Journal of Manufacturing Systems 56 (2020) 58–71

Contents lists available at ScienceDirect

Journal of Manufacturing Systems

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

MES-integrated digital twin frameworks

Elisa Negri*, Stefano Berardi, Luca Fumagalli, Marco Macchi

Dept. of Management, Economics and Industrial Engineering, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133, Milano, Italy

ARTICLE INFO

Keywords:
Manufacturing execution system (MES)
Cyber physical systems (CPS)
Digital twin
Industry 4.0
Intelligence layer
Error States detection

ABSTRACT

Industry 4.0-based manufacturing systems are equipped with Cyber-Physical Systems that are characterized by a strong interlinkage between the real world and the digital one: actions in one world have an impact on the other. In this paradigm, Digital Twins (DT) are defined as simulation models that are both getting data from the field and triggering actions on the physical equipment. However, most of the claimed DT in literature are only replicating the real system in a synchronized fashion, without feeding back actions on the control system of the equipment. In literature, these are referred to as Digital Shadows (DS). The paper proposes a way to integrate a DS simulation model with the Manufacturing Execution System (MES) in this way creating a DT. The MES-integrated DT is used for decision making thanks to the presence of an intelligence layer that hosts the rules and the knowledge to choose among alternatives. The paper proposes two frameworks based on the MES-integrated DT: one for managing error states and one for triggering disassembly processes as a consequence of low assembly quality. The DT simulation is developed and integrated with the MES of the Industry 4.0 Laboratory at the School of Management of Politecnico di Milano, where the proposed frameworks have been tested and validated.

Introduction

During the last decade, a rampant process of digital transformation of production plants has taken place, referred to as Industry 4.0 [1,2]. This paradigm is vastly studied and practiced by researchers and practitioners, and it fosters the concept of “smart factory”, a flexible environment that creates the conditions for a highly modular and digitalised production facility [3].

Under the Industry 4.0 paradigm many enabling technologies can be included. These can be used to aid manufacturing companies in many levels, from the corporate aspect to the shop-floor [4,5]. The frameworks proposed in this work are designed for production systems based on Cyber Physical Systems (CPS), that are defined as *physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing communication core* [6]. CPS are also featured by several characteristics like: *real-time capability* to detect any change in the physical process, and to react to them; *intelligence* meant as the capability to identify and sense relevant events [7]. The “cyber” side of the CPS are typically the hosting environment of the so-called Digital Twin (DT), that is defined as a digital copy of a physical asset, being conceived as a system that can replicate, plan, control and directly interact with its physical side [8]. A univocal definition of DT does not exist, and no common understanding concerning this term can be found since it is used differently over disparate disciplines [9]. Thanks to the use of simulation techniques, DT not only are linked to the physical object in the digital world and emulate it, but also include algorithms and sensor-based data gathering in order to provide tools that enable performing experiments in certain environments [10–13]. Overall, a DT typically leverages on the bridging between the digital and real worlds inside the CPS and “*exploits sensed data, mathematical models and real-time data elaboration in order to forecast and optimise the behaviour of the production system at each life cycle phase, in real time*” [14,15]. In alignment to the Industry 4.0 paradigm, it transforms data into a business advantage and added value for the enterprise; all of this in real-time [16].

Another crucial aspect to be discussed lies in the *magnitude of integration* and *connectivity* of a DT in the physical world [17,18]. According to Kritzinger et al., a new perspective based on integration level permits to fully understand how a DT should be defined [9]. In fact, a DT can be either highly or poorly integrated to its physical counterpart, generating three possible levels of digital replicas [9]:

- Digital Model (DM): a digital representation of an existing physical object that does not use any form of automated data exchange between the physical object and the digital one;
- Digital Shadow (DS): it is a DM with an additional automated one-

* Corresponding author.
E-mail addresses: elisa.negri@polimi.it (E. Negri), stefano.berardi@polimi.it (S. Berardi), luca1.fumagalli@polimi.it (L. Fumagalli), marco.macchi@polimi.it (M. Macchi).

https://doi.org/10.1016/j.jmsy.2020.05.007
Received 3 December 2019; Received in revised form 30 April 2020; Accepted 11 May 2020
Available online 25 May 2020
0278-6125/

way flow between the state of an existing physical object and a digital one; therefore, if the state of the physical object changes, then the state of the digital object automatically changes as well, but not vice versa;
- Digital Twin (DT): if the data flow between an existing physical object and a digital one is fully integrated in both directions – from physical to digital and vice versa – the object can be referred to as DT; accordingly, the digital object might also act as a controlling instance of the physical object; therefore, a change of state of the physical object leads to a change in state of the digital object *and* vice versa.

Based on this distinction, most of the publications found in literature that claim to have built DT, actually fall in the DM [19–21] and DS [22–24] categories [9].

The proposed work aims at contributing to this understanding, by constructing a DT according to the definition of Kritzinger [9], using communication protocols in order to connect the digital to the physical side of a given equipment in a manufacturing environment in a full *bilateral* communication. Specifically, the work proposes two general DT-based communication frameworks that are applicable in multiple production facilities. These conceptual models exploit communication protocols in order to integrate the DT and the control layer of the shop floor, i.e. the Manufacturing Execution System (MES). As expected benefits, the application of such a DT-based tool to a production system, if properly designed, helps streamlining production by exploiting the reactive nature that a MES integration functionality provides to feedback actions on the control system of the equipment.

The remainder of the paper is organised as follows. In Section 2, background on DT and MES is provided in terms of their possible applications in the industrial environment; the possibility to integrate the MES into a DT simulation model is also elaborated; the definition of the research objectives of the paper is eventually set out. In Section 3, two MES-DT bilateral communication frameworks are proposed. Section 4 reports the implementation of the DT through a MATLAB/Simulink modelling and the integration with a MES software. Section 5 follows with the application case in the Industry 4.0 Lab at the School of Management of Politecnico di Milano. Section 6 is devoted to discussing the results of the application case and Section 7 sets out some concluding remarks placing the contribution into a wider research perspective.

Research framework

In this section some background information on the use of DT is given together with the possibility to use it in an integrated way with a MES software. This leads to the definition of a set of research gaps that, in the end, will define the research objectives of this paper.

Background information

Being the CPS generally reckoned as a tool that bridges between physical and computing levels of a resource at the same time, [25], its concept must be mapped against the classical Automation Pyramid (ANSI/ISA-95, IEC 62264) [26]. This normative separates a generic manufacturing company information and control systems into 5 different levels – starting from level 0 and ending at level 4 – that compose a structured reference model [27,28].

Monostori et al. [29] suggest a novel definition of the control architecture basing on the characteristics of CPS, that are typically more flexible and knowledge-based [7,30–33]. By exploiting CPS, the hierarchical setting of a production system can be disrupted, enabling the possibility to reach more than one level of the pyramid (up to the point of reaching all five of them) from any data source of the CPS, like depicted in Fig. 1 [34–36,29].

The MES is the level in which the contribution of the present paper is operating. It has been well described by the Manufacturing Enterprise Solution Association (MESA) [37] and in CPS-based systems it embodies a completely new role. The MES can be developed both as proprietary and/or as open-source system and can communicate with external software tools. This whole procedure usually exploits communication protocols that, in turn, use specific programming languages. In the industrial perspective, the main communication protocol that seems to be mostly used to reach the MES from external software packages is the Industrial Ethernet. From a language point of view, several standards exist also regarding the used languages, like eXtensible Markup Language (XML) – that, for instance, run on MTConnect – and Structured Query Language (SQL) [38,39]. This element is crucial for the proposed research work, since the implementation is established on communication between MES and external engineering tools, such as e.g. MATLAB & Simulink (www.mathworks.com), where the DT simulation can be implemented.

DT have in fact the simulation and computing capabilities to support industrial decision making and, potentially, to act on the MES directly. It is worth, then, making a reflection based on the decision-making solutions in literature currently claiming the exploitation of technologies and enablers that fall into the Industry 4.0 paradigm, in particular CPS-based automation, to enhance the MES functionality.

- A first scenario regards solutions that deal with error states, downtimes or quality issues that might affect manufacturing plants. These exploit a *lean manufacturing* paradigm, specifically with respect to avoidance of dead times and scraps throughout all the production flow. Therefore, even if merely theoretical, the possibility to create a middle ground between a MES and a set of *lean* objectives is put forward by Cottyn et al. [40,41]. Indeed, MES is reckoned to be better in acknowledging the real-time perspective, in accordance with the continuous improvement offered by the *lean manufacturing* paradigm: the use of a MES is more appropriate than Enterprise Resource Planner (ERP) systems, that are, in turn, more "batch-oriented" and with less granularity. As a matter of fact, multiple MES vendors have developed *lean* concepts in their software tools. Still, the adoption of a *lean*-based framework reaching the shop floor level – that considers and exploits extensive information coming from CPS – is still lacking [42,43].
- Another scenario that can be studied, in order to have a stronger decision-making capability on production management, is the use of CPS to enhance *human-machine interfaces*. A MES-based measure can effectively regulate the behaviour of employees within manufacturing processes [18]. Furthermore, MES interfaces can be used also for data insertion and data visualisation; this can be validated according to a CPS point of view. In fact, according to Lee et al. CPS have a high level of self-awareness [44]; this translates into a new improved capability for the MES coming precisely from the CPS. Besides another important insight on the topic has been brought forward by Zhou et al. in terms of Human-Cyber-Physical Systems (HCPS) [45]. In fact, in contrast with the traditional view of manufacturing, where a considerable amount of activities is performed by the human, a new "digital" and "digital-networked" manufacturing paradigm is proposed for the future of production environment. HCPS are proposed as a composite intelligent system, with three main actors: humans, Cyber Systems and Physical Systems [46]. This kind of intelligent manufacturing systems are designed to achieve specific manufacturing goals under an optimised point of view. Here, activities such as sensing, analysis and decision making migrate from the human to the CPS, enhancing both work efficiency and human knowledge management, transfer and application [45].
- Other scenarios lie in *re-scheduling* and *re-working* possibilities. At the moment, the execution process inside companies is usually relying on manual activities, leading to low automation of the process [47]. CPS permit a more distributed control architecture where

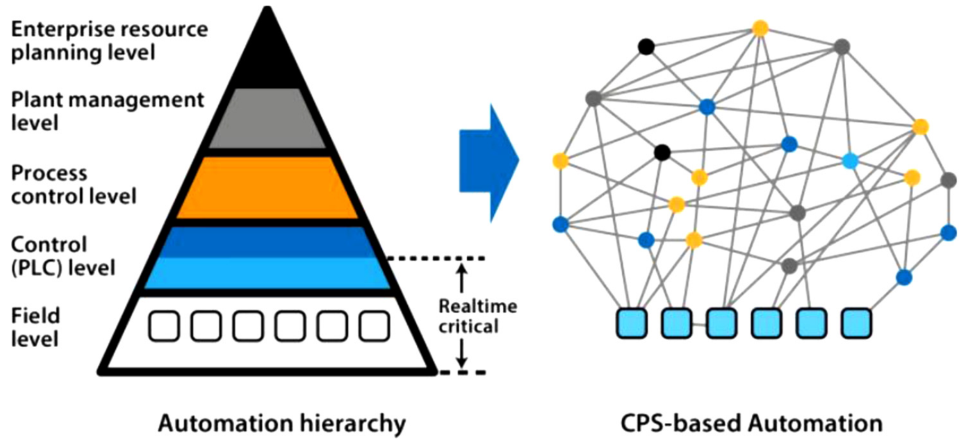

Fig. 1. CPS-based automation pyramid [29].

many resources of a device can be used for multiple tasks inside the facility; subsequently decision making can be performed also at the CPS level [48]. This new vision is also in contrast with the old MES-based central control architecture. Proposals of event-driven re-scheduling have been performed, for the use cases of metal sheet production lines [49].

