Organisational Mode Switching for Controlled Run-Time Adaptation in Industrial Multi-Agent Systems

Michael Köhler-Bußmeier[1,†], Jan Sudeikat[1,†] and Finn Wellershaus[1,†]

[1]*University of Applied Science Hamburg*
Berliner Tor 7, D-20099 Hamburg, Germany

Abstract
Cyber-physical systems operate within and manipulate physical environments. Thus, these systems require differing levels of adaptivity. In addition, cutting-edge systems require system elements to show differing levels of autonomy in order to address distribution and robustness aspects. In earlier works, we proposed organizational modeling as a vehicle to explicitly manage system assignments. Consequently, designing run-time adjustments of organizations inherently involves balancing individual autonomy and organizational obedience. However, it is far from obvious how to design appropriate, meaningful, goal-directed, high-level transformations. Especially Industrial Agent Systems need simple and predictable forms of self-modification. Therefore, we extend the *Self-Modification@Run-Time* approach with *organizational modes* of agent-interplay that can be switched at run-time to self-organize the system. This enables a light-weight design approach to balance the interplay of architectural adaptation and individual agent autonomy in a developer-friendly way.

Keywords
Autonomous Systems, Adaptive Systems, Cyber-Physical Systems, Industrial Agents, Organizations, Self-X

1. Introduction

Constructing cyber-physical systems (CPS) means interconnecting multitudes of physical and logical system elements. When these multitudes operate under changing environment conditions, adaptability by autonomous operation of elements is typically required. However, controlling ensembles of autonomous system elements is in itself a design challenge, continuously explored in the *multi-agent* field of research.

Usually, CPS are owned and operated by more than one entity. Thus, governance of system functions becomes a central issue for the survivability of a CPS. Textbook examples are *centralized control*, the use of *market-based* mechanisms and *commons-based* approaches [1]. These governance models provide means for bridging the micro-macro gap [2] of multi-agent systems. Here, we assert that structured development practices [3] implicitly or explicitly select a governance model and systematically derive system implementations. Indeed, the successful design and implementation of agent-based CPS following a single governance model is a development challenge. However, real-world CPS require adjustments, e.g., electric power systems have to adapt their operating mode to handle resource shortages (We will come back to this scenario for our case study in Sec. 3).

A major challenge is to engineer adaptation and self-organization. To this end, we propose the concept of *mode switching* as an approach that balances engineering demands (i.e., predictability) with ability to self-adaptation at run-time (i.e., flexibility). In earlier works, it has been argued that organizational models of MAS are suitable abstractions for guiding the run-time adaptation of systems, particularly CPS [4, 5]. Building on this perspective, we propose organizational modes as a concrete realization of this idea. Since searching for the optimal organization for a given context at run-time is computationally expensive, we outline a lightweight approach to context-based reorganization, where

International Workshop on Petri Nets for Adaptive Systems, PNAS'26
*Corresponding author.
†These authors contributed equally.
michael.koehler-bussmeier@haw-hamburg.de (M. Köhler-Bußmeier); jan.sudeikat@haw-hamburg.de (J. Sudeikat); janne.wellershaus@haw-hamburg.de (F. Wellershaus)
0000-0002-3074-4145 (M. Köhler-Bußmeier); 0009-0007-7871-9006 (J. Sudeikat); 0009-0006-8729-038X (F. Wellershaus)

so-called *organizational modes* are predefined and switched. We relate this approach to the principal adaptation levels of CPS and outline how modes can be switched at run-time through the exchange of organizational models. Since this viewpoint allows reorganization, governance rules can be enforced by constraining the modes available to a system.

Concretely, our control and governance models are based on Petri nets [6, 7]. One part of our specification, the organization net, describes the network of agents, and another part, the protocols, define the orchestrated interaction between them (cf. Figure 1). The organization net defines the structure of the network, which is a hierarchy in the example below, where the position *broker* is placed 'above' *firm 3*, *virtual firm*, and *requester*. This means that the broker can coordinate its subordinates. At run-time, the broker connects to subordinate agents to form a team. The team then interacts according to the agent interaction protocol associated with it. In the example below, the AIP is a simple producer-consumer interaction. For simplicity, arc expressions, guards, and other inscriptions have been omitted from the net models. Details on the formalism can be found in [8, 9, 10]. However, our concept of *mode switching* can be considered a general approach applicable to any agent-oriented system, as it mainly relies on concepts such as roles, which are common to all agent-based approaches.

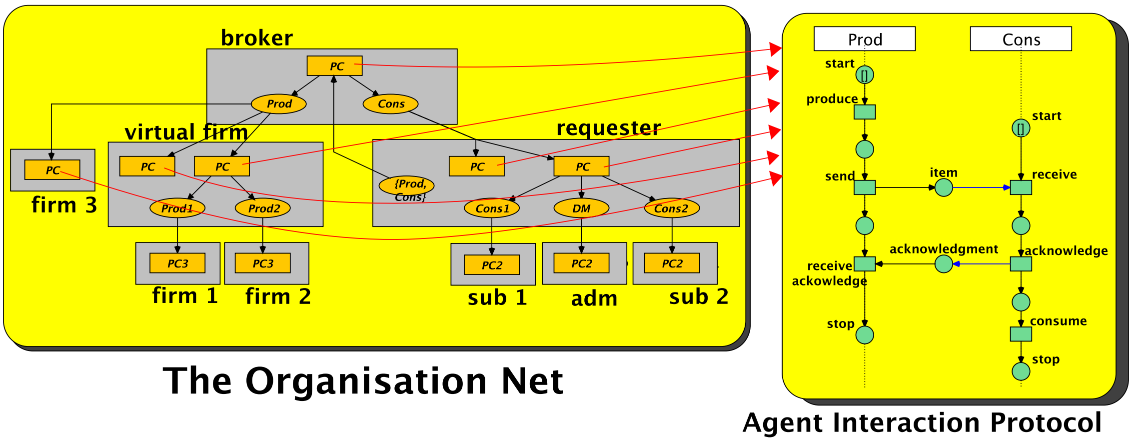

Figure 1: Sonar: Agent Interaction Protocols are orchestrated by Organizational Models (adapted from [9])

