



Article

# Executable Digital Process Twins: Towards the Enhancement of Process-Driven Systems

Flavio Corradini <sup>1\*</sup>, Sara Pettinari <sup>1,\*</sup>, Barbara Re <sup>1</sup>, Lorenzo Rossi <sup>1,\*</sup> and Francesco Tiezzi <sup>2</sup>

<sup>1</sup> School of Science and Technology, University of Camerino, 62032 Camerino, Italy; flavio.corradini@unicam.it (F.C.); barbara.re@unicam.it (B.R.)

<sup>2</sup> Dipartimento di Statistica, Informatica, Applicazioni, University of Florence, 50121 Florence, Italy; francesco.tiezzi@unifi.it

\* Correspondence: sara.pettinari@unicam.it (S.P.); lorenzo.rossi@unicam.it (L.R.)

**Abstract:** The development of process-driven systems and the advancements in digital twins have led to the birth of new ways of monitoring and analyzing systems, i.e., digital process twins. Specifically, a digital process twin can allow the monitoring of system behavior and the analysis of the execution status to improve the whole system. However, the concept of the digital process twin is still theoretical, and process-driven systems cannot really benefit from them. In this regard, this work discusses how to effectively exploit a digital process twin and proposes an implementation that combines the monitoring, refinement, and enactment of system behavior. We demonstrated the proposed solution in a multi-robot scenario.

**Keywords:** digital process twin; process-driven system; BPMN; IoT; multi-robot system



**Citation:** Corradini, F.; Pettinari, S.; Re, B.; Rossi, L.; Tiezzi, F. Executable Digital Process Twins: Towards the Enhancement of Process-Driven Systems. *Big Data Cogn. Comput.* **2023**, *7*, 139. <https://doi.org/10.3390/bdcc7030139>

Academic Editor: Nik Bessis

Received: 30 June 2023

Revised: 25 July 2023

Accepted: 31 July 2023

Published: 8 August 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Process-driven systems have emerged for organizing and managing complex workflows within various domains. These systems are based on the concept of modeling a system's processes to enable the optimization of the operations and better control over the activities [1]. At the heart of process-driven systems is business process management [2], which involves the identification, analysis, and improvement of business processes to enhance overall performance. Specifically, business process models can be exploited to express the system workflow, communication, tasks, and data required to achieve the goal in a unique model [3].

In this domain, the advancements of digital twins (DTs), i.e., virtual replicas of a system acting in a real environment, are opening up new possibilities for advanced system monitoring, analysis, and optimization. Indeed, a DT gathers flows of data collected from the environment and provides a reliable replica of a system. It can be used to represent, simulate, and predict changes in the physical system to foster decision making and provide the opportunity to apply preventive actions for optimizing system behavior [4].

With the creation of DTs for process-driven systems, also referred to as digital process twins (DPTs), organizations gain valuable insights into their operations, enabling them to make decisions, optimize their performance, and predict possible outcomes based on the data coming from both the process model and the physical system [5]. Therefore, a DPT can both permit developers and end-users to interpret the behavior of a system and facilitate the interpretation and analysis of data. At the same time, a DPT can allow the exploration of various designs and operational alternatives related to the behavior of the whole system [6].

The application of DTs is spreading in a variety of fields, especially in the domains of IoT and robotic systems [7]. The convergence of DTs with model-driven techniques, which leverage business processes to define the behavior of IoT and robotic

systems [8–10], makes them a suitable target for the application of DPTs. Indeed, considering the execution perspective, IoT and robotic devices function in real-world scenarios that may involve several errors not considered during system modeling. In such scenarios, the DPT can be exploited to enhance the whole system by providing the developers with a reliable representation of the system devices' execution.

Nevertheless, although the application of DPTs is promising, current solutions mainly focus on conceptual aspects and lack effective ways of virtually representing a system and its operational process [11].

This paper aimed to fill the gap between the concept of DPTs and their practical implementation in physical systems. The aim was to enable developers to effectively utilize the functionalities of a DPT to enhance physical systems. In line with this objective, we conducted an investigation into existing DT concepts, exploring their potentialities and limitations. This led to the proposal of the novel concept of an executable DPT (xDPT). An xDPT enriches the monitoring and analysis functionalities of a DPT with the possibility of actively driving the execution of the entire system. Our solution comprises a methodology guiding the application and construction of xDPTs. Additionally, it provides developers with a comprehensive toolset to not only monitor and analyze a system's behavior but also drive its operations. As a result, our contribution opens up new possibilities for enhanced process-driven system management and optimization.

We obtained concrete evidence of the benefits brought by our solution through a demonstrative multi-robot scenario. The solution relied on (i) BPMN collaboration diagrams [12] for representing the system; (ii) the FAME framework [13] for refining and enacting the process model; and (iii) a business model animator and a 3D simulator for monitoring the system's execution to show the xDPT in a unique interface.

This work was based on the approach presented in [5] and extended its scope as follows:

1. We present and discuss the concept of xDPT and how it can be used to actually execute a process-driven system.
2. We illustrate the concepts and technologies supporting the application of an xDPT to IoT and robotic systems.
3. We present the implementation and application of an xDPT in a multi-robot scenario.

The rest of the paper is structured as follows: Section 2 discusses different definitions of the DT and DPT concepts to identify current limitations and present the novel concept of an xDPT. Section 3 presents the xDPT in an IoT and robotic context. Section 4 implements the xDPT in a multi-robot scenario running in a warehouse. Section 5 discusses approaches in the literature for exploiting DTs in IoT and robotic systems. Finally, Section 6 concludes the paper and explores directions for future works.

## 2. Towards Executable Digital Process Twins

To better understand how the concept of a DPT can be integrated with process-driven systems, this section introduces the emerging trends for the application of the DT concept in different contexts and scenarios. We first introduce the application of DTs to distributed and interactive organizations, leading to the emerging concept of digital process twins. Then, we present the application of DTs for directly implementing industrial devices, referred to as executable digital twins. Finally, we propose a concept that combines these two DT types, introducing the executable digital process twin, which aims to enhance process-driven systems.

### 2.1. Digital Process Twin

The concept of the DT is evolving to include system information models that describe the properties and relationships of all data and information required to achieve a task, as well as the elements and connections of organizational systems, to understand the factors affecting a decision and to predict decision outcomes [14]. In the literature, this concept is referred to as a DPT [15]. Notably, DPTs are considered part of a wider concept, known

as the digital twins of an organization [14]. DPTs still generate many concerns among organizations trying to figure out the benefits they bring to the whole system and how to overcome challenges such as implementing real-time changes and developing the DPT to reflect these changes [16]. To overcome these concerns, Becker et al. [17] prescribed a set of criteria to provide clarification and suggestions regarding emerging DPTs. Specifically, a DPT should involve:

- Multiple and interdependent processes;
- Agents capable of making choices, learning from experience, and achieving goals;
- Models showing different aspects of the organization;
- Simulation and prediction capabilities to understand how the organization might behave when it faces different situations.