- Lastly, a further scenario regarding *disassembly* can be brought forward. Endeavours are carried out in research, in regards of the possibility to tackle *high-quality* issues using intelligent techniques [50–52]. Anyway, when in a production or assembly system some specific quality requirements still cannot be met, in order not to scrap materials, some research endeavours address also the possibility to *disassemble* a workpiece. As a matter of fact, the plant itself can be reconfigured for this specific aim, integrating disassembly methodologies and frameworks to enable them into different phases of a product lifecycle [53]. In line with Iarovyi et al., also in this case the use of different devices to enable a distributed CPS-based control architecture can be validated [48].

The works presented in literature that apply the scenarios explained above do not present a capability to actuate any action on the field, for example through the MES, only supporting the monitoring side of the aspect. To this end, other works endow a DT with a direct control decision-making functionality within industrial systems. For instance, Zhang et al. developed a DT-driven methodology for a rapid individualised design of automated flow-shop manufacturing systems [54]. Zhang's work paves the way to possible integrations of a MES to the DT in order to generate detailed work instructions, and also provide a more robust decision-making support [54]. Other proposals come from the research environment to integrate the DT to a concept of "smart factory" in order to cope with operational issues on production lines and to enrich connectivity, proactivity and agility of production, see e.g [54,55].

Autonomy is also one of the core characteristics of a DT, in order to have a deeper understanding and control in the whole product lifecycle, especially in the production phase [56]. As a matter of fact, the challenge is to create autonomous systems that are able to control the production processes and to react to variations during the course of action. In general, several endeavours are addressed in order to combine data during service runs and simulation models in order to have systems that might support operators and planners during operations [56]. This leads to the possibility for a DT to become a more and more integrated system to the production environment. This integration might happen both under a data point of view – for example by using data analytics and deep learning approaches that increase interoperability within the apparatus – or by physically linking the model via sensors etc. Though, research efforts are to be performed in this way for the formalization and experimental setting of this version of the DT [57–59].

Research gaps

On the whole, the just reviewed applications strongly lack in autonomy and real-time supervision capability of production tasks, especially in terms of shop-floor events management *et similia*. To fulfil this gap, more power could be given to a DT by integrating not only the MES, but also several control and decision-making models. Thus, two relevant open questions are addressed by the present work:

1 Which is an appropriate architecture to connect physical and digital levels in order to apply DT for decision-making in an industrial environment?
2 How to enable autonomy and real-time supervision based on the adoption of a DT, integrating an independent and self-governing control decision-making functionality to the current developed models?

The first question relates to the need of a communication in *both directions* with respect to the physical world, as conceived by [9]. The second question assumes that DT are a means to bring the decision-making support, integrating existent models and MES functionality.

Research objectives

Literature has been found to widely lack, concerning the application of autonomous integrated DT with control systems in production environments, as discussed by Cimino et al. [60]. Subsequently, a main objective for the proposed work is here showcased with the aim of fulfilling the topics in which the existent research endeavours fall short.

The *objective of the work* is then creating a DT, able to allow communication between the digital and physical side of manufacturing assets in both directions. In terms of research background, digital models commonly found can be reckoned as *shadows* (i.e. DS) which gather data, but do not have any control or decision-making ability over physical systems and equipment. In spite of this, most of the applications and use cases found in literature fall in this limited category. This paper proposes to overcome this limit with two frameworks that leverage on a digital model that communicates in a *bilateral way* between the digital and physical side of the manufacturing assets. This paves the way for a better and enhanced integration level between the two sides of the CPS, transforming the DS into a DT according to Kritzinger et al. definition [9]. Furthermore, the functionality, that enables the supervisory control over the physical system, should pass through an *intelligence layer*. This entity takes the information coming from the monitoring level and elaborates them in order to be able to make the upper level decisions, acting as a controlling agent. This is all located in an intelligence layer, that is *per se* separated from the DT layer, but that factually enables the controlling feature of the DT, making it possible to close the control loop. The two DT-MES bilateral communication frameworks are implementable in a CPS environment and are later demonstrated through an application at the Industry 4.0 Laboratory of the School of Management of Politecnico di Milano.

Accordingly, the work contributes to enhancing the management of a production system by endowing it with a reactive nature, together with autonomous control and decision-making ability. In particular, comparing with the current scenarios found in the background (Section 2.1), this research work proposes a DT:

- to have a better real-time grasp of what is taking place in the field level, considering the DT as the most suitable CPS-based tool to support such an integration work;
- that enhances human-machine interaction, in compliance with the HCPS theory [46];
- to have event-driven reactive scheduling, with decision making performed on CPS level;
- that proposes integrated disassembly methodologies.

Proposed DT-MES bilateral communication frameworks

This part of the work aims at defining two reactive DT-MES bilateral communication frameworks that work combining the physical and digital level of a certain asset, with the aim of streamlining the production within the manufacturing facility it is placed in. The models are based on CPS capabilities of *bilateral* communication between the physical and digital sides, thus leading to exploit a full DT in Kritzinger's view [9]: this means a DT which forces actions to the MES according to what the intelligence layer defines.

The following features can be pointed out for the intelligence layer:

- it is built to be able to communicate densely and to be integrated with the DT model;
- it is enabling a decision-making ability that is conceptually detached from the DT;
- it is designed with the purpose to make decisions on whether to apply the aforementioned frameworks.

The intelligence layer, and its role with respect to the DT model, are in line with the proposed reference architecture by Valckenaers [61,62].

Framework 1: Error states management

The first proposed framework deals with *error states management* (see Fig. 2). The overall objective for this framework is to deal with error states that stop the production flow by solving them either with a fully automated method or with the help of an operator. The DT and the intelligence layer on top of that, by exploiting the sensors embedded in the line, identify those error states and either solve them right away (error states of type B in the figure) or support the human operator in solving them, if an external intervention is mandatory to resume production (error states of type A in the figure). As it is shown in the figure, there are many levels on which the framework can operate, that are named differently according to their specific nature and their structural function in the framework functioning. However, the overall structure is inspired by the classical CPS structure [44].

The main macro-areas of the framework are the *Physical World* that stands for the actual physical infrastructure, and the *Cyber World.* The latter is made of different and more detailed levels and sub-levels (called layers) that work as follows:

- there is a Decision Making Level, composed of:
- a DT layer where the simulation model is used for detection purposes; as a matter of fact, sensor reading/overwriting takes place in this level;
- an intelligence layer that makes decisions based on the nature of the error states identified in real-time [62];
- the MES integration Level, which takes as inputs the decisions made by the intelligence layer and practically represents the bridge between the DT simulation and the MES; the MES integration level is hosted in the same simulation environment as the DT;
- in the end, the MES Level represents the MES and its functions that are automatically called thanks to the MES integration level.

The flow starts in the physical level with the arrival of the order at a workstation, and a check on whether it is processable or not. If it is not processable, an error state that blocks the production flow arises at physical level. At this point the DT, by reading the sensors embedded in the facility in real time, is aware of what is happening and replicates it in a synchronized simulation model (i.e. the actual DT). At this point the intelligence layer, by receiving the values of a combination of sensors provided by the DT layer, identifies them as a specific error state that is getting the production blocked, according to their specific nature and to embedded rules in this level. The error states can be classified by the intelligence layer according to the necessity of human intervention to solve them (*type A*: human is necessary; *type B*: human is not necessary). The actions of the intelligence layer are diversified according to the type of error state:

- for *type A* error states, the intelligence layer communicates with the MES software through the MES integration level, making it follow a series of actions aimed at helping operators in solving the error (as an example: alert messages on the Human Machine Interface screen – HMI – or automated movement of materials); it is lastly responsibility of the operator to solve the error;
- for *type B*, the error state can be automatically solved by instantly communicating from the DT to the MES – passing through intelligent and MES integration levels – the action to be performed to solve the issue.

In both cases, the DT makes the intelligent layer recognise the type of error that, in turn, triggers the MES to perform the required actions (either as support to human or physical action on field) through the MES integration. Besides, in terms of helping humans in the solving process, this framework proposes a practical translation of the HCPS view [45,46].

Downstream of the error solution, the DT gathers sensors values in order for the intelligent layer to both verify if the processing conditions are met and eventually trigger the MES to resume the production process.

Framework 2: reactive disassembly

The second proposed framework deals with *reactive disassembly* during assembly of products that do not respect specified quality standards. As it is shown in the graphical representation of the framework (see Fig. 3), the same four-level structure of the framework is again exploited.

In order to be applied, this framework needs the presence of processing steps that are carried out by physical resources in the plant:

- an assembly step, made by assembly machines or equipment (e.g. robotic arms);
- a quality check step, made by a resource able to check pieces' quality (e.g. visually, via proper sensors) that are being assembled;
- a disassembly step made by a disassembly machine or equipment that could in certain cases also coincide with the assembly one (e.g.

