RESEARCH ARTICLE

Adaptation Strategy Planning for Collaborative Cyber-Physical Systems Using Goal Models

Marian Daun[1,a], Bastian Tenbergen[2,*], Torsten Bandyszak[3,b], Jennifer Brings[3,c], Linda Feeken[4,d]

[1]*Technical University of Applied Sciences Würzburg-Schweinfurt, Schweinfurt, Germany*

[2]*State University of New York at Oswego, Oswego, United States*

[3]*University of Duisburg-Essen, Essen, Germany*

[4]*German Aerospace Center (DLR), Oldenburg, Germany*

[a]*Email: marian.daun@thws.de*

[b]*Email: torsten.bandyszak@paluno.uni-due.de*

[c]*Email: jenniferbrings@gmail.com*

[d]*Email: linda.feeken@dlr.de*

ABSTRACT

Collaborative systems such as cyber-physical systems (CPS) dynamically form runtime networks to achieve goals that cannot be achieved by the individual system alone. While much research has been done on architectures for adaptive systems, these approaches do not consider the fact that goals exist on network level and thus need specific adaptation strategies. In decentralized settings where there is no central mechanism for orchestrating and coordinating the collaboration between systems, failure to adapt individual systems' behaviors without considering the network's overall outcome may result in a functional degradation of the entire network, possibly to the point of injuring humans and harming individual systems in the network. In this article, we propose a goal-based adaptation approach that takes the goals of the individual systems as well as the goals of the overall network into account. Our approach allows considering adaptation strategies based on the network goals which are fulfilled by automated behavioral adaptations of the individual system of the network. Results from a proof-of-concept application to an industrial case example show applicability and usefulness of the approach.

1. INTRODUCTION

One way to look at decentralized collaborative systems is to understand each individual system as participating in a collaborative network of interconnected, adaptive systems which closely interact at runtime [1,2,3]. Examples for such system collaboration are cyber-physical systems (CPS), which communicate with other CPS and render decisions based on remotely sensed data or the need to locally actuate servos, valves, or the like. Networks of CPS must be considered highly dynamic [4], as network nodes may join or leave the network, resulting in a plethora of possible network configurations the individual CPS might face at runtime [5]. For instance, autonomously driving vehicles equipped with a cooperative adaptive cruise control (CACC) can form platoons on the highway [6,7]. In a platoon, vehicles drive with close distances at a high speed, saving fuel due to wind shields and slipstreams [7], and optimize road usage for high throughput, reducing traffic jams and rear-end collisions [6].

Yet, since not all vehicles travel between the same origin and destination, vehicle nodes leave and join the network constantly. At runtime different numbers of vehicles equipped with CACCs from different manufacturers may join together and form a platoon, leading to many possible configurations of a platoon that need to be considered.

In previous work, we investigated the model-based specification of goal models for collaborative CPS forming CPS networks. We found goal modeling to be a very good approach to support requirements engineering and early analysis of collaborative CPS [8]. As collaborative CPS

---

*Corresponding author. Email: bastian.tenbergen@oswego.edu

operate in dynamic contexts, they need to be self-adaptive in such a way that they best fulfill their goals in different contexts. In addition, through collaboration of systems within the network, the network is able to achieve goals the individual systems cannot. As the network changes at runtime, this means that system behavior may require adaptation such that not only the system-level goals, but also the network-level goals are satisfied. In decentralized settings where there is no central mechanism exists to orchestrate and coordinate the collaboration between systems such that the satisfaction of network goals at runtime is guaranteed, failure to adapt individual CPS behavior may result in a functional degradation of the entire network, possibly to the point of injuring humans and harming individual systems in the network [9]. Thus, developers define goals to be achieved by the individual systems as well as with the collaborative network. To support this we defined a goal modeling approach that takes the characteristics of collaborative CPS into account [10].

In the following, we focus on the use of goal models for the definition of adaptation strategies to be implemented in the individual systems. These adaptation capabilities need to be systematically designed into each system during the later phases of development. Particularly, there is a need to reason about the impact the adaptation of an individual system has for the overall network and for other systems, with regard to their goals. While concrete adaptations occur at runtime, this reasoning is an inherently human engineering effort and requires careful planning of adaptation strategies to enable the system to select and execute adequate adaptations during development [9]. To assist this manual engineering task, we propose the use of established goal models [11] and recommend explicitly specifying system-level goals and network-level goals. Doing so makes obvious the dependencies on possible collaborative network configurations and possible adaptation conflicts. This allows to systematically analyze the relations between goals of individual systems and the collaborative network for different network configurations and contexts. Hence, this allows making engineering decisions regarding suitable adaptation strategies already during early development phases.

In this article, we contribute a goal-based approach for identifying and deciding upon suitable adaptation strategies taking system-level goals and network-level goals into account. The key feature of our approach is the differentiation between system-level and network-level goals as well as their interrelation. Doing so assists in making overt conflicts on those two levels. Such a conflict typically manifests itself in different alternative goals that can be activated according to the strategy selected. Thus the approach supports defining adaptive CPS capable of autonomously selecting the most suitable adaptation strategy according to relevant context phenomena, such as aforementioned changing network configurations. The approach, furthermore, enables individual systems not only to adapt to maximize their own goal fulfillment but also takes the goals of the network into account as well. We illustrate our approach and show proof of concept by means of an industrial case example from the industry automation domain.

In the following, Section 2 briefly introduces the state of the art and introduces goal modeling foundations for CPS. Section 3 explains the approach in detail. Section 4 presents a proof-of-concept case example. We discuss findings from applying the approach in Section 5. Section 6 concludes the article.

2. FOUNDATIONS AND STATE OF THE ART

Model-based engineering is considered a promising approach towards handling the complexity of modern CPS [12]. Especially collaborative CPS resemble many typical characteristics of systems of systems, such as emergent behavior resulting from interactions between autonomous systems (cf. [13]). Such systems form networks with a highly dynamic topology, and are able to dynamically reconfigure during operation [14]. Hence, it is essential to develop collaborative CPS with capabilities allowing them to adapt to changing environmental properties. Dynamic reconfiguration and adaptation have been extensively discussed in the area of self-adaptive systems (cf. e.g. [15]). In principle, the term "self-adaptiveness" describes systems that select from different, normally pre-programmed behaviors by making use of a MAPE-K feedback control loop [16,17]. Goal-orientation plays a major role in the development of self-adaptiveness [16]. While the managed system has certain goals to be achieved, the adaptation process itself fulfills specific goals by selecting the appropriate pre-programmed behvaior given the operational situation (cf. [17]). However, while systematic, model-based approaches have been developed to aid in the development of self-adaptive system properties [18], these approaches are based on a "closed world" assumption [19]. For cyber physical systems that form dynamic networks with an arbitrary number of nodes and heterogeneous behavior (i.e., CPS that exist in an "open world" [3]), predicting all possible adaptation scenarios is not possible. Therefore, we propose runtime reasoning based on network goal satisfaction.