Indeed, a DPT can be seen as a digital replica of the business processes of an organization. Therefore, it enables runtime monitoring, analysis, simulation, and improvements, providing an extension to the current state of the art in process modeling and mining [18]. Moreover, DPTs are more effective when applied to organizational contexts described by an enterprise architecture modeling language, such as BPMN [12]. Riss et al. [19] evaluate the characteristics and benefits of using a DPT in these contexts. By exploiting a continuous data flow, the DPT shows the real-time performance of its actual counterpart and associates the process with the current behavior of the entities. Additionally, the collection of data on past behavior enables simulations and predictions of future process behavior under changed conditions. Notably, when the DPT is applied to a process-driven system, we refer to it as DPT.

### 2.2. Executable Digital Twin

Traditional DTs serve as simulation and predictive tools to enable engineers and developers to optimize performance, conduct simulations, and predict failures without risking damage to the actual physical object. However, DTs remain separate entities, only influencing their real-world counterparts through insights and adjustments [20]. To bring the DT closer to the physical side, Hartmann and Van der Auweraer [21] introduced the concept of an Executable Digital Twin (xDT) for the application of DTs in industrial contexts. Indeed, an xDT aims to bridge the gap between a DT and the corresponding execution on the physical side. It can be seen as a specific DT whose models can be executed by dedicated execution engines both in simulation environments and in physical devices.

The core goal of exploiting xDT is to support the whole lifecycle of a system, from its design to the analysis steps. Therefore, an xDT is a high-fidelity simulation model that expands its applications beyond the design and validation phases to also manage in-operation and service phases [22]. Currently, xDTs are proposed for industrial scenarios as self-contained, executable, digital representations of a specific behavior of a physical device [23], with the aim of testing configurations that can be complex to recreate in real life, validating the system dynamics, and predicting future behaviors [24]. Indeed, the main goal of an xDT is to deploy the digital twin itself into the physical world, enabling the knowledge contained in the digital representation to create value for the physical counterpart directly.

### 2.3. Executable Digital Process Twin

Currently, with the rising adoption of business processes for specifying and driving the behavior of distributed systems [8–10], it becomes increasingly evident that the application of DPTs is essential to enhance the whole system. Indeed, the ability to monitor, analyze, and understand the complex behavior of process-driven systems makes DPTs the key to gaining insight into processes and improving their decision-making skills. Therefore, a DPT of a process-driven system should combine a virtual representation of each participant with the representation of the process itself. This combined representation shows the interdependencies between the participants and the process [25].

Nevertheless, a DPT capable of virtually representing an organizational system and its operational processes is still a theoretical concept and is far away from reality [11]. Therefore, there is a need to understand how it is possible to exploit DPTs, and in particular DPTs, not only for performing monitoring and analysis steps but also for executing the whole system. In this regard, the process should serve both as a DPT, for performing monitoring and analysis steps, and as an xDT, for enacting and controlling the whole system’s execution. We refer to the application of an xDT in a process-driven organizational context as an executable digital process twin (xDPT).

Notably, since the reference organization is built upon a process model, the proposed xDPT is known in advance. Therefore, it does not need to rely on techniques for process discovery from streams of logs, which currently are not mature enough to allow the discovery of processes representing collaborative distributed systems [11,26]. The authors of Figure 1 present an overview of an xDPT representing a process-driven organization. The workflow of this organization is described in BPMN, whose semantics will be detailed in Section 3. Specifically, the xDPT receives real-time event logs from the running organization and allows direct monitoring by marking the status of the system in the running process. Moreover, it can exploit existing process mining techniques to check the conformance of the executed behavior [27], to provide an enriched view over multiple system perspectives [28], and to store historical data to predict process evolution [29]. Finally, the process driving the system behavior can be refined and deployed into the organization. Summing up, Table 1 compares DPT, xDT, and xDPT to show the characteristics and potentiality of applying an xDPT. In particular, the comparison is made with respect to their distinctive features of (i) representing the system behavior (representation), (ii) knowing in advance the model defining the system behavior (prior knowledge), (iii) simulating and predicting future states of the system (prediction), (iv) improving the system at run-time (improvements), and (v) deploying the behavioral model in the physical system (deployment). Notably, features supported by a DT are represented by a check mark, while unsupported features are indicated by a cross symbol.

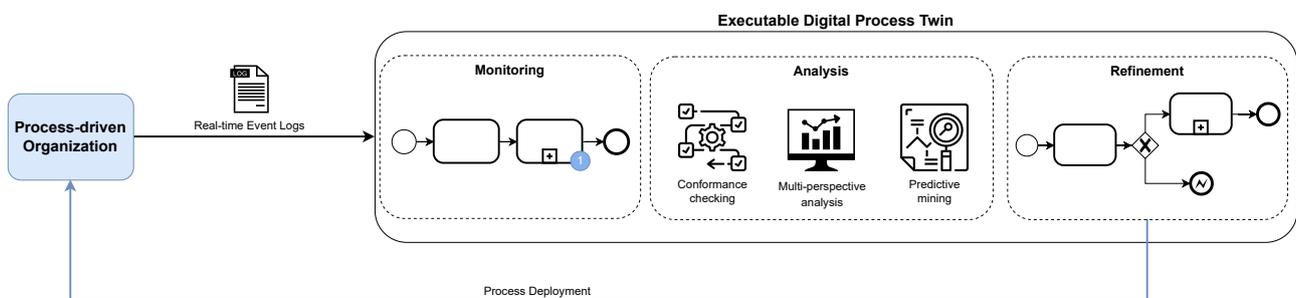


Figure 1. Overview of the functionalities of an executable digital process twin.

Table 1. Comparison of the three types of DTs.

	DPT	xDT	xDPT
Representation	✓	✗	✓
Prior knowledge	✗	✓	✓
Prediction	✓	✓	✓
Improvements	✗	✗	✓
Deployment	✗	✓	✓

### 3. Supporting Executable Digital Process Twins for IoT and Robotic Systems

The development of xDPTs can enhance the application of business processes for specifying the behavior of IoT and robotic systems [8,9]. Therefore, we reduce the principles of Becker et al. [17] closer to this application domain. Indeed, a DTO of a process-driven IoT and robotic system should involve the following:

- Multiple and cooperative processes representing the cooperative behavior of the devices;
- Devices capable of achieving a mission by making autonomous decisions;
- A combination of different visualizations of the system's execution;
- Simulation and prediction capabilities to understand how the devices might behave when different conditions (e.g., space, time, resources) are changed.