This paper is structured as follows. First, we outline the context of this work and discuss levels of adaptivity in Sec. 2. Then, we discuss designing and embedding organizations within CPS development practices and exemplify the switching approach in Sec. 3. In Sec. 4, we study how organizational modes can be integrated into engineering methods for CPS, like the Use Case Methodology. Finally, we present conclusions and outline directions for future work.

2. Related Work

Before developing an applied model, existing models and previous work need to be considered. First, an overview of the use of Multi-Agent System (MAS) in Cyber-Physical System (CPS) is given in Section 2.1. Next, the existing testbed is presented in Section 2.2. To lead the way towards tackling the first and second challenges described in the introduction, an overview of organization in MAS is given in Section 2.3. Adaptation in organizations is described in Section 2.4. Finally, the importance of organizational adaptation for governance of systems is presented in Section 2.5.

2.1. Multi-agent Systems and Cyber-Physical Systems

CPS combine physical devices with logical systems to manage distributed systems. One application scenario are power grids. These devices can be viewed as Industrial Agents (IAs) interacting with each other and their environment. By doing so, the design and development of CPS can be related to MASs.

The complexity and importance of large-scale systems call for governance. It is not just a number of IAs interacting freely with each other, but there are rules governing which interactions should take place as well as parameters and thresholds for desired system states. These aspects, which can be viewed as an additional layer on top of the CPS, correspond to organizations within MAS. These can be used to capture roles, hierarchies, groups and interaction protocols of a system. There are established models for the development of Organization-Centered Multi-Agent Systems (OCMAS) and proposed approaches to combining it with CPS [11].

Real-world CPS like smart-grids exist in changing environments. This poses a challenge to those systems, since they might need to adapt to be able to fulfill their goals within those changing conditions. Failure to adapt risks non-optimal results or even system failures. Furthermore, adding self-adaptation reduces workload at run-time from the system designers and maintainers, since the system can react autonomously and maintain the desired behavior even in challenging situations. However, adaptation comes at a cost. It needs to be evaluated which adaptations are possible and which ones to choose in a given situation. Once adaptation is possible within a system, there is a risk that it may not improve the system but instead break it by leading to invalid or undesired states. Lastly, adding the flexibility required for adaptation to a system poses engineering challenges in itself, including the determination of the level of adaptation supported by the system. There is ongoing research into the application of self-adaptation in CPS. Multiple approaches [12, 13, 14] focus on enabling agents to adapt and either make centralized or decentralized decisions about optimal adaptations. This means that they focus on the micro level, but the macro level receives less attention. However, when considering complex real world scenarios involving multiple stakeholders, governance is an important issue to consider. The proposed approach using organizational modes can help address these challenges and integrate governance models into the CPS. Organizations as part of a macro view can be used to realize different governance models within MASs. When developing and designing CPS, these organizations can serve as documentation of the system architecture and, furthermore, be used at run-time as artifacts for analyzing and evaluating possible adaptations and optimizations. When considering adaptations, changes within the agents can trigger changes in the organization, and the other way around. This concept is known as the micro-macro link. The modes help prevent undesired adaptations by limiting the system to predefined adaptations and states that can be analyzed independently. They provide defined guidelines for the system to follow and structure its adaptability.

When considering self-adaptation in CPS, the MAPE-K [15] model is established for centralized as well as decentralized applications. It differentiates between a managed system and a managing system. The managed system is the system of software and physical agents that run and can be changed via adaptation mechanisms. The managing system runs through a feedback loop of monitoring, analyzing, planning and executing to identify the need for the adaptation of the managed system and perform it. There are multiple integrations of MAPE-K for CPS [12] as well as in Organization-Based MAS (Org-MAS) [16].

2.2. Existing Testbed Infrastructure

Within this context of OCMAS for smart grids and CPS, specialized tooling can help with developing and evaluating organizations [11]. A testbed that simulates self-adaptive organizations in different cyber-physical scenarios can provide data for designers to evaluate these organizations. To support this, it must able to simulate varied and potentially complex scenarios. Co-simulations combine multiple smaller simulators to achieve accurate representation of more complex scenarios with multiple distinct components like weather, traffic, grid simulation and household demand [17]. From an engineering perspective, utilizing a framework to enable co-simulation introduces a layer of abstraction between the actual scenario and a MAS testbed. Interactions with the environment take place via the framework, which allows simulators used in the background for physical components to be replaced independently of the MAS. Furthermore, every simulation software or already established scenario using the co-simulation could be extended by plugging in the testbed, which can act as a new layer of control. The basic testbed infrastructure was previously developed and is currently being extended to include the mode switching presented within this paper. It is being developed as federate to be run within a HELICS [18] co-simulation. On top of the HELICS layer used for the environment, an agent layer is used for control and management of the agents within the system. The Jadex-V framework is used as the software basis. There is also support for roles that agents can adapt to perform actions connected with their corresponding roles to enable later addition of the organizational layer, as introduced within this paper. The testbed consists of multiple parts, a MAS federate, the HELICS broker and the other federates used for system simulations.

