Towards Adaptive Workflow Management by Case-Based Reasoning and Automated Planning

Lukas Malburg[1,2,*], Ralph Bergmann[1,2]

[1] Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany
[2] German Research Center for Artificial Intelligence (DFKI), Branch University of Trier, 54296 Trier, Germany

Abstract

Adaptive workflow management is an important topic in recent years, as increasing dynamics due to growing customer demands and faster changing market conditions require more flexibility in workflows. This is especially the case for rigid and rather standardized production processes that cannot be easily modified, if, for example, a breakdown occurs in one of the production lines. The goal of the *Fourth Industrial Revolution (Industry 4.0)*, is to provide, among others, more flexible and cost-effective processes in companies by using *Artificial Intelligence (AI)* methods. In this paper, we present 1) a framework for adaptive workflow management for IoT-enhanced manufacturing processes and 2) an idea for a new adaptation method that combines *Case-Based Reasoning (CBR)* and automated planning. In this context, we discuss the benefits of such a synergistic combination and introduce the framework and the individual phases according to the 4R (*R*etrieve, *R*euse, *R*evise, *R*etain) CBR cycle. In addition, we present our physical smart factory model that can be used to evaluate the suitability of developed research artifacts.

Keywords

Case-Based Reasoning, Automated Planning, Industry 4.0, Adaptive Workflow Management, Cyber-Physical Workflows

1. Introduction

The industry is in a major transformation towards more autonomous and intelligent production systems, a process known as the *Fourth Industrial Revolution (Industry 4.0)* [1]. In this context, the use of *Artificial Intelligence (AI)* methods is required [2] but still in early stages. In addition, current production systems are rather closed systems that work in isolation with predefined interfaces and only with low interoperability [2, 3]. To enable a more intelligent production in context of Industry 4.0, a closer connection between low-level devices and higher level systems (e. g., *Enterprise Resource Planning, ERP;* Manufacturing Execution Systems, MES*; or* Workflow Management Systems, WfMSs) is required for real-time decision-making and to control the rigid and rather standardized production processes in a more flexible way [3, 4, 1, 5]. However, state-of-the-art WfMSs are rather limited w. r. t. flexibility and often only provide means to handle simple, mostly expected exceptions that must be fully specified in the workflow model

ICCBR POCBR'22: Workshop on Process-Oriented Case-based Reasoning at ICCBR-2022, September, 2022, Nancy, France

[*] Corresponding author.

malburgl@uni-trier.de (L. Malburg); bergmann@uni-trier.de (R. Bergmann)
0000-0002-6866-0799 (L. Malburg); 0000-0002-5515-7158 (R. Bergmann)

[6, 7]. Unanticipated exceptions that occur during workflow execution, usually require more sophisticated (structural) workflow adaptations taking into account the environmental context of execution (cf. [8, 9]) to enable adaptive workflow management [6].

To date, advanced workflow adaptation methods rely either on a knowledge-intensive approach (e. g., [10, 7]), such as by using expert knowledge in *Case-Based Reasoning (CBR)*, or on a search-intensive approach by using a generative problem solver (e. g., [11, 12]) such as AI planning [13, 14]. Knowledge-intensive approaches such as CBR require experience knowledge from experts and search-intensive techniques need a comprehensive, formal domain description for the use of AI planning. In addition, solving a planning problem is an NP-complete problem [14]. Connecting both ends of the problem-solving reasoning continuum [13] as in *Case-Based Planning (CBP)* [15, 16, 14] combines the advantages of both approaches and could lead to time savings and better results during problem-solving as well as to reduced knowledge acquisition and modeling efforts.

