IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 September 2023, accepted 30 October 2023, date of publication 8 November 2023,
date of current version 16 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3331221

RESEARCH ARTICLE

Digital Twin Framework for Reconfiguration Management: Concept & Evaluation

BIRTE CAESAR[1], **KIRA BARTON**[2,3], **(Senior Member, IEEE), DAWN M. TILBURY**[3], **(Fellow, IEEE),**
AND ALEXANDER FAY[1], **(Senior Member, IEEE)**
[1]Institute of Automation Technology, Helmut Schmidt University, 22043 Hamburg, Germany
[2]Mechanical Engineering Department, University of Michigan, Ann Arbor, MI 48109, USA
[3]Robotics Department, University of Michigan, Ann Arbor, MI 48109, USA

Corresponding author: Birte Caesar (birte.caesar@hsu-hh.de)

This work was supported by the Fulbright Visiting Student Researcher Program (Doktorand:innenprogramm), which is sponsored by the U.S. Department of State and the German-American Fulbright Commission.

ABSTRACT To remain competitive in a highly dynamic environment, manufacturing companies have to quickly react to disturbances or changing customer requirements. To enable manufacturing systems to cover these dynamics, the concept of reconfigurable manufacturing systems was introduced. From a technical point of view, this concept has been exploited for the past 20 years, revealing several different design solutions. However, industrial application is still an exception. Our analysis led to the assumption that this is due to a lack of operator support for reconfiguration management. In addition, mostly individual aspects of reconfiguration are considered instead of exploiting the entire reconfiguration space at the system and machine level. Therefore, in this paper, we present a digital twin framework for reconfiguration management considering reconfiguration as a holistic problem. We evaluate the framework by conducting a case study and challenging it by evaluating the completeness based on a systematic literature review, and analyze if it follows good practice based on 32 requirements for digital twin frameworks.

INDEX TERMS Reconfigurable manufacturing systems, reconfiguration planning, cyber-physical production system, configuration selection, adaptation.

I. INTRODUCTION

Industrial production is in a state of change. This includes, on the one hand, the individual desires every one of us, which increases the importance of and demand for individualized and customized products. On the other hand, environmental awareness and the demand for resource-friendly production are increasing, as about a third of the energy consumption in the world is emitted by the manufacturing domain [1]. These social changes and the scarcity of resources pose new challenges to the manufacturing domain. In particular static production lines are not prepared for these challenges, as varying product requests require flexible and reconfigurable manufacturing systems (RMSs), which provide the required manufacturing capabilities and can prevent energetic and resource inefficient configurations of the manufacturing systems. Accordingly, dedicated production lines are not suitable for reconfiguration and reuse in the production of individualized products. Non-optimal operating production lines increase environmental impact and lead to a waste of resources. Currently, most existing production systems are designed for single-purpose usage with limited or no flexibility [2], [3], [4].

One solution to address this need is RMSs, which can be composed of various machines from different vendors providing varying manufacturing capabilities [5]. During the past 20 years, the concept of RMSs has been exploited from a technical point of view, presenting several design solutions for hardware and software modularity. But for industrial applications there are still several barriers to overcome [4], [5], [6]. In particular, reconfiguration management during operation remains mainly a manual task and is individually triggered [5], [7]. Reconfiguration management includes the identification of reconfiguration needs, reconfiguration

The associate editor coordinating the review of this manuscript and approving it for publication was Binit Lukose.

planning, and finally reconfiguration execution [8]. Without systematic and methodical support, reconfiguration management is error prone and the configuration selection process during reconfiguration planning remains subjective based on incomplete knowledge of individual workers [7], [9].

RMSs are reconfigurable at both the system and machine level. At the system level, machines with required capabilities are selected and arranged, as mentioned by Nakano et al. [10]. At the machine level, a machine configuration must be selected that provides the capability with the right manufacturing parameters [5]. To underline the need for reconfiguration management, Haddou Benderbal et al. [9] states that configuration selection is a complex task, with complexity increasing if both system and machine reconfigurations are considered, as the number of configurations increases exponentially with the number of machines in a RMS [9]. The effect is magnified when considering that the RMS consists of machines that are themselves reconfigurable, such as reconfigurable machine tools (RMTs). Furthermore, to support reconfiguration management, economic criteria are important as reconfiguration decision support can also be used as economical reconfiguration triggers. It should be noted, however, that the relevant economic criteria can differ greatly depending on the organizational level and must be taken into account [3], [11], [12].

This leads to the research question of *how RMS operators can be supported in reconfiguration management such that the solution space is fully exploited and various economic factors, as well as reconfiguration triggers, can be considered appropriately*. Specifically this means, given an existing system and its set of possible configurations, a desirable, optimal, or most cost-effective configuration needs to be selected depending on a strategic reconfiguration trigger, i.e. new product request or an operational reconfiguration trigger, such as a machine breakdown. Further, the selection of a configuration has to take into account operator defined optimization criteria. To deal with the different time requirements resulting from different reconfiguration triggers and the heterogeneous information that needs to be combined for reconfiguration management at the system and machine level, the concept of digital twins (DTs) comes into play. A DT should help to understand and optimize the performance of the real world entity and combine data from different data sources, like planning data with data from the physical entity, e.g., measurement data [13], [14]. Therefore, to improve reconfiguration management for RMS operators, we propose a DT framework for reconfiguration that enables the consideration of operational and strategic reconfiguration triggers, while taking into account the reconfiguration space on both the system and machine levels. Further, the reconfiguration space is dynamically updated and adapted based on the current state of the physical system.

This article is an extended version of previous conference papers [15], [16]. In [15] we present a detailed analysis of reconfiguration management, highlighting the diversity of reconfiguration objectives and triggers, thus making reconfiguration management a complex task. In [16] we introduced our solution of a digital twin framework for reconfiguration management. In this paper, we present a detailed description of the updated framework and its most important components, including behavior flow charts. Additionally, we present a detailed evaluation of the framework, which was missing in the previous papers.

The remainder of the article is structured as follows. Section II presents a detailed analysis of related work highlighting the research gap. Section III presents the developed digital twin framework for reconfiguration, including a detailed explanation of the important components of the framework, such as the reconfiguration trigger analyzer (Section III-A), reconfiguration planning (Section III-B), and the decision maker (Section III-C). Section IV presents a detailed evaluation of the DT framework for reconfiguration consisting of a conducted case study for functional validation (Section IV-A), a redundancy analysis of the framework components based on a systematic literature review (Section IV-B), and a good practice analysis based on 32 requirements for digital twin frameworks (Section IV-D). The article concludes with a summary in Section V.

II. RELATED WORK

This section presents a brief introduction to reconfiguration management (Section II-A), followed by a review of the state of the art regarding digital twin frameworks for reconfiguration management (Section II-C). In Section II-B the requirements derived from the research question, clustered into functional and non-functional requirements, are presented. The section is followed by a presentation of the state of the art approaches, which are assessed by the requirements in Section II-D concluding with a research gap.

A. BACKGROUND

Reconfiguration management includes the identification of reconfiguration needs Ⓐ, reconfiguration planning Ⓑ, configuration selection Ⓒ, and finally reconfiguration execution Ⓓ [8], [17]. Reconfiguration triggers are used to identify reconfiguration needs and are monitored during operation. Additionally, reconfiguration triggers play an important role in determining the required reconfiguration tasks of a reconfigurable manufacturing system, as different disturbances require different solutions.

Therefore, reconfiguration planning Ⓑ includes a total of six tasks (*i*-*vi*) that are executed depending on the reconfiguration trigger. For reconfiguration of a manufacturing system, the product plays a significant role as it specifies the required manufacturing processes. Therefore, the first required task of reconfiguration planning is *(i) process planning*, which is the creation of a bill of processes (BOP) based on the product features to be manufactured and is sometimes part of production planning [18]. Many features of a product can be created by different manufacturing processes. This consideration results in several BOPs, which enable full exploitation of the manufacturing capabilities of the RMS and offer more flexibility for reconfiguration planning.

At the system level, reconfiguration can take different forms and require different tasks. The task *(ii) layout design* has the goal of maximizing the efficiency of the workflow through the specific arrangement of machines and equipment in a manufacturing system [19]. Layout design is the assignment of a machine to a dedicated spot on the shop floor taking into account several constraints and optimization of the layout considering transportation time, material handling cost, etc. [20]. Layout design includes hardware and software reconfigurations. The optimal layout is strongly dependent on the planned paths a product can take through the manufacturing system.

This leads to the task *(iii) path planning*, which refers to the selection of machines to manufacture a certain product or product feature. Path planning is also known as dispatching or machine job allocation [21]. The path can differ if several machines can provide the same manufacturing capability, or if a product feature can be created by different manufacturing capabilities.

With several path plans for different products, the task *(iv) scheduling* comes into play. Scheduling refers to a time-wise decision about which product feature is created on a certain machine [18]. Path planning and scheduling include only software reconfigurations. At the machine level, reconfiguration planning includes the *(v) capability management* and the *(vi) interoperability management*. Capability management includes the selection of configurations for a reconfigurable machine tool (RMT). Selection includes a check between the required capability and the available capability considering the tools and modules that can be selected. An RMT can have several configurations that provide the required capabilities. The required capabilities are specified during process planning and include the manufacturing processes and additional attributes, such as required quality. The interoperability management considers the constraints an RMT has regarding the fulfillment of its tasks, i.e. processing a product feature. These constraints could be the requirement of an RMT for a separate loading and unloading system or a connection to a transportation unit.

The configuration selection Ⓒ takes the generated configurations of the reconfiguration planning into account and selects a configuration according to predefined objectives. Finally, reconfiguration execution Ⓓ implements the selected configuration by, for example, adding hardware components and executing planned and scheduled processes.

B. REQUIREMENTS REGARDING DIGITAL TWIN FRAMEWORKS FOR RECONFIGURATION MANAGEMENT

According to Scully [22] there are 252 existing DT categories, resulting from a combination of the hierarchical level considered, e.g., product, process, or system; the life cycle phase, e.g., design, operation, or maintenance; and the defined purpose of the digital twin, e.g., prediction, optimization, or reconfiguration. Different digital twin frameworks, e.g. [14], [23], and [24], have been developed to provide a digital twin development blueprint applicable to these categories. However, the goal to cover this diversity of digital twin applications makes the existing digital twin frameworks rather abstract. This in turn leads to little guidance when it comes to a specific application, such as reconfiguration management. Accordingly, the focus in this section is on existing DT frameworks for reconfiguration management. It is assumed that the specifics of the DT concept are considered by all DT frameworks for reconfiguration management and therefore only the reconfiguration management-specific requirements are examined here. This section presents a detailed description of the requirements for a DT framework for reconfiguration management. These requirements are derived from the research goal and reinforced by literature. The requirements are grouped as functional requirements (*F*) and non-functional requirements (*NF*). The requirements are used to compare and evaluate the existing DT frameworks for reconfiguration management, which are presented in Section II-C.

1) F1 (OPERATOR SUPPORT)

Operator support is defined as any assistance provided to the operator to perform the operator's duties [25]. This becomes more important as the complexity of the tasks entrusted to the operator increases. As stated by Marks et al. [7] and Andersen et al. [5] the lack of operator support is one of the barriers why RMS are still not used very frequently in industry. As described in Section I, reconfiguration management is affected by the different characteristics of the RMS, the operational objective, and the triggers of the reconfiguration. This means that the operator support is characterized differently depending on the application. Therefore, the first functional requirement is that a DT framework for reconfiguration management incorporates any means to support the operator.

2) F2 (CUSTOMIZED OPTIMIZATION)

The reasons and goals for reconfiguration are manifold and so are optimization criteria helping to select a suitable configuration. Brahimi et al. [19] and Yelles-Chaouche et al. [20] both present detailed literature reviews to identify the most common optimization objectives for RMS. In a previous publication, we have assigned the optimization objectives to the individual reconfiguration tasks, which revealed a non exclusive assignment [15]. This finding results in the requirement for customized optimization so that the operator can specify the optimization criteria as needed.

3) F3 (OPERATIONAL & STRATEGIC RECONFIGURATION TRIGGERING)

Wiendahl et al. [26] introduced an overview of manufacturing change drivers clustered into internal and external triggers. Erinakis et. al categorize the triggers as well based on the source, but choose as categories: production, demand, and supply [27]. Both underline the relevance of the different triggers to be able to initiate appropriate reconfiguration measures. However, this classification does not take into account the temporal aspect of the triggers. Therefore, in the following, the categories of operational and strategic triggers are used. Operational triggers can be summarized as triggers that require short-term adjustments to production or lead to disruptions in production. Strategic triggers, in contrast, can be long-term corporate decisions or new objectives that the production has to achieve. Thus, to fully exploit the potential of RMS, it should be possible to consider different types of reconfiguration triggers [27].

4) F4 (DYNAMIC RECONFIGURATION SPACE)

For each reconfiguration the reconfiguration space needs to be updated. This includes the consideration of the current configuration, as well as the current system state. The set of feasible configurations that can be applied at any given point in time is influenced by the current state of the system [28]. Further, this enables the consideration of machine replacement or additional machines. Consequently, the consideration of a dynamic reconfiguration space increases the opportunity to find a feasible solution and also increase the responsiveness.

5) NF1 (SOLUTION NEUTRAL)

A framework is a conceptual presentation that serves as a guideline for implementation. Companies often have to follow company guidelines for implementation and software, as well as they have to consider the existing software landscape. As the goal of a framework is to provide insights and explanations to a concept and to enable broad applicability, the framework is required to be solution neutral.

6) NF2 (DETAILED DESCRIPTION)

The description in Section I shows that reconfiguration management can differ greatly. The aim is to give the user an impression of the functionality of the reconfiguration management and how it fits into its environment. Consequently, for each component of the reconfiguration management framework, it is required that there is a description of the functionality, the motivation, and the integration into the overall context. Only then the framework can be used as a blueprint for requirements specification.

7) NF3 (DEFINITION OF RECONFIGURATION TASKS)

To be able to assess the relevance of a framework for a particular use, it is not only important to describe the individual framework components in detail, but also to explicitly specify reconfiguration and what reconfiguration tasks are considered. As stated in Section I, reconfiguration can be conducted on the system or machine level and therefore is used for different reconfiguration requests.

8) NF4 (MODULARITY)

Reconfiguration management has to deal with several aspects of RMS resulting in complex and heterogeneous reconfiguration tasks [29]. The systematic literature review of reconfiguration management revealed that each solution for reconfiguration management is different and tailored for a specific purpose [30]. To meet the individual reconfiguration management requirements of a specific RMS, the framework is required to incorporate modularity aspects [29]. This enables a wide range of application of the framework and a custom tailored solution based on selectable framework components.

C. EXISTING DIGITAL TWIN FRAMEWORKS FOR RECONFIGURATION MANAGEMENT

Qamsane et al. [31] present a unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems to create digital twins that accurately represent the physical system's dynamics so that real-time optimization and the generation of effective production control actions are possible. The authors present in [24] and [32] a requirements driven methodology to develop purpose driven digital twins and provide general guidance for the development of digital twins. The publications serve as a general basis but must be adapted for the explicit application of reconfiguration management.

Poudel et al. [33] extend the work of Qamsane et al. to include reconfiguration aspects for dynamic manufacturing planning. For reconfiguration, layout planning and resource selection are included. Both provide reconfiguration functionality to cover strategic reconfiguration triggers, such as a new manufacturing order. The reconfiguration selection uses ontological descriptions to map machine capabilities to the required manufacturing processes provided by the operator. No machine level reconfiguration is included, but the framework includes means to introduce further DT functionality via a digital twin manager. Layout planning takes into account the specific requirements of each machine when generating shop floor layouts. The machine requirements are contained in the ontological machine description. The framework includes an application layer that provides the ability to use simulations to evaluate the generated configurations or generate KPIs of current system performance.

