ON LEARNING FROM PAST EXPERIENCE AS A META-METHODOLOGY FOR THE APPLICATION OF SELF-OPTIMIZING WORKING PRINCIPLES WITHIN COMPLEX HIERARCHICAL MECHATRONIC SYSTEMS

Peter Scheideler and Andreas Schmidt
Heinz Nixdorf Institute, University of Paderborn
http://wwwhni.upb.de
Fürstenallee 11, D-33102 Paderborn, Germany
{Peter.Scheideler,Andreas.Schmidt}@hni.upb.de

ABSTRACT

Modern mechatronic products make use of the close interaction between mechanics, electronics, control engineering and software. As those systems increase in complexity and the interaction with its environment demands autonomous behavior, the need for inherent intelligence becomes ever urgent. Based on a framework for self-optimizing mechatronic systems, this paper introduces *Working Principles of Self-Optimization* as intelligent building blocks for self-optimizing module-agents. As a consequence of changing influences on the technical system, the working principles allow for an endogenous modification of the module-agents' multi-objective system and for an autonomous adaptation of its parameterization, behavior and structure. Knowledge bases of working principles together with an extended process model for learning from past experience enables the mechatronic agents to apply stored working principles to current situations and to learn from the outcome of their execution. The proposed approach is verified within a railway application scenario. The linear-motor drive of shuttle trains is self-optimized by the working principle of *Preview Control*.

Keywords: Hybrid System Applications; Case-Based Reasoning; Intelligent Agents

1 INTRODUCTION

Mechatronic systems comprise the domains of mechanical and electrical engineering as well as control systems and software. About forty researchers from various disciplines such as mathematics, computer science, mechanical and electrical engineering investigate methods and tools for the development of self-optimizing systems within the "Collaborative Research Center 614 – Self-Optimizing Concepts and Structures in Mechanical Engineering" [SFB614-ol] set-up at the University of Paderborn, Germany. The "New Railway-Technology Paderborn" project [NBP-ol] serves as a demonstration object for self-optimizing mechatronic systems.

This paper introduces working principles of self-optimization as building blocks for intelligent self-optimizing systems. Functional agent-modules learn from past experience by applying working principles stored in the agents' knowledge base to current situations. By communicating with other agent-modules and by revising own behavioral adaptation strategies, the agent-modules optimize themselves with time passing. The remainder of this paper is organized as follows: The second chapter introduces the architectural framework of self-optimizing mechatronic systems. Derived from the proposed functional module-hierarchy, the third chapter elaborates on the novel idea of working principles of self-optimization. The fourth chapter incorporates the working principles of self-optimization into knowledge bases of agent-modules together with a process-model based on an extended case-based reasoning approach. The fifth chapter exemplifies the usage of the process-model and the knowledge base of working principles with the help of an application scenario – the preview control of shuttles. The last chapter gives a short summary of the findings and an outlook on future work.

2 SELF-OPTIMIZING MECHATRONIC SYSTEMS

In order to cope with the inherent complexity, large mechatronic systems are usually decomposed into functional units which create a multi-hierarchical system of superordinated and subordinated modules [Gau02]. [SFB-ol] defines self-optimizing technical systems as follows: "Self-optimization of a technical system refers to the endogenous modification of the target vector due to changing environmental conditions and the resulting target-compliant, autonomous adaptation of the structure, the

