Journal of Manufacturing Systems 77 (2024) 903–918

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

journal homepage: www.elsevier.com/locate/jmansys

Simulation-based Digital Twin for enhancing human-robot collaboration in assembly systems

Antonio Cimino [a], Francesco Longo [b,c,*], Letizia Nicoletti [c], Vittorio Solina [b,c]

[a] Department of Innovation Engineering, University of Salento, Italy
[b] Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy
[c] CAL-TEK S.r.l., Italy

ARTICLE INFO

Keywords:
Digital Twin
Human-Robot Collaboration
Assembly Systems
Automotive Industry
Interoperability

ABSTRACT

The advent of new technologies and paradigms such as the Internet of Things (IoTs), Digital Twin (DT), Human-Robot Collaboration (HRC), is offering immense opportunities to improve the performance of manufacturing systems, but also opening new challenges. The current scientific literature highlights the presence of numerous theoretical studies, but limited real-life applications, and the need to address interoperability issues, with the aim of valorizing the data continuously generated by humans, robots, machines. This research presents a novel simulation-based DT, designed for supporting HRC optimization in assembly systems. The proposed approach is tested and validated, through a case study in the automotive sector, specifically focusing on an assembly line for car front doors. The results show that it is possible to achieve HRC improvements through the assessment of different working configurations. Furthermore, it is explained how the simulation-based DT, by leveraging the FIWARE/FIROS paradigm, can effectively and efficiently interact with other systems, to enable real-time data exchange, which is nowadays one of the main open research challenges.

1. Introduction

The Fourth Industrial Revolution, often referred to as Industry 4.0 (I4.0), marks a profound shift in the manufacturing landscape, characterized by the rapid emergence of novel digital technologies, the widespread adoption of automation, and the pervasive use of data-driven methodologies [1–4]. Key paradigms such as Digital Twin (DT) [5], Extended Reality (XR) [6], Internet of Things (IoTs) [7], Human-Robot Collaboration (HRC) [8] are gradually gaining importance in ensuring the competitiveness and sustainability of companies [9]. The twin concept dates back to the 1960s when the National Aeronautics and Space Administration (NASA) constructed twin space vehicles to replicate the conditions of one in flight with remarkable accuracy [10]. Today, DT finds widespread applications across various sectors [11], particularly in manufacturing, where it facilitates efficient product design, aids production planning, optimizes human resource utilization, and guides maintenance activities [12,13]. Concurrently, robotics is revolutionizing operations in technologically advanced firms, enhancing production safety and efficiency, reducing human errors, and increasing product quality [14]. Consequently, it is increasingly common to find environments in manufacturing where humans and robots coexist harmoniously.

While I4.0 technologies offer significant opportunities, they also present new challenges. DT requires real-time data exchange between physical and digital worlds to accurately mirror reality [15]. Thus, ensuring data interoperability emerges as a critical imperative in this era [16]. To address this, the concept of an orchestrator has recently gained attention, serving as a centralized entity coordinating and managing different components or systems to achieve specific objectives [17,18]. Similarly, while robots can augment human tasks in the workplace, managing such collaborations present significant challenges, influenced by economic, security, and social factors [19,20].

The main purpose of this paper is to propose, test and validate a simulation-based DT, for enhancing the HRC in an assembly system. It is also explained how a DT can successfully interact with a set of other entities, including an orchestrator, through the FIWARE/FIROS standards. A real-life case study in the automotive industry demonstrates the effectiveness of the proposed solution. The rest of this paper is structured as follows. Section 2 reviews the current scientific literature regarding DTs and HRC, with a focus on the manufacturing field. Crucial research gaps are identified and described to clarify the innovative contribution. Section 3 provides relevant information on the research framework that

---

* Corresponding author at: Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy.
E-mail address: francesco.longo@unical.it (F. Longo).

https://doi.org/10.1016/j.jmsy.2024.10.024
Received 29 February 2024; Received in revised form 9 September 2024; Accepted 25 October 2024

this study if part of. Section 4 describes in detail the simulation-based DT model, while Section 5 explains its testing and validation through a case study. Conclusions are in Section 6.

2. Literature review

DTs and HRC represent two highly emerging and interesting concepts, falling within the I4.0 umbrella [21–23]. Subsections 2.1 and 2.2 explore the current state-of-the-art regarding these two paradigms, respectively. Subsection 2.3 highlights the research gaps identified and the innovative contribution of this paper.

2.1. Digital Twins in manufacturing

The concept of DT is based on the creation of a digital replica of a physical object, process, or system. Such a digital replica is powered by real-time data coming from the corresponding physical object and can be used to monitor, analyze, and optimize its functioning [5]. With the advent of IoT, advanced sensors, cloud computing and artificial intelligence, the DT has gained increasing interest in the scientific field. Today, DTs are used across various industries, including manufacturing, energy, healthcare, construction, agriculture, and many others [5,24,25]. DTs can monitor the status and operation of machinery and equipment, optimize production processes, simulate different scenarios and predict future behavior. These capabilities enable companies to make smart decisions and improve the overall performance of their systems and/or processes. Within DT-driven approaches, the use of simulation-based DT, which replicate real systems using the Modeling & Simulation (M&S) paradigm, is gaining significant relevance [26]. Basically, through simulation, it is possible to improve the performance of the corresponding physical asset, by exploring various what-if scenarios. Below, the main recent scientific contributions in terms of adoption of DTs in the manufacturing sector are reviewed. The focus is on two main concepts aligned with the scope of this paper: (i) the use of M&S to enable DTs, and (ii) DT applications in assembly systems. Židek et al. [27] have recently proposed a DT of an experimental assembly system, created as a 3D model through a CAD design software and then imported it into a Tecnomatix platform, in order to simulate and optimize online all the manufacturing processes involved. Sun et al. [28] address the issue of assembly process of high precision products. Specifically, a DT-driven approach is proposed, to improve overall assembly efficiency and quality consistency, which are traditionally quite limited, considered the manual nature of the operations. An important research effort was recently made by Yi et al. [29], who proposed a digital twin reference model for smart assembly process design and an application framework, based on three layers (i.e., physical space layer, interaction layer, virtual space layer), successfully tested in a simplified satellite case study. Similarly, Roque-Rolo et al. [26] proposed a simulation-based DT framework consisting of three key layers: simulation, integration, and distributed control. They demonstrated that such an approach can be extremely useful for predicting the behavior and performance of a distributed production control system. According to Coelho et al. [30], simulation-based decision support tools, which are representative of reality, can be used as digital-twinning for operations improvement. Korth et al. [31] propose a DT for real-time management and highlight the importance of combining discrete-event simulation for performance enhancement. For further information on the current role of DT in manufacturing systems, the reader is referred to some recent and comprehensive literature reviews [32–34]. Although the concept of DT is quite widespread in the manufacturing sector, the study of the scientific literature reveals some important research gaps. According to Soori et al. [35], further investigation is required to enhance real-time data exchange and synchronization between physical assets and their corresponding DTs, with the aim to ensure accurate representation and timely decision-making. Furthermore, Kritzinger et al. [12] argue that there is currently a lack of real-life case studies, as most research focuses only theoretical concepts.

2.2. Human-robot collaboration: Focus on assembly manufacturing systems

Human-robot collaboration refers to the cooperative interaction between humans and robots to carry out tasks or solve problems. In these collaborations, humans and robots work synergistically, combining their respective strengths to improve overall performance and productivity. This concept is based on the idea that robots complement human abilities rather than replacing them. It recognizes that humans bring cognitive skills, creativity, adaptability, and emotional intelligence, while robots offer precision, strength, speed, and endurance in tasks that may be repetitive, dangerous, or alienating for humans [36,37]. The HRC paradigm appears extremely promising today with applications in various fields. The most significant and recent scientific contributions are reviewed below, with a focus on assembly systems in manufacturing industry. Cherubini et al. [38] have proposed a human-robot manufacturing cell for homokinetic joint assembly, with the aim of reducing the human workload and lowering the risk of strain injuries. Michalos et al. [39] presented the implementation of a system for an advanced HRC assembly, where each task is assigned to human and robot, based on their capabilities. Using Augmented Reality (AR) glasses and smartwatches, human-robot interaction is facilitated, in a very safe environment, validated through a real-life case study in the automotive industry. In the context of HRC, a key challenge is task allocation in terms of determining which tasks should be assigned to humans and which to robots. Malik and Bilberg [40] proposed a methodology for task distribution between humans and robots, based on their complexity, with the aim of balancing the overall workload. The approach proposed by Ranz et al. [41] is instead based on the capabilities of human and robot, and specifically on a two-staged decision-making process: (i) assignment of tasks based on the unique capabilities of humans and robots; (ii) distribution of the remaining tasks, based on the respective capabilities and the impacts on cost, time, quality. Tsarouchi et al. [42] proposed and tested in an automotive industry, an intelligent decision-making method for the allocation of sequential tasks, with the aim of increasing the level of automation in an environment where human can interact with robot through body gestures. For a more comprehensive analysis of the current state of the art relating to HRC with a focus on manufacturing, the reader is addressed to some recent and comprehensive literature reviews [43,44]. Despite the numerous scientific contributions related to HRC, significant research efforts are still needed to achieve large-scale acceptance of this revolutionary paradigm. According to Arents et al. [45], the sustainability of a system based on HRC is guaranteed by the efficient interconnectivity of the system itself with other elements within the smart factory, which is still an open issue. Simões et al. [44] argue that under the manufacturing companies' perspective, future work should focus more on tools capable of supporting decision-makers in the implementation of an HRC that jointly guarantees workers physical and mental well-being, performance, and productivity.

Regarding the use of DTs to support HRC, it is important to underline that this is a quite new research topic. The joint use of the keywords "Digital Twin" and "Human Robot Collaboration" within the scientific database Scopus returns a very limited number of documents, which were only published from 2017 onwards. To date, a significant research branch focuses primarily on using DTs to support safe and efficient HRC-oriented layouts [46,47]. According to Malik and Brem [48], DT technology can be used to digitally perform human tasks and then evaluate biomechanical loads, stress levels and fatigue under different scenarios. This provides crucial insights for solving the human-robot task allocation problem, with the aim to improve performance and human well-being. DT-HRC coupling also enables intelligent motion planning of human and robot ensuring workplace safety [22,49]. Basically, risk levels regarding multiple working configurations can be effectively

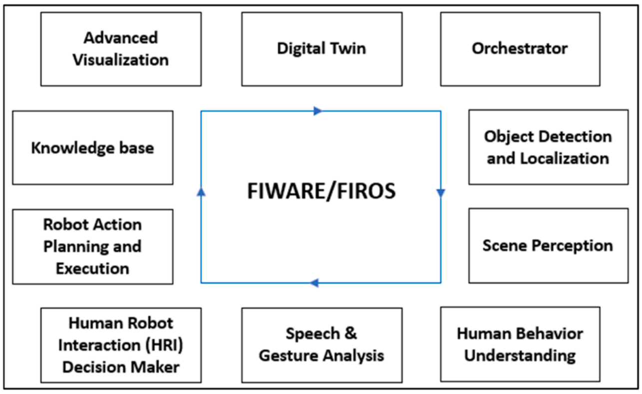


Fig. 1. High-level visualization of FELICE architecture.

assessed for making smart decisions [50,51]. Zhu et al. [52] recently demonstrated the potential of DT in the reconfiguration of HRC manufacturing systems, using optimization models supported by novel algorithms. Furthermore, by using DT, the monitoring and control of HRC in assembly processes can be achieved through the continuous collection of real-time data from the field. Therefore, in case of issues, countermeasures can be taken effectively and efficiently [22]. Overall, it can be stated that the use of DT technology to replicate HRC in a virtual environment represents today a research area of significant scientific interest, which can lead to countless benefits not only from the point of view of the performance of manufacturing systems, but also to improve the health of the human operator [53–55].