Several approaches have been presented in the area of CBP (e. g., [17, 12, 13, 18, 11]). However, there is no approach in the context of Industry 4.0 and *Cyber-Physical Production Systems (CPPSs)* [2] by using *Business Process Management (BPM)* solutions and semantic technologies. In addition, only a few evaluate their developed research artifacts in physical environments with real problem situations instead of using synthetically generated data or simplified sample domains. The contribution of this paper is twofold: 1) we present a generic architectural framework for adaptive workflow management in smart environments, and 2) we propose an idea for a new adaptation method that combines CBR and automated planning by enabling runtime adaptations of manufacturing workflows. In this context, we discuss how experience-based adaptation methods can be combined with generative problem-solving and how this approach differs from existing related work. In order to validate developed research artifacts for workflow adaptation under real-world conditions, we use a physical *Fischertechnik (FT)* smart factory model [3, 19, 5] that emulates two independent production lines and is controlled by a WfMS in a process-oriented manner. The advantages of using such physical models are that it is possible to imitate real production in a protected environment at rather low costs but to maintain the runtime characteristics for evaluation and the transferability to real environments [3, 5].

In the following, Sect. 2 describes the used physical smart factory and related work for adaptive workflow management in context of BPM and the *Internet of Things (IoT)* as well as approaches that use CBR for adaptive workflow management. The architectural framework for adaptive workflow management and the idea for a new adaptation method are presented in Sect. 3. Finally, Sect. 4 summarizes the workshop paper and gives an outlook for future work.

2. Foundations and Related Work

To conduct research with real production lines is often difficult due to safety concerns and industry secrets. For this reason, physical models that emulate real production environments find their way into research (e. g., [5, 20, 3, 21]). They enable the development and evaluation of research artifacts in a closed, protected environment but at much lower prices before transferring to real-world production [3]. In Sect. 2.1, we introduce our Fischertechnik physical smart factory for emulating production environments. In addition, we present how this physical factory can be controlled in a process-oriented way and how semantic annotations are used for process execution and for planning purposes. In Sect. 2.2, we present how experiential knowledge can be used for problem-solving in context of *Case-Based Reasoning (CBR)*. Finally, we discuss in Sect. 2.3 relevant related work.

2.1. Physical Smart Factory Model for Industry 4.0 Research

For conducting practice-oriented research in the field of Industry 4.0, physical smart factories can be used for the evaluation and demonstration of novel research artifacts, before implementing them in a complex real-world manufacturing environment [3, 5]. In our research [3, 19, 21, 5, 20], we use a *Fischertechnik (FT)*[1] smart factory model that consists of two similar shop floors connected for the exchange of workpieces, as shown in Fig. 1. There are four workstations on each shop floor with six identical machines: a sorting machine with color detection, a multi-processing workstation with an oven, a milling machine and a workstation transport connecting the two, a high-bay warehouse, and a vacuum gripping robot. In addition, there are individual machines on each shop floor, i. e., a punching machine and a human workstation on the first shop floor and a drilling machine on the second one. Several light barriers, switches, and capacitive sensors are installed for control purposes on each shop floor. The workpieces used for simulating the production are small cylindrical blocks (height = $\sim 1.4\,\mathrm{cm}$, diameter = $\sim 2.6\,\mathrm{cm}$) in white, red, or blue color. Each workpiece is equipped with an NFC tag that contains information about the individual workpiece such as an identifier, the current production state (i. e., color, position), and the production history with time stamps. There are RFID readers/writers integrated on both

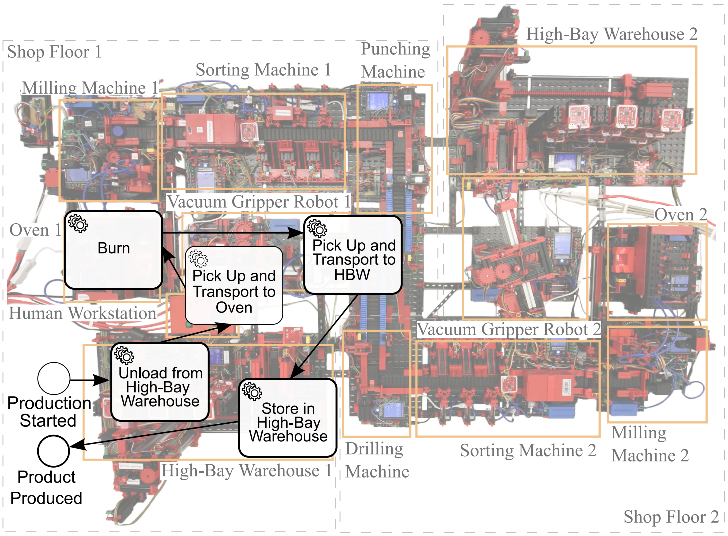

Figure 1: Workflow-based Control of the Fischertechnik Factory Model. (Based on: [5])