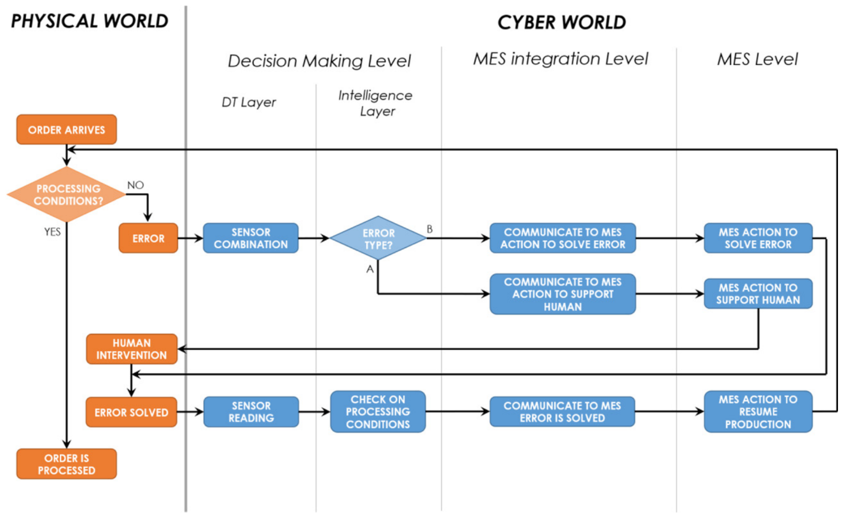

Fig. 2. Error states management framework.

flexible robot arm able both to assemble and disassemble).

Moreover, it is necessary to have specified quality criteria which require the assembled product to be disassembled to rework the piece or to re-use its components.

As in Fig. 3, the DT runs a synchronized simulation in parallel to the physical assembly of products and is continuously updated according to sensors that detect assembled product quality. According to the sensor readings provided by the DT layer, the intelligence layer understands – through its embedded rules – whether the quality standards are met or not. In the first case, the assembly process can proceed, while in the second case the subsequent set of actions are automatically triggered by the intelligence layer. Specifically, the DT must collect information about the order in question, for instance it understands which is the wrongly assembled component. Then, the MES is automatically triggered to abort the current assembly order, and automatically schedules a customised disassembly order on the aborted product.

Simulation models for digital twins

Both aforementioned frameworks can lead back to an architecture that shows how the DS and the DT interact with other levels including also the computational models that have been developed in order to work properly. The idea is represented by a block-like schema (see Fig. 4), where the functions at the basis of the model are shown. In Fig. 4, like the legend suggests, the colour of the blocks describes their nature. Blue blocks refer to computational and digital entities; the orange block stands for physical resources. By grouping the elementary blocks/functions differently, a DS or a DT can be generated. The following list describes the blocks in the Fig. 4.

- *DM–Digital Model*, that is at the basis of the DS itself. This is a simulation model designed to read and overwrite in real time sensors values that are embedded in the production facility (i.e. the field devices). However, as long as this block is not connected to the field devices, it is nothing but a digital copy of the physical resource that does not vary its status in an autonomous way, consistently with Kritzinger's definition [9].
- When the DM is linked to the physical resources, like in Fig. 4, a *DS–Digital Shadow* is built. This is characterized by a real-time reading of what is taking place and therefore an automated and real-time synchronization of the simulation with respect to the physical status changes (one-sided communication). This model is explained in Section 4.1.
- The DS communicates to an intelligence layer, that is the place where higher-level decisions are made, about the actions to be taken in order to react to physical events. The intelligence layer takes as an input a combination of sensors values readings coming from the DS. This allows the identification of a set of states that automatically trigger a set of solving techniques that were previously studied,

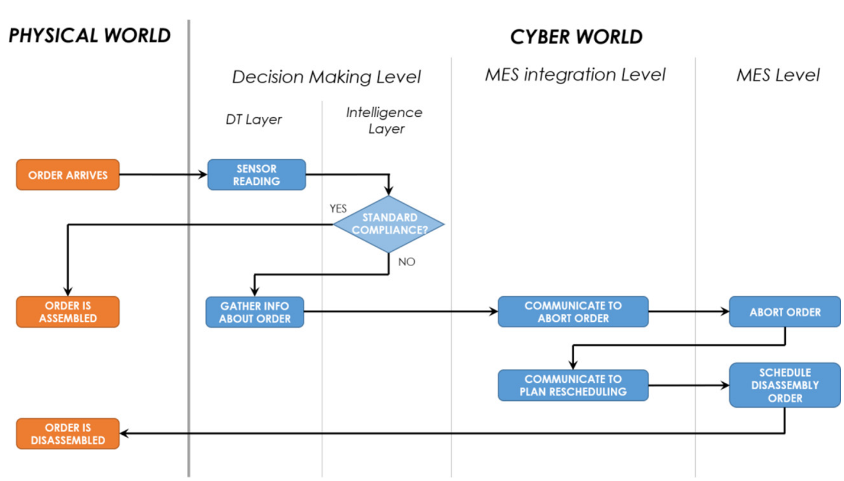

Fig. 3. Reactive disassembly framework.

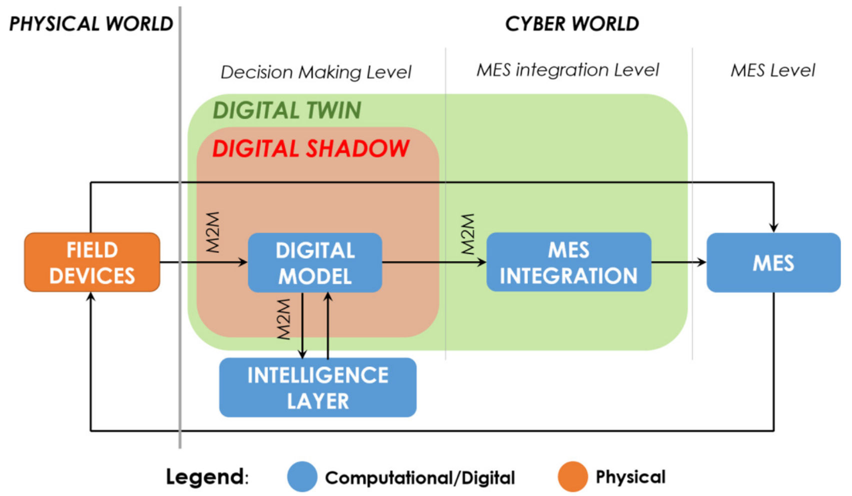

Fig. 4. DT vs. DS model basic architecture.

modelled and embedded in the code. This kind of decision making is the core of the intelligence layer and can be reckoned as its output.

- The DS communicates with the MES through the MES integration level. By integrating this connection with the DS, a *DT–Digital Twin* is created. In particular, the *MES integration* function is used in both the error states management framework and the reactive disassembly one. It is designed to communicate with the MES software in order to manage the communication with the MES and to trigger the desired MES behaviour that is decided by the *intelligence layer*, in order to solve the incurred error (framework 1) or to force the customised disassembly of the assembled product, downstream to the occurrence of quality discrepancies that the model is designed to autonomously identify (framework 2). This functionality is described in Sections 4.2 and 4.3.

The novelty proposed by the two reactive DT-MES bilateral communication frameworks lies in the fact that the DT can control the physical system, according to rules embedded in the intelligence layer (i.e. forcing it to do certain actions through the MES). The proposed functionalities of the DT do not only give the possibility to create a *bilateral* communication, but also streamline production, decreasing considerably the impact of error states and production quality issues.

The chosen simulation software to implement the DT is *Simulink*, following the methodology by Fumagalli et al. [63], that allows to exploit the MATLAB coding embedded capabilities in order to model the system and to mimic the behaviour of the manufacturing assets as physical resources in question.

Communication with the field

According to Fig. 4, the first step for the development of a DT is the creation of the DS, which implies communication between the field level of a production facility and a simulation model (which is the simple DM).

This model can be created by exploiting a communication methodology that essentially extracts in real-time the values of the sensors that are embedded in a certain facility. The connection between the two levels presented in Fig. 4 may be done with communication protocols like Open Platform Communication (OPC) Unified Architecture (UA), or with other machine-to-machine (M2M) protocols such as Robot Operating System (ROS), Message Queue Telemetry Transport (MQTT), Data Distribution Service for Real Time Systems (DDS), Transmission Control Protocol (TCP)-based [64,65]. The choice of a protocol depends on the features of the physical equipment that the DS mirrors [66].

These protocols enable the real-time reading of values, thanks to the use of specific Simulink MATLAB functions. Specifically, in Simulink, the choice falls in the use of *Level-2 MATLAB S-Functions*. This specific function allows to use the MATLAB language to create custom blocks with multiple input and output ports, which can process any type of signal produced by a Simulink model. The function itself comprises a set of callback methods that the Simulink engine invokes when updating or simulating the model (https://it.mathworks.com/help/simulink/sfg/writing-level-2-matlab-s-functions.html). In this specific case, the *Level-2 MATLAB S-Functions* are used in the simulation environment to read, use and overwrite the values of the sensors of interest also in terms of identification of machine states, such as idle, working, error etc. [60]. The function, thanks to the use of OPC UA protocol, is able to create inside the simulation environment, a *client/server* connection, allowing the model to know the value of a certain sensor at a given time. Inside Simulink environment, the *Level-2 MATLAB S-Function* is represented on the left-hand side of Fig. 5. The code inside the block function is built as on the right-hand side of the Figure. The nature of the code is modular and highly customisable; as a matter of fact, the user can choose whichever number of input and output ports in order to respectively overwrite and read the sensors values. As the graphical part of Fig. 5 suggests, the output sensors values (i.e. the names of the sensors are '*CarrierID*', '*xBG1*', '*xQA1_A1*'…) are linked, combined and then exploited for subsequent scopes in the remainder of the simulation tool. The ways in which these operations are carried out is explained in the next paragraph.

Specifically, a DS simulation model with embedded field-connections through the *Level-2 MATLAB S-Function* has been fully designed and proposed by Negri et al. [67]. More specifically, an automatic update of the *machine states* of the shop-floor equipment, starting from the identification and combination of the values of the sensors on the production system is provided. This specific functionality of the work has paved the way for the implementation of the work proposed in this paper. Besides, in the work of Negri et al., additional features have been provided to the DS, such as calculation of *energy consumption* of the physical resources, and some specific *KPIs* of the production system; for more details, please refer to [67].

The MES integration level of the Digital Twin

By integrating the DS to the MES, it is possible to talk about DT since communication becomes bilateral between physical and digital sides, [9], allowing the latter to take corrective actions when the physical side is in need.

