Proceedings of the 2024 Winter Simulation Conference
H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.

TOWARDS STANDARDIZING THE INTEGRATION OF DIGITAL TWINS IN MANUFACTURING SYSTEMS

Hedir Oukassi[1], Amel Jaoua[1], Soumaya Yacout[2], Elisa Negri[3], and Mehdi Jaoua[4]

[1]Dept. of Industrial Eng. National Engineering School of Tunis, University of Tunis ElManar, TUNISIA
[2]Dept. of Mathematics and Industrial Engineering, École Polytechnique de Montréal, CANADA
[3]Dept. of Management, Economics and Industrial Engineering, Politecnico di Milano, ITALY
[4]Independant Researcher, Florida, USA

ABSTRACT

Industrial sectors are increasingly prioritizing the integration of Digital Twin (DT) technology into their operations to enhance manufacturing system decision-making. However, implementing DTs remains highly challenging due to its novelty, highlighting the pressing need for the development of more formal methodologies to guide its effective implementation. Therefore, the aim of this paper is to present ongoing research concerning DT design and implementation in accordance with the ISO 23247 standard. The objective of this DT is to embed real-time Autonomous Mobile Robot dispatching decision in a small-scale production system. The present work outlines the essential steps for transitioning from a traditional Discrete Event Simulation model to a Real-Time Simulation, ensuring connectivity and synchronization with the physical system to achieve efficient implementation. It also exhibits the benefit of using rigorous formalism clarifying the complex sequencing involved in capturing and transmitting real-time sensor data between physical and cyber system.

1 INTRODUCTION

The rise of Industry 4.0 technologies offers exciting prospects for the manufacturing industry. However, it also poses significant challenges, particularly in integrating Cyber-Physical Systems (CPS). As defined by Cimino et al. (2019) CPS involve the fusion of information technology with embedded system processes. Within CPS, Digital Twins (DT) emerge as a powerful tool for empowering stakeholders in decision-making processes (Negri et al. 2017). To integrate DTs into CPSs, it is essential to consider diverse complex requirements, such as modeling methods and interactions, from the design phase through implementation (Fett et al. 2023).

In recognition of this challenge, many international organizations have proposed standards and reference architectures, encouraging companies to adopt them to enhance their integration of DTs. A comprehensive review of the technology standard for enabling DT is given by Wang et al. (2022). The International Organization for Standardization (ISO) was the first committee to develop the DT project and published the digital twin standard series in the field of smart manufacturing namely the ISO 23247 (ISO, 2020). While it is important to follow standardized reference architectures to ensure efficient integration of DTs, this practice is still uncommon in many research papers. Ferko et al. (2023) reviewed a substantial body of literature, pointing out the misalignment with the standard. The need for standardization has also been recently highlighted by Aboelhassan et al. (2023) and Jaoua et al. (2024), particularly in the context of DT for production planning and control.

Thus, the objective of this work is to develop and implement a DT which adheres to the ISO 23247 standard for addressing Autonomous Mobile Robot (AMR) dispatching problem in a small-scale production system. Specifically, our focus will be on the data model specification and data exchange discussed in Part 3 and Part 4 of the ISO 23247 series. We will use ISO 23247-3 to define the Observable Manufacturing

Elements (OME), their attributes, and the corresponding data structure. Additionally, information exchange and connectivity will be outlined according to ISO 23247-4. A practical implementation of this designed DT will then be conducted through the integration of a Real-Time Simulator (RTS). An RTS is a continuously synchronized Discrete Event Simulation (DES) model connected with the physical system (Lugaresi and Matta 2018). The present work outlines the essential steps for transitioning from a traditional DES model to a RTS model, ensuring fundamental aspects of connectivity and synchronization with physical system for an efficient DT implementation. According to Wooley et al. (2023) and Matta and Lugaresi (2023), Many papers on DTs lack clarity on how to continuously synchronize information flow between cyber and physical systems. Therefore, the implementation phase described herein aims to clarify this issue through a real case study, demonstrating how real-time data exchange is embedded.

This paper is structured as follows: Section 2 reviews the most related works. Section 3 introduces the designed DT aligned with the ISO 23247 standard. Section 4 presents the implementation of phases in the small-scale production system. Finally, Section 5 provides the conclusion, as well as future research directions.

2 RELATED WORKS

Over the past five years many definitions have been proposed to describe the challenging DT concepts due to its interdisciplinary appeal to researchers from various fields. Lim et al. (2020) describe it as a high-fidelity replica of a physical asset that interacts with its digital counterpart in real time, while Barbieri et al. (2021) provide further clarification by defining a DT as a concept characterized by the bidirectional exchange of information between the physical entity and its corresponding digital replica in cyberspace. In alignment with these definitions, the process of building a digital twin involves specific levels of integration, as highlighted by several researchers, including Aboelhassan et al. (2023). The first level of integration includes the digital model (DM), which serves as a digital representation of the physical system with a manual data flow. The second level of integration is the Digital Shadow (DS), characterized by an automated unidirectional data flow from the physical object to the virtual object without feedback. Finally, the DT represents the highest level of integration, characterized by an automated bidirectional data flow between the physical object and its virtual counterpart, which is the focus of this study.

For better standardization, major organizations such as the International Organization for Standardization (ISO), the International Electrotechnical Commission (IEC), the International Telecommunication Union (ITU), and the Institute of Electrical and Electronics Engineers (IEEE), have focused on defining DT standards (Wang et al. 2022). For smart manufacturing the first work was proposed by ISO 23247, titled "Automation systems and integration - Digital twin framework for manufacturing". This ISO 23247 series establishes a framework to support the creation of DT based on the OME including personnel, equipment, materials, manufacturing processes, etc. This series is composed of four parts addressing different scopes: Part 1 and 2 deals with general principles and functional views, whereas Part 3 (ISO 23247-3) depicts OME and Part 4 (ISO 23247-4) deals with information exchange. In the field of modeling, ISO encourages the use of existing modeling standards for implementing the standard, specifically the IEC 62264 series for Enterprise-control system integration (IEC. 2013).