Haddou Benderbal et al. [9] present a first draft of a digital modular framework for RMS with a specific focus on software reconfiguration. The framework follows the basic definition of a digital twin, distinguishing between a physical and a virtual space including a live data acquisition and a feedback loop to the physical system. The alignment with the digital twin architecture is based on the digital twin advantages of transparency and real-time feedback to explicitly consider the RMS characteristic diagnosability. Further, an optimization based simulation is introduced to explore the configuration space. The optimization based simulation is considered replaceable for different applications so that each of the main characteristics of an RMS can be addressed. This approach shows on an abstract level how the digital twin architecture can be used for an RMS, but is not pointing out if there are specific adaptions for an RMS or how the diagnosis could be performed. However, it is not clear how the individual RMS characteristics can be integrated into the framework and whether multiple characteristics can be addressed simultaneously.

In contrast, Tang et al. [6] present a more detailed RMS-DT framework showing how the core characteristics of an RMS could be consolidated with a DT. The framework is aligned with the ISA95 architecture as an important foundation of operation views in manufacturing and providing an overview of the decision timing requirements. The authors point out that the reconfiguration decisions require different decision time frames and assign the RMS characteristics to ISA95 layers. Both, Haddou Banderbal et al. and Tang et al., showed the theoretical application, but lacked an implementation, which helps to understand which functionalities are needed to understand how the RMS characteristics can be managed.

In turn, Leng et al. [3] present an implementation of a digital-twin-driven platform for rapid reconfiguration for new product orders to reduce the overhead of the reconfiguration process. Therefore an open architecture and REST-based interfaces are developed to enable the reconfiguration of the RMS industrial internet of things-system based on the current integrated RMTs. The approach covers software-based challenges as well as a configuration selection integrating layout design, path planning, and capability selection. The provided platform functionalities are executed in an integrated manner triggered by a new product order. This combination of functionalities requires the system to be in an idle state as hardware reconfiguration is included. The approach lacks the consideration of the different decision time frames resulting from the RMS characteristics and reconfiguration triggers.

Eirinakis et al. [27] present a generic framework for disruption handling including supply chain disruption for Situation-Aware Manufacturing System (SAMS). The authors introduce the term SAMS as a new paradigm, but the definition is comparable with DTs. The framework consists of four main components, monitoring and analysis, data provision and control execution, optimization, and simulation. The monitoring and analysis is the centerpiece of the presented work. Prognosis models are trained using historical data to predict disruptions during operation considering real-time data. The disruptions can be operational or strategic. Data provision and control execution describes the requirement that the system under consideration provides means to enable this. The optimization is comparable to the reconfiguration planning, but unfortunately, no further details are presented. From the considered use cases, it becomes clear that only scheduling is considered as a reconfiguration task. The simulation is used to evaluate the generated schedule based on various KPIs. The framework is used to instantiate different use cases, but lacks the possibility to adapt the reconfiguration planning as it only comprises one reconfiguration task.

Kombaya Touckia et al. [34] present a DT framework for RMS design and simulation. The framework includes three detailed meta-models to describe the structure, the operations, and the configurations of an RMS. The meta-models are modeling guidelines and enable the inclusion of machines, transport systems, and humans as well as logic aspects. Based on the models, the authors developed a Matlab/Simulink simulation that can be executed to evaluate the current configuration or to create a new configuration. The authors mention that the usage of real-time data is possible, but don't describe how these are incorporated in the simulation. The approach focuses on the modeling of RMS rather than on reconfiguration. In addition, the type of reconfiguration that can be achieved based on the simulation is not described in detail.

Müller et al. [35] present an architecture for self-organised reconfiguration management. The architecture is aligned with the RAMI 4.0 framework and is complemented by UML meta-models and a modeling approach for knowledge specification for easy machine integration. The required models are located in the proxy layer and reconfiguration management in the management layer. The reconfiguration management consists of four tasks; the identification of reconfiguration demand, the generation of alternative configurations, the evaluation of configurations, and the selection of a new configuration. The authors have published a detailed description of the first two tasks in [36]. This publication makes it evident that the generation of the alternative configurations is an integrated approach consisting of layout design, path planning and capability management. The manufacturing request of a new product and a machine failure are considered reconfiguration triggers. A selection of evaluation criteria can be made by the operator via a GUI. The approach requires an implementation of all architecture components. However, modularization may be possible in the future because the reconfiguration management layer is implemented as a multi-agent system. First steps towards modularization are shown by the integration of different simulations that can be chosen dynamically.

Mo et al. [37] present a framework for managing manufacturing system reconfiguration optimization. The framework is based on a graph database providing among others information about the current configuration of the system, as well as KPIs to assess the configurations. Altogether the framework incorporates three reconfiguration aspects. First, the matchmaking between requested and provided manufacturing capabilities including the selection of the best fitting machine for each task. Second, optimization of the layout and machine parameters for the selected configuration. The optimization method is selected based on the user defined KPIs. And lastly, the control code update. A reconfiguration is triggered by changed product requests. Even though the authors use data from the physical system to form different DT based simulations, they do not consider the update of the reconfiguration space based on the current system state. The framework cannot be individualized for different applications.

D. RESEARCH GAP

Table 1 shows the summary of the requirements satisfied by the approaches presented in Section II-C. All approaches, except for Tang et al. [6], offer support to the operator even though the support takes different forms, (satisfaction of F1). Conversely, the framework presented by Tang et al. aims to make the data structure and reconfiguration functionalities compatible with the existing ISA-95 architecture, rather than providing a direct benefit to the operator. The approaches mainly present solution neutral frameworks, so that they are generally applicable, (satisfaction of NF1). Seven of the approaches include means for dynamically updating the reconfiguration space (satisfying F4). Furthermore, all approaches do consider optimization goals, but only Haddou Benderbal et al. [9], Erinakis et al. [27], Müller et al. [38], Poudel et al. [33], and Mo et al. [37] consider the possibility for the operator to adapt the optimization goals (satisfaction of F2) by altering the input parameters for the optimization based simulation. Requirement F3 shows that only four of the approaches, Tang et al. [6], Erinakis et al. [27], Müller et al. [38], and Poudel et al. [33], include both operational and strategic reconfiguration triggers. The other approaches focus on either operational or strategic triggers. Based on the operational data available due to the DT concept, Tang et al. and Erinakis et al. explicitly encourage the use of these data to consider, for example, quality deviations or machine failures as triggers for reconfiguration in addition to the product. Even though almost all approaches describe their frameworks in detail (satisfaction of NF2), four of the approaches remain at an abstract level in their description of the framework, leaving the reconfiguration tasks considered by the framework unclear. At best Qamsane et al. [31], Tang et al. [6], Müller et al. [38], Poudel et al. [33], and Mo et al. [37] clearly describe the considered reconfiguration tasks (satisfaction of NF3). For example, Tang et al. describes the specific functionalities for each RMS characteristic. Qamsane et al. [31], Haddou Benderbal et al. [9], Tang et al. [6], and Poudel et al. [33] consider a modularity concept (satisfaction of NF4). Whereas Haddou Benderbal et al. only describe that the framework components can be replaced by another. Qamsane et al. and Tang et al. offer a detailed concept for integrating different components into the framework. Qamsane et al.'s solution is based on a software-defined control concept that is also used by Poudel et al. [33]. Tang et al.'s approach is a modularization concept based on the ISA 95 automation pyramid.

TABLE 1. Analysis of requirements satisfaction in the relevant literature.

Approaches	Requirements							
	F1	F2	F3	F4	NF1	NF2	NF3	NF4
Qamsane et al. 2019 [31]	✓	-	-	✓	✓	✓	✓	✓
Haddou Benderbal et al. 2020 [9]	✓	✓	-	✓	✓	✓	-	✓
Tang et al. 2020 [6]	-	-	✓	✓	✓	✓	✓	✓
Leng et al. 2020 [3]	✓	-	-	✓	-	✓	-	-
Eirinakis et al. 2021 [27]	✓	✓	✓	✓	✓	✓	-	-
Kombaya Touckia et al. 2022 [34]	✓	-	-	-	✓	-	-	-
Müller et al. 2022 [38]	✓	✓	✓	✓	✓	✓	✓	-
Poudel et al. 2022 [33]	✓	✓	✓	✓	✓	✓	✓	✓
Mo et al. 2023 [37]	✓	✓	-	-	✓	✓	✓	-

Table 1 shows that Poudel et al. [33] is the only approach that satisfies all of our stated requirements. The approach was published parallel to our first publication of the framework [16] and is therefore analyzed here in detail. The goal of our framework is to provide comprehensive knowledge for all reconfiguration management tasks, including the broad facets of reconfiguration planning. Table 2 shows an overview of the reconfiguration planning tasks considered by Poudel et al. [33]. Poudel et al. do not consider the process planning, but instead take the process plan as a starting point for a reconfiguration. During layout design, interoperability aspects of single machines are considered. But as the capabilities of a machine are considered fixed, so are the interoperability aspects of a machine. Therefore, interoperability management is partially satisfied because no changes to interoperability aspects are considered here. The same applies for capability management, as only static capabilities are considered and there is no reconfiguration at the machine level. The completeness evaluation in Section IV-C underlines that process planning and capability management are among the most important tasks for reconfiguration management.

TABLE 2. Completeness analysis of Poudel et al. [33].

Reconfiguration Planning Task	Poudel et al. [33]
Process Planning	-
Layout Design	✓
Path Planning	✓
Scheduling	✓
Capability Management	(✓)
Interoperability Management	(✓)

In summary, no framework takes reconfiguration management into account in all facets, but Poudel et al. present an approach considering the relevant aspects of a good digital twin framework for reconfiguration. Therefore, the aim of this paper is to present more information our developed DT Framework for reconfiguration management that includes all reconfiguration planning tasks and highlights how the operator will be supported depending on the components included in a solution. Additionally, we present the relationship between reconfiguration planning tasks and the information required for the execution of the tasks.

III. DIGITAL TWIN FRAMEWORK FOR RECONFIGURATION MANAGEMENT

System reconfiguration is a complex and multidimensional task that includes different decision making processes. The design and implementation of reconfigurable manufacturing systems can take many forms and is highly context dependent [5]. However, reconfiguration planning can be accomplished by different tasks described in Section III-B. Literature reviews, such as Müller et al. [30] and Yelles-Chaouche et al. [20], show that a lot of approaches exist that present solutions for specific tasks and task combinations considering different optimization goals. In addition, tasks such as scheduling or path planning are often considered as flexibility separate from reconfiguration. What is missing, however, is assistance in deciding when which tasks are required and a precise description of how these are related and influence each other [5]. Furthermore, as shown in [15], there are many different triggers for reconfiguration that have a strong impact on the tasks, but have not been sufficiently considered [11]. Reconfiguration needs based on different triggers may have different time horizons. At the same time, the solutions to these reconfiguration needs may also have different time horizons. For example, increasing capacity by purchasing new machines is a strategic decision, but optimizing the schedule in response to a new order is an operational decision. The same applies to the optimization objectives considered for the configuration selection of the RMS. Both in some cases affect multiple system levels. To meet the different time horizons and system level requirements, an additional integration of live data from the shop floor is required [11]. The concept of the digital twin is therefore a suitable foundation [39]. Our aim is to present a holistic framework to release the multilevel potential by presenting a guideline for capability selection based on desired optimization objective, reconfiguration triggers, and the current state of the manufacturing system. Therefore, the digital twin of reconfiguration is a holistic approach that includes flexibility decisions and provides the necessary capabilities based on the current situation and system state.

The integration of reconfiguration and DTs is still little considered in research [40]. Therefore, a general analysis of DT architectures was conducted to create a foundation and align the framework with it. Accordingly, the general structure of the framework is based on the classic architecture of a digital twin, which distinguishes between the physical space and the virtual space [41] (Fig. 1). The physical space represents the RMS, which consists of several reconfigurable machine tools (RMTs), non reconfigurable machines and reconfigurable transport systems (RTSs). An RMS is described as $M_i = \{M_{ia}, \ldots, M_{in}\}$, whereas M_{ia} represents a specific RMT, RTS, or a non reconfigurable machine a, which is available and belongs to the RMS i. Each element within an RMS provides an interface for sensor and system data acquisition, as well as a control interface to enable the reconfiguration decision.

The collected data is stored and processed in the *data layer*, also named digital shadow, of the virtual space. The data collection receives periodic or event-based data in the form of measured variables from the physical system and is responsible for storing them. Likewise, information about the current system configuration is collected and stored in the data layer, represented in Fig. 1 as *configuration knowledge*. The real-time process data can be used to define reconfiguration triggers to be monitored by the reconfiguration trigger analyzer, such as energy consumption or processing time. In addition, the system state can be monitored in case of system failure. By system state we mean the system states defined in the ISA 88 state machine, such as Idle, Execute, Aborting, and so on. These system states can also be used to define when a reconfiguration can be performed. Section III-B will present more insights, into what information is considered as configuration knowledge to describe the current configuration of the RMS. The *data processing* prepares the data so that it is available for the reconfiguration trigger analyzer, as well as for the reconfiguration planning. Machine data is included and can be used by different DTs, e.g., to monitor the behavior of a certain machine or process.

The *reconfiguration trigger analyzer* takes on an important role, as it triggers the reconfiguration planning by requesting a reconfiguration and at the same time integrating information from different sources and putting it into context for reconfiguration. Strategic trigger data, such as production plans or data from the enterprise resource planning (ERP) system, must be combined with operational trigger data, such as current system state or utilization, to generate a reconfiguration request that specifies the possible reconfiguration planning tasks to be executed. This allows a narrowing down of the reconfiguration space. Section III-A presents a detailed description of the reconfiguration trigger analyzer.

The *reconfiguration planning* provides the tasks *(i) process planning*, *(ii) layout design*, *(iii) path planning*, *(iv) scheduling*, *(v) capability management*, and *(vi) interoperability management*, which can be executed separately or in combination to generate a new configuration. Current system information, as well as identified requirements and constraints provided by the reconfiguration request of the reconfiguration trigger analyzer, determine the valid reconfiguration space and which reconfiguration tasks to consider. In addition, the optimization goals of the operator are included for configuration generation provided by the decision maker. In Fig. 1 the reconfiguration planning is provided with a system state by the data layer. Under the term system state, various information is summarized, e.g., machine failure, utilization, etc. The interaction of the individual tasks and their required input information and their provided output is described in detail in Section III-B.

