Adaptive Management of Cyber-Physical Workflows by Means of Case-Based Reasoning and Automated Planning*

Lukas Malburg[1,2], Florian Brand[1], and Ralph Bergmann[1,2]

[1] Artificial Intelligence and Intelligent Information Systems, University of Trier, 54296 Trier, Germany, http://www.wi2.uni-trier.de
[2] German Research Center for Artificial Intelligence (DFKI)
Branch University of Trier, Behringstraße 21, 54296 Trier, Germany
{malburgl,s4fnbran,bergmann}@uni-trier.de;
{lukas.malburg,ralph.bergmann}@dfki.de

Abstract Today, it is difficult for companies to react to unforeseen events, e. g., global crises. Highly standardized manufacturing processes are particularly limited in their ability to react flexibly, creating a demand for more advanced workflow management techniques, e. g., extended by artificial intelligence methods. In this paper, we describe how *Case-Based Reasoning (CBR)* can be combined with automated planning to enhance flexibility in cyber-physical production workflows. We present a compositional adaptation method complemented with generative adaptation to resolve unexpected situations during workflow execution. This synergy is advantageous since CBR provides specific knowledge about already experienced situations, whereas planning assists with general knowledge about the domain. In an experimental evaluation, we show that CBR offers a good basis by reusing cases and by adapting them to better suit the current problem. The combination with automated planning further improves these results and, thus, contributes to enhance the flexibility of cyber-physical workflows.

Keywords: Case-Based Reasoning · Automated Planning · Industry 4.0 · Adaptive Workflow Management · Cyber-Physical Workflows

1 Introduction

Recently, global crises have shown that manufacturing processes and in general supply chains cannot easily be adapted to respond to unforeseen and dynamic events. This is among others because manufacturing processes are often highly standardized and therefore only provide a limited degree of flexibility [12, 16]. One of the goals of the *Fourth Industrial Revolution (Industry 4.0)* [12] is to enhance this limited flexibility by applying *Artificial Intelligence (AI)* methods [13]

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-031-26886-1_5

in *Cyber-Physical Production Systems (CPPSs)* [20]. Consequently, failure situations that occur during production should be resolved in an autonomous, self-adaptive way, strengthening resilience of workflows against failures and unexpected situations during their execution [15, 25]. To adapt workflows automatically and, in turn, to ensure their continuing execution in problem situations, current research applies search-intensive approaches such as AI planning (e. g., [7,17,24]) and knowledge-intensive approaches such as *Case-Based Reasoning (CBR)* [1] (e. g., [19,21,28,29]). However, search-intensive techniques require fully observable planning domain descriptions, which are rather rare and difficult to obtain in real-world applications, to generate appropriate solutions [22, 30]. In addition, AI planning is sometimes not applicable for large problems due to the high computational complexity [4, 5, 27]. To remedy these issues, the combination of AI planning and CBR offers significant potential for improvement, leading to research directions such as *Case-Based Planning* [4,5,27,30] in which plans are reused in similar situations. CBR, in this regard, provides specific experience knowledge that can be utilized in similar problem situations, limiting the knowledge acquisition and modeling effort for a comprehensive planning domain. However, today, existing approaches are not examined for production planning or for advanced adaptive workflow management in which currently either pure planning techniques or pure CBR methods are applied (e. g., [17, 18, 28, 29]). For this purpose, this paper presents application scenarios in which *Case-Based Reasoning (CBR)* [1] can contribute to enhance the flexibility of cyber-physical manufacturing workflows. Thereby, we focus on automatic workflow adaptations in the *Reuse* phase to resolve unexpected failure situations that can occur during workflow execution (see [15] for the architectural framework). Compositional adaptation is used with AI planning to overcome the outlined disadvantages and to apply both methods in a synergistic way. A key advantage of our approach is that it limits the knowledge acquisition and modeling effort typically required for creating comprehensive planning domains by incorporating experiential knowledge during problem-solving. To evaluate the approach, we use a physical smart factory model from *Fischertechnik (FT)*, which enables us to conduct laboratory experiments while maintaining real world environmental conditions of production lines. In the following, Sect. 2 presents the used physical smart factory and describes application scenarios in which CBR can contribute to enhance flexibility of production workflows. Our approach for automatic workflow adaptation by combining compositional adaptation and AI planning is presented in Sect. 3. To measure the effectiveness and suitability of the approach, we present an experimental evaluation in Sect. 4. Finally, Sect. 5 summarizes the paper and gives an outlook on future research directions.

2 Foundations and Related Work

We describe the main characteristics of cyber-physical workflows and present the used smart factory model in Sect. 2.1. Afterwards, we discuss in Sect. 2.2 related work using AI-based methods for adaptive workflow management. Finally, Sect. 2.3 introduces *Process-Oriented Case-Based Reasoning (POCBR)* [3] as a special kind of CBR for reusing procedural experiential knowledge. In addition, application scenarios in which POCBR can contribute to enhance flexibility in cyber-physical workflows are described.

2.1 Cyber-Physical Workflows and Physical Smart Factory

Cyber-Physical Workflows (CPWs) [18, 25] are a new branch of workflows in which the presence of *Internet of Things (IoT)* technologies influences the execution of workflows in the real world and vice versa. For example, actuators such as machines are used to execute tasks in the environment and sensor data from IoT devices can be used for guiding workflow execution or to detect failures during production. Based on these detected situations, AI techniques can be applied to resolve problems and to continue workflow execution in the physical world [16, 17]. These advantages can be achieved for workflow management by exploiting the bidirectional coupling between process-based systems and the smart environment [26] and, thus, by using the variety of IoT sensor data from it. The environment itself benefits by using well-established methods from workflow management research [10]. However, to profit from these advantages, several challenges must be addressed (cf. [10]), which is why the use of advanced AI methods in this area is still in its infancy.

