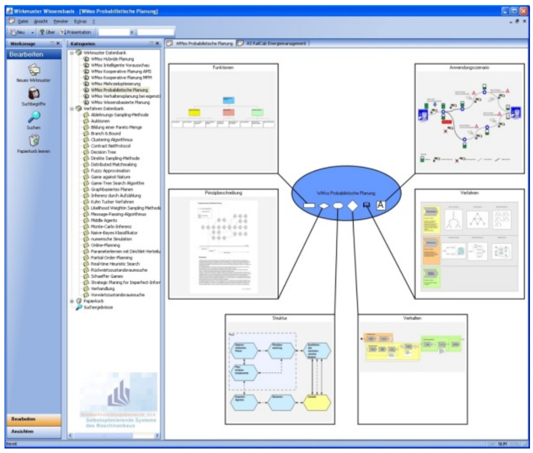

Figure 6: GUI of the active pattern knowledge base

The Active Pattern Knowledge Base links all the aspects of an active pattern to the respective partial models, which are specified in Microsoft Visio. However, the user can see a preview of the partial model. Furthermore, information regarding the various methods is stored directly within the program. Both, the active pattern and the methods are stored as a list to enable a quick access. To find an appropriate active pattern, a fulltext search is available, which looks for matches between the search word and the notions used in the partial models. The interface to an ontology based search is also implemented. A preliminary ontology for matching related technical functions supports the fulltext search. Search results are presented with a percentaged quality.

IV Application Example: Hybrid Energy Storage System (HES)

Currently we are using the presented methods to design an intelligent hybrid energy storage system, which can optimize its modes of ooperation by itself according to different consumer loads and other system surroundings. The system shall supply both train vehicles and motor vehicles. The project is part of an innovative railway prototype called "Neue Bahntechnik Paderborn/RailCab" [15]. RailCabs are autonomous vehicles that supply transport for both passengers and cargo on demand without any stops or changes of train and reduce energy consumption by forming convoys. They feature a substantially higher riding comfort due to active damping of the coaches and active guidance. The vehicles are propelled by a doubly fed linear motor drive, which offers the possibility to build or clear convoys while driving due to a high and defined propulsive force. It is further possible to transfer energy by this linear motor drive, thus there is no catenary or conductor rail to supply the loads on board.

Unfortunately, the power transfer is not possible under all driving conditions. An on-board energy storage backs up the supply of the vehicle. The main requirements are the ability to store a high amount of energy to offer long operation as well as to provide a high power while charging and discharging to cover all demands of the loads and to store the maximum transferred power of the linear motor drive. As the energy storage is installed on a vehicle, its mass and volume has to be small as well. Low costs, a high efficiency and a long life-span without maintenance are further requirements.

To meet these requirements, we decided to combine different energy storage devices with complementary characteristics into a hybrid energy storage system (HES), consisting of a NiMH-battery and a double layer capacitor (DLC) (Fig. 7) [16].

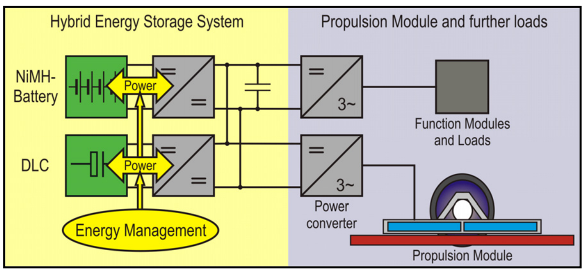

Figure 7: Principle concept of the hybrid energy storage system (HES)

The HES combines long term storage – batteries – featuring high energy density with short term storage – double layer capacitors (DLC) – offering high power density and high cycliability.