Overall, the *decision maker* is provided with a selection of configurations created by the tasks of reconfiguration planning. The decision to execute a reconfiguration can either be with the user or with the decision maker. For critical decisions, decision options or explicit advice are presented to the user, who selects and starts the reconfiguration. For non critical and recurring decisions the decision maker selects the configuration and triggers the execution. At the same time, the selected configuration is stored as the current

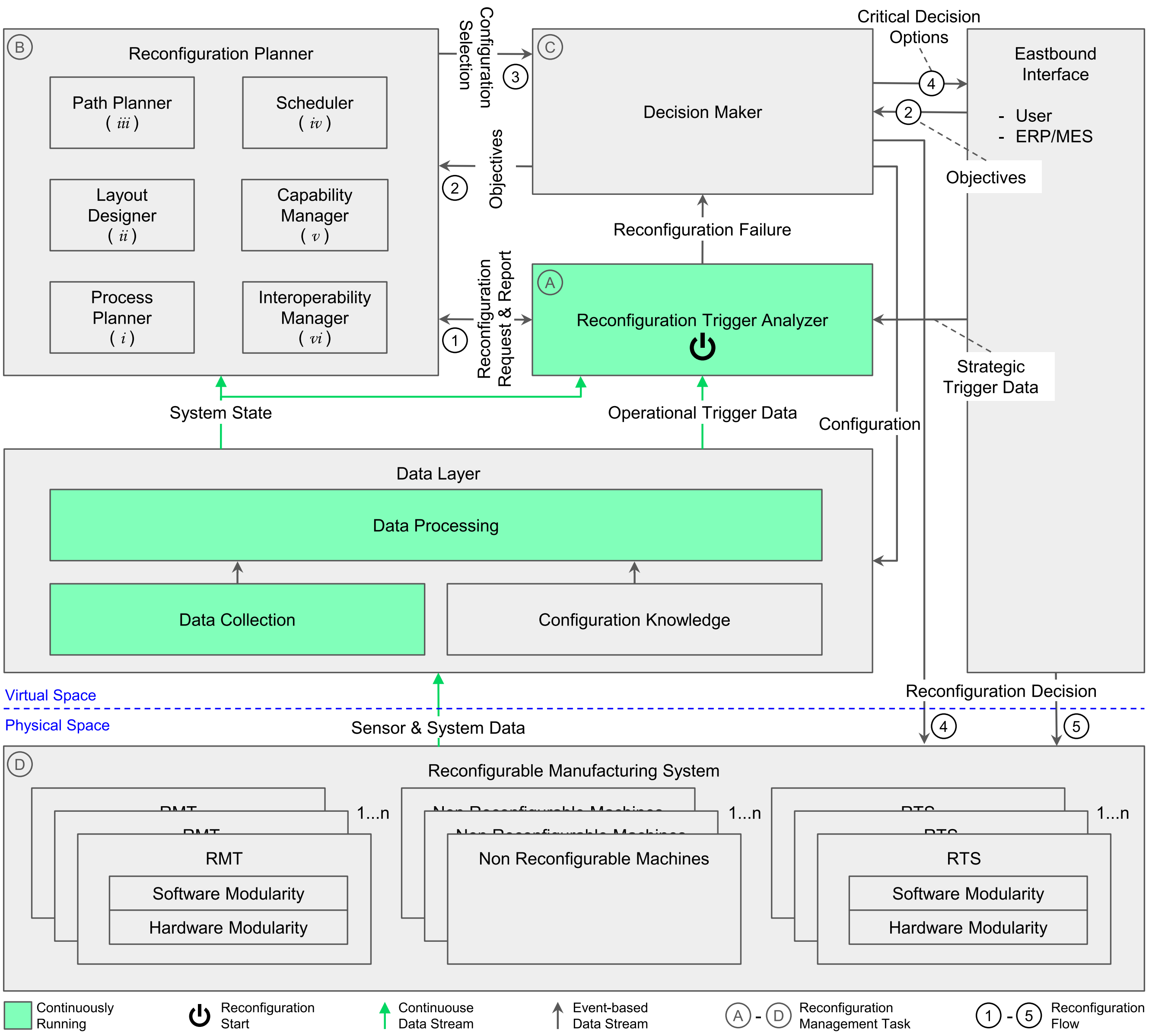

FIGURE 1. **DT framework for reconfiguration.** ***Note: The framework can be instantiated as one DT or several DTs providing different functions of the framework, each provided with data by a shared data layer.***

configuration in the configuration knowledge base of the data layer. The operator must define which decisions are considered critical and which are not. This cannot be defined in general, as it depends strongly on the physical system.

The *eastbound interface* enables the interaction of users with the DT and the connectivity to higher-level systems. The eastbound interface is the access point for users to configure and manage the decision maker, get recommendations, as well as connection to MES or ERP systems. Further, the decision maker is capable of requesting a certain reconfiguration task to meet the required objectives, even though the reconfiguration trigger analyzer may not request that specific task.

The framework design considers the management of the RMS characteristics described in [42]. The characteristics modularity, integrability, scalability, convertability and customization are managed by the reconfiguration planning tasks. Diagnosability is considered through the reconfiguration trigger analyzer and the decision maker. There is no claim of completeness of the framework so that it can be extended as needed.

A. RECONFIGURATION TRIGGER ANALYZER

The reconfiguration trigger analyzer (RTA) forms the foundation for automated reconfiguration management. On the one hand, the RTA is responsible for recognizing that a reconfiguration is required, and on the other hand, it is responsible for analyzing which reconfiguration tasks to perform in the current situation. To identify a reconfiguration need, operators define beforehand what the reconfiguration triggers are to consider. Reconfiguration triggers can be determined with the help of a context analysis procedure, as published in [43]. Whereas the context represents the specific environment and application scenario the RMS will be used in. This means that the exact definition of the reconfiguration triggers is an individual task, an example of one reconfiguration trigger can be found in Section IV-A.

Assuming that a reconfiguration need is identified and based on the premise that the overall goal is to first exploit the flexibility of the current configuration before a reconfiguration is planned, a general procedure to generate a reconfiguration request is provided. Taking advantage of system flexibility helps keep reconfiguration efforts low. Fig. 2 shows the general procedure of the RTA to create a reconfiguration request after a reconfiguration need is identified. Fig. 2 refers to the reconfiguration planning tasks (*i* - *vi*) defined in Section I and specified in more detail in the following section. In general, the focus is on the availability of the required manufacturing capabilities, as this reflects the basic function of a manufacturing system. After ensuring that the manufacturing capabilities are available, the reconfiguration tasks can be used to optimize the configurations. To analyze the availability of the required capabilities, current system capabilities are compared to required capabilities. Only machines that are currently up and running are taken into account. In addition, only the capabilities of the current configuration of each machine are considered. This information is accessed via the data layer and is depicted in Fig. 1 as *configuration knowledge* and *system state.*

If the capabilities are available, the system flexibility is exploited by considering a reconfiguration on the system software level, so that no structural changes of the system are considered and the reconfiguration effort is minimized. For this purpose, the reconfiguration tasks path planning (*iii*) and scheduling (*iv*) are consulted. If the capabilities are not available, on the machine level the flexibility is exploited. Machine flexibility includes the change of parameter settings as well as automated structural changes, e.g., tool turret. These changes must be explicitly specified in the capability models for the capability management (*v*) task. If the capabilities can be provided after exploiting the flexibility on the machine level, the reconfiguration tasks path planning (*iii*) and scheduling (*iv*) can be used to optimize the configuration on the system level. If the capabilities are not available through the flexibility of the current system configuration, a reconfiguration on machine and system level including hardware changes is required. First, the reconfiguration tasks on the machine level are executed. The capability management (*v*) identifies the machine configurations to provide the required capability. Next, interoperability management (*vi*) identifies constraints between machines that may affect system-level reconfiguration. If no configuration is found that provides the required capabilities, the RTA informs the decision maker that no reconfiguration can achieve the required goal. If, by reconfiguration on the machine level, the capabilities can be provided, a reconfiguration on the system level is conducted. In this case, the layout design is also included, as the shop floor may require machines that were not previously included in the current configuration. The reconfiguration analysis is thus structured to first exploit fast solutions, which can be applied without human intervention, which is of explicit importance for operational reconfiguration triggers, and only if these fail, reconfiguration solutions with higher costs and effort are exploited.

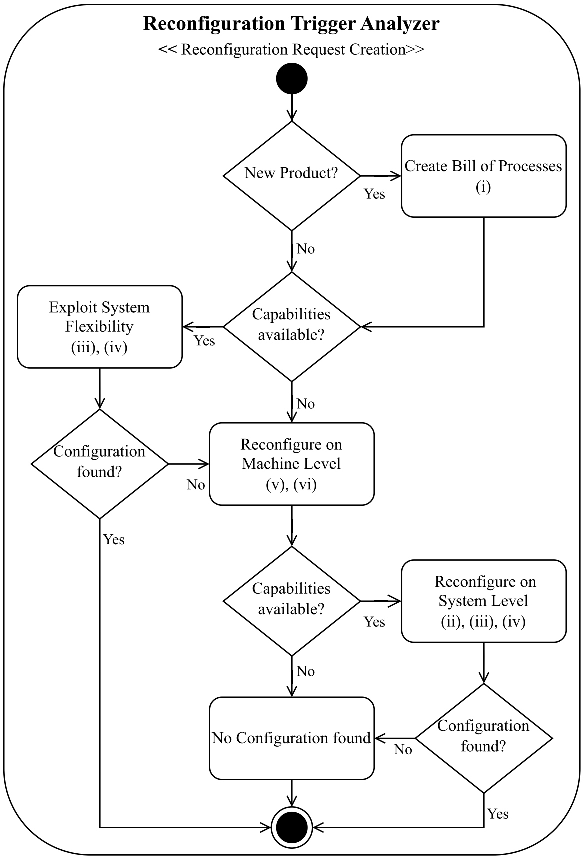

FIGURE 2. Reconfiguration request creation.

B. RECONFIGURATION PLANNING

Reconfiguration planning incorporates the actual tasks to create new configurations on the system and machine level. The tasks represented in the digital twin framework for reconfiguration are the six reconfiguration tasks (*i*-*vi*) described in detail in [16]. However, this does not mean that all tasks have to be implemented. The framework is intended to show what benefits the various tasks can have. Each task requires specific input information to compute a configuration. Some input information is generated by different tasks of the reconfiguration planning itself. If the task that generates the input information for a particular task is not implemented, the required input information must be provided by an external source, e.g., user input. Fig. 3

depicts the in- and output of each framework component of the reconfiguration planning. In addition, the corresponding reconfiguration task performed by the respective component is indicated by the numbering $i - vi$ according to the description in Section III-A.

The process planner needs as input a 3D CAD drawing of the final product. In general, a product p is a final produced good and consists of several components. A component is a discrete manufactured good, which is an assembled part to create the final product p. Each component c_i consists of several features f_{ia}, whereas a feature is a distinctive geometric characteristic of the component. Accordingly, $c_i = \{f_{ia}, \ldots, f_{in}\}$ is a list representing the features that need to be manufactured to create a component. The process planner extracts the product structure including its components and features. Further, for each feature, the possible manufacturing processes are listed, also referred to as a bill of processes (BoP). The list can be created based on a mapping between product features and manufacturing processes, as, for example, described in [44]. Depending on the implementation of the other reconfiguration planning tasks, additional information may be needed that also could be generated by the process planner, e.g., the feature dimensions, the requested raw material or quality requirements. Therefore, each feature is described by attributes relevant to configuration selection and follows $BoP = ci : \{f_{ia} : \{MP_{ia}, RQ_{ia}, RM_{ia}\}\}$. MP_{ia} represents a list of manufacturing processes, which can create the feature. RQ_{ia} represents a list of relevant attributes for the configuration selection of the capability management, e.g., expected quality, feature dimensions, etc. RM_{ia} describes the raw material of the component the feature belongs to.

The capability manager needs as minimal input information the required manufacturing processes, as well as the available machines M_i. The capability manager generates a list of machines capable of providing the required manufacturing processes $MS = ci : \{f_{ia} : \{M_i, \ldots, M_n\}\}$. With further details like a quality requirement or the feature dimensions, defined by RQ_{ia}, the machine configuration selection is more precise. Further, if no selection objective is given as input, by default a configuration is selected, which fulfills the requirements with minimal change to the current configuration. Taking into account the current state of the system, the capability management requests a list M_i with the available machines from the data layer, to only consider the capabilities of the up and running machines. The output list of the capability manager can be enriched with evaluation information, e.g., estimated processing time for feature f_{ia}.

The interoperability manager takes as input the machine list MS provided by the capability manager and provides as output, constraints Co_{iM_i} for each machine M_i on the list, regarding predecessor or successor, as well as requirements for specific connectors or installation area. These constraints are taken into account by the path planner and the layout designer.

The path planner takes into account the constraints Co_{iM_i} provided by the interoperability manager, as well as the list of machines capable of manufacturing a certain feature (MS), provided by the capability manager. As output, the path planner provides a set of feasible configurations $SC = \{S_i, \ldots, S_n\}$, each including a selection of machines, each assigned to manufacture specific features. The selection can be based on the evaluation information of the capability manager or on other operational criteria like machine utilization, reliability, etc.

The set of configurations (SC) is the input for the layout designer besides the information of available shop floor positions SF_i and their attributes, e.g., provided connections, sizes, etc., as well as the constraints Co_{iM_i} of the selected machines provided by the interoperability manager. The output of the layout designer is a layout plan L_i for each configuration S_i with the machines assigned to dedicated positions, satisfying all requirements and optimized to minimize the transportation time and/or handling time. For a minimization of these, a sequence Sq_i representing an order in which the features of each component can be manufactured must be taken into account. It may be that multiple sequences are feasible for the features of one component. If no dedicated sequence is provided, it is assumed that the features need to be manufactured following the order the features are listed and that each component can be manufactured in parallel.

The scheduler creates a schedule Sch_i representing the information when is which feature manufactured on which machine, for each configuration of the set SC in accordance to the sequence Sq_i and layout L_i. This takes into account the amount of requested products n_p in an order, as well as already scheduled orders.

The described sequence of reconfiguration planning tasks represents an idealized procedure that takes into account one product at a time. Considering multiple products at a time requires several iterations of the reconfiguration planning tasks, especially an advanced scheduling algorithm.

It should be noted that the decisions of the respective reconfiguration planning tasks are interdependent, so a better solution may be achieved if an integrated approach is applied. An integrated approach can be appropriate when it has been identified that a new product is being introduced and the production system is in a corresponding phase where an all-encompassing reconfiguration is possible. However, a specific response to certain triggers can only be achieved by modularizing reconfiguration planning into tasks. The modularization can be enforced by a separate implementation or by an integration of a selective execution of certain functions in a monolithic implementation.

C. DECISION MAKER

The decision maker is responsible for the selection of one configuration from the set provided. The first step is to evaluate each configuration of the configuration set based on the operator defined evaluation criteria. Depending on the consideration of single or multiple evaluation criteria,

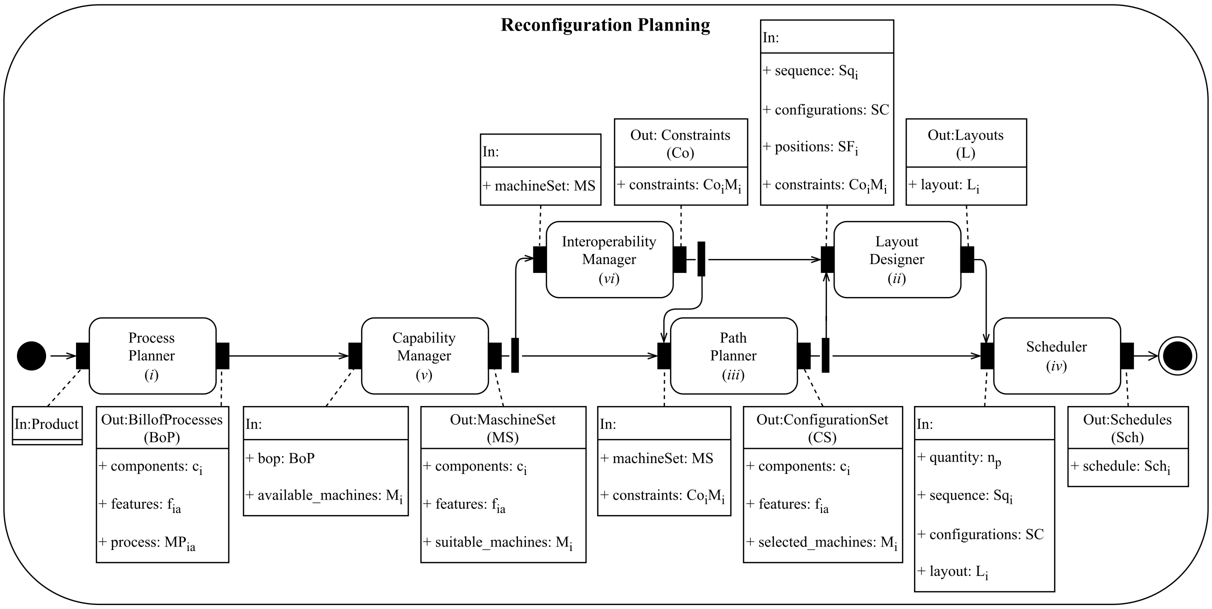

FIGURE 3. Relation of reconfiguration planning tasks.

the decision procedure is distinguished (Fig. 4). If multiple evaluation criteria are used, it is first analyzed whether there is one configuration that dominates all other configurations. A dominant configuration reaches better scores for each evaluation criterion than the other configurations in the set. If the configuration set does not contain a dominant configuration, a Pareto frontier is created. A Pareto frontier is composed of the configurations that dominate other configurations in one or more evaluation criteria, a so-called non-dominant set. A non-dominant set includes the best configurations, but does not allow any conclusions about a definite winner. To make a decision, other decision aspects must be taken into account. Strategic aspects are suitable here, e.g., reduce the number of used machines, always select a solution with the least reconfiguration effort, etc. [45]. Strategic aspects for instance can be expressed in the form of weights for each individual evaluation criterion. This enables a definite selection of one configuration of the non-dominant set.

