COORDINATION OF SELF-OPTIMIZING MECHATRONIC SYSTEMS

A New Application for Multi-Agent Planning

Benjamin Klöpper and Wilhelm Dangelmaier
Heinz Nixdorf Instiute, University of Paderborn, Fürstenallee 11, Paderborn, Germany

Keywords: Self-optimization, Mechatronics, Multi-agent systems, Planning and coordination.

Abstract: The paradigm of self-optimization introduces flexible and highly adaptive mechatronic systems. During the exploiation of this flexibility, new problems arise. One of these problems is the coordination of mechatronics systems and subsystems. This paper introduces the application area *self-optimizing mechatronic systems* and identifies the arising coordination problems. Two main scenarios are identified: coordination of autonomous mechatronic systems and coordination of several subsystems within an autonomous mechatronic system. We will show that multi-agent technology and in particular multi-agent planning can be applied to solve both coordination scenarios.

1 INTRODUCTION

Technical systems and machines are designed to fulfill tasks for humans. Technological progress continuously extends the spectrum of tasks and improves the quality of tasks fulfillment. The quality of tasks fullfillment can be measured in various dimensions, depending on the current area of application. Examples for such dimensions are: timeliness, resource consumption, processing accuracy (e.g. in case of machining tools), or comfort and driving pleasure (in case of vehicles). Mechatronic systems are a relatively new class of technical systems. The term mechatronics refers to the close integration of electromechanical systems, electronic and information technology (Bradley, 1997).

Further advancement in information and communication technology enables systems to adapt their behavior to changing environmental settings and user preferences. We use the term self-optimization to characterize such systems. Self-optimizing systems (Gausemeier et al., 2008) are able to adapt their objectives autonomously. This includes modifying the relative weighting or ranking of the objectives. Adapting the objectives results in an adaptation of the system behavior. To determine the suitable adaption of the system behavior, the objectives are used to formulate corresponding optimization problems. Their solutions problems indicate the suitable behavior adaptations. The adaptations are realized by adapting parameters (e.g. changing a control parameter) or the structure of the system (e.g. replacing the current controller).

We understand self-optimization as an iterative sequence of three actions:

1. Situation-Analysis – includes the state of the system and observations about its environment.
2. Determining the system of objectives.
3. Adapting the system behavior according to the new objectives.

In this paper we will explore two main theses regarding self-optimizing mechatronic systems:

1. Planning extends the steps of self-optimization to futures states and helps to implement autonomous determination of objectives
2. The structure of mechatronic systems requires multi-agent planning

The paper is structured as followed: First it introduces mechatronic systems and their structure in more details. Then, the following section explains how planning can be used to implement the determination of objectives. From classical planning we proceed to multi-agent planning and show that this discipline has great potential within the context of self-optimizing mechatronic systems. Finally we will conclude with example that illustrates this new application area.

Klöpper B. and Dangelmaier W. (2009).
COORDINATION OF SELF-OPTIMIZING MECHATRONIC SYSTEMS - A New Application for Multi-Agent Planning.
In *Proceedings of the International Conference on Agents and Artificial Intelligence*, pages 312-317
DOI: 10.5220/0001794903120317

2 STRUCTURING OF MECHATRONIC SYSTEMS

Figure 1 shows an proposal for a hierarchical structuring for mechatronic systems ((VDI), 2004). *Mechatronic Function Modules* (MFM) are on the lowest level of the hierarchy. MFM are the elementary type of mechatronic systems and consist of a mechanic structure, actuating elements and sensors. The sensors provide information about the environment and the system for a superimposed information processing, which controls the actuating elements. On the next level *autonomous mechatronic systems* (AMS) integrate several function modules. AMS and MFM interact by exchange of information. Finally, several AMS can establish networked mechatronic systems (NMS). The structuring shows versatile relationships between mechatronic systems by information, energy and material flow. These relationships and interactions require coordination.

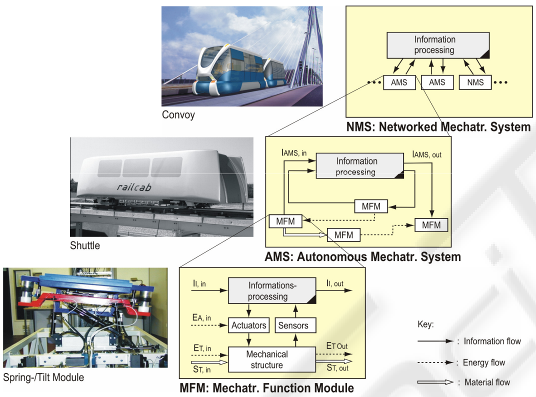

Figure 1: Structure of a complex mechatronic system.