Using IoT environments such as real manufacturing shop floors for research purposes poses many difficulties [16]. Thus, small-scale physical models can be used for executing manufacturing workflows. We use a smart factory model from *Fischertechnik (FT)* to conduct process-oriented research for Industry 4.0 [14,16, 26]. The custom model[1] emulates two independently working production lines consisting of two shop floors that are linked to exchange workpieces (see Fig. 1). Each production line consists of six identical machines. In addition, there are individual machines on each shop floor, i. e., a *Punching Machine (PM)* and a *Human Workstation (HW)* on the first shop floor and a *Drilling Machine (DM)* on the second one. For control purposes, there are several light barriers, switches, and capacitive sensors on each shop floor. The workpieces used for simulating manufacturing are small cylindrical blocks. Each workpiece is equipped with an NFC tag with information about the individual workpiece such as the current production state and the production history with time stamps.

2.2 AI-Based Methods for Adaptive Workflow Management

To enhance workflow flexibility by automatic workflow adaptations, two types of situations [17] that can occur during execution must be handled: *expected* and *unexpected* situations. Whereas expected situations that are known in advance can be handled by appropriate exception handling techniques (cf. [23]), unexpected

[1] More information about the smart factory model and a video can be found at https://iot.uni-trier.de.

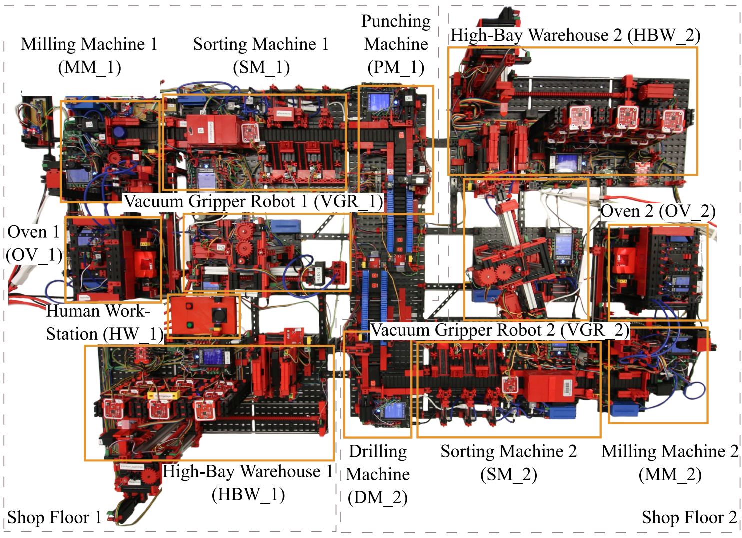


Fig. 1. The Physical Fischertechnik Smart Factory Model [16].

situations require *ad-hoc changes* during runtime and, thus, more advanced techniques. The *ADEPT* framework by Dadam and Reichert [6] is one of the earliest and most prominent approaches for increasing process flexibility via ad-hoc changes of running processes. Thereby, monitoring for process failures and adapting a process is mainly performed manually. If an error occurs, the user selects a suitable ad-hoc change pattern such as inserting and deleting tasks or changing their execution order. To enable fully automatic ad-hoc changes, AI-based methods, which can be divided into *search-intensive* and *knowledge-intensive* [27], can be applied (e. g., [7, 17–19, 24, 28, 30]): Search-intensive techniques aim at finding a solution in the search space, which is usually implemented by using AI planning. In addition to the high search effort that is needed to generate solutions for large problems, a full and comprehensive planning domain description is required for problem-solving. However, such comprehensive planning descriptions are rather rare in real-world application domains. In addition, planning domain descriptions are sometimes sparse and only incomplete domain models with partial knowledge are available for planning [22, 30]. Knowledge-intensive methods such as CBR use experiential knowledge, e. g., gained from employees working on the shop floor, to generate solutions. This significantly reduces the effort required to solve problems but also means that sufficient experiential knowledge must be available. This is especially difficult in dynamic IoT environments, where many unexpected problem situations can occur. Consequently, the resulting adapted workflows are sometimes not executable and require additional manual adjustments by users (e. g., [21, 29]). Even though the individual methods provide good adaptation results, executability and correctness of adapted cyber-physical workflows are important as improperly configured and adapted workflows can cause considerable damage. To overcome the weaknesses of the methods in this regard, the combination of search-intensive and knowledge-intensive techniques promises valuable advantages [4, 5, 27, 30].

2.3 Process-Oriented Case-Based Reasoning for Cyber-Physical Workflows

In our research, we apply *Process-Oriented Case-Based Reasoning (POCBR)* [3] as a knowledge-intensive method for advanced workflow management. POCBR integrates *Case-Based Reasoning (CBR)* [1] with process-aware information systems [23] such as workflow management systems or enterprise resource planning systems. A case in POCBR expresses procedural experiential knowledge gained in previous problem-solving situations, and the case base consists of best-practice workflows for reuse. To represent procedural experiential knowledge, we use a semantic workflow graph representation called *NEST graph* introduced in [3]: A NEST graph is a quadruple $G = (N, E, S, T)$ where N is a set of nodes and $E \subseteq N \times N$ represents the edges between nodes. Semantic descriptions S can be used for semantic annotations of individual nodes or edges. T specifies the type of the node or edge.