2.1. Goal Modeling

During requirements engineering goal modeling (cf. [20,21,22]) is used to capture and reason among goals. Goal modeling approaches commonly used in the literature include GRL [11], i* [23,24] and its extension Tropos [25], or KAOS [26]. These approaches not only systematically support the definition, analysis and

evaluation of goal models, but often also include facilities to graphically represent goal relationships and formal foundations that allow reasoning about goals (see e.g. [27,28]). Goal modeling approaches often share a formal basis and are closely linked to other approaches. For example, the modeling language GRL (Goal-oriented-Requirements Language) [11] is based on the i* [23,24] modeling framework.

Studies have shown that in direct comparison of goal modeling with textual requirement specifications, compact goal models are better suited to uncovering possible conflicts than text documents (see e.g. [29]). Goal models are also used as the basis for automated reasoning techniques such as [27] or [28]. A more detailed overview over goal-oriented RE is given in [21].

Many extensions of established goal modeling techniques and languages have been proposed for specific purposes and types of systems, e.g. aspect-oriented extension of GRL [30]. Moreover, goal modeling has been successfully applied to drive adaptation strategies, such as in multi-agent systems (e.g. [31,32]), self-adaptive systems ([20,33,34]), systems of systems (e.g. [35,36]) or, like in this work, CPS (e.g. [37]).

2.2. Goal Modeling for Self-Adaptive CPS

Especially in self-adaptive cyber physical systems, goal models are heavily used to drive the adaptation decisions. For example, in [20] goal models are used to describe the current goals of a system and potential goals that characterize the system after adaptations (which are depicted in separate goal models). The Tropos4AS approach [34,38] proposes several extensions that consider states of environmental entities and potential failures to adapt autonomous agents. Others propose formal approaches that involve executable runtime models [39,40] to reason about goal satisfaction during operation. In [41,42] goal models are considered evolvable runtime entities to drive adaptations, considering adaptive goals that may induce changes in the goal models. Similarly, the OMACS approach [43] involves runtime goal modeling specific for multi-robot systems. In [44] goal models are used complementarily with other artifacts to reason about the impacts of different possible adaptation decisions, which is also supported by respective infrastructure at runtime. The FLAGS approach [45] employs fuzzy extensions to goal models to account for different degrees of goal satisfaction in the context of dynamic runtime adaptations. Another related approach that utilizes goal models, Adapt Cases [46], provides a framework for analyzing and documenting desired adaptivity during the development.

All of these approaches exclusively consider system-level goals for system-level adaptation. None of the above approach considers network-level goals in collaborative scenarios, like for CPS. Yet, designing collaborative systems adaptive is necessary to ensure that the CPS network meets system goals and network goals alike.

Modeling and reasoning over collaborative CPS requires considering multiple, distributed MAPE-K feedback control loops [5,47]. Regarding the use of goal models, this means that each CPS uses its own goal model for adaptation. State-of-the-art approaches typically only focus on single self-adaptive systems. For example, the underlying assumption of the OMACS approach [43] is that overall goals can be achieved by individual robots. In contrast, in a collaborative system with emergent properties, the individual goals of different partaking CPS cannot be considered in isolation. Rather, there are cross-dependencies between individual goals and the overall goals to be achieved through collaboration (cf. [35]). Hence, goals of the individual systems and goals of the overall system network should be explicitly distinguished in model-based manner [8] in order to monitor the adaptation of system behavior using, for example, a MAPE-K feedback control loop [47]. Hence, by contrast in [10], we proposed an approach for explicitly modeling these different goals and their relationships through clearly separating the individual systems and the overall CPS network.

In this article, we build upon existing goal modeling approaches and introduce a distinction between goal models of the CPS network and goal models of the individual CPS. This serves as a foundation to guide the systematic adaptation of collaborative CPS. Our approach specifically supports the systematic derivation of adaptation strategies in early phases based on such goal models. State-of-the-art approaches do not systematically support such derivation process in detail either. Rather, the identification of strategies as well as the identification of relationships to underlying goals and other relevant parameters of the adaptation context is often not considered in a structured manner from a requirements engineering (most notably, elicitation) perspective.

3. GOAL-BASED ADAPTATION STRATEGIES FOR COLLABORATIVE CPS

To support the systematic adaptation planning of collaborative CPS in early phases, we propose a goal-based approach. Based on GRL goal models that specify system-level goals as well as network-level goals and dependencies between them (Section 3.1). Adaptation strategies are developed that aid in fulfilling the network-level goals in the interplay of the individual systems (Section 3.2).

3.1. Network Goal, System Goal, and Adaptation Strategy Modeling

Fig. 1 gives an overview of the core concepts that are underlying our approach, as well as their relationships. The approach interrelates the CPS network goal model (containing network-level goals) with the CPS goal models (containing system-level goals) specifying goals of the individual systems, and relevant context parameters to be sensed. Together, this allows the definition of adaptation strategies. We define the following *Goal Model Types:*

- The *CPS network goal model* defines the goals of the collaborative CPS network. Collaborative CPS interact with other CPS in a network of collaborative CPS. This CPS network is formed to achieve goals the individual systems can only achieve in collaboration, but not on their own. As collaborative networks do not have a management instance monitoring goal fulfillment at runtime, the CPS network goals must be defined during development. The CPS network goal model contains every goal the CPS network shall be able to achieve eventually during its operation, independent from specific CPS network configurations, and specific context situations.

- The *CPS goal models* define the goals of the individual systems. The individual CPS strive to contribute to CPS network goals. Hence, the goals of the individual CPS are defined to fulfill the CPS network goals, which depend upon the individual CPS. Moreover, the individual CPS have their own goals they try to achieve in addition to contributing to the CPS network goals. In doing so, dependency relations between CPS network goals and individual CPS goals as well as conflicts are documented.

Hence, goal modeling is used to explicitly specify goals of the collaborative CPS network and the goals of the individual CPS. Furthermore, there exist dependency relations (modeled through respective links), as an individual CPS must contribute to the CPS network goals, and, consequently, satisfaction of CPS network goals depend on the satisfaction of certain individual CPS goals.

Naturally, not all goals can be fulfilled all the time. For instance, it is possible to have contradictory goals or goals that are specific for certain context situations. The goal models specify all the possible goals of the CPS network and its constituent CPS, and thus include goals that may not be achieved by a specific CPS network. Rather, the goal models can be seen as the basis to perform adaptations by selecting which goals to achieve as reactions to changing context situations. The goal models of individual CPS may even include goals that have no relation to CPS network goals, as each CPS is seen as an autonomous unit that aims to achieve individual goals as part of an overall system (cf. [48]). Hence, during requirements engineering, it can be challenging to define desired collaborations considering such trade-offs between network and individual goals.