2.3. Research gaps and our contribution

Based on the literature review, several significant research gaps have been identified:

- Although the concept of DT is quite common today, there is still a lack of real case studies, as much research is only theoretical [12]. A significant effort is needed to show that this technology is feasible in reality, in order to push its large-scale diffusion. Indeed, Shao and Helu [56] argue that there is still a lot of confusion about how to implement a DT in a real manufacturing system.
- The real-time data exchange between the real and digital world is still an open issue due to interoperability issues [35]. Standardization efforts are necessary to enable the effective and efficient exchange of data between different systems, in order to enable timely and intelligent decision-making, which is crucial in manufacturing ecosystems. Solving interconnectivity issues is critical also for an effective HRC [45].
- Simões et al. [44] argue that future work should focus more on tools capable of supporting decision-makers in the implementation of an HRC that jointly guarantees workers physical and mental well-being, performance, and productivity.
- According to Soori et al. [35], research is needed to explore ways to integrate DT technology with human-machine interfaces to improve collaboration and decision-making.

Our innovative contribution can be summarized as follows:

- Design of a simulation-based DT, capable of investigating and enhancing HRC in the context of assembly systems, with the aim to address the current needs for integration of DTs with human-machine interfaces.
- Testing and validation of the proposed approach through a case-study, belonging to the automotive sector, bridging the current gap regarding the lack of real implementations.
- Development and validation of an innovative way to enable real-time data exchange between a DT and any external system (e.g., an orchestrator). This method addresses current interoperability issues and provides the reader with useful information to replicate it in other production contexts.

3. Research framework

This research is part of the European project “FlExible assembly with human-robot Collaboration and digital twin modEls” (FELICE), funded by the European Union’s Horizon 2020 Research and Innovation Programme. The project integrates multidisciplinary research in robotics, Artificial Intelligence (AI), computer vision, data analytics, process optimization, and ergonomics, aiming to develop a modular platform that integrates and harmonizes an array of autonomous and cognitive technologies. The primary goal is to increase the agility, productivity, safety, and well-being of human workers within an assembly production system. Specifically, by combining human and robot skills, the project aims to improve manufacturing performance and work ergonomics while establishing a safe, mentally satisfying environment for HRC. The project’s main pilot features an assembly line for automobile front doors, consisting of three workstations where a set of assembly operations are performed. Each workstation is staffed with an operator, and a robot is available to assist and facilitate operations along the assembly line. The project operates on two distinct levels: a local level, which pertains to the actual assembly line in the physical environment, and a global level, which interacts with the real world through an actionable digital replica (Digital Twin - DT) of the entire physical assembly line. This paper focuses on the DT module, and more specifically on one of its key components, the Discrete Event Simulation (DES) module. The study begins by providing an overview of the project, including its modules and their integration (Section 3.1), with the aim to increase the reader’s understanding. Following this, Section 3.2 presents the DT module, while Section 4 focuses on the development and implementation of the DES module.

3.1. The FELICE project

FELICE project aims to increase the agility and productivity of an assembly production system and also ensure the safety and physical and mental well-being of human workers. To achieve these goals, 10 specialized modules have been developed with the aim of trying to combine the accuracy and endurance of robots with the cognitive ability and flexibility of humans. Fig. 1 provides a high-level visualization of the FELICE architecture, while Table 1 briefly summarizes the main functionalities of each of the module.

The integration of these modules, coupled with their dynamic data exchange and interaction, lays the groundwork for accomplishing the objectives of this research. Although this aspect is beyond the scope of the current research, the authors consider it relevant to offer a brief description of the integration approach used to implement and structure the platform. To address this effectively, the platform utilizes FIWARE, an open-source framework for IoT platforms (further information on FIWARE can be found out at https://www.fiware.org/), along with FIROS, a software tool for connecting Robot Operating System (ROS) (further information on FIROS can be found at https://firos.readthedocs.io/en/latest/index.html). While FIWARE provides a foundation for interoperability, modern robotics software often relies on ROS for communication between modules. To bridge ROS and FIWARE, FIROS is employed, serving as a connection tool with ROS-based robots. Specifically, FIWARE/FIROS enable data exchange between modules by receiving, storing, and transmitting data through a publish/subscribe mechanism [3]. This mechanism relies on a set of defined data models that standardize data formats and semantics across the platform’s software applications for each module. Essentially, a data model defines all entities (messages) that can be published and/or subscribed by the different modules within FIWARE/FIROS, serving as a common language for ensuring both syntactic and semantic interoperability. Each module has its data model presented in a table format, as reported in

Table 1
Platform modules functionalities.

Module Name	Functionalities description
Digital Twin	This module focuses on the development of the virtual representation of the real assembly line to enable the management of operating conditions, the simulation of the assembly process and the optimization of various aspects of its performance. It has been specifically developed for conducting what-if analyses, providing real-time decision support, and facilitating control and monitoring of the assembly processes.
Orchestrator	This module automatically optimizes assembly task distribution between human workers and robots by considering workstation characteristics, such as minimizing fatigue and respecting time limits. The module instructs robots to perform tasks at predefined workstations and continuously monitors the production line to ensure tasks are implemented on time. Additionally, it dynamically reallocates robot assistance to different workstations as needed, based on worker requests or predefined criteria, such as providing support when explicitly requested by a worker or deciding whether a worker should take a break or be assisted by a robot. The decisions made by the orchestrator module are guided by Key Performance Indicators (KPIs) calculated through the DT module.
Object Detection and Localization	This module focuses on identifying and estimating the position and orientation of significant objects, such as tools and parts for assembly. This data enables the robot to effectively grasp, pick up, and deliver these objects to human workers in accordance with the assembly sequence.
Scene Perception	This module is dedicated to constructing a spatial map of the environment to ensure safe movement for the robot within and between workstations. Utilizing its onboard sensors, the robot aggregates multiple observations to create a map of its surroundings, even in the presence of independently moving objects.
Human Behavior Understanding	This module is designed to detect human presence, estimate body posture, interpret actions, and characterize behavior by comparing observed actions with expected behavior. This task is crucial for ensuring safety by detecting human presence and location. Understanding actions and intentions is essential for monitoring assembly task progress and worker state including its ergonomics.
Speech & Gesture Analysis	This module focuses on enhancing communication between humans and the environment, including robots and workstations. Regarding speech, the module develops algorithms for command-based interaction and commands are employed either to assign task to robot or ask for their assistance. Concerning gesture recognition, predefined sets of gestures are defined for communication (body language) with the robot.
Human Robot Interaction (HRI) Decision Maker	This module essentially translates all vocal or gestural commands, as well as commands given by the orchestrator, into a language understandable to the robot.
Robot Action Planning and Execution	This module, informed by the data provided by the HRI Decision Maker, empowers the robot to carry out received commands including movements, operator assistance, and task execution.
Knowledge base	This module serves as a secure and accessible repository for consolidating, mining, and managing various types of information generated by all the platform modules. These include information such as the robot and objects positions, worker's body postures, orchestrator commands, etc. Basically, this module facilitates storage and retrieval of supporting operation and decision-making processes.
Advanced Visualization	This module presents information related to the assembly line data in terms of a predefined set of KPIs in a visually accessible format, assisting end-users in making informed decisions regarding process optimization, resource allocation, and other relevant factors

Table 2. To illustrate this concept further, Section 5 reports an extract of the data model of the DES module, which was specifically defined for implementing the case study investigated in this research.

3.2. The Digital Twin module

The DT module is a virtual representation of the physical assembly line which enables the management of operating conditions, the simulation of the assembly process and the optimization of performance indicators. As part of a larger platform integrating multiple modules, the DT module can be fed with real-time data necessary to accurately represent any physical assembly line, including its operators, robots, and operations. This capability enables conducting what-if analyses for optimizing assembly operations, providing real-time (online) decision support, and facilitating control and monitoring of the assembly processes. Fig. 2 provides a visualization of the DT module main components.

The DT module consists of two main components: the Digital Model and the Digital Mirror. Specifically, the Digital Model is further divided into two modules: the DES module and Real-Time Virtual Simulation (RTVS) module. The DES module allows to conduct fast-time simulations (discrete event simulation) of the assembly line to support what-if analyses. Its description is presented in detail in Section 4.

The RTVS module facilitates the evaluation of decision effectiveness and outcomes in assembly line management through a real-time virtual simulation. It presents a 3D virtual representation of all assembly line operations, including tasks executed by operators and robots. Users can access this simulation to assess the impact of decisions made, including task allocation between humans and robots based on predefined KPIs. These KPIs measure both workstation productivity and operator ergonomics, enabling the analysis of decision impacts.

As concern the digital mirror, its Interactive Virtual Reality Environment (IVRE) module offers a dynamic, real-time 3D representation that mirrors the actual assembly line, providing an advanced tool for monitoring operations. Unlike the RTVS module, which provides a real-time virtual simulation but does not replicate the actual occurrences on the assembly line, the IVRE module faithfully reproduces the current events in the assembly line virtually, providing an accurate representation of currently on-going real-time operations. The IVRE module features a dashboard showcasing the current status of each twinned entity, including human operators and robots, alongside the progress of task completion time and error count for each worker, with or without robot support. The module's objective is to develop a tool that, through the evaluation of predefined KPIs, enables supervision, control, and monitoring of assembly operations. If decision-makers find KPI values unsatisfactory, they can promptly implement decisions in real-time within the assembly line and improve its overall performance.

Both the Digital Model (DES and RTVS sub-modules) and the Digital Mirror (IVRE module) are accessible either manually by the end-user or automatically by the orchestrator module of the FELICE platform. It is important to note that the primary focus of this research is on the development, implementation and utilization by end user of the DES module, while the other platform modules fall outside the scope of this research.

This section concludes by presenting Table 3, which summarizes the main features and objectives of the components within the DT module. This table provides readers with a concise overview, enabling quick comparison of the various modules at a glance.

Table 2
Data model table format.