A similar procedure is applied if only a single evaluation criterion is considered. Instead of creating a Pareto frontier, a ranking is created based on the scored value of each configuration of the configuration set. If no unique winning configuration can be selected, the configurations with the same highest score are evaluated based on strategic aspects. Additional evaluation criteria may need to be introduced if the selected evaluation criteria cannot be broken down into several values, e.g., production time.

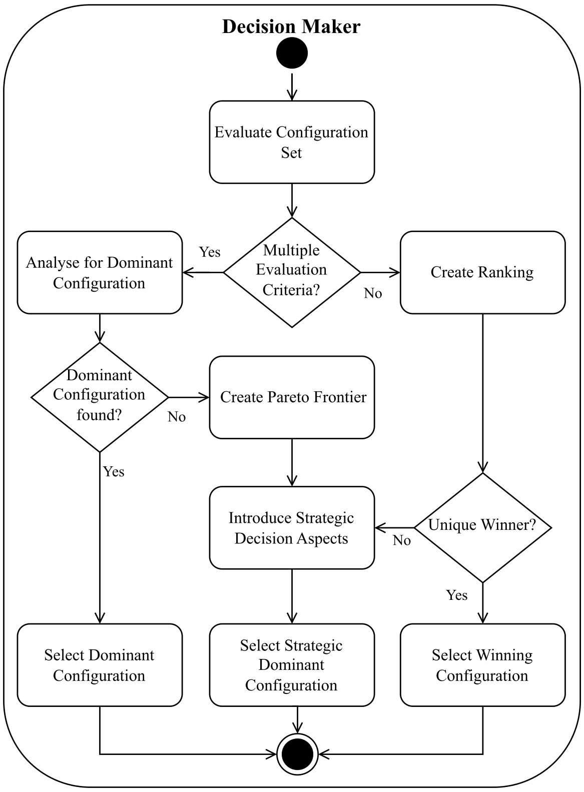

FIGURE 4. Decision procedure of the decision maker.

IV. EVALUATION OF THE DIGITAL TWIN FRAMEWORK FOR RECONFIGURATION MANAGEMENT

To evaluate the digital twin framework for reconfiguration management, a four-stage approach is applied. In order to validate if the required functions, as stated in the research question, are included in the developed framework, a case study is conducted. Based on the case study, no statement about the quality of the presented solution can be made. Therefore, the framework will be analyzed for redundant components, completeness, and if it follows good practice.

The remainder of this section is structured as follows. Section IV-A presents the conducted case study based on the laboratory factory of the University of Michigan. Section IV-B presents an analysis of whether each component of the framework contributes to the fulfillment of at least one of the required framework functions. Section IV-C analyzes, based on a systematic literature review, which framework components are used by existing reconfiguration management approaches. Finally, in Section IV-D it is analyzed if the framework follows good practice.

A. CASE STUDY

In this section, a case study implementation is presented to show the application of the presented framework. Therefore, in this case study, a representative use case is selected, where the interaction between reconfiguration planning tasks and the reconfiguration trigger analysis, as well as the decision maker, is shown.

The considered manufacturing system is the test bed of the smart manufacturing group of the University of Michigan. The test bed M_{UM} consists of two CNC machines M_1 and M_2, two 3D printers from different brands M_3 and M_4, as well as a mobile Kuka robot R_1 to load and unload the machines and an assembly desk A_1. In the initial configuration the test bed includes one 3D printer, one CNC machine, the mobile Kuka robot, and the assembly work station $M_{UM} = \{M_1, M_3, R_1, A_1\}$. The layout of the test bed is shown in Fig. 5, including three unused spaces for extension and the travel time of the robot between the machines and the assembly workstation.

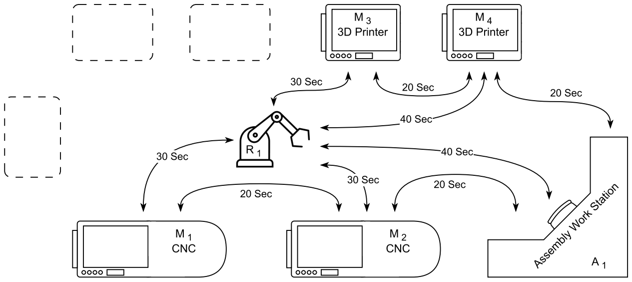

FIGURE 5. **Testbed Layout t_0.**

The test bed produces a model truck p consisting of seven parts, a chassis c_1, a trunk c_2, a cabin c_3, and four wheels $\{c_4, c_5, c_6, c_7\}$ (Fig. 6). The chassis is made of a raw part by face milling the assembly pins $MP_{11-12} = \{FaceMilling\}$ and drilling the wheel assembly holes $MP_{13-16} = \{Drilling\}$. The trunk and the cabin can be either 3D printed or milled, i.e. MP_{21-22} and $MP_{31-33} = \{Milling, 3DPrinting\}$. Since the wheels are purchased parts, therefore no further component features or manufacturing processes are listed (cf. Table 3 and Table 4).

As a strategic trigger, the demand change is considered. In this case, the demand change results in the same requested product, but a higher quantity is requested. Table 3 shows the exemplary product structure of the model truck and each component's features. In the following subsections, it is demonstrated how the capability manager interacts with the path planner and the decision maker. Each represents one possible solution to fulfill the respective task. There is a special focus on the distribution of tasks and the openness of the manufacturing system to take into account that with each reconfiguration planning machines can be added or removed. The modularization shows in particular the adaptability of the framework to different needs. An integrated optimization algorithm tailored to an RMS with a fixed number of machines and known capabilities would likely produce better results.

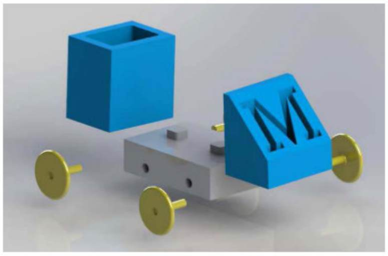

FIGURE 6. **Exploited view of the model truck.**

TABLE 3. **Overview of the model truck product structure.**

- p: Model Truck
  - c_1: Chassis
    - f_{11-12}: Assembly Pin
    - f_{13-16}: Wheel Assembly Hole
  - c_2: Trunk
    - f_{21}: Trunk Cutout
    - f_{22}: Assembly Cutout
  - c_3: Cabin
    - f_{31}: M Engraving
    - f_{32}: Inclined Surface
    - f_{33}: Assembly Cutout
  - c_{4-7}: Wheel

1) CAPABILITY MANAGER IMPLEMENTATION

To describe the capabilities of each machine, we use context-sensitive attributed feature models (CAFMs). Feature models capture common and variable parts of a machine on a conceptual level as features in a hierarchical tree. A machine is described by several features, each feature representing significant modules or components that can be selected or deselected to adapt to new requirements by altering the functionality of the machine [46]. A feature can be either ''mandatory'', i.e., must be selected upon parent selection, or ''optional'', i.e., can be selected upon parent selection. Optional features can be structured in ''or'' groups, i.e., at least one group feature must be selected, or ''alternative'' groups, i.e., exactly one group feature must be selected. Additional relations between features are expressed using cross-tree constraints, which are logical formulas between features. Additionally, each feature can be described with attributes to distinguish certain features, e.g., the diameter of a tool, and support the configuration selection based on an objective function. With the use of feature models, it is determined which configurations a machine can take. A valid configuration is described by the sum of the selected features [47], [48]. Mauro et al. [49] introduced a concept to relate the features of a feature model to context information, which can trigger a reconfiguration. This context information can be used in cross-tree constraints to restrict or enforce the selection of features in a certain context. To select a configuration, we make use of the HyVarRec[1] reconfiguration engine. To derive a configuration, HyVarRec encodes feature models into propositional formulas. Each feature is translated into a variable and each relationship between features or constraints is translated into a formula. To use attributes or arithmetic constraints, the satisfiability modulo theory (SMT) solver Z3[2] is used, which allows the replacement of variables by predicates [50].

TABLE 4. Model truck process plan.

p: Model Truck		
	c_1: Chassis	
		MP_{11-12}: Face Milling
		MP_{13-16}: Drilling
	c_2: Trunk	
		MP_{21-22}: 3-Angle Milling, 3D Printing
	c_3: Cabin	
		MP_{31}: 5-Angle Milling, 3D Printing
		MP_{32}: 5-Angle Milling, 3D Printing
		MP_{33}: 3-Angle Milling, 3D Printing
	c_4 - c_7: Wheel	None

TABLE 5. Minimal capability management output.

p: Model Truck		
	c_1: Chassis	
		f_{11}: $\{M_1, M_2\}$
		f_{12}: $\{M_1, M_2\}$
		f_{13}: $\{M_1, M_2\}$
		f_{14}: $\{M_1, M_2\}$
		f_{15}: $\{M_1, M_2\}$
		f_{16}: $\{M_1, M_2\}$
	c_2: Trunk	
		f_{21}: $\{M_1, M_2, M_3, M_4\}$
		f_{22}: $\{M_1, M_2, M_3, M_4\}$
	c_3: Cabin	
		f_{31}: $\{M_1, M_2, M_3, M_4\}$
		f_{32}: $\{M_1, M_2, M_3, M_4\}$
		f_{33}: $\{M_1, M_2, M_3, M_4\}$

To calculate the estimated production time for each manufacturing process provided by a machine a suitable calculation formula is needed. The considered case study includes CNC machines that provide a milling process, which will be evaluated based on the material removal rate Q. The material removal rate describes the removed material volume per minute and is a common evaluation parameter for milling processes.

For the 3D printing process of the 3D printers, a different measure is necessary, as it is an additive process where the component is built layer by layer instead of removing material from a raw part. For the calculation of the estimated processing time a slicer software is used. A slicer software creates based on the 3D drawing and the parameter settings like filling density, layer height, nozzle diameter, printing speed, printing material, and the used 3D printer, a control code for the printer and estimates the printing time. A difference from the estimated processing time for milling is that it is not individual features that are manufactured but whole components. Therefore, the comparison and selection of the machines have to be based on the component level instead of the feature level.

2) PATH PLANNER IMPLEMENTATION

As the implementation of the path planning task, a genetic algorithm (GA) is selected. A GA is a popular meta-heuristic algorithm and is an evolutionary-inspired approach [51]. In general, to initialize a GA, a starting population (Y) is generated randomly. Each Y consists of n chromosomes (X_i) [52]. A chromosome consists of genes. A binary encoding is used, so that the input scheme, provided by the capability manager, needs to be translated into binary form. Therefore, each gene follows the encoding: M_i produces f_{ia}, which can be true (value = 1) or false (value = 0). Based on the provided input (Table 5), the computed chromosome consists of 32 genes. Each chromosome X_i of a population Y is evaluated based on a fitness function f. For this case study, the minimization of the unit processing time f_{UPT} (Eq. 1) is selected as the fitness function. The unit processing time is the sum of the production time for each feature f_{PT} (Eq. 2) and the set up time f_{SUT} (Eq. 3), which is the sum of the reconfiguration time that each machine needs to provide the required manufacturing process and the set up time of each machine for each component that is processed by that machine. Assuming that each component only requires one clamping per machine, neglecting further clamping may be needed for different features.

After the first population is generated and evaluated, the next generation is created. Therefore, two chromosomes are selected based on their fitness, i.e. the scored value of the fitness function. These chromosomes are used to create offsprings based on a defined cross-over probability (CX_p). Furthermore, with the help of a defined mutation probability (M_p), some of the genes of the offspring are flipped randomly. Following this procedure, a new population is created. The use of a population, as well as cross-over and mutation, ensures that the solution does not approximate a local optimum [51]. The repetition of creating new generations is stopped when a certain threshold or the maximum of generations is reached.

The problem considered here, the selection of machines to manufacture a particular product feature, is a non-linear constrained problem. To use GAs to solve constraint problems, constraint-handling means need to be incorporated [51]. To transform the constraint problem into a non-constraint problem, a static penalty function is introduced. The penalty function is the most common technique, but

---

[1] https://github.com/HyVar/hyvar-rec

[2] https://github.com/Z3Prover/z3

requires a high population size and a high number of generations [51]. To ensure the manufacturability of the product, each feature needs to be manufactured by at least one machine. For each chromosome, this constraint is checked, and if it is violated, a penalty time is assigned to the fitness of the chromosome instead of the fitness value. The constraint that if 3D printing is selected for a feature of a component, all features of that component must be 3D printed is also covered by a penalty function. The penalty time is assigned to the fitness of a chromosome if this constraint is violated, e.g., if a component is manufactured by milling and 3D printing.

For this case study, a two-point crossover with a crossover probability CX_p of 0.5%, a mutation rate M_p of 0.2%, and the tournament chromosome selection technique are used. A starting population size of 600 and a number of 100 generations are used. The ten best chromosomes are saved and passed to the decision maker. The implementation is based on DEAP[3] an evolutionary computation framework [53]. The execution of the GA took 4.353424 seconds running on a standard PC equipped with a 1.8GHz CPU and 16GB RAM. Fig. 7 shows the performance of the GA. The blue line represents the average fitness (Eq. 1) of the considered generation. It shows that after about ten generations, much fewer non-valid configurations are created.

$$min f_{UPT} = \sum (f_{PT} + f_{SUT}), \tag{1}$$

$$f_{PT} = \sum_{i=1}^{n} PT_{M_i f_{ia}} \cdot x_{f_{ia}}, \tag{2}$$

$$f_{SUT} = \sum_{i=1}^{n} SUT_{M_i} \cdot x_{c_i} + \sum_{i=1}^{n} RT_{M_i f_{ia}} \cdot x_{f_{ia}} \tag{3}$$

where:

f_{PT}: is the estimated production time
f_{SUT}: is the estimated set up time including the reconfiguration time
$PT_{M_i f_{ia}}$: is the production time of machine M_i to manufacture feature f_{ia}
$x_{f_{ia}}$: 1 if feature f_{ia} is manufactured on machine M_i
SUT_{M_i}: set up time of machine M_i
x_{c_i}: 1 if machine M_i manufacture any of the features of component c_i
$RT_{M_i f_{ia}}$: reconfiguration time of machine M_i to change from initial configuration into the configuration to manufacture feature f_{ia}

3) DECISION MAKER IMPLEMENTATION

To select a configuration the decision maker needs to be provided with one or several selection criteria and a set of feasible configurations. As a selection criterion, the throughput rate per day is selected. The throughput rate is calculated according to ISO 22400-2 based on actual production times and delay times. However, reconfiguration planning requires working with predicted times, Fig. 8 shows the simplified representation of the throughput rate per day (p/d). This use case considers the throughput rate per day, where one day contains three 8-hour shifts. To follow the ISO 22400-2 the reconfiguration time is considered as part of the set up time. The decision maker receives a 10 best-of feasible configurations list determined by the path planner. Each of the configurations is evaluated in the same manner as described in Equations 2 and 3. Additionally, the transportation time needs to be calculated.