Parameters. In addition to the goal models, our approach considers parameters that have an impact on the specific adaptations to be carried out during runtime. In our approach, the following parameter types are considered, based on the distinction between a CPS network and the individual CPS engaged in this network:

- *Context parameters* define the current contextual circumstances in which the CPS network operates at a certain point in time. These context parameters define relevant attributes of the physical surroundings, business strategies, or user demands to be fulfilled. For instance, in the platooning example from Section 1, the desired speed or the maximum allowed speed on a specific position on a highway are context parameters in this sense.
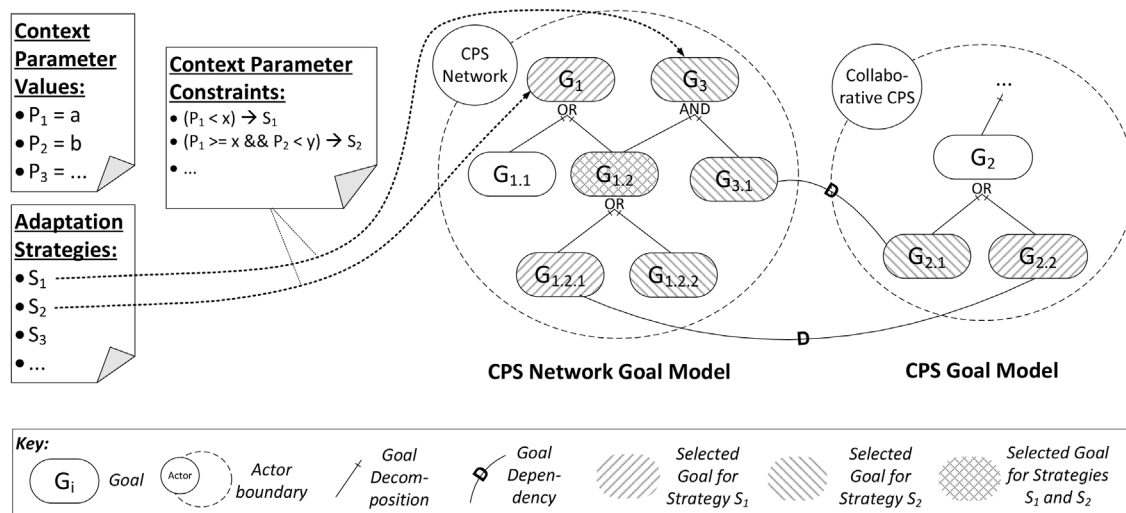

Figure 1. Overview of the concepts underlying our goal-based approach.

- *CPS operational parameters* capture the execution context an individual CPS faces w.r.t. fulfilling its goals. This includes resources that are used by CPS, such as computing or communication capabilities. Hence, positive or negative impacts of selecting a specific CPS goal for adaptation on such operational parameters have to be systematically captured as well.

Our approach emphasizes the role of context parameters that characterize the environment of the CPS network in order to monitor network-level goal satisfaction at runtime. While individual operational parameters of collaborating CPS may influence the possible adaptations as well, our approach most notably elaborates on how context parameters constrain the adaptation strategies to be defined. Depending on the context parameters it is defined under which conditions a certain strategy shall be chosen. Hence, constraints are defined over the context parameters, e.g. specifying a range of values a context parameter can take, given which strategy is selected. Furthermore, explicitly capturing effects on CPS operational parameters allows investigating the impacts of selecting a certain adaptation strategy.

Adaptation Strategies. Adaptation strategies define the adaptation behavior of CPS in dependence of the behavior of its context. A strategy takes into account context parameters, a set of system-level goals which are achievable by behaving according to the strategy, and a set of network-level goals that the CPS is contributing to, given the strategy. When context parameters or goals change at runtime, a strategy might no longer be sufficient for contributing to the goals, and a new strategy has to be chosen. This reflects in adaptation scenarios describing which CPS and network goals are to be pursued under certain circumstances. The sensed context parameters, the goals of the individual CPS and the goals of the CPS network in relation to the possible evolution of context parameters over time lead to the definition of adaptation strategies.

Strategies are selected based on observations of the environment, i.e., concrete values of the context parameters that characterize the current environment situation. Changes in the context may require adaptations to be performed in order to maintain operation of the overall CPS network, i.e., to achieve its goals. However, depending on the specific context situation, some goals may not be achievable. Such situations are specified using adaptation scenarios that describe how the CPS network needs to behave in terms of switching between different goal alternatives, as reactions to changes in context parameters. For instance, a corresponding strategy may focus merely on some minimal set of goals (e.g. defining degradation modes to avoid safety hazards) that can be achieved while other important goals may be ignored. Goal modeling decreases the number of strategies that have to be generated for a CPS at design time by identifying sets of goals that are not achievable in specific context situations (e.g. due to conflicts between goals).

3.2. Goal-Based Adaptation Planning

To support the systematic definition of adaptation strategies as introduced above, we identified a set of essential steps that should be followed when eliciting such strategies early in development. Fig. 2 shows a UML Activity Diagram that outlines the process for deriving adaptation strategies based on goal models.

Step 1. Initial versions of the CPS and the CPS network goal model are created. These do not necessarily have to be consolidated, i.e., there might be inconsistencies w.r.t. the possible adaptations of individual CPS that take the overall network goals into account. This might be especially the case when the goal models are created concurrently, e.g. by different development teams. This is a predominantly manual engineering process facilitated during typical system development.

Output: The outcome of this step is a set of goal models that allow defining the relevant functions and entities that will participate in the collaboration.

Step 2. Based on the goals models of the previous step, the contextual and CPS operational parameters are defined to systematically enhance and amend these models for the purpose of enabling the individual CPS to orchestrate and adapt according to strategies that fulfill the CPS network goals. As described in Section 3.1, these parameters can be captured in a textual representation. Furthermore, possible impacts of individual CPS goals on the operational parameters are defined as well. We argue that our goal-based approach also helps identifying the relevant parameters to be considered for potential adaptations.

Output: The outcome of this step is a set of consolidated goal models aligning adaptation goals of individual systems with desired adaptation goals on network-level.

Step 3. Using these parameters and narrowed-down entities involved in an adaptation, dependencies between CPS network goals and individual CPS goals are identified next. Here the focus is on defining which overall CPS network goal can be achieved by which individual CPS goal(s). Goal dependencies can be modeled graphically using GRL [11], as explained in Section 3.1.

Output: The outcome of this step is a GRL specification relating concrete goals of individual CPS with adaptation goals of the CPS network. This lays the foundation for selecting the ideal network adaptation.

