Design Framework for the Integration of Cognitive Functions into Intelligent Technical Systems

R. Dumitrescu[1]*, H. Anacker[1] and J. Gausemeier[2]

[1]Fraunhofer Institute for Production Technology IPT, Project Group Mechatronic Systems Design, Zukunftsmeile 1, 33102 Paderborn, Germany
[2]Heinz Nixdorf Institute, University of Paderborn, Fürstenallee 11, 33102 Paderborn, Germany
*E-Mail: roman.dumitrescu@ipt.fraunhofer.de, juergen.gausemeier@hni.uni-paderborn.de[2]

Summary: The integration of cognitive functions will enable mechatronic systems to be superiorly embedded into their environment and to follow their system objectives independently. The intention is to develop intelligent technical systems, which can optimize their behavior by themselves to become more flexible, robust and user-friendly. Numerous challenges, however, become apparent on the way to such self-optimizing systems. For instance, there is a lack of a systematic coupling of those disciplines, which are relevant for the exploration of cognitive functions, with the general engineering approach in product development. To rise to these challenges, the integration of cognitive functions has to be supported already during the early stages of the development with some kind of methodology. Important requirements occur in terms of the intensified interdisciplinarity of the development and the increasing system complexity. Therefore a design framework for the integration of cognitive functions into self-optimizing systems has been developed, which integrates both existing and newly developed methods in a well-structured procedure. This contribution presents this approach and its exemplary application.

Keywords: Design Methodology, Mechatronics, Conceptual Design, Cognitive Functions.

1. Introduction

Today mechanical engineering products are characterized by the close interaction of mechanics, electronics, control engineering and software engineering. This interaction is expressed by the term mechatronics. In order to categorize the variety of applications, the diversity of mechatronic systems can be centralized and expressed by the three categories "Electromechanics", "Multibody Systems" and "Intelligent Systems" (Figure 1).

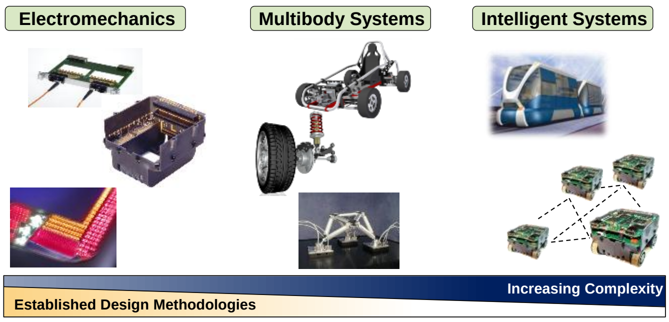

Figure 1. Variety of Mechatronic Systems.

The first category is based on the integration of mechanics and electronics. The aim is to reach a high density of mechanical and electronic functions within a limited space. The second category deals with the controlled movements of multibody systems. The objective is to improve the system's behavioral movement. Mechatronic systems are always part of the two categories.

The conceivable development of information technology opens up fascinating perspectives, which have the potential to go far beyond current standards. Keywords as "self-optimization", "things that think", "Cyber-Physical Systems" or "Industry 4.0" express this perspective on Intelligent Technical Systems. Such systems are characterized by the integration of cognitive functions, methods and technologies of non-technical disciplines, e.g. cognitive science or neurobiology. Based on the implementation of self-x properties or intelligence for monitoring and controlling processes, these systems go far beyond current standards: they are adaptive, robust, fore-sighted and user-friendly [1], [2].

However, the intention is to develop systems with inherent partial intelligence. The paradigm of *self-optimization* is the main focus of the Collaborative Research Centre (CRC) 614 "Self-Optimizing Concepts and Structures in Mechanical Engineering" at the University of Paderborn [3]. Self-optimizing (s.o.) systems are based on mechatronic systems, but have the ability to react autonomously and flexibly on changing systems or environmental conditions during operation mode. Such intelligent technical systems lead to a considerably more complex system due to their new functionality and decentralized system structure.