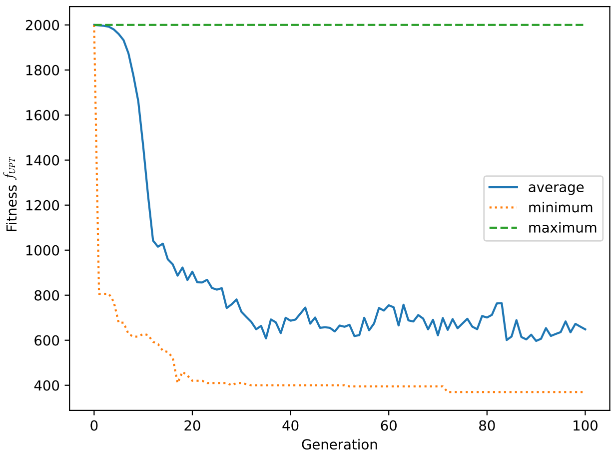

FIGURE 7. Overview of the GA performance.

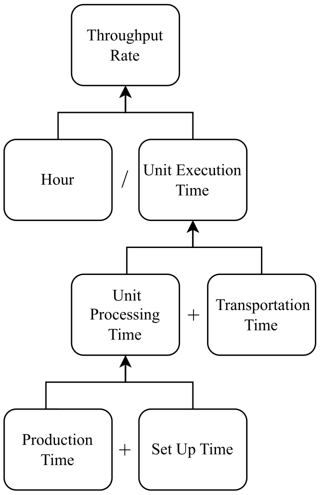

FIGURE 8. Throughput rate, simplified representation based on ISO 22400-2.

As in this case study, no layout designer is considered, the layout for each configuration remains the same as depicted in Fig. 5. Based on the layout and current transportation times, a matrix is created (Table 6). The matrix reads as follows; the transport from a machine of a certain row to a machine in a certain column takes x seconds. It is assumed that the robot is positioned in the middle between all machines. The robot provides the machines with the raw material and

---

[3] https://github.com/DEAP/deap

picks up the components. Components are stored within the robot and transported to the assembly workstation when all components are ready for assembly. Furthermore, it is assumed that all features are manufactured in the sequence listed in Table 3 and, therefore, do not include the acceleration of the unit processing time. The transportation time for each configuration is calculated accordingly, whereas it is assumed the robot waits in front of the machine until it is finished.

TABLE 6. **Transportation time matrix according to Fig. 5 (in s).**

to from	M_1	M_2	M_3	M_4	R_1	A_1
M_1	0	20	60	70	30	70
M_2	20	0	60	70	30	20
M_3	60	60	0	20	30	70
M_4	70	70	20	0	40	20
R_1	30	30	30	40	0	40
A_1	70	20	70	20	40	0

The selection procedure follows the defined flow for a single selection criterion presented in Section III-C, Fig. 4. If one configuration reaches a higher throughput rate than the other configurations, this configuration is selected. If there are several configurations with the same throughput rate, further criteria or operator preferences can be considered to select one configuration. In this case study the configuration with the lowest reconfiguration time is selected, as reconfiguration is still assumed to have the greatest deviation in execution time and generally is considered as a non-value-adding activity.

Table 7 represents a comparison of the current configuration t_0 and the new configuration t_1 selected by the decision maker. Instead of using the 3D printer M_3 for components c_2 and c_3, the 3D printer M_4 is selected for component c_3. Furthermore, the CNC machine M_1 is replaced by the CNC machine M_2 manufacturing component c_1.

4) CASE STUDY SUMMARY

This case study successfully demonstrated that an individual solution for reconfiguration management can be derived from the framework. Two tasks of reconfiguration planning have been implemented independently using different solution approaches. The interaction of these tasks is based on the definition of input and output schemata respectively. The functional requirements operator support, including different economic factors, considering different reconfiguration triggers, as well as updating the reconfiguration space can be satisfied by using the framework as a blueprint.

In general, the case study revealed aspects that could be improved by considering further reconfiguration planning tasks. For example, the inclusion of scheduling could allow the production of individual components to be parallelized.

TABLE 7. **Comparison of configuration t_0 & configuration t_1.**

			Configuration t_0		Configuration t_1	
Performance			TR	2.75 p/d	TR	3.9 p/d
			PT	281 min	PT	253 min
			RT	90 min	RT	60 min
			SUT	150 min	SUT	120 min
			TT	3.33 min	TT	2.83 min
Machine Usage	c_1	f_{11}	M_1		M_2	
		f_{12}	M_1		M_2	
		f_{13}	M_1		M_2	
		f_{14}	M_1		M_2	
		f_{15}	M_1		M_2	
		f_{16}	M_1		M_2	
	c_2	f_{21}	M_3		M_3	
		f_{22}	M_3		M_3	
	c_3	f_{31}	M_3		M_4	
		f_{32}	M_3		M_4	
		f_{33}	M_3		M_4	

TR: Throughput Rate, according to Fig. 8
PT: Production Time, according to Eq. 2
RT: Reconfiguration Time, according to Eq. 3
SUT: Set Up Time, according to Eq. 3
TT: Transportation Time, according to Table 6

The implemented solution always considers the entire product at once. In addition, the reconfiguration time is calculated based on the configuration of the machine at the time of planning. Considering the sequence of the process steps could lead to a reduction in reconfiguration time. Whereas the consideration of the transport time is negligible here. Thus an implementation of a layout designer would have little effect.

B. REDUNDANCY ANALYSIS

As stated in Section II-B, four functional requirements are derived from the research question stated in Section I. To access the redundancy, each framework component must contribute to at least one of the functional requirements. An overview of the redundancy analysis is presented in Table 8.

1) F1 (OPERATOR SUPPORT)

The reconfiguration planning tasks are each represented by one component: path planner, scheduler, layout designer, interoperability manager, process planner, and capability manager. Each of these helps to explore the solution space and generate solutions for reconfiguration. Therefore, the reconfiguration planning does not have to be conducted by the operator. Further, the components reconfiguration trigger analyzer and decision maker also support the operator. The reconfiguration trigger analyzer monitors the defined triggers and initiates the reconfiguration planning. The latter instead presents solutions via the component eastbound interface to the operator and if enabled selects a reconfiguration solution for the operator.

2) F2 (CUSTOMIZED OPTIMIZATION)

The components decision maker and eastbound interface support customized optimization by enabling an adaptation of the optimization criteria and optimization function. However, only optimization criteria can be introduced for which the data is available.

3) F3 (OPERATIONAL AND STRATEGIC RECONFIGURATION TRIGGERING)

To fulfill this requirement, data needs to be collected and processed from the RMS itself and peripheral systems, e.g., MES or ERP. Therefore, the components data processing, data collection, and eastbound interface are required. Finally, the component reconfiguration trigger analyzer uses this data to monitor the triggers and initiate reconfiguration planning when needed.

4) F4 (DYNAMIC RECONFIGURATION SPACE)

To obtain a dynamic reconfiguration space, i.e. adapted to the current system state, the components data processing, data collection, and reconfiguration trigger analyzer are required. As stated above, data processing and data collection provide the corresponding data. Further, the reconfiguration trigger analyzer takes these into account to decide which reconfiguration planning task to initiate based on the current system state and the current trigger. This dynamically adjusts the reconfiguration space depending on the trigger and system state.

TABLE 8. **Redundancy analysis.**

	Reconfiguration Planning						Data Layer					
Requirement	Path Planner	Scheduler	Layout Designer	Interoperability Manager	Process Planner	Capability Manager	Data Processing	Data Collection	Configuration Knowledge	Decision Maker	Reconfiguration Trigger analyzer	Eastbound Interface
F1	✓	✓	✓	✓	✓	✓				✓	✓	✓
F2										✓		✓
F3							✓	✓			✓	✓
F4							✓	✓	✓		✓	

C. COMPLETENESS ANALYSIS

To assess whether the framework is complete or misses important components, a systematic literature review (SLR) is conducted. In general, an SLR is defined as a structured strategy for assessing previous literature findings. There are various methods for conducting an SLR, depending on the research domain and the intended outcome. The conducted SLR is based on [54] and [55] for detailed insights see [30]. To identify the relevant literature the following search strings resulting from the combination of the two categories are defined:

- **Reconfiguration Management** and related concepts: reconfiguration management, reconfiguration, adaption, self-adaption, self-adapting, self-configuration, self-configuring, adaptation, self-adaptiveness
- **Domains** in which reconfiguration is predominant: cyber-physical system/s, cyber-physical production system/s, manufacturing system/s, reconfigurable manufacturing system/s, flexible manufacturing system/s, adaptable manufacturing system/s, cpps, cps, rms, rmt

Based on the defined search stings, an automatic extraction of the literature from the scientific databases IEEEXplore, SpringerLink, and ScienceDirect was conducted with the help of the tool described in [56]. The search was limited to publications published no earlier than 2018, as this was the first year that a significant increase in interest in digital twins and manufacturing was evident in Google search trends. As shown in Fig. 9, a total of 404 publications were retrieved from the scientific databases and screened for inclusion. For an inclusion decision, the abstract of each publication is independently screened by two researchers to decide whether it should be included in the full-text analysis or not. In particular, a publication was included if it considers at least one of the above stated aspects of reconfiguration management and one of the following domains: Discrete Manufacturing, Cyber-Physical Systems, or Reconfigurable Manufacturing Systems. Additionally, publications were excluded if they were surveys themselves or if the provided use case was not related to reconfiguration at the machine/shop floor level (e.g., FPGA reconfiguration). Generally, if there were any doubts about the eligibility of a publication, a third researcher was consulted for the final decision, to decide if the publication is suitable for the full-text review. A detailed description of the inclusion and exclusion criteria can be found in [30]. Of the 404 publications, 67 publications were included in the detailed literature analysis.

The 67 publications were read carefully and if one of the components was addressed in the present approach a point was awarded for the respective component. In total, 67 points could be scored for each component, which was not reached by any of the components. That means that no component of the digital twin framework for reconfiguration management was included in each of the publications. At the same time, no component was not considered at all. Fig. 10 presents the results of the analysis in total numbers and percentages. The most considered reconfiguration planning

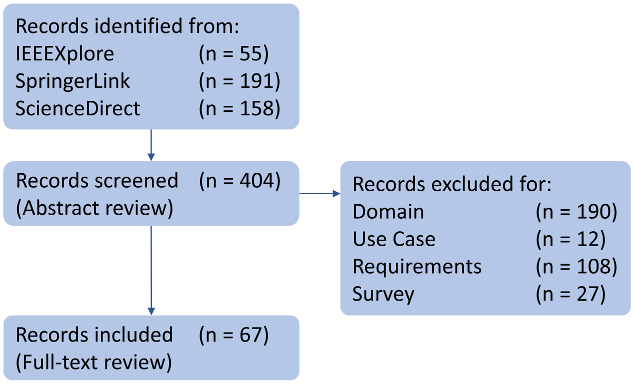

FIGURE 9. **Overview of the SLR method.**

component is the capability manager, considered by about 80% of the publications. Interoperability management and data processing are considered much less frequently. Especially interesting is that more often data collection is taken into account than data processing. Further, configuration evaluation should be added as subcomponent of the decision maker to the framework, as this was mentioned explicitly in a high amount of publications, but is only considered implicit in the presented digital twin framework for reconfiguration management. The results for each publication are presented in Table 10 in Appendix A.

In summary, none of the defined framework components are identified as not relevant and no further component could be identified that needs to be included. The latter was concluded because reading the papers did not reveal any functionality not covered by the framework components.

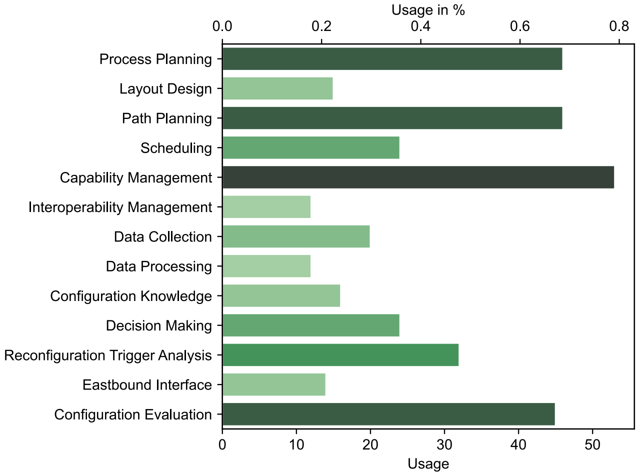

FIGURE 10. **Result of completeness analysis.**

D. GOOD PRACTICE ANALYSIS

Moyne et al. [24] present a set of 32 requirements a digital twin framework should comply with to follow good practice. These requirements are clustered into three categories: requirements related to the DT definition and its today usage (Table 9, i.1-i.11), requirements related to DT applications in the near future (Table 9, ii.1-ii.16), and requirements related to DT applications in the longer term (Table 9, iii.1-iii.4). The requirements from category *i.* are considered mandatory, whereas each fulfilled requirement from category *ii.* and *iii.* as a benefit to complement the best practice solution. Table 9 lists all 32 requirements and the analysis result. If the developed DT framework for reconfiguration management meets the requirement, a ■ is set, otherwise a □. In the following, the mandatory requirements of category *i* are described, as well as the fulfilled requirements of category *ii* and *iii*.

The modular design of the framework in general and especially of the reconfiguration planning requires to specify the interfaces of each framework component. This allows the reuse and portability of each component solution. At the same time, this has the effect that the framework can be extended as desired. Modularization is the foundation of individualized adaptation of reconfiguration management with respect to project needs. Therefore, requirement *i.1* is fulfilled. Requirement *i.2* is fulfilled as the foundation of the framework is built by data, information, and models stored and provided by the data layer. The framework does not prescribe explicit models, but system state models are required for the reconfiguration planning and the RTA. Furthermore, the configuration knowledge needs to be captured, where configuration models can be used. The framework includes a decision maker that provides the DT client with the reconfiguration solution. Further, the execution of the reconfiguration planning is supported and triggered by the RTA, which leads to the fulfillment of requirement *i.3*. In the current state requirement *i.4* is not fulfilled as the framework does not include narrow DT intelligence, but does provide means to integrate it. For example machine learning algorithms can be integrated to estimate the processing time based on actual data. The RTA combines analytical functions with subject-matter-expertise (SME), fulfillment of *i.5*. Each trigger has to be defined by experts, but automatic analyses observe these during operation and call for action if required. The decision maker selects a reconfiguration plan based on defined objectives. The values of the objectives are estimated based on the currently provided data. These estimated values can be compared to the exact values of execution and therefore deliver quantifiable metrics to assess the DT capability. In the same manner, the precision of the DT can be determined; fulfillment of requirement *i.6* and *i.7*. Requirement *i.8* is satisfied, since the added value can be calculated on the basis of the selected optimization objectives. The evaluation of the added value created by the DT is therefore individually based on the operator preferences. For this purpose, the objectives can be integrated into a company-valid KPI system. Correct or incorrect operation of the DT is reflected in the achieved added value and is thus ascertainable and quantifiable, which fulfills requirement *i.9*. Requirements *i.10* is fulfilled due to the modular design of the framework and the interface definition, which allows to evolve individual components without causing a complete

TABLE 9. Evaluation of best practice, requirements copied from [24, p.8].