Step 4. Based on the information elicited so far, requirements engineers can then systematically explore adaptation scenarios. In these adaptation scenarios, possible connections and interrelations between
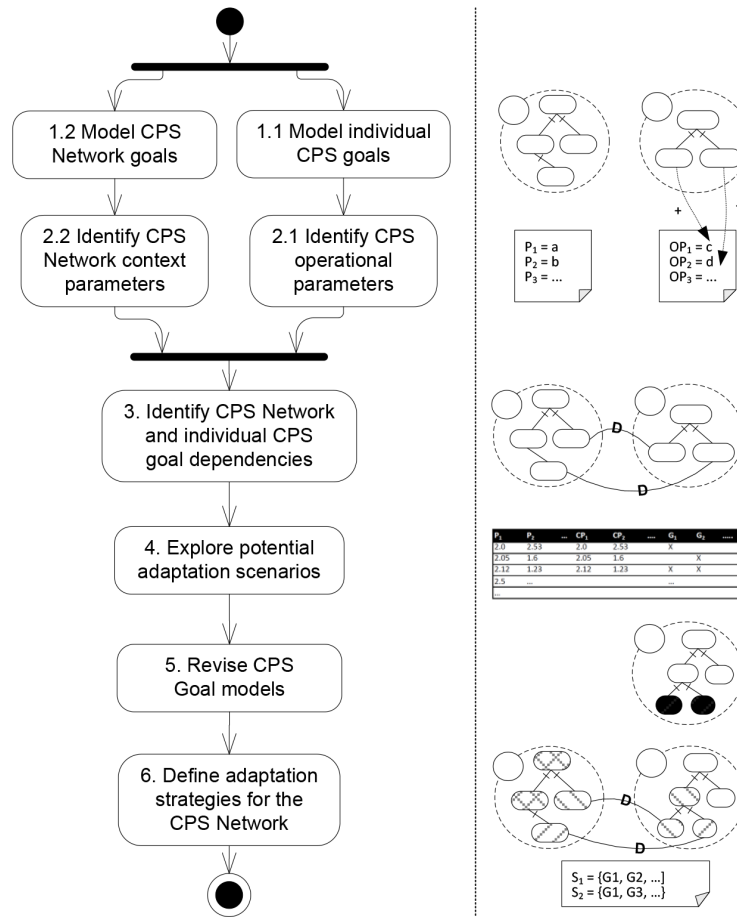
Figure 2. Process for goal-based definition of adaptation strategies.

the parameters and the goals that can be selected or deselected for adapting the system, as well as associated impacts on CPS performance are illustrated. Adaptation scenarios specify concrete examples how context parameter values may evolve over time. They consist of a sequence of time steps, each illustrating CPS network context parameter values and CPS operational parameter values, as well as the CPS goals that are selected to be "active" in order to achieve certain network goals (at each specific scenario step) in order to react upon context changes.

Output: The outcome of this step is a set of concrete adaptation scenarios showing the execution of adaptations at runtime.

Step 5. Technical solution examples specified in the scenarios in the previous step allows revisions of the initially devised CPS goal model from a more concrete, technical perspectives. Alternatives can be added, inconsistencies resolved, and goals can be relaxed in order to enable adaptations (cf. [49,50]) and make it technically feasible. Following this approach, individual CPS goals and CPS network goals are systematically aligned so that the identified strategies consider not only the individual CPS goals, but also the satisfaction of the overall CPS network goals.

Output: The outcome of this step are systematic revisions, additions, and deletions to the previously identified adaptation scenarios. This allows identifying missing adaptation goals for individual CPS or the entire network, discard superfluous goals, and identify missing scenarios to ensure successful adaptation at runtime.

Step 6. The aligned goal models and the adaptation scenarios then constitute the basis for defining strategies in the form of subsets of all the goals specified in the CPS network goal model. A strategy is then selected based on the identified relationships to relevant context parameters, i.e., when certain context parameter constraints are met.

Output: The outcome of this step is the specific strategies in which a CPS network-wide adaptation shall occur at runtime and which roles individual CPS will play.

Our approach has been originally devised for application in early requirements phases of collaborative CPS development processes, where adaptation needs to be planned in order to enable capabilities to engage and collaborate in a system network, and satisfaction of overall CPS network goals. Nevertheless, the approach in principle also helps when monitoring an existing CPS network, which may be open and consist of several individual systems whose contribution to the overall

network can only be assessed on the level of potential goals they strive for.

4. PROOF-OF-CONCEPT APPLICATION

We applied our approach to an industrial case example from the industry automation domain to assess the usefulness of the approach.

4.1. Case Example: Autonomous Transport Robots

Autonomous transport robots can form so-called fleets in order to process transport tasks together as a collective. Such robot fleets, consisting of individual transport robots, are used in production logistics to ensure prompt delivery and collection of material to and from various locations in production processes and are seen as enabler for smart factories [51]. Developing and operating autonomous transport robots poses many challenges, including a highly dynamic operating environment and the demand for reacting flexibly to changes in business constraints and new goals to be fulfilled [52].

There is a trend towards organizing transport robot fleets in a decentralized, collaborative manner, in order to allow for more flexibility and achieve better scalability through adding or removing individual robots (cf. [53]). In such a setting, agent-oriented techniques such as auctions, where the individual robots place bids for taking over a transport task, are typically employed to coordinate the fleet [54]. The individual robots taking part in a fleet may differ from each other in features such as load capacities, for example. While a non-collaborative robot typically optimizes its own route for goods transportation, the fleet allows the robots to optimize their routes taking into account the routes of other robots in order to avoid collisions. In addition, the fleet optimizes the transport of goods because in a collaborative network the task assignment to individual robots is handled flexibly through collaboration.

4.2. Goal Models of the Autonomous Transport Robots

Fig. 3 shows a goal model for the autonomous transport robots. The model represents two actors, the Robot Fleet and a Transport Robot belonging to the fleet. While the overall goal of the fleet is the fulfillment of all transport tasks, which can be divided into several subgoals, the overall goal of the robot is "task transport" in general. The goal of the individual robot is also divided into many sub goals, such as "maintain battery utilization".