Module Name	Sub-Module Name (optional)	Entity Name	Entity Type	Attribute Name	Attribute Type	Short Description
It defines the name of the platform module (e.g. Digital Twin, Orchestrator, Scene Perception, etc.). It is important to note that a single module may encompass multiple sub-modules	It defines the name of a sub-module. For example, the DT module has three sub-modules (DES, RTVS and IVRE)	It defines the name of the entity containing the message that a module or a submodule is going to exchange with FIWARE/ FIROS and in turn with other platform modules	It defines the type of the entity being exchanged with FIWARE/ FIROS	It defines the attribute name that is part of the entity being exchanged with FIWARE/ FIROS	It defines the attribute type that is part of the entity being exchanged with FIWARE/ FIROS (e.g., double, integer, etc.)	It reports a short description explaining the meaning of the message being exchanged with FIWARE/ FIROS

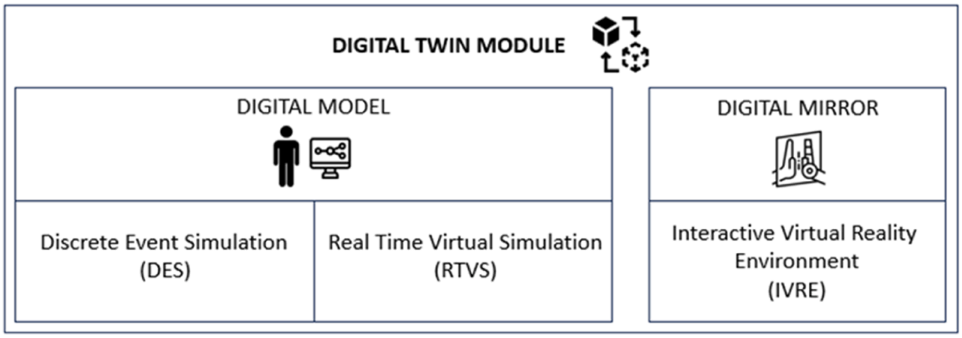


Fig. 2. DT module main components.

Table 3
DT Module components: features and objectives.

Platform Module	Module Component	Component sub-module	Feature	Objective
Digital Twin	Digital Model	DES	Conducts fast-time simulations of the assembly line	Supports what-if analysis scenarios for assembly line optimization
		RTVS	Conducts real-time virtual simulation	Evaluates decision effectiveness and outcomes in assembly line management
	Digital Mirror	IVRE	Reproduces faithfully and virtually real-time events of the assembly line	Supports monitoring and control of on-going operations

4. The DES module

The DES module is a simulation-based digital twin model used to simulate the assembly line based on inputs from the end-user or the orchestrator module through FIWARE/FIROS, supporting what-if analyses. Essentially, various assembly line configurations can be tested to assess the effects of different inputs, including task allocation between operators and robots and task execution via active collaboration between human operators and robots. The evaluation is based on a set of predefined KPIs related to assembly line productivity and the ergonomics of human operators. Followings, Section 4.1 presents the requirements and main features of the DES module. Subsequently, Section 4.2 provides an overview of the DES module’s data exchange capabilities and presents a solution for addressing interoperability issues. Section 4.3 summarizes the data framework used to develop the DES module, while Section 4.4 outlines the DES architecture and key implementation steps.

Table 4
DES module requirements (DesReq).

Requir. ID	Requir. description
DesReq1	The DES module shall enable to perform online simulation experiments
DesReq2	The DES module shall be used by the orchestrator and human user via a list of interfaces/APIs to interact with it and make tests in a synthetic scenario about the choices of the orchestrator and human users (e.g. effectiveness of the subdivision of the task between operators and robot).
DesReq3	The DES module shall include the digital model of the robot
DesReq4	The DES module shall include the digital model of each human worker
DesReq5	The DES module shall have the capability to carry out a fast time simulation (discrete event simulation) of the assembly line for a given time span (e.g. 8 h ahead) to check some decisions taken by the orchestrator and human user
DesReq6	The DES module shall be based on a discrete-event and agent-based simulation model of the assembly line capable also of receiving real-time data form the physical system
DesReq7	The DES module shall be used in automatic mode (robot re-allocation options are tested by the orchestrator) or manual mode (robot re-allocation options are tested by a human user)
DesReq8	The DES module shall provide interfaces to include 3D models
DesReq9	The DES module shall only contain 3D models of a maximum size in order to not cause performance issues in other simulation environments
DesReq10	The DES module shall have interfaces with FIWARE/FIROS
DesReq11	The DES shall provide an interface to submit actions by both an automated and manual decision-maker (i.e. orchestrator module and human user)

4.1. *DES requirements and features*

The authors based the identification of the DES module requirements on the following sources: (1) the overall technical research project proposal, (2) feedback and insights from potential end-users, (3) suggestions and ideas from the research project consortium members, and (4) analysis of relevant scientific literature in this field. Table 4 presents the 11 identified requirements that guided the development of the DES module’s main features, which are detailed in Table 5.

Table 5
DES main features in response to the identified requirements.

DES Features	DesReq#
It is based on Discrete Event Simulation (DES) and Agent Based model that can be used to perform simulations according to input received, also real-time, by the orchestrator and human users through FIWARE/FIROS	DesReq1, DesReq6, DesReq10
It is responsible for the execution of fast time simulations for what-if analysis. The simulations include the robot and the human operators. Such simulations can be executed either by the orchestrator or by a human user through FIWARE/FIROS	DesReq2, DesReq3, DesReq4, DesReq5
It facilitates what-if analysis and experimentations in automatic (orchestrator) or manual modes (human users) and provides an interface for both automated and manual decision-makers to submit actions.	DesReq7, DesReq11
It can include 3D animation	DesReq8, DesReq9

4.2. DES module data exchange capabilities

The DES module has been developed by using the AnyLogic software, a powerful tool for discrete event simulation and agent-based modeling (further information about AnyLogic can be found at https://www.anylogic.com/). Built on the Java Standard Edition (Java SE) platform, the DES module required the design and the implementation of a custom Java library to facilitate the publish/subscribe communication mechanism between the AnyLogic DES environment and FIWARE/FIROS, as well as other platform modules. This library consists of three modules: a web server for Java, a parser module and an HTTP Request Java module to interact with the FIWARE/FIROS API. This library solves any kind of interoperability issue and allows the DES module to communicate real-time with FIWARE/FIROS according to the architecture depicted in Fig. 3.

The web server module includes various functionalities that provide several APIs through which FIWARE/FIROS can send all entity data required by the DES module. Upon receiving a message, the web server forwards it to the parser module for interpretation. These messages adhere to the NGSIv2 standard and are structured in JSON format, which is not directly compatible with the DES module. Thus, the parser module manages the conversion of data from NGSIv2 objects to Java objects, which are readable by the DES module. Conversely, when the DES module needs to send a message, it is routed to the parser module, which converts the DES Java objects into NGSIv2 format, compatible with FIWARE/FIROS. Subsequently, the data in NGSIv2 format are handled by the HTTP request module, which transmits them to FIWARE/FIROS via the HTTP protocol.

4.3. DES development data framework

A key step in developing the DES module is collecting data, information, and logic that represent the real system. This data will later serve as the foundation for developing the DES module. To this end, the authors focus on the fundamental nature of an assembly process. An assembly process consists of several workstations where a sequential and coordinated series of operations aimed at assembling individual components or parts form a final product or subassembly. Within this context, three hierarchical levels can be identified:

- Workstations: these represent the number of workstations (physical locations) within an assembly process, where one or more macro-operations can be executed.
- Macro-Operations: these are high-level operations of a workstation that encompass multiple related micro-operations. They represent significant steps in the assembly sequence, each potentially subject to precedence constraints, meaning that certain macro-operations must be completed before others can begin. Each macro-operation is associated with a specific workstation.
- Micro-Operations: these consists of smaller and more specific actions contributing to the completion of a macro-operation. Each micro-operation can be assigned to either an operator and/or a robot, or may involve collaboration between both, offering flexibility in execution throughout the assembly process. Additionally, each micro-operation can be executed either manually or with the use of specific tools, and it may or may not involve the handling of a component to be assembled or parts necessary for the assembly. In this context, to standardize all types of assembly micro-operations, the authors propose a categorization into four groups: (1) taking a component to be assembled, (2) taking a tool to perform the assembly operation, (3) performing the assembly operation and (4) releasing the tool used to perform the assembly operation.

Essentially, an assembly process may entail one or multiple workstations, representing physical locations where one or more macro-operations, adhering to a predetermined sequence, are carried out. Each macro-operation consists of one or more micro-operations, which can be executed by human operators, robots, or a collaborative effort between both. The diverse combinations of macro-operation sequences and micro-operation allocations to human operators, robots or both of them define the full spectrum of potential pathways (workflows) for completing the assembly process.

With a clear understanding of these concepts, the authors have identified the essential data, along with their format, needed for

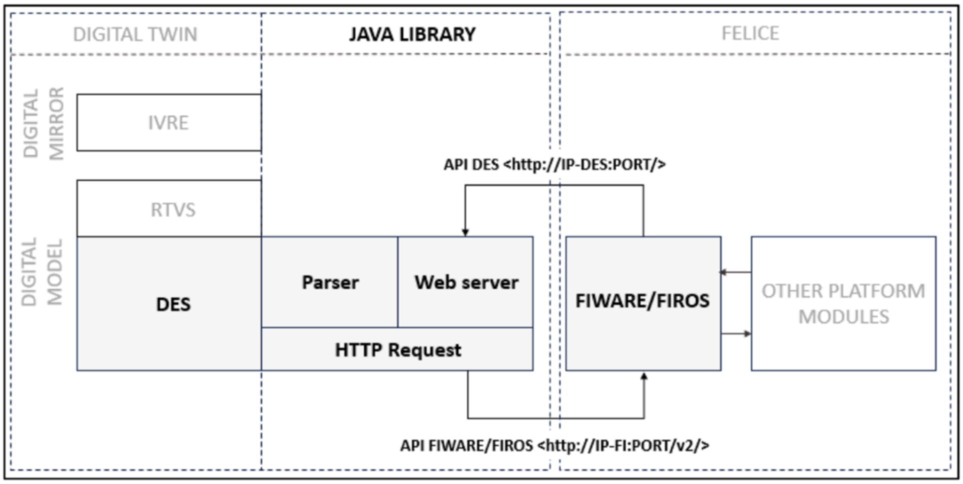


Fig. 3. DES communication architecture.

Table 6
Essential Data for DES Model Development.