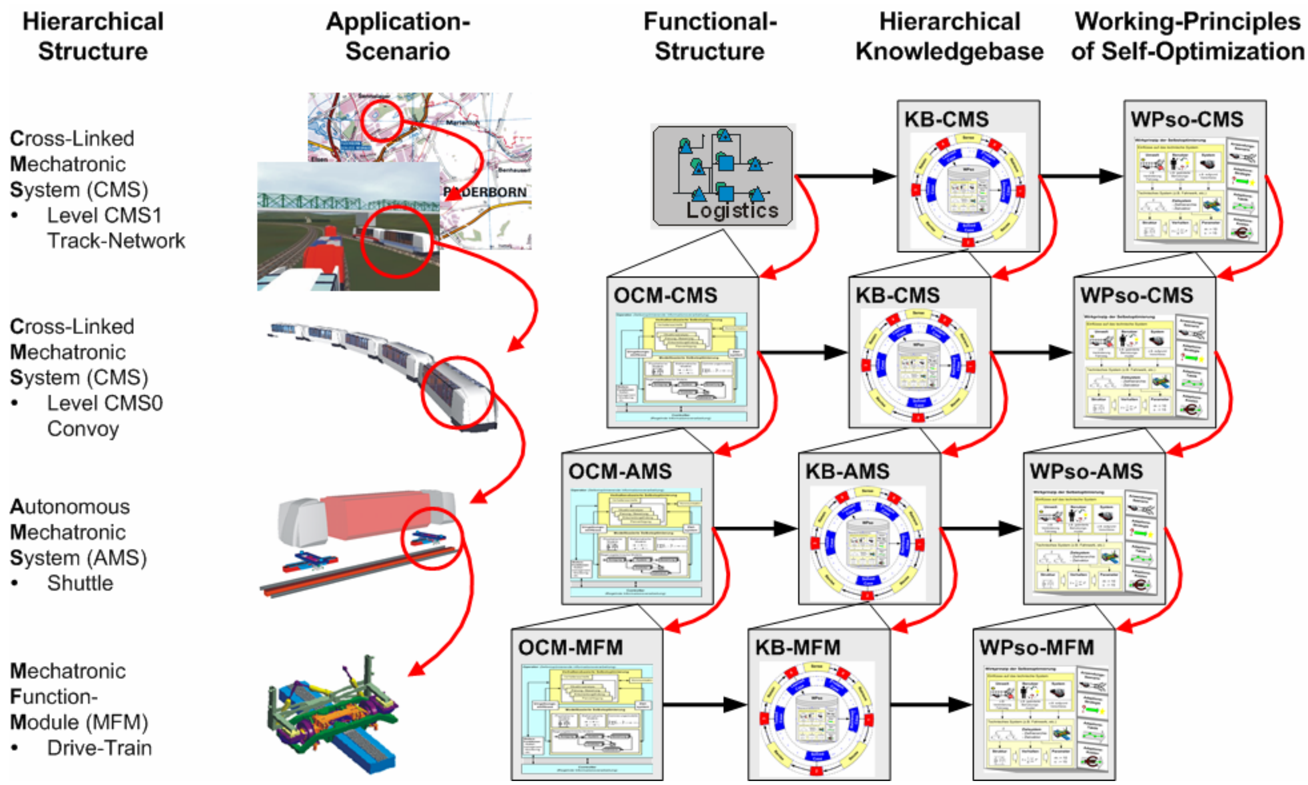

Figure 1: Hierarchical Mechatronic System

behavior and the parameters of this system. Self-optimization therefore far exceeds known control and adaptation strategies; self-optimization enables empowered systems with inherent intelligence which is able to react autonomously and flexibly to changing environmental conditions." The approach deployed by the Collaborative Research Center is based on a hierarchical structure of cooperating and communicating module-agents on three main-levels, see figure 1. The *Mechatronic Function Module* (MFM), e.g. a drive-train, represents the lowest-level with assembly groups consisting of sensors, actuators, and other subassembly-groups. The *Autonomous Mechatronic System* (AMS), e.g. a train shuttle, is made up of multiple MFM's and is characterized by its autonomous behavior which allows for interaction with the environment and reaction to external influences. The *Cross-Linked Mechatronic System* (CMS), e.g. a convoy of some shuttles or a track-network, constitutes the highest level of the hierarchical structure. The CMS consists of AMS's that may not physically be connected any more but cross-linked by exchange of information only. The lower the level the more strict the real-time constraints and the less intelligent – in the sense of simple reactive behaviors – the modules are. The higher the level the more deliberative and autonomous intelligent behavior the modules show. So-called *Operator Controller Modules* (OCM) realize the functional structure of the mechatronic system on each level. The functionality of OCM's is depicted in the subsequent paragraph. Each module-agent owns a *Knowledge-Base* (KB) of past experienced *Working Principles of Self-Optimization* (WPso). Facing a new situation the agent selects the most similar and promising working principle experienced in the past and adapts it to the current needs. By evaluating the outcome of the employed working principle, new behavioral patterns can be learned and the knowledge base can be enhanced. The methodology of using past experience to solve current requirements is also known as *Case-Based Reasoning* (CBR) [Ber02] and will be addressed in chapter four.

Operator-Controller-Modules (OCM) control the behavior of mechatronic systems. The OCM consists of so-called Operators and Controllers, see figure 2. The Operator incorporates the intelligent self-optimizing information processing unit of the mechatronic system. Considering the influences from the environment as well as a multi-objective system provided externally by a user or another OCM via communication, the *Behavior-based Self-Optimization* determines the next optimum action on a behavioral model. The optimization process cycles through a *Loop of Behavior* where it analysis the current situation, plans the next steps and evaluates the possible outcomes, decides on the optimal action and executes the calculated plan. The *Model-Based Self-Optimization* incorporates physical, mathematical and optimization models of the mechatronic system to decide on one of various possible control system models to be employed.

