Design and implementation of a digital twin application for a connected micro smart factory

Article *in* International Journal of Computer Integrated Manufacturing · April 2019

DOI: 10.1080/0951192X.2019.1599439

International Journal of Computer Integrated Manufacturing

ISSN: 0951-192X (Print) 1362-3052 (Online) Journal homepage: https://www.tandfonline.com/loi/tcim20

Design and implementation of a digital twin application for a connected micro smart factory

Kyu Tae Park, Young Wook Nam, Hyeon Seung Lee, Sung Ju Im, Sang Do Noh, Ji Yeon Son & Hyun Kim

INTERNATIONAL JOURNAL OF COMPUTER INTEGRATED MANUFACTURING
2019, VOL. 32, NO. 6, 596–614
https://doi.org/10.1080/0951192X.2019.1599439

ARTICLE

Design and implementation of a digital twin application for a connected micro smart factory

Kyu Tae Park[a], Young Wook Nam[a], Hyeon Seung Lee[a], Sung Ju Im[a], Sang Do Noh[a], Ji Yeon Son[b] and Hyun Kim[b]

[a]Department of Industrial Engineering, SungKyunKwan University, Suwon-si, Republic of Korea; [b]IOT Research Division, Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

ABSTRACT
Recently, manufacturing concepts, such as personalized production and distributed manufacturing, have attracted attention owing to the ongoing revolution in industrial technology. Connected micro smart factories in factory-as-a-service system with these new manufacturing paradigms and Industrial Internet of Things (IIoT) are inefficient in terms of cost and production. To solve these problems, a digital twin, which uses a digital representation of a process, with the same configuration of manufacturing elements, synchronized information, and functional units, was designed and implemented. The digital twin utilizes the latest information from the Internet to gather data from IIoT devices and interoperates in a variety of applications. In addition, it derives the components of a detailed design of the digital twin application, to which it performs procedure definition. This research differs from other digital twin studies that concentrate on the prognostic health management of only a single machine. This study could help managers organize the benefits of utilization through a digital twin based on a hierarchy as they could receive real-time monitoring of the present, tracking information from the past, and operational decision-making support for the future. In addition, the proposed application reduces the cost and production inefficiencies, ultimately resulting in the efficient operation of a manufacturing system.

ARTICLE HISTORY
Received 15 August 2018
Accepted 27 January 2019

KEYWORDS
Connected micro smart factory; digital twin; factory-as-a-service; industrial internet of things; smart manufacturing; personalized production

Introduction

The recent industry-wide revolution has prompted a variety of transformations in the manufacturing paradigm (Yao and Lin 2015). Manufacturing increasingly relies on customer-oriented mass customization and logistics networks, unlike existing mass-production technologies. Thus, the advancement of computational applications and Industrial Internet of Things (IIoT) technology have increasingly attracted interest in distributed manufacturing. However, such distributed manufacturing requires integrated management and the collection of practical manufacturing information from factories (Fan and Huang 2007; Doukas, Psarommatis, and Mourtzis 2014; Yao et al. 2007; Huang et al. 2007).

Mass customization, among the two key manufacturing paradigms, is a manufacturing concept in which the participants in the value chain produce goods with guaranteed low prices while providing customers with highly differentiated value propositions (Huang et al. 2007; Spring and Araujo 2013; Mai et al. 2016). Since its introduction, this concept has evolved into personalized production, which is an advanced concept resulting from continuous development (Yao and Lin 2015). However, personalized production is currently too limited to be considered an effective manufacturing concept in itself (Yao and Lin 2015; Kumar 2007; Silveira, Borenstein, and Fogliatto 2001; Son et al. 2015). Employing this strategy requires considering many factors or responding to various concerns to achieve affordable price levels while providing the highest quality, fastest delivery, and highest level of customization, all of which are the ultimate goals of personalized production (Kumar 2007).

Distributed manufacturing is a manufacturing concept that aims to increase production efficiency on the strength of the supply chain and value-added networks of manufacturing elements (Durão et al. 2017). Adopted mainly on the grounds of logistic, economic, and environmental constraints (Durão et al. 2017), the distributed manufacturing concept supports product customization through a distributed factory network (Fan and Huang 2007). However, managing distributed processes is difficult. Therefore, several studies have aimed to develop promising network architectures and applications that would enable product monitoring and tracking for efficient management (Fan and Huang 2007; Durão et al. 2017).