Data name	Data description	Data type
Product	This field indicates the name of the final product to be assembled	STRING
Workstation ID	This integer value uniquely identifies a specific workstation within the assembly process where one or more macro-operations can be executed	INT
Macro-operation ID	This unique integer value, which is a multiple of 10, serves to identify each macro-operation in the assembly process. Each macro-operation encompasses one or more micro-operations	INT
Macro-operation constrains	This data consists of a list of integer values associated with each macro-operation. Each integer value represents the ID of a macro-operation that must be completed before the corresponding macro-operation can begin. If there are no precedence constraints for a particular macro-operation, the value is set to 0	ARRAYLIST
Micro-operation description	This refers to a detailed description of the micro-operation that needs to be performed	STRING
Micro-operation orders execution	This is an integer value indicating the sequential order followed for the execution of micro-operations. For instance, a micro-operation with an order of N = 1 will be performed first in the sequence	INT
Micro-operation assignment	This field indicates whether the robot is designated to carry out the operation. It is marked as YES if the robot is capable of performing the micro-operation, and NO if the task is intended solely for human execution.	STRING
Assembly item	This field indicates the name of the component to be assembled as part of the final product and/or parts and tools needed for the assembly	STRING
Micro-operation human operator component time	This field denotes the average time, measured in seconds, required for human operators to take a component to be assembled	DOUBLE
Micro-operation robot component time	This field denotes the average time, measured in seconds, required for the robot to take a component to be assembled and release it to the operator for the assembly	DOUBLE
Micro-operation human operator grasping tool time	This field denotes the average time, measured in seconds, required for the human operator to take a tool necessary to perform a micro-operation	DOUBLE
Micro-operation robot grasping tool time	This field denotes the average time, measured in seconds, required for the robot to take a tool necessary to perform a micro-operation	DOUBLE
Micro-operation human operator assembly time	This field denotes the average time, measured in seconds, required for the human operator to complete the assembly micro-operation	DOUBLE
Micro-operation robot assembly time	This field denotes the average time, measured in seconds, required for the robot to complete the assembly micro-operation	DOUBLE
Micro-operation human operator releasing tool time	This field denotes the average time, measured in seconds, required for the human operator to release a tool used to perform a micro-operation	DOUBLE
Micro-operation robot releasing tool time	This field denotes the average time, measured in seconds, required for the robot to release a tool either to the operator or on a tool holder	DOUBLE

developing a DES model for any assembly process. Table 6 summarizes the list of identified data, including their descriptions and types.

By inputting this data during the development of the DES model, it becomes possible to define all the rules governing the execution of operations within each workstation of an assembly process. Furthermore, it is important to note that the DES module has been also developed to account for ergonomic factors concerning the operators. In order to assess and optimize operator ergonomics, a custom-designed mathematical model has been developed and implemented, correlating operator efficiency with operator fatigue. However, due to its complexity and depth, this topic deserves a dedicated research article. Therefore, at this stage, it falls beyond the scope of this study.

4.4. DES architecture and implementation

The DES module consists of the following four main components: process logics, input parameters, output performance measures and 3D animation. Fig. 4 illustrates the DES architecture. Followings, each of these components will be thoroughly described to provide an overview of the DES module's characteristics and implementation logic.

Process Logic.

The process logic defines the rules that govern the simulation model. By using AnyLogic software, an agent-based tool, the authors chose a modular approach to develop a simulation model. This approach allows easy extension and scalability regarding the number of workstations and the macro- and micro-operations performed at each workstation. In this regard, as an assembly process may entail one or multiple workstations, where one or more operations are executed, the author decided to develop AnyLogic simulation modules reproducing workstations along with its operations. The sequential or parallel connection of those modules allow the reproduction of the overall assembly process. Each AnyLogic simulation module has always the same structure and consists of a series of blocks and several agent entities. The interconnection of blocks and the movement of agent entities through the blocks simulate the workstation assembly operations. Following the micro-operations categorization presented in Section 4.3, within the simulation environment provided by AnyLogic, *service* blocks have been used to simulate the time to take components to be assembled (1) and to perform assembly operations (3), while *Seize-type*, *MoveTo-type*, and *Release-type* blocks have been used to simulate the movements of either the operator or the robot to take and then release a tool necessary to complete an assembly operation (2)(4). As concerns module agent entities, three types characterize it. The agent entity *product* represents the item being assembled and acts as a dynamic entity moving through the assembly line's process logic. The agent entity *product* receives services from the static agent entities *operator* and *robot*, which act as resources. After an *operator* or *robot* completes a single assembly operation (micro-operation), the agent entity *product* passes through a *finished* block within that module to determine if the agent entities *operator* and *robot* have completed their list of micro-operations for that specific product within that specific workstation. If the list of micro-operations remains incomplete, the agent entity *product* is reassigned to the agent entities *operator* and/or *robot* (depending on input data values) and *service* blocks are again used to simulate the time needed to perform the operation. This modelling structure allows also to easily change the number of micro-operations to be executed within each workstation by solely acting on the input data (adding or removing micro-operations from the input data table) and without varying the modeling structure of the module workstation.

This ensures that the proposed simulation model is both flexible and adaptable to various types of assembly process. This iterative cycle continues until all micro-operations are completed for each macro-operation at the specific workstation/module. As soon as the list is complete, the entity *product* moves to the next workstation/module. Fig. 5 visually illustrates this iterative cycle.

Once the process logic is understood, to run the simulations,

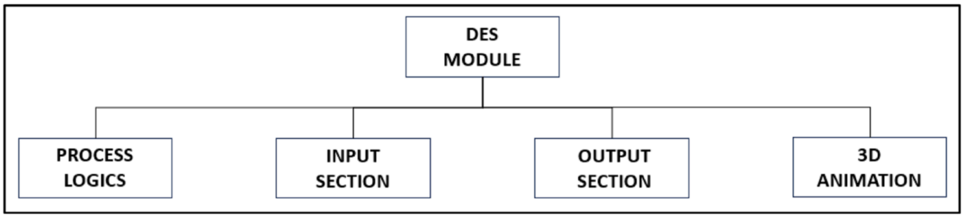


Fig. 4. DES architecture.

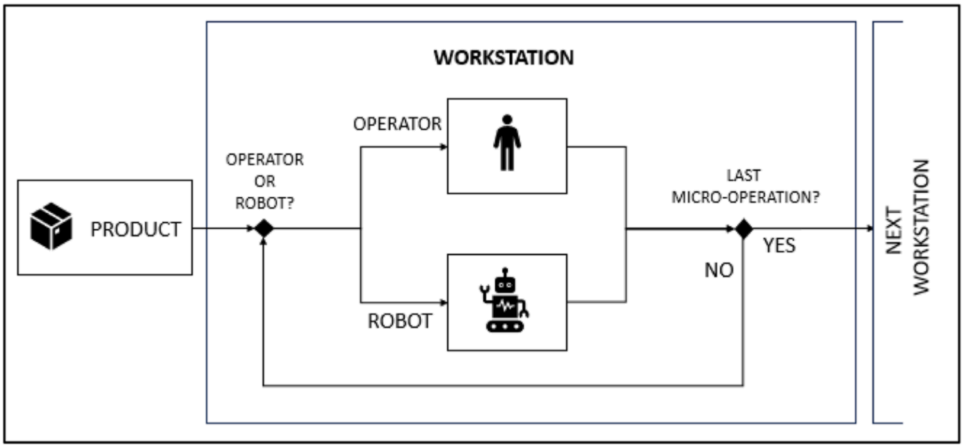


Fig. 5. Iterative cycle for agent entity *product*.

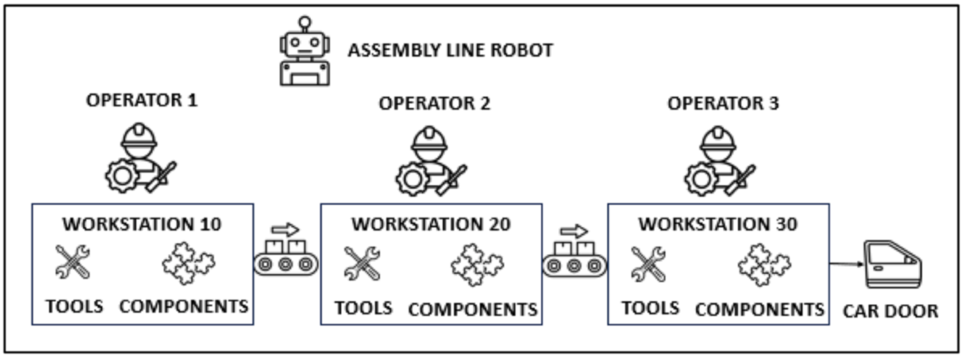


Fig. 6. Assembly line schematic visualization.

Table 7
Workstations macro-operations overview.

Workstation #	Macro-operation ID	Macro-operation description
1	10	Glass front door assembly
	20	Posterior vertical coverage assembly
	30	Left front door window seal assembly
	40	Anterior vertical coverage assembly
	50	Weather strip front door belt end assembly
	60	Rearview mirror assembly
2	10	Tie rod assembly
	20	Plugs assembly
	30	Door carrier assembly
	40	Lock system assembly
3	10	Front door speaker assembly
	20	Door panel assembly to the door frame
	30	Window control panel and plug assembly
	40	Speaker frame assembly
	50	Anti-vibration pad assembly

predefined standard input data must be accessible in a structured table stored in the AnyLogic internal database. An example of this table will be presented in Section 5 along with the case study. This data can be inputted to the DES module either manually by the end-user or automatically by the orchestrator via FIWARE/FIROS.

Finally, this section introduces an additional functionality of the DES module. Leveraging on the input data provided to the simulation model and, specifically, on macro-operation sequence constraints and human/robot task assignment possibilities, an algorithm within the AnyLogic environment can automatically generate all possible workflows. As discussed in Section 4.3, these workflows represent alternative scenarios for executing the same assembly process, determined by changing the sequence of macro-operations to be executed and by assigning the same task either to human operators or robots, or both of them. This functionality enables the end-user and/or orchestrator to analyze each workflow and select the optimal one based on a predefined set of KPIs related to assembly process productivity, efficiency, and operator ergonomics.

Input section.