Due to the participation of the different disciplines (technical and non-technical) the development of s.o. systems is also more complex. Even though design methodologies of conventional mechanical engineering (e.g. PAHL/BEITZ [4]) and mechatronics (e.g. VDI-guideline 2206 "Design methodology for mechatronic systems" [5]) exist, they cannot come up to the demand of s.o. systems. As a consequence within the CRC 614 a new design methodology is getting developed in order to support engineers to develop s.o. systems in a comprehensive way.

The focus of this contribution lies on a design framework for the integration of cognitive functions into technical systems. We will explain the general concept of self-optimization and its connections to the paradigm of cognition in section two. Section three will present a design framework consisting of the system specification as well as our methodology.

2. Self-optimization and Cognitive Functions

The key aspects and the mode of operation of a s.o. system are illustrated in Figure 2. The system determines its currently active objectives on the basis of the encountered influences on the technical system of its environment. New objectives can be added, existing objectives can be rejected or the priority of objectives can be modified during operations. Therefore the system of objectives and its autonomous changing is the core of self-optimization. Adapting the objectives in this way leads to a continuous adjustment of the system`s behavior to the environmental state. This is achieved by adapting parameters or reconfiguration of the structure (e.g. adapting controller parameter).

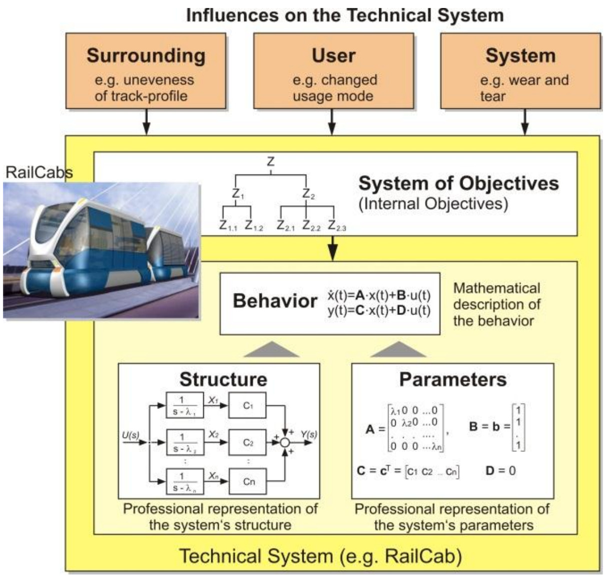

Figure 2. Aspects of Self-Optimizing Systems [2].

The s.o. process consists of the three following actions: Analyzing the current situation, determining the system`s objectives and adapting the system`s behavior. In the first phase the observation differs in recalling the sensor data, analyzing the fulfillment of objectives and the indirect release of information by communication with other systems. After analyzing this information the system independently determines objectives. The loop of self-optimization is finished by adapting the behavior.

In order to realize such a self-optimization process s.o. systems perform information processing functions such as *to communicate*, *to share knowledge* or *to extract information*. Such functions are known as cognitive functions. Even though there is no unified definition of cognition, there is a common sense that cognition intervenes between the perception and the behavior of a system in such a way, that a certain stimuli does not result always in the same reaction, but in the best possible [6]. Therefore 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 [7].

STRUBE distinguishes following **cognitive functions** on a psychological level [6]: *to observe*, *to recognize*, *to encode*, *to store*, *to remember*, *to think*, *to solve problem*, *to control motor function* and *to use language*. Thus, cognitive functions are basically information processing functions, which not only formalize new information, but also connect new information with existing internal information. Since cognitive functions process information– and this is the main assumption of cognitive science – they are calculation processes and can be implemented in technical systems, too [8].

3. Framework for the conceptual design

We developed a design framework, which supports the developer already during the conceptual design in order to identify, select and specify solutions for the integration of cognitive functions into technical systems. The result is the early specification of the cognitive and non-cognitive information processing, which is the basis for concretization and implementation. We will first describe how to describe the system and how once successfully proven solutions should be documented for the reuse. Afterwards we will present the procedure model, which is the core of the design framework. The framework was used to design the s.o. operating strategy of a hybrid energy storage system (HES) of an innovative railway vehicle. The HES combines two different technologies – conventional batteries (NiMH) for a high energy density and double layer capacitors (DLC) for a high power density. This is necessary in order to offer a continuous power supply for the on-board loads of the vehicle [3].