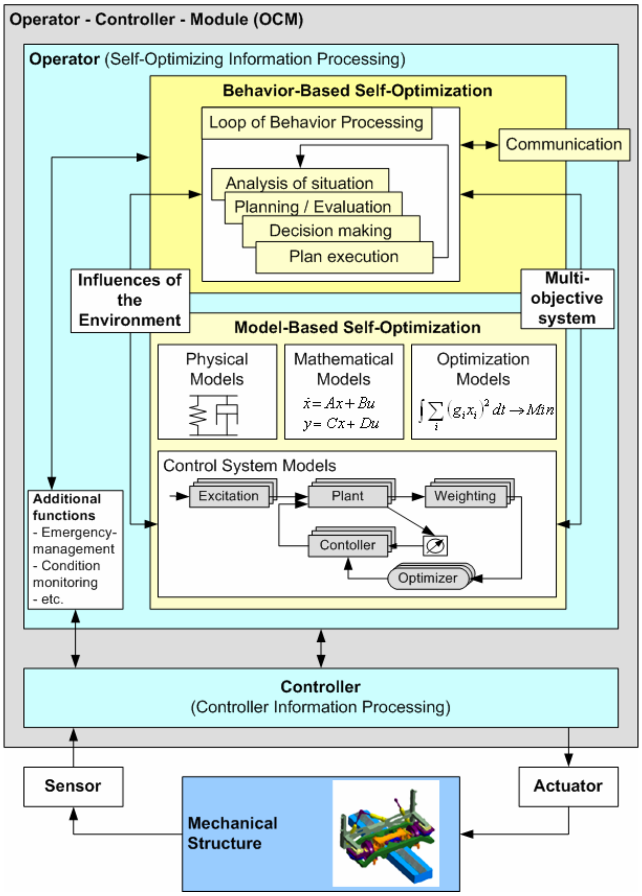

Figure 2: Operator-Controller-Module

By applying one of those models, the Controller part of the OCM directly impacts the respective mechatronic structure via actuators and sensors. [Kle+02] elaborates in more detail on the OCM and the design of self-optimizing agent-based controllers based upon OCM's. [SS03] presented an approach for an ontology for the above mentioned decentralized intelligent self-optimizing mechatronic system at the CMS level. The proposed ontology will be used for terminological references in the remainder of this paper.

3 WORKING PRINCIPLES OF SELF-OPTIMIZATION

Working Principles of Self-Optimization (WPso) form the building blocks of intelligent mechatronic systems. [SFB-ol] defines working principles of self-optimization as a combination of the application scenario of the working principle (e.g. driving on a track or crossing a switch) together with a technical system (e.g. drive-train) as well as the influences on the technical system from the environment, users or other system elements and adaptation-components as depicted in figure 3. The technical system constitutes of a structural model (e.g. topology of mechanical components, sensors and actuators, or physical models of the mechatronic system) as well as behavioral models (e.g. differential equation systems) and their parameterization. A multi-objective system prescribes necessary constraints imposed from subordinate systems and goals given by superordinated systems of the technical system at hand. The working principle of self-optimization allows for the endogenous modification of the multi-objective system based on changing influences on the technical system, as well as for the multi-objective-compliant, autonomous adaptation of parameters, behavior and structure. Adaptation strategies and tactics define the kind and process of modifications whereas adaptation costs represent the effort of adaptation in terms of e.g. energy-consumption, time-delays, or monetary payments.