An exemplary cyber-physical manufacturing workflow represented in the *NEST* graph format is depicted in Fig. 2. It represents a sheet metal production process that can be physically executed in the smart factory model[2] (see Sect. 2.1). We use these kinds of manufacturing workflows because they are well suited for the factory layout used, and they are highly customized for a client, which also implies increased flexibility during execution. However, the used sheet metal workflows are placeholders for other arbitrary industrial processes. In the shown workflow, an unprocessed steel slab is unloaded from the high-bay warehouse and transported to the oven. In the oven, the steel slab is burned and rolled into a thick, middle-sized sheet metal. Afterwards, the processed sheet metal is transported to and stored in the high-bay warehouse.

In cyber-physical workflows, task nodes ($TN \subseteq N$) denote the production steps that are executed by actuators in the physical IoT environment. The semantic descriptions of task nodes can be used to further describe the properties of each activity, e. g., the concrete machine parameters. In the illustrated workflow, the *Burn* task is configured by the size and thickness parameters to produce the required sheet metal. In addition to this, the state of the task is captured in the semantic description (*COMPLETED*, *ACTIVE*, *EXECUTABLE*, *FAILED*, or *BLOCKED*). A task is blocked or fails during execution if the IoT resource needed to perform the activity is not functional, e. g., due to a defect. Data nodes ($DN \subseteq N$) can be consumed or produced by task nodes. Data-flow edges ($DE \subseteq E$) represent a consumption of a data node with an ingoing edge to the task node and a production of a data node by an outgoing one. A data node represents the state of the workpiece during this point in the manufacturing

[2] More information about the execution of workflows in the Fischertechnik smart factory model can be found in [14, 16, 26].

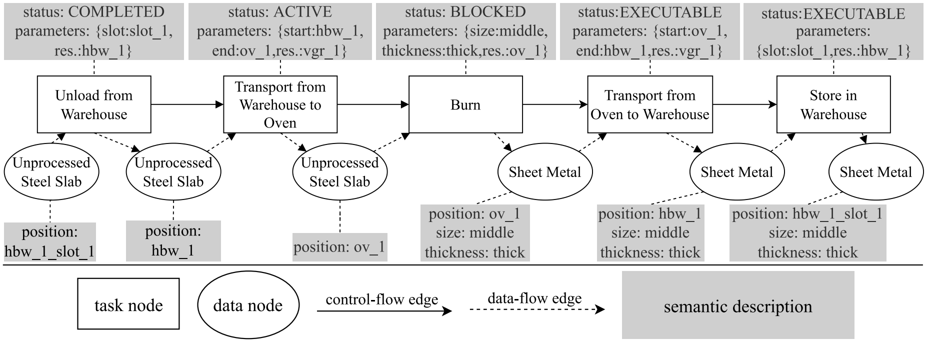


Fig. 2. Graph Representation of a Cyber-Physical Workflow.

workflow. The semantic description of a data node features the properties of the workpiece, e. g., the position of the workpiece and the production properties such as size and thickness. Consequently, state changes of a single workpiece are represented by data nodes in the context of the overall workflow (similar to the usage of data nodes in [29]). Thus, a task node can only be executed after a previous task has produced the required properties. In this context, the control-flow can be derived from the data-flow by using control-flow edges ($CE \subseteq E$) that are specified between task nodes. One important aspect of cyber-physical workflows is the point of execution: As they are workflows executed by actuators in a physical IoT environment, the workflows need to be safely executable. For the smart factory used, it means that the machines need to be active and the activities represented by task nodes need to be configured correctly. Additionally, the state of the workpiece must be a valid input for the executing task. For example, the drilling machine can only drill holes into the workpiece if it is a sheet metal and not an unprocessed steel slab. In addition, the workpiece must be located at the drilling machine.

Application Scenarios for Cyber-Physical Workflows One application scenario of POCBR for cyber-physical workflows is the retrieval of suitable workflows for execution based on a product specification and the currently available production capacities. For example, a client can specify the properties of the desired product in an order. These requirements are then used as a query to retrieve a suitable production workflow that can satisfy it. Afterwards, the workflow can be executed by a workflow management system. A further application scenario for using POCBR for cyber-physical workflows is the topic of workflow adaptation in the *Reuse* phase (see [15] for a generic architectural framework). Much of the current work in POCBR deals with retrieving similar workflow cases (e. g., [3, 9, 11]) and only few approaches (e. g., [19, 21, 28, 29]) investigate the complex topic of automatic adaptation of retrieved workflows. Assume that the exemplary manufacturing workflow depicted in Fig. 2 is currently executed in the smart factory (see Sect. 2.1). During the execution, a failure due to a defect of the required oven occurs which leads to the task being blocked and not executable, i. e., the production cannot be continued. The goal of using POCBR in this scenario is to retrieve a case where a similar situation occurred previously, involving similar states of the machines and a similar not executable workflow. This retrieved case builds the basis for solving the current problem. The stored solution in the case is a *Change Plan* [17] that, when applied to the non-executable workflow, recovers the workflow to be further executed and, thus, to produce the final product. However, it is rather unlikely that a retrieved case solution solves the problem completely (see Sect. 2.2). For example, a similar case for the given problem is retrieved in which the oven in the second production line (*OV_2*) is used as alternative. However, the transports from the current position to the second oven and back are not contained. In these cases, the change plan should be adapted to better suit the current problem situation.