[1] Fischertechnik is a company that produces modules for emulating factories on a small scale. More general information can be found at https://www.fischertechnik.de/en/simulating/industry-4-0.

shop floors, creating 28 communication points. This enables tracking of each workpiece and retrieving the required manufacturing operations and parameters, which can be modified during production if necessary. A video of the smart factory executing a manufacturing workflow and tracking workpieces by an object-detection framework [21] can be found at https://iot.uni-trier.de.

In order to control actuators such as manufacturing resources and to process real-time IoT sensor data from the smart factory, the functionalities of actuators and sensors must be available in a coarse-grained manner at a higher level [3, 19, 5]. As a result, it is possible to react to events that occurred in the smart factory at higher level systems, e. g., in *Workflow Management Systems (WfMSs)*. In our previous work [3, 19, 5], we present a service-oriented architecture that enables us to control the smart factory in a process-oriented way. In Fig. 1, an exemplary sheet metal manufacturing workflow is shown as an overlay above the individual stations in the factory. In this workflow, an unprocessed steel slab is unloaded from the *High-Bay Warehouse (HBW)* and transported by the *Vacuum Gripper Robot (VGR)* to the oven in which the steel slab is burned and formed into a sheet metal. Afterwards, it is transported back to the HBW and stored. We use the *Camunda BPM Platform*[2] as WfMS to execute workflows in the factory. The workflows are modeled with *Business Process Model and Notation (BPMN)*[3] 2.0 compliant *Service Tasks* that invoke RESTful web services (see [19, 5] for more details). The web server in turn executes the corresponding method of the individual manufacturing resource in the factory. For example, if the service *Burn* is invoked by the WfMS, the corresponding method *Burn* is executed at the controller of the oven (see [5] for more details about the granularity of services). In addition, we semantically enrich each web service with semantic annotations [19] and connect them to the developed domain ontology FTOnto [20]. As a result, it is possible to check during workflow execution whether the preconditions are satisfied and whether the corresponding effects after execution have occurred. To enable the use of automated planning techniques, we convert the semantic web services into a formal planning domain description with the equivalent number of classical planning operators and one domain description with durative actions for temporal planning. By converting the semantic services automatically into a formal planning domain description, it is possible to remedy the time-consuming knowledge modeling effort that is typically needed to construct comprehensive planning domains.

2.2. Using Experience Knowledge for Adaptive Workflow Management

Process-Oriented Case-Based Reasoning (POCBR) [22] is a special kind of CBR that deals with the integration of CBR in *Process-Aware Information Systems (PAISs)* [6], e. g., WfMSs. A case in POCBR expresses procedural experiential knowledge, in our case represented as semantic workflow graphs, also called *NEST* graphs [22]. A *NEST* graph is a semantically annotated graph that consists of *N*odes, *E*dges, *S*emantic Descriptions for each node and edge, and *T*ypes of different nodes and edges. Figure 2 depicts the modeled BPMN process from Fig. 1 as a *NEST* graph enriched with semantic descriptions and with explicitly modeled data nodes. By using POCBR, it is possible to reuse experiential knowledge in the form of semantic workflow graphs in similar problem situations. However, instead of getting a case that satisfies the current