The structure of the hybrid energy storage system offers a degree of freedom for the distribution of the power flow of the two storage devices. An energy management is necessary to benefit from this. This energy management should be able to react to varying influences from the surroundings of the vehicle and to adapt its behavior adequately. Depending on the situation different objectives may be important, e.g. a high efficiency, low deterioration or increased availability [17]. Operating strategies for the energy management can be realized by miscellaneous ways, e.g. by the limitation of battery power, often combined with a velocity-dependent adaptation of the state of charge of the DLC [18], [19]. Though, such conventional strategies offer only adaptation by static rules; they neglect changes due to the system's surroundings. Therefore we systematically designed and implemented a complete self-optimization process in a test-rig including both continuous and discrete optimization methods for the HES.

In order to realize self-optimization in the HES, the cognitive functions have to be identified from the described requirements. Additional functions, which result from a chosen part-solution, have to be added in order to specify the complete partial model "functions". Figure 8 shows a simplified cut-out of this final partial model of the HES, focusing on the cognitive functions for the mechatronic control loop in respective to the presented classification from Table 1.

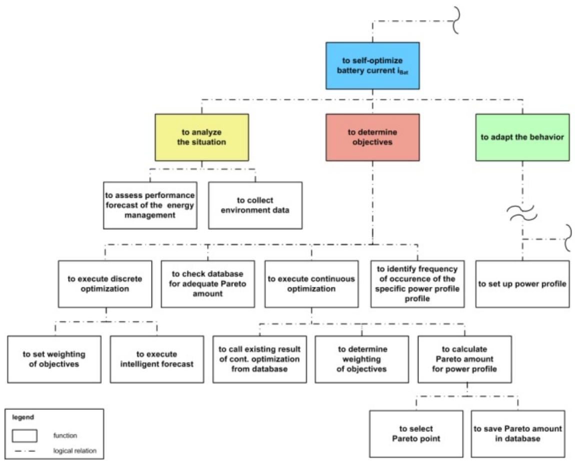

Figure 8: Cut-out of the partial model "functions" of the HES

The pigmented functions are the main functions for the self-optimization process. They are broken down into sub-functions regarding possible system elements, which realize those functions that cannot be further detailed by a sub-function. The description and arrangement of the system elements defines the basic structure of the self-optimizing HES. Figure 9 visualizes the simplified partial model "active structure" of the HES, consisting of those system elements.

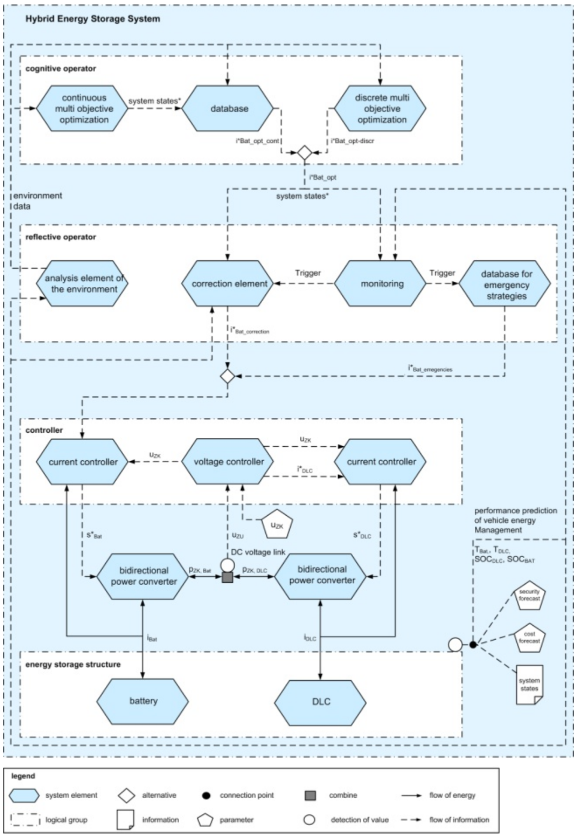

Figure 9: Simplified partial model "active structure" of the HES

There are logical groups (cognitive operator, reflective operator, controller and energy storage structure), which specify the self-optimization information processing architecture (cp. operator-controller-module Fig. 2). The "cognitive cooperator" of the HES consists of two different types of optimizer (continuous and discrete) and a database for the continuous optimizer. The "controller" is realized by current and voltage control loops for the HES. The "reflective operator" contains system elements for the monitoring and correction, but also a database with emergency strategies in order both optimizer fail. The "structure" of the HES consists of the two types of energy storage devices. Two bidirectional power converters control the power flows to/from the storage.