Using such standards can facilitate the design and implementation of DTs in manufacturing, promoting efficiency and consistency in production planning and control. However, according to Ferko et al. (2023), very few works have conformed to these standards. Pystina et al. (2022) have described a design process toward implementing a DT in existing production system. Reference Architectural Model Industry 4.0 (RAMI 4.0) is used as a reference architecture to describe asset characteristics on different layers. This work is based on models from the IEC standard. The application in a flexible production line with several working posts equipped with a robot is described. Vasilev et al. (2023) have utilized the aforementioned standard in the design and development of Manufacturing Operations Management systems, which involve mandatory Unified Modeling Language (UML) modelling. They showed that UML can describe resource behavior comprehensively across various use cases, facilitating the construction of DTs. Also, Marah et al. (2023), introduced the Multi-Agent DT framework, operating within the domain of cyber-physical systems in manufacturing. This framework, driven by a Multi-Agent Systems paradigm, features a flexible architecture for modeling agent-based DTs. Through a case study, they demonstrate the feasibility and applicability of this framework.

While the benefits of adhering to standards are obvious from these studies, they primarily focus on the design phases of DTs, with limited exploration into implementation and corresponding complexity of simulation models connectivity. In fact, to the best of our knowledge and supported by recent research (Wooley et al. 2023; Abouelhassan et al. 2023; Jaoua et al. 2024), the effective implementation of formally designed DTs remains a challenge. Wooley et al. (2023) conducted a comprehensive review of the proposed DT application and noticed that many DT-related papers are vague about implementation details and cannot be replicated. More specifically, it remains ambiguous how real-time connectivity is established between physical and virtual systems within software tools. Also, according to Matta and Lugaresi (2023), it remains unclear how the continuous synchronization of information flow is accomplished. Thus, the purpose of the present work is to elucidate these implementation issues while aligning with the reference architecture presented in the ISO 23247. Furthermore, to assure replicability the developed code to establish continuous synchronization of information flow with the RTS is provided.

3 DESIGN OF THE DT'S OBSERVABLE ELEMENTS AND DATA EXCHANGE

3.1 Description of the Learning Factory

The objective of this work is to solve the AMR dispatching problem arising in a small-scale production system reproduced at a Learning Factory (LF) located at the National Engineering School of Tunis, Tunisia. At this LF, the Festo Modular Production System, MPS® system 403-1, from Festo® Didactic is used as an Industry 4.0 learning system as well as for research purposes. The MPS 403-1 is composed of three automated workstations: Distribution, Joining and Sorting, Figure 1-a. The role of these workstations is to produce the customer order received at the Manufacturing Execution System (MES) to mimic the make-to order approach in the e-commerce context. We consider three types of customized order quantity corresponding to a batch of 2, 4 or 6 workpieces. Once these customized orders are produced in the MPS 403-1, they are transported for packing at one of the two Manual Assembly Stations. The pickup and delivery are conducted by an Autonomous Mobile Robot (AMR) named Robotino®, from Festo® didactic. The corresponding layout is given in Figure 1-b. The AMR navigates within an aisle, colored in blue in Figure 1-b, to perform pickup from the MPS 403-1 and delivery to the operator at Manual workstations.

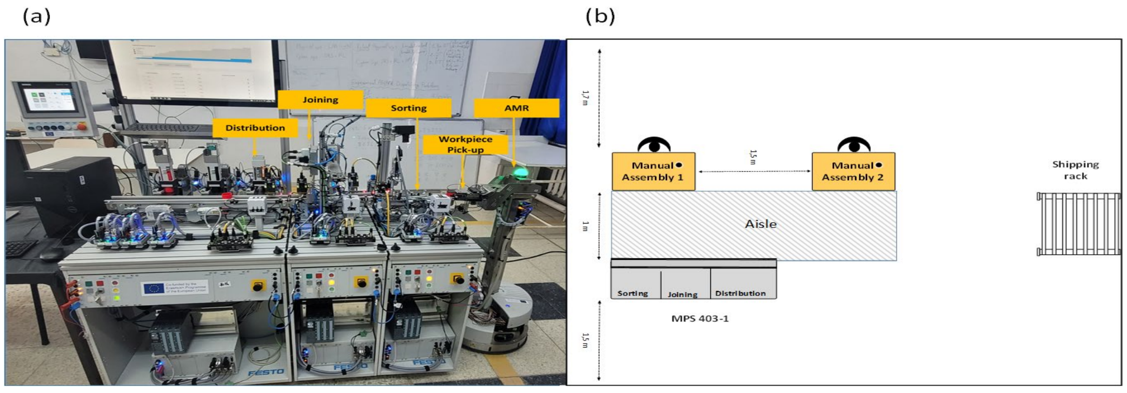


Figure 1: Learning Factory (a) The MPS with the AMR; (b) Layout of the use case in the LF.

The dispatching problem addressed herein involves sending the AMR to one of the Manual Assembly stations with the objective of minimizing the makespan. This decision has to be made by the DT, based on a pre-trained RL algorithm on the corresponding DES model. Herein the RL training is conducted offline on the DES model with the objective to find the best action, i.e. assigning the AMR to Manual assembly 1 or 2, giving the State of the physical system. The State is a vector composed of the Number of workpieces waiting in the Queue of each Manual Assembly Station, the type of the last waiting order in each queue, and the type of the next order. During the training a feedforward Neural Network is used to take the environment's State and to approximate the Q-value for each Action at the current state to choose the one that gives a higher Reward. The weights of the Neural Network are updated along this training process. Once the training is completed, the DES model will no more be invoked, and the DT will use the RTS in real-time. The role of the RTS in the DT architecture is to continuously provide a snapshot of the physical system's state. This state is used by the DT to determine from the pre-trained RL on the DES model, the corresponding optimal real-time dispatching decisions for the AMR. This decision will be sent from the DT to the AMR via the RTS as specified in the following section 4. In fact, within this architecture, there is no need to accelerate the RTS for online prediction, as the training with the traditional DES model has already been completed offline. For more insights the used RL optimization mechanism, interested readers could consult (Jaoua et al. 2024). In this paper, we will focus on the implementation of this decision with the DT architecture and the related synchronization issues for state tracking.