[2] https://camunda.com/
[3] https://www.omg.org/spec/BPMN/2.0.2/

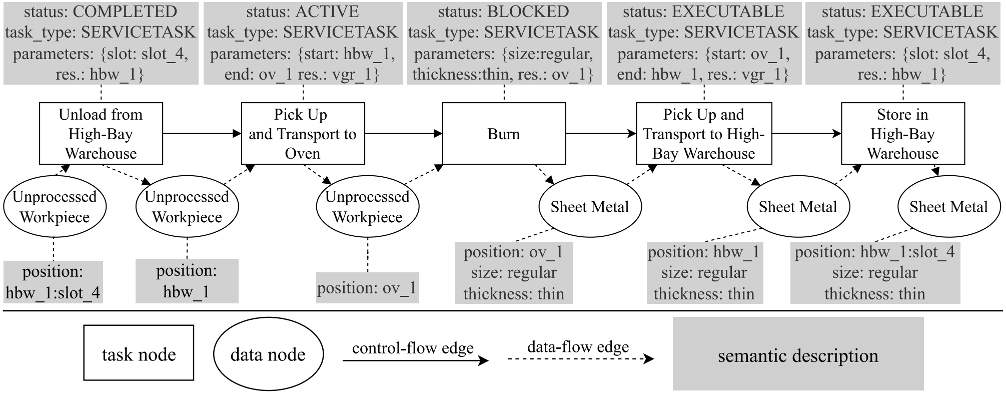
status: COMPLETED
task_type: SERVICETASK
parameters: {slot: slot_4, res.: hbw_1}
status: ACTIVE
task_type: SERVICETASK
parameters: {start: hbw_1, end: ov_1 res.: vgr_1}
status: BLOCKED
task_type: SERVICETASK
parameters: {size:regular, thickness:thin, res.: ov_1}
status: EXECUTABLE
task_type: SERVICETASK
parameters: {start: ov_1, end: hbw_1, res.: vgr_1}
status: EXECUTABLE
task_type: SERVICETASK
parameters: {slot: slot_4, res.: hbw_1}
Unload from High-Bay Warehouse
Pick Up and Transport to Oven
Burn
Pick Up and Transport to High-Bay Warehouse
Store in High-Bay Warehouse
Unprocessed Workpiece
Unprocessed Workpiece
Unprocessed Workpiece
Sheet Metal
Sheet Metal
Sheet Metal
position: hbw_1:slot_4
position: hbw_1
position: ov_1
position: ov_1 size: regular thickness: thin
position: hbw_1 size: regular thickness: thin
position: hbw_1:slot_4 size: regular thickness: thin
task node
data node
control-flow edge
data-flow edge
semantic description

Figure 2: Semantic Workflow Graph representing a Sheet Metal Manufacturing Process.

problem completely, it is sometimes only possible to retrieve similar solutions that cannot be used without any modifications to solve the current problem [10]. This is especially the situation in dynamic environments such as smart factories, where storing all possible cases in a case base is practically intractable and leads to unacceptable retrieval times [14]. There exist two main types of adaptation techniques in CBR that can be used to adapt the retrieved case to better suite the requirements of the current problem situation: *transformational* and *generative* adaptation [10]. *Transformational* adaptation modifies the retrieved case directly in order to be used for the current problem. *Generative* adaptation applies a knowledge-based problem solver, e. g., an AI planner, that can solve the problem, i. e., it can build a solution from scratch.

2.3. Related Work

In this section, we divided related work into two groups based on their main contribution: The first group consists of *Business Process Management (BPM)* approaches that are used in smart environments such as smart home [9] or smart health and emergency management [8] but all considering exceptions and deviations during workflow execution and thus provide a form of adaptive workflow management. The second group contains works that apply CBR to increase workflow adaptability (e. g., [7, 10, 23]) or that apply CBR to planning (e. g., [17, 12, 13, 18, 11]).

Adaptive Workflow Management in BPM and IoT: The *PROtEUS* system by Seiger et al. [9] enables the execution and adaptation of cyber-physical workflows in smart homes. A resource-based adaptation is applied to search for similar replacement resources in case of exceptions. The *SmartPM* system by Marrella et al. [8] uses automated planning techniques to adapt emergency management processes in which unanticipated exceptions occurred. The adaptation of the process resolves the exception by a sequence of actions, i. e., a plan.