At the same time, we aim to bring these concepts closer to the actual system execution. Therefore, this section introduces the methodologies enabling the implementation of an xDPT that combines the representation, monitoring, analysis, enactment, and refinement of IoT and robotic systems' behavior.

### 3.1. Representing an xDPT

The representation of the behavior of IoT and robotic systems can be supported by the adoption of the BPMN standard [12]. Indeed, it is a well-established notation for modeling the behavior of complex systems. It is easy to interpret and is compliant with the implementation of the system [9]. Moreover, BPMN models can be exploited to directly execute the system, thus shifting the focus of the system development from code-centric approaches to model-driven ones [30].

More in detail, BPMN is an OMG standard notation that provides different diagrams for modeling business activities [31]. Among the others, collaboration diagrams are the most suited for representing different perspectives of the systems altogether, such as the control flow, communication, and data manipulation [32,33]. More in detail, a collaboration diagram consists of a collection of pools, which contain processes. A process consists of activities, gateways, and events connected to each other by means of sequence flows, i.e., edges. Activities are represented as rectangles and are tasks to be performed within a process. Gateways, represented as diamonds, are used to control the execution flow of a process by managing parallel activities and choices. Events, represented as circles with possible symbols inside, represent something that can happen at the beginning, during, or at the end of the process execution. Finally, data objects are information that can be input or output with respect to activities and events.

Therefore, BPMN can be exploited to faithfully model the system behavior, thus being suitable for representing an xDPT.

### 3.2. Enacting and Refining an xDPT

The representation of the system behavior using the BPMN standard enables a direct system execution based on the modeled process. Indeed, BPMN models can be enacted to drive the system's execution, thus easing the development of IoT and robotic systems. Specifically, the model enactment can be achieved by exploiting a model-to-code approach or a direct model execution.

In the model-to-code approaches, such as the ones proposed in [34,35], the BPMN model is translated into code that can be injected into the devices. Therefore, the developer can specify the behavior of the devices via BPMN and automatically translate it into executable code. Differently, the approaches that exploit a direct model execution [13,36,37], use a BPMN engine capable of receiving the model, interpreting its elements, and, consequently, driving the behavior of the devices. Notably, the model-to-code and the direct model execution approaches can be complementary, based on the computational capabilities of the devices or on the requirements of the whole system.

Moreover, the concept of DT states that any changes to the DT must be reflected in the actual behavior [38]. This requirement can be achieved by exploiting a direct model refinement and enactment. Indeed, a developer can dynamically change the behavior of the system by refining the existing process model and deploying the updated version into the BPMN-driven system. This approach allows for enhancing the devices' behavior, according to the evolution of the system.

### 3.3. Monitoring an xDPT

The monitoring feature of an xDPT is in charge of showing at runtime the current status of the system. Based on the BPMN model used to represent and drive the system execution, it is possible to highlight in the process the activities that devices are performing. Indeed, concerning the execution semantics, BPMN relies on the notion of token (Section 7.1.1 [12]). A token traverses, from a start event, the sequence edges of the process and passes through its elements, enabling their execution, and an end event consumes it when it terminates. The distribution of tokens in the process elements refers to an execution state; therefore, the process execution is defined in terms of state evolution, as described in [32]. The concept of tokens flowing through a process model is very useful for checking if the modeler's intentions are reflected in how the process is actually performed [39]. Several tools [40–43] implement token simulators to show the evolution of the system and effectively track the interplay between control flow, data handling, and exchange of messages, specifically in complex scenarios [32]. The main goal is to detect undesired executions, where, e.g., a control flow is blocked, and deduce the cause beyond them by possibly checking the values of the involved data [41]. Therefore, the integration of a token simulator enables debugging facilities on the top of a BPMN diagram to help the developer easily understand and debug the underlying system.

Moreover, since we aim to apply the concept of xDPT to systems capable of moving and interacting with the environment, the xDPT should also integrate realistic monitoring of the system. Indeed, to provide a comprehensive visualization of the system (real or simulated) executions, a 3D simulator can be used to visualize how the devices act in the environment and interact with each other.

The main objective of 3D simulators is to provide a low-cost, yet effective test bed of the targeted system [44]. Therefore, thanks to the capability of faithfully representing a complex system, a simulator can enable research activities that would not happen without them, due to limited resources or financial restrictions [45], but can also be exploited to study the impact, in terms of wireless signal overload and security, before its real deployment [46]. In general, a simulator provides developers with a graphical user interface showing how the system is performing in order to help them enhance its whole execution.

Finally, the advantage of combining this visualization with process monitoring is the possibility of concurrently seeing the realistic execution of the system behavior and the visualization of the corresponding states of the involved devices.

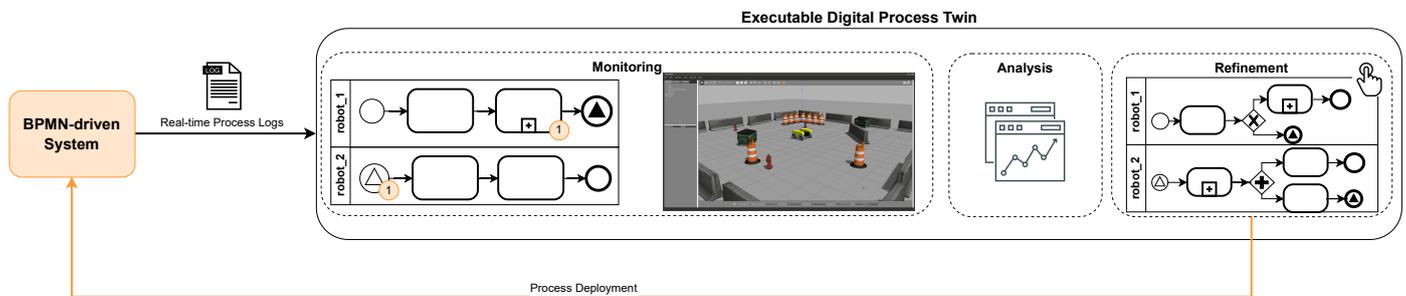
### 3.4. Analyzing an xDPT

The continuous flow of data from the physical system enables an xDPT to show the real-time performance of its counterpart. Therefore, the xDPT can serve for performing real-time analysis, as well as simulations and predictions of future process behavior under changed conditions [17].

The analysis of real-time event logs produced during the execution of the system allows for reducing the delay between the time when the event has happened in the real world and when useful information is obtained [47]. IoT and robotic systems can benefit from the analysis of a stream of data by exploiting streaming process mining techniques. These can enrich the systems with adaptability features by capturing dynamic behaviors and accommodating changes related to the environment [48].