3.2 Design of the DT Observable Manufacturing Elements

According to the definition given by ISO 23247-3 the herein OME, are the "Order", the "AMR" and "Workstations" which represent physical components integrated into the digital twin's data architecture. An Order is a type of a workpiece. The "Digital Twin Services" class acts as a mediator, ensuring that each physical action and state change is accurately captured and reflected in the DT in real-time. The corresponding attributes are depicted in the following UML Class diagram; Figure 2. ISO have also recommended to base the modelling approach of the DT on general standard such as IEC 62264, which consists of using the robust UML formalism. For more insight, the UML formalism reader could consult (Miles and Hamilton 2006).

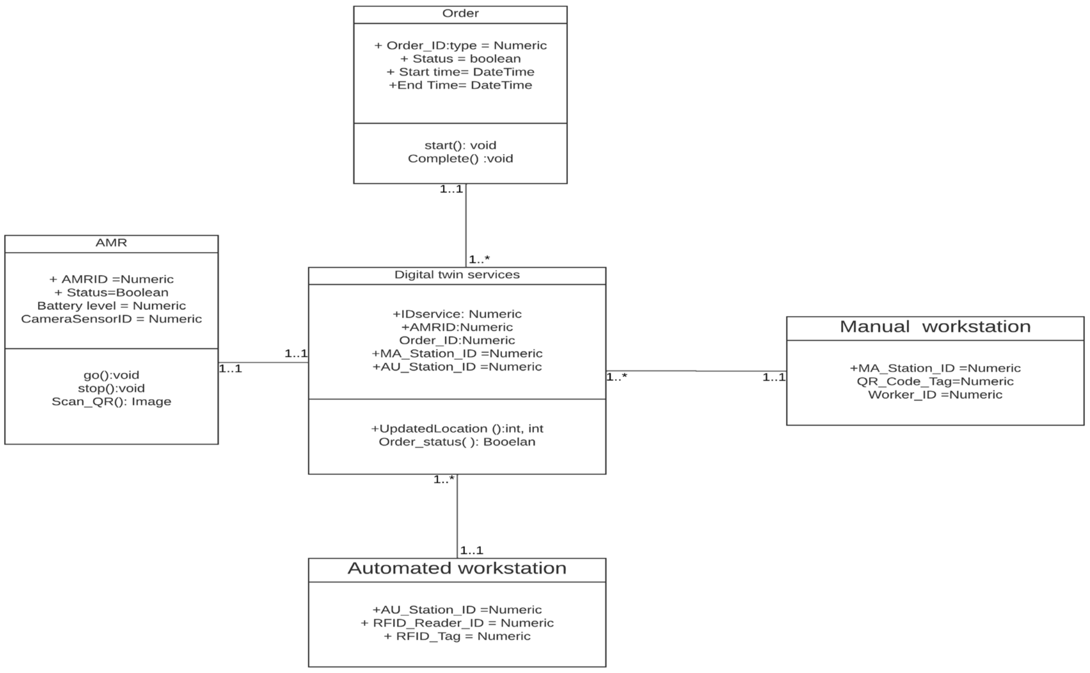


Figure 2: DT services UML class diagram.

As depicted in Figure 2, the "Digital Twin Services" class is central to this structure. It coordinates the workflow and ensures synchronization with real-time sensor data from physical components. There are two types of workstations, distinguished by their tracking technologies. Herein the Distribution, Joining and Sorting are instances of the "Automated workstation" class. These stations are equipped with Radio Frequency Identification (RFID) Readers. The DT communicate with the "AMR" which is equipped with Camera Sensor to reflect its position. The Assembly stations 1 and 2 belongs to the class "Manual workstation" and are identified with their QR_Code_Tag attribute. A customer "Order" is characterized by its type which indicates the number of workpieces to produce. The following subsection will illustrate the connectivity technologies and their usage.

3.3 Connectivity and Information Exchange

In this DT application two types of connectivity between the OME will be established. The first one is between the DT, the "Order" and the "Automated workstation" and is based on the OPC-UA protocol embedded within the Turck's RFID system. The second is between the DT, the "AMR" and the "Manual Workstation" and is based on the REST API communication protocol. According to the ISO 23247-4 these two protocols ensure reliable data exchange between manufacturing devices and DT.

The first communication allows to track the position of the workpiece when it moves through the three automated workstation. The exit of each workstation is equipped with an RFID write/read device, and each workpiece has a tag inserted into it, Figure 3-a. The vendor of this RFID system, Turck, provides the middleware and establishes connectivity to track and store RFID data in real-time.

To establish the second communication with the AMR, we use the REST API protocol due to its compatibility with the Robotino system for communication, control, and data access. Also, the AMR, Figure 3-b, is equipped with camera. The REST API streamlines operations by sending requests to the Robotino for navigating to specific "Manual Station" and to the camera sensor for activating QR code scanning. A specific QR code is associated to each manual workstation for delivery and also another QR code is placed at the dispatching point i.e. for the pickup at the exit of the MPS 403.1, Figure 3-b.

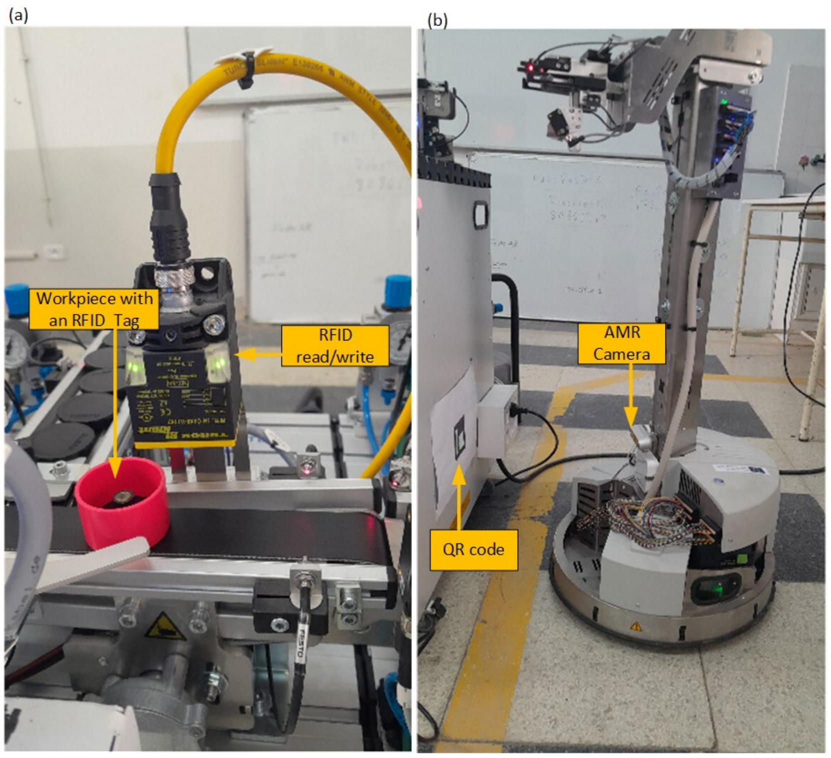


Figure 3: (a) RFID read/write device at the exit of joining station, (b) Camera sensor of the AMR.