Case-Based Reasoning Methodology for Adaptive Workflow Management: POCBR can be applied to reuse procedural experiential knowledge (see Sect. 2.2). Such experiential knowledge from a case base filled with best-practice workflows can be used to increase workflow adaptability and thus builds the basis for adaptive workflow management [23]. Müller [10] presents three non-generative experience-based adaptation methods in which adaptation knowledge is inductively learned from the case base and afterwards applied to a workflow. Thus, it is possible to adapt workflows to resolve current exceptional situations. Müller demonstrates the approach in the context of cooking recipes in order to adapt the workflow to the user's taste. The used cooking recipes are driven by their defined control-flow, i. e., their execution order. In contrast to control-flow workflows, Zeyen et al. [24] modify these adaptation methods to be applicable for data-driven data mining workflows. It is important to note that the *NEST* graph in Fig. 2 is also a kind of data-driven workflow, since data nodes are explicitly modeled for state changes of products even if no data is processed directly. Similarly to these approaches, Weber et al. [7] present the *CBRFlow* system for adaptive workflow management by using conversational CBR. If changes to a workflow become necessary due to exceptions or environmental changes and the deviation is not defined in the workflow model, the user adds a case to the case base by answering corresponding questions. The case from the case base can be retrieved in future and describes how the situation can be handled, e. g., by skipping a task, if this or a similar situation occurs again.

Another major branch of work that applies CBR in combination with automated planning is *Case-Based Planning (CBP)* [14, 15, 16]. In CBP, a case specifies a plan and sometimes additional information, e. g., a reasoning trace, that can be reused in similar problem-solving situations instead of planning from scratch. A plan as a sequence of actions is very similar to a workflow that is composed of activities that should be executed by a WfMS. Thus, also CBP approaches that adapt retrieved plans are similar to the previously described workflow adaptation techniques. Veloso [13] present the *PRODIGY/Analogy* system that uses derivational analogy for plan adaptation. In the approach, reasoning traces of plan generations are stored in the case base besides the plans themselves in order to be able to guide the search of the planner in a new problem-solving situation. Ros et al. [18] present a case-based approach to select actions during robot soccer. In their approach, they do not adapt the sequence of actions for the robot team as a solution of the retrieved case, but adapt the current problem, e. g., the positions of the involved robots, so that the real-world problem match the problem in the retrieved case. Muñoz-Avila et al. [12] present the *SiN* system that uses a conversational CBR component if it is not possible to decompose tasks by hierarchical planning due to an incomplete planning domain description. The *PARIS* system proposed by Bergmann [17] is a case-based planner that reuses cases at different levels of abstraction. If a new problem situation occurs, the system retrieves the best-matching case at the lowest possible level of abstraction. A generative planner replaces remaining abstract operators with specific actions that can be executed in the real world. *OAKPlan* introduced by Serina [11] applies kernel functions during plan retrieval to rapidly find similar and well-suited plans as a basis for reuse. A generative planner is used for modifying the plan to better reflect the current problem situation.

3. Adaptive Workflow Management by Case-Based Reasoning and Automated Planning

In this section, we present an architectural framework for adaptive workflow management in smart environments and, in this context, an idea for an adaptation method that combines CBR and automated planning. Figure 3 depicts the architecture that is described in more detail in the following. At the top of the framework, a state-of-the-art WfMS is used. The