Other approaches exploit event logs simulated or produced during system execution to predict future evolutions of the system [29,49]. Specifically, the combination of event logs produced both by the executed process and by the data perceived by the devices allows the application of process mining techniques to continuously improve processes and respond to changes [50]. Considering the security aspect, IoT and robotic systems have an impact both on the surrounding environment and on other devices. Hence, a detailed analysis of system security could fix a vulnerable device before it can infect others while supporting the predictive detection of attacks and misbehavior [51].

Summing up, Figure 2 shows the features of an xDPT for BPMN-driven IoT and robotic systems. Specifically, the enacted BPMN process can be monitored by utilizing tokens to mark the current status of the system and a 3D simulator to concurrently provide a realistic visualization of the ongoing processes. Process mining techniques enable the analysis of real-time event logs. Finally, the process can be refined to update and enhance the system behavior.



**Figure 2.** Proposed features of an xDPT for IoT and robotic systems.

#### 4. PROWIN: An xDPT for Robotic Systems

To effectively support a developer in the application of an xDPT to an IoT and robotic system, we propose the PROWIN tool, a unique view of the system execution that implements the features of an xDPT. This section first describes the MRS scenario, and then the implementation of the PROWIN tool is described to better explain how to combine different technologies to develop an xDPT applied to the MRS scenario. Notably, the integration of the analysis component in the xDPT is out of the scope of this work.

##### 4.1. Robotic Scenario

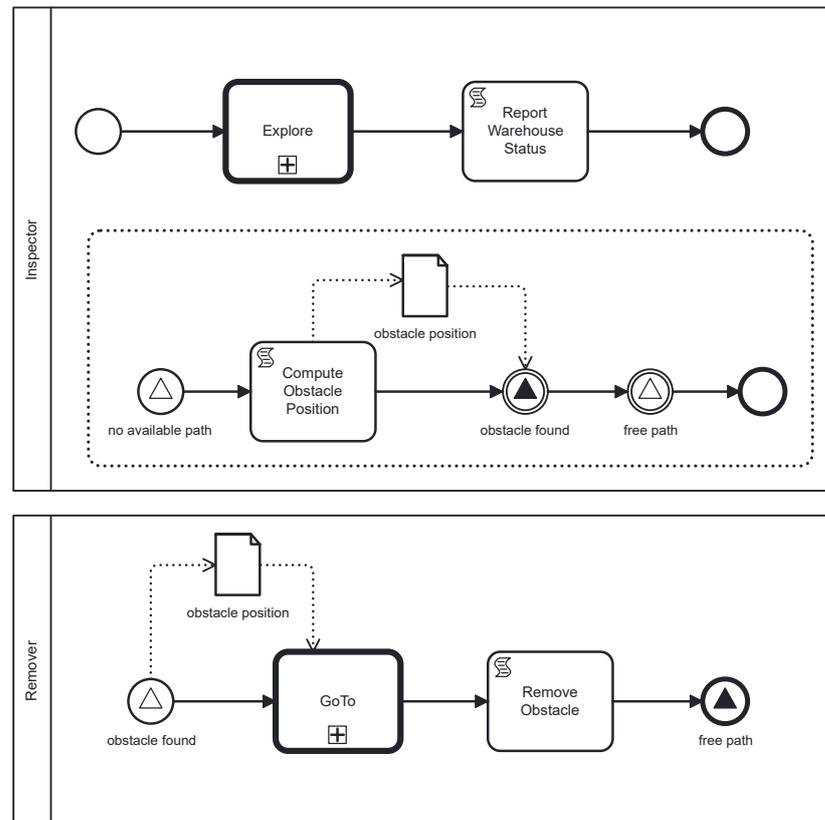
The scenario used to demonstrate the capabilities of the PROWIN tool involves a MRS operating in a warehouse setting. The MRS consists of two ground robots working collaboratively to perform various tasks. One of the robots, referred to as the *Inspector*, has the responsibility of inspecting the warehouse and providing detailed reports on the goods and the overall status of the warehouse. While executing its tasks within the warehouse, the Inspector may encounter obstacles that hinder its movement and disrupt the inspection process.

To overcome the obstacles, the Inspector robot triggers the intervention of another robot known as the *Remover*. The Remover robot is specifically designed and equipped to reach the identified obstacle and effectively remove it from the Inspector's path. Once the Remover robot successfully eliminates the obstacle, it notifies the Inspector robot, allowing it to resume the warehouse inspection. This coordinated effort between the Inspector and Remover robots ensures the smooth progression of the inspection tasks, improving overall efficiency and productivity in the warehouse environment.

##### 4.2. Implementation

To **represent** the xDPT of the scenario, we followed the guidelines presented in [9]. This resulted in the BPMN collaboration diagram depicted in Figure 3, where each robot is represented as a pool. The *Inspector* starts its mission with a call activity that corresponds to the exploration of the place. During this mission, it can find an obstacle blocking its way. This event is handled by an event sub-process in which the robot computes the obstacle position, sends it as a signal event, and, finally, moves to an idle state waiting for a free path. Once the exploration of the whole warehouse is finished, the robot performs the reporting of the warehouse status, by means of a script task, and ends its execution. Meanwhile, the *Remover* robot is triggered by a signal start event when the Inspector sends the position of an obstacle. So, it first reaches the position, represented as a call activity, and then performs

the script task corresponding to the obstacle removal. Its mission ends with the sending of a signal stating a free path.



**Figure 3.** MRS mission in a warehouse scenario.

The **enactment** of the MRS collaboration is driven by the FAME framework [13]. It is a framework for specifying and enacting MRS missions driven by BPMN collaboration diagrams. Specifically, each robot of the collaboration is built upon the ROS framework. Indeed, ROS is one of the most famous and used open-source frameworks for programming robots. It provides an abstraction layer on top of which developers can build robotics applications. Exploiting the FAME framework, it is possible to embed in each robot a BPMN engine capable of interpreting the model and drive the behavior of the robot. This allows an MRS fully driven by the enactment of the BPMN collaboration.

Once the process model is sent to the ROS network, the BPMN collaboration is split based on the robots in the system. Therefore, each robot receives the process model corresponding to the mission it should perform and exploits the built-in BPMN engine to interpret the BPMN elements and consequently perform a mission. During the system execution, the BPMN engines share with the PROWIN tool real-time event logs representing the current element enacted by the engine. Notably, the connection between the ROS network and PROWIN is enabled by the *ros2 web bridge* (<https://github.com/RobotWebTools/ros2-web-bridge>, accessed on 3 February 2023), i.e., a package that exposes the messages exchanged in the ROS network, so that it is possible to access robotic data from a web interface.