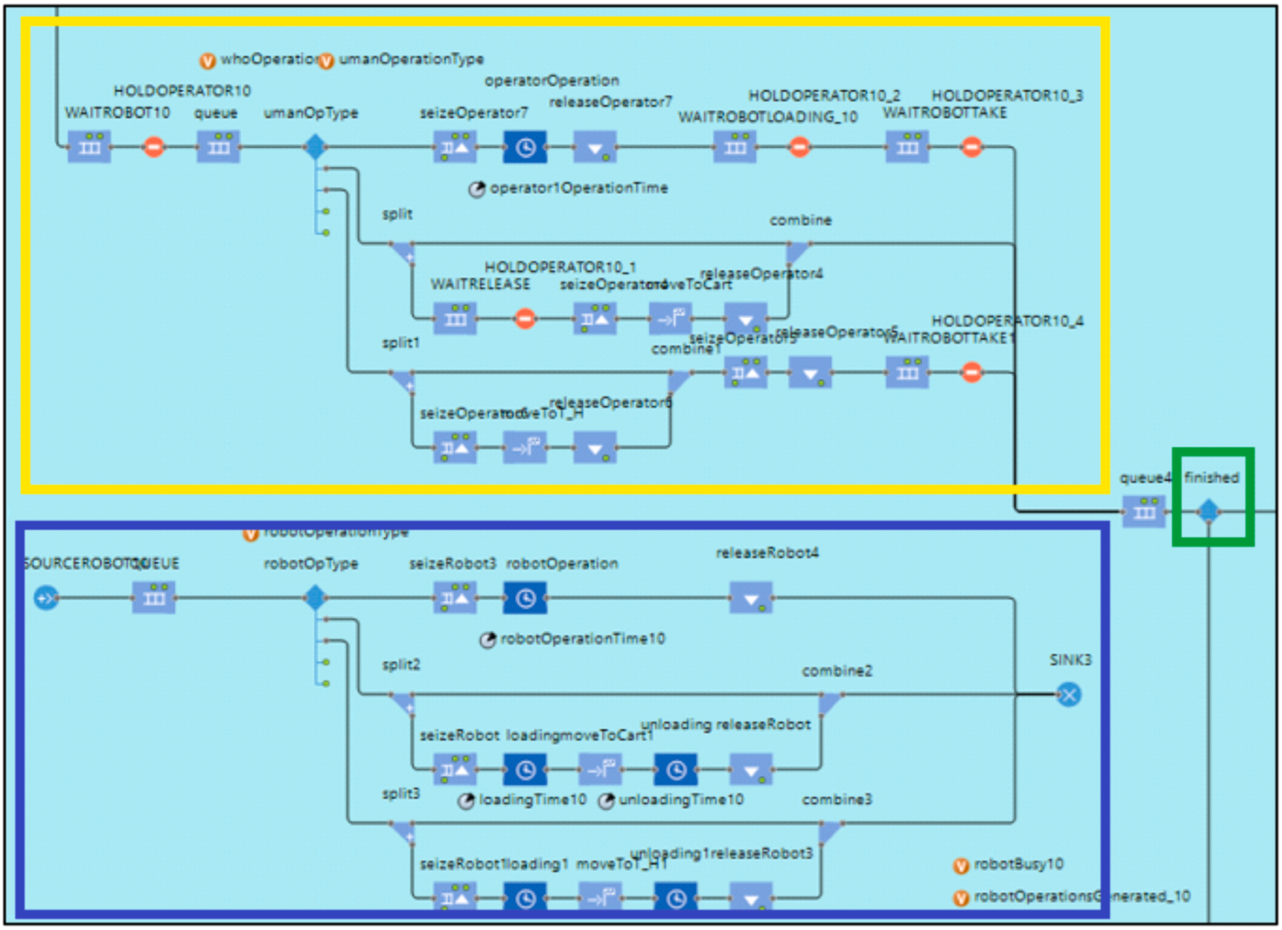


Fig. 7. Module macro-sections within AnyLogic environment.

The input section serves the purpose of enabling the end users and the orchestrator to conduct what-if analyses and experiments by interacting with a series of commands linked to various parameters. Through this section, users, whether humans or the orchestrator module, gain access to a dedicated Graphical User Interface (GUI). Here, they can configure several parameters both before starting a simulation and during its execution. The GUI offers users the flexibility to specify the types of analyses and experiments they wish to conduct and make corresponding adjustment to all necessary parameters. Below, the authors provide a short description of the commands available within the input section. The GUI visualization will be presented in Section 5, along with the case study.

- Operators' commands: users can utilize slider-type objects to adjust both the speed and efficiency of operators working on the assembly line.
- Robots' commands: users can utilize slider-type objects to adjust both the speed and efficiency of robots working on the assembly line.
- Ergonomic problems commands: in this section, users find several check box commands, each corresponding to an operator on the assembly line. When one of these commands is activated, it triggers a change in the efficiency value of the respective operator. This adjusted efficiency value is calculated based on a specific mathematical model implemented within AnyLogic. On the other side, if the check boxes are deactivated, the operators' efficiency remains constant, retaining the value set by the user through the operators' commands.
- Collaboration command: in this section, manual users find a single check box command titled "Start Collaboration with Orchestrator". When activated, two actions are initiated: firstly, activation of connection to FIWARE/FIROS, enabling the DES module to publish and subscribe messages; secondly, transmission of messages from the DES module to the orchestrator, containing information on assembly line output measures. This transmission enables the orchestrator to autonomously make decisions aimed at improving assembly line productivity and ergonomics based on the output measure values.
- Robots' position management: this section enables users to determine the workstations where a robot can be assigned to assist operators in executing operations. The objective is to evaluate the impact of human-robot collaboration on workstation productivity and ergonomics.
- Working time randomness: in this section, users can adjust the variability of both operators' and robots' time in order to generate output values closer to those observed in the real system.
- Workflow management: within this section, users can select for each workstation a specific workflow from the range of available options to conduct what-if analyses and experimentation on the assembly process.
- Work shift management: users have the ability to adjust parameters concerning work shifts before starting the simulation. This section includes three commands: (1) a command to set the duration of each work shift in hours, (2) an edit box to determine the number of working shifts within a 24-hour period, and (3) a command enabling users to specify the number of shifts to be simulated, thus defining the total duration of the simulation.
- Advanced controls for operators' efficiency: in this section, users can access a variety of edit box commands to adjust parameters associated with the mathematical model used to assess operators'

Table 8
DES module input data – structured data table for workstation 10.

Workstation ID	Macro-operations		Micro-operations			Assembly item	Component time [sec]		Tool grasping time [sec]		Assembly time [sec]		Tool release time [sec]	
	ID	Constraints	Description	Order	Assignment		Operator	Robot	Operator	Robot	Operator	Robot	Operator	Robot
10	10	0	Take glass front door from cart	1	NO	glass front door	7							
			Insert it in internal door channel	2	NO	-					6,5			
			Push glass front door towards anterior side of door	3	NO	-					7,5			
			Manually fit internal door gasket on sheet metal edge	4	NO	-					7,5			
			Take screwdriver from line side	5	NO	screwdriver			7,5					
			Take one screw from pouch	6	NO	screw	7,5							
			Place screw on screwdriver tip	7	NO	-					7,5			
			Fix superior part of glass front door	8	NO	-					7,5			
			Take one screw from pouch	9	NO	screw	7,5							
			Place it on screwdriver tip	10	NO	-					7,5			
			Fix inferior part of glass front door	10	NO	-					7,5			
			Leave screwdriver on workstation	11	NO	screwdriver							7,5	
	20	0	Take posterior vertical coverage from cart (if performed by the robot, hand over the component to the operator)	1	YES	posterior vertical coverage	7,5	10,5						
			Manually position posterior vertical coverage	2	NO	-					7			
			Check position of posterior vertical coverage	3	NO	-					6,5			
			Align upper profile of coverage with gasket and ensure hook is correctly positioned	4	NO	-					7,5			
			Take screwdriver from line side (if performed by the robot, hand over the screwdriver to the operator)	5	YES	screwdriver			6,5	5,5				5,5
			Take three screws from pouch	6	NO	screw	7,5							
			Place screws on screwdriver tip, insert one at a time	7	NO	-					7			
			Fix panel starting from bottom	8	NO	-					6,5			
			Leave screwdriver on line side (if performed collaboratively, the robot receives the screwdriver from the operator and leaves it on the line side)	9	YES	screwdriver				6			7,5	6
	30	20	Take left front door window seal from cart (if performed by the robot, hand over the component to the operator)	1	YES	left front door window seal	7	11						
			Start inserting manually from belt side	2	NO	-					6,5			
			Continue manually insertion to other side	3	NO	-					7,5			
			Using manual pressure, start inserting left front door window seal from rear area of B pillar door by coupling it with sheet metal flap	4	NO	-					7			
			Apply left front door window seal inside duct	5	NO	-					6,5			
			Using manual pressure, complete insertion of left front door window seal by arranging it up to B pillar	6	NO	-					7,5			
			Visually check correct positioning of left front door window seal in anterior part of door	7	NO	-					7			
	40	30	Take anterior vertical coverage from cart	1	NO	anterior vertical coverage	6,5							
			Place anterior vertical coverage	2	NO	-					7,5			
			Fix anterior vertical coverage by inserting five joints in appropriate slots	3	NO	-					7			
			Check if anterior vertical coverage is correctly placed	4	NO	-					6,5			
	50	40	Take weatherstrip front door belt end from cart	1	NO	weatherstrip front door belt end	7,5							
			Manually put it on glass sliding seat in belt line area, starting from front door area	2	NO	-					7			
			Place flocked lip of weatherstrip front door belt end in contact with glass	3	NO	-					6,5			
			Using manual pressure, slide weatherstrip front door belt end onto door sheet along entire length up to B pillar	4	NO	-					7,5			
			Check correct alignment between weatherstrip front door belt end and door sheet from front door area to B pillar	5	NO	-					7,5			
	60	50	Take one rearview mirror from cart (if performed by the robot, hand over the component to the operator)	1	YES	rearview mirror	7,5	20,5						
			Take wire, fold it on itself, and insert its connector into box of door through external hole of front door	2	NO	-					7,5			
			Manually couple mirror into slot below	3	NO	-					7,5			

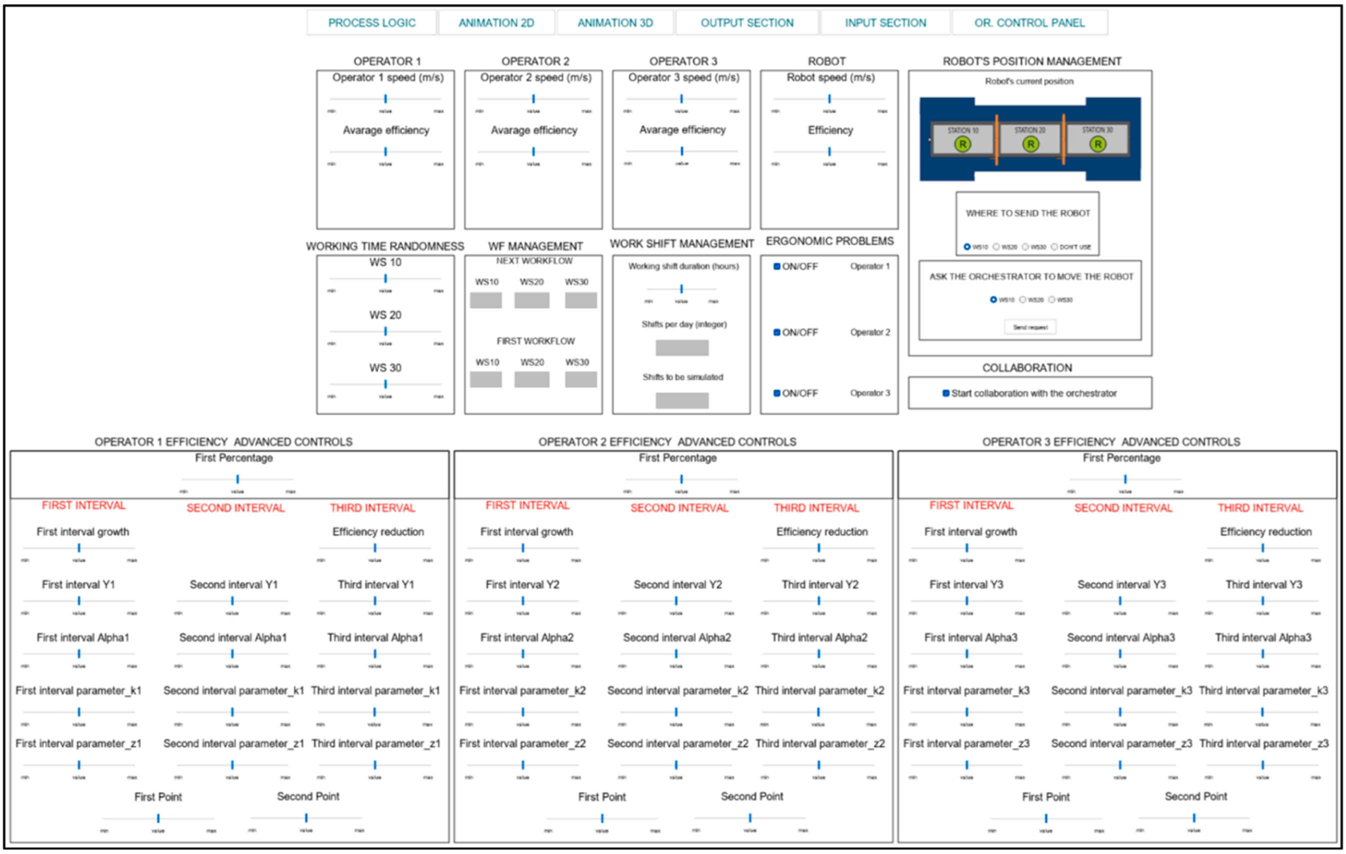


Fig. 8. DES Module input section.

efficiency.