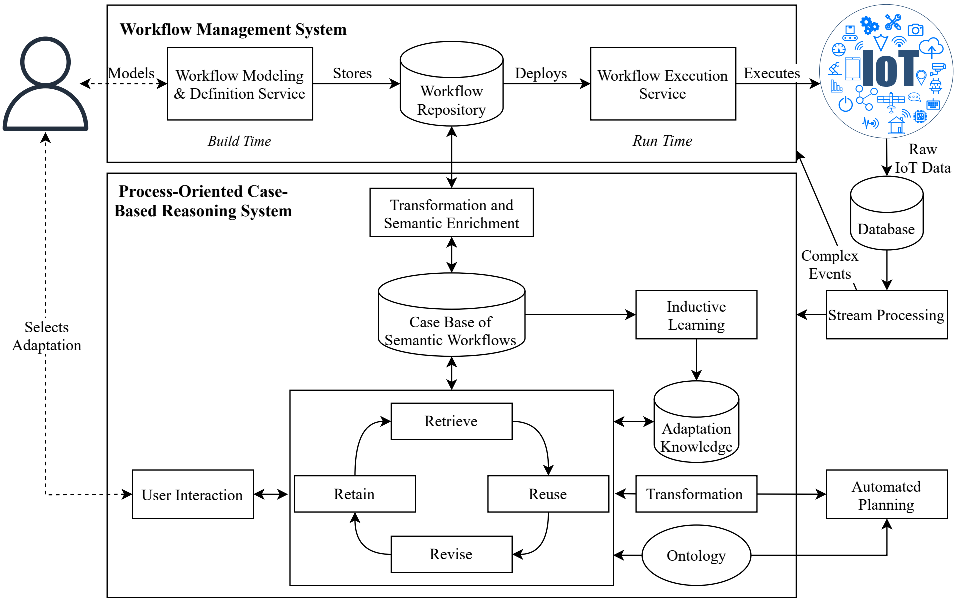

Figure 3: Architecture for Adaptive Workflow Management by CBR and Automated Planning. (Based on: [25])

expert can model manufacturing workflows by using a corresponding service, i. e., the *Workflow Modeling & Definition Service*. Afterwards, the workflow is stored in a workflow repository. If a production order is received, the corresponding workflow from the workflow repository is retrieved, deployed, and executed in the smart factory (cf. Sect. 2.1) by the *Workflow Execution Service*. During execution, the smart factory generates raw IoT sensor data that is passed to a database and processed immediately by a *Stream Processing* engine. By using a stream processing engine, it is possible to develop queries that send higher level events to other systems if certain patterns are detected in the sensor data [3, 5]. In our case, the WfMS receives events for workflow execution and the POCBR system ProCAKE[4] [26] is informed about the current state of the currently executed workflows in the smart factory. In order to check the execution of the modeled workflows in the smart factory, the BPMN 2.0 workflows are semantically enriched and transformed into their corresponding *NEST* graph representation (cf. Sect. 2.2). To enable the use of experience-based adaptation methods [10], inductive learning methods are applied to the workflows in the case base to automatically learn *Adaptation Knowledge*. During execution of a workflow in the smart factory, ProCAKE is notified by the stream processing engine when activities of workflows have been completed successfully or when this is not the case and unexpected exceptions have occurred. This enables a more comprehensive monitoring of the workflow execution besides the monitoring in the WfMS. In case of an exception, e. g., due to a machine breakdown, ProCAKE receives a notification of the stream processing engine and starts the 4R (*R*etrieve, *R*euse, *R*evise, *R*etain) cycle in the POCBR system.

Retrieve: The notification from the stream processing engine is analyzed and a query that captures the current problem in the smart factory is automatically generated. The query contains

[4] https://procake.uni-trier.de

the state of the currently executed workflow and the remaining activities of the workflow as well as further metadata, e. g., which machines are currently available in the smart factory or defect. Afterwards, a retrieval is performed to determine if a similar problem situation has already occurred in the past. During the retrieval, a graph matching based on a semantic similarity measure [22] is performed to find the best-matching case in the case base that corresponds to the currently executed workflow and its metadata.