To **monitor** the system execution, we make use of the bpmn-token-simulator (<https://github.com/bpmn-io/bpmn-js-token-simulation>, accessed on 30 January 2023) to highlight in the process model the activities performed by the devices. The usage of a token simulator enables a graphical visualization of the system execution status on top of the BPMN process model, giving a precise understanding of its behavior. Indeed, this tool shows the system evolution in terms of token flow that states which activity is currently performed. Moreover,

as presented in Section 3, this monitoring can be enriched with realistic visualization of the physical system. Therefore, the visualization of the running MRS is implemented with the *Gazebo* simulator (<https://gazebo.org>, accessed on 16 December 2022). It is the reference simulator for systems developed in ROS and is commonly used to visualize systems showing how the devices act in the environment. Indeed, it permits testing applications rapidly and in a safe, yet realistic, environment. Moreover, Gazebo is embedded in the ROS architecture as a node; therefore, developers can use the same code for controlling the robots in both real and simulated environments.

Finally, to effectively support the **refinement** of the xDPT, PROWIN provides an additional interface for updating the MRS behavior by exploiting a BPMN modeler capable of directly communicating with the ROS network. Indeed, this component is composed of a web-based BPMN modeler (<https://bpmn.io>, accessed on 10 February 2023) that allows the manipulation of the current process model and is directly connected with the MRS so that the developer can easily change the robotic mission by updating the model and deploying it into the BPMN-driven MRS.

The proposed solution is depicted in Figure 4, showing that the PROWIN tool provides both a unique interface that combines the monitoring of the system evolution in terms of token flow, data values, and message interactions and the visualization of realistic robots acting in the environment and interacting with each other, and a modeling interface for applying changes to the process and thus to the MRS behavior.

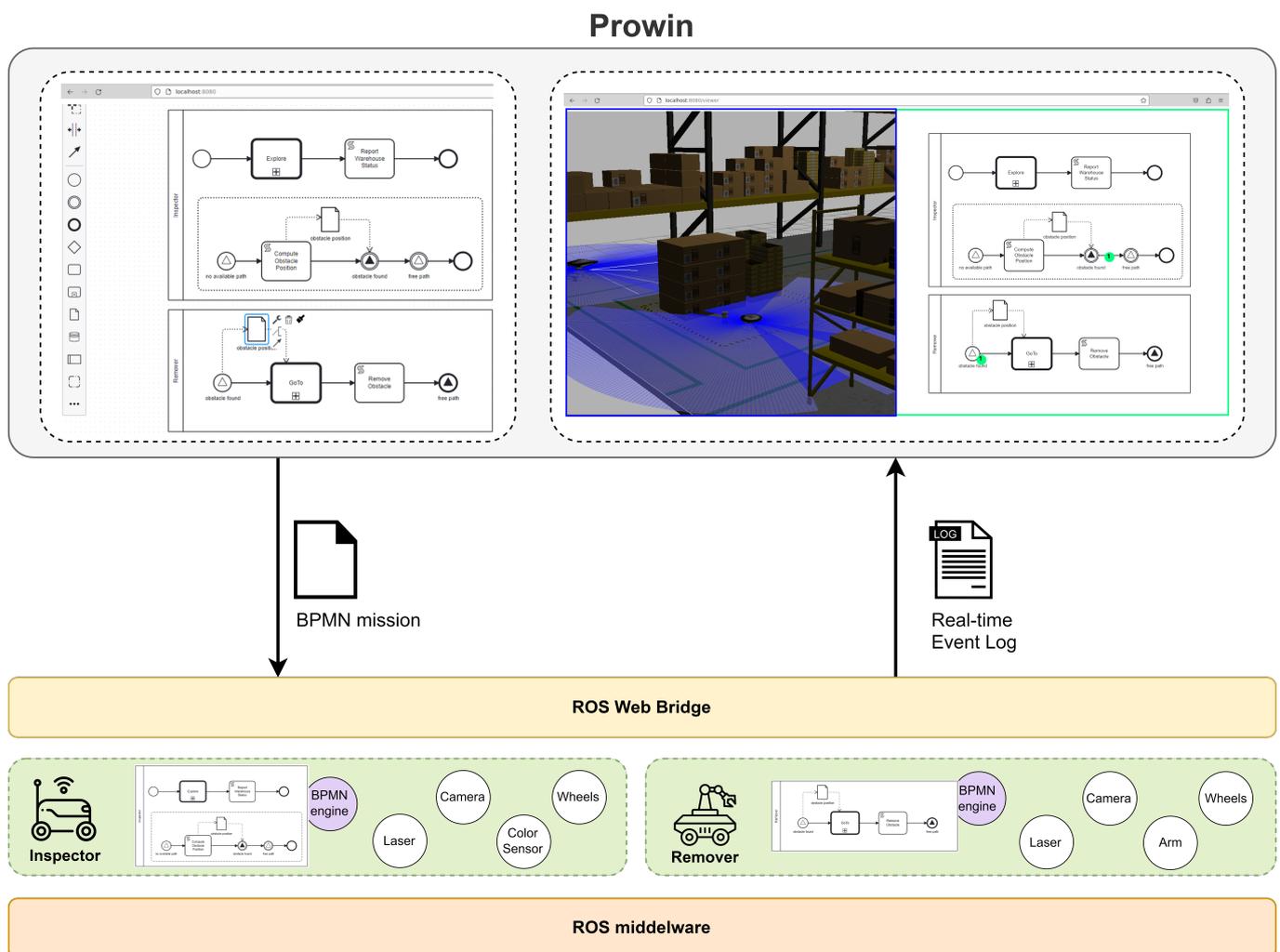


Figure 4. The PROWIN architecture.

## 5. Related Works

In the literature, DPTs are just presented theoretically [11], whereas the implementation of DTs for IoT and robotic systems is increasingly popular [7,15]. Among all the application scenarios, in industrial contexts, effective collaboration between humans and autonomous devices is crucial to enhance the productivity of the entire system. Specifically, DTs have the potential to give real-time status on devices' performances, as well as production line feedback [7].

Most works exploit 3D environments to create a DT that faithfully shows the physics of system devices and simulates possible execution outcomes. In particular, DTs can be developed to effectively train machine learning algorithms before deploying them into devices [52], thus integrating autonomous devices with efficient decision-making and problem-solving skills [53,54]. The combination of a 3D environment and a robotic application can additionally enable the analysis of system performance as well as the graphical verification of the robot's behavior [55]. Differently, a realistic reproduction of the system exploiting augmented reality enables developers to interact with the system and monitor the corresponding actions in real-time [56]. Moreover, the realistic reproduction of an industrial production line allows the simulation of system behavior without the risk of financial loss in the real production line [57].