4 IMPLEMENTATION OF THE PROPOSED DT FRAMEWORK

This section details the practical steps taken to implement the designed DT. It will first introduce the developed DES model and the embedded information exchange mechanism. Then, it will describe the steps taken to transform this DES into a RTS model.

4.1 DES Model Specification and Data Synchronization

4.1.1 DES Model Specification

At this stage we will translate the previously defined Class diagram of DT services with the corresponding identified OME, Figure 2, into a DES model. For that purpose, we will use the Simio simulation software. Herein the ISO 23247 does not prescribe specific software tools or platforms for developing simulation models. In our application we choose this tool given its good performance in data exchange which facilitates cyber-physical system modelling (Devanga et al. 2022). For readers unfamiliar with Simio, please consult Smith and Sturrock (2023).

We firstly modeled the Order Arrivals through a data table that contains the list of orders. The “Arrival Mode” property of this object was configured as “Arrival Table” enabling the retrieval of all order data from the arrival table for use during run-time. Then each of the three Automated Workstation: Distribution, Joining and Sorting are modeled as “Server objects”. The AMR is modeled as a vehicle that transports the manufactured workpiece from the last MPS 403-1 station, i.e. the Sorting Automated Workstation, to one of the Manual Workstations. At these two Manual Workstation customer orders are combined to be assembled in boxes according to the predefined order type, i.e. workpiece quantity. To implement this process, we use the “Combiner objects”. Once assembled, these orders are placed for delivery. The corresponding 2D visualization of this small-scale production system is given in Figure 4.

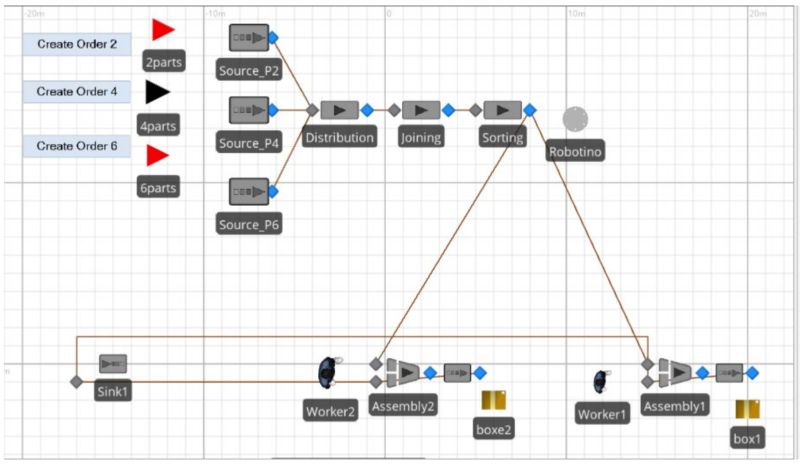


Figure 4: The 2D Simio model.

After modeling the DM of the OME, the next step will deal with the identification of synchronization points and the data exchange embedment.

4.1.2 Data Synchronization Mechanism

To establish data exchange, the first step is to identify the synchronization points for this DT. The decision regarding the necessary synchronization points is intrinsically tied to the defined DT objective which is herein the AMR dispatching. According to ISO 23247-4 the rate of synchronization depends on the application context. The update frequency must be aligned with the real-time data needs, ensuring accurate representation of the physical system. Lugaresi and Matta (2021) also highlight that the rationale for this decision is to reduce the frequency of updates in the RTS by eliminating the capture of redundant sensor data.

Based on this principle, we defined seven synchronization points for our DT. The first synchronization point is related to the MES to capture the arrival of customer orders in real-time. Then three synchronization points are associated to the RFID reader placed at the Exit of each Automated workstation. These points allow to capture when the workpiece finished each manufacturing step. Finally, the last three synchronization points are associated to the AMR arrival and departure from the pickup to the delivery manual workstations. For that purpose, the synchronization is conducted by having the AMR camera sensor scan the three QR codes placed respectively at the Exit of the MPS 403-1, at the Manual workstation 1 and the last one at the Manual workstation 2. The following UML sequence diagram, Figure 5, describes the information exchange between the virtual system and the OMEs within the DT framework.

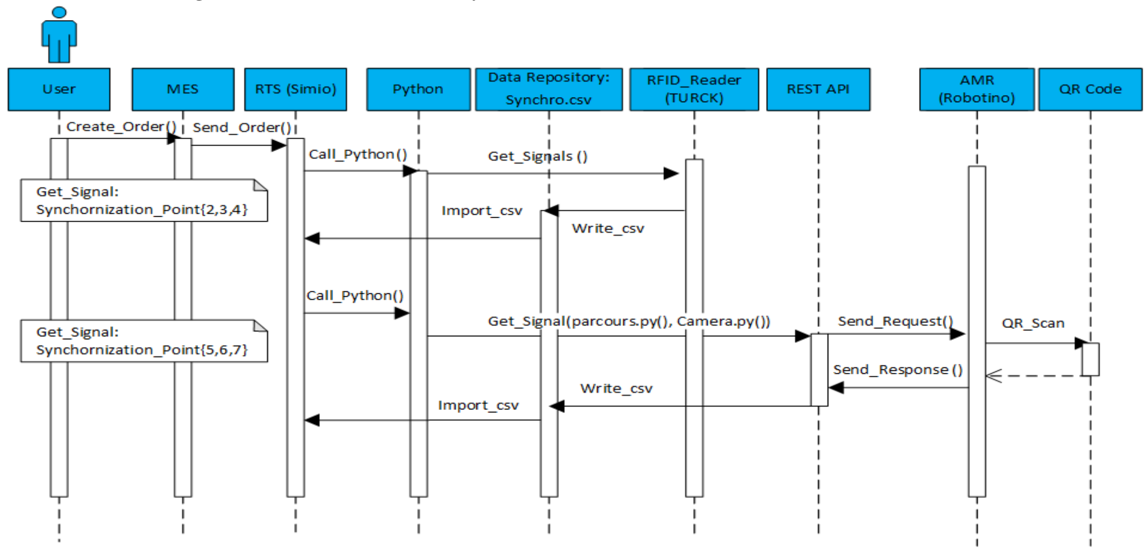


Figure 5: Sequence Diagram for data exchange within the DT.