This additional capability can be given to the DT always by exploiting *M2M* communication protocols that, in turn, are based on

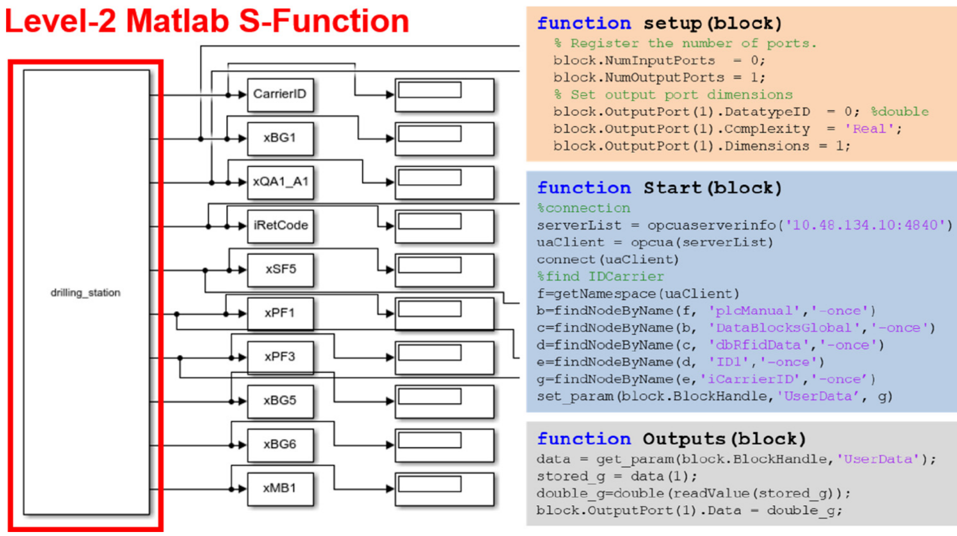

Fig. 5. Level-2 MATLAB S-Function (left-hand side); Related code (right-hand side).

certain languages, that may either be proprietary or open source (i.e. XML). These protocols allow external sources to communicate with software tools; in this peculiar case, MES software tools. In the present work, according to the schema proposed in Fig. 4, based on events that are taking place in the physical level the DT will interact with the intelligence layer to know what should be done and then will autonomously communicate it to the MES.

The protocols in question make it possible for the DT to either request or send information to the MES server, opening the possibility to create a communication channel between the two levels.

However, communication between the simulation environment and the MES software needs a precise language and syntax. The protocols used by the vendors of industrial equipment can either be proprietary or open source, in order to aid integration with other instruments. In both of these cases, commands must be sent via communication channels like TCP/IP or Industrial Ethernet [38,39]; in turn, commands can be sent to the MES server using strings or similar syntactic structures. Information stored in the exchanged commands need to be properly parsed in order to be useful for the implemented algorithms.

The Digital Twin simulation model with the MES integration

As presented in Sections 4.1 and 4.2, the proposed DT may read sensors values and eventually overwrite them (i.e. via OPC UA) and communicate autonomously with the MES software when needed (i.e. via TCP/IP).

Based on these channels, Simulink function blocks can be properly designed to tackle technical issues on the shop-floor. In fact, real-time synchronized simulated sensors can be used to identify "*trigger states*" that, in turn, – according to the rules dictated by the intelligence layer – activate the MES communication channel by creating communication ports to send commands to take the desired measures. In other words, these trigger states basically enable the intelligence layer in a computational way, in order for it to decide the solving measure to be applied; this becomes then an actionable information to be sent through communication ports to an external software tool as MES, in order to finally take the measure in the physical world.

In Simulink environment, the modelling of the MES-integrated DS, that is now referred to as DT, is graphically represented like in Fig. 6. The left-hand side of the Figure is featured with the *Level-2 MATLAB S-Function*, typical of the DS functions previously described (in section 4.1). On the right-hand side, instead, the MES integration functionality stands for the computational link between the digital model and the MES software. Specifically, the inputs to the MES integration functionality block are the sensors that are needed to detect a specific error state (in case of Error states management framework) or a specific non-compliance with quality standards (in case of Reactive Disassembly framework), while the outputs are specific commands to be sent to the MES server. As it can be seen in the Figure, a dense web of branches between the simulation blocks related to the sensor values determine the aforementioned set of "*trigger states*". Both this aspect, and the related code will be described in the Application Case paragraph.

This whole proposed model has been in fact validated in the application case that is showcased in the next section, together with the reached results coming from the application of the reactive DT-MES bilateral communication frameworks.

Application case

A proof of concept of the proposed DT model has been developed at the Industry 4.0 Laboratory of the School of Management of Politecnico di Milano. The laboratory is endowed with a prototypical assembly line, used for research and didactic activities [68].

The assembly line in question is configured and installed with a mix of modular resources and machinery by *Festo*, *Mitsubishi Electric*, and *Siemens*. The line assembles a simplified mobile phone, that can be tracked in every processing step thanks to RFID-tagged carriers that follow a standard conveyor path. The production line is embedded with a set of sensors, for each module of the line, that can be reached and read in real-time via OPC UA protocol.

The production is integrated with two servers that connect the line to a MES software and to an energy monitoring platform. The MES in question is called *MES4* and it is provided by *Festo*; this makes it a proprietary software. Besides, its server can be reached from external software tools using a string-based communication protocol that runs on TCP/IP channels [69].

The digital twin in the laboratory

The line is made of seven workstations, each dedicated to one or more assembly steps for the mobile phone. The stations are here listed, following the numeric reference of Fig. 7: *Manual (1)*, *Front Cover (2)*, *Drilling (3)*, *Robot Assembly (4)*, *Camera Inspection (5)*, *Back Cover (6)*, *Press (7)*.

DT of the single workstations were modelled in *Simulink* environment, and then put into the overall DT of the production line, as shown in Fig. 8 and as described in detail in [67]. As already explained in the Section 4.1, communication with the field has been granted by using

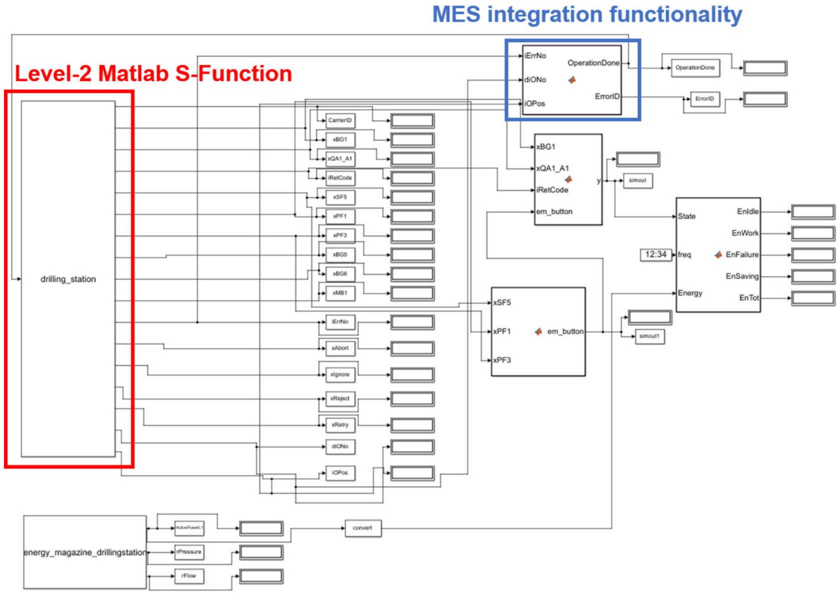

Fig. 6. DT model in Simulink environment.

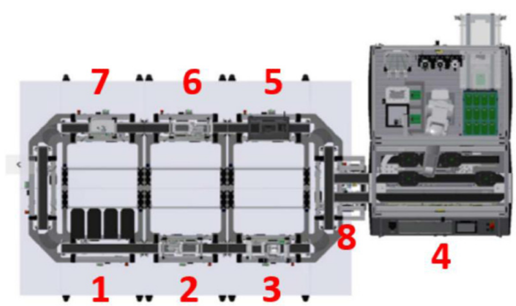

Fig. 7. Plant view of the assembly line of the I4.0 Lab.

the OPC UA communication protocol. All of the sensors of the line are, in fact, accessible by means of this protocol, and they were thus read and overwritten thanks to the *Level-2 MATLAB S-Function* that has been modelled in order to be able to communicate with the sensors by using OPC UA protocol.

More specifically, to respect the communication channels relative to Fig. 4, the work proceeded as illustrated in the reminder.

- For the sensors data reading and overwriting made by the DS, the *Level-2 MATLAB S-Function* has been used and modelled following a structure that enables the creation of *input/output* ports to which a specific sensor value was associated. The inputs allow the overwriting of sensors; the outputs, in turn, make it possible to read the sensors values. Accordingly, Fig. 9 shows graphically how the block-based model is made. On the other hand, the code respects the structure shown in Fig. 5.
- For the *MES integration* functionality, exploiting the sensors values coming from the DS, a set of "*trigger states*" has been identified for each workstation. When one of these trigger states is active, according to the rules decided in the intelligence layer, a connection channel is automatically created with the MES software through the creation of a TCP/IP object. A string-based command related to the specific trigger state is sent to the MES, exploiting the just created communication channel. More specifically, trigger states were identified to apply both DT-MES bilateral communication frameworks, i.e. for error states management and for reactive disassembly.

Based on this, the trigger states of the *Robot Assembly* workstation are shown in Table 1. The sensors that are being considered in this

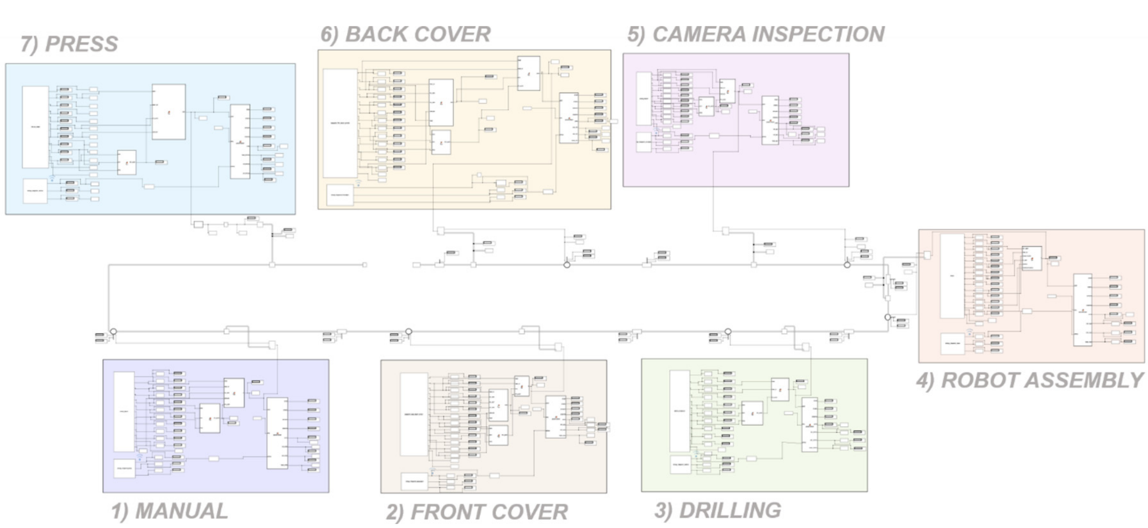

Fig. 8. DT simulation model for a production facility.