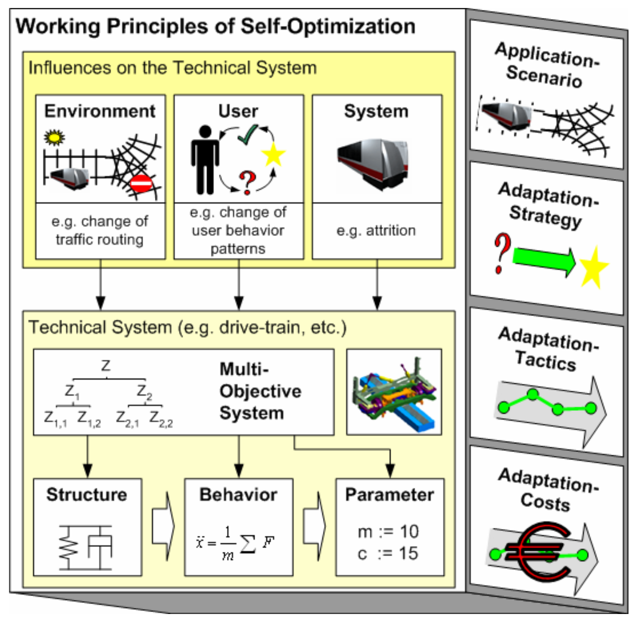

Figure 3: Working Principles of Self-Optimization

Altogether the working principles reflect a structure of detailed or generalized behavioral patterns for a mechatronic system which may be used by the Operator module to guide the behavior of the OCM in a specific situation [GS03]. Typical examples for low-level working principles may be *Hierarchical Clustering* and *Focus-Search-Group* [MA02] to build-up internal world-models of the environment and *Reinforcement-Learning* [KLM96] as a method for the OCM-agent to explore unknown state-spaces via stimulus-response schemata. The subsequent chapter on the application scenario of driving on a track presents the higher level working principle of *Preview-Control* by *Pareto-Optimization* [Del+03] at the autonomous mechatronic systems (AMS) level. Future work will deal with working principles at the level of cross-linked mechatronic systems such as *Auction- or Negotation-*

based Crossing of Switches by the autonomous coordination of multiple shuttles. For example [Ger03] propose a so called *negotiation-graph* that represents the characteristics of the trading-agents or [MP99] describe a multi-agent bidding mechanism based on a cooperative Case-Based Reasoning (*CoopCBR*) model. Both the negotiation graph and the CoopCBR approach may be interpreted as working principles of self-optimization within the application scenario of auction- and negotiation-based crossing of switches.

4 KNOWLEDGE BASE OF WORKING PRINCIPLES

Each OCM-agent owns an individual knowledge-base KB of working principles of self-optimization that represent past experiences. The OCM-agent employs a working principle in a specific application scenario. When facing a new situation, the OCM agent selects the most similar historic application scenario from the knowledge-base and adapts the respective working principle to the new situation. After having employed the working principle, the OCM agent evaluates the outcome and stores the new experience if necessary. The methodology of using past experience to solve current problems is known as *Experience Management* [Ber02] or *Case-Based Reasoning* (CBR). [AP94] describe the process from retrieving a situation or problem case up to the stage of storing a learned case with the help of a Case-Based Reasoning Cycle. Figure 4 depicts an adapted version of the CBR-cycle as a novel process model which controls the application of working principles within the scope of knowledge bases of mechatronic systems. A set of n sensors s_i (i=1, …, n) deliver a sensor input vector $\mathbf{i} = (\mathbf{i}_1, \mathbf{i}_2, \ldots, \mathbf{i}_n)$ in the *Sense* step. The sensor vector **i** is decomposed into an environmental vector **e**, a vector for the user demands **u**, and a vector standing for superordinated systems **s** enhanced by an externally given multi-objective system **z**. The tupel $\mathbf{C_{PC} = (e, u, s, z)}$ represent the influences on the technical system and make up the initial problem case C_{PC}. A similarity vector function $\mathbf{sim: C_{PC} \times KB \rightarrow [0;1]}$ calculates the weighted similarity value of all working principle cases

> $C_{wp} \in$ KB = {(C_{HPC},C_{HSC}) | C_{HPC}=historical problem case; C_{HCS} = historical solution cases}

that are part of the agent's knowledge base KB with respect to the problem case C_{PC}. The similarity values range from 0 (not similar) to 1 (full similarity). The *Retrieve* step returns a ranked list of similarity-judged cases where only the most similar case $\mathbf{C_{Sim} = (C_{SimPC}, C_{SimSC})}$ is used. The solution case tupel $\mathbf{C_{SimSC} = (S, B, P, A_S, A_T, A_C)}$ includes the solution of the historical most similar case C_{Sim} as vector-elements of *Structure* S, *Behavior* B, *Parameters* P, *Adaptation-Strategy* A_S, *Adaptation-Tactics* A_T, and *Adaptation-Costs* A_C. With the help of the adaptation vector function $\mathbf{adapt: C_{PC} \times C_{SimPC} \times C_{SimSC} \rightarrow C_{SC}}$ the *Reuse* step adapts the historical solution C_{SimSC} according to the functional differences between the initial problem case C_{PC} and the historical solution case C_{SimSC} and thus returns the solved case C_{SC}. The *Revise* step applies the solved case by uploading the working principle into the *Operator* of the respective OCM agent. After execution of the working principle a new tested case C_{TC} is constructed from the outcome of the executed solved case C_{SC}.