The process, Figure 5, begins with an order triggered by a User via the MES which transmits this customer order to the first synchronization point placed in the Arrival module of the Simio model. At first, since the workpiece has to be produced at the MPS 403-1, its movement in the physical system will be synchronized in the digital model using the RFID sensors signals captured at the *Synchronization_points*: {2, 3, 4}. To execute the corresponding functions, given in Figure 5, Python scripts are requested via Simio ‘Call Python’ API. When the workpiece reaches the pickup zone its synchronization in the RTS will be tied to the AMR position, which is captured through QR code camera recognition related to *Synchronization_points*: {5, 6, 7}. The 'Parcours.py' script determines the path for workstation delivery, while 'Camera.py' is responsible for image scanning. As mentioned earlier, these scripts interface with the AMR through a REST API protocol. The ‘Synchro.csv’ file serves as a repository for all captured sensor data from the production system, continuously updating to reflect the real-time state of the sensors. Through this mechanism when the AMR is at *synchronization point* 5 i.e. able to pick up, the DT will use the real-time state and determine accordingly, from the pre-trained database, the best dispatching action. This control decision is applied concurrently in the RTS and transmitted to the AMR through the REST API protocol.

It is worth mentioning that our choice to limit tracking the AMR position to only three synchronization points—corresponding to the pickup and the two delivery stations—is in line with the principle of reducing the frequency of updates in the RTS to avoid latency. With this choice we did not track the instantaneous position of the AMR during travelling. In other types of problems, such as AMRs routing, a higher number of synchronization points would be necessary to avoid problems such as traffic congestion. The next section details implementation challenge of this data exchange mechanism in the DES model to transform it to an RTS.

4.2 Transitioning from Traditional DES to RTS

It is worth noting that the ISO 23247 standard does not offer detailed implementation guidance for using a specific simulation modeling approach. Therefore, the following work aims to clarify these implementation issues concerning the transformation of a DES model into an RTS.

4.2.1 Establishing Clock Synchronization

To ensure that the RTS advances continuously with the physical system, it is important to use world wall clock time instead of relying on the *Simulation clock* formerly used for the predictive DES model. In traditional DES model, the *Simulation clock* advances by jumping to the times of occurrence of future events scheduled in the *Event list* using *random variate*. But in the RTS these future events are not generated from probability distributions, but their occurrence is triggered from sensor data. Then the RTS must advance in real-time, matching the progression of the world clock. Although software programs allow adjustment of the running speed, this feature is insufficient to ensure continuous consistency between the physical and its digital counterparts.

Then to use the world wall clock time in the RTS, herein developed with Simio, we recourse to the feature *'EmulationSyncStep'*. Also, the following new value must be assigned to the *'DelaySeconds'* variable based on this expression: *[3600 * (TimeNow - DateTime.SystemNow)]*. This setting allows to run the DES model with real-world time. The following snapshot, Figure 6, illustrates how the current simulation time in the model aligns with the system clock of the computer.

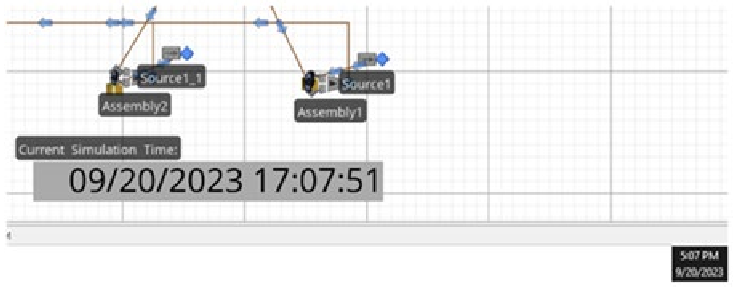


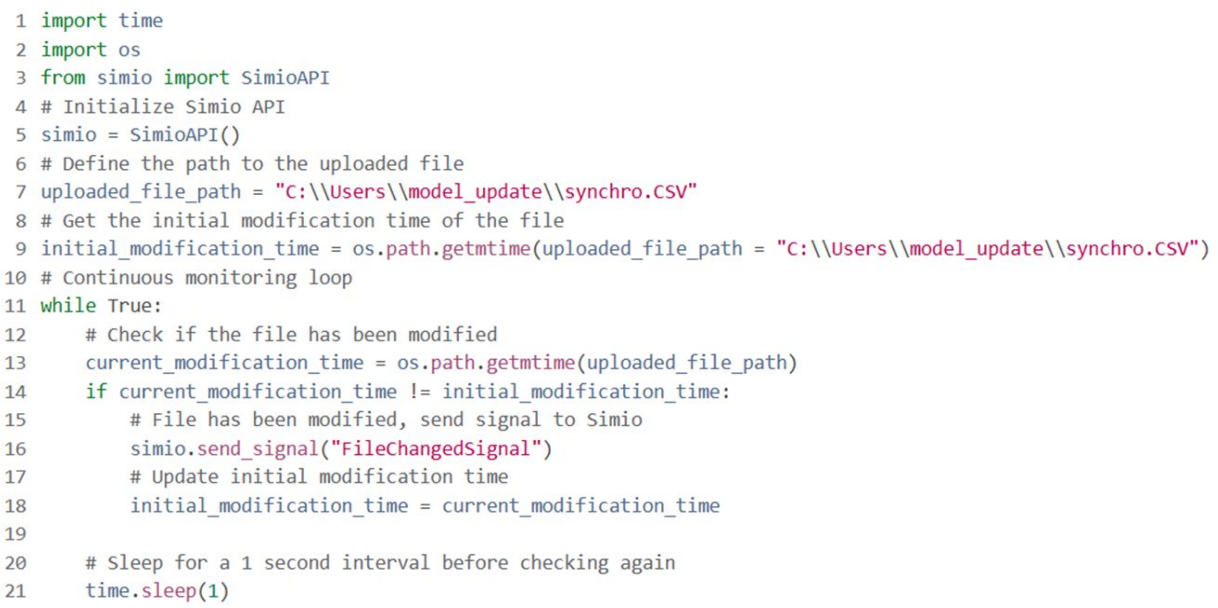
Figure 6: Clock time of the RTS model with world wall clock time.