2.3. Multi-Agent System Organizations

To properly apply an adaptive organization model, an overview of relevant existing organization-based MAS is required. This section does not aim to give a comprehensive overview, but to present the relevant organizations for this paper, which include AGR [19], Moise [20] and Sonar [21]. Generally, organizations for MAS impose a structure on the agents, which puts constraints on them and reduces their action space [22]. There is tension between a fully permissive organization, which puts no constraints on the agents and might as well not exist, and an organization which is too limiting and leaves no choice and options to the agents. The latter form loses any benefits that might be gained from having theoretically autonomous intelligent and proactive agents. This tension is mirrored by the energy grid governance models of free market and centralized control. A balance must be struck between benefiting from the organization and preserving sufficient agent autonomy. This holds true independent of the model used. However, other differences must also be considered. Various dimensions can be used to describe the modeling capabilities of an organization. These include *functional*, *structural*, *deontic* and *interaction*. Not every organization model explicitly differentiates between all dimensions.

AGR stands for Agent-Group-Role and is focused on basic components of an organization [19]. In its *structural* dimension, it describes organizations as a set of agents and the roles they can adopt as well as group structures and corresponding groups that consist of roles, which in turn consist of constraints and interactions. In the *functional* dimension, it proposes sequence diagrams for different group structures and their roles and how they work for a given task or adopt and leave roles.

Moise is a model that can be described along three dimensions: The *structural* dimension describes roles and their interactions and relationships, as well as groups. Groups are described as aggregate of roles with specified cardinality. This is called structural specification and it is described by using a specific notation [22]. Groups are separated into group specifications, which can be understood as a class descriptions, and the groups, which are instances of those classes. The *functional* dimension describes the goals and the ways to achieve them. This is done via goal decomposition trees called schemes. To this end, plans and missions for the agents are described. These schemes can be linked to different group specifications. This relationship is expressed via the functional specification [22]. The *deontic* dimension is used to tie *functional* and *structural* together by assigning missions and plans to the roles via permissions and obligations. On an architectural level, Moise has three layers: a communication layer, an agent layer and an organization layer. In the organization layer, each agent gets a so called "OrgBox" assigned which act as proxies to the organization for the agents to act according to the organizational specifications. A centralized "OrgManager" is used to manage the organization and control reorganization requests. These practical aspects are realized in the software implementation of Moise, called S-Moise [22].

Sonar models are organizational specifications based on Petri nets [21]. As part of the *interactional* dimension, there are Distributed Workflow Nets (DWFNs) describing the possible interactions between positions. Nets describing the tasks and their execution as well as potential refinement and delegations combine the *structural* and *functional* dimension by also integrating information about the existing positions and their tasks and responsibilities. Deontic information is only captured implicitly within these nets. The model separates between Organizational Position Agents (OPAs) and Organizational Member Agents (OMAs). The OMAs are the agents performing the actions and following the action corridors presented by the organization. The way these are communicated is through the OPAs, which represent positions within the organization. They are responsible for changes to this organization and enforce its rules. These positions come with roles and obligations. As part of the team formation, the actual groups of agents that are going to interact with each other at run-time are determined via negotiation between the affected OPAs. As part of the DWFN, the interactions of agents are defined by following the execution of the team operators.

2.4. Adaptive Organizations

Adaptation within an Org-MAS can happen on the level of agents that change their behavior. This accounts for changes like an agent performing new actions and discarding old actions. Within an organization, this can be represented by agents adopting new roles and discarding old ones. This process of adaptation within an organization is called reorganization. There are different approaches to model this reorganization.

Reorganization in Moise is integrated on an organizational level as presented in [23]. Since functional and structural specifications are mostly independent of each other, they can be altered by only requiring an adaptation of the deontic specification, which ties them together. To perform a reorganization, a reorganization group is specified. It defines roles for a centralized *OrgManager*, Historian for the adaptation history, Designer and Monitoring. In accordance with Moise, the reorganization process is also described as a goal decomposition tree. Based on the roles, there are agents that monitor the organization, analyze it, identify issues and propose alternatives to adopt. The actual adaptation is only performed by the one "OrgManager" agent.

Within the Sonar model, there is a proposed distinction of adaptation into levels [24]. The first order adaptation considers modifications to an agent's decision logic as well as the allocation of roles, affecting their environment. These are changes that occur within a given organization without altering it as a whole. Changes to the organization model are encapsulated within the second level. This includes adjustments to the interaction protocols, delegation and decision-making processes, and structural changes. An integration of a MAPE-K loop is [16], which also applies the different adaptation levels to a MAPE loop.

For the application of integrating MAPE-K within CPS and Sonar, a metamodel was developed [11]. This metamodel covers the relationship between physical devices, organization, legacy software agents and coordination processes. It defines resources which are provided by physical devices, owned by organizational positions and modified via the interactions of roles.

2.5. Adaptive Governance via Organizational Adaptation

In earlier works [24], we distinguished between first- and second-order adaptations. Using the *probabilistic workflow net* formalism [24] (an extension of inter-organizational workflow nets [25] where choices are modeled with probabilities), the adaptation of a multi-agent system can be characterized by changes in agent decision logic or by the adjustment / refinement of constituent workflows. This resembles the configuration of industrial agents as well as the iterative adjustment of organizational structures within MAS.