While the sub goals of "transport task request" have to be fulfilled in order to fulfill the super goal as well, the goal "maintain battery utilization" is fulfilled when only one of its illustrated sub goals is achieved. Similarly, the goal "reduction of maintenance costs" of the overall robot fleet has two alternative realizations, defining the different ranges of the distance to be covered by each of the constituent robots. The maximum difference of the covered distance among the individual robots can
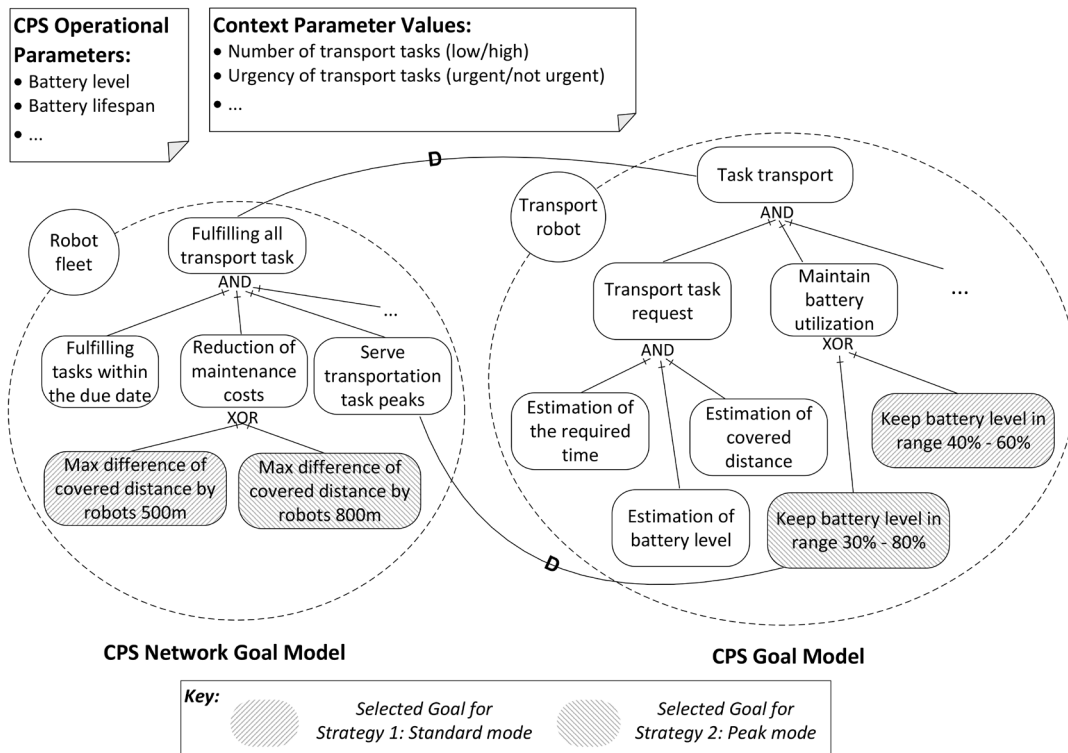

Figure 3. Goal model of individual transport robot (CPS) and the robot fleet (CPS network).

be either 500 m or 800 m, depending on the strategies defined by the plant manager, as will be explained below.

4.3. Definition of Adaptation Strategies

Based on the goal models introduced above, two exemplary strategies have been defined. There is one strategy incorporating goals to be achieved in normal operation (i.e., in standard mode), and one for peak mode, where the number of transport tasks to be fulfilled is significantly increased. Table 1 compares these two strategies, and lists an excerpt of the goals selected in each strategy.

As introduced in Section 3, context parameters are identified in addition to goal models. One context parameter is the number of transport tasks that are issued due to production demands. The standard mode assumes that neither urgent tasks nor too many tasks have to be carried out by the robots at once. Therefore, the goal "reduction of maintenance costs" of the robot fleet can be achieved, if the total covered distance differs between the robots by a maximum of 500 m, while being able to fulfill the normal transportation demands. Furthermore, the goal "keep battery level in range 40–60%" of the CPS is selected to allow for optimal battery usage to increase the lifespan of the battery. Thus, battery level and lifespan are two relevant CPS operational parameters that are considered in this case example.

The strategy defined for the peak mode includes a different subset of goals. The goal "reduction of maintenance costs" of the overall system network is in conflict with serving a high amount of incoming transport tasks. Hence, there needs to be another alternative sub goal that, when selected, reduces the distance in order to adapt to the high load. For this strategy it is more important to fulfill all tasks than to reduce maintenance costs of the robots. The goal of the system network is then achieved when the total distance covered by individual robots differs from the distances covered by other robots by a maximum of 800 m. Similarly, as each individual robot has to accomplish more transportation tasks, another desired average battery level is selected

for the peak mode strategy. The goal to serve transportation task peaks of the overall robot fleet can thus only be reached when the desired battery level is in between the broadened range of 30% to 80%.

4.4. Adaptation Scenarios

The fleet has the goal of fulfilling all transport tasks requested by the machines of the production plant without violating the due dates of the tasks. One possible adaptation scenario for illustrating the actual application of the defined adaptation strategies (based on the goals and parameters that have been identified in the first place) considers switching from standard mode to peak mode. This means that initially all robots of the fleet shall cover similar distances (e.g. at each point in time the difference between the minimum and maximum of the distances the individual robots covered on the same day is smaller than 500 m), so they have equal wear and tear and the mechanics can maintain all robots at the same time instead of having to visit the plant several times. According to the current strategy, each robot has the goal of holding its battery level in the range of 40% and 60%, so that the lifespan of the battery is high. Equipped with those two goals on the fleet level and one individual goal for each robot, each robot chooses a strategy that allows fulfilling the goals. Those strategies determine which robot is taking over which transport task at runtime. We assume that each robot has a task queue and is able to estimate the time, the covered distance and its battery level which it will have after fulfilling the last task of its queue. For example, each transport robot can choose the strategy of taking over a transport task if and only if all of the following conditions hold according to estimations of the robot:

- By adding the transport task to the end of its transport queue, it can fulfill the transport task in time.

- Directly after having fulfilled the task, the battery level of the robot is between 40% and 60%.

- According to the actual distribution of transport task over the fleet, all other robots will cover a longer

		Strategy 1: Standard Mode	Strategy 2: Peak Mode
Robot fleet	Goals	Reduction of maintenance costs	
		Difference of covered distance between robots max. 500 m per day	Difference of covered distance between robots max. 800 m per day
		...	...
Transport	Robot goals	Maintain battery utilization	
		Keep battery level in range 40–60%	Keep battery level in range 30–80%
		...	...

Table 1. Strategies of the system network and of individual CPS.

distance to fulfill the task, i.e. the difference between the covered distance of the robot and the robot with the least covered distance after having fulfilled the task is minimal among all robots.