Once this step is accomplished the next one consists of letting the RTS evolve according to the real occurrence of external events captured continuously at the synchronization points. To track these events occurrence from the physical system the following synchronization mechanism must be embedded in the model.

4.2.2 Embedding RTS Synchronization

The objective of the following mechanism is to poll the sensor data tracked at each synchronization point to capture in the RTS the occurrence of the corresponding events. These sensors data, from the seven synchronization points, are captured in the data repository, synchro.csv file, as depicted earlier by the sequence diagram in Figure 5. Under this mechanism, for example, if a workpiece reaches the Joining workstation, its replica in the RTS will be placed on hold, awaiting a release signal from the corresponding RFID sensor at synchronization point 3. This signal indicates that the workpiece has completed the Joining process and is entering the next Sorting workstation. Upon receiving this signal, the RTS releases the workpiece from the Joining process to advance it to the next workstation.

To embed this mechanism, we have first to do the binding process in Simio which ensures that the change in the data table will be considered in the RTS. This binding is insufficient because we require also continuous tracking of real-time changes in the physical system to update tables while the model is running. For this purpose, we developed a Python script to continuously monitor the synchro.csv file for updates and promptly send signals to Simio whenever changes occur. Upon receiving this signal, Simio imports the updated file, ensuring that real-time data is consistently integrated into the RTS. For illustrative purpose a simplified version of this script is given in Figure 7.



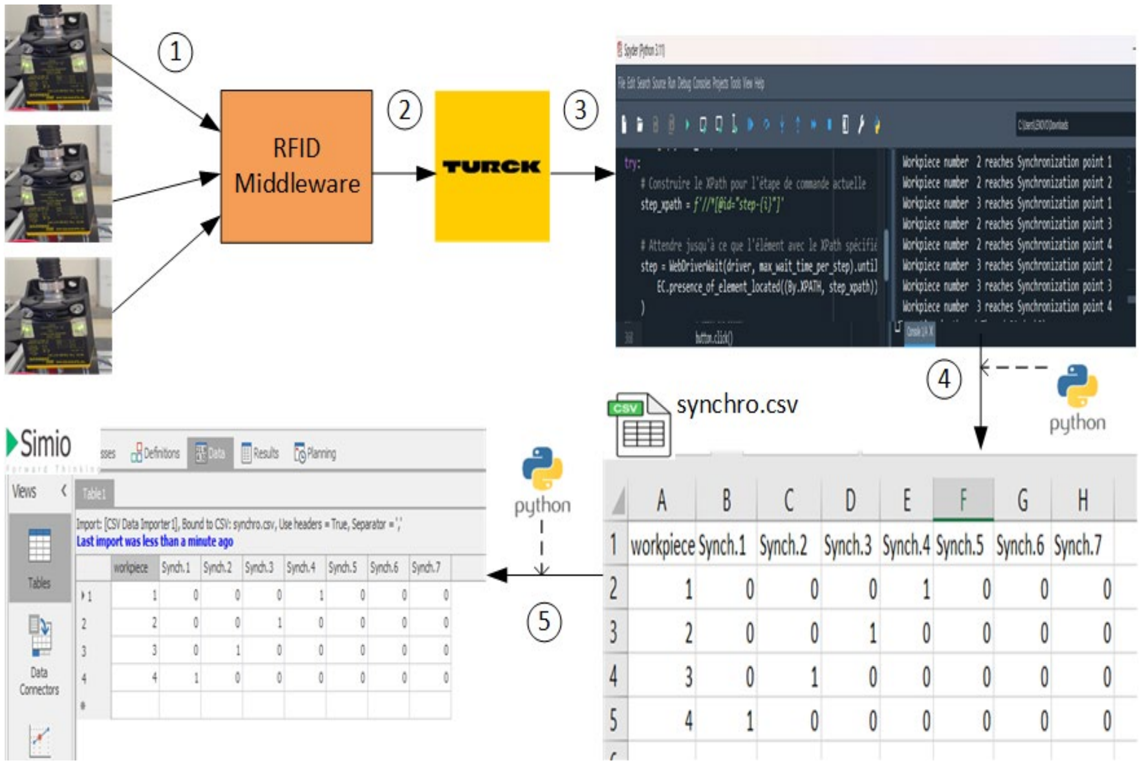
```
import time
import os
from simio import SimioAPI
# Initialize Simio API
simio = SimioAPI()
# Define the path to the uploaded file
uploaded_file_path = "C:\\Users\\model_update\\synchro.CSV"
# Get the initial modification time of the file
initial_modification_time = os.path.getmtime(uploaded_file_path = "C:\\Users\\model_update\\synchro.CSV")
# Continuous monitoring loop
while True:
    # Check if the file has been modified
    current_modification_time = os.path.getmtime(uploaded_file_path)
    if current_modification_time != initial_modification_time:
        # File has been modified, send signal to Simio
        simio.send_signal("FileChangedSignal")
        # Update initial modification time
        initial_modification_time = current_modification_time

    # Sleep for a 1 second interval before checking again
    time.sleep(1)
```

Figure 7: [Python Code] Script for continuous update check.

In the RTS, when an entity i.e. a workpiece is awaiting a synchronization signal, the process remains on hold, continually scanning the csv file until its corresponding position value shifts from '0' to '1'. As indicated, in the code, Figure 7, this update check is conducted regularly, every 1 second.