3.1. System specification

Within the conceptual design, the basic structure and the operation modes of the system are defined. The result is called principle solution. It describes not only the physical, but also the logical operating characteristics. For this purpose a dedicated specification technique was developed within the CRC 614 0. The description of the principle solution is divided into several aspects. The most important and for the integration of cognitive functions relevant ones are illustrated in figure 3.

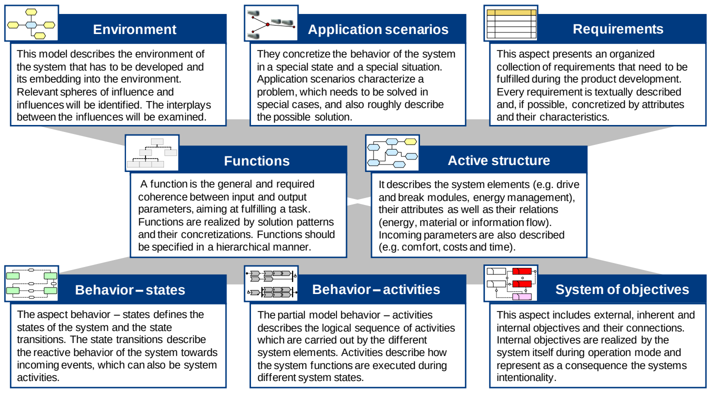

Figure 3. Short description of the used specification technique [2].

To integrate established solutions of all involved disciplines, we develop a uniform specification of solution patterns consisting of six aspects. A pattern describes a recurring problem and the core of its solution [9]. The aspect **characteristics** describe the properties of the pattern. They allow inferences on which requirements the pattern can meet. Examples of the characteristics are the processing speed and the type of calculation. The aspect **functions** list and describe the functions which can be implemented. Thus, this aspect expresses the problem description. The aspect **active structure** is the core of the solution. It specifies which system elements are necessary in order to implement the functionality and how those system elements are interrelated. With the aspect **behavior**, the description of the solution is completed. For this purpose, the behavior of the system elements and logical groups of several system elements is described. In **solution principles** relevant methods (e.g. Lorentz force or learning algorithm) are saved. C**ontext** specifies applications, in which the solution pattern was successfully implemented.

3.2. Methodology

The procedure model of the design framework (figure 4) gives an overview of the phases and tasks that have to be carried out to specify the cognitive information processing during the conceptual design. The design framework distinguishes four essential phases.

Systems analysis (1): At first an environment model is getting specified to identify the need for cognitive functions and respective requirements. According the HES, its structure offers a degree of freedom for the distribution of the power flows to the two storage devices. The main systems objectives are: minimize the degradation of energy, minimize the battery deterioration and maximize the power reserve. Important influences from the surroundings are the temperature of the battery, the state of charge of both storage devices and power prediction (incl. its quality).

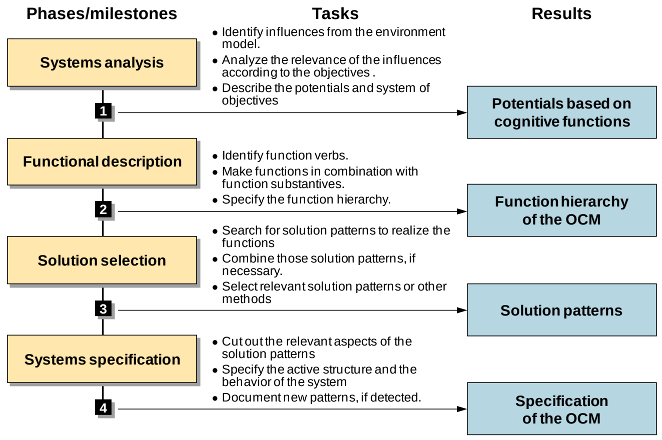

Figure 4. Procedure model of the design framework.

Functional description (2): In order to realize the s.o. operating strategy, it is necessary to concretize it with a functional description. For this purpose, we additionally develop design templates and a function catalogue to describe the information processing of s.o. systems in form of a hierarchy.