This strategy for standard mode might be suitable as long as the goals of the fleet or of at least one robot is not changing and the number of transport tasks is so low, that the goals can be fulfilled. In reality, a fleet of transportation robots is regularly facing short time periods in which the amount of transport tasks is significantly higher than in the rest of the time, e.g. the number of transport tasks peaks each morning when all machines are turned on and require material at the same time. This becomes visible in relevant context parameters, in this case an increase in the number of transport tasks, which is evaluated based on defined constraints. For enabling the fleet to adapt to such situations, the plant manager may decide to adapt the strategy to peak mode, i.e., to weaken some of the goals to be fulfilled by the fleet and the individual robots before such peaks occur by increasing the permissible battery level to 30–80% and the difference in the covered distance to 800 m. This is done by activating the respective goals according to the new strategy selected, and could also be automated based on the defined context parameter constraints. By relaxing the fleet's goals, the robots can choose better strategies according to their individual goals, e.g. by charging all batteries up to 80% before the peak in the amount of transport tasks occurs, when the respective goal related to battery utilization has been switched (cf. Fig. 3). Hence, the adaptation of behavior to new goals increases the effectiveness of the robot fleet.

Another trigger for adaptation of the behavior is an update of the context model of the CPS network or of at least of one CPS. Such an update can be made manually (e.g. a new machine or charging station is installed in the plant, so that the internal map of the robots has to be updated) or by learning processes. For example, the fleet of robots can learn that a certain machine is requesting a specific transport good from the stock every 10 minutes. By adding this information to the context models of the robots, a change of the strategy can increase the performance of the fleet. For example, if the machine will request for the transport good within the next minute, a robot which is idling near the stock should not leave its position to take transport task at the other side of the plant, but should wait for the transport task to come and leave other transport tasks to the other robots.

5. DISCUSSION

Using the industrial case example, we showed that this approach is applicable—at least in this case—for the development of collaborative CPS. To evaluate its industrial usefulness, we applied our approach to several use cases during informal workshops with industry partners [55]. These workshops were conducted as part of a research project involving several academic and industrial partners.

5.1. Observations From Application With Industry Representatives

We proposed the specification of the individual CPS goals and the CPS network goals in GRL goal models, and explicitly documenting contribution links between these goal models. The majority of practitioners found the explicit specification of goal models useful and intuitive as it reflects their thinking about the system, which, due to the collaborative nature, mostly happens in terms of system goals. The explicit distinction between system goals and network goals extends the typical real of investigation and was well appreciated. Industry partners stressed the value of these strategies to reason about adaptation needs and problems arising in certain situations at early stages as well as for driving the further development process. Although not executed in all stages yet, industry partners have developed a prototype transport robot fleet using this goal-based approach for strategy definition. The defined strategies have further been used to instantiate monitoring and adaptation mechanism due to changed context values. So far, it has been confirmed that this approach aids in keeping track of the goals to be fulfilled for different configurations and situations.

5.2. Limitations

There remain several limitations and threats to validity. First of all, as is always the case for case study research, it cannot be ensured that the approach is applicable to other case examples as well. This is particularly the case, as the approach itself has been developed in close collaboration with industry partners, always having the transport robot case example in mind, which is comparable to design science research principles [56,57]. However, this increases the risk of having developed a specific solution approach only applicable to this case or to a limited number of comparable cases.

Discussions with other industry partners from the same domain, working on similar systems, confirmed the applicability of the approach for the development of industrial transport robots. However, discussions with practitioners from other domains or developing very different types of systems showed slightly different results. For instance, in case of a production system that relies on conveyor belts for transportation but places emphasis on the overall production process,

practitioners preferred a scenario-based approach, rather than reasoning about goals first. The approach has also been considered partly helpful by partners of the automotive domain. While the idea of using goals and explicitly specifying the interplay between the goals of the individual system and the network were very well received, automotive engineers expressed the desire for additional scenario representations, as this more closely related to the feature-centric development of automotive systems. We assume that the explicit expression of the adaptation scenarios with common model-based scenario languages can foster the application of the approach in these areas.

5.3. Inferences

Based on our findings we are confident that the approach is applicable and useful for the development of the transport robot case. We have also found evidence of the usefulness for the development of collaborative CPS from other domains. However, for now, we assume, that some kind of "goal thinking" must already have been established to make the utmost benefit of the approach. For other cases we aim at combining the strategy definition and the visualization of strategies with common approaches for scenario modeling and linking these to the goals. So far, we assume that this helps the transfer of the approach to the development of other system types.

6. CONCLUSION

In this article, we described a goal-based approach for systematically defining adaptation strategies for collaborative CPS. The approach takes individual CPS goals as well as goals of the overall system network and dependencies between these into account. Strategies are defined as subsets of the goals specified in the CPS goal model, which are to be achieved under given context circumstances, reflected in monitored context parameters. The latter are evaluated against defined constraints in order to allow for systematic adaptations, i.e., transitions between different strategies to be achieved. Hence, changes of context parameters may cause the change of the strategies of both the individual CPS and the CPS network. Such changes are systematically explored by defining adaptation scenarios, also taking into account operational parameters of individual CPS.

Using an autonomous transport robot case example, we demonstrated the applicability of our approach. Observations from introducing our approach with industry practitioners show that the principles of our approach are sensible and help in reasoning about adaptation strategies. However, scenario-centric reasoning

support is desired by practitioners working with manufacturing or automotive systems.

Initial results from a proof-of-concept application presented herein is qualitative and lacks quantitative, generalizable evidence to its effectiveness. Future work shall more rigorously evaluate the benefits of our approach, e.g. through experimentation and through application to other case examples, such as vehicle platooning [55]. Furthermore, we aim to extend our approach to also take dynamic adaptation scenarios into account that become necessary when a CPS network is already in operation. In such situations, the goal models and defined strategies may evolve at runtime, based on changes in the context. Hence, the relevant context parameters may change as well, which needs to be considered for evaluating the suitability of predefined strategies, and, if necessary, adapting the strategies as well.

Conflict of Interest

The authors declare that they have no conflicts of interest.

Data Availability

Expert evaluations through industrial application of our approach in a case study support the findings and are published OpenAccess as part of the CrESt project effort: [58]. Technical reports containing anonymized artifacts of the case study "Cooperating Transport Robots" are available upon request via https://crest.in.tum.de/lizenz.php.

Funding

Authors' Contribution

Daun, Bandyszak, and Brings developed the solution approach and saw through the implementation. Feeken provided feedback at early stages before assisting Bandyszak and Brings with application on the case example and authoring the initial technical report for the funded project. Daun supervised and provided revisions, as did Feeken. Tenbergen independently validated results before preparing the technical report for scientific publication. Tenbergen and Daun authored the initial version of this article, with revisions to all sections by Bandyszak, Brings, and Feeken.

REFERENCES

[1] P.J. Mosterman, J. Zander. Cyber-Physical Systems Challenges: A Needs Analysis for Collaborating Embedded Software Systems. Software & Systems Modeling, 2016, 15(1): 5–16.