In this paper, we refine these levels of adaptation (cf. Figure 2). The *Managed System* [26] comprises software and hardware agents that can be remotely reconfigured. The higher layers of adaptivity define the levels of influence in *managing* [26] software functions. *First order* adaptivity addresses the changes within the current organization model. Options are (1a) to adapt the internal agent decision logic, (1b) adapt role allocations within the organization as well as the interplay between roles, and (1c) adapting the generation of AIPs (agent interaction protocols) within the organization.

Second order adaptivity refines the switching logic, i.e., how to adapt the Model of the organization. This breaks down to (2a) adapt how decisions are made or delegated within the organization, (2b) transform AIPs and (2c) change the organizational structures, i.e., add or remove teams and (re-)configure organizational elements. Thus, the model of the organization is subject to adaptation.

Consequently, a *third* order adaptation would concern the adjustment of the modeling elements of the organization. This relates to software *evolution* [26], since this level does not address automated adjustments, but rather development efforts be the development team. It adjusts the modeling elements for the system and the organization as well as the logic for deciding how to adjust switching logic at run-time.

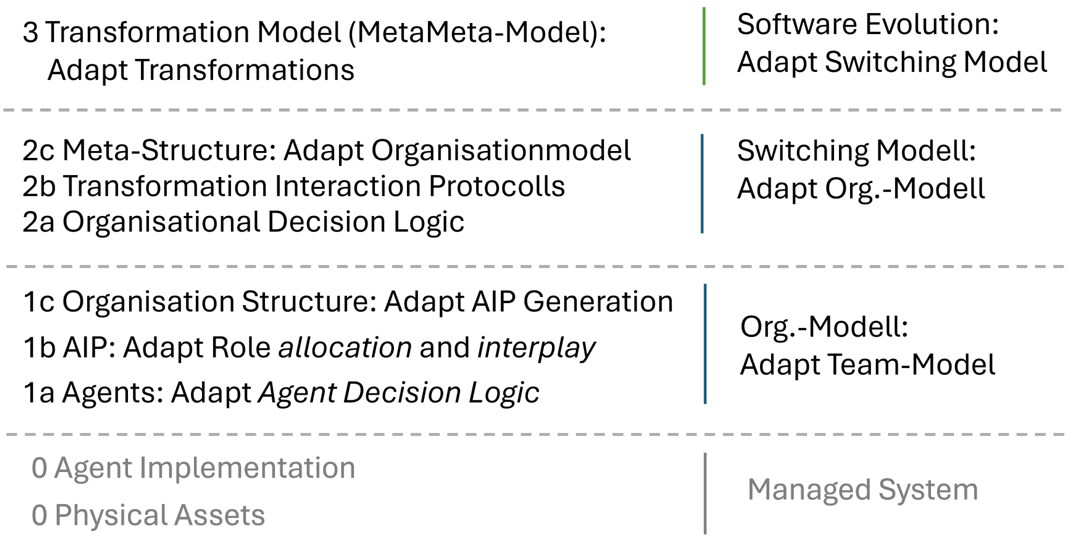

Figure 2: Levels of adaptivity and their high level view within the organizational model.

3. Re-Organization: From Reconfiguration Planning to Mode Switching

The self-modification at run-time is a challenging task. (For a discussion of challenges for the run-time adjustment of CPS via self-modifying inter-agent organizations, we refer to [4].) A prominent architecture-based adaptation approach using MAPE-based reasoning is proposed in [27, 26]. While this approach is conceptually interesting and performance-capable, it challenges engineering practices. Detrimental are, among other things, the efforts required to construct appropriate formal models of possible organizations within an application domain as well as the configuration of self-adaptive adjustments when systems deviate from these organizations. Self-planned reconfigurations also involves a loss of control concerning the run-time properties of the system itself, since planning techniques resolve sequences of desirable transformations (cf. Section 2).

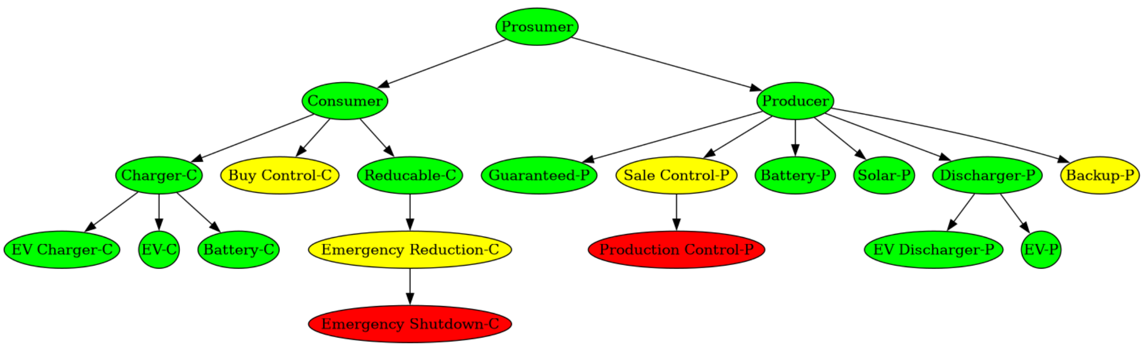

Figure 3: Inheritance hierarchy of roles.