3 Adaptive Management of Cyber-Physical Workflows by CBR and Automated Planning

Based on the presented application scenarios of POCBR for cyber-physical workflows, we present our approach for combining experience-based adaptation by POCBR with automated planning for adaptive management of cyber-physical workflows in this section. The approach can be applied in the *Reuse* phase in the generic architectural framework presented in [15]. First, we examine how the compositional adaptation with workflow streams [21] can be used for cyber-physical workflows (see Sect. 3.1). Afterwards, we describe how experience-based adaptation can be enhanced by using AI planning (see Sect. 3.2).

3.1 Compositional Adaptation with Workflow Streams for Cyber-Physical Workflows

Compositional adaptation by using *Workflow Streams* [21] decomposes a workflow into smaller suitable sub-workflows, each of which produces a *partial workflow output* that is essential for achieving the overall workflow goal. More precisely, partial workflow outputs are intermediate steps in the workflow that are combined for achieving the final workflow goal such as the end product. Manufacturing workflows often consist of such intermediate produced subcomponents that are finally combined into an end product. Each partial output and, thus, each workflow stream represents a self-contained part of the workflow that can be replaced by another stream, e. g., with a different task sequence or parameterization, but producing the same output during adaptation. Learned streams from several previously experienced cases can be used for: 1) replacing streams in the case solution, 2) adding new streams to the case solution, or 3) deleting not needed streams in the case solution. The goal of adaptation is to modify the change plan, i. e., the solution in the retrieved case, in such a way that, on the one hand, the workflow goal of the current problem workflow is still reached and, on the other hand, only currently functional machines are used (see Sect. 2.3). In the following, we describe how the compositional adaptation method works in detail for cyber-physical workflows. Thereby, we distinguish between the automatic learning of workflow streams and applying them during adaptation in the *Reuse* phase.

Learning Workflow Streams Each workflow stream produces a partial output of the workflow, i. e., a data node called *creator data node*. The task node that produces this special data node is called *creator task* and also marks the end of each workflow stream. In contrast to the native approach by Müller and Bergmann [21] in which creator tasks are defined by using the data-flow edges in the graph, this definition cannot be used for cyber-physical workflows. This is because data nodes in cyber-physical workflows are used to represent the state changes of the workpieces (see Sect. 2.3). Definition 1 specifies the modified definition for creating appropriate creator tasks in cyber-physical workflows:

Definition 1 *A task node t is a creator task ct, iff it adds at least one new property to the manufactured product. In the graph representation used, a new property can be determined by a larger number of attributes in the semantic description of the produced data node, i. e., the creator data node, compared to the consumed data node. The set of creator tasks CT is defined as follows:*

$$CT = \{t \in TN \mid \exists d_1, d_2 \in DN : ((d_1, t) \in DE \wedge (t, d_2) \in DE) \wedge |S(d_1)| < |S(d_2)|\}$$

Figure 3 depicts a manufacturing workflow with its marked creator tasks ($\odot$) and corresponding workflow streams. The tasks *Burn*, *Deburr*, and *Drill* are creator task nodes since they add a relevant property to the produced workpiece. Following Def. 1, the data nodes produced by the creator tasks have a larger number of attributes in their semantic descriptions than the previously consumed data nodes. For example, the *Burn* activity adds a concrete size and thickness of the produced sheet metal to the state of the workpiece. After specifying the tasks in the workflow that are creator tasks, the workflow can be partitioned into a set of workflow streams with the restriction that each task node $t \in TN$ is exclusively assigned to one workflow stream. The streams are constructed by applying the following rules [21, 29]: A task node $t \in TN \setminus CT$ is assigned to a stream WS, 1) iff t is executed before the creator task $ct \in CT$ in the workflow, 2) iff t is

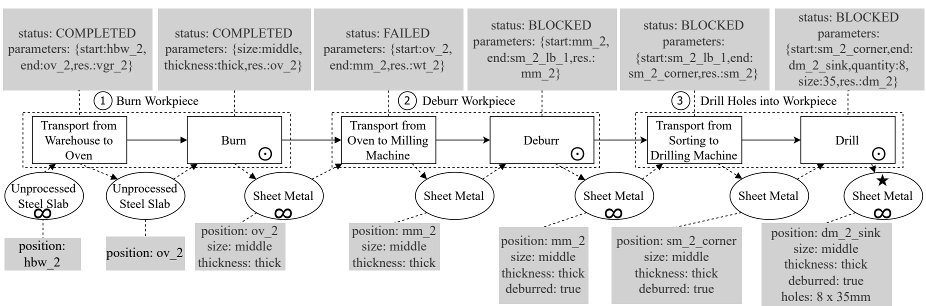
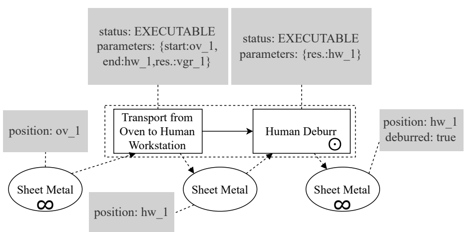


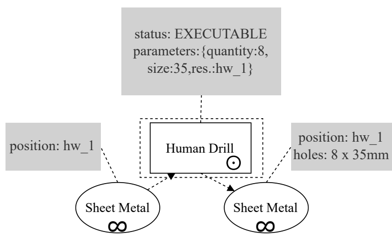
Fig. 3. Problem Manufacturing Workflow As Part of the Query.

not already contained in another stream, and 3) iff t is directly or transitively data-flow connected (cf. [21]) to the creator task $ct \in CT$. For example, the *Transport from Oven to Milling Machine* task is executed before the creator task *Deburr*, it is not already assigned to a stream, and there exists a data node that connects the *Transport from Oven to Milling Machine* task directly with the *Deburr* creator task (see Fig. 3). However, it could also be the case that several normal tasks are between this task node and the next creator task. In this case, the data-flow connectedness is transitively given.