Solution selection (3): The function hierarchy is the basis for the solution selection. For the implementation of the functions of the HES, two solution patterns can be used. The solution pattern “Multi-Objective-Optimization” is based on continuous optimization methods and the solution pattern “Intelligent Preview” relies on discrete optimization methods. Both solution patterns were derived from other s.o. systems in the past.

Systems specification (4): The selected solution patterns are used to specify the information processing. Figure 10 visualizes a cut-out and simplified presentation of the specified HES. The *active structure* covers the description and arrangement of the system elements to realize a self-optimizing operating strategy. There are logical groups (cognitive operator, reflective operator and controller), which specify the self-optimization information processing architecture. The “cognitive cooperator” of the HES consists of two different types of optimizers (continuous and discrete) according to the selected solution patterns 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 and also a database.

The *behavior – activities* describe how the self-optimization is executed. Only a very few activities are shown here. In the case that a power profile is contained in the database, the continuous optimization element is active and the upper series of activities is carried out. If there is no profile, the discrete online optimization is performed by the system element “Discrete MOO” (lower series of activities). In contrast *behavior – states* describe states and state transitions of the system elements and due to which events those states change. In figure 5 the *behavior – states* describe the interaction between both optimizations.

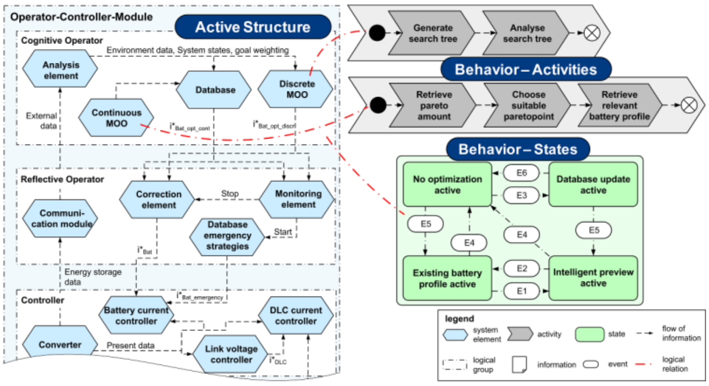

Figure 5. Specification of the information processing of the HES (cut-out and simplified illustration).

3.3. Application

The cognitive functions including self-optimization and prediction of future power demands are implemented in soft real-time on a standard personal computer. Pre-calculated operating strategies can be stored and retrieved, new strategies via continuous optimizer can only be calculated offline. The computation time of the discrete optimization typically ranges from 25 to 50 seconds and is performed online before driving the respective track sections. As in-time results cannot be absolutely guaranteed for demanding power profiles, in case of failure emergency strategies ensure safe operation until valid optimization results are available. The self-optimization process has been tested on a detailed loss model of the energy storages including the necessary power electronics as well as on the test-rig hardware. Compared to an operating strategy optimized based on a static system of objectives, with self-optimization the energy losses of the HES have been reduced by up to 13 % on the one hand or the availability of peak power has been increased significantly on the other hand, considering the system surroundings and the corresponding relevance of the objectives during run-time.

Acknowledgements

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) under the grant number SFB 614.

References

[1] Strube, G., 1998, Modelling Motivation and Action Control in Cognitive Systems, in: Schmid, U. et al., ed., Mind Modelling, Pabst, Berlin.

[2] Gausemeier, J., Frank, U., Donoth, J., and Kahl, S., 2009, Specification Technique for the Description of Self-Optimizing Mechatronic Systems” Research in Engineering Design, 20/4:201-223.

[3] Website of the CRC 614: http:\\www.sfb614.de/en/sfb614.

[4] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.-H., 2007, Engineering Design - A Systematic Approach, 3rd ed. Springer Verlag, London.

[5] Verein Deutscher Ingenieure (VDI), 2004, VDI-Guideline 2206 – Design methodology for mechatronics systems, Beuth Verlag, Berlin.

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

[7] Vernon, D., Metta, G. and Sandini, G., 2007, A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents, IEEE Transactions on Evolutionary Computation, 11/2:151-180.

[8] Lenzen, M., 2002, Natürliche und künstliche Intelligenz – Einführung in die Kognitionswissenschaft, Campus Verlag, Frankfurt am Main.

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