Factory-as-a-Service (FaaS) is an open manufacturing service that promotes personalized production by addressing the difficulties of initiating production that an individual or company may encounter while commercializing ideas. FaaS also faces access, cost, and performance hurdles. An access hurdle is a difficulty in understanding the detailed product requirements of other manufacturing customization/personalization concepts, a cost hurdle signifies the ever-growing cost issue, and a performance hurdle signifies manufacturing inefficiencies. FaaS addresses these difficulties through the connected micro smart factory (CMSF), which is a distributed manufacturing system, and by using the interconnection with an application utilizing the IIoT. This concept is based on three key

CONTACT Sang Do Noh sdnoh@skku.edu

components: the open manufacturing service platform (i.e. an interaction space for the participants in the value chain), additive manufacturing, and a modular layout design (Silveira, Borenstein, and Fogliatto 2001; Kang et al. 2018). Finally, one more crucial advancement is the digital twin concept, one of the key elements in the research and application of IIoT, is a model that the virtual factory of the information on and functional unit of the physical world. The digital twin also aims to ensure the efficiency of a manufacturing site based on interconnection and interaction via the real-world IIoT network (Silveira, Borenstein, and Fogliatto 2001; Rosen et al. 2015; Kühn 2006; Gabor et al. 2016).

This study considers FaaS, which applies the concepts of personalized production and distributed manufacturing, as the research domain, while the research goals were resolving the performance hurdle and increasing efficiency to mitigate the cost hurdle by applying the integrated automation system. Additionally, the proposed digital twin is a solution for distributed manufacturing systems with heterogeneous manufacturing systems. To simultaneously solve the cost and performance hurdles of personalized production and overcome the difficulties of integrated management, which is a limitation of distributed manufacturing using the factory network, a digital twin based on the IIoT network is applied to FaaS. In addition, to solve the problems of the personalized production and distributed manufacturing system, this study applied the time machine approach, which is among the many approaches of digital twins and can perform monitoring, tracking, and decision-making support roles. Moreover, the design of the digital twin application for performing these roles was studied.

The following tasks will be performed in this study to efficiently design and implement the aforementioned digital twin application. (1) The paper discusses the scenario of FaaS, the production phase of the scenario, and problems associated with the manufacturing elements at this stage. (2) Next, the possibility of solving the problems derived from a digital twin and its time machine approach is examined. (3) A system scenario was constructed to be executed by the digital twin application to solve the defined problems. (4) Furthermore, an interoperability-context system architecture was designed to execute this scenario. (5) Moreover, this paper details the designs of the application configuration and system architecture and (6) defines the procedures to efficiently operate and manage the system.

Background

Factory-as-a-service (FaaS)

This section defines personalized production and then discusses previous studies on FaaS, to which this manufacturing concept was applied. In this section, access, cost, and performance hurdles are defined, which are three key issues within the scope of the FaaS: (Fan and Huang 2007; Mai et al. 2016; Silveira, Borenstein, and Fogliatto 2001; Son et al. 2015; Du, Jiao, and Tseng 2006).

To realize personalized production by resolving the access hurdle, which is the first constraint, detailed, accurate information on the product that the customer wishes to produce should be shared among the participants in the value chain (Silveira, Borenstein, and Fogliatto 2001; Du, Jiao, and Tseng 2006). In addition, a considerable amount of feedback and engineering knowledge is required at the design stage (Tseng, Jiao, and Wang 2010) to satisfy customers across various market segments, and a space or platform for sharing such information should be provided (Mourtzis, Doukas, and Vandera 2017). The information-sharing medium, which is essential for enabling personalized production, is constructed to function across various platforms, such as in smart mobile apps (Mourtzis, Doukas, and Vandera 2017), product data management systems, computational systems (Silveira, Borenstein, and Fogliatto 2001), and Web-based platforms (Kumar 2007).

Similarly, the FaaS system also incorporates a FaaS platform for sharing and utilizing information among the participants in the value chain (Son et al. 2015; Han et al. 2016). Figure 1 shows a conceptual diagram of the service scenario of the FaaS platform. The service platform receives an order from the