3 PLANNING FOR SELF-OPTIMIZATION

Planning in *Artificial Intelligence* refers to the task of finding a sequence of actions which lead from a given initial state to a desired goal state (Russell and Norvig, 2003). Planning languages are used to formalize this problem and enable the algorithmic solution. Planning languages usually consist of two main elements: state representation and action representation. While the states represent snapshots of the world, actions define the dynamic component of the planning problem. An action defines which activities are applicable in a given state (via precondition) and which consequences arise from this activity (via effects). During the last years, the *Planning Domain Definition Language* (PDDL) (Gerevini and Long, 2005) established a de facto standard in academic research.

In the context of self-optimization *planning* can be interpreted in a different way. For each action planning locally determines (on a time line) the objectives. Since the planning procedure builds a complete sequence of activitivies, this local system of objectives is coherent with future steps and situations in the plan. Thus, planning is a powerful mechanism in the context of self-optimization.

To use this powerful mechanism, the planning models from artificial intelligence must be mapped onto the abilities and behaviors of mechatronic systems. Thus, the next section introduces a classical methodology for the design of mechatronic systems. Subsequently, the central modeling elements of the methodology are mapped onto the planning of artificial intelligence.

3.1 Classic Design of Mechatronic Systems

A technical system fulfills its tasks by the implementation of a behavior. During the design of a mechatronic system the behavior is usually modeled as a black box. The transformation of incoming information, energy and material flow into outgoing flows is focussed. The system's designer defines a function hierarchy and splits the main tasks into sub-functions until a sufficient amount of details is achieved (Pahl and Beitz, 07). According to the definition of the function hierarchy, the subfunctions are examined for relationships, dependencies and conflicts. The designer searches for a feasible sequence of subfunctions to implement the desired behavior. This sequence may contain loops and branches. Finally, the designer identifies possible implementations and solutions of the subfunctions.

3.2 Planning Models for Self-Optimizing Mechatronic Systems

Self-optimizing mechatronic systems are able to adapt their behavior to the current operation condition and varying external preferences (e.g. defined by a user or surrounding systems) regarding the possible quality dimensions of their behavior. For this purpose, these mechatronic systems feature several implementations of their sub-functions. We refer to these implementations as *operation modes*.

To achieve maximum flexibility and adaptability the mechatronic systems must be able to identify the sequence of operation modes that provides the best quality regarding the current external objective and the current environmental influences. This problem leads directly to the planning problem in artificial intelligence.

If the operation modes of a mechatronic system are mapped onto the action definition of a planning language, artificial intelligence planning can be applied to determine the currently best sequence of operation for a given overall task. An example for such a task is the transportation of a passenger from a start station to a destination station by a railway vehicle. The initial state is the current position of the vehicle, the goal state is the passenger at his destination. Possible quality dimension or *objectives* are travel time, comfort and energy consumption. The vehicle's active suspension may feature different combinations of comfort and energy consumption for certain environmental circumstances (e.g. track conditions). The following listing shows an action definition for a driving on track section with an active suspension:

```
(:action drive
 :parameters
(?v vehicle ?l1 ?l2 - location)

 :precondition
(and at ?r ?l1)
    (>=(energy_storage ?v)
            (*distance ?l1 ?l2)
            (suspension-active ?v)

 :effect
    (and (not (at ?v ?l1))
    (at r? ?l2)
    (increase (total-energy-used)
            (*(distance ?l1 ?l2)
            (suspension active ?v)))
    (decrease (energy_storage ?v)
            (*(distance ?l1 ?l2)
            (suspension-active *v)))
)
```

The precondition of the action assures that the vehicle has sufficient energy to travel between the two locations $l1$ and $l2$. In the effects the vehicle moves from its origin $l1$ to $l2$ and the state of charge of the energy storage is reduced by the energy consumption of travelling the distance between $l1$ and $l2$ with activated suspension.

Beside fundamentally different implementations of sub-functions (e.g. active and passive suspension) the activities can also differ just in certain parameters (e.g. operating pressures). To select appropriate parameter settings to define actions is difficult. Figure 2 shows a possible systematic approach to determine reasonable operation modes. The first steps in the definition of operation mode is the selection of characteristic environmental influences and a reasonable configuration of objectives (e.g. weighting of energy consumption vs. comfort). The environmental influences will be part of the precondition of the actions while the objectives configuration *label* the actions. An optimization (e.g. (Witting et al., 2008) is used to determine continuous parameters of the system behavior and to provide information on how good the objectives can be achieved under the characteristic environmental influences and with the predetermined weightings. Optimal control (Kirk, 1970) even offers a way to predetermine optimal continous state trajectories regarding a cost functions. During operations the expected environmental influences must be mapped to a characteristic influence from optimization to determine the effects of an action. This problem is a classification problem.