Applying Workflow Streams After the partitioning of the workflows in the case base and the construction of workflow streams, the learned streams stored in a stream repository can be applied in the *Reuse* phase. During adaptation, streams in the retrieved case solution are replaced or deleted, or new streams are added. The goal of the adaptation is to modify the case solution in such a way that it resolves the current problem by continuing workflow execution. More precisely, the adapted change plan leads to the fact that the currently not executable workflow can be continued while retaining the workflow goal (see Sect. 2.3). As an example, assume that the illustrated manufacturing workflow in Fig. 3 is currently executed in the smart factory. During production, a failure occurs at the *Transport from Oven to Milling Machine* task since the workstation transport machine in the second shop floor is not functional. After detecting this problem situation, the statuses of all tasks are updated. During this process, it is determined that the following tasks are blocked since the milling machine, the sorting machine, and the drilling machine in the second shop floor also cannot be used. A query consisting of the problem workflow graph and the described factory states is generated and a retrieval for a similar problem situation is performed.



(a) Change Plan As Case Solution From the Retrieved Case.



(b) Learned Workflow Stream.

Fig. 4. Compositional Adaptation of Cyber-Physical Workflows.

Figure 4a illustrates the change plan as case solution from the most similar retrieved case. Instead of using the currently defective milling machine, the change plan uses the human workstation, which is currently ready to deburr the workpiece. Although no defective machines are used anymore, inserting this change plan into the currently non-executable workflow and, thus, replacing streams 2 and 3 would not lead to the desired workflow goal marked with ★ in the problem workflow. For this reason, the stream repository is searched for further streams that are executable based on the current machine states and that generate the goal data node of the problem workflow. Figure 4b depicts a workflow stream learned from another case from the case base. Adding this stream to the retrieved change plan fully satisfies the adaptation goal. For this reason, it is added to the change plan and the modified change plan replaces the currently non-executable parts, i. e., streams 2 and 3, in the problem workflow. Adding a stream to the change plan requires maintaining the connection to the other nodes in the change plan. The *creator data node* is the final partial output of a stream. In addition to this output data node, the data node consumed by the first task in each stream is also marked with ∞ since it is required that this input must match with the output of the previous stream during insertion. These data nodes are called *anchor data nodes*. This means that the output of the change plan must be a valid input for the inserted stream. In the example, it is the case since the only condition for processing the workpiece during human drill is that the workpiece must be a sheet metal. Please note that for the validity check, only the workpiece attributes and not the *position* that represents the current physical location of the workpiece are considered. This is because otherwise it would prevent using suitable streams achieving the workflow goal. For example, if the learned stream from the repository (see Fig. 4b) uses the milling machine in the first shop floor for drilling holes, it could never be added to the change plan since the positions did not match. However, it would also achieve the goal data node that is required in the problem workflow.

To ensure that only streams are replaced or added in which no defective machines are used and by considering the goal data node of the problem workflow, a semantic similarity measure based on the measure presented in [3] is used. This similarity measure assesses the similarity between the goal data node of the problem workflow and the partial workflow output of the stream. In addition, the similarity between the current machine states and the machines needed to execute the stream is determined. If the similarity increases after a replacement or an addition of a stream, the modification is suitable for solving the current problem. At the end, we check if unnecessary streams can be deleted from the change plan while still achieving the workflow goal.

3.2 Integrating Automated Planning for Resolving Inconsistencies

Cyber-physical workflows are executed in physical IoT environments by actuators (see Sect. 2.1). For this reason, adapted workflows must be valid for execution since misconfigured workflows could lead to damage to machines or products or could lead to dangerous situations for humans. However, the compositional adaptation method cannot guarantee that adapted workflows are finally semantically correct and, thus, executable [21,29]. In addition, appropriate adaptation knowledge for solving the problem situation may not be available. One possibility to overcome these issues is to present the adapted workflows before execution to users. In this process, users can fix inconsistent parts (similar to [28]). Although this is a viable possibility, it requires a domain expert to perform these adjustments manually (cf. [6]), which can be complex and time-consuming. Another possibility is to use a further adaptation technique to fix inconsistencies in the adapted workflow that cannot be solved by compositional adaptation. To overcome these problems, we propose to combine POCBR as a knowledge-intensive technique and automated planning as a search-intensive method. The main advantage of this combined approach is that the complete adaptation problem is divided into smaller and, thus, easier to solve sub-problems (*Divide and Conquer*), some of which are solved by the POCBR system and the compositional adaptation and some by AI planning. This combined approach also compensates incomplete planning domain models that are often common in real-world applications [22, 30]. In the following, we present how this combination can be used for automatic workflow adaptations.