Fundamental prerequisites for organizational planning P, as part of the planning phase in the MAPE-Loop [4], are (1) an appropriate goal description, concerning the set of permitted organizations, (2) the specification of available transformation operators, and (3) the definition of the formal semantics of both. These definitions have to be grounded on a formal organizational model, like in [28], where the transformational operations for our Sonar-model (cf. Section 2) are defined. However, machine readable specifications and application specific supplements have to be supplied to enable run-time planning of organizational changes.

In this work, we advocate a more light-weight approach. We propose defining an organization as a predefined set of related organizations, each called an *organization mode*. Additionally, we define *mode transitions*, i.e., when and how our adaptation engine performs a switch from the current mode to an appropriate next one.

An example of two organizations and the switching logic is given in the following section. Deploying a new organization within an application requires an organizational overlay. According to the Sonar-formalism, entities within the organization are represented as *positions*, i.e., OPAs. These entities represent the actual physical and computational agents that may be part of an organizational construct, e.g., team or group. Being part of this construction requires the agents to play specific roles. Thus, changing the allocation of roles to entities is a fundamental mechanism required for reorganization. Redeploying an organization can be done either by changing the association of agents to these OPAs or reassigning the roles agents play on the fly.

A *role hierarchy* (Fig. 3) is a basic design construct for our approach. At run-time, it has to be decided which roles can be played by specific entities. This can be done by depth-first searching and backtracking over role hierarchies. The resulting switching of modes is exemplified in Fig. 4.

The actual allocation of agents to positions, i.e., obligations to play specific roles, is based on their individual capabilities. Foundational modes for deciding agent allocations to OPAs can be imposed by a *centralized* adaptation engine or decided by *decentralized* inter-agent protocols [8] (cf. Section 2), e.g., using auction mechanisms. The latter approach enhances the ability of the system to self-organize by decentralizing the allocation logic. These deployment options (e.g., based on [29]) will be explored in future works.

3.1. Case Study: Exemplifying Adaptive Reconfiguration in CPS

This scenario is roughly inspired by grid traffic light approaches [30, 31]. This exemplifies an addition of *modes* because functioning CPS is supplemented with additional operation phases in order to prevent unwanted effects. Green operation modes resemble market-based governance while yellow and red modes denote differing grades of interventions in order to prevent failures. In our scenario, agents can group together and form coalitions to achieve goals. When joining those groups, agents accept a limitation for their valid action space in exchange for security, support, and more bargaining power. In the first organization, shown in the left part of Figure 4, no relevant limitations are initially present. This means that agents can act as they wish and are able to.

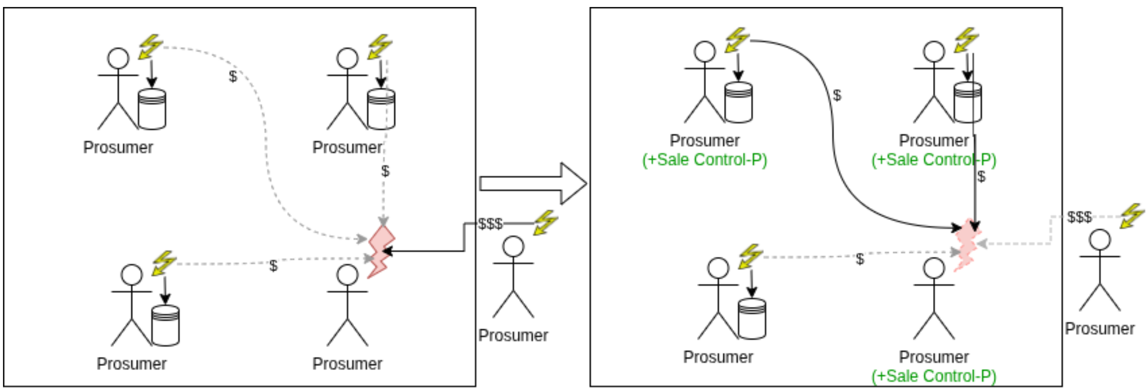

Figure 4: Transition from old organization to new reconfigured organization. Dashed arrows indicate possible but not chosen interactions, while solid arrows indicate chosen interactions. Agents choose not to cooperate or share power, which results in a mode escalation that enforces a degree of power sharing.

A mode switch, here from green to yellow, is a change in the environment triggered within the system. It affects roles, groups and adaptation processes differently. For example, certain roles may be active only in certain modes and be forced to stay inactive in others. Green roles can be active in the green mode. Other more controlling or emergency roles can require a higher mode (yellow) to be enabled. This is highlighted in Figure 3 and Tables 1 and 2.

Table 1
Producer role descriptions

Roles	Mode	Parameter	Description
Producer	both	production	Generic role for producers
Guaranteed-P	yellow	guaranteedAmount	Producer guarantees to provide a certain surplus.
Sale Control-P	yellow	amount	Controls how much power an agent either has to buy or may only buy.
Production Control-P	red		All of the agents decisions regarding the production can be taken from it.
Battery-P	both	level; rate	Unload a battery.
Solar-P	both	amount	Retrieve power from solar cells.
EV-P	both	DischargingRate	Role for EVs to be able to discharge at stations and provide energy.
EV Discharger-P	both	DischargingRate	Role to mark prosumers as being able to receive power from EVs.

When switching to the yellow mode, yellow roles are activated and can be used. Groups, as community of prosumers with different roles, can impose conditions and role requirements on their members. In addition, new group constraints are introduced. One example is an enforced minimum number of agents adopting (either new roles or activate old) yellow roles to ensure proper reactivity to the mode escalation. One the one hand, these constraints can already be part of the group specification. On the other hand, they can be the result of new adaptation processes that get unlocked by the new mode and result in a different organization. After a change of group structure, agents must adapt to new conditions for the group to continue operating.