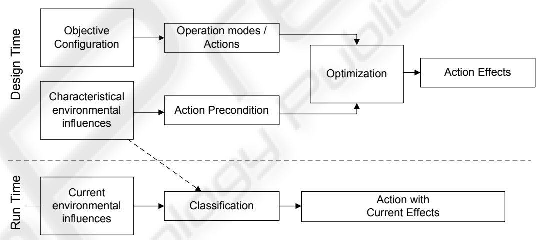

Figure 2: Action Definition by Offline Optimization.

4 MULTI-AGENT PLANNING FOR SELF-OPTIMIZATION

The structuring of mechatronic systems clearly shows that mechatronic systems do not act isolated in their environment. Instead, there are various interactions and interdependencies. Thus, the activities and the planning of mechatronic systems have to be coordinated. Witteveen et. al. define multi-agent planning as the combination of planning and coordination (Weerdt et al., 2005).

Multi-agent systems (MAS) and agent based systems are recognized as a new approach to the control and coordination of mechatronic systems (cf. (Baum et al., 2002; Al-Safi and Vyatkin, 2007)). Anyhow, multi-agent planning is rarely used in mechatronics today.

In the context of mechatronic systems two fundamentally different application scenarios can be distinguished:

1. Multi-agent planning for autonomous mechatronic systems
2. Multi-agent planning for function modules within an autonomous mechatronic systems

4.1 Multi-agent Planning for AMS

Each autonomous mechatronic system possesses an individual system of objectives. Thus, they act selfishly and cooperation is an instrument to improve their local plan quality. In the context of self-optimizing mechatronics we distinguish three types of cooperative actions. In *job swapping* an AMS transfers a job (e.g. a transport job or machining workpiece) to another AMS, which is more suitable. Reasons may be the abilities of the systems or their current local plans. For instance, a machining center offers better processing accuracy or a vehicle already passes the start and destination of a transport job. AMS may also *outsource operations*. This cooperation action requires extenisve coordination. The planning system has to embed an externally executed operation into its current local plan. Thus, timing is crucial. The third coordination activity is the *alignment of activities*. The alignment of activities changes the cost-benefit ratio of an activity. Mainly, the alignment of activities refers to a chronological rearrangement of activities or a change of the activity's subject (e.g. a workpiece). For example such rearrangements help to avoid set-up cost.

To realize these cooperation actions during the planning process, a phase model according the Wooldridge and Jennings (Wooldridge and Jennings, 1998) seems suitable: Recognize cooperation potential,team formation, plan formation, and team action.

To recognize possible cooperation action, these actions have to be integrated into the local planning problem of the AMS. In case of the *exchange of jobs* the integration is straight forward. The systems can compare the quality of a plan with and without a job. Data clustering can be used to select candidate jobs. Under the assumption that jobs which differ largely from the other job cause large costs, small clusters apart from the residual data set are good candidates. The external execution of sub-functions is just a special type of sub-function. They can be integrated with expected cost-benefit ration in the local planning process. Conditional actions can model the alignment of activities, with special effects for aligned and unaligned actions.

Multi-agent technology offers various methods to implement *team formation* and *plan formation*. Matchmaking mechanisms like middle agents (Sycara et al., 1997) or distributed matchmaking (Ogston and Vassiliadis, 2002) can identify possible cooperation partners. Negotiations (Faratin et al., 1998), auctions (David et al., 2002) and voting (Conitzer, 2008) can be used to provide an agreement about joined plans and compensations.

4.2 Multi-agent Planning for Function Modules

The coordination of function modules differs largely from the coordination of AMSs. The composition of the systems is already known, no method for the identification of cooperation partners is required. Furthermore all function modules are committed to the system of objectives of the AMS. Thus, the purpose of the coordination of function modules is to achieve a behavior that optimizes the system of objectives of the overall system.

For this problem, centralized planning could be a feasible approach. However, this approach has several disadvantages. First, the definition of complex centralized planning models is more time-consuming than the definition lean localized planning models. Secondly, the centralized approach is less flexible. Every change in the composition of the function modules requires a redesign of the planning models. Finally, the paradigm of *divide-and-conquer* offers better performance. In distributed planning, parallelization speeds-up the processing and splitting up a centralized planning problem into several smaller local problems reduces the complexity. For an example, we consider a system with two function modules. Each function module can carry out different modes of operations. In a decentralized formulation two state-action trees with 2^d nodes can be constructed. In a centralized 4^d state-action combinations have to be considered. The better performance of localized and distributed planning has to be balanced with the coordination efforts.