#	Requirement	☐
i.1	The DT capability is modular; boundaries and relationship with the interacting physical twinned system are clearly defined.	■
i.2	The DTs use models to replicate an aspect of an asset, process, system or product.	■
i.3	The DT solution must also include some computational engine that supports the model and provides the required capability to the DT client.	■
i.4	The DT must be able to use some form of narrow DT intelligence that allows the DT to provide its capability in a specified application domain.	☐
i.5	The DT must be able to incorporate analytics and subject-matter-expertise in a complementary fashion to realize intelligence in solutions.	■
i.6	The DT must deliver its capability in terms of one or more quantifiable metrics.	■
i.7	The accuracy of the DT in providing these quantifiable metric(s) must be ascertainable and quantifiable.	■
i.8	The DT must deliver a quantifiable net value-add in its application environment.	■
i.9	The financial benefit of correct operation of the DT in its environment as well as the cost of incorrect operation in that environment must be ascertainable and quantifiable.	■
i.10	The framework must support an evolution rather than revolution of capabilities, especially supporting the evolution of existing capabilities to align with the ultimate DT framework vision.	■
i.11	The framework must support the entire DT lifecycle from envisioning and design through development, validation, deployment and maintenance.	■
ii.1	Re-usability: DT solutions must be portable, re-usable and scalable.	■
ii.2	Re-usability: Degree and method of re-usability (e.g., data translation, sub-set of metrics) must be definable.	■
ii.3	Interoperability: Multiple instances of the same DT class must be allowed to interact in a coordinated fashion.	☐
ii.4	Interoperability: Integration of and coordination between instances of different DT classes must be supported.	■
ii.5	Interoperability: Integration of and coordination between DT and non-DT components must be supported.	■
ii.6	Interchangeability: Interchangeability of different instances of the same DT class must be supported.	■
ii.7	Interchangeability: The framework must support standardized definitions of DT structure, baseline minimum abilities, quantifiable capabilities metrics, exposed interfaces, services provided, and behavior exhibited.	■
ii.8	V&V capability: Standardized, reusable and quantifiable V&V processes must be supported.	☐
ii.9	DT maintainability: The minimum required DT output quality (see requirement i.7) to provide its intended capability must be quantifiable. Diagnosability of lack of sufficient quality of DT output (e.g., outside of threshold limits) should be identifiable in a time-critical fashion.	☐
ii.10	DT maintainability: the DT should be updated to continue to provide sufficient output quality if that level of maintenance is a requirement for the DT (e.g., as a form of scheduled maintenance, or as a consequence of the DT providing lack of sufficient output quality, see requirement ii.9).	☐
ii.11	DT maintainability: DT improvement support must include solution model tuning as well as model re-building or replacement; this capability should trend toward being fully automated.	☐
ii.12	DT capability and accuracy: DT solutions must be able to use evolving analytics techniques, including improvements on existing techniques and novel new techniques.	☐
ii.13	DT capability and accuracy: DT solutions must support structured and automated integration of analytics and subject-matter-expertise information.	☐
ii.14	DT extensibility: DT framework must be extensible to support DT solutions across the entire smart manufacturing ecosystem.	■

TABLE 9. ***(Continued.)*** **Evaluation of best practice, requirements copied from [24, p.8].**

#	Requirement	☐
ii.15	DT extensibility: DT framework must address security requirements, including data partitioning and IP security required for DT operation across the entire SM ecosystem.	☐
ii.16	Sustaining a DT technology community: DT framework must provide a common DT definition, taxonomy and other mechanisms that allow the community to collaborate on DT technology, from DT fundamental research through applied research, development, deployment, and maintenance.	■
iii.1	DT framework must support virtual counterparts across the entire smart manufacturing ecosystem (including the full supply chain), that can be used for detection, prediction, prescription and analysis of all aspects of operation.	☐
iii.2	Automation of DT creation, validation and integration must be supported; generally, automation of the entire DT lifecycle must be supported, except for aspects of envisioning and design.	☐
iii.3	DT framework must support an evolution from narrow intelligence towards more general intelligence with fewer context restrictions (e.g., narrow AI to AGI).	☐
iii.4	DT framework must support the marrying of SME and analytics as a continuing integral part of DT capability and evolution.	☐

revision of the DT. At the same time, further capabilities can be added to the DT. The developed framework serves as a blueprint for the engineering of a DT for reconfiguration management. During envisioning the framework can be used to show what a full-fledged reconfiguration management solution looks like and relevant framework components can be selected as needed. The system architecture and interface definition provided by the framework support design and development. The framework can also be used to validate different designs if combined with simulation models. In addition, the modular character also supports deployment and maintenance, as individual components can be deployed and maintained step by step. Therefore, requirement *i.11* is marked as fulfilled.

Next, consider the optional requirements. Requirement *ii.1* and *ii.2* are fulfilled as the framework follows a modular approach that enables re-usability. The reusable components include reconfiguration planning since the interfaces and the required models are defined. These models are provided via the data layer and can be added or updated at any time. The other framework components must be adapted to the respective RMS, as well as the operator goals and preferences. The DT classes referred to in [24] are equal to the components of the framework. Multiple instances of one component are not explicitly foreseen, but for the reconfiguration planning components it can provide benefits if for specific reconfiguration triggers different solutions are required. In exchange, it is required to implement the RTA so that the different instances can be included, which allows to fulfill requirement *ii.3*. Due to the interface definition and specification, the interoperability of the components is ensured. For the interoperability of non-DT components, the eastbound interface is included. At the same time, this enables the interchangeability of the component instances and the extensibility to include not yet foreseen components, fulfillment of *ii.4, ii.5, ii.6, ii.7*, and *ii.14*. Requirement *ii.16* is fulfilled as together with the framework, a detailed definition of a DT was prepared based on relevant publications. In addition, a definition of reconfiguration and associated tasks is provided. Both, as well as the interface description, enable DT community collaboration.

Finally, ten of eleven mandatory requirements are fulfilled. Beyond that nine of sixteen requirements related to the DT application in the near future can be satisfied by the current framework state. Unfortunately, so far none of the requirements for the DT application over the long term can be fulfilled. Requirement *iii.2*, the automation of the DT creation, is considered and supported by the developed method published in [57]. In summary, the degree of compliance with the mandatory requirements is 91% and with the optional requirements is 45%. This allows the conclusion that the DT framework follows a good practice for the current deployment and can be improved for future usage.

V. SUMMARY

In this article, the formation of a digital twin framework to approach reconfiguration management as a holistic problem was presented. The presented framework helps to gain a deep understanding of the tasks required for reconfiguration management. The most relevant tasks and framework components are described and explained in detail. Flow charts are included to present the general procedure, which needs to be customized for the respective application. Each task, depending on the reconfiguration trigger, can be executed to find a solution for the changed requirements of the system configuration. The potential of a DT for reconfiguration management as a solution for considering strategic and operational reconfiguration triggers is highlighted. Furthermore, a modularization of reconfiguration planning is proposed

TABLE 10. Results of the systematic literature review.

#	Process Planner	Layout Designer	Path Planner	Scheduler	Cap. Manager	Int. Manager	Data Collection	Data Processing	Config. Knowledge	Decision Maker	Recon. T. Analyzer	Eastbound Interf.	Config. Evaluation
[58]	0	0	1	1	0	0	0	0	0	0	0	0	0
[59]	1	0	0	0	1	0	0	0	0	0	0	0	1
[60]	1	1	1	0	1	0	0	0	0	0	0	0	1
[61]	0	1	1	0	1	0	0	0	1	0	0	0	1
[62]	1	1	1	0	1	0	0	0	0	0	0	0	1
[63]	1	0	1	1	1	0	0	0	0	1	0	0	1
[64]	1	1	1	0	1	0	0	0	1	0	1	0	0
[65]	1	1	1	1	1	0	0	0	0	0	0	0	1
[66]	0	0	0	1	1	0	1	0	0	0	1	0	1
[67]	1	1	1	0	0	1	0	0	0	0	0	0	0
[68]	1	0	1	0	1	0	0	0	0	0	1	0	1
[69]	1	0	0	0	1	0	1	1	0	0	1	0	1
[70]	1	0	1	1	1	0	1	0	0	0	1	0	1
[71]	1	1	1	1	1	0	0	0	0	0	1	1	1
[72]	0	0	0	0	0	0	0	0	0	0	0	0	0
[73]	0	0	1	0	1	1	0	0	1	0	1	0	0
[74]	1	1	1	1	1	1	0	0	1	1	1	0	1
[75]	1	0	0	1	1	1	1	0	0	0	0	1	0
[39]	1	1	0	1	1	0	1	0	0	1	1	0	1
[76]	1	0	1	1	1	0	0	0	0	1	0	0	0
[77]	1	0	1	1	1	0	1	1	0	0	1	1	0
[78]	1	1	0	1	1	0	0	0	0	0	1	1	0
[79]	1	0	1	0	1	0	0	0	1	1	1	0	1
[80]	1	0	1	0	1	0	1	0	0	0	0	1	0
[81]	1	0	1	0	1	0	0	0	1	1	0	0	1
[82]	0	0	1	0	1	1	1	1	0	0	0	1	0
[83]	0	0	0	0	1	0	0	0	0	0	0	1	0
[84]	1	0	1	0	1	0	0	0	0	0	0	1	1
[85]	1	0	1	0	1	1	0	0	0	1	0	0	1
[86]	1	1	1	1	1	1	1	1	0	1	0	0	1
[87]	1	1	1	0	1	0	0	0	0	1	1	0	0
[88]	1	0	1	1	1	0	0	0	0	0	1	1	1
[89]	0	0	0	0	1	0	0	0	0	0	1	0	0
[90]	1	0	1	0	1	0	1	1	1	1	1	0	0
[91]	1	0	0	1	0	0	1	1	1	1	1	0	0
[92]	0	0	1	0	1	0	1	1	1	1	1	0	1
[93]	1	0	1	0	1	0	0	0	0	1	0	0	1
[94]	1	1	1	1	1	0	0	0	0	0	0	1	1
[95]	1	0	1	0	1	0	0	0	0	0	0	1	1
[96]	1	0	1	0	1	0	0	0	0	1	0	0	1
[97]	0	0	0	0	0	1	0	0	0	0	1	0	0
[98]	1	0	0	0	1	0	0	0	0	1	0	0	1
[99]	0	0	0	1	0	0	0	0	0	0	0	0	1
[100]	1	0	0	0	1	0	1	1	1	1	1	0	1
[101]	1	0	1	1	1	0	0	0	0	1	1	0	1
[102]	1	0	0	0	1	0	1	1	1	1	1	0	1

to allow flexible consideration of different triggers. The framework should serve as a template for the development of reconfiguration management solutions and should not be seen as a closed solution, but as extensible and customizable. Each of the presented components and tasks can be implemented with different solutions and must be supplied with appropriate models and data for use during operation.

In summary, the evaluation of the presented DT framework for reconfiguration management has shown that it includes the required functions as stated in the research question, is complete and doesn't include redundant components, and finally follows good practice, for the current deployment of DTs. The practical implementation of the DT framework for reconfiguration management was shown to a limited extent. In conclusion, further research on its practical application must be conducted.

APPENDIX A
RESULTS OF THE SYSTEMATIC LITERATURE REVIEW

TABLE 10. *(Continued.)* Results of the systematic literature review.

#	Process Planner	Layout Designer	Path Planner	Scheduler	Cap. Manager	Int. Manager	Data Collection	Data Processing	Config. Knowledge	Decision Maker	Recon. T. Analyzer	Eastbound Interf.	Config. Evaluation
[103]	1	0	1	1	1	0	0	0	0	1	0	0	1
[72]	1	0	1	0	0	1	0	0	0	0	0	1	1
[104]	1	0	1	0	0	0	0	0	0	0	0	0	1
[105]	0	0	0	0	0	0	1	0	0	0	1	0	0
[106]	0	0	1	1	0	0	0	0	0	0	1	1	1
[107]	0	0	1	1	0	0	1	0	0	0	1	0	1
[108]	0	0	0	0	1	0	0	0	0	0	0	0	0
[8]	1	1	1	0	1	0	0	0	1	0	1	1	1
[109]	1	0	1	1	1	1	1	0	0	0	0	0	1
[3]	1	1	1	0	1	1	1	1	1	1	1	0	1
[110]	0	0	1	0	1	0	1	1	0	0	1	0	1
[111]	1	0	1	1	0	0	0	0	0	0	0	0	1
[112]	1	0	1	0	1	0	0	0	1	0	0	0	1
[113]	0	0	1	0	1	0	0	0	0	0	1	0	0
[114]	0	0	0	0	1	0	0	0	0	0	1	0	0
[115]	1	0	1	0	1	0	0	0	0	1	0	0	1
[116]	0	0	0	0	0	0	0	0	0	0	0	0	1
[117]	1	0	1	0	1	0	0	0	0	1	0	0	1
[118]	0	0	0	1	1	0	0	0	0	1	0	0	1
[119]	0	0	1	0	0	0	1	1	1	1	1	0	1
[120]	1	0	0	0	1	1	0	0	1	0	1	0	0

REFERENCES

[1] A. Ebrahimi, H. W. Jeon, S. Lee, and C. Wang, "Minimizing total energy cost and tardiness penalty for a scheduling-layout problem in a flexible job shop system: A comparison of four metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 141, Mar. 2020, Art. no. 106295.

[2] X. L. Hoang, A. Fay, P. Marks, and M. Weyrich, "Systematization approach for the adaptation of manufacturing machines," in *Proc. IEEE 21st Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2016, pp. 1–4.

[3] J. Leng, Q. Liu, S. Ye, J. Jing, Y. Wang, C. Zhang, D. Zhang, and X. Chen, "Digital twin-driven rapid reconfiguration of the automated manufacturing system via an open architecture model," *Robot. Comput.-Integr. Manuf.*, vol. 63, Jun. 2020, Art. no. 101895.

[4] J. Morgan, M. Halton, Y. Qiao, and J. G. Breslin, "Industry 4.0 smart reconfigurable manufacturing machines," *J. Manuf. Syst.*, vol. 59, pp. 481–506, Apr. 2021.

[5] A.-L. Andersen, T. D. Brunoe, B. Christensen, M. Bejlegaard, D. G. H. Sorensen, and K. Nielsen, "Tailored reconfigurability: A comparative study of eight industrial cases with reconfigurability as a key to manufacturing competitiveness," in *Reconfigurable Manufacturing Systems: From Design to Implementation* (Springer Series in Advanced Manufacturing), vol. 18, L. Benyoucef, Ed. Cham, Switzerland: Springer, 2020, pp. 209–245.

[6] J. Tang, C. Emmanouilidis, and K. Salonitis, "Reconfigurable manufacturing systems characteristics in digital twin context," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 10585–10590, 2020.

[7] P. Marks, X. L. Hoang, M. Weyrich, and A. Fay, "A systematic approach for supporting the adaptation process of discrete manufacturing machines," *Res. Eng. Des.*, vol. 24, no. 3, pp. 1–21, 2018.

[8] T. Müller, N. Jazdi, J.-P. Schmidt, and M. Weyrich, "Cyber-physical production systems: Enhancement with a self-organized reconfiguration management," *Proc. CIRP*, vol. 99, pp. 549–554, Jan. 2021.

[9] H. H. Benderbal, A. R. Yelles-Chaouche, and A. Dolgui, "A digital twin modular framework for reconfigurable manufacturing systems," in *Advances in Production Management Systems. Towards Smart and Digital Manufacturing* (IFIP Advances in Information and Communication Technology), vol. 592, B. Lalic, V. Majstorovic, U. Marjanovic, G. von Cieminski, and D. Romero, Eds. Cham, Switzerland: Springer, 2020, pp. 493–500.

[10] T. Nakano, K. Daiki, H. Chen, I. Kovalenko, E. Balta, Y. Qamsane, and K. Barton, "Manufacturing line design configuration with optimized resource groups," in *Proc. IEEE 17th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2021, pp. 625–632.

[11] A. Singh, S. Gupta, M. Asjad, and P. Gupta, "Reconfigurable manufacturing systems: Journey and the road ahead," *Int. J. Syst. Assurance Eng. Manage.*, vol. 8, no. S2, pp. 1849–1857, Nov. 2017.

[12] L. Qasim, "System reconfiguration: A model based approach; from an ontology to the methodology bridging engineering and operations," Ph.D. dissertation, Laboratoire Génie Industriel, Université Paris-Saclay, Paris, France, Dec. 2020.