[2] M. Törngren, U. Sellgren. Complexity Challenges in Development of Cyber-Physical Systems. In: M. Lohstroh, P. Derler, M. Sirjani (Eds.), Principles of Modeling: Essays Dedicated to Edward A. Lee on the Occasion of His 60th Birthday. Lecture Notes in Computer Science, LNPSE, Vol. 10760. Cham: Springer, 2018, pp. 478–503.

[3] B. Tenbergen, M. Daun, P.A. Obe, j. Brings. View-Centric Context Modeling to Foster the Engineering of Cyber-Physical System Networks. 2018 IEEE International Conference on Software Architecture (ICSA). Seattle: IEEE, 2018, pp. 206– 20609.

[4] M. Broy, A. Schmidt. Challenges in Engineering Cyber-Physical Systems. Computer, 2014, 47(2): 70–72.

[5] V. Klös, T. Göthel, S. Glesner. Comprehensible and Dependable Self-Learning Self-Adaptive Systems. Journal of Systems Architecture, 2018, 85–86: 28–42.

[6] B. van Arem, C.J.G. van Driel, R. Visser. The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics. IEEE Transactions on Intelligent Transportation Systems, 2006, 7(4): 429–436.

[7] C. Bergenhem, H. Pettersson, E. Coelingh, C. Englund, S. Shladover, S. Tsugawa. Overview of Platooning Systems. Proceedings of the 19th ITS World Congress. Vienna: Chalmers Publication Library (CPL), 2012.

[8] M. Daun, V. Stenkova, L. Krajinski, J. Brings, T. Bandyszak, T. Weyer. Goal Modeling for Collaborative Groups of Cyber-Physical Systems with GRL: Reflections on Applicability and Limitations Based on Two Studies Conducted in Industry. Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC '19). New York: ACM, 2019, pp. 1600–1609.

[9] T. Weyer, M. Daun, B. Tenbergen. The Changing World and the Adapting Machine: How Digital Transformation Changes Requirements Engineering in the Embedded and Cyberphysical Systems Industry. IEEE Software, 2021, 38(5): 83–91.

[10] M. Daun, J. Brings, L. Krajinski, V. Stenkova, T. Bandyszak. A GRL-Compliant iStar Extension for Collaborative Cyber-Physical Systems. Requirements Engineering, 2021, 26(3): 325–370.

[11] International Telecommunication Union (ITU-T), Z.151 : User Requirements Notation (URN) - Language Definition. Geneva: ITU, 2018.

[12] P.G. Larsen, J. Fitzgerald, J. Woodcock, P. Fritzson, J. Brauer, C. Kleijn, T. Lecomte, M. Pfeil, O. Green, S. Basagiannis, A. Sadovykh. Integrated Tool Chain for Model-Based Design of Cyber-Physical Systems: The INTO-CPS Project. 2016 2nd International Workshop on Modelling, Analysis, and Control of Complex CPS (CPS Data). Vienna: IEEE, 2016, pp. 1–6.

[13] W.C. Baldwin, B.J. Sauser, J. Boardman. Revisiting "The Meaning of Of" as a Theory for Collaborative System of Systems. IEEE Systems Journal, 2017, 11(4): 2215–2226.

[14] C.B. Nielsen, P.G. Larsen, J. Fitzgerald, J. Woodcock, J. Peleska. Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Computing Surveys, 2015, 48(2): 18.

[15] H. Horikoshi, H. Nakagawa, Y. Tahara, A. Ohsuga. Dynamic Reconfiguration in Self-Adaptive Systems Considering Non-Functional Properties. Proceedings of the 27th Annual ACM Symposium on Applied Computing (SAC '12). New York: ACM, 2012, pp. 1144–1150.

[16] J.O. Kephart, D.M. Chess. The Vision of Autonomic Computing. Computer, 2003, 36(1): 41–50.

[17] S. Shevtsov, M. Berekmeri, D. Weyns, M. Maggio. Control-Theoretical Software Adaptation: A Systematic Literature Review. IEEE Transactions on Software Engineering, 2018, 44(8): 784–810.

[18] T. Vogel, H. Giese. Model-Driven Engineering of Self-Adaptive Software with EUREMA. ACM Transactions on Autonomous and Adaptive Systems, 2014, 8(4): 18.

[19] L. Baresi, E. Di Nitto, C. Ghezzi. Toward Open-World Software: Issues and Challenges. Computer, 2006, 39(10): 36–43.

[20] H.J. Goldsby, P. Sawyer, N. Bencomo, B.H.C. Cheng, D. Hughes. Goal-Based Modeling of Dynamically Adaptive System Requirements. 15th Annual IEEE International Conference and Workshop on the Engineering of Computer Based Systems (ECBS 2008). Belfast: IEEE, 2008, pp. 36–45.

[21] J. Horkoff, F.B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini. Goal-Oriented Requirements Engineering: An Extended Systematic Mapping Study. Requirements Engineering, 2019, 24(2): 133–160.

[22] A. van Lamsweerde. Goal-Oriented Requirements Engineering: A Guided Tour. Proceedings Fifth IEEE International Symposium on Requirements Engineering. Toronto: IEEE, 2001, pp. 249–262.

[23] F. Dalpiaz, X. Franch, J. Horkoff. iStar 2.0 Language Guide. arXiv:1605.07767 [cs], 2016.

[24] E.S.K. Yu. Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering. Proceedings of ISRE '97: 3rd IEEE

International Symposium on Requirements Engineering. Annapolis: IEEE, 1997, pp. 226–235.

[25] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos. Tropos: An Agent-Oriented Software Development Methodology. Autonomous Agents and Multi-Agent Systems, 2004, 8(3): 203–236.

[26] A. Dardenne, A. van Lamsweerde, S. Fickas. Goal-Directed Requirements Acquisition. Science of Computer Programming, 1993, 20(1–2): 3–50.

[27] P. Giorgini, J. Mylopoulos, E. Nicchiarelli, R. Sebastiani. Reasoning With Goal Models. In: S. Spaccapietra, S.T. March, Y. Kambayashi (Eds.), Conceptual Modeling – ER 2002, Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2002, pp. 167–181.

[28] C.M. Nguyen, R. Sebastiani, P. Giorgini, J. Mylopoulos. Multi-Objective Reasoning With Constrained Goal Models. Requirements Engineering, 2018, 23(2): 189–225.

[29] J. Hassine, D. Amyot. An Empirical Approach Toward the Resolution of Conflicts in Goal-Oriented Models. Software and Systems Modeling, 2017, 16(1): 279–306.

[30] G. Mussbacher, D. Amyot, J. Araújo, A. Moreira, M. Weiss. Visualizing Aspect-Oriented Goal Models With AoGRL. Second International Workshop on Requirements Engineering Visualization (REV 2007). New Delhi: IEEE, 2007, pp. 1.