Reuse: It is rather unlikely to retrieve a case that completely solves the current problem, i. e., to resolve the exceptional situation and, thus, to continue workflow execution to achieve the workflow goal. Therefore, adaptation methods are required that modify the retrieved case to better suite the current problem (cf. Sect. 2.2). For this purpose, the adaptation methods proposed by Müller [10] can be transferred and applied to the manufacturing domain similar to the transformation performed by Zeyen et al. [24]. Since it is difficult to physically change the real world in production, i. e., the state of the workflows and the positions of the workpieces or the current state of machines, an approach as presented by Ros et al. [18] cannot be used. In contrast, a solution adaptation in which the retrieved workflow is adapted to resolve the exception should be used and, thus, allow the workflow to be continued. For example, if the oven on the first shop floor is broken, the manufacturing workflow as depicted in Fig. 2 cannot be continued as planned. To further execute the workflow and to achieve the overall workflow outcome, i. e., the final product, an alternative resource that can perform the *Burn* activity should be searched and used. The adaptation in the *Reuse* phase applies the learned adaptation knowledge and attempts to resolve the exception. Afterwards, a validator checks and determines for safety reasons whether the adapted workflow is executable in the smart factory. This may not always be the case, as a sufficient number of cases is required to learn appropriate adaptation knowledge automatically by using the stored cases in the case base [10]. Even if a sufficient number of cases is available, not all situations in dynamic environments can be captured by experienced cases. For this reason, a generative planner to support the adaptation in scenarios where the experience-based adaptation does not provide a full and appropriate solution should be used. In addition, this synergistic method is appropriate, since problem-solving from scratch is sometimes difficult w. r. t. the computation time and the effort required to model a comprehensive planning domain. By using the combined method, the whole adaptation problem can be divided into smaller sub-problems (*divide and conquer*) that are only partly solved by AI planning to fix the remaining *gaps* in the adapted workflow. For example, in the case of a breakdown of the oven on the first shop floor, the POCBR system might provide a case that is similar to the current problem but instead of being executed on the first production line, it is executed on the second one. In this case, using the oven on the second shop floor solves the problem but the transport routes to the oven are not yet included in the adapted workflow. If the learned adaptation knowledge contains these transport routes, they can be added by the POCBR system during adaptation. Otherwise, the validator determines that there are misalignments w. r. t. the shop floor positions in the adapted workflow and AI planning is started with the corresponding initial state and the desired goal state to resolve these misalignments (cf. *Transformation* in Fig. 3). In the complete adaptation process, the modeled knowledge in the domain ontology FTOnto [20] about the production environment is essential for the POCBR system but also for using AI planning. The advantage of the used transformational adaptation compared to a derivational adaptation as in [13] is that state-of-the-art AI planners can be used without major modifications to obtain their reasoning traces. This also permits to swap the planners flexibly or to use several planners or planning configurations, e. g., by changing the used heuristics, in a planning ensemble. In the whole adaptation procedure, it is also possible to include users in the loop and to let them participate by selecting adaptations by using an interactive approach (cf. [25, 7, 12]).

Revise: In this phase, the adapted workflow is checked for applicability in the smart factory. As a first step, the validator is used to check the syntactic and semantic correctness of the adapted workflow. In addition, the adapted workflow can be presented to a domain expert, who must confirm the performed adaptations before the workflow is further executed. Afterwards, the modified workflow is continued in the smart factory. In case exceptions occur again in the continued workflow, the CBR cycle is again triggered to resolve them.

Retain: In the final *Retain* phase, it is checked whether the adapted workflow should be stored as a learned problem situation in the case base. This is especially useful if, for example, the exception could only be solved with the help of the generative planner or by a human. In this case, the competence of the POCBR system can be increased for future problem situations. Consequently, the POCBR system can learn for future problem situations by incorporating knowledge from AI planning. In contrast, the planner's domain description can be extended by case-specific expert knowledge to complement incomplete planning domains (cf. [12]).

4. Summary and Outlook

We present a first step towards adaptive workflow management for IoT-enhanced cyber-physical manufacturing workflows by using CBR and automated planning in a synergistic approach. We show how such an approach can be realized in the 4R CBR cycle. Relevant related work in the area of IoT and business process management and several approaches using case-based reasoning or case-based planning is presented. However, most of the approaches are not applied in the context of Industry 4.0 or evaluated in physical real-world environments. In this context, the combination of problem-solving and (workflow/plan) execution is an interesting research topic. For our research, we use a physical Fischertechnik smart factory that allows to conduct such experiments in protected but close-to-reality environment.

In future work, we want to further implement and validate the proposed adaptive workflow management framework. For this purpose, we currently implement the experience-based adaptation methods by Müller [10] for the manufacturing domain. In addition, we plan to conduct extensive experimental evaluations with the physical smart factory model.

References

[1] M. Rüßmann, et al., Industry 4.0: The future of productivity and growth in manufacturing industries, Boston Consulting Group 9 (2015) 54–89.

[2] L. Monostori, Cyber-physical Production Systems: Roots, Expectations and R&D Challenges, Procedia CIRP 17 (2014) 9–13.