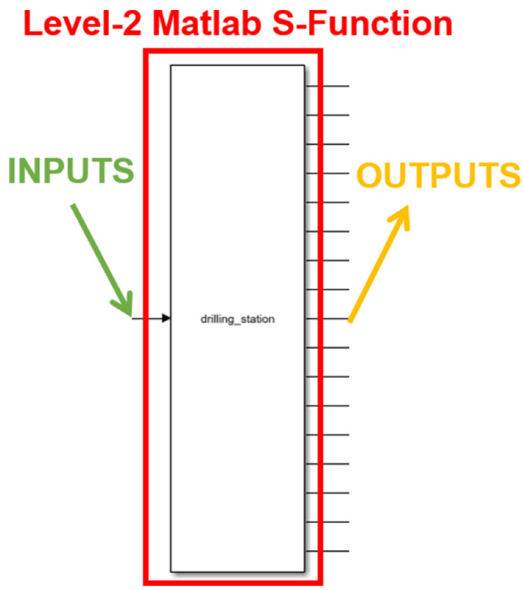

Fig. 9. Level-2 MATLAB S-Function inputs/outputs example on drilling station.

Table 1
Example of Trigger states IDs for the Robot Assembly workstation.

Error ID – Trigger state	xG1_BG50	xG1_BG51	xDisableMenu
5	0	0	1
6	1	1	1
7	0	1	1

workstation, with their relative meaning, are the following:

- *xG1_BG50*: it senses the pallet at the working position;
- *xG1_BG51*: it senses the front cover at the working position;
- *xDisableMenu*: it indicates when an error is being displayed at the HMI.

Basically, for each combination of sensors on the columns, an error type is identified: the trigger states are the corresponding computational side of the error states, that are therefore read by the DT.

Similar tables have been elaborated for all the other workstations of the assembly line.

In computational terms, Table 1 translates in the extract of code shown in Fig. 10: when the condition is met, the communication port is opened, and the string relative to the solving technique is sent to the MES. Then, right after solving the error – either automatically or by human intervention – the output values to the MES integration block are looped back to the *Level-2 MATLAB S-Function* in order to overwrite the sensors values that hold the piece in working position. Once these sensors are overwritten, the pallet is released, and the workstation is cleared.

Similarly, the reactive disassembly framework exploits the value of a specific sensor to be triggered. More specifically, the sensor *'xResult'* of the Camera Inspection workstation indicates if the assembly process is compliant with the workplan so far (i.e. it senses the presence of a

```
if (xDisableMenu == 1)
    if (xG1_BG50 == 0 & xG1_BG51 == 0) %condition in which we have PCB missing
        ErrorID = 5;
        t=tcpip('10.48.134.1',2000);
        fopen(t);
        fprintf(t,sprintf('444;RequestID=0;MClass=150;MNo=71;ErrorState=0;#MaxRecords=
```

Fig. 10. Extract of code that verifies the "trigger states" in the MES integration block of the Robot Assembly.

Table 2
Possible values of the xResult sensor, with relative meaning, at the Camera Inspection workstation.

xResult	Meaning
0	*The workpiece is* <u>*not*</u> *compliant with the workplan*
1	*The workpiece is compliant with the workplan*

```
if(xResult == 0)
    t=tcpip('10.48.134.1',2000);
    fopen(t);
    fprintf(t,sprintf('444;RequestID=0;MClass=101;MNo=21;ErrorState=1;
                        #ONo=%d;#OPos=%d\r',diONo,iOPos));
    if(diONo~=0)
        t1=tcpip('10.48.134.1',2000);
        fopen(t1);
        fprintf(t1,sprintf('444;RequestID=0;MClass=101;MNo=2;ErrorState=0;
                            #PNo=1231;#Aux1Int=1\r'));%set new order
```

Fig. 11. Extract of code that verifies the workpiece compliance in the MES integration block of the Camera Inspection.

fuses, if the workplan includes the assembly of fuses).

As in Table 2 and in Fig. 11, if the sensor *xResult* is 0, the workpiece is not compliant with the workplan, so two strings are sent to the MES: the first one to abort the current assembly workplan; the second one to schedule a customised disassembly order to that specific workpiece.

To grant a common understanding between the MES server and the *Simulink* model, a proper parsing technique has been developed. In fact, information can be either sent or received from the MES to the DT. If the DT just sends information to the MES when certain conditions are met, no parsing is needed. On the other hand, if the DT sends an inquiry to request information to the MES, the latter stores it back in a string form. Subsequently, the string needs to be parsed for the information to be readable in computational terms by the DT. For this aim, MATLAB functions like *textscan* and *cell2mat* were used in the code.

Application of the DT-MES bilateral communication frameworks

The application of the *error states management* framework consisted in the identification of a set of eleven error states that stop production, for all the workstations in the production line. Each of these states can be, in turn, separated into two sub-categories, according to the need of human intervention to solve the issue; the *type A* error states do need human intervention for solving, while the *type B* error states can be fully solved in an automated way. The error states found in the assembly line and subsequently considered are summed up in Table 3.

Each one of these errors was identified by using a combination of sensors, in order to univocally understand whether an error is occurring. For instance, with reference to Table 1, the ErrorID 5 is taking place when *'xG1_BG50'* and *'xG1_BG51'* sensors are equal to 0 and the *'xDisableMenu'* sensor is equal to 1. With the DT running, when one of these errors occurs, it is tackled right away through the error states management framework.

For *type A* error states, the DT supports the operator in charge of

Table 3
Error states list.

Error ID	Machine	Description	Type
1	*Front cover*	Buffer of front covers is empty	A
2	*Front cover*	Front cover already assembled on the piece	B
3	*Front cover*	Pallet not available on the carrier	B
4	*Drilling*	Front cover not assembled on the piece	B
5	*Robot assembly*	PCB not available in the PCB Box	A
6	*Robot assembly*	Pallet not available on the carrier	B
7	*Robot assembly*	Front cover not assembled on the piece	B
8	*Back cover*	Front cover not assembled on the piece	B
9	*Back cover*	Buffer of back covers is empty	A
10	*Back cover*	Back cover already assembled on the piece	B
11	*Press*	Back cover not assembled on the piece	B

Table 4
Framework 1, type A error example.

Level	Action
Physical Level	The carrier arrives at the Robot Assembly and production blocks.
DT layer	The DT identifies the combination of sensors that are read in real-time. The error is taking place when *'xG1_BG50'* and *'xG1_BG51'* sensors are equal to 0 and the *'xDisableMenu'* sensor is equal to 1. The first sensor is 0 if there is a pallet in working position; the second is 0 if the front cover of the smartphone is detected on the pallet in working position. The last sensor is 1 if there is an error message displayed on the HMI.
Intelligence layer	The intelligence layer acknowledges that an error state is happening and associates it to a specific classification and triggers a solving technique to be operated. In this case the error is due to the lack of PCB in the physical box, that must be refilled by the operator (error of type A).
MES integration level + MES level	The simulation model follows the instructions given by the intelligence layer and automatically triggers the button that expels the physical PCB box from the Robot cell, thanks to sensor overwriting, and fills up the virtual PCB box on the MES platform. (For the sake of completeness, it must be said that, in addition to the physical PCB box, there is a virtual PCB box encoded on the MES software, that gets emptied when a physical PCB is used for assembling a piece, and needs to be filled up manually on the MES software when the physical PCB box is filled. In this case, one of the MES integration actions done by the DT is the automatic filling of the virtual PCB box.)
Physical level	The physical PCB box is autonomously expelled from the Robot cell, and the operator notices more easily that the production got blocked. Also, the operator fills up the physical PCB box, and presses the button to put it back in working position, inside the Robot cell.
DT layer	After, the DT updates in real-time the value of the sensors.
Intelligence layer	This level checks if the PCBs have been refilled and that the physical box is back in working position.
DT layer	When the intelligence layer states the proper working conditions are met, the DT communicates to the MES to resume the work order that has been idling the Robot.
Physical level	The work order resumes the assembly process with the newly filled PCBs in the box.

controlling the line to notice the error states by making it more evident. When the error state condition is solved by the operator, the DT autonomously makes the MES resume the production.

For instance, the *error state number 5* consists in the lack of Printed Circuit Boards (PCB) in the PCB box of the *Robot Assembly* workstation. In this specific step the assembly process includes the placement of a PCB inside the front cover of the smartphone; the PCB are stored in a physical box inside the Robotic cell, called PCB box. If the box is empty at the assembly moment in the Robot workstation, the processing step interrupts, and cannot resume until the box is filled. When this peculiar error state occurs, based on the five levels of the previously described framework (cf. Fig. 2), the DT takes actions in an automatic fashion, like in Table 4.

For the *type B* error states, the DT solves the error in full autonomy by forcing the MES to clear the workstation by ignoring the production step (if the assembly step was already performed) or to abort the order in question (if the production step was not performable). As an example, in *error state number 4*, when the pallet arrives at the Drilling station to perform the drilling operation, the production interrupts since there is no front cover present on the pallet due to assembly errors upstream on the line. The way the DT tackles this kind of error is explained in the list reported in Table 5.

Notice that, in case the order is aborted, all the phases of the branch downstream to the "Error Solved" block in the Framework 1 (cf. Fig. 2), are skipped.

For the *reactive disassembly* framework, the DT exploits two particular workstations: the *Robot Assembly* and the *Camera Inspection*. Table 6 reports the actions performed by the five levels of the Framework 2 (cf. Fig. 3).

Results and discussion

A set of simulation runs were performed to quantify the magnitude of the impact of the application of the DT in regards of both frameworks. In fact, specifically for the *error states management*, the magnitude can be effectively computed by measuring the mean downtime before and after the application of the reactive DT-MES bilateral communication framework.