In the example in Sect. 3.1, a learned stream from the stream repository (see Fig. 4b) is added to the retrieved change plan (see Fig. 4a) to achieve the adaptation goal. However, the modified change plan requires that the workpiece is located at the oven in the first shop floor and not at the oven in the second shop floor. Thus, the adapted workflow is syntactically correct but not semantically and, therefore, not executable in the smart factory. Since the change plan does not include a transport, it must be added to the adapted workflow to ensure executability. In this context, we check after replacing the failed part of the problem workflow with the change plan whether the semantic correctness is given or whether inconsistencies exist. An inconsistency exists, for example, if the workpiece is not located at the correct position or if the workflow goal is not achieved with the change plan used. In these cases, we automatically determine the inconsistencies and generate a corresponding planning problem that consists of the current environmental conditions, i. e., the machine states, the initial state, and the goal state that should be reached by planning. Figure 5 depicts the adapted workflow with marked inconsistencies. To generate the planning problem, we use the output anchor data node of the first stream as the initial state and the input anchor data node of the second stream as the desired goal state of the planning problem. To maintain executability, the current state of the IoT environment, i. e., the machine states, are also defined in the planning problem so that only executable tasks can be used by the planner. Giving the generated planning problem to a state-of-the-art planner, it can easily solve the problem by adding actions/tasks that transport the workpiece from the oven in the first shop floor to the oven in the second shop floor. By adding these tasks, the adapted workflow is finally semantically correct and, thus, executable. The generative adaptation in our approach mainly needs knowledge about possible transportation routes since the knowledge about production steps such as *Burn*, *Drill*, or *Deburr* is already considered by the creator tasks in the workflow streams. In this way, the laborious and error-prone task of creating a complete planning domain as needed to generate appropriate solutions from scratch can be limited [4,17,22,24,27,30]. However, it could be possible that some situations cannot be solved since the required knowledge is not available in the POCBR system and in the (incomplete) planning domain. In these cases, the approach can support users with a pre-adapted workflow that builds the basis for performing final modifications to ensure executability manually (cf. [6, 28]).

4 Experimental Evaluation

In this section, we present the experimental evaluation of the proposed approach conducted in our physical smart factory (see Sect. 2.1). For this purpose, we implemented the approach in the open-source POCBR framework ProCAKE[3] [2]. We use Fast Downward [8] with an A* search using the landmark-cut heuristic (*lmcut*) as a planner. Moreover, we create a full planning domain description since it is manageable for our rather small smart factory use case, and it allows us to use AI planning as gold standard in the experiment. The domain is written in PDDL 2.1 by using non-durative actions with action costs and negative preconditions. In total, 256 planning actions with several parameters, 27 relational predicates, and one functional numeric fluent, i. e., the total-cost function for the action costs, are contained in the domain[4]. To evaluate the approach, we measure the fulfillment of the performed adaptations by measuring the semantic similarity and the costs for executing the change plan in the factory. In addition, we check whether the adapted workflows are semantically correct and, thus, executable or not. In the experiment, we investigate the following hypotheses:

H1 The compositional adaptation (CA) results in equal or better adapted workflows w. r. t. the described criteria than only using the retrieved case without adaptation (w/o).

H2 Using the combined approach consisting of compositional and generative adaptation (CGA) leads to better results in terms of executability and semantic correctness than the pure compositional adaptation (CA).

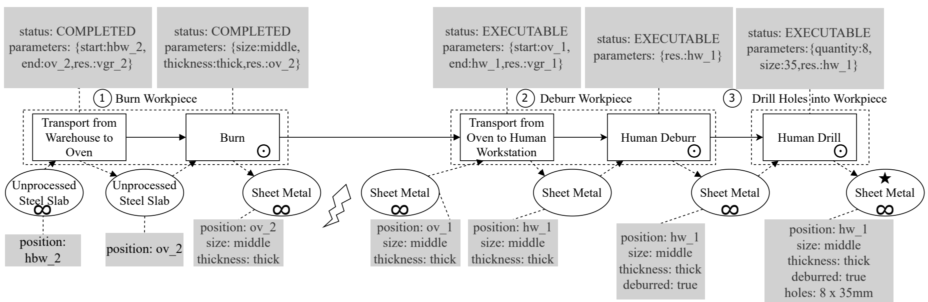


Fig. 5. Manufacturing Workflow As Result of Compositional Adaptation.

---

[3] http://procake.uni-trier.de

[4] The PDDL 2.1 domain and all planning problems are available at https://gitlab.rlp.net/iot-lab-uni-trier/edoc-2022-idams-workshop.

H3 The results generated by the combined approach (CGA) have similar total costs than solving the adaptation problem from scratch by using AI planning (GA).

4.1 Experimental Setup

The Fischertechnik smart factory (see Sect. 2.1) represents two independent production lines and is used for the experiment. The goal of the experiment is to check whether it is possible to modify currently executed workflows after a failure so that they can still be executed, e. g., by using machines from the other production line that can perform the required activities (see application scenario in Sect. 2.3). To obtain real-world problems, a failure during the production is injected in a manufacturing resource of the smart factory. For this purpose, we use a *failure generation engine* that randomly selects machines and switches them to defective. We parameterized the engine to have at most two failures simultaneously, where each individual failure lasts at least 25 and at most 45 seconds. During the entire runtime of the workflows (approx. 6 to 9 minutes for each run), several machine resource failures are generated. If a failure occurs in the smart factory, the affected workflow is stopped and its current state is captured to use it for AI planning (GA) and as a query for the POCBR system (w/o, CA, and CGA) [15]. We utilize *four different production workflows* throughout the experiment that are executed in pairs of two. The used workflows deal with sheet metal production, as already introduced before (see Fig. 2 for an example). Thereby, each shop floor executes a single production workflow at a time: W 1.1 and W 1.2 are executed on the first shop floor and use 6 out of 7 different machines; both containing 12 tasks. W 2.1 and W 2.2 are executed on the second shop floor, use all 7 different machines, and contain 16 and 19 tasks. We apply a *Train and Test* scenario in which we first generate 20 random problems (10 for each pair of workflows) that are solved by using AI planning (GA) if failures occur during execution. Four generated problem situations could not be solved, since no other machines are available as alternative. However, these four problems and the 16 adapted workflows are stored in the case base as best-practice solutions. Based on these 16 correctly adapted workflows, we partitioned each workflow into its workflow streams and store them as adaptation knowledge (*Train* phase) in a stream repository. Finally, we generate 10 (5 for each pair of workflows) further problems that are used for evaluation (*Test* phase).