Table 2
Consumer role descriptions

Roles	Mode	Parameter	Description
Consumer	both	consumption	Generic role for consumers
Buy Control-C	yellow	amount	Controls how much power an agent has to or may only buy.
Reducable-C	green	reducableAmount	Provides the information how much the agent is willing to reduce its usage.
Emergency Reduction-C	yellow	forcedReduction	Requires agents to reduce their usage by a specific amount, based on the roles parameters.
Emergency Shutdown-C	red		Marks agents as able to get shut down whenever this is required.
EV Charger-C	both	ChargingRate	Role responsible for marking a Prosumer as being able to connect to an EV to charge it.
EV-C	green	ChargingRate	EV s with this role are able to charge at another prosumer with the EV charger role.
Flex-C	green	FlexibleComponent	Agents with this role are allowed flexibility in their usage.
Battery-C	both	level; rate	Agents can store energy in batteries.

3.2. Transformation as Organizational Mode Switching

In the in Figure 4 presented configuration, scenarios may arise that appear individually beneficial to most of the agents but are not optimal for the system as a whole, for instance with respect to total group spending. For some reason, one agent has a negative power-balance. The other agents in the group have a positive balance and could support their peer, but they choose not to do so and instead recharge their batteries. This forces agent A to buy power from another agent outside of the group that charges a higher price (maybe because of lack of supply or because it is an agent from another group). In the optimal case, the group would not need to spend any money, since demand can already be met locally.

As part of a market-based mechanism, the agent could offer more money for power within the group to convince its peers to sell at least some part of their surplus. However, depending on the market price, this may cause the agent to leave the group whenever selling is not beneficial. Furthermore, there is no guarantee that the situation would be resolved, since the other agents may also decide to accept bids from external agents. This highlights that the issue is less a problem of prices and more one of a lack of cooperation and support guarantees provided by the group.

The *price of anarchy* (PoA) [32] – the proportion of the worst possible outcome compared to the best possible outcome – can be used to identify these situations. This problem can be solved by changing the group structure to motivate or enforce corresponding behavior. One option is to *add obligations* to a subset of agents within the group. This can be understood as a formalization of the promise to support each other. The “Sale Control-P” role can be used to this end. It can specify a minimum amount of power (calculated in the form of packages) that agents must offer for sale within the group. This way, a potential baseline within the group can be established resulting in an organization that better conforms to the given environment. Note that a change in the environment may lead to another suboptimal situation, requiring a reconfiguration of the organization.

4. Integrating Organizational Modes into CPS

The mode switching approach proposed in this paper is a natural extension to industrial agent-based systems [3], i.e., a design inspiration is mapped to an application architecture via reference architectures, and agent behaviors are interpreted as additional software functions for the CPS to be [3]. In this respect, modes are elaborations of the underlying design approaches. Due to the explicit switching, they can be developed and refined independently form one another. The connecting piece is the configuration of the mode transitions, which have to be consistent with governance rules and must not overwhelm the system.

We found that the modes described here fit well within development concepts for CPS. When refining these modes, standard engineering approaches should be employed.

Following the IEC 62443 *Use Case Methodology* [33], modes can be described as individual Use Cases. This allows for a consequent mapping of modes to associated reference architectures. The use case template already allows for specifying additional constraints and assumptions, where the required absolute or relative role allocations can be denoted. Since the organizational overlay structures how (software) functions are invoked, OMAs and OPAs concepts structure the *Function* layers and relate to the *Business* and *Information Layers* within the Smart Grid Architecture Model [33]. This adds anther perspective on integrating organizations in reference Architectures, which can be understood as software design [34] or social (e.g., [35]) modeling extensions.

IEEE 2660.1 [36] provides a structured way to identify agent deployments in industrial settings, and it has been shown how this can be mapped to typical reference architectures [37]. The deployment modes can serve as specifications and contain the reconfigurations of already running agents.

5. Conclusions

In this contribution we proposed organizational mode switching as a conceptual approach to enable engineering of run-time adaptations. We have shown that switching organizational models in industrial agent-based systems allows for the exchange of governance modes in a predictable and manageable way. This viewpoint enables the design of enhancements to conventional governance modes.

Mode switching allows the designer to structure the adaptation process in a clear way that is compatible with the MAPE-loop pattern. We monitor the system and analyze it. Whenever we conclude that the current mode is no longer appropriate, the planning phase must determine which of the possible "successor modes" is the most suitable for a mode switch. Note that mode switching is not fully predefined, as the specification contains declarative elements (e.g., it specifies that one agent must change its role, without specifying which one). The planning phase generates concrete steps for the execution phase. We would like to mention that modeling self-adaptivity using mode switching provides for a compositional verification of systems, since we can analyze each mode independently. This aspect is quite beneficial since the state-space of self-adaptive systems is likely to explode.

In future work, we will explore tool support for enacting these exchanges. This comprises a corresponding information modeling that serves as a language support for configuring, for example, role constraints and transportation rules. In addition, a *mechanism library* is required to enact group decisions (as discussed in [38]). In order to evaluate the practicality of these biding blocks, a co-simulation framework will be revised and extended as part of an currently ongoing Master's project. Besides that, we particularly plan to further investigate using game theoretic concepts, such as the Price of Anarchy, and related concepts to initiate role switches.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] P. H. J. Nardelli, Cyber-Physical Systems : Theory, Methodology, and Applications, John Wiley & Sons, Incorporated, Newark, 2022.