Output section

The output section enables users of the DES module to evaluate the effects of input parameters on various KPIs, facilitating what-if analyses and the selection of optimal configurations for assembly process productivity and ergonomics. Six KPIs can be monitored, as described below.

- Utilization levels: this section enables the evaluation of the utilization levels of both operators and robots involved in the assembly process.
- Productivity: this section presents productivity values in terms of number of products assembled per hour at each workstation and across the entire assembly line.
- Assembly times: this section provides for each product the assembly time at workstation level and the total assembly time at assembly line level.
- Average assembly time: this section displays the average assembly time at the workstation level, as well as the average total assembly time at the assembly line level.
- Operators' efficiency: this section displays the efficiency of each operator individually, as well as the average efficiency across all operators on the assembly line.

The GUI of the output section will be visualized in Section 5, along with the case study.

3D animation.

The final section of the DES module features a 3D animation component, providing users with a clear visual representation of the assembly line's operations and performance. Leveraging on AnyLogic's capabilities, users can create 3D animations using geometric objects that support the third dimension (Z-height). In the AnyLogic environment, at least one object *camera* is required to visualize 3D animations. Within the DES module, four objects *camera* are employed, offering 3D visualization of the assembly processes from different angles. Each camera is linked to an object *3D window*, which displays the user what camera is capturing. The 3D animation of the DES module is shown in Section 5, along with the case study presentation.

5. Case study

This section presents a case study to demonstrate the capabilities and functionalities of the DES module. Section 5.1 provides a description of the reference context, outlining the assembly process under consideration along with its operations. Additionally, the authors introduce an example of a structured data table utilized to provide input data to the DES module. Moving forward, Section 5.2 outlines the implementation of the DES module within the case study framework, offering details on its process logics, input and output sections, and the 3D animation features. Finally, Section 5.3 showcases a practical application of the DES module, highlighting its interaction with the orchestrator and analyzing the main results and key findings.

5.1. Reference context

The case study focuses on a line dedicated to the assembly of car front doors. The line includes three workstations, each performing several macro-operations. One operator is assigned to each workstation and, a conveyor belt moves the car front doors through these stations. All necessary parts and tools for assembly are readily available at each workstation. Additionally, a robot is available for the entire line and can be assigned to any workstation as needed to support operators. Fig. 6 depicts schematically the assembly line, while Table 7 provides a

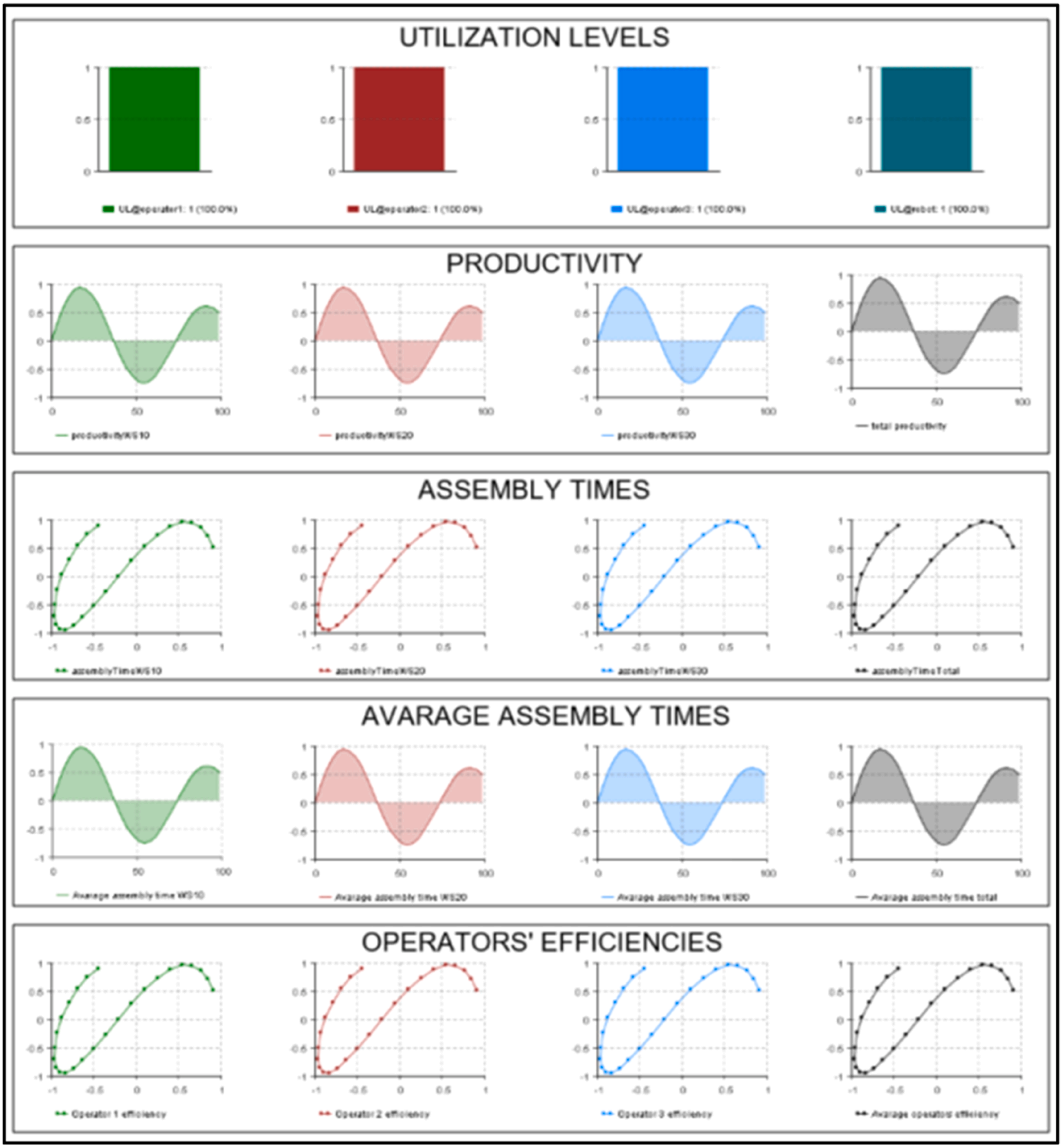


Fig. 9. Output section.

summary of the macro-operations performed at each workstation.

5.2. DES module implementation

This section provides an overview of the DES module's implementation, designed to mirror the dynamics of the considered case study. The implementation started with the deployment of the process logics that precisely reflected the specifics of the case study. To this end, three distinct AnyLogic simulation modules, following the same structure, were created to reproduce the assembly operations at each of the three workstations on the assembly line. Fig. 7 depicts in details the module related to workstation 10, along with its blocks and interconnections. Similar modules have been also developed for workstation 20 and 30.

Each module consists of two macro-sections. The first one is dedicated to operations performed solely by either the operator or the robot, as shown in the yellow box in Fig. 7. The second one considers operations involving human-robot collaboration, represented by the blue box in Fig. 7. In these collaborative scenarios, the robot actively assists the human operator. Specifically, when collaboration is required, the robot can: (1) retrieve assembly components and/or equipment from their designated positions within the workstation and hand them to the

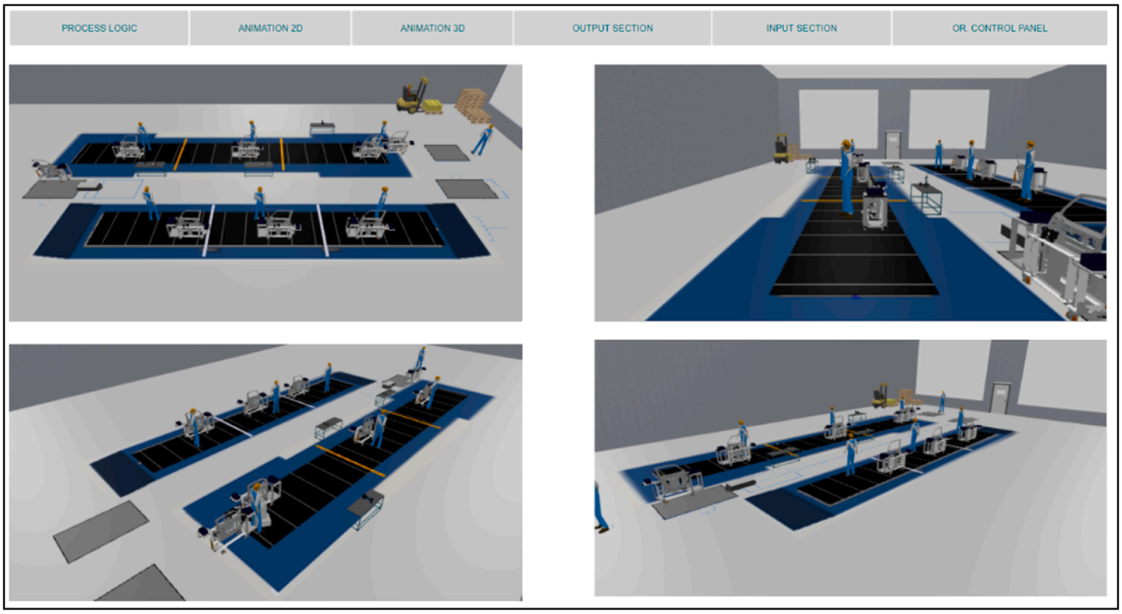


Fig. 10. Different views of the assembly line 3D virtual environment.

human operator, which completes the assembly tasks; (2) after the assembly is finished, the human operator returns the assembly equipment to the robot, which then places it back in its original location within the workstation. For the selected case study, Table 8 in Section 5.3 lists the micro-operations where human-robot collaboration is feasible, indicated by a “YES” in the micro-operation assignment column.