4.2 Experimental Results

The experiment focuses on the executability of adapted cyber-physical workflows for resolving failure situations during runtime. Table 1 depicts the experimental results of 10 random-generated failure situations conducted in the smart factory. We measure the semantic similarity between the solution and the goal to achieve (first row for each method) and the total-costs for executing the change plan (second row for each method). The total-costs reflect the execution time of the adapted workflow in seconds, i. e., the time required in the smart factory to

Tab. 1. Results of Adapting Cyber-Physical Workflows

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Avg.
w/o	1.0 ✓	0.95 ✗	1.0 ✓	0.98 ✗	1.0 ✗	0.86 ✗	0.97 ✗	1.0 ✓	1.0 ✗	0.96 ✗	0.972
	456	./.	456	342	451	416	./.	451	416	./.	
CA				1.0 ✗		0.89 ✗					0.977
				530		419					
CGA					1.0 ✓	1.0 ✓			1.0 ✓		0.988
					484	817			493		
GA	456	223	456	./.	416	382	376	451	451	376	

execute the adapted part of the production process. A space in the table indicates that the measured values are equal to the method above it. In addition, we indicate whether the adapted workflows are semantically correct (✓) and, thus, executable or not (✗). As a gold standard, we use generative adaptation (*GA*) to compare the POCBR approach against it. For queries *Q1*, *Q3*, and *Q8*, the failure situation can be solved immediately, since similar problems already occurred during training. In this context, complete solutions are contained in the case base and, thus, no further adaptations are required. For the generated failure situations *Q2*, *Q7*, and *Q10*, the retrieved cases are part of the 4 problems that could not be solved by AI planning in the *Train* phase before. For this reason, the POCBR system can only provide limited support to the user, since no more similar cases exist that can be used to solve the problem. However, the system could provide a complete solution for *Q10* if it had previously included *Q7* as a new experienced case, which shows the potential of the proposed self-learning approach. The workflows adapted in *Q5*, *Q6*, and *Q9* are executable but only after they were modified by the combined approach. The sole use of *CA* only leads to an increase in similarity for *Q4* and *Q6*. *Q4* is a special case since a rather similar case is retrieved and adapted, but no solution could be generated as all required machines for recovery are not functional, i. e., workflow executability cannot be achieved. All in all, the experiment shows high potential for using POCBR for adaptive cyber-physical workflows. Some problems can be solved even without adaptation since the solution has already been experienced during training. Whenever an adaptation has been performed, it always results in the same or a slightly higher similarity in contrast to pure retrieval (see **Avg.** column). Thus, we accept H1. In addition, the combined approach *CGA* leads to a higher number of executable workflows compared to *CA* (see *Q5*, *Q6*, and *Q9*). However, during *CGA* adaptation, the total costs sometimes increase strongly and are significantly higher than the costs for generating the solution from scratch. For this reason, we accept H2 but reject H3. To conclude, the use of the combined adaptation approach leads to adapted workflows that can be further executed by achieving their goals in 6 out of 10 cases in the experiment.

5 Conclusion and Future Work

We present an approach for using *Case-Based Reasoning (CBR)* to enhance the flexibility of cyber-physical workflows. In this context, we focus on combining compositional adaptation with generative AI planning for resolving failures during manufacturing. The proposed approach utilizes procedural experiential knowledge and, thus, limits the typically high knowledge acquisition and modeling effort to create comprehensive planning domains. In this context, complete planning domains are required for planning from scratch but often only incomplete domain models are available for planning in real application domains [22, 30]. In contrast, planning in the proposed approach is used to solve smaller sub-problems, requiring general knowledge about transportation routes rather than specific knowledge as stored in cases. However, AI planning is needed in the approach to satisfy the challenging requirement of executability of automatic adapted cyber-physical workflows (see Sect. 2.1 and 2.2). Otherwise, improperly configured and adapted workflows can lead to considerable damage. In an experimental evaluation conducted in a physical smart factory, we showed that the approach can solve most problem situations and can adapt workflows suitably. This is performed either by reusing the case without modifications or by subsequent adaptation with the combined approach. Although the proposed approach has been implemented and validated in the domain of cyber-physical workflows, it can also be applied to other workflow domains (e. g., [21, 29]). This is because compared to other domains, cyber-physical workflows have more specific requirements w. r. t. executability and correctness of the adapted workflows.

In the future, more than one case should be used for adaptation and the best adaptation result should be returned. Moreover, using conversational techniques (e. g., [28, 29]) may promise better adaptation results while integrating domain experts. Transferring the proposed approach to real production lines with larger production workflows and more possible actions is also an interesting aspect for future work but also requires faster workflow retrieval methods (cf. [9]). In this context, it should be investigated how scalable the approaches are w. r. t. the workflow size and the domain complexity. We expect an improved solution quality since the case base contains more and larger workflows of already solved problems and a significantly better computation time than applying a solely generative approach (cf. [4, 5, 27]). Finally, it should be examined how much formalized knowledge is inevitably required in the planning domain, i. e., how incomplete the planning domain can be, for the approaches.