Focusing on the monitoring of the system behavior, a DT should clearly and unambiguously specify and represent current and expected device behavior. Therefore, the monitoring of the system operation by constructing a DT from the data collected in real-time gives reasons for the effects of interventions, including those that have not been observed before [58]. More in detail, a DT representing the system behavior can be exploited to detect and resolve anomalies [59] and enhance the operational efficiency while performing a what-if analysis to simulate the hypothetical situation in preparation for the possible change of the external factors [60], but can also be combined with model-driven solutions to develop and maintain digital twin infrastructures to automatically link the physical side with the digital one [61]. Moreover, if the physical system is driven by processes, it is possible to exploit processes to easily create the behavioral DT and allow the runtime synchronization between the digital and physical parts [25].

In summary, the existing literature proposes different approaches regarding the realistic representation of the physical system and the monitoring of system behavior, but lacks a unified solution that effectively integrates them. Our work improves the state-of-the-art by proposing the novel concept of xDPT, which combines the monitoring, analysis, and refinement of the physical system behavior and is supported by the PROWIN tool to show how to apply it in a multi-robot case study.

## 6. Conclusions

Considering the spread of process-driven systems, this paper presents the novel concept of executable digital process twins to effectively enable the monitoring, analysis, and refinement of such systems. In this regard, we analyze the concept of both the digital twin of an organization and of the executable digital twin to underline their advantages and limitations, thus showing how process-driven systems can benefit from their combination. Moreover, we illustrate how to implement an xDPT in a cooperative multi-robot scenario. The implementation results in the PROWIN tool implementing the monitoring of the executed system from the process and the physical perspectives. The tool also allows the deployment of a refined process model into the robots, thus enabling the synchronization between the physical and the digital systems. Finally, we assess the tool in a BPMN-driven MRS deployed in a warehouse.

### 6.1. Discussion

The definition and the implementation of an xDPT result from the need to integrate DTs in process-driven systems. Specifically, when creating a DPT solely based on real-time event logs, process mining techniques are applied to discover and analyze the system's

execution. However, existing techniques often lack adequate solutions for distributed scenarios involving multiple entities, such as IoT devices and robots [26]. Consequently, process models derived from current process discovery techniques may not faithfully represent the dynamic nature of the running system, rendering them unsuitable for direct system enhancement. On the other hand, when the system's execution is driven by a pre-defined process, the refinements made to the DPT can be readily utilized to enhance the system. In such cases, where the process is known in advance, the modifications and optimizations applied to the DPT can be employed to improve the system's performance and capabilities. This targeted approach allows for more effective integration of the DPT with the underlying process-driven system, enabling valuable insights and enhancements to be directly derived from the DPT.

Additionally, the case study we presented is from a warehouse management scenario. Notably, the application of an xDPT can enhance different fields that already benefit from the application of DTs, such as healthcare, smart cities, and agriculture [62]. For instance, the integration of IoT devices and robots in healthcare has led to the development of advanced solutions, such as wearable monitoring devices and surgical robots. Similarly, their collaboration in smart cities can be exploited to optimize transportation and public services. Finally, in agriculture, IoT devices and robots enhance crop productivity and sustainability, reducing manual labor and increasing efficiency. Moreover, the increasing utilization of BPMN-driven approaches in IoT and robot development makes these systems well-suited for the application of xDPTs. This integration will further enhance system efficiency, monitoring capabilities, and decision making, regardless of the specific application scenario.

## 6.2. Future Work

In future research, we plan to integrate the PROWIN tool with process mining techniques to enhance the analysis capabilities of the xDPT. Indeed, the integration of process mining techniques will enable us to discover hidden patterns, bottlenecks, and inefficiencies in the execution of processes within the xDPT. We will explore methods for automatically extracting process models from event logs and aligning them with the represented process models. This integration will provide a complete view of the system's behavior, allowing for more accurate simulations, predictions, and optimizations. Moreover, we plan to explore the integration of predictive analytics techniques. By leveraging historical event logs and combining them with machine learning algorithms, we can develop predictive models that anticipate future system behavior, enabling decision making and preventive actions.

**Author Contributions:** All authors of this manuscript have equal contributions in conceptualization. Methodology, S.P. and L.R.; software, S.P. and L.R.; validation, S.P. and L.R.; investigation, S.P.; resources, S.P.; writing—original draft preparation, S.P.; writing—review and editing, S.P., B.R., L.R. and F.T.; supervision, B.R., L.R. and F.T.; project administration, F.C.; funding acquisition, F.C. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by PNRR MUR project ECS\_00000041-VITALITY.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The code is available from the authors. The corresponding author will provide the code on request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

MRS	Multi-Robot System
BPMN	Business Process Model and Notation
ROS	Robot Operating System 2
DTO	Digital Twin of an Organization
xDT	Executable Digital Twin
DPT	Digital Process Twin
xDPT	Executable Digital Process Twin