In (Klöpper et al., 2008a) we introduced a formal framework, which enables the function modules to consider the social consequences of their activities. The basic model element in this formal framework are *conditional evaluation functions*: $eval^{\omega}_{\Delta}(om_{fm}|om'_{fm'},e)$ defines how good an operation mode of function module fm achieves an objective ω under the environmental influences e if function module fm' carries out an operation mode om'. In (Klöpper et al., 2008b) we introduced a systematic approach to map the objectives of the AMS to the objectives of the function modules and thus provided (in combination with the formal framework) planning metrics for local planning processes.

We suggest a planning process that resembles the popular Generalized Partial Global Planning (GPGP) (Lesser et al., 2004). The *conditional evaluation functions* are equivalent to the relationships of GPGP. In difference to GPGP, the *conditional evaluation functions* are already considered during the local planning based on the expected behavior of other function modules. The seamless integration of the interaction into the local planning requires no special coordination strategies like in GPGP. The GPGP coordination strategies work only on single actions, while the *conditional evaluation functions* in combination with replanning and full knowledge of all local plans, enables simultaneous coordination of all actions within a plan horizon. The feasibility of each local plan can also be assured.

Another problem is to assure feasibility over all local problems. This problem occurs, if the function modules share some resources like energy. Here we suggest a central coordination agent which governs the shared resources. If the coordination agent detects a shortage, it asks the function module agents for a modification of their behavior. The *conditional evaluation functions* enable the comparison of the suggested behaviors. Thus, the coordination agent selects the behavior modifications, which have the smallest negative effect on the overall system performance.

5 APPLICATION SCENARIO

Figure 3 shows how the two multi-agent planning problems in the context of the RailCab-system can be coupled in a hierarchical process. The RailCab-system is an innovative railway system. Autonomous vehicles – RailCabs – fulfill transportation jobs with a demand oriented transport disposition. Thus, no fixed schedule exists. The RailCabs feature a number of innovative function modules: an active suspension, an air gap adjustment systems (AGAS) and a linear drive.

The process hierarchy follows the structuring of mechatronic systems. On the top level (networked mechatronics systems) the RailCabs perform the cooperative actions *transportation job swapping* and *convoy formation*. In the decentralized job swapping each vehicle constantly analyzes it's local plan. It searches for jobs that cause inappropriate costs, e.g. jobs that cause large detours for picking up and/or delivering a single passenger. This kind of job is advertised for bids of other vehicles. An implementation of the job swapping reduces the cost of track utilization by 15% in average during simulation experiments. Details about job swapping can be found in (Danne et al., 2006). The formation of convoys reduces energy costs for non-leading vehicles in the convoy. Assuming a cost reduction of 50% [1], a planning procedure reduces travelling costs from 15% up to 20% over all vehicles in the convoy, considering costs for necessary detours in order to join the convoy (Dürksen et al., 2008).

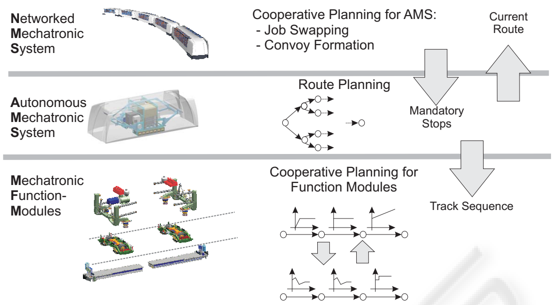

Figure 3: Hierarchical Planning Process in the RailCab Scenario.

The cooperative planning process for AMS creates a number of mandatory stops for each vehicle (to pick-up and disembark passengers). A conventional route planning process creates a route to connect the mandatory stops. The resulting sequence of tracks is input for the cooperative planning process for the function modules. Each function module locally constructs a plan for the activities on this track sequence with respect to the expected behavior of its fellow modules. An example of a planning process on this level is the linear drive, which has to select an acceleration profile for each track section. The propulsion module may take the effects on the active suspension (high velocity implies high energy consumption) into consideration. When each local plan is constructed, the *coordination after planning* takes places.

6 CONCLUSIONS

In this paper we introduced a new application area for multi-agent planning: self-optimizing mechatronic systems. The new application area was structured and first results of multi-agent planning in self-optimizing mechatronic systems were introduced. In our opinion the application of multi-agent technology to mechatronic systems enables the full potential of the paradigm *self-optimization*. Two different application scenarios for multi-agent planning were introduced and first results were presented. With the combination of self-optimization and multi-agent planning a new class of technical products with new levels of utility becomes possible.