Within this mechanism, the RTS is synchronized with the real-time physical system state through its synchronization points. This constitutes the main difference from the classic DES model, in which events are typically predicted using *random variates* and scheduled in an *Event list.* In the RTS, events' occurrences are tied to the synchronization mechanism and are updated based on real-time sensor data received from the physical system. For further insights, Figure 8 illustrates an example of how the embedded mechanism continuously updates the synchro.csv file based on sensor data captured from the Turck's RFID system. This sensor data is integrated into the Simio model to achieve the RTS synchronization.



```
try:
    # Construire le XPath pour l'étape de commande actuelle
    step_xpath = f'//*[@id="step-{i}"]'

    # Attendre jusqu'à ce que l'élément avec le XPath spécifié
    step = WebDriverWait(driver, max_wait_time_per_step).until(
        EC.presence_of_element_located((By.XPATH, step_xpath))
    )
            button.click()
```

```
Workpiece number  2 reaches Synchronization point 1
Workpiece number  2 reaches Synchronization point 2
Workpiece number  3 reaches Synchronization point 1
Workpiece number  2 reaches Synchronization point 3
Workpiece number  2 reaches Synchronization point 4
Workpiece number  3 reaches Synchronization point 2
Workpiece number  3 reaches Synchronization point 3
Workpiece number  3 reaches Synchronization point 4
```

	A	B	C	D	E	F	G	H
1	workpiece	Synch.1	Synch.2	Synch.3	Synch.4	Synch.5	Synch.6	Synch.7
2	1	0	0	0	1	0	0	0
3	2	0	0	1	0	0	0	0
4	3	0	1	0	0	0	0	0
5	4	1	0	0	0	0	0	0

Import: [CSV Data Importer1], Bound to CSV: synchro.csv, Use headers = True, Separator = ';'
Last import was less than a minute ago

	workpiece	Synch.1	Synch.2	Synch.3	Synch.4	Synch.5	Synch.6	Synch.7
1	1	0	0	0	1	0	0	0
2	2	0	0	1	0	0	0	0
3	3	0	1	0	0	0	0	0
4	4	1	0	0	0	0	0	0
*								

Figure 8: An execution example of sensor data tracking and RTS update.

Figure 8 gives an overview on the five steps involved in the tracking and synchronization of the RTS towards an efficient DT. At step 3 it is possible to see the continuous evolution of each workpiece through the different synchronization point related to the RIFD placed at the exit of the three Automated workstations captured from the Turck's RFID system. As previously designed in the sequence diagram, Figure 5, this data is continuously written into the synchro.csv file tracking the trace of each workpiece over the seven fixed Synchronization point. For example, in Figure 8, the information provided for workpiece 2 indicates that it has currently reached synchronization point 3, which is associated with the exit from the Joining station. At that moment in the RTS model, when the corresponding position value switches from '0' to '1', workpiece 2 is released from the joining station and moves to the next workstation. This data structure allows for tracking the progress of each workpiece in the physical system, enabling straightforward data exchange within this DT application.

5 CONCLUSION

In this work, a DT application is developed for real-time dispatching decisions in a small-scale production system, following the design principles of ISO standards. Aligning with the reference architecture outlined in the ISO 23247 series, this work provides a methodological implementation through the formal design of the complex data exchange mechanism. The application benefited from the rigorous formalism of UML, greatly clarifying the sequencing involved in capturing and transmitting real-time sensor data for cyber-physical system synchronization. The present work also elucidated coding issues to transform traditional DES into a RTS for efficient DT deployment. In this DT architecture, the traditional DES model is used for offline training of RL optimization algorithms, while the RTS advances using the world wall clock time to maintain temporal consistency with the physical system. Unlike the DES model, in the RTS, events are not predicted using *random variates* and scheduled in an *Event list*. Instead, they are tied to the synchronization mechanism and updated based on real-time sensor data received from the physical system.

The main limitation of the proposed DT lies in its scalability to larger production systems involving a higher number of synchronization points. Latency issues may arise, necessitating further investigation in this area, particularly from DES simulation software providers. Another important aspect to explore is conducting online validation of the RTS. Lugaresi et al. (2023) have highlighted the importance of DT validation and proposed novel methods for conducting it. The future work of this research will focus on this phase of online validation.

ACKNOWLEDGMENTS

Authors are grateful to Simio for providing the discrete-event simulation software under a grant from Simio LLC to the National Engineering School of Tunis, Tunisia.

REFERENCES

AboElHassan, A., A. H. Sakr, and S. Yacout. 2023. “General Purpose Digital Twin Framework Using Digital Shadow and Distributed System Concepts”. *Computers & Industrial Engineering* 183:109534.

Barbieri, G., A. Bertuzzi, A. Capriotti, L. Ragazzini, D. Gutierrez, E. Negri *et al.* 2021. “A Virtual Commissioning Based Methodology to Integrate Digital Twins into Manufacturing Systems”. *Production Engineering* 15(3):397-412.

Cimino, C., E. Negri, and L. Fumagalli. 2019. “Review of Digital Twin Applications in Manufacturing”. *Computers in Industry* 113:103130.

Devanga, A., E. D. Badilla, and M. Dehghanimohammadabadi. 2022. “Applied Reinforcement Learning for Decision Making in Industrial Simulation Environments”. In *2022 Winter Simulation Conference (WSC), 2819-2829* https://doi.org/10.1109/WSC57314.2022.10015282.

Ferko, E., A. Bucaioni, P. Pelliccione, and M. Behnam. 2023. “Standardisation in Digital Twin Architectures in Manufacturing”. In *2023 IEEE 20th International Conference on Software Architecture (ICSA)*, 70-81. Institute of Electrical and Electronics Engineers, Inc.

Fett, M., F. Wilking, S. Goetz, E. Kirchner, and S. Wartzack. 2023. "A Literature Review on the Development and Creation of Digital Twins, Cyber-Physical Systems, and Product-Service Systems." *Sensors* 23(24):9786.

Lugaresi, G., and A. Matta. *2018*. “Real-Time Simulation in Manufacturing Systems: Challenge and Research Directions” .In *2018 Winter Simulation Conference (WSC), 3319-3330* https://doi.org/10.1109/WSC.2018.8632542.