## References

1. Weske, M. *Business Process Management Architectures*; Springer: Berlin/Heidelberg, Germany, 2007.
2. Dumas, M.; La Rosa, M.; Mendling, J.; A Reijers, H. *Fundamentals of Business Process Management*; Springer: Berlin/Heidelberg, Germany, 2013.
3. Aguilar-Savén, R.S. Business process modelling: Review and framework. *Int. J. Prod. Econ.* **2004**, *90*, 129–149. [[CrossRef](#)]
4. Corradini, F.; Fedeli, A.; Polini, A.; Re, B. Towards a Digital Twin Modelling Notation. In Proceedings of the Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, Falerna, Italy, 12–15 September 2022; pp. 1–6.
5. Corradini, F.; Pettinari, S.; Re, B.; Rossi, L.; Tiezzi, F. An Approach to Support Digital Process Twin. In Proceedings of the Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress, Falerna, Italy, 12–15 September 2022; pp. 1–4.
6. Bordeleau, F.; Combemale, B.; Eramo, R.; van den Brand, M.; Wimmer, M. Towards Model-Driven Digital Twin Engineering: Current Opportunities and Future Challenges. In Proceedings of the Systems Modelling and Management, Bergen, Norway, 25–26 June 2020; Volume 1262, pp. 43–54.
7. Fuller, A.; Fan, Z.; Day, C.; Barlow, C. Digital twin: Enabling technologies, challenges and open research. *IEEE Access* **2020**, *8*, 108952–108971. [[CrossRef](#)]
8. Valderas, P.; Torres, V.; Serral, E. Modelling and executing IoT-enhanced business processes through BPMN and microservices. *J. Syst. Softw.* **2022**, *184*, 111139. [[CrossRef](#)]
9. Bourr, K.; Corradini, F.; Pettinari, S.; Re, B.; Rossi, L.; Tiezzi, F. Disciplined use of BPMN for mission modeling of Multi-Robot Systems. In Proceedings of the Forum at Practice of Enterprise Modeling, Riga, Latvia, 24–26 November 2021; Volume 3045, pp. 1–10.
10. Corradini, F.; Marcelletti, A.; Morichetta, A.; Polini, A.; Re, B.; Tiezzi, F. A Choreography-Driven Approach for Blockchain-Based IoT Applications. In Proceedings of the 2022 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), Pisa, Italy, 21–25 March 2022; pp. 255–260.
11. van der Aalst, W. Concurrency and objects matter! Disentangling the fabric of real operational processes to create digital twins. In Proceedings of the Theoretical Aspects of Computing, Tbilisi, Georgia, 27–29 September 2022; pp. 3–17.
12. OMG. *Business Process Model and Notation (BPMN)*, version 2.0; OMG: Milford, MA, USA, 2011.
13. Corradini, F.; Pettinari, S.; Re, B.; Rossi, L.; Tiezzi, F. A BPMN-driven framework for Multi-Robot System development. *Robot. Auton. Syst.* **2023**, *160*, 104322. [[CrossRef](#)]
14. Parmar, R.; Leiponen, A.; Thomas, L.D. Building an organizational digital twin. *Bus. Horiz.* **2020**, *63*, 725–736. [[CrossRef](#)]
15. Caesar, B.; Hänel, A.; Wenkler, E.; Corinth, C.; Ihlenfeldt, S.; Fay, A. Information Model of a Digital Process Twin for Machining Processes. In Proceedings of the Emerging Technologies and Factory Automation, Vienna, Austria, 8–11 September 2020; pp. 1765–1772.
16. Caporuscio, M.; Edrisi, F.; Hallberg, M.; Johannesson, A.; Kopf, C.; Perez-Palacin, D. Architectural concerns for digital twin of the organization. In Proceedings of the Software Architecture, L'Aquila, Italy, 14–18 September 2020; pp. 265–280.
17. Becker, M.C.; Pentland, B.T. Digital twin of an organization: Are you serious? In Proceedings of the Business Process Management Workshops, Rome, Italy, 6–10 September 2021; pp. 243–254.
18. Park, G.; Comuzzi, M.; van der Aalst, W. Analyzing Process-Aware Information System Updates Using Digital Twins of Organizations. In Proceedings of the Research Challenges in Information Science, Barcelona, Spain, 17–20 May 2022; pp. 159–176.
19. Riss, U.V.; Maus, H.; Javaid, S.; Jilek, C. Digital twins of an organization for enterprise modeling. In Proceedings of the Practice of Enterprise Modeling, Riga, Latvia, 25–27 November 2020; pp. 25–40.
20. Lo, C.; Chen, C.; Zhong, R.Y. A review of digital twin in product design and development. *Adv. Eng. Inform.* **2021**, *48*, 101297. [[CrossRef](#)]
21. Hartmann, D.; der Auweraer, H.V. Digital Twins. *arXiv* **2020**, arXiv:abs/2001.09747.
22. Scurria, L.; Risaliti, E.; Buss, D.; Kubo, P.; Tamarozzi, T.; Cornelis, B. Executable digital twin-prevent the early failure of a truck anchorage using smart virtual sensors. *Int. J. Adv. Curr. Pract. Mobil.* **2022**, *4*, 1309–1317. [[CrossRef](#)]
23. Hartmann, D.; der Auweraer, H.V. The Executable Digital Twin: Merging the digital and the physics worlds. *arXiv* **2022**, arXiv:abs/2210.17402.