Finally, each module, as detailed in Section 4.4, incorporate a finished block to ensure the DES module’s flexibility and scalability, as denoted by the green box within Fig. 7.

Following the implementation of the process logics, next step was to develop the input and output sections, customized to fit the requirements of the case study under investigation. Fig. 8 and Fig. 9 depict the GUI of the DES module’s input and output sections within the AnyLogic environment, respectively.

Table 9
Macro-operation sequences for workstation 10.

Macro-operations sequence I	Macro-operations sequence II
10	20
20	10
30	30
40	40
50	50
60	60

As the final step of the DES implementation, the authors focused on creating the 3D virtual environment for the entire assembly lines. Four camera objects were utilized to provide 3D visualization of the assembly processes from various angles. Fig. 10 provides four different views of the same assembly line. It is worth noting that 3D models were also developed for the assembly line of rear door car assembly, which are displayed but are outside the scope of this study.

5.3. *Calculation data model*

In this section, the authors demonstrate the utilization of the DES module by an end-user for conducting what-if analysis scenarios, with a specific focus on workstation 10 of the assembly line under investigation. It is important to note that similar analyses can also be carried out for workstations 20 and 30. The standardized input data for workstation 10, required to run the DES module, is presented in its structured format in Table 8.

Based on the data presented in Table 8, the total number of workflows associated with this workstation can be computed. The factors considered in determining all potential workflows for each station include: (a) the sequence of macro-operations to be performed, and (b) the potential involvement of the robot in specific micro-operations. At workstation 10, where macro-operation 10 and macro-operation 20 have no precedence constraints, there are two possible scenarios for completing all operations: macro-operation 10 can either precede

Table 10
Comparative Analysis of AT Values for Workflow IDs 01, 32, 33, and 64.

Replication [#]	Workflow ID	Macro-operations sequence	Robot collaboration level [%]	AT [sec]	Reduction [%]
1	01	I	0 %	181,528	-
	32	I	100 %	173,299	−4533 %
2	01	I	0 %	181,242	-
	32	I	100 %	173,04	−4525 %
3	01	I	0 %	181,012	-
	32	I	100 %	172,994	−4430 %
1	33	II	0 %	181,121	-
	64	II	100 %	173,076	−4442 %
2	33	II	0 %	181,121	-
	64	II	100 %	172,734	−4631 %
3	33	II	0 %	181,05	-
	64	II	100 %	172,826	−4542 %

Table 11
Extract of the DES data model.

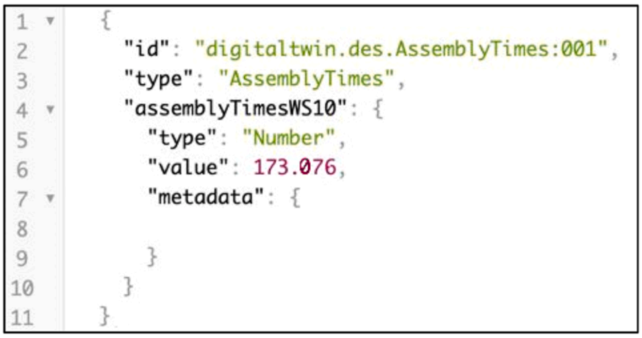
Module Name	Sub-module name	Entity name	Entity type	Attribute name	Attribute type	Short description
Digital Twin	des	digitaltwin.des.UtilizationLevels	Utilization Levels	utilizationLevelOperator1	double	This number, ranging from 0 to 1, indicates the simulated utilization level of operator 1 at station 10
				utilizationLevelOperator2	double	This number, ranging from 0 to 1, indicates the simulated utilization level of operator 2 at station 20
				utilizationLevelOperator3	double	This number, ranging from 0 to 1, indicates the simulated utilization level of operator 3 at station 30
				utilizationLevelRobot	double	This number, ranging from 0 to 1, indicates the simulated utilization level of the robot
Digital Twin	des	digitaltwin.des.Productivity	Productivity	Productivity1	double	This represents workstation 10 simulated productivity, measured by the number of completed doors per hour
				Productivity2	double	This represents workstation 20 simulated productivity, measured by the number of completed doors per hour
				Productivity3	double	This represents workstation 30 simulated productivity, measured by the number of completed doors per hour
				TotalProductivity	double	This represents the assembly line productivity, measured by the number of completed doors per hour
Digital Twin	des	digitaltwin.des.AssemblyTimes	Assembly Times	assemblyTime1	double	This is the simulated assembly time at station 10 (in seconds) for a specified door (doorID). It reflects the time taken by the operator to complete workstation 10 operations for that specific door
				assemblyTime2	double	This is the simulated assembly time at station 20 (in seconds) for a specified door (doorID). It reflects the time taken by the operator to complete workstation 20 operations for that specific door
				assemblyTime3	double	This is the simulated assembly time at station 30 (in seconds) for a specified door (doorID). It reflects the time taken by the operator to complete workstation 30 operations for that specific door
				totalAssemblyTime	double	This is the simulated total assembly time (in seconds) for a specific door (doorID)
				doorID	int	This ID corresponds to the door for which the assembly times are referenced
Digital Twin	des	digitaltwin.des.OperatorsEfficiency	Operators Efficiency	operatorsEfficiency1	double	This represents the operator simulated efficiency at station 10, ranging from 0 to 1
				operatorsEfficiency2	double	This represents the operator simulated efficiency at station 20, ranging from 0 to 1
				operatorsEfficiency3	double	This represents the operator simulated efficiency at station 30, ranging from 0 to 1
Digital Twin	des	digitaltwin.des.OrchestratorCollaboration	Orchestrator Collaboration	Collaborate	Boolean	This variable indicates if collaboration between the digital twin and the orchestrator is needed (true for required, false for not required)
Digital Twin	des	digitaltwin.des.WhereSendRobot	WhereSend Robot	timestamp	DateTime	This is the timestamp (in UTC format) when the new collaboration message is sent
				positionWS	Integer	This variable indicates the workstation where the robot has to go (10, 20 or 30)

```
{
  "id": "Orchestrator.Planner.Plannedworkflow:001",
  "type": "Plannedworkflow",
  "digitaltwinworkflowId": {
    "type": "Number",
    "value": 64,
    "metadata": {

    }
  },
  "refStation": {
    "type": "Text",
    "value": "Felice.common.Station:001",
    "metadata": {

    }
  }
}
```

Fig. 11. Orchestrator publish message.



```
{
  "id": "digitaltwin.des.AssemblyTimes:001",
  "type": "AssemblyTimes",
  "assemblyTimesWS10": {
    "type": "Number",
    "value": 173.076,
    "metadata": {

    }
  }
}
```

Fig. 12. DES module publish message.

macro-operation 20, or vice versa. As a result, there are two distinct sequences for completing all the operations of workstation 10, as outlined in Table 9.

Taking into account factor (b), where there are five identified operations at station 10 that the robot can perform (see Table 8, column assignment), 2^5 possible configurations from each sequence of macro-operations can be generated. Therefore, the total number of potential workflows for station 10 will be:

$$\#workflows\ for\ station 10 = 2^5 * n.of\ sequences = 2^5 * 2 = 32 * 2 = 64$$

In conclusion, there are a total of 64 different scenarios available for evaluation, each assessing the combination of macro-operation sequences and task allocations between human operators and the robot. Here, the DES module provides invaluable support to the end user, enabling swift and efficient testing of each scenario. Essentially, within the “WF management” section of the input GUI, the end user can input for each workstation the desired workflow ID for assessment and receive pre-defined KPI values from the DES module. Table 10 exemplifies the Assembly Time (AT) for workstation 10 across macro-operation sequences I and II. For both of them, it details scenarios without collaborative robot support (workflow ID01 and workflow ID33), where operations are solely handled manually by the operator, and with 100 % collaborative robot support (workflow ID32 and workflow ID64), where the robot assists in all five operations it can perform. In addition, the evaluation of each workflow has been replicated three times (see the column Replication in Table 10), to consider the stochastic nature of the assembly line operations.

The analysis of the table results reveals that changing the macro-operation sequences has minimal effect on the AT values. However, the integration of robot support in operation execution leads to more significant reductions in AT values, ranging from −4.43 % to −4.631 % compared to scenarios without robot involvement. It is important to note that selecting the optimal workflow should also consider the evaluation of other KPI values, aligning with the specific targets and requirements set for the end-users.

It is worth mentioning that the what-if analysis scenarios can be also executed through the automated communication between the DES and orchestrator modules. Although the specific dynamics and logic governing the orchestrator module’s functionalities are beyond the scope of this research, the authors will briefly outline how communication between these two modules is established. Firstly, the authors provide an extract of the DES data model in Table 11, serving as a common language for ensuring both syntactic and semantic interoperability. This structured data is necessary to allow message exchange between the DES and other modules, including the orchestrator, via FIWARE/FIROS. Next, Fig. 11 depicts an example of publish message sent from the orchestrator to the DES module via FIWARE/FIROS, seeking to assess the workflow ID64 for workstation 10. Conversely, Fig. 12 depicts a publish message from the DES module via FIWARE/FIROS in response to the orchestrator’s publish request. The data exchanged pertains to the assembly time for workstation 10 under workflow ID64.

6. Conclusions

This research is part of the European Union Horizon 2020 Project “FlExible assembLy manufacturIng with human-robot Collaboration and digital twin modEls (FELICE),” an initiative that integrates various disciplines such as robotics, artificial intelligence, computer vision, data analytics, process optimization, and ergonomics. The project aims to develop a modular platform that combines autonomous and cognitive technologies to enhance the agility, productivity, safety, and well-being of human workers in assembly production systems. The focus of this research lies specifically on the DT module, with particular emphasis on one of its key components, the DES module. The DES module facilitates fast-time simulations of assembly lines to support what-if analysis scenarios. Specifically, it enables testing of various assembly line configurations to evaluate the impact of different inputs, including task allocation between operators and robots, and task execution via active collaboration between human operators and robots. The evaluation is based on predefined KPIs related to the productivity of the assembly line and the ergonomics of human operators.

The authors started by presenting the DES module’s requirements and main features, followed by an overview of its data exchange capabilities, including the DES communication architecture. Subsequently, the implementation of the DES module has been presented. The DES module consists of four components: process logics, input section, output section, and 3D animation. The process logic defines the rules governing simulation execution, while the input section facilitates what-if analyses by configuring various input parameters to generate different assembly scenarios. On the other side, the output section enables users to assess the effects of input parameters on diverse KPIs, supporting end-users’ decision on assembly configuration selection. Finally, the 3D animation component provides users with a clear visual representation of the assembly line’s operations.

In the final part of the study, the authors present a case study to demonstrate the capabilities and functionalities of the DES module. This includes a description of the assembly process under consideration, the DES module implementation details within the case study framework, and the practical application of the DES module. Additionally, the authors demonstrate how the DES module can interact with other platform modules, particularly showcasing its interaction with the orchestrator module within the case study context.