[31] M. Morandini, L. Penserini, A. Perini. Operational Semantics of Goal Models in Adaptive Agents. Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '09). Richland: AAMAS, 2009, pp. 129–136.

[32] Z. Shen, C. Miao, X. Tao, R. Gay. Goal Oriented Modeling for Intelligent Software Agents. Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2004). Beijing: IEEE, 2004, pp. 540–543.

[33] Y. Jian, T. Li, L. Liu, E. Yu. Goal-Oriented Requirements Modelling for Running Systems. 2010 First International Workshop on Requirements@Run. Time. Sydney: IEEE, 2010, pp. 1–8.

[34] M. Morandini, L. Penserini, A. Perini. Towards Goal-Oriented Development of Self-Adaptive Systems. Proceedings of the 2008 International Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '08). New York: ACM, 2008, pp. 9–16.

[35] E. Cavalcante, T. Batista, N. Bencomo, P. Sawyer. Revisiting Goal-Oriented Models for Self-Aware Systems-of-Systems. 2015 IEEE International Conference on Autonomic Computing. Grenoble: IEEE, 2015, pp. 231–234.

[36] E. Silva, T. Batista, E. Cavalcante. A Mission-Oriented Tool for System-of-Systems Modeling. 2015 IEEE/ACM 3rd International Workshop on Software Engineering for Systems-of-Systems (SESoS '15). Florence: IEEE, 2015, pp. 31–36.

[37] J. Park, S. Kang, I. Chun, W. Kim. Goal Modeling Method for Autonomie Control of Cyber-Physical Systems (CPS). 16th International Conference on Advanced Communication Technology. Pyeongchang: IEEE, 2014, pp. 295–299.

[38] M. Morandini, F. Migeon, M.-P. Gleizes, C. Maurel, L. Penserini, A. Perini. A Goal-Oriented Approach for Modelling Self-Organising MAS. In: H. Aldewereld, V. Dignum, G. Picard (Eds.), Engineering Societies in the Agents World X (ESAW 2009). Lecture Notes in Computer Science, LNAI, Vol. 5881. Berlin, Heidelberg: Springer, 2009, pp. 33–48.

[39] M.U. Iftikhar, D. Weyns. ActivFORMS: Active Formal Models for Self-Adaptation. Proceedings of the 9th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS 2014). New York: ACM, 2014, pp. 125–134.

[40] K. Welsh, P. Sawyer, N. Bencomo. Towards Requirements Aware Systems: Run-Time Resolution of Design-Time Assumptions. 2011 26th IEEE/ACM International Conference on Automated Software Engineering (ASE 2011). Lawrence: IEEE, 2011, pp. 560–563.

[41] L. Baresi, L. Pasquale. Adaptive Goals for Self-Adaptive Service Compositions. 2010 IEEE International Conference on Web Services. Miami: IEEE, 2010, pp. 353–360.

[42] L. Baresi, L. Pasquale. Live Goals for Adaptive Service Compositions. Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '10). New York: ACM, 2010, pp. 114–123.

[43] C. Zhong, S.A. DeLoach. Runtime Models for Automatic Reorganization of Multi-Robot Systems. Proceedings of the 6th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS '11). New York: ACM, 2011, pp. 20–29.

[44] X. Franch, P. Grunbacher, M. Oriol, B. Burgstaller, D. Dhungana, L. Lopez, J. Marco, J. Pimentel. Goal-Driven Adaptation of Service-Based Systems From Runtime Monitoring Data. 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops. Munich: IEEE, 2011, pp. 458–463.

[45] L. Baresi, L. Pasquale, P. Spoletini. Fuzzy Goals for Requirements-Driven Adaptation. 2010 18th IEEE International Requirements Engineering Conference. Sydney: IEEE, 2010, pp. 125–134.

[46] M. Luckey, B. Nagel, C. Gerth, G. Engels. Adapt Cases: Extending Use Cases for Adaptive Systems. Proceedings of the 6th International Symposium on Software Engineering for

Adaptive and Self-Managing Systems (SEAMS '11). New York: ACM, 2011, pp. 30–39.

[47] S. Wätzoldt, H. Giese. Modeling Collaborations in Adaptive Systems of Systems. Proceedings of the 2015 European Conference on Software Architecture Workshops (ECSAW '15). New York: ACM, 2015, 3.

[48] M.W. Maier. Architecting Principles for Systems-of-Systems. Systems Engineering, 1998, 1(4): 267–284.

[49] M. Ahmad, J.-M. Bruel, R. Laleau, C. Gnaho. Using RELAX, SysML and KAOS for Ambient Systems Requirements Modeling. Procedia Computer Science, 2012, 10: 474–481.

[50] E.M. Fredericks, B. DeVries, B.H.C. Cheng. AutoRELAX: Automatically RELAXing a Goal Model to Address Uncertainty. Empirical Software Engineering, 2014, 19(5): 1466–1501.

[51] H. ElMaraghy, W. ElMaraghy. Smart Adaptable Assembly Systems. Procedia CIRP, 2016, 44: 4–13.

[52] H. Andreasson, A. Bouguerra, M. Cirillo, D.N. Dimitrov, D. Driankov, L. Karlsson, A.J. Lilienthal, F. Pecora, J.P. Saarinen, A. Sherikov, T. Stoyanov. Autonomous Transport Vehicles: Where We Are and What Is Missing. IEEE Robotics & Automation Magazine, 2015, 22(1): 64–75.

[53] B.-H. Schlingloff, H. Stubert, W. Jamroga. Collaborative Embedded Systems - A Case Study. 2016 3rd International Workshop on Emerging Ideas and Trends in Engineering of Cyber-Physical Systems (EITEC). Vienna: IEEE, 2016, pp. 17–22.

[54] M.B. Dias, R. Zlot, N. Kalra, A. Stentz. Market-Based Multirobot Coordination: A Survey and Analysis. Proceedings of the IEEE, 2006, 94(7): 1257–1270.

[55] H. Schlingloff. CrESt Use Cases. In: W. Böhm, M. Broy, C. Klein, K. Pohl, B. Rumpe, S. Schröck (Eds.), Model-Based Engineering of Collaborative Embedded Systems. Cham: Springer, 2021, pp. 1–14.

[56] A.R. Hevner, S.T. March, J. Park, S. Ram. Design Science in Information Systems Research. MIS Quarterly, 2004, 28(1): 75–105.

[57] K. Peffers, T. Tuunanen, M.A. Rothenberger, S. Chatterjee. A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 2007, 24(3): 45–77.

[58] W. Böhm, M. Broy, C. Klein, K. Pohl, B. Rumpe, S. Schröck. Model-Based Engineering of Collaborative Embedded Systems - Extensions of the SPES Methodology. Cham: Springer, 2021, pp. 404.