[2] M. Schillo, K. Fischer, C. Klein, The micro-macro link in DAI and sociology, in: S. Moss, P. Davidsson (Eds.), Second International Workshop on Multi-Agent Based Simulation, volume 1979 of *Lecture Notes in Computer Science*, Springer-Verlag, 2000, pp. 133–148.

[3] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, A. W. Colombo, Smart agents in industrial cyber–physical systems, Proceedings of the IEEE 104 (2016) 1086–1101. doi:10.1109/jproc.2016.2521931.

[4] J. Sudeikat, M. Köhler-Bußmeier, Controlled run-time adaptivity in industrial agent systems - challenges and research prospects (2023). doi:10.18420/INF2023_200.

[5] J. Sudeikat, M. Kohler-Bußmeier, Towards integrating multi-agent organizations in opc ua for developing adaptive cyber-physical systems, in: INFORMATIK 2022, Gesellschaft für Informatik, Bonn, 2022. doi:10.18420/INF2022_135.

[6] J. Sudeikat, M. Köhler-Bußmeier, Controlled run-time adaptivity in industrial agent systems: challenges and research prospects, in: Informatik 2023 - Designing Futures: Zukünfte gestalten, volume 337 of *Lecture Notes in Informatics*, 2023, pp. 2003–2014.

[7] M. Köhler-Bußmeier, J. Sudeikat, H. Rölke, L. Capra, Petri nets as run-time models for self-adaptive cyber-physical systems, in: Petri Nets and Software Engineering 2024, PNSE'24, volume 3730, CEUR, 2024. URL: https://ceur-ws.org/Vol-3730/paper10.pdf.

[8] M. Köhler-Bußmeier, M. Wester-Ebbinghaus, D. Moldt, A formal model for organisational structures behind process-aware information systems, Transactions on Petri Nets and Other Models of

Concurrency. Special Issue on Concurrency in Process-Aware Information Systems 5460 (2009) 98–114. doi:10.1007/978-3-642-00899-3_6.
[9] M. Köhler-Bußmeier, E. Deliu, Sonar/oredi: A tool for creation and deployment of organisation models, in: J. Dix, W. Cees (Eds.), Conference on Multi-Agent System Technologies, MATES 2010, volume 6251 of *Lecture Notes in Artificial Intelligence*, 2010, pp. 76–87.
[10] M. Köhler-Bußmeier, J. Sudeikat, Defining adaption measures for organizational multi-agent systems, International Journal of Parallel, Emergent and Distributed Systems (2023). doi:10.1080/17445760.2023.2243005.
[11] J. Sudeikat, M. Köhler-Bußmeier, Controlled run-time adaptivity in industrial agent systems : challenges and research prospects, in: Informatik 2023, volume 337 of *Informatik 2023 : Designing Futures: Zukünfte gestalten ; 26. – 29. September 2023 Berlin*, 2023, pp. 2003–2014. URL: http://hdl.handle.net/20.500.12738/14282, iSSN: 1617-5468.
[12] A. Yassine, M. S. Hossain, Cooperative Distributed Multiagent System for Carbon-Aware IIoT in Smart Grids, IEEE Internet of Things Journal 12 (2025) 34863–34874. doi:10.1109/JIOT.2025.3586089.
[13] J. Päßler, M. H. t. Beek, F. Damiani, E. B. Johnsen, S. L. T. Tarifa, Analysing Self-Adaptive Systems as Software Product Lines, Journal of Systems and Software 222 (2025) 112324. URL: https://www.sciencedirect.com/science/article/pii/S0164121224003686. doi:https://doi.org/10.1016/j.jss.2024.112324.
[14] S. Ilager, J. Fahringer, A. Tundo, I. Brandić, A Decentralized and Self-Adaptive Approach for Monitoring Volatile Edge Environments, 2024. URL: https://arxiv.org/abs/2405.07806, _eprint: 2405.07806.
[15] Danny Weyns, An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective, John Wiley & Sons, 2020.
[16] M. Köhler-Bußmeier, J. Sudeikat, Studying the Micro-Macro-Dynamics in MAPE-Like Adaption Processes, in: M. Köhler-Bußmeier, W. Renz, J. Sudeikat (Eds.), Intelligent Distributed Computing XVI, Springer Nature Switzerland, Cham, 2024, pp. 285–300.
[17] F. Tuxworth, A. Aijaz, Vehicle-to-Grid Technology meets Packetized Energy Management: A Co-Simulation Study, 2024. URL: https://arxiv.org/abs/2406.19296. doi:10.48550/ARXIV.2406.19296, version Number: 1.
[18] T. D. Hardy, B. Palmintier, P. L. Top, D. Krishnamurthy, J. C. Fuller, HELICS: A Co-Simulation Framework for Scalable Multi-Domain Modeling and Analysis, IEEE Access 12 (2024) 24325–24347. doi:10.1109/ACCESS.2024.3363615.
[19] J. Ferber, O. Gutknecht, F. Michel, From Agents to Organizations: An Organizational View of Multi-agent Systems, in: P. Giorgini, J. P. Müller, J. Odell (Eds.), Agent-Oriented Software Engineering IV, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 214–230.
[20] J. F. Hübner, J. S. Sichman, O. Boissier, A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems, in: G. Bittencourt, G. L. Ramalho (Eds.), Advances in Artificial Intelligence, Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, pp. 118–128.
[21] M. Köhler-Bußmeier, M. Wester-Ebbinghaus, SONAR*: A Multi-Agent Infrastructure for Active Application Architectures and Inter-organisational Information Systems, in: L. Braubach, W. van der Hoek, P. Petta, A. Pokahr (Eds.), Multiagent System Technologies, volume 5774, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 248–257. URL: http://link.springer.com/10.1007/978-3-642-04143-3_27. doi:10.1007/978-3-642-04143-3_27, series Title: Lecture Notes in Computer Science.
[22] J. F. Hübner, J. S. Sichman, O. Boissier, S-Moise+: A Middleware for Developing Organised Multi-agent Systems, in: O. Boissier, J. Padget, V. Dignum, G. Lindemann, E. Matson, S. Ossowski, J. S. Sichman, J. Vázquez-Salceda (Eds.), Coordination, Organizations, Institutions, and Norms in Multi-Agent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 64–77.
[23] J. Hübner, J. Sichman, Using the MOISE + for a cooperative framework of MAS reorganisation, volume 3171, 2004, pp. 506–515. doi:10.1007/978-3-540-28645-5_51.