[13] A. Madni, C. Madni, and S. Lucero, "Leveraging digital twin technology in model-based systems engineering," *Systems*, vol. 7, no. 1, p. 7, Jan. 2019.

[14] B. A. Talkhestani, T. Jung, B. Lindemann, N. Sahlab, N. Jazdi, W. Schloegl, and M. Weyrich, "An architecture of an intelligent digital twin in a cyber-physical production system," *Automatisierungstechnik*, vol. 67, no. 9, pp. 762–782, Sep. 2019.

[15] B. Caesar, D. M. Tilbury, K. Barton, and A. Fay, "Requirements for reconfiguration management for manufacturing systems," *IFAC-PapersOnLine*, vol. 55, no. 10, pp. 3214–3219, 2022.

[16] B. Caesar, D. M. Tilbury, K. Barton, and A. Fay, "Digital twin framework for reconfiguration management," in *Proc. IEEE 18th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2022, pp. 1807–1813.

[17] F. Karl, G. Reinhart, and M. F. Zaeh, "Strategic planning of reconfigurations on manufacturing resources," *Proc. CIRP*, vol. 3, pp. 608–613, Jan. 2012.

[18] M. L. Pinedo, "Introduction," in *Scheduling*. Cham, Switzerland: Springer, 2016, pp. 1–10.

[19] N. Brahimi, A. Dolgui, E. Gurevsky, and A. R. Yelles-Chaouche, "A literature review of optimization problems for reconfigurable manufacturing systems," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 433–438, 2019.

[20] A. R. Yelles-Chaouche, E. Gurevsky, N. Brahimi, and A. Dolgui, "Reconfigurable manufacturing systems from an optimisation perspective: A focused review of literature," *Int. J. Prod. Res.*, vol. 59, no. 21, pp. 6400–6418, 2021.

[21] M. P. Groover, *Automation, Production Systems, and Computer-Integrated Manufacturing*, vol. 29, no. 4. London, U.K.: Pearson, 2009.

[22] P. Scully, "How the world's 250 digital twins compare? Same, same but different," IOT Anal., Hamburg, Germany, Tech. Rep., 2020.

[23] G. Steindl, M. Stagl, L. Kasper, W. Kastner, and R. Hofmann, "Generic digital twin architecture for industrial energy systems," *Appl. Sci.*, vol. 10, no. 24, p. 8903, Dec. 2020.

[24] J. Moyne, Y. Qamsane, E. C. Balta, I. Kovalenko, J. Faris, K. Barton, and D. M. Tilbury, "A requirements driven digital twin framework: Specification and opportunities," *IEEE Access*, vol. 8, pp. 107781–107801, 2020.

[25] J. Abonyi, P. Arva, S. Nemeth, C. Vincze, B. Bodolai, Z. D. Horváth, G. Nagy, and M. Németh, "Operator support system for multi product processes-application to polyethylene production," *Comput. Aided Chem. Eng.*, vol. 14, pp. 347–352, Jan. 2003.

[26] H.-P. Wiendahl, J. Reichardt, and P. Nyhuis, "Factory change drivers," in *Handbook Factory Planning and Design*, H.-P. Wiendahl, J. Reichardt, P. Nyhuis, and R. Rossi, Eds. Berlin, Germany: Springer, 2015, pp. 1–12.

[27] P. Eirinakis, G. Kasapidis, I. Mourtos, P. Repoussis, and E. Zampou, "Situation-aware manufacturing systems for capturing and handling disruptions," *J. Manuf. Syst.*, vol. 58, pp. 365–383, Jan. 2021.

[28] U. Gangoiti, A. López, A. Armentia, E. Estévez, and M. Marcos, "Model-driven design and development of flexible automated production control configurations for industry 4.0," *Appl. Sci.*, vol. 11, no. 5, p. 2319, Mar. 2021.

[29] E. Hajjem, H. H. Benderbal, N. Hamani, and A. Dolgui, "Digital twin framework for reconfigurable manufacturing systems: Challenges and requirements," in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*, vol. 631, A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero, Eds. Cham, Switzerland: Springer, 2021, pp. 553–562.

[30] T. Müller, B. Caesar, W. Matthias, S. Ferhat, F. Sahlab, A. Fay, R. Oger, N. Jazdi, and M. Weyrich, "Reconfiguration management in manufacturing: A systematic literature review," *Automatisierungstechnik*, vol. 55, no. 10, pp. 3214–3219, 2023.

[31] Y. Qamsane, C.-Y. Chen, E. C. Balta, B.-C. Kao, S. Mohan, J. Moyne, D. Tilbury, and K. Barton, "A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems," in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Piscataway, NJ, USA, Aug. 2019, pp. 1394–1401.

[32] Y. Qamsane, J. Moyne, M. Toothman, I. Kovalenko, E. C. Balta, J. Faris, D. M. Tilbury, and K. Barton, "A methodology to develop and implement digital twin solutions for manufacturing systems," *IEEE Access*, vol. 9, pp. 44247–44265, 2021.

[33] L. Poudel, I. Kovalenko, R. Geng, M. Takaharu, Y. Nonaka, N. Takahiro, U. Shota, D. M. Tilbury, and K. Barton, "An integrated framework for dynamic manufacturing planning to obtain new line configurations," in *Proc. IEEE 18th Int. Conf. Autom. Sci. Eng. (CASE)*, Piscataway, NJ, USA, Aug. 2022, pp. 328–334.

[34] J. K. Touckia, N. Hamani, and L. Kermad, "Digital twin framework for reconfigurable manufacturing systems (RMSs): Design and simulation," *Int. J. Adv. Manuf. Technol.*, vol. 120, nos. 7–8, pp. 5431–5450, Jun. 2022.

[35] T. Müller, S. Kamm, A. Löcklin, D. White, M. Mellinger, N. Jazdi, and M. Weyrich, "Architecture and knowledge modelling for self-organized reconfiguration management of cyber-physical production systems," *Int. J. Comput. Integr. Manuf.*, pp. 1–22, Sep. 2022.

[36] T. Müller, S. Walth, N. Jazdi, and M. Weyrich, "Identification of reconfiguration demand and generation of alternative configurations for cyber-physical production systems," in *Advances in Automotive Production Technology—Theory and Application*, P. Weißgraeber, F. Heieck, and C. Ackermann, Eds. Berlin, Germany: Springer-Vieweg, 2021, pp. 63–70.

[37] F. Mo, H. U. Rehman, F. M. Monetti, J. C. Chaplin, D. Sanderson, A. Popov, A. Maffei, and S. Ratchev, "A framework for manufacturing system reconfiguration and optimisation utilising digital twins and modular artificial intelligence," *Robot. Comput.-Integr. Manuf.*, vol. 82, Aug. 2023, Art. no. 102524.

[38] B. Maschler, T. Müller, A. Löcklin, and M. Weyrich, "Transfer learning as an enhancement for reconfiguration management of cyber-physical production systems," *Proc. CIRP*, vol. 112, pp. 220–225, Jan. 2022.

[39] T. Müller, B. Lindemann, T. Jung, N. Jazdi, and M. Weyrich, "Enhancing an intelligent digital twin with a self-organized reconfiguration management based on adaptive process models," *Proc. CIRP*, vol. 104, pp. 786–791, Jan. 2021.

[40] J. Leng, D. Wang, W. Shen, X. Li, Q. Liu, and X. Chen, "Digital twins-based smart manufacturing system design in industry 4.0: A review," *J. Manuf. Syst.*, vol. 60, pp. 119–137, Jul. 2021.

[41] M. W. Grieves, "Product lifecycle management: The new paradigm for enterprises," *Int. J. Product Develop.*, vol. 2, nos. 1–2, p. 71, 2005.

[42] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP Ann.*, vol. 48, no. 2, pp. 527–540, 1999.

[43] B. Caesar, A. Valdezate, J. Ladiges, R. Capilla, and A. Fay, "A process for identifying and modeling relevant system context for the reconfiguration of automated systems," *IEEE Trans. Autom. Sci. Eng.*, early access, Jul. 7, 2023, doi: 10.1109/TASE.2023.3291394.

[44] S. Wolf, B. Böhm, B. Caesar, and A. Fay, "Creating domain models to describe capabilities of manufacturing plants for the order-controlled production," in *Automation 2020*. Düsseldorf, Germany: VDI Wissensforum, 2020, pp. 531–548.

[45] S. Ben Cheikh, S. Hajri-Gabouj, and S. Darmoul, "Reconfiguring manufacturing systems using an analytic hierarchy process with strategic and operational indicators," in *Proc. Int. Conf. Ind. Eng. Oper. Manage. (IEOM)*, Piscataway, NJ, USA, Mar. 2015, pp. 1–6.

[46] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," Softw. Eng. Insitute, Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep CMU/SEI-90-TR-021, 1990.

[47] I. Schaefer, R. Rabiser, D. Clarke, L. Bettini, D. Benavides, G. Botterweck, A. Pathak, S. Trujillo, and K. Villela, "Software diversity: State of the art and perspectives," *Int. J. Softw. Tools Technol. Transf.*, vol. 14, no. 5, pp. 477–495, Oct. 2012.

[48] C. Seidl, I. Schaefer, and U. Aßmann, "Deltaecore—A model-based delta language generation framework," in *Modellierung 2014*, H.-G. Fill, Ed. Bonn, Germany: Köllen, 2014, pp. 81–96.

[49] J. Mauro, "The reconfiguration engine HyVarRec," Concept Reply, Rari, Italy, Tech. Rep., Apr. 2016.

[50] J. Mauro, M. Nieke, C. Seidl, and I. C. Yu, "Context-aware reconfiguration in evolving software product lines," *Sci. Comput. Program.*, vol. 163, pp. 139–159, Oct. 2018.

[51] A. Chehouri, R. Younes, J. Perron, and A. Ilinca, "A constraint-handling technique for genetic algorithms using a violation factor," *J. Comput. Sci.*, vol. 12, no. 7, pp. 350–362, Jul. 2016.

[52] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.

[53] F.-A. Fortin, F.-M. De Rainville, M.-A. G. Gardner, M. Parizeau, and C. Gagné, "DEAP: Evolutionary algorithms made easy," *J. Mach. Lang. Res.*, vol. 13, pp. 2171–2175, Jul. 2012.

[54] Y. Xiao and M. Watson, "Guidance on conducting a systematic literature review," *J. Planning Educ. Res.*, vol. 39, no. 1, pp. 93–112, Mar. 2019.

[55] M. J. Page et al., "PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews," *BMJ*, vol. 372, p. n160, Mar. 2021.

[56] N. Sahlab, H. Kahoul, N. Jazdi, and M. Weyrich, "A knowledge graph-based method for automating systematic literature reviews," *Proc. Comput. Sci.*, vol. 207, pp. 2814–2822, Jan. 2022.

[57] B. Caesar, N. Jansen, M. Weigand, M. Ramonat, C. S. Gundlach, A. Fay, and B. Rumpe, "Extracting functional machine knowledge from STEP files for digital twins," in *Proc. IEEE 27th Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2022, pp. 1–4.

[58] H. Zhu, M. Chen, Z. Zhang, and D. Tang, "An adaptive real-time scheduling method for flexible job shop scheduling problem with combined processing constraint," *IEEE Access*, vol. 7, pp. 125113–125121, 2019.

[59] I. S. Khan, U. Ghafoor, and T. Zahid, "Meta-heuristic approach for the development of alternative process plans in a reconfigurable production environment," *IEEE Access*, vol. 9, pp. 113508–113520, 2021.

[60] E. Asghar, U. K. U. Zaman, A. A. Baqai, and L. Homri, "Optimum machine capabilities for reconfigurable manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 95, nos. 9–12, pp. 4397–4417, Apr. 2018.

[61] H. H. Benderbal and L. Benyoucef, "Machine layout design problem under product family evolution in reconfigurable manufacturing environment: A two-phase-based AMOSA approach," *Int. J. Adv. Manuf. Technol.*, vol. 104, nos. 1–4, pp. 375–389, Sep. 2019.

[62] R. C. Sabioni, J. Daaboul, and J. L. Duigou, "An integrated approach to optimize the configuration of mass-customized products and reconfigurable manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 1–2, pp. 141–163, Jul. 2021.

[63] M. A. Yazdani, A. Khezri, and L. Benyoucef, "Process and production planning for sustainable reconfigurable manufacturing systems (SRMSs): Multi-objective exact and heuristic-based approaches," *Int. J. Adv. Manuf. Technol.*, vol. 119, nos. 7–8, pp. 4519–4540, Apr. 2022.

[64] Z. Moukkar, A. Cauvin, and A. Ferrarini, "Design approach for assembly lines reconfiguration process," in *Artificial Intelligence and Industrial Applications* (Lecture Notes in Networks and Systems), vol. 144, T. Masrour, I. E. Hassani, and A. Cherrafi, Eds. Cham, Switzerland: Springer, 2021, pp. 307–322.

[65] M. Uzunosmanoglu, B. Raa, V. Limère, A. de Cock, Y. Singh, A. J. Lopez, S. Gautama, and J. Cottyn, "Aggregate planning for multi-product assembly lines with reconfigurable cells," in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems* (IFIP Advances in Information and Communication Technology), vol. 631, A. Dolgui, A. Bernard, D. Lemoine, G. von Cieminski, and D. Romero, Eds. Cham, Switzerland: Springer, 2021, pp. 525–534.

[66] A. Maldonado-Ramirez, I. Lopez-Juarez, and R. Rios-Cabrera, "Reconfigurable distributed controller for welding and assembly robotic systems: Issues and experiments," in *Transactions on Intelligent Welding Manufacturing*, S. Chen, Y. Zhang, and Z. Feng, Eds. Singapore: Springer, 2019, pp. 29–49.

[67] S. Sierla, V. Kyrki, P. Aarnio, and V. Vyatkin, "Automatic assembly planning based on digital product descriptions," *Comput. Ind.*, vol. 97, pp. 34–46, May 2018.

[68] C. He, Z. Guan, Y. Gong, C. Wang, and L. Yue, "Automated flexible transfer line design problem: Sequential and reconfigurable stages with parallel machining cells," *J. Manuf. Syst.*, vol. 52, pp. 157–171, Jul. 2019.

[69] J. C. Serrano-Ruiz, J. Mula, and R. Poler, "Development of a multidimensional conceptual model for job shop smart manufacturing scheduling from the industry 4.0 perspective," *J. Manuf. Syst.*, vol. 63, pp. 185–202, Apr. 2022.

[70] P. Li and P. Jiang, "Enhanced agents in shared factory: Enabling high-efficiency self-organization and sustainability of the shared manufacturing resources," *J. Cleaner Prod.*, vol. 292, Apr. 2021, Art. no. 126020.

[71] G. Hoellthaler, M. Schreiber, K. Vernickel, J. B. Isa, J. Fischer, N. Weinert, R. Rosen, and S. Braunreuther, "Reconfiguration of production systems using optimization and material flow simulation," *Proc. CIRP*, vol. 81, pp. 133–138, Jan. 2019.

[72] B. Kaiser, A. Reichle, and A. Verl, "Model-based automatic generation of digital twin models for the simulation of reconfigurable manufacturing systems for timber construction," *Proc. CIRP*, vol. 107, pp. 387–392, Jan. 2022.

[73] B. Caesar, M. Nieke, A. Köcher, C. Hildebrandt, C. Seidl, A. Fay, and I. Schaefer, "Context-sensitive reconfiguration of collaborative manufacturing systems," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 307–312, 2019.

[74] I. B. Garcia, J. Daaboul, A. Jouglet, and J. L. Duigou, "An approach to jointly optimize the process plan, scheduling, and layout design in reconfigurable manufacturing systems," in *Service Oriented, Holonic and Multi-agent Manufacturing Systems for Industry of the Future*, vol. 1034, T. Borangiu, D. Trentesaux, P. Leitão, O. Cardin, and L. Joblot, Eds. Cham, Switzerland: Springer, 2022, pp. 403–415.