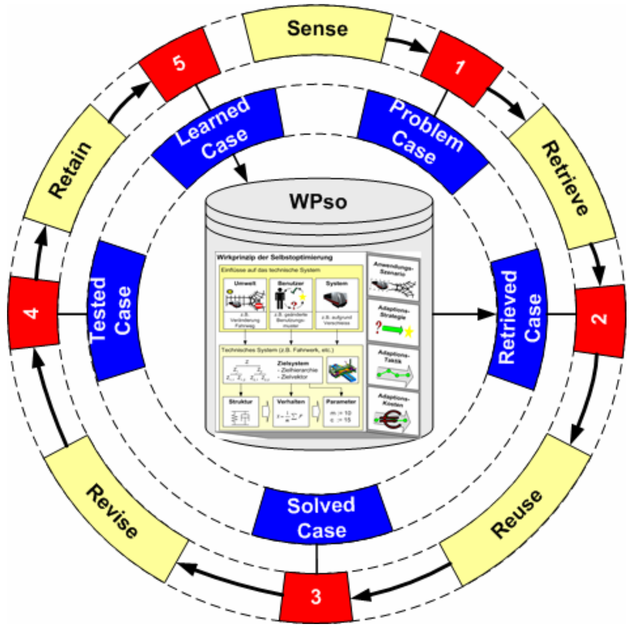

Figure 4: CBR-Cycle (adapted from [AP94])

The real outcome of the executed working principle versus its expected outcome is evaluated by the function $\mathbf{judge: C_{SC} \times C_{TC} \rightarrow A_{CTC}}$ and leads to adaptation-costs A_{CTC} of the tested case C_{TC}. The *Retain* step decides on whether the tested case shall be added to the knowledge-base as a new learned case C_{LC}.

The experience management setting proposed in this paper differs from traditional approaches in many respects. Traditional approaches of CBR in mechatronics are typically used within a single domain only. For example, [SV98] formulates and processes design knowledge in fluidics by the use of CBR. [SKC01] utilizes hierarchical case-based reasoning and decompositional problem-solving techniques for plant-control software design. Our work not only proposes working principles of self-optimization as connecting elements to cover all mechatronic domains – mechanical and electrical engineering as well as control systems and software. But also we put forward the CBR method as a fully automatically running coordination and communication process among all OCM agents on all levels of the hierarchical mechatronic system framework – CMS, AMS and MFM.

5 APPLICATION SCENARIO

This chapter exemplifies the use of the OCM-agents' knowledge base of working principles of self-optimization with the help of a real-world application scenario – the optimized driving of a shuttle on a track. Figure 5 depicts the application scenario. For a detailed technical description of the shuttle system see [NBP-ol]. The track is divided up into several track sections. Each track section is controlled by a specific track control which controls the track current (phase and amplitude) that builds up a magnetic field on the track section. The track sections communicate with each other to generate a continuous magnetic wave especially at the connecting elements of neighboring track sections. The track section agents may also communicate with other actors of the scene, e.g. the shuttle or the station. The goal system of the track-control mainly includes the setting of an optimal track-current which allows for a minimum attrition of the track when shuttles cross the track. The shuttle accelerates and decelerates on a track by generating an own magnetic field which is functional dependent on the magnetic field of the track. Let the multi-objective system of the shuttle on the AMS level consist of 1) the desired manoever – driving single which leads to more freedom of choice regarding velocity etc. or driving in a convoy of shuttles which is energy saving – 2) the desired energy management and 3) the desired safety.