[1] on perfect flat tracks the reduction is 80% due to slipstream effect

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). Further information can be found at www.sfb614.de/en.

REFERENCES

Al-Safi, Y. and Vyatkin, V. (2007). An ontology-based reconfiguration agent for intelligent mechatronic systems. In *Holonic and Multi-Agent Systems for Manufacturing*, number 4569 in Lecture Notes In Artificial Intelligence, Berlin. Springer Verlag.

Baum, W., Bredenfeld, A., Hans, M., Hertzberg, J., Ritter, A., Schonherr, F., Christaller, T., and Schraft, R. (2002). Integrating heterogeneous robot and software components by agent technology. In *Robotik 2002*, volume 1679 of *VDI-Berichte*. VDI Verlag GmbH.

Bradley, D. (1997). The what, why and how of mechatronics. *Engineering Science and Education Journal*, 6(2):81–88.

Conitzer, V. (2008). Comparing multiagent systems research in combinatorial auctions and voting. In *10th International Symposium on Artificial Intelligence and Mathematics (ISAIM-08)*.

Danne, C., Dück, V., and Klöpper, B. (2006). Selfish motivated cooperative planning in cross networked mechatronic system. *International Transactions on Systems Science and Applications*, 2(2):221–225.

David, E., Azoulay-Schwartz, R., and Kraus, S. (2002). Protocols and strategies for automated multi-attribute auctions. In *Proc. First Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 77–85.

Dürksen, D., Klöpper, B., Ruth, D., and Thonemann, C. (2008). Combining distributed matchmaking and clustering to prune the solution space in distributed optimization problems – demonstrated in the railcab system. In *Eighth International Conference on Hybrid Intelligent Systems*, pages 108–113,, Barcelona. IEEE Computer Society, IEEE Computer Society Press.

Faratin, P., Sierra, C., and Jennings, N. R. (1998). Negotiation decision functions for autonomous agents. *International Journal of Robotics and Autonomous Systems*, 24:3–4.

Gausemeier, J., Kahl, S., and Pook, S. (2008). From mechatronics to self-optimizing systems. In *7. Internationales Heinz Nixdorf Symposium*, volume 223 of *HNI-Verlagsschriftenreihe, Paderborn*. Heinz Nixdorf Institut, HNI Verlagsschriftenreihe, Paderborn.

Gerevini, A. and Long, D. (2005). Plan constraints and preferences in pddl3. Technical report, Department of Electronics for Automation, University of Brescia.

Kirk, D. E. (1970). *Optimal control theory*. Prentice-Hall, Englewood Cliffs, NJ.

Klöpper, B., Romaus, C., Schmidt, A., and Vöcking, H. (2008a). A multi-agent planning problem for the coordination of functions modules. In *Self-Optimizing Mechatronic Systems: Design the Future*.

Klöpper, B., Romaus, C., Schmidt, A., Vöcking, H., and Donoth, J. (2008b). Defining a system of objectives for multi-agent coordination of function modules within mechatronic systems. In *: Proceedings of IDETC/CIE 2008 ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, volume . uk, uk.

Lesser, V., Decker, K., Wagner, T., Carver, N., Garvey, A., Neimann, D., Prassad, M. N., Raja, A., Vincent, R., Xuan, P., and Zhang, X. (2004). Evolution of the gpgp/taems domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems*, 9:87–143.

Ogston, E. and Vassiliadis, S. (2002). Unstructured agent matchmaking: experiments in timing and fuzzy matching. In *SAC '02: Proceedings of the 2002 ACM symposium on Applied computing*, pages 300–305, New York, NY, USA. ACM.

Pahl, G. and Beitz, W. (07). *Engineering Design: A Systematic Approach*. Springer.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Sycara, K., Decker, K., and Williamson, M. (1997). Middle-agents for the internet. In *Proceedings of IJCAI-97*.

(VDI), V. D. I. (2004). Vdi-guideline 2206 - design methodology for mechatronic systems. Technical report, Verein Deutscher Ingenieure (VDI), Beuth-Verlag, Berlin.

Weerdt, M., Mors, A., and Witteveen, C. (2005). Multi-agent planning - an introduction to planning and coordination. Technical report, Dept. of Software Technology, Delft University of Technology.

Witting, K., Schulz, B., Dellnitz, M., Bcker, J., and Frhlecke, N. (2008). A new approach for online multiobjective optimization of mechatronic systems. *International Journal on Software Tools for Technology Transfer STTT*, 10:223–231.

Wooldridge, M. and Jennings, N. R. (1998). *Formalizing the cooperative problem solving process*, pages 430–440. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.