24. Allamaa, J.P.; Patrinos, P.; der Auweraer, H.V.; Son, T.D. Sim2real for Autonomous Vehicle Control Using Executable Digital Twin. *IFAC-PapersOnLine* **2022**, *55*, 385–391. [[CrossRef](#)]
25. Valderas, P. Supporting the Implementation of Digital Twins for IoT-Enhanced BPs. In Proceedings of the International Conference on Research Challenges in Information Science, Corfu, Greece, 23–26 May 2023; pp. 222–238.
26. Corradini, F.; Re, B.; Rossi, L.; Tiezzi, F. A Technique for Collaboration Discovery. In Proceedings of the Business Process Modeling, Development and Support, Leuven, Belgium 6–7 June 2022; Volume 450, pp. 63–78.
27. Carmona, J.; van Dongen, B.; Weidlich, M. Conformance checking: Foundations, milestones and challenges. In *Process Mining Handbook*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 448, pp. 155–190.
28. Fahland, D. Process mining over multiple behavioral dimensions with event knowledge graphs. In *Process Mining Handbook*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 448, pp. 274–319.
29. Di Francescomarino, C.; Ghidini, C. Predictive process monitoring. In *Process Mining Handbook*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 448, pp. 320–346.
30. Hasić, F.; Asensio, E.S. Executing IoT processes in BPMN 2.0: Current support and remaining challenges. In Proceedings of the International Conference on Research Challenges in Information Science, Brussels, Belgium, 21 October 2019; pp. 1–6.
31. Corradini, F.; Polini, A.; Re, B.; Rossi, L.; Tiezzi, F. Consistent modelling of hierarchical BPMN collaborations. *Bus. Process Manag. J.* **2022**, *28*, 442–460. [[CrossRef](#)]
32. Corradini, F.; Muzi, C.; Re, B.; Rossi, L.; Tiezzi, F. Formalising and animating multiple instances in BPMN collaborations. *Inf. Syst.* **2022**, *103*, 101459. [[CrossRef](#)]
33. Muzi, C.; Pufahl, L.; Rossi, L.; Weske, M.; Tiezzi, F. Formalising BPMN Service Interaction Patterns. In Proceedings of the Practice of Enterprise Modeling, Vienna, Austria, 31 October–2 November 2018; Volume 335, pp. 3–20.
34. Domingos, D.; Martins, F. Using BPMN to model Internet of Things behavior within business process. *Int. J. Inf. Syst. Proj. Manag.* **2017**, *5*, 39–51. [[CrossRef](#)]
35. Caracas, A. From business process models to pervasive applications: Synchronization and optimization. In Proceedings of the International Conference on Pervasive Computing and Communications Workshops, Lugano, Switzerland, 19–23 March 2012; pp. 320–325.
36. Kirikkayis, Y.; Gallik, F.; Winter, M.; Reichert, M. BPMNE4IoT: A framework for modeling, executing and monitoring IoT-driven processes. *Future Internet* **2023**, *15*, 90. [[CrossRef](#)]
37. Otsu, K.; Tepsuporn, S.; Thakker, R.; Vaquero, T.S.; Edlund, J.A.; Walsh, W.; Miles, G.; Heywood, T.; Wolf, M.T.; Agha-Mohammadi, A.-A. Supervised Autonomy for Communication-degraded Subterranean Exploration by a Robot Team. In Proceedings of the AeroConf, Big Sky, MT, USA, 7–14 March 2020; pp. 1–9.
38. Singh, M.; Fuenmayor, E.; Hinchy, E.P.; Qiao, Y.; Murray, N.; Devine, D. Digital twin: Origin to future. *Appl. Syst. Innov.* **2021**, *4*, 36. [[CrossRef](#)]
39. Allweyer, T.; Schweitzer, S. A tool for animating BPMN token flow. In Proceedings of the Business Process Model and Notation, Vienna, Austria, 12–13 September 2012; pp. 98–106.
40. Signavio. Available online: <https://www.signavio.com> (accessed on 5 April 2023).
41. Corradini, F.; Muzi, C.; Re, B.; Rossi, L.; Tiezzi, F. MIDA: Multiple Instances and Data Animator. In Proceedings of the BPM (Dissertation/Demos/Industry), Sydney, Australia, 9–14 September 2018; pp. 86–90.
42. Abdul, B.M.; Corradini, F.; Re, B.; Rossi, L.; Tiezzi, F. UBBA: Unity Based BPMN Animator. In Proceedings of the Information Systems Engineering in Responsible Information Systems, Rome, Italy, 3–7 June 2019; Volume 350, pp. 1–9.
43. Visual Paradigm, Business Process Design with Powerful BPMN Software. Available online: <https://www.visual-paradigm.com/features/bpmn-diagram-and-tools> (accessed on 5 April 2023).
44. Lee, W.; Cho, S.; Chu, P.; Vu, H.; Helal, S.; Song, W.; Jeong, Y.S.; Cho, K. Automatic agent generation for IoT-based smart house simulator. *Neurocomputing* **2016**, *209*, 14–24. [[CrossRef](#)]
45. De Melo, M.S.P.; da Silva Neto, J.G.; Da Silva, P.J.L.; Teixeira, J.M.X.N.; Teichrieb, V. Analysis and comparison of robotics 3d simulators. In Proceedings of the Symposium on Virtual and Augmented Reality, Rio de Janeiro, Brazil, 28–31 October 2019; pp. 242–251.
46. Bounceur, A.; Clavier, L.; Combeau, P.; Marc, O.; Vauzelle, R.; Masserann, A.; Soler, J.; Euler, R.; Alwajeel, T.; Devendra, V.; et al. CupCarbon: A new platform for the design, simulation and 2D/3D visualization of radio propagation and interferences in IoT networks. In Proceedings of the Annual Consumer Communications & Networking Conference, Las Vegas, NV, USA, 12–15 January 2018; pp. 1–4.
47. Burattin, A. Streaming process mining. In *Process Mining Handbook*; Springer: Berlin/Heidelberg, Germany, 2022; Volume 448, pp. 349–372.
48. Tavares, G.M.; Ceravolo, P.; Da Costa, V.G.T.; Damiani, E.; Junior, S.B. Overlapping analytic stages in online process mining. In Proceedings of the International Conference on Services Computing, Milan, Italy 8–13 July 2019; pp. 167–175.
49. Burattin, A.; Re, B.; Rossi, L.; Tiezzi, F. A purpose-guided log generation framework. In Proceedings of the Business Process Management; Springer: Münster, Germany, 2022; Volume 13420, pp. 181–198.
50. Brockhoff, T.; Heithoff, M.; Koren, I.; Michael, J.; Pfeiffer, J.; Rumpel, B.; Uysal, M.S.; Van Der Aalst, W.; Wortmann, A. Process prediction with digital twins. In Proceedings of the International Conference on Model Driven Engineering Languages and Systems Companion, Fukuoka, Japan, 10–15 October 2021; pp. 182–187.

51. Hemmer, A.; Badonnel, R.; Chriment, I. A process mining approach for supporting IoT predictive security. In Proceedings of the Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–9.
52. Matulis, M.; Harvey, C. A robot arm digital twin utilising reinforcement learning. *Comput. Graph.* **2021**, *95*, 106–114. [[CrossRef](#)]
53. Alexandru, M.; Dragoş, C.; Bălă-Constantin, Z. Digital Twin for automated guided vehicles fleet management. *Procedia Comput. Sci.* **2022**, *199*, 1363–1369. [[CrossRef](#)]
54. Jaensch, F.; Csiszar, A.; Scheifele, C.; Verl, A. Digital Twins of Manufacturing Systems as a Base for Machine Learning. In Proceedings of the Mechatronics and Machine Vision in Practice, Stuttgart, Germany, 20–22 November 2018; IEEE: Stuttgart, Germany, pp. 1–6.
55. Martínez-Gutiérrez, A.; Díez-González, J.; Ferrero-Guillén, R.; Verde, P.; Álvarez, R.; Perez, H. Digital twin for automatic transportation in industry 4.0. *Sensors* **2021**, *21*, 3344. [[CrossRef](#)]
56. Jiang, Y.; Yin, S.; Li, K.; Luo, H.; Kaynak, O. Industrial applications of digital twins. *Philos. Trans.* **2021**, *379*, 20200360. [[CrossRef](#)]
57. Vachálek, J.; Bartalský, L.; Rovný, O.; Šišmišová, D.; Morhác, M.; Lokšík, M. The digital twin of an industrial production line within the industry 4.0 concept. In Proceedings of the International Conference on Process Control, Strbske Pleso, Slovakia, 6–9 June 2017; pp. 258–262.
58. Dumas, M. Constructing Digital Twins for Accurate and Reliable What-If Business Process Analysis. In Proceedings of the Workshop on BPM Problems to Solve before We Die, Rome, Italy, 6–10 September 2021; Volume 2938, pp. 23–27.
59. Sleuters, J.; Li, Y.; Verriet, J.; Velikova, M.; Doornbos, R. A Digital Twin Method for Automated Behavior Analysis of Large-Scale Distributed IoT Systems. In Proceedings of the System of Systems Engineering, Anchorage, AK, USA, 19–22 May 2019; pp. 7–12.
60. Lee, H.D.; Guo, K.; Souza, L.F.; Lee, J.M. Application of Digital Twin to Monitor and Optimize Utility Process. In Proceedings of the Control, Automation and Systems, Jeju, Republic of Korea, 12–15 October 2021; pp. 376–381.
61. Lehner, D.; Sint, S.; Vierhauser, M.; Narzt, W.; Wimmer, M. AML4DT: A Model-Driven Framework for Developing and Maintaining Digital Twins with AutomationML. In Proceedings of the IEEE Emerging Technologies and Factory Automation, Vasteras, Sweden, 7–10 September 2021; pp. 1–8.
62. Zhang, R.; Wang, F.; Cai, J.; Wang, Y.; Guo, H.; Zheng, J. Digital twin and its applications: A survey. *J. Adv. Manuf. Technol.* **2022**, *123*, 4123–4136. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.