[3] L. Malburg, et al., Using Physical Factory Simulation Models for Business Process Management Research, in: BPM Workshops, volume 397 of *LNBIP*, Springer, 2020, pp. 95–107.

[4] D. A. Rossit, et al., Production planning and scheduling in Cyber-Physical Production Systems: a review, Int. J. Computer Integr. Manuf. 32 (2019) 385–395.

[5] R. Seiger, et al., Integrating process management and event processing in smart factories: A systems architecture and use cases, J. Manuf. Syst. 63 (2022) 575–592.

[6] M. Reichert, B. Weber, Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies, Springer, 2012.

[7] B. Weber, W. Wild, R. Breu, CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning, in: 7th ECCBR, volume 3155 of *LNCS*, Springer, 2004, pp. 434–448.

[8] A. Marrella, M. Mecella, S. Sardiña, Intelligent Process Adaptation in the SmartPM System, ACM Trans. Intell. Syst. Technol. 8 (2017) 25:1–25:43.

[9] R. Seiger, et al., Toward a framework for self-adaptive workflows in cyber-physical systems, Softw. Syst. Model. 18 (2019) 1117–1134.

[10] G. Müller, Workflow Modeling Assistance by Case-based Reasoning, Springer, 2018.

[11] I. Serina, Kernel functions for case-based planning, Artif. Intell. 174 (2010) 1369–1406.

[12] H. Muñoz-Avila, et al., SiN: Integrating Case-based Reasoning with Task Decomposition, in: 17th IJCAI, Morgan Kaufmann, 2001, pp. 999–1004.

[13] M. M. Veloso, Planning and Learning by Analogical Reasoning, volume 886 of *LNCS*, Springer, 1994.

[14] R. Bergmann, H. Muñoz-Avila, M. M. Veloso, E. Melis, CBR Applied to Planning, in: Case-Based Reasoning Technology, volume 1400 of *LNCS*, Springer, 1998, pp. 169–199.

[15] M. T. Cox, et al., Case-based planning, Knowledge Eng. Review 20 (2005) 283–287.

[16] D. Borrajo, A. Roubícková, I. Serina, Progress in Case-Based Planning, ACM Comput. Surv. 47 (2014) 35:1–35:39.

[17] R. Bergmann, Effizientes Problemlösen durch flexible Wiederverwendung von Fällen auf verschiedenen Abstraktionsebenen, volume 138 of *DISKI*, Infix Verlag, 1996.

[18] R. Ros, J. L. Arcos, R. López de Mántaras, M. M. Veloso, A case-based approach for coordinated action selection in robot soccer, Artif. Intell. 173 (2009) 1014–1039.

[19] L. Malburg, et al., Semantic Web Services for AI-Research with Physical Factory Simulation Models in Industry 4.0, in: 1st IN4PL, ScitePress, 2020, pp. 32–43.

[20] P. Klein, L. Malburg, R. Bergmann, FTOnto: A Domain Ontology for a Fischertechnik Simulation Production Factory by Reusing Existing Ontologies, in: 21st LWDA, volume 2454, CEUR-WS.org, 2019, pp. 253–264.

[21] L. Malburg, et al., Object Detection for Smart Factory Processes by Machine Learning, Procedia Comput. Sci. 184 (2021) 581–588.

[22] R. Bergmann, Y. Gil, Similarity assessment and efficient retrieval of semantic workflows, Inf. Syst. 40 (2014) 115–127.

[23] M. Minor, et al., Case-based adaptation of workflows, Inf. Syst. 40 (2014) 142–152.

[24] C. Zeyen, et al., Adaptation of Scientific Workflows by Means of Process-Oriented Case-Based Reasoning, in: 27th ICCBR, volume 11680 of *LNCS*, Springer, 2019, pp. 388–403.

[25] C. Zeyen, R. Bergmann, Towards an Interactive Workflow Modeling Assistance by Means of Case-Based Reasoning, in: 20th LWDA, CEUR-WS.org, 2018, pp. 355–360.

[26] R. Bergmann, et al., ProCAKE: A Process-Oriented Case-Based Reasoning Framework, in: ICCBR Workshop, volume 2567, CEUR-WS.org, 2019, pp. 156–161.