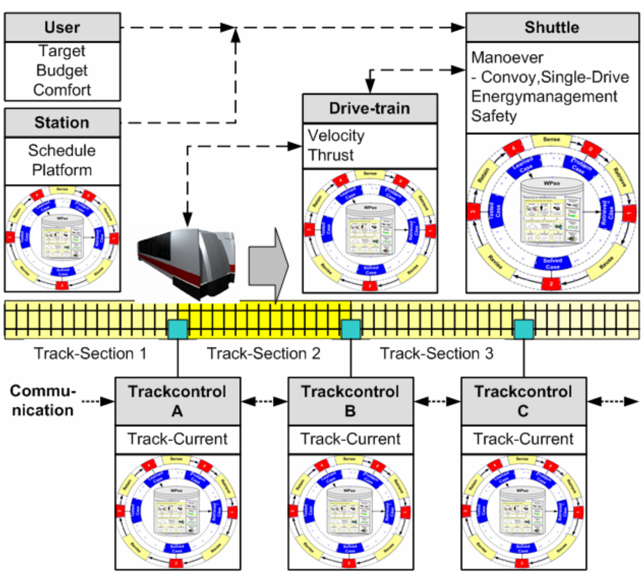

Figure 5: Application Scenario – Preview Control

On a lower hierarchical level – the MFM level – the shuttle constitutes of the drive-train module. The objectives of the drive-train include parameters such as velocity and thrust which are imposed by the shuttle objective system. Other agents that may influence the technical system can be identified as users driving in a shuttle and stations where users may enter and exit the shuttles. The typical multi-objective system of users consist of the target location the user wants to go to, the budget the user is willing to spend for the trip and the comfort the user wishes to enjoy. This example shows clearly that budget and comfort are contradictory goals which must be balanced. The station's objectives may include the optimum scheduling of its platforms. However, the station-agent is only depicted to show a complete picture of the whole scenario. It will be neglected for the further studies. Each OCM-agent, the track-control, the shuttle, the drive-train, the user, and the station, exhibit its own knowledge base of working principles of self-optimization. The agent-specific working principles control the specific behavior of each agent.

Figure 6 depicts the process of applying the knowledge base of working principles on the application scenario. Let the user u impose a certain level of comfort C on the shuttle. The system scenario reflects the current state of the shuttle and drive-train state-variables. In particular, let the system scenario s comprise the safety objective S of the shuttle. Let the shuttle sense its environment E leading to an environmental vector e. Using this information, step *1) Sense / Query* builds up the problem case C_{PC} = (e, u, s, z), where z := Z1 = (C, S). Let the most similar historical case of the shuttle's knowledge base be the working principle of self-optimization C_{Sim} = (C_{SimPC}, C_{SimSC}) consisting of the solution tupel C_{SimSC} = (S, B, P, A_S, A_T, A_C). The adaptation strategy A_S comprises a *Preview-Control* behavior. Preview control incorporates knowledge about the track sections ahead in order to modify the current and future behavior of the drive-train, see [HO03]. The shuttle actually driving from track section 1 to track section 2 may have communicated with track section 3 in order to receive information about the current situation on that segment. Other adaptation strategies such as reinforcement learning may favor the immediate information of the current situation when other information about upcoming segments is not available because of communication failure or the like. The adaptation tactics A_T = (P1,M1,O1,Z1) comprise information about the adaptation of the physical models P1, the mathematical models M1, the optimization model O1 and the multi-objective system Z1 of the selected drive-train working principle of self-optimization. Instead of the real shuttle model, a simpler vertical dynamics car model is used in the remainder of this paper (see [Ril02]). The simplified structure P1 of the physical model is depicted as a chassis P1.1 and a wheel P1.2. The parameterization of P1 consists of the mass m_A, the spring-constant c_A the damper constant d_A of the chassis as well as the mass m_R and the spring-constant c_R of the wheel. The movement z_A of the chassis, z_R of the wheel and z_S of the drive-train can be observed. The mathematical models M1 are derived from the physical models as differential equation systems. The optimization model O1 subsumes the optimization of 1) the comfort C and 2) the safety S objectives as minimizing the area of the squared 1) weighted movements of the chassis z_A and the ac-