Both 6.1 and 6.2 paragraphs are aimed at validating and discussing the fulfilment of the objective of the work stated at the beginning of the paper. As a matter of fact, the formulation of the MES-DT bilateral communication frameworks with related validation has paved the way to the creation of a DT. This model reaches a reactive nature and an autonomous decision-making capability thanks to the formulation of an intelligence layer that is conceptually separated.

Results and discussion on Framework 1: Error states management

The performed simulation runs are here reported and described; they are meant to prove that, thanks to the application of the DT-MES bilateral communication framework, the mean downtime for each of the eleven error states is reduced. Thus, the chosen KPI to prove this is the *percentage downtime reduction*. For each one of the error types, a total of 60 observations on the real line equipment has been performed, in order to be able to calculate the percentage downtime reduction: 30 of these observations have been carried out as in the Scenario 1, and 30 as in the Scenario 2. These are described in the list below:

- SCENARIO 1: running of work orders *without* the application of the

Table 5
Framework 1, type B error example.

Level	Action
Physical Level	The pallet arrives at the working position at the Drilling station and production gets blocked since there is no front cover.
DT layer	The DT automatically provides a sensors combination in order for the intelligence layer to interpret them. In this case if the *'xCL_BG4'* is equal to 1, and *'iErrNo'* is equal to 4, the error is taking place. The first sensor detects whether the front cover is on the pallet, when in working position, and it is equal to 1 when there is no front cover. The second sensor is equal to 4 when the HMI is displaying an error message.
Intelligence layer	The sensors combination provided by the DT is elaborated and associated automatically to the solving action. In this case the order in question must be aborted and, for that specific semi-assembled product, a customised disassembly order must be planned.
MES integration level + MES level	The DT communicates to the MES to abort the work order, since there is no way to resume the assembly of the piece in a compliant way.
Physical level	The workstation is cleared from the aborted work order, and production of other pieces can resume.

Table 6
Framework 2 example.

Level	Action
Physical Level	One of the sensors of the *Camera Inspection* workstation acknowledges that the piece is not compliant to certain standards.
DT layer	The DT provides the sensor value to the intelligence layer.
Intelligence layer	The sensor value provided by the DT is elaborated and associated automatically to the solving action. In this case the order in question must be aborted and a customised disassembly order must be planned.
MES integration level + MES level	The DT communicates to the MES to abort the work order and to reactively schedule a disassembly order on that piece when the carrier passes again through the *Robot Assembly* station (*N.B.* the conveyor belt is a loop in the laboratory).
Physical level	The workstation is cleared from the aborted work order, and production of other pieces can resume.

DT-MES bilateral communication framework; error states in this scenario are tackled only thanks to the intervention of an operator.

- SCENARIO 2: running of work orders *with* the application of the DT-MES bilateral communication framework; error states in this scenario are solved either thanks to the operator warned through the HMI communication by the proposed framework (for type A errors) or in a fully automated DT-MES action (for type B errors).

All of the observations in question have been carried out in conditions of actual malfunctioning of the production line. This made it possible to compare how the line behaved with and without the proposed DT-MES bilateral communication framework, under the same starting conditions, and to compare the mean downtime in the two scenarios.

The results of the tests are summarised in Table 7. The Mean Downtime Reduction has been calculated as the reduction of the Mean Downtime by comparing the time to solve the error state in Scenario 1 with the one in Scenario 2. As it can be acknowledged from the table, the time spent in each of the error states was greatly reduced thanks to the application of the DT-MES bilateral communication framework.

Errors of *type A* have a downtime percentage reduction lower in magnitude with respect to those of *type B*. This is due to the fact that the unpredictability of human behaviour is still relevant in *type A* errors. In fact, the specific task to be performed by the operator to solve a *type A* error can be executed in a variable and occasionally long time, affecting the stopping time of the production line, and subsequently downsizing the reduction of the error state time. The occasional long time of the humans to solve manually the error also leads to a high variance. That is why a rough quantification of both variance and skewness of the out-coming statistical distributions of mean downtime for type A errors is provided in Table 6. The out-coming statistical distributions for *type A* errors have a positive skewness, hence they are asymmetrical to the left. This further validates the long time that sometimes is required to solve the error by the operator; the graphical representation of the downtimes of *Scenario 2* for ErrorID = 5 is given in Fig. 12.

On the other hand, for *type B* errors, the error is solved in a very short time, that is almost deterministic for all of them. This is because

Table 7
Mean downtime reduction for the error states.

Error ID	Type	Mean downtime scenario 1 [s]	Mean downtime scenario 2 [s]	Mean downtime reduction [%]	Variance	Skewness
1	A	16.28	12.12	25.55	22.6	0.42
2	B	12.86	1.59	87.54	–	–
3	B	13.23	1.39	89.53	–	–
4	B	13.86	1.69	87.83	–	–
5	A	66.75	27.00	60.64	101.3	0.86
6	B	14.89	2.40	83.83	–	–
7	B	11.84	2.10	82.18	–	–
8	B	14.39	2.09	85.55	–	–
9	A	19.74	13.96	29.27	47.7	1.5
10	B	13.05	2.09	85.55	–	–
11	B	13.81	2.13	84.59	–	–

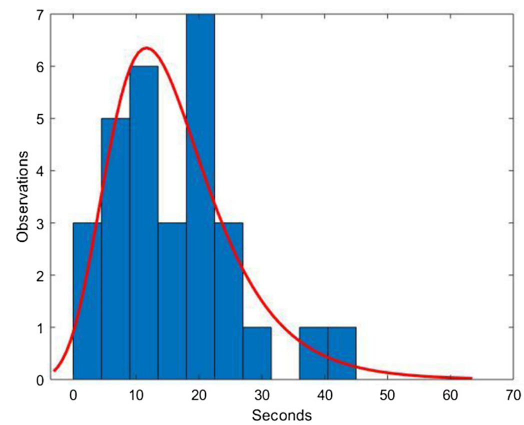

Fig. 12. Histogram of the solved downtimes of ErrorID no. 5.

the computational time to process the solving operation by the DT is almost the same for all the analysed error states. Furthermore, for all the *type B* error states, because of their semi-deterministic solving time, no indication of variance or skewness of the solved scenario results was interesting to report.

Overall, the application of this framework to the laboratory case grants the possibility to streamline the production and contextually reduce the dead times that may occur during production and, in turn, the overall lead time of production as well. Everything is performed in an autonomous and reactive way by the DT framework. Also, a better *human-machine* interaction is proven, in line with the path opened up by the HCPS studies [45,46]. Peculiarly, in the Error State Management framework, the improvements were validated by successfully managing to make the error states more noticeable to the eyes of the operators in charge of controlling the line.

Discussion on framework 2: reactive disassembly

The main objective and benefit of this framework is the ability of the proposed DT to reactively schedule disassembly work-plans, according to quality non-compliances. The validated industrial outcomes of the application of this framework can be summarised as follows:

- the possibility to react to shop-floor events that, differently from the error states framework, do not block the production flow;
- the possibility to avoid useless and costly further assembly steps on a defective product, provided thanks to the ability to immediately notice the quality error, and automatically abort the order;
- the possibility to reactively schedule the disassembly of the mis-assembled component and the assembly of the new component in substitution of the mis-assembled one;
- the possibility to place back for re-use the disassembled component, in a quick way in order to avoid non-added value work in progress in the shop-floor.

In practical terms, the *lack of fuses* in the PCB assembly on the phone was an example of quality problem that was occurring at the lab operations and that was managed by the proposed framework that enabled to detect all such cases. In fact, if the *Robot Assembly* station does not assemble a planned fuse inside of the PCB for any reason, and the assembly continues nonetheless, the carrier arrives at the *Camera Inspection* station. Here the DT-MES integration proposed framework identifies the ongoing non-compliance in the assembly process and aborts automatically the order; therefore, the scheduling procedure is done immediately in real-time since there are no computational burdens. So, the customised disassembly order is associated in an automated fashion to that specific carrier.

Benefits coming from the application of this framework are measurable in terms of *lead time* as a result of downtime reduction. For a given work order inside the assembly line, in fact, lead time is for certain reduced. As a matter of fact, if incompliances are autonomously detected and reactively managed via scheduling of disassembly tasks, the work orders will spend for sure less time inside the production plant.

Besides, in the totality of the cases, by applying the reactive disassembly framework, disassembly is planned for pieces that do not comply with the standards.

Conclusions

The objective of the research work was to elaborate a *DT – Digital Twin*, starting from a *DS – Digital Shadow,* since the majority of the claimed DT in literature do not present the bilateral communication of information from physical to the digital worlds and vice versa, limiting them to be simple DS in the paradigm by Kritzinger [9]. The objective has been reached by connecting the CPS-synchronized simulation (i.e. the DS) to the MES, in this way closing the control loop and achieving a real DT [9]. Therefore, the DT not only mimics the behaviour of the real physical asset, but also enables autonomous and real-time *two-way* communication between the physical and digital sides, thus turning the bilateral communication into action by triggering some actions on the MES. This capability translates into the possibility to not only monitor the physical asset in real-time, but also to react to shop-floor events that might affect the supervised production environment.

The proposed contributions in this regard are two DT-MES bilateral communication frameworks that can be used to streamline production by proper decision making.

- The *error states management* framework exploits the DT to solve error states that might affect a manufacturing facility. This requires the MES integration functionality, besides the identification of the error state and the decision on the way to solve it. This framework enhances also the human-machine interaction, based on the view proposed by Zhou et al. [45,46].
- The *reactive disassembly* framework schedules disassembly orders for assembled products that do not comply with quality standards. This is applicable only if the facility is endowed with tools that can assemble, control, and disassemble work pieces.

In both frameworks the decision making relies on an external layer strictly connected to the DT, the so-called intelligence layer that embeds the rules and the knowledge to select the proper action to be done and have the DT triggering it to the MES through its bilateral connection.

The proposed frameworks were validated in laboratory environment, for a didactic assembly line located at the Industry 4.0 Laboratory of School of Management of Politecnico di Milano. Here the running of a tailored DT, modelled and implemented following the proposed frameworks, makes it possible to improve production performances. In fact, the mean downtime due to a set of identified error states regarding the assembly line is proven to be considerably reduced thanks to the DT integration with MES that, when a certain error state is taking place, sends command strings directly to the MES to tackle the issue. Also, the possibility to reactively schedule disassembly programs is validated by exploiting a robotic arm that can perform both assembly and disassembly actions – if properly configured – and a camera that checks on assembly standards.