For well known track sections travelled frequently, we chose to implement an offline "continuous multi-objective optimization". For these frequently applied power profiles, according precalculated optimization results in terms of Pareto sets are retrieved from a database [16]. Since there is no guarantee that the required power profile of the HES is already stored, a second optimization was implemented, a "discrete optimization" for the online calculation of the operating strategy [20]. How the HES executes those optimizations is illustrated in Figure 10.

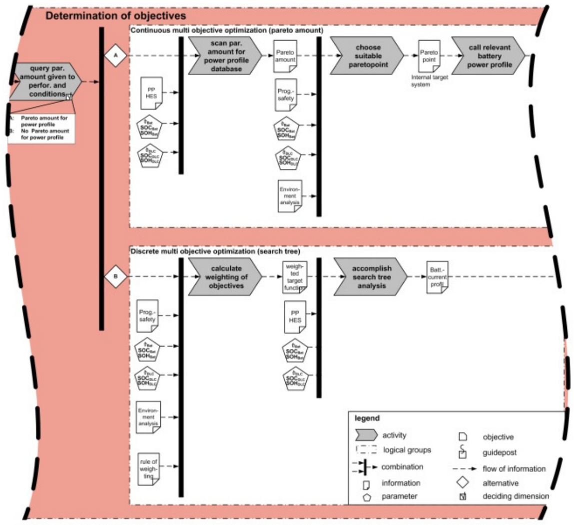

Figure 10: Cut-out of the partial model "Behavior – activitiy" of the HES

The activities (grey arrows) are system functions which are executed during operation. The depicted partial model "behavior-activity" shows only the second action of the self-optimization process, the determination of objectives. In the case a power profile is in the database the continuous optimization starts; if not, the discrete optimization is performed (lower series of activities).

In order to support the reuse of the specified and evaluated solution, we concretized two active pattern for self-optimization and stored them in the active pattern knowledge base (cp. Fig. 6): AP_{SO} "Multi-Objective-Optimization" based on the continuous optimization method and AP_{SO} "Intelligent Preview" based on the discrete optimization method. Both active patterns were applied successfully in other technical system [20].

V RÉSUMÉ

We believe that self-optimizing systems are the mechatronic systems of tomorrow. In our contribution we introduce an approach to design cognitive functions in such systems. First we declared self-optimization as a paradigm for innovative mechatronic systems and looked into the complex field of cognitive functions. Accordingly we provided not only a generic procedure model, but also our hybrid energy storage system (HES) as an application example to demonstrate the development of an intelligent technical system. To guarantee the effective reuse of once successfully proven solutions for the design of cognitive functions in self-optimizing systems, we presented a new type of solution pattern called Active Pattern for Self-Optimization (AP_{SO}).

ACKNOWLEDGMENT

This contribution was developed in the course of the Collaborative Research Centre 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" funded by the German Research Foundation (DFG). One author obtains a research fellowship by the International Graduate School "Dynamic Intelligent Systems" supported by the federal state North Rhine-Westphalia, Germany.

REFERENCES

[1] G. Pahl, W. Beitz, J. Feldhusen, K.-H. Grote. *Engineering Design – A Systematic Approach*. Springer Verlag, London, 3th edition, 2007

[2] Verein Deutscher Ingenieure (VDI). *VDI- Richtlinie 2206 – Design methodology for mechatronic systems*. Beuth Verlag, Berlin, 2004

[3] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, A. Angel. *A Pattern Language*. Oxford University Press, New York, 1977.