CRediT authorship contribution statement

Vittorio Solina: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Francesco Longo:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Letizia Nicoletti:** Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Antonio Cimino:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research work was partly supported by the FELICE Project – Flexible Assembly Manufacturing with Human-Robot Collaboration and Digital Twin Models – funded from the European Union’s Horizon 2020 Research and Innovation program under grant agreement No. 101017151.

References

- [1] Masood T, Sonntag P. Industry 4.0: Adoption challenges and benefits for SMEs. Comput Ind 2020;121:103261.
- [2] Longo F, Mirabelli G, Nicoletti L, Solina V. An ontology-based, general-purpose and Industry 4.0-ready architecture for supporting the smart operator (Part I–Mixed reality case). J Manuf Syst 2022;64:594–612.
- [3] Cimino A, Gnoni MG, Longo F, Solina V. Integrating multiple industry 4.0 approaches and tools in an interoperable platform for manufacturing SMEs. Comput Ind Eng 2023;186:109732.
- [4] Schulze F, Dallasega P. Lean and Industry 4.0 mitigating common losses in engineer-to-order theory and practice: an exploratory study. Flex Serv Manuf J 2023:1–41.
- [5] Tao F, Zhang H, Liu A, Nee AY. Digital twin in industry: state-of-the-art. IEEE Trans Ind Inform 2018;15(4):2405–15.

[6] Cimino A, Longo F, Mirabelli G, Solina V, Verteramo S. An ontology-based, general-purpose and Industry 4.0-ready architecture for supporting the smart operator (Part II–virtual reality case). J Manuf Syst 2024;73:52–64.
[7] Pivoto DG, de Almeida LF, da Rosa Righi R, Rodrigues JJ, Lugli AB, Alberti AM. Cyber-physical systems architectures for industrial internet of things applications in Industry 4.0: a literature review. J Manuf Syst 2021;58:176–92.
[8] Karuppiah K, Sankaranarayanan B, Ali SM, Bhalaji RKA. Decision modeling of the challenges to human–robot collaboration in industrial environment: a real world example of an emerging economy. Flex Serv Manuf J 2023;35(4):1007–37.
[9] Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. J Clean Prod 2020;252:119869.
[10] Li L, Aslam S, Wileman A, Perinpanayagam S. Digital twin in aerospace industry: a gentle introduction. IEEE Access 2021;10:9543–62.
[11] Jones D, Snider C, Nassehi A, Yon J, Hicks B. Characterising the Digital Twin: a systematic literature review. CIRP J Manuf Sci Technol 2020;29:36–52.
[12] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: a categorical literature review and classification. Ifac-Pap 2018;51(11):1016–22.
[13] Park Y, Woo J, Choi S. A cloud-based digital twin manufacturing system based on an interoperable data schema for smart manufacturing. Int J Comput Integr Manuf 2020;33(12):1259–76.
[14] Liu Z, Liu Q, Xu W, Wang L, Zhou Z. Robot learning towards smart robotic manufacturing: a review. Robot Comput-Integr Manuf 2022;77:102360.
[15] Wang KJ, Lee YH, Angelica S. Digital twin design for real-time monitoring–a case study of die cutting machine. Int J Prod Res 2021;59(21):6471–85.
[16] Weichhart G, Panetto H, Molina A. Interoperability in the cyber-physical manufacturing enterprise. Annu Rev Control 2021;51:346–56.
[17] Liu B, Zhang Y, Zhang G, Zheng P. Edge-cloud orchestration driven industrial smart product-service systems solution design based on CPS and IIoT. Adv Eng Inform 2019;42:100984.
[18] Mo F, Querejeta MU, Hellewell J, Rehman HU, Rezabal MI, Chaplin JC, Ratchev S. PLC orchestration automation to enhance human–machine integration in adaptive manufacturing systems. J Manuf Syst 2023;71:172–87.
[19] Saenz J, Behrens R, Schulenburg E, Petersen H, Gibaru O, Neto P, Elkmann N. Methods for considering safety in design of robotics applications featuring human-robot collaboration. Int J Adv Manuf Technol 2020;107:2313–31.
[20] Xu W, Cui J, Liu B, Liu J, Yao B, Zhou Z. Human-robot collaborative disassembly line balancing considering the safe strategy in remanufacturing. J Clean Prod 2021;324:129158.
[21] Bilberg A, Malik AA. Digital twin driven human–robot collaborative assembly. CIRP Ann 2019;68(1):499–502.
[22] Lv Q, Zhang R, Sun X, Lu Y, Bao J. A digital twin-driven human-robot collaborative assembly approach in the wake of COVID-19. J Manuf Syst 2021;60:837–51.
[23] Zhang R, Lv J, Bao J, Zheng Y. A digital twin-driven flexible scheduling method in a human–machine collaborative workshop based on hierarchical reinforcement learning. Flex Serv Manuf J 2023:1–23.
[24] Alazab M, Khan LU, Koppu S, Ramu SP, Iyapparaja M, Boobalan P, Aljuhani A. Digital twins for healthcare 4.0-recent advances, architecture, and open challenges. IEEE Consum Electron Mag 2022.
[25] Purcell W, Neubauer T. Digital Twins in agriculture: a state-of-the-art review. Smart Agric Technol 2023;3:100094.
[26] Roque Rolo G, Dionisio Rocha A, Tripa J, Barata J. Application of a simulation-based digital twin for predicting distributed manufacturing control system performance. Appl Sci 2021;11(5):2202.
[27] Židek K, Piteľ J, Adámek M, Lazorík P, Hošovský A. Digital twin of experimental smart manufacturing assembly system for industry 4.0 concept. Sustainability 2020;12(9):3658.
[28] Sun X, Bao J, Li J, Zhang Y, Liu S, Zhou B. A digital twin-driven approach for the assembly-commissioning of high precision products. Robot Comput-Integr Manuf 2020;61:101839.
[29] Yi Y, Yan Y, Liu X, Ni Z, Feng J, Liu J. Digital twin-based smart assembly process design and application framework for complex products and its case study. J Manuf Syst 2021;58:94–107.
[30] Coelho F, Relvas S, Barbosa-Póvoa AP. Simulation-based decision support tool for in-house logistics: the basis for a digital twin. Comput Ind Eng 2021;153:107094.
[31] Korth B, Schwede C, Zajac M. Simulation-ready digital twin for realtime management of logistics systems. 2018 IEEE International Conference on Big Data (Big Data). IEEE; 2018, December. p. 4194–201.
[32] Negri E, Fumagalli L, Macchi M. A review of the roles of digital twin in CPS-based production systems. Procedia Manuf 2017;11:939–48.
[33] He B, Bai KJ. Digital twin-based sustainable intelligent manufacturing: a review. Adv Manuf 2021;9:1–21.
[34] Somers RJ, Douthwaite JA, Wagg DJ, Walkinshaw N, Hierons RM. Digital-twin-based testing for cyber–physical systems: a systematic literature review. Inf Softw Technol 2023;156:107145.
[35] Soori M, Arezoo B, Dastres R. Digital Twin for smart manufacturing, a review. Sustain Manuf Serv Econ 2023:100017.
[36] Mukherjee D, Gupta K, Chang LH, Najjaran H. A survey of robot learning strategies for human-robot collaboration in industrial settings. Robot Comput-Integr Manuf 2022;73:102231.
[37] Segura P, Lobato-Calleros O, Ramírez-Serrano A, Soria I. Human-robot collaborative systems: structural components for current manufacturing applications. Adv Ind Manuf Eng 2021;3:100060.
[38] Cherubini A, Passama R, Crosnier A, Lasnier A, Fraisse P. Collaborative manufacturing with physical human–robot interaction. Robot Comput-Integr Manuf 2016;40:1–13.
[39] Michalos G, Kousi N, Karagiannis P, Gkournelos C, Dimoulas K, Koukas S, Makris S. Seamless human robot collaborative assembly–an automotive case study. Mechatronics 2018;55:194–211.
[40] Malik AA, Bilberg A. Complexity-based task allocation in human-robot collaborative assembly. Ind Robot: Int J Robot Res Appl 2019;46(4):471–80.
[41] Ranz F, Hummel V, Sihn W. Capability-based task allocation in human-robot collaboration. Procedia Manuf 2017;9:182–9.
[42] Tsarouchi P, Matthaiakis AS, Makris S, Chryssolouris G. On a human-robot collaboration in an assembly cell. Int J Comput Integr Manuf 2017;30(6):580–9.
[43] Villani V, Pini F, Leali F, Secchi C. Survey on human–robot collaboration in industrial settings: safety, intuitive interfaces and applications. Mechatronics 2018;55:248–66.
[44] Simões AC, Pinto A, Santos J, Pinheiro S, Romero D. Designing human-robot collaboration (HRC) workspaces in industrial settings: a systematic literature review. J Manuf Syst 2022;62:28–43.
[45] Arents J, Abolins V, Judvaitis J, Vismanis O, Oraby A, Ozols K. Human–robot collaboration trends and safety aspects: a systematic review. J Sens Actuator Netw 2021;10(3):48.
[46] Kousi N, Gkournelos C, Aivaliotis S, Lotsaris K, Bavelos AC, Baris P, Makris S. Digital twin for designing and reconfiguring human–robot collaborative assembly lines. Appl Sci 2021;11(10):4620.
[47] Wang Y, Feng J, Liu J, Liu X, Wang J. Digital twin-based design and operation of human-robot collaborative assembly. IFAC-Pap 2022;55(2):295–300.
[48] Malik AA, Brem A. Digital twins for collaborative robots: a case study in human-robot interaction. Robot Comput-Integr Manuf 2021;68:102092.
[49] Wang S, Zhang J, Wang P, Law J, Calinescu R, Mihaylova L. A deep learning-enhanced Digital Twin framework for improving safety and reliability in human–robot collaborative manufacturing. Robot Comput-Integr Manuf 2024;85:102608.
[50] Zhang Z, Ji Y, Tang D, Chen J, Liu C. Enabling collaborative assembly between humans and robots using a digital twin system. Robot Comput-Integr Manuf 2024;86:102691.
[51] Li H, Ma W, Wang H, Liu G, Wen X, Zhang Y, Yang M, Luo G, Xie G, Sun C. A framework and method for human-robot cooperative safe control based on digital twin. Adv Eng Inform 2022;53:101701.
[52] Zhu Q, Huang S, Wang G, Moghaddam SK, Lu Y, Yan Y. Dynamic reconfiguration optimization of intelligent manufacturing system with human-robot collaboration based on digital twin. J Manuf Syst 2022;65:330–8.
[53] Ma X, Qi Q, Cheng J, Tao F. A consistency method for digital twin model of human-robot collaboration. J Manuf Syst 2022;65:550–63.
[54] Müller M, Ruppert T, Jazdi N, Weyrich M. Self-improving situation awareness for human–robot-collaboration using intelligent Digital Twin. J Intell Manuf 2024;35(5):2045–63.
[55] Ma X, Qi Q, Tao F. A Digital Twin–based environment-adaptive assignment method for human–robot collaboration. J Manuf Sci Eng 2024;146(3).
[56] Shao G, Helu M. Framework for a Digital Twin in manufacturing: scope and requirements. Manuf Lett 2020;24:105–7.