This work contributes to the research on DT by proposing examples of the possible decision making that can be obtained using the combination of DT and intelligence layer. The proposed contribution wants to claim the role of the DT into the decision-making frameworks. In particular, the intelligence (i.e. the rules and the knowledge to choose between alternative options) is hosted on a separated layer with respect to the DT, while the DT remains a replica that is able to communicate in two ways with the physical world. For more details on this, the ARTI (Activity Resource Type Instance) architecture proposes a very interesting perspective on the topic [62]. This may also pave the path for distributed DT architectures, composed of various DT modules. The single DT modules may be hosted by the single physical resources and may be used for various functionalities, mainly related to a local decision making (e.g. monitoring and forecasting of the behaviour of the physical twin [61] or may be used as a means to populate data repositories to be used on higher levels [70]). When integrating the various DT modules into a system architecture, this could be used for a global decision making and integrated into a higher-level control system, such as the MES [71,72].

Future research directions may investigate the distributed DT architectures and the related local and global decision making that they enable, assessing their impact on production and on further decisions to be made on the operations. Another future direction may enlarge the basis of functionalities of the DT and intelligence layer, through other frameworks of integration with the MES; some examples are (i) the reactive scheduling of orders that embed maintenance actions according to predictive models elaborated through the data available from the DT synchronization with the field, in line with [73]; (ii) the monitoring of energy consumption anomalies of the production operations that may be considered as symptoms of future failures or quality deviations, or may offer a basis to evaluate actions to reduce environmental impacts; (iii) the use of DT during the production phase of a product to ease the end-of-life of the product itself (e.g. the DT of the assembly process may encounter phenomena that may impact the disassembly of the same product).

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 has been possible thanks to the use of the Industry 4.0 Laboratory of the School of Management of Politecnico di Milano (lab site: www.industry40lab.org). This work was supported by the Ministry of Education, University and Research (Italy), in relation to the project “Smart Manufacturing 2020” (grant number CTN01_00163_216744).

References

[1] Qin J, Liu Y, Grosvenor R. A categorical framework of manufacturing for industry 4.0 and beyond. Procedia Cirp 2016;52:173–8. https://doi.org/10.1016/j.procir.2016.08.005.
[2] Jazdi N. Cyber physical systems in the context of industry 4.0. Autom. Qual. Testing, Robot. 2014 IEEE Int. Conf. 2014:1–4. https://doi.org/10.1109/AQTR.2014.6857843.
[3] Castelo-Branco I, Cruz-Jesus F, Oliveira T. Assessing Industry 4.0 readiness in manufacturing: evidence for the European Union. Comput Ind 2019;107:22–32. https://doi.org/10.1016/j.compind.2019.01.007.

[4] Zheng P, Wang H, Sang Z, Zhong RY, Liu Y, Liu C, et al. Smart manufacturing systems for Industry 4. 0 : conceptual framework, scenarios, and future perspectives. Front Mech Eng China 2018;13:137–50.
[5] De Carolis A, Macchi M, Negri E, Terzi S. Guiding manufacturing companies towards digitalization. 23rd ICE/IEEE Int Technol Manag Conf 2017:503–12. https://doi.org/10.1109/ICE.2017.8279925.
[6] Raj R, Lee I, Sha L, Stankovic J. Cyber-physical systems: the next computing revolution. Des Autom Conf 2010:731–6.
[7] Napoleone A, Macchi M, Pozzetti A. A review on the characteristics of cyber-physical systems for the future smart factories. Int J Ind Manuf Syst Eng 2020;54:305–35. https://doi.org/10.1016/j.jmsy.2020.01.007.
[8] Borangiu T, Trentesaux D, Thomas A, Leitão P, Barata J. Digital transformation of manufacturing through cloud services and resource virtualization. Comput Ind 2019;108:150–62. https://doi.org/10.1016/j.compind.2019.01.006.
[9] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: A categorical literature review and classification. IFAC-PapersOnLine 2018;51:1016–22. https://doi.org/10.1016/j.ifacol.2018.08.474.
[10] Garetti M, Rosa P, Terzi S. Life cycle simulation for the design of product-service systems. Comput Ind 2012;63:361–9. https://doi.org/10.1016/j.compind.2012.02.007.
[11] Yang W, Yoshida K, Takakuwa S. Digital twin-driven simulation for a cyber-physical system in industry 4.0 era. DAAM int. Sci. Book. 2017. p. 227–34. https://doi.org/10.2507/daaam.scibook.2017.18 Chapter 18.
[12] Cattaneo L, Fumagalli L, Macchi M, Negri E. Clarifying data analytics concepts for industrial engineering. IFAC-PapersOnLine 2018;51:820–5. https://doi.org/10.1016/j.ifacol.2018.08.440.
[13] Grieves M, Vickers J. Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. Transdiscipl. Perspect. Complex Syst. New Find. Approaches 2017:85–113. https://doi.org/10.1007/978-3-319-38756-7_4.
[14] Kritzinger W, Karner M, Traar G, Henjes J, Sihn W. Digital Twin in manufacturing: a categorical literature review and classification. IFAC-PapersOnLine 2018;51:1016–22. https://doi.org/10.1016/j.ifacol.2018.08.474.
[15] 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. https://doi.org/10.1016/j.promfg.2017.07.198.
[16] Peres RS, Dionisio Rocha A, Leitao P, Barata J. IDARTS – towards intelligent data analysis and real-time supervision for industry 4.0. Comput Ind 2018;101:138–46. https://doi.org/10.1016/j.compind.2018.07.004.
[17] Qi Q, Tao F, Hu T, Anwer N, Liu A, Wei Y, et al. Enabling technologies and tools for digital twin. Int J Ind Manuf Syst Eng 2019:0–1. https://doi.org/10.1016/j.jmsy.2019.10.001.
[18] You F, Zhao X, You Y. Production management and optimization in injection molding companies based on MES. DEStech Trans Eng Technol Res 2018:253–9. https://doi.org/10.12783/dtetr/mcae2017/15973.
[19] Gyulai D, Pfeiffer A, Kádár B, Monostori L. Simulation-based production planning and execution control for reconfigurable assembly cells. Procedia Cirp 2016;57. https://doi.org/10.1016/j.procir.2016.11.077.
[20] Salleh NAM Kasolang S, Mustakim MA, Kuzaiman NA. The study on optimization of streamlined process flow based on delmia quest simulation in an automotive production system. Procedia Comput Sci 2017. https://doi.org/10.1016/j.procs.2017.01.206.
[21] Kaylani H, Atieh AM. Simulation approach to enhance production scheduling procedures at a pharmaceutical company with large product mix. Procedia Cirp 2016;41:411–6. https://doi.org/10.1016/j.procir.2015.12.072.
[22] Simons S, Abé P, Neser S. Learning in the AutFab – the fully automated industrie 4.0 learning factory of the university of applied sciences darmstadt. Procedia Manuf 2017;9:81–8. https://doi.org/10.1016/j.promfg.2017.04.023.
[23] Vachalek J, Bartalsky L, Rovny O, Sismisova D, Morhac M, Loksik M. The digital twin of an industrial production line within the industry 4.0 concept. Proc. 2017 21st Int. Conf. Process Control. PC 2017 2017. https://doi.org/10.1109/PC.2017.7976223.
[24] Weyer S, Meyer T, Ohmer M, Gorecky D, Zühlke D. Future modeling and simulation of CPS-based factories: an example from the automotive industry. IFAC-PapersOnLine 2016. https://doi.org/10.1016/j.ifacol.2016.12.168.
[25] Wang L, Törngren M, Onori M. Current status and advancement of cyber-physical systems in manufacturing. Int J Ind Manuf Syst Eng 2015;37:517–27. https://doi.org/10.1016/j.jmsy.2015.04.008.
[26] Meyer H, Fuchs F, Thiel K. Manufacturing Execution Systems – Optimal Design, Planning, and Deployment. 2009.
[27] ANSI/ISA. Enterprise control system integration part 1. Models and Terminology 1999:77.
[28] ANSI/ISA. Enterprise control system integration part 3: activity models of manufacturing operations management. 2005.
[29] Monostori L. Cyber-physical production systems: roots from manufacturing science and technology. At-Automatisierungstechnik 2015;63:766–76. https://doi.org/10.1515/auto-2015-0066.
[30] Mohammed WM, Ramis Ferrer B, Iarovyi S, Negri E, Fumagalli L, Lobov A, et al. Generic platform for manufacturing execution system functions in knowledge-driven manufacturing systems. Int J Comput Integr Manuf 2017;32:262–74. https://doi.org/10.1080/0951192X.2017.1407874.
[31] Negri E, Fumagalli L, Macchi M, Garetti M. Ontology for service-based control of production systems. APMS 2015, Part II, IFIP AICT 2015;460. p. 484–92.
[32] Fumagalli L, Pala S, Garetti M, Negri E. Ontology-based modeling of manufacturing and logistics systems for a new MES architecture. APMS 2014 Adv. Prod. Manag. Syst 2014;438:192–200. https://doi.org/10.1007/978-3-662-44739-0_24.
[33] Riedl M, Zipper H, Meier M, Diedrich C. Automation meets CPS. IFAC Proc. Vol. 2013;46:216–21. https://doi.org/10.3182/20130522-3-BR-4036.00071.
[34] Riedl M, Zipper H, Meier M, Diedrich C. Cyber-physical systems alter automation architectures. Annu Rev Control 2014;18:123–33. https://doi.org/10.1016/j.arcontrol.2014.03.012.
[35] Monostori L. Cyber-physical production systems: roots, expectations and R&D challenges. Procedia Cirp 2014;17:9–13. https://doi.org/10.1016/j.procir.2014.03.115.
[36] Monostori L, Kádár B, Bauernhansl T, Kondoh S, Kumara S, Reinhart G, et al. Cyber-physical systems in manufacturing. CIRP Ann Manuf Technol 2016;65:621–41. https://doi.org/10.1016/j.cirp.2016.06.005.
[37] MESA. Mesa International – White Paper Number 6 MES explained : a high level vision. Financ manag. 1997.
[38] Iwatsu S, Watanabe Y, Kodama K. Development of communications protocols between manufacturing execution system and production equipment. IEEE Int Conf Emerg Technol Fact Autom ETFA 2007:280–7. https://doi.org/10.1109/EFTA.2007.4416779.
[39] Urbina Coronado PD, Lynn R, Louhichi W, Parto M, Wescoat E, Kurfess T. Part data integration in the Shop Floor Digital Twin: mobile and cloud technologies to enable a manufacturing execution system. Int J Ind Manuf Syst Eng 2018;48:25–33. https://doi.org/10.1016/j.jmsy.2018.02.002.
[40] Cottyn J, Van Landeghem H, Stockman K, Derammelaere S. A method to align a manufacturing execution system with Lean objectives. Int J Prod Res 2011;49:4397–413. https://doi.org/10.1080/00207543.2010.548409.
[41] D'Antonio G, Bedolla JS, Chiabert P. A novel methodology to integrate manufacturing execution systems with the lean manufacturing approach. Procedia Manuf 2017;11:2243–51. https://doi.org/10.1016/j.promfg.2017.07.372.
[42] Masson C, Jacobson S. Lean planning and execution software: extending lean thinking across enterprise. AMR Res; 2007 (accessed October 30, 2019). http://www.amrresearch.com/Content/View.aspx?CompURI=tcm:7-18992.
[43] Cattaneo L, Rossi M, Negri E, Powell D, Terzi S. Lean thinking in the digital era. Ríos J., Bernard A., Bouras A., foufou S. Prod. Lifecycle manag. Ind. Futur. PLM 2017. IFIP Adv Inf Commun Technol 2017;517:1–11. Springer, Cham.
[44] Lee J, Bagheri B, Kao H. A Cyber-Physical Systems architecture for Industry 4. 0-based manufacturing systems. Manuf Lett 2015;3:18–23. https://doi.org/10.1016/j.mfglet.2014.12.001.
[45] Zhou J, Li P, Zhou Y, Wang B, Zang J. Meng L. Toward new-generation intelligent manufacturing. Engineering 2018;4:11–20. https://doi.org/10.1016/j.eng.2018.01.002.
[46] Zhou J, Zhou Y, Wang B, Zang J. Human–Cyber–Physical systems (HCPSs) in the context of new-generation intelligent manufacturing. Engineering 2019;5:624–36. https://doi.org/10.1016/j.eng.2019.07.015.
[47] Novas JM, Bahtiar R, Belle JV, Valckenaers P. An approach for the integration of a scheduling system and a multi-agent manufacturing execution system towards a collaborative framework. IFAC Proc. Vol. (IFAC-PapersOnline) 2012;45(6):728–33.
[48] Iarovyi BS, Mohammed WM, Lobov A, Ferrer BR, Lastra JLM. Cyber – physical systems for manufacturing execution systems. 2016. 104.
[49] Pfitzer F, Mieth C, Liertz W, Provost J. Event-driven production rescheduling in job shop environments Application to sheet metal production processes. 14th IEEE Int Conf Autom Sci Eng 2018:939–44.
[50] Wang J, Fu P, Gao RX. Machine vision intelligence for product defect inspection based on deep learning and Hough transform. Int J Ind Manuf Syst Eng 2019;51:52–60. https://doi.org/10.1016/j.jmsy.2019.03.002.
[51] Yin X, He Z, Niu Z, (Steven) Li Z. A hybrid intelligent optimization approach to improving quality for serial multistage and multi-response coal preparation production systems. Int J Ind Manuf Syst Eng 2018;47:199–216. https://doi.org/10.1016/j.jmsy.2018.05.006.
[52] Babu M, Franciosa P, Ceglarek D. Spatio-Temporal Adaptive Sampling for effective coverage measurement planning during quality inspection of free form surfaces using robotic 3D optical scanner. Int J Ind Manuf Syst Eng 2019;53:93–108. https://doi.org/10.1016/j.jmsy.2019.08.003.
[53] Chang MML, Ong SK, Nee AYC. Approaches and challenges in product disassembly planning for sustainability. Procedia Cirp 2017;60:506–11. https://doi.org/10.1016/j.procir.2017.01.013.
[54] Zhang H, Liu Q, Chen X, Zhang D, Leng J. A digital twin-based approach for designing and decoupling of hollow glass production line. IEEE Access 2017;3536. https://doi.org/10.1109/ACCESS.2017.2766453. 1–1.
[55] Lee H, Kim T. Smart factory use case model based on digital twin. ICIC Express Lett Part B Appl 2018;9:931–6. https://doi.org/10.24507/icicelb.09.09.931.
[56] Rosen R, Von Wichert G, Lo G, Bettenhausen KD. About the importance of autonomy and digital twins for the future of manufacturing. IFAC-PapersOnLine, vol. 48. Elsevier Ltd.; 2015. p. 567–72. https://doi.org/10.1016/j.ifacol.2015.06.141.
[57] Lenz J, Wuest T, Westkämper E. Holistic approach to machine tool data analytics. Int J Ind Manuf Syst Eng 2018;48:180–91. https://doi.org/10.1016/j.jmsy.2018.03.003.
[58] Rosen R, Boschert S, Sohr A. Next generation digital twin. Atp Mag 2018;60:86. https://doi.org/10.17560/atp.v60i10.2371.
[59] Wang J, Ma Y, Zhang L, Gao RX, Wu D. Deep learning for smart manufacturing: methods and applications. Int J Ind Manuf Syst Eng 2018;48:144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.
[60] Cimino C, Negri E, Fumagalli L. Review of digital twin applications in manufacturing. Comput Ind 2019;113. https://doi.org/10.1016/j.compind.2019.103130.
[61] Cardin O, Castagna P, Couedel D, Plot C, Launay J, Allanic N, et al. . Energy-aware resources in digital twin: the case of injection moulding machines. SOHOMA 2019 2020:183–94. https://doi.org/10.1007/978-3-030-27477-1_14. SCI 853.
[62] Valckenaers P. ARTI reference architecture-PROSA revisited. Int. Work. Serv.