References

1. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. AI Commun. **7**(1), 39–59 (1994)
2. Bergmann, R., et al.: ProCAKE: A Process-Oriented Case-Based Reasoning Framework. In: 27th ICCBR Workshops. vol. 2567, pp. 156–161. CEUR-WS.org (2019)
3. Bergmann, R., Gil, Y.: Similarity assessment and efficient retrieval of semantic workflows. Inf. Syst. **40**, 115–127 (2014)

4. Bergmann, R., Muñoz-Avila, H., Veloso, M.M., Melis, E.: CBR Applied to Planning. In: CBR Technology, LNCS, vol. 1400, pp. 169–199. Springer (1998)
5. Borrajo, D., Roubícková, A., Serina, I.: Progress in Case-Based Planning. ACM Comput. Surv. **47**(2), 35:1–35:39 (2014)
6. Dadam, P., Reichert, M.: The ADEPT project: a decade of research and development for robust and flexible process support. Comp. Sci. Res. Dev. **23**(2), 81–97 (2009)
7. Gil, Y., et al.: Wings: Intelligent Workflow-Based Design of Computational Experiments. IEEE Intell. Syst. **26**(1), 62–72 (2011)
8. Helmert, M.: The Fast Downward Planning System. J. Artif. Intell. Res. **26**, 191–246 (2006)
9. Hoffmann, M., Malburg, L., Bach, N., Bergmann, R.: GPU-Based Graph Matching for Accelerating Similarity Assessment in Process-Oriented Case-Based Reasoning. In: 30th ICCBR. LNCS, vol. 13405, pp. 240–255. Springer (2022)
10. Janiesch, C., et al.: The Internet-of-Things Meets Business Process Management. A Manifesto. IEEE Syst. Man Cybern. Mag. **6**(4), 34–44 (2020)
11. Kendall-Morwick, J., Leake, D.: A Study of Two-Phase Retrieval for Process-Oriented Case-Based Reasoning. In: Successful CBR Applications-2, vol. 494, pp. 7–27. Springer (2014)
12. Lasi, H., et al.: Industry 4.0. BISE **6**(4), 239–242 (2014)
13. Lee, J., Kao, H.A., Yang, S.: Service Innovation and Smart Analytics for Industry 4.0 and Big Data Environment. Procedia CIRP **16**, 3–8 (2014)
14. Malburg, L., et al.: Semantic Web Services for AI-Research with Physical Factory Simulation Models in Industry 4.0. In: 1st IN4PL. pp. 32–43. ScitePress (2020)
15. Malburg, L., Bergmann, R.: Towards Adaptive Workflow Management by Case-Based Reasoning and Automated Planning. In: 30th ICCBR Workshops. CEUR-WS.org (2022), Accepted for Publication.
16. Malburg, L., Seiger, R., Bergmann, R., Weber, B.: Using Physical Factory Simulation Models for Business Process Management Research. In: BPM Workshops, LNBIP, vol. 397, pp. 95–107. Springer (2020)
17. Marrella, A., Mecella, M., Sardiña, S.: Intelligent Process Adaptation in the SmartPM System. ACM Trans. Intell. Syst. Technol. **8**(2), 25:1–25:43 (2017)
18. Marrella, A., Mecella, M., Sardiña, S.: Supporting adaptiveness of cyber-physical processes through action-based formalisms. AI Commun. **31**(1), 47–74 (2018)
19. Minor, M., et al.: Case-based adaptation of workflows. Inf. Syst. **40**, 142–152 (2014)
20. Monostori, L.: Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. Procedia CIRP **17**, 9–13 (2014)
21. Müller, G., Bergmann, R.: Workflow Streams: A Means for Compositional Adaptation in Process-Oriented CBR. In: 22nd ICCBR. LNCS, vol. 8765, pp. 315–329. Springer (2014)
22. Nguyen, T.A., Sreedharan, S., Kambhampati, S.: Robust planning with incomplete domain models. Artif. Intell. **245**, 134–161 (2017)
23. Reichert, M., Weber, B.: Enabling Flexibility in Process-Aware Information Systems - Challenges, Methods, Technologies. Springer (2012)
24. Rodríguez-Moreno, M.D., Borrajo, D., Cesta, A., Oddi, A.: Integrating planning and scheduling in workflow domains. Expert Syst. Appl. **33**(2), 389–406 (2007)
25. Seiger, R., et al.: Toward a framework for self-adaptive workflows in cyber-physical systems. Softw. Syst. Model. **18**(2), 1117–1134 (2019)
26. Seiger, R., et al.: Integrating process management and event processing in smart factories: A systems architecture and use cases. J. Manuf. Syst. **63**, 575–592 (2022)

27. Veloso, M.M.: Planning and Learning by Analogical Reasoning, LNCS, vol. 886. Springer (1994)
28. Weber, B., et al.: CBRFlow: Enabling Adaptive Workflow Management Through Conversational Case-Based Reasoning. In: Adv. in CBR, LNCS, vol. 3155, pp. 434–448. Springer (2004)
29. Zeyen, C., Malburg, L., Bergmann, R.: Adaptation of Scientific Workflows by Means of Process-Oriented Case-Based Reasoning. In: 27th ICCBR. LNCS, vol. 11680, pp. 388–403. Springer (2019)
30. Zhuo, H.H., Nguyen, T.A., Kambhampati, S.: Model-Lite Case-Based Planning. In: 27th AAAI. AAAI Press (2013)