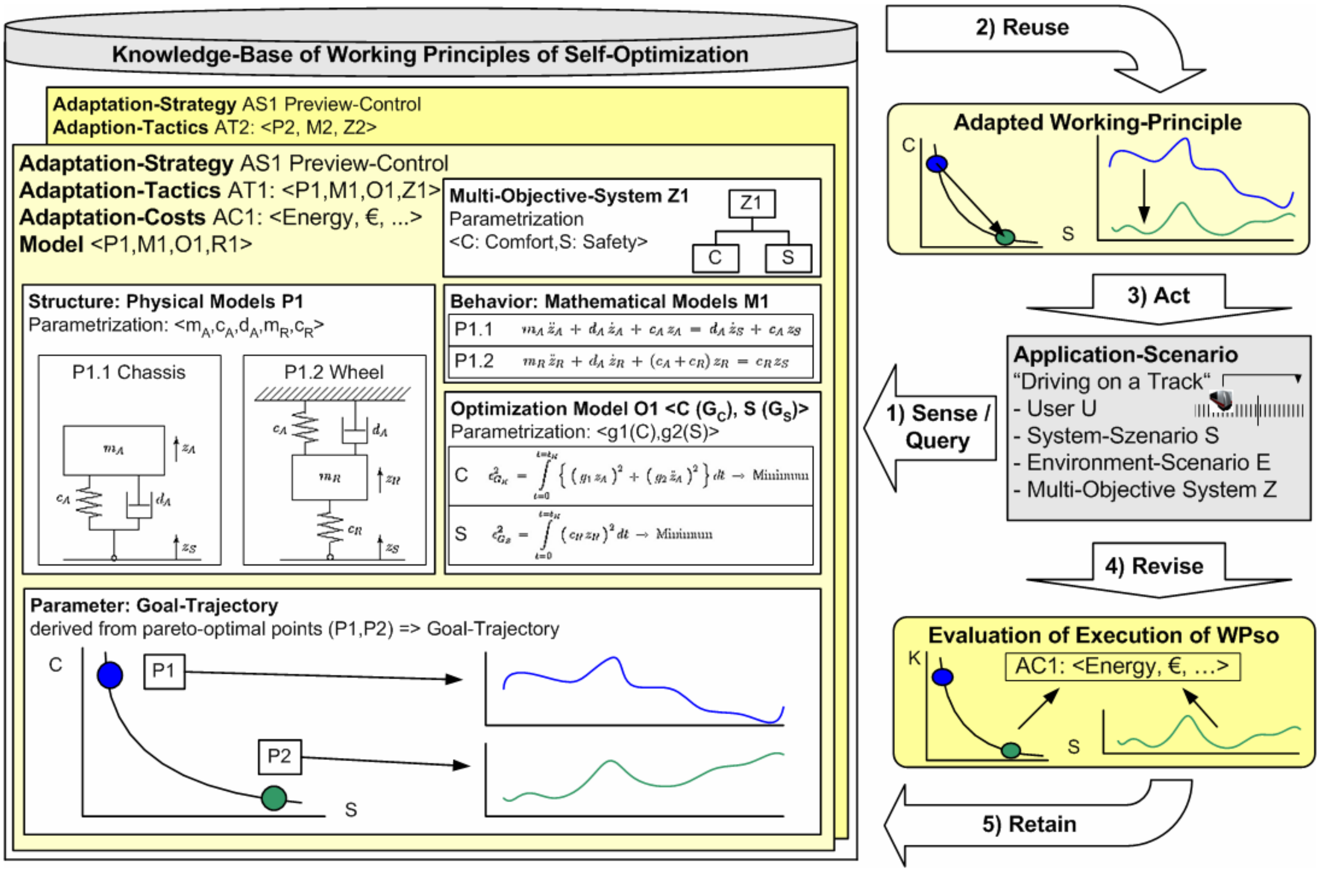

Figure 6: CBR-Process for applying Working Principles of Self-Optimization

celeration of the chassis $\ddot{z}_A$ and 2) the movement of the chassis z_R within a time-interval $[0;t_E]$. However, the multi-objective system Z1 prescribes a pareto-optimality of both objectives C and S. With the help of the evolutionary pareto-algorithm put forward in [Del+03] a pareto-curve of pareto-optimal states is calculated and conceptional depicted in the lower left part of figure 6. Let the blue P1 = (C1,S1) and the green P2 = (C2,S2) be two pareto-optimal states for comfort and safety settings. P1 and P2 lead to two goal-trajectories, blue and green, which control the behavior of the preview-control mechanism where P1 was used in the historical most similar solution case C_{SimSC}. The *Reuse* step adapts C_{SimSC} according to the current situation leading to the adapted working principle as a solved case C_{SC}. For example more safety was demanded which must be traded for less comfort. Within the mathematical and optimization model, more safety induces stronger spring-damper components which in turn reduce the comfort of driving. The adaptation to the current situation leads to discarding the pareto-optimal blue point P1 and using the green point P2 instead. P2 again involves the application of the green goal trajectory in favor of the blue one.

The drive-train acts according to the chosen and adapted working principle and leads to a real-world tested case C_{TC}. The *Revise* step evaluates the real outcome of this action versus the expected outcome presumed by the adapted working principle. Among others the value judgment leads to an update of the adaptation costs A_C, e.g. energy-consumption, monetary payments, and the adaptation tactics A_T. Using this information, the *Retain* step decides whether the so constructed learned case C_{LC} shall be stored in the knowledge base or not.

6 SUMMARY

This paper introduced working principles of self-optimization as basic building blocks of intelligent self-optimizing systems. It has been shown, how the working principles of self-optimization can be incorporated into the knowledge bases of functional-module-agents of self-optimizing systems on all levels – ranging from the Mechatronic Function Module (MFM) via the Autonomous Mechatronic Module (AMS) to Cross-Linked Mechatronic Systems (CMS). A process model was proposed that constitute an extension to the common case-based reasoning approach. This process model controlled the application of the working principles of self-optimization according to the principle of using past experience to solve current problems. An adaptation of the working principles and the incremental learning of the functional module agent by incremental storing new experiences in the knowledge base were presented.