[4] A. Schmidt. *Wirkmuster zur Selbstoptimierung: Konstrukte für den Entwurf selbstoptimierender Systeme*. HNI- Verlagsschriftenreihe, Band 204, Paderborn, 2006.

[5] J. Gausemeier, D. Zimmer, U. Frank, B. Klöpper, A. Schmidt. *Using Active Patterns for the Conceptual Design of Self-Optimizing Systems Examplified by an Air Gap Adjustment System.* 27th Computer and Information in Engineering Conference, Las Vegas, 2007.

[6] P. Adelt, J. Donoth, J. Gausemeier, J. Geisler, S. Henkler, S. Kahl, B. Klöpper, A. Krupp, E. Münch, S. Oberthür, C. Paiz, M. Porrmann, R. Radkowski, C. Romaus, A. Schmidt, B. Schulz, H. Vöcking, U. Witkowski, K. Witting, O. Znamenschykov: *Selbstoptimierende Systeme des Maschinenbaus – Definitionen, Anwendungen, Konzepte*. HNI-Verlags-schriftenreihe, Band 234, Paderborn, 2009

[7] G. Strube. *Modelling Motivation and Action Control in Cognitive Systems*. in: U. Schmid, J. Krems & F. Wysocki (Eds.), Mind Modelling, Pabst, Berlin, 1998.

[8] J.R. Anderson, C. Lebiere. *The Atomic Components of Thought*. Lawrence Erlbaum, New Jersey, 1998

[9] B. S. Müller. *Identifikation elementarer kognitiver Leistungen*. GMD Report 17. GMD- Forschungszentrum Informationstechnik GmbH, Sankt Augustin, 1998.

[10] G. Strube. *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta, 1996.

[11] D. Vernon, G. Metta, G. Sandini. *A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents*. IEEE Transactions on Evolutionary Computation, Vol. 11, No. 2, pp. 151-180, 2007

[12] J. Gausemeier, D. Steffen, J. Donoth, S. Kahl: *Conceptual Design of Modularized Advanced Mechatronic Systems*. International Conference on Engineering Design (ICED), Stanford, 2009

[13] F. Kallmeyer: *Eine Methode zur Modellierung prinzipieller Lösungen mechatronischer Systeme.* Dissertation, Fakultät für Maschinenbau, Universität Paderborn, HNI-Verlagsschriftenreihe, Band 42, Paderborn, 1998

[14] J. Gausemeier, U. Frank, J. Donoth, S. Kahl: *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), 2008

[15] Website of the RailCab project; http://www.railcab.de/

[16] C. Romaus, J. Böcker, K. Witting, A. Seifried, O. Znamenshchykov. *Optimal Energy Management for a Hybrid Energy Storage System Combining Batteries and Double Layer Capacitors.* Accepted by IEEE Energy Conversion Congress and Exposition (ECCE2009), 2009

[17] S. Suzuki, J. Baba, K. Shutoh, E. Masada. *Effective application of superconducting magnetic energy storage (SMES) to load leveling for high speed transportation system.* IEEE Transactions on Applied Superconductivity, Volume 14, 2004

[18] H. Li; D. Liu. *Power Distribution Strategy of Fuel Cell Vehicle System with Hybrid Energy Storage Elements Using Triple Half Bridge (THB) Bidirectional DC-DC converter*. 42nd Industry Applications Conference, pp. 636-642, New Orleans, September 23-27, 2007

[19] D. Heinemann. *Strukturen von Batterie- und Energiemanagementsystemen mit Bleibatterien und Ultracaps*. PHD-Thesis, Technical University of Berlin, 2007

[20] J. Gausemeier, O. Znamenschykov, S. Oberthür, H. Podlogar. *An Approach for Achieving Self-Optimization in Mechatronic Systems by Active Pattern*. 8th International Conference on Intelligent System Design and Applications, November 26-28, Kaoshing, 2008