[24] M. Köhler-Bußmeier, H. Rölke, Petri-nets@run.time: Handling uncertainty during run-time adaptation using digital twins, in: Petri Nets and Software Engineering 2023, PNSE'23, volume 3430, CEUR, 2023, pp. 34–52. URL: http://ceur-ws.org/Vol-3430/.

[25] W. v. d. Aalst, Interorganizational workflows: An approach based on message sequence charts and Petri nets, Systems Analysis - Modelling - Simulation 34 (1999) 335–367.

[26] D. Weyns, An Introduction to Self-Adaptive Systems: A Contemporary Software Engineering Perspective, John Wiley & Sons Ltd, 2020.

[27] J. Kephart, D. Chess, The vision of autonomic computing, Computer 36 (2003) 41–50. doi:10.1109/mc.2003.1160055.

[28] L. Capra, M. Köhler-Bussmeier, H. Rölke, J. Sudeikat, Petri nets as run-time models for self-adaptive cyber-physical systems, in: M. Köhler-Bussmeier, D. Moldt, H. Rölke (Eds.), Proceedings of the International Workshop on Petri Nets and Software Engineering 2024 co-located with the 45th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2024), June 24 - 25, 2024, Geneva, Switzerland, volume 3730 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024, pp. 164–181. URL: https://ceur-ws.org/Vol-3730/paper10.pdf.

[29] J. Sudeikat, M. Köhler-Bußmeier, Assets, artifacts and (Multi-)agents: On enacting adaptive organizations in cyber-physical systems, in: Informatik 2024, Lecture Notes in Informatics, 2024, pp. 1887–1892. URL: https://doi.org/10.18420/inf2024_163, 9th IACS Workshop 2024.

[30] G. A. o. E. BDEW, W. Industries, Smart Grid Traffic Light Concept, Technical Report, 2015. URL: https://www.bdew.de/media/documents/Stn_20150310_Smart-Grids-Traffic-Light-Concept_english.pdf.

[31] C. Exner, M.-A. Frankenbach, A. von Haken, A. Höck, M. Konermann, Practical implementation of the management of local flexible generation and consumption units using a quota-based grid traffic light approach, in: CIRED 2020 Berlin Workshop (CIRED 2020), volume 2020, 2020, pp. 432–435. doi:10.1049/oap-cired.2021.0082.

[32] Y. Shoham, K. Leyton-Brown, Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations, Cambridge University Press, 2008. doi:10.1017/cbo9780511811654.

[33] M. Gottschalk, M. Uslar, C. Delfs, The Use Case and Smart Grid Architecture Model Approach, Springer International Publishing, 2017. doi:https://doi.org/10.1007/978-3-319-49229-2.

[34] J. Sudeikat, M. Köhler-Bußmeier, On combining domain modeling and organizational modeling for developing adaptive cyber-physical systems, in: Proceedings of the 14th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART,, INSTICC, SciTePress, 2022, pp. 330–336. doi:10.5220/0010881200003116.

[35] S. Paustian, J. Köhlke, J. Mattes, S. Lehnhoff, The (still unexplored) social side of smart grid development: towards a social layer for the smart grid architecture model (SGAM), in: Proc. of the Thirteenth ACM International Conference on Future Energy Systems, ACM, Virtual Event, 2022, pp. 521–531. doi:10.1145/3538637.3539585.

[36] P. Leitão, T. I. Strasser, S. Karnouskos, L. Ribeiro, J. Barbosa, V. Huang, Recommendation of best practices for industrial agent systems based on the ieee 2660.1 standard, in: 2021 22nd IEEE International Conference on Industrial Technology (ICIT), volume 1, 2021, pp. 1157–1162. doi:10.1109/ICIT46573.2021.9453511.

[37] P. Leitão, S. Karnouskos, T. I. Strasser, X. Jia, J. Lee, A. W. Colombo, Alignment of the ieee industrial agents recommended practice standard with the reference architectures rami4.0, iira, and sgam, IEEE Open Journal of the Industrial Electronics Society 4 (2023) 98–111. doi:10.1109/ojies.2023.3262549.

[38] J. Sudeikat, M. Köhler-Bußmeier, Assets, artifacts and (multi-)agents: On enacting adaptive organizations in cyber-physical systems: Invited talk, in: INFORMATIK 2024, Gesellschaft für Informatik e.V., Bonn, 2024, pp. 1887–1892. doi:10.18420/inf2024_163.