Future work will include the extension of the idea of working principles of self-optimization on the cross-linked mechatronic system level, in particular the evolutionary and evolving behavior patterns of communicating and cooperating functional-module agents. Also the need of fuzzy representations of mechatronic systems and variables will drive the research into the fuzzification of the process model of retrieval, adaptation and learning of working principles. Last but not least, new application scenarios within the Collaborative Research Center 614 and the New Railway Technology Paderborn will require the investigation of new working principles of self-optimization.

ACKNOWLEDGEMENTS

This work was developed in the course of the Collaborative Research Center 614 – Self-Optimizing Concepts and Structures in Mechanical Engineering – University of Paderborn, and was published on its behalf and funded by the Deutsche Forschungsgemeinschaft (DFG).

REFERENCES

[AP94] Aamodt, A.; Plaza E.: Case-Based Reasoning: Foundational issues, methodological variations, and system approaches, AI Communications, Vol. 7, No. 1, pp. 39-59, 1994

[Ber02] Bergmann, R.: Experience Management – Foundations, Development Methodology, and Internet-Based Applications, Springer Lecture Notes in Artificial Intelligence 2432, 2002

[Del+03] Dellnitz, M.; Teich, J.; Schütze, O.; Mostaghim, S.: Covering Pareto-Sets by multilevel evolutionary subdivision techniques. In: C.M. Fonseca; P.J. Fleming; E. Zitzler; K. Deb and L. Thiele (eds.), Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, 2003

[Gau02] Gausemeier, J: From Mechatronics to Self-Optimization, 20th CAD-FEM Users Meeting 2002, International Congress on FEM Technology, Oct. 9-11, 2002, Friedrichshafen, Germany

[Ger03] Gerst, R. et al: Modellierung der praktischen Rolle in Verhandlungen mit einem erweiterten Verfahren des fallbasierten Schliessens, TUTS-WP-1-2003. Institut für Soziologie, Technical University of Berlin

[GS03] Gausemeier, J; Schmidt, A.: Wirkprinzipien der Selbstoptimierung, 14th Symposium „Design for X“, Oct. 13.-14., 2003, Erlangen, Germany

[HO03] Hestermeyer, T.; Oberschelp, O.: Modeling, Analysis, and Optimization of Time-discrete Preview-Strategies by Superimposition of Elementary Functions on Active Driving Modules, to be published

[Kle+02] Kleinjohann, B.; Kleinjohann, L.; Oberschelp, O.; Hestermeyer, T.: Design of Self-Optimizing Agent-Based Controllers, CfP Workshop 2002 – Agent-Based Simulation, April 7-9, 2002, Passau, Germany

[KLM96] Kaelbling, L.P.; Littman, M.L.; Moore, A.W.: Reinforcement Learning: A Survey, In: Journal of Artificial Intelligence Research, Vol. 4, pp. 237-285, 1996

[MA02] Meystel, A. M., Albus, J. S.: Intelligent Systems – Architecture, Design, and Control, Wiley Series on Intelligent Systems, 2002

[MP99] Martin, F.J.; Plaza, E.: Auction-Based Retrieval, In: Proceedings of the 2nd Congrès Català d'Itelligència Artificial (CCIA99), Girona, Spain, pp. 136-145, Oct. 25-27, 1999

[NBP-ol] NBP-online: Homepage of the "New Railway Technology Paderborn", http://nbp-www.upb.de, 2003

[Ril02] Rill, R.: Vehicle Dynamics, Lecture Notes of the University of Applied Sciences Regensburg, Germany, 2002, http://homepages.fh-regensburg.de

[SFB-ol] SFB 614-online: Homepage of the Collaborative Research Center 614 „Self-Optimizing Concepts and Structures in Mechanical Engineering", http://www.sfb614.de, 2003

[SKC01] Smyth, B.; Keane, M.T.; Cunningham, P.: Hierarchical Case-Based Reasoning Integrating Case-Based and Decompositional Problem-Solving Techniques for Plant-Control Software Design, In: IEEE Transactions on Knowledge and Data Engineering, Vol. 13, No. 5, pp. 793-812, Sept. / Oct. 2001

[SS03] Scheideler, P., Schmidt, A.: Entwicklung einer Modellwelt für ein dezentrales, intelligentes, mechatronisches System, in: Intelligente mechatronische Systeme, HNI-Verlagsschriftenreihe Band 122, pp. 81-91, 2003

[SV98] Stein, B.; Vier, E.: An Approach to Formulate and to Process Design Knowledge in Fluidics, In: Mastorakis, N.E. (ed.) Recent Advances in Information Science and Technology, pp. 237-242, Oct. 1998, World Scientific Publishing Co. Pte. Ltd