[75] B. Dafflon, N. Moalla, and Y. Ouzrout, "Cyber-physical systems network to support decision making for self-adaptive production system," in *Proc. 12th Int. Conf. Softw., Knowl., Inf. Manage. Appl. (SKIMA)*, Dec. 2018, pp. 1–8.

[76] C. Mejía-Moncayo, J.-P. Kenné, and L. A. Hof, "A hybrid architecture for a reconfigurable cellular remanufacturing system," in *IFIP Int. Conf. Adv. Prod. Manage. Syst* Cham, Switzerland: Springer, 2021, pp. 488–496.

[77] D. Guo, M. Li, S. Ling, R. Y. Zhong, Y. Rong, and G. Q. Huang, "Synchronization-oriented reconfiguration of FPAI under graduation intelligent manufacturing system in the COVID-19 pandemic and beyond," *J. Manuf. Syst.*, vol. 60, pp. 893–902, Jul. 2021.

[78] A. B. D. Eynaud, N. Klement, L. Roucoules, O. Gibaru, and L. Durville, "Framework for the design and evaluation of a reconfigurable production system based on movable robot integration," *Int. J. Adv. Manuf. Technol.*, vol. 118, nos. 7–8, pp. 2373–2389, Feb. 2022.

[79] S. Pellegrinelli, "Configuration and reconfiguration of robotic systems for waste macro sorting," *Int. J. Adv. Manuf. Technol.*, vol. 102, nos. 9–12, pp. 3677–3687, Jun. 2019.

[80] Y.-J. Lin, Z.-X. Chen, and C.-Y. Huang, "Knowledge reasoning for intelligent manufacturing control system," *Proc. Manuf.*, vol. 39, pp. 1880–1888, Jan. 2019.

[81] G. Kumar, K. K. Goyal, N. K. Batra, and D. Rani, "Single part reconfigurable flow line design using fuzzy best worst method," *Opsearch*, vol. 59, no. 2, pp. 603–631, Jun. 2022.

[82] C. Yang, F. Liao, S. Lan, L. Wang, W. Shen, and G. Q. Huang, "Flexible resource scheduling for software-defined cloud manufacturing with edge computing," *Engineering*, vol. 22, pp. 60–70, Mar. 2023.

[83] B. I. Epureanu, X. Li, A. Nassehi, and Y. Koren, "Self-repair of smart manufacturing systems by deep reinforcement learning," *CIRP Ann.*, vol. 69, no. 1, pp. 421–424, 2020.

[84] H. H. Benderbal, M. Dahane, and L. Benyoucef, "Modularity assessment in reconfigurable manufacturing system (RMS) design: An archived multi-objective simulated annealing-based approach," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 1–4, pp. 729–749, Jan. 2018.

[85] M. Engelsberger and T. Greiner, "Dynamic reconfiguration of service-oriented resources in cyber–physical production systems by a process-independent approach with multiple criteria and multiple resource management operations," *Future Gener. Comput. Syst.*, vol. 88, pp. 424–441, Nov. 2018.

[86] M. Li and G. Q. Huang, "Production-intralogistics synchronization of industry 4.0 flexible assembly lines under graduation intelligent manufacturing system," *Int. J. Prod. Econ.*, vol. 241, Nov. 2021, Art. no. 108272.

[87] S. Ding, H. Zhang, W. Wu, E. Guo, Y. Zhang, and A. Song, "A digital and structure-adaptive geometric error definition and modeling method of reconfigurable machine tool," *Int. J. Adv. Manuf. Technol.*, vol. 112, nos. 7–8, pp. 2359–2371, Feb. 2021.

[88] M. Bi, I. Kovalenko, D. M. Tilbury, and K. Barton, "Dynamic resource allocation using multi-agent control for manufacturing systems," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 488–494, 2021.

[89] K. Balzereit and O. Niggemann, "Automated reconfiguration of cyber-physical production systems using satisfiability modulo theories," in *Proc. IEEE Conf. Ind. Cyberphys. Syst. (ICPS)*, vol. 1, Jun. 2020, pp. 461–468.

[90] J. Wan, S. Tang, D. Li, M. Imran, C. Zhang, C. Liu, and Z. Pang, "Reconfigurable smart factory for drug packing in healthcare industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 15, no. 1, pp. 507–516, Jan. 2019.

[91] A. Bunte, P. Wunderlich, N. Moriz, P. Li, A. Mankowski, A. Rogalla, and O. Niggemann, "Why symbolic AI is a key technology for self-adaption in the context of CPPS," in *Proc. 24th IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2019, pp. 1701–1704.

[92] K. Balzereit and O. Niggemann, "AutoConf: New algorithm for reconfiguration of cyber-physical production systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 739–749, Jan. 2023.

[93] M. Ashraf and F. Hasan, "Configuration selection for a reconfigurable manufacturing flow line involving part production with operation constraints," *Int. J. Adv. Manuf. Technol.*, vol. 98, nos. 5–8, pp. 2137–2156, Sep. 2018.

[94] M. Bortolini, F. G. Galizia, C. Mora, and F. Pilati, "Reconfigurability in cellular manufacturing systems: A design model and multi-scenario analysis," *Int. J. Adv. Manuf. Technol.*, vol. 104, nos. 9–12, pp. 4387–4397, Oct. 2019.

[95] A. S. Khan, L. Homri, J. Y. Dantan, and A. Siadat, "Modularity-based quality assessment of a disruptive reconfigurable manufacturing system—A hybrid meta-heuristic approach," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 5–6, pp. 1421–1444, Jul. 2021.

[96] M. Ameer and M. Dahane, "Reconfigurability improvement in industry 4.0: A hybrid genetic algorithm-based heuristic approach for a co-generation of setup and process plans in a reconfigurable environment," *J. Intell. Manuf.*, vol. 34, pp. 1445–1467, Nov. 2021.

[97] M. Manzini and M. Urgo, "An approximate approach for the verification of process plans with an application to reconfigurable pallets," in *Selected Topics in Manufacturing*. Cham, Switzerland: Springer, 2021, pp. 105–120.

[98] O. Battaïa, A. Dolgui, N. Guschinky, and F. Makssoud, "Mathematical model for processing multiple parts on multi-positional reconfigurable machines with turrets," in *Proc. IFIP Int. Conf. Adv. Prod. Manage. Syst.*, 2021, pp. 563–573.

[99] A. Kumar, L. N. Pattanaik, and R. Agrawal, "Multi-objective scheduling model for reconfigurable assembly systems," in *Innovations in Soft Computing and Information Technology*. Singapore: Springer, 2019, pp. 209–217.

[100] N. Rodrigues, E. Oliveira, and P. Leitão, "Decentralized and on-the-fly agent-based service reconfiguration in manufacturing systems," *Comput. Ind.*, vol. 101, pp. 81–90, Oct. 2018.

[101] A. Ma, M. Frantzén, C. Snider, and A. Nassehi, "Anarchic manufacturing: Distributed control for product transition," *J. Manuf. Syst.*, vol. 56, pp. 1–10, Jul. 2020.

[102] N. Nikolakis, R. Senington, K. Sipsas, A. Syberfeldt, and S. Makris, "On a containerized approach for the dynamic planning and control of a cyber–physical production system," *Robot. Comput.-Integr. Manuf.*, vol. 64, Aug. 2020, Art. no. 101919.

[103] M. Mahmoodjanloo, R. Tavakkoli-Moghaddam, A. Baboli, and A. Bozorgi-Amiri, "Flexible job shop scheduling problem with reconfigurable machine tools: An improved differential evolution algorithm," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106416.

[104] J. Hermann, P. Rübel, M. Birtel, F. Mohr, A. Wagner, and M. Ruskowski, "Self-description of cyber-physical production modules for a product-driven manufacturing system," *Proc. Manuf.*, vol. 38, pp. 291–298, Jan. 2019.

[105] A. Villalonga, E. Negri, L. Fumagalli, M. Macchi, F. Castaño, and R. Haber, "Local decision making based on distributed digital twin framework," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 10568–10573, 2020.

[106] K. A. Kurniadi, S. Lee, and K. Ryu, "Digital twin approach for solving reconfiguration planning problems in RMS," in *Proc. IFIP Int. Conf. Adv. Prod. Manag. Syst.* Cham, Switzerland: Springer, 2018, pp. 327–334.

[107] R. Santos, C. Toscano, and J. P. de Sousa, "A new simulation-based approach in the design of manufacturing systems and real-time decision making," *IFAC-PapersOnLine*, vol. 54, no. 1, pp. 282–287, 2021.

[108] M. Bortolini, R. Accorsi, M. Faccio, F. G. Galizia, and F. Pilati, "Toward a real-time reconfiguration of self-adaptive smart assembly systems," *Proc. Manuf.*, vol. 39, pp. 90–97, Jan. 2019.

[109] C. El Zant, K. Benfriha, S. Loubère, A. Aoussat, and O. Adjoul, "A design methodology for modular processes orchestration," *CIRP J. Manuf. Sci. Technol.*, vol. 35, pp. 106–117, Nov. 2021.

[110] D. Zhang, M. Xie, H. Yan, and Q. Liu, "Resilience dynamics modeling and control for a reconfigurable electronic assembly line under spatio-temporal disruptions," *J. Manuf. Syst.*, vol. 60, pp. 852–863, Jul. 2021.

[111] O. Antons and J. C. Arlinghaus, "Data-driven and autonomous manufacturing control in cyber-physical production systems," *Comput. Ind.*, vol. 141, Oct. 2022, Art. no. 103711.

[112] M. Ashraf, A. Gola, F. Hasan, and A. S. AlArjani, "An adaptive algorithm for multiple part families manufacturing selection in reconfigurable flow lines," in *Intelligent Information and Database Systems*, vol. 1178, P. Sitek, M. Pietranik, M. Krótkiewicz, and C. Srinilta, Eds. Singapore: Springer, 2020, pp. 133–144.

[113] C. Abadi, I. Manssouri, and A. Abadi, "Reconfiguration of flexible manufacturing systems considering product morpho-dimensional characteristics and modular design," in *Advances in Integrated Design and Production*, A. Saka, J.-Y. Choley, J. Louati, Z. Chalh, M. Barkallah, M. Alfidi, M. Amar, F. Chaari, and M. Haddar, Eds. Cham, Switzerland: Springer, 2021, pp. 559–565.

[114] S. Tang, Y. Zhang, Z. Zheng, T. Chen, H. Zheng, and J. Zhang, "Critical path control of dynamic reconfigurable manufacturing systems based on NCESs," in *Proc. China Autom. Congr. (CAC)*, Oct. 2021, pp. 3474–3479.

[115] I. Khettabi, L. Benyoucef, and M. A. Boutiche, "Sustainable reconfigurable manufacturing system design using adapted multi-objective evolutionary-based approaches," *Int. J. Adv. Manuf. Technol.*, vol. 115, nos. 11–12, pp. 3741–3759, Aug. 2021.

[116] A. Abid, M. Hammadi, M. Barkallah, J.-Y. Choley, J. Louati, A. Rivière, and M. Haddar, "Generic framework for holonic modelling and multi-agent based verification of reconfigurable manufacturing systems," *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 12, pp. 1793–1809, Dec. 2018.

[117] E. Massimi, A. Khezri, H. H. Benderbal, and L. Benyoucef, "A heuristic-based non-linear mixed integer approach for optimizing modularity and integrability in a sustainable reconfigurable manufacturing environment," *Int. J. Adv. Manuf. Technol.*, vol. 108, nos. 7–8, pp. 1997–2020, Jun. 2020.

[118] A. Kumar, L. N. Pattanaik, and R. Agrawal, "Optimal sequence planning for multi-model reconfigurable assembly systems," *Int. J. Adv. Manuf. Technol.*, vol. 100, nos. 5–8, pp. 1719–1730, Feb. 2019.

[119] Z. Guo, Y. Zhang, X. Zhao, and X. Song, "CPS-based self-adaptive collaborative control for smart production-logistics systems," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 188–198, Jan. 2021.

[120] X. L. Hoang, A. Fay, P. Marks, and M. Weyrich, "Industrial application of a MDM-based approach for generation and impact analysis of adaptation options—A case study," in *Proc. IEEE 23rd Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2018, pp. 1244–1247.

BIRTE CAESAR received the integrated B.Sc. and M.Sc. degrees in industrial engineering from the Inter-University Program HWI, University of Hamburg, the Hamburg University of Applied Sciences, the Technical University of Hamburg, and Helmut Schmidt University, Hamburg, Germany, in 2013 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Institute of Automation Technology, Helmut Schmidt University. In 2021, she spent six months as a Fulbright Fellow with the University of Michigan, Ann Arbor, MI, USA, where she investigated the suitability of digital twins for reconfiguration management. She is a Research Group Leader for digital twins with the Institute of Automation Technology, Helmut Schmidt University. Her research interests include reconfiguration management in the manufacturing domain as well as engineering support to provide engineers with systematic methods to create models needed for reconfiguration management during operation.

KIRA BARTON (Senior Member, IEEE) received the B.S. degree in mechanical engineering from the University of Colorado, in 2001, and the M.S. and Ph.D. degrees in mechanical engineering from the University of Illinois at Urbana–Champaign, in 2006 and 2010, respectively. She joined the Mechanical Engineering Department, University of Michigan, Ann Arbor, MI, USA, in 2011. She is currently a Professor with the Robotics Department and the Mechanical Engineering Department. She is also the Associate Director of the Automotive Research Center, a University-based U.S. Army Center of Excellence for modeling and simulation of military and civilian ground systems. Her research interests include advancements in modeling, sensing, and control for applications in smart manufacturing and robotics. She was a recipient of an NSF CAREER Award, in 2014, the SME Outstanding Young Manufacturing Engineer Award, in 2015, the University of Illinois, Department of Mechanical Science and Engineering Outstanding Young Alumni Award, in 2015, the University of Michigan, Department of Mechanical Engineering Department Achievement Award, in 2016, and the ASME Dynamic Systems and Control Young Investigator Award, in 2017. She was named one of the 25 leaders transforming manufacturing by SMEs, in 2022, and was selected as one of the 2022 winners of the Manufacturing Leadership Award from the Manufacturing Leadership Council.

DAWN M. TILBURY (Fellow, IEEE) received the B.S. degree in electrical engineering from the University of Minnesota, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA. From 2017 to 2021, she was the Assistant Director of Engineering with the National Science Foundation, where she oversaw a federal budget of nearly U.S. $1 billion annually while maintaining her position with the University of Michigan. She is currently the inaugural Ronald D. and Regina C. McNeil Department Chair of Robotics with the University of Michigan and the Herrick Professor of Engineering. She has published more than 200 articles in refereed journals and conference proceedings. Her research interests include control systems, including applications to robotics and manufacturing systems. She is a fellow of ASME and a Life Member of SWE.

ALEXANDER FAY (Senior Member, IEEE) was born in 1970. He received the Diploma and Ph.D. degrees in electrical engineering from the Technical University of Braunschweig, Germany, in 1995 and 1999, respectively. He had worked for five years with ABB Corporate Research Laboratories in Heidelberg and Ladenburg before he was appointed as a Full Professor with the Institute of Automation Technology, Helmut Schmidt University, Hamburg, Germany, in 2004. His main research interests include models and methods for the engineering of large automated systems, especially in the process and manufacturing industries and in buildings and transportation systems, with a special focus on autonomous and decentralized systems, model-based engineering, and semantic technologies. He has served as a member for the AdCom of the IEEE Industrial Electronics Society. He was the General Co-Chair of the IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), in 2008, and the Program Co-Chair of the IEEE International Conference on Automation Science and Engineering (CASE), in 2018. From 2011 to 2017, he served as an Associate Editor for the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS.

• • •