IEC. 2013. IEC 62264-2: Enterprise-control System Integration – Part 2: Object and attributes for enterprise-control system integration

ISO. 2020a. “ISO (DIS) 23247-1: Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 1: Overview and general principles”. ISO/TC 184/SC4/WG15.

ISO. 2020b. “ISO (DIS) 23247-2: Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 2: Reference Architecture”. ISO/TC 184/SC4/WG15.

ISO. 2020c. “ISO (DIS) 23247-3: Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 3: Digital Representation”. ISO/TC 184/SC4/WG15.

ISO. 2020d. “ISO (DIS) 23247-4: Automation Systems and Integration - Digital Twin Framework for Manufacturing - Part 4: Information Exchange”. ISO/TC 184/SC4/WG15.

Jaoua, A., S. Masmoudi, and E. Negri. 2024. “Digital Twin-Based Reinforcement Learning Framework: Application to Autonomous Mobile Robot Dispatching”. *International Journal of Computer Integrated Manufacturing* 1-24.

Lim, K. Y. H., P. Zheng, and C. H. Chen. 2020. “A State-of-the-Art Survey of Digital Twin: Techniques, Engineering Product Lifecycle Management and Business Innovation Perspectives”. *Journal of Intelligent Manufacturing* 31(6):1313-1337.

Lugaresi, G., and A. Matta. 2021. “Automated Manufacturing System Discovery and Digital Twin Generation”. *Journal of Manufacturing Systems* 59(2021):51-66.

Lugaresi, G., Gangemi, S., Gazzoni, G., & Matta, A. 2023. “Online Validation of Digital Twins for Manufacturing Systems”. *Computers in Industry* 150(2023):103942.

Marah, H., and M. Challenger. 2023. “Madtwin: A Framework for Multi-Agent Digital Twin Development: Smart Warehouse Case Study”. *Annals of Mathematics and Artificial Intelligence* :1-31.

Matta, A., and G. Lugaresi. 2023. “Digital Twins: Features, Models, And Services”. In *2023 Winter Simulation Conference (WSC), 46-60* https://doi.org/10.1109/WSC60868.2023.10407260.

Miles, R., and K. Hamilton. 2006. *Learning UML 2.0*. New York: O'Reilly Media, Inc.

Negri, E., L. Fumagalli, and M. Macchi. 2017. “A Review of the Roles of Digital Twin in CPS-Based Production Systems”. *Procedia Manufacturing* 11(2017):939-948.

Pystina, X., L. Gzara, V. Cheutet, and A. Sekhari. 2022. “A Conceptual Framework Elements for Digital Twin Deployment in Production Systems Domain”. In *12th International Conference on Information Society and Technology*, 39–44. Kopaonik, Croatia: Institute of Electrical and Electronics Engineers, Inc.

Smith, J. S., and D. T. Sturrock. 2023. “Simio and Simulation: Modeling, Analysis, Applications”. 6th ed. Pittsburgh: Simio LLC.

Vasilev, P., and E. Koleva. 2023. “IEC 62264 Standard-Based Manufacturing Operations Management Resource Modelling for Electron Beam Welding”. *Journal of Physics: Conference Series* 2443(1):012011.

Wang, K., Y. Wang, Y. Li, X. Fan, S. Xiao, and L. Hu. 2022. “A Review of the Technology Standards for Enabling Digital Twin”. *Digital Twin* 2:4.

Wooley, A., D. F. Silva, and J. Bitencourt. 2023. “When Is a Simulation a Digital Twin? A Systematic Literature Review”. *Manufacturing Letters* 35(2023):940-951

AUTHOR BIOGRAPHIES

HEDIR OUKASSI has obtained a Bachelor’s degree in Industrial Engineering from the National Engineering School of Tunis, Tunisia. Now pursuing her Master of Science in Industrial Engineering at Polytechnique Montréal, Canada. Her research interests include Digital Twin, simulation and optimization modeling, Artificial Intelligence applied to Smart manufacturing. Her email address is hedir-2.oukassi@polymtl.ca.

AMEL JAOUA is Professor of Industrial Engineering in National Engineering School of Tunis, Tunisia. She received her PhD in Industrial Engineering at Polytechnique Montréal, Canada. She co-authored 30+ scientific publications and she is Expert in developing cutting-edge curriculum in Industrial Engineering for the emerging era of interconnected industries. She is Director of the Industry 4.0 laboratory for the Next Production Revolution Erasmus project. Her research interests include Smart Digital Twins, Reinforcement Learning, Simulation-based Optimization, and Cyber-Physical Systems implementation. Her email address is amel.jaoua@enit.utm.tn.

SOUMAYA YACOUT is a full Professor in the Department of Mathematics and Industrial Engineering at Polytechnique Montreal in Canada since 1999. She is the Scientific Director of the FCI-CÉOSnet network, Director of the Physical Asset Integrity Management (PAIM) laboratory, Director of the Intelligent Cyber-Physical Systems laboratory, member of the Institut de Valorisation des Données (IVADO), of Industry 4.0 Network, and of Industry 4.0 laboratory She is also the founder, President and CEO of DEXIN Inc., an enterprise dedicated in offering state of the art technologies for data-driven solutions to help companies in achieving the highest level of value added performance by keeping their physical assets in good health. Her email address is soumaya.yacout@polymtl.ca.

ELISA NEGRI is Professor of Industrial Plants Management and of Modeling & Data Analysis for Complex Systems at Politecnico di Milano. She is Scientific Coordinator of the Extended Partnership “Made in Italy Circular and Sustainable”. She researches on digital twins for production planning and control in the manufacturing sector. She co-authored 50+ scientific publications. She is active in international scientific communities, being Vice-chair for Social Media in IFAC TC5.1 Manufacturing Plant Control, Associate Editor in Flexible Services & Manufacturing and Track Chair in ANNSIM2024. Her email address is elisa.negri@polimi.it.

MEHDI JAOUA is a research consultant in Simulation Modelling and Optimization of complex Systems. He holds a Bachelor's degree in Computer Science and a Research Master’s degree in Mathematics and Industrial Engineering, with a specialization in Operations Research and Optimization, from Polytechnique Montréal, Canada. He has conducted many projects in emulation modeling, and virtual reality test benches for both training and virtual commissioning in manufacturing and warehouse industries. His email address is mehdi.jaoua@polymtl.ca.