Orientat. Holonic Multi-Agent Manuf. 2018:1–19.

[63] Fumagalli L, Polenghi A, Negri E, Roda I, Fumagalli L, Polenghi A, et al. Framework for simulation software selection Framework for simulation software selection. J Simul 2019;13:1–18. https://doi.org/10.1080/17477778.2019.1598782.

[64] Cavalieri S, Salafia MG, Scroppo MS. Integrating OPC UA with web technologies to enhance interoperability. Comput Stand Interfaces 2019;61:45–64. https://doi.org/10.1016/j.csi.2018.04.004.

[65] Negri E, Fumagalli L, Cimino C, Macchi M. FMU-supported simulation for CPS digital twin. Procedia Manuf 2019;28:201–6. https://doi.org/10.1016/j.promfg.2018.12.033.

[66] Profanter S, Tekat A, Dorofeev K, Rickert M. OPC UA versus ROS, DDS, and MQTT: performance evaluation of industry 4.0 protocols. Proc IEEE Int Conf Ind Technol 2019:0–8.

[67] Negri E, Assiro G, Caioli L, Fumagalli L. A machine state-based digital twin development methodology. XXIV summer sch. “Francesco Turco” – Ind. Syst. Eng. 2019:34–40.

[68] Fumagalli L, Macchi M, Pozzetti A, Taisch M, Tavola G, Terzi S. New methodology for smart manufacturing research and education: the lab approach. Proc Summer Sch Fr Turco 2016;13-15(September):42–7.

[69] Wascher F, Bolla D. Communication between MES 4 and PLC. FESTO Didact SE Mater 2014:1–14.

[70] Taylor N, Human C, Kruger K, Bekker A, Basson A. Comparison of digital twin development in manufacturing and maritime domains. Int Work Serv Orientat Holonic Multi-Agent Manuf. 2020:158–70. https://doi.org/10.1007/978-3-030-27477-1.

[71] Villalonga A, Beruvides G, Castaño F, Haber R. Industrial cyber-physical system for condition-based monitoring in manufacturing processes. 2018 IEEE Ind. Cyber-Physical Syst., IEEE 2018:637–42. https://doi.org/10.1109/ICPHYS.2018.8390780.

[72] Raileanu S, Borangiu T, Morariu O, Anton F. Integrating the digital twin of a shop floor conveyor in the manufacturing control system. Int Work Serv Orientat Holonic Multi-Agent Manuf. 2019;2020:134–45.

[73] Negri E, Davari Ardakani H, Cattaneo L, Singh J, Macchi M, Lee J. A digital twin-based scheduling framework including equipment health index and genetic algorithms. 13th IFAC Work. Intell. Manuf. Syst. 2019:43–8.

Elisa Negri received her PhD in 2016, with a thesis about the role of ontologies for smart manufacturing. She now works at the Department of Management, Economics and Industrial Engineering of the Politecnico di Milano in industrial, national and European projects on ontology-based engineering, industrial systems modeling, shopfloor control, digital twin and Cyber-Physical Systems supporting advanced manufacturing solutions in the frame of Industry 4.0. During her studies and research work, she spent periods at prestigious international institutions, such as EPFL (Lausanne, Switzerland), FIR (Aachen, Germany), Tongji University (Shanghai, China) and Rutgers University (New Brunswick, USA).

Stefano Berardi took his Master Degree in Mechanical Engineering at Politecnico di Milano in April 2019, specialising in Production Systems Design and Management. Since then, he started collaborating with the Manufacturing Group of the School of Management of Politecnico di Milano as research fellow, focusing his research program on the study and development and use of technologies and informative systems in manufacturing environment.

Luca Fumagalli is Assistant Professor in the Department of Management, Eco-nomics and Industrial Engineering of Politecnico di Milano. After his Master of Science in Mechanical Engineering, he completed his PhD studies in 2010 about Innovation in Maintenance Management at the Department of Management, Economics and Industrial Engineering. His research interests are advanced production management, industrial services and maintenance management, with a specific concern on smart technological solutions. His research activity has been related also with European research funded projects. Luca Fumagalli is TeSeM observatory research responsible (www.tesem.net) and MEGMI Master vice-director (Executive Master on Industrial Maintenance Management)(http://www.mip.polimi.it/megmi).

Marco Macchi is currently Full Professor at Department of Management, Economics and Industrial Engineering of Politecnico di Milano. His teaching and research activities regard industrial technologies, asset lifecycle management, operations and maintenance management. Serving the scientific community, he is currently Vice-chair of the technical committee IFAC TC 5.1 Manufacturing Plant Control, and Chair of the IFAC Working Group A-MEST (Advanced Maintenance Engineering, Services and Technology), WG affiliated to the IFAC TC 5.1; he is Member of the IFIP WG 5.7 Advances in Production Management Systems and the IFAC TC 5.3 Enterprise Integration and Networking; besides, he is Book Reviews Editor and Editorial Board Member of the International Journal Production Planning & Control: The Management of Operations, Taylor & Francis.