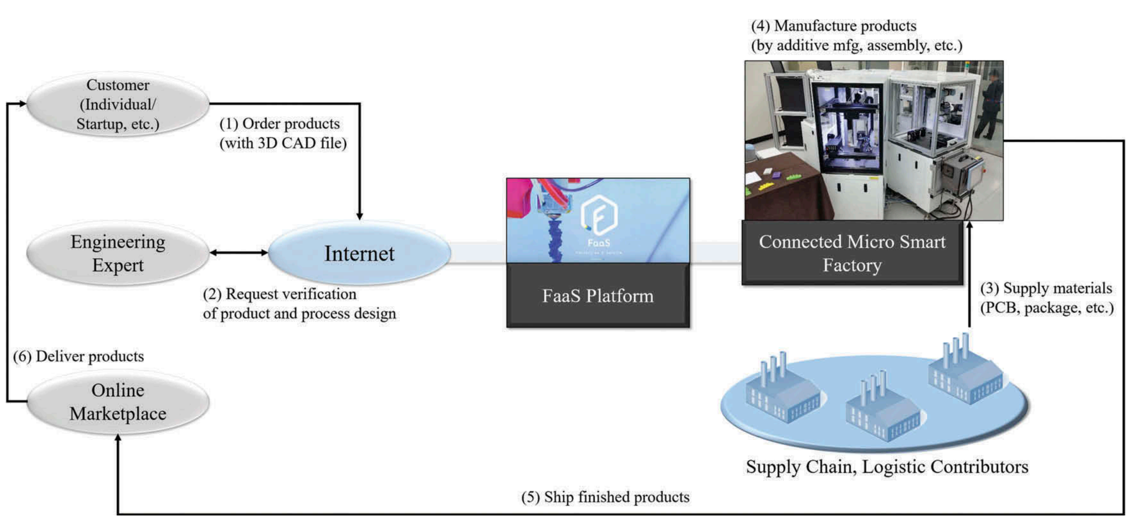

Figure 1. Conceptual diagram of the FaaS platform for personalized production (Son et al. 2015).

customer via the Internet, and the product design is finalized through interactions with engineering experts based on the customer order. The entire service scenario is completed when the product is manufactured by operating the manufacturing elements connected with the customer via IIoT. The product is then delivered to the customer (Son et al. 2015).

In the fourth stage of the above-described FaaS scenario, i.e. the manufacturing stage, products are produced in the micro smart factory (MSF), where personalized production occurs. The cost and performance hurdles are the constraints shadowing the entire scenario at this stage, particularly with regard to efficiency. These issues can be partially resolved by configuring the limited and modular manufacturing elements to increase efficiency; however, additional efficiency is still required owing to the insufficient integration and utilization of information (Son et al. 2015; Kang et al. 2018).

The MSF is a manufacturing system designed to create optimal production efficiency in an effort to overcome the cost hurdle, as shown in Figure 2. The cost hurdle signifies a problem in which the production cost continues to increase structurally when producing a variety of personalized products in small amounts. In this system, both the production cost and period are lower than those of the conventional mold making method owing to the adoption of the laminated production method. In addition, the optimal production efficiency is guaranteed by components that help managers deal with complex processes (Son et al. 2015; Kang et al. 2018). To achieve logistical efficiency, a distributed manufacturing system is also introduced, while a logistic hub is selected as the factory site, thus connecting the MSF to complete a CMSF (Son et al. 2015).

The manufacturing elements of an MSF can be categorized broadly into production, post-processing, assembly, and packaging. These processes consist of automated facilities and are designed to be easily removed or added as modules (Kang et al. 2018). In addition, information on the factories located in the MSF is gathered, shared, and input via the FaaS platform, thereby establishing an infrastructure for designing and applying various applications (Han et al. 2016).

Although MSF-based methods and technologies have been applied to overcome the cost and performance hurdles by capitalizing on the logistical advantage according to the location as well as by promoting the base environment for sharing and utilizing data, few studies have focused on efficiently managing and utilizing the data. In particular, most of the data are not used after their vaporization or storage, thereby preventing the IIoT network from creating added value.

Industrial Internet of Things (IIoT) and digital twin

This section describes IIoT-based manufacturing applications and examples. It also defines the digital twin, a technology utilized by advanced IIoT-based manufacturing applications, and it describes some applications. MSF was selected as the physical domain to be replicated in this study, and the digital twin was generated by virtualizing its components and reference data. As such, this study aimed to promote monitoring, tracking, and decision-making to overcome the performance hurdle of FaaS that incorporated the personalized production concept via the time-machine approach; a digital twin application approach.

The IIoT discussed here is based on the Internet of Things (IoT), which is defined as 'a global infrastructure for the information society, enabling advanced services by interconnecting physical and virtual things based on the existing and evolving interoperable information and communication technologies' (Mulani and Pingle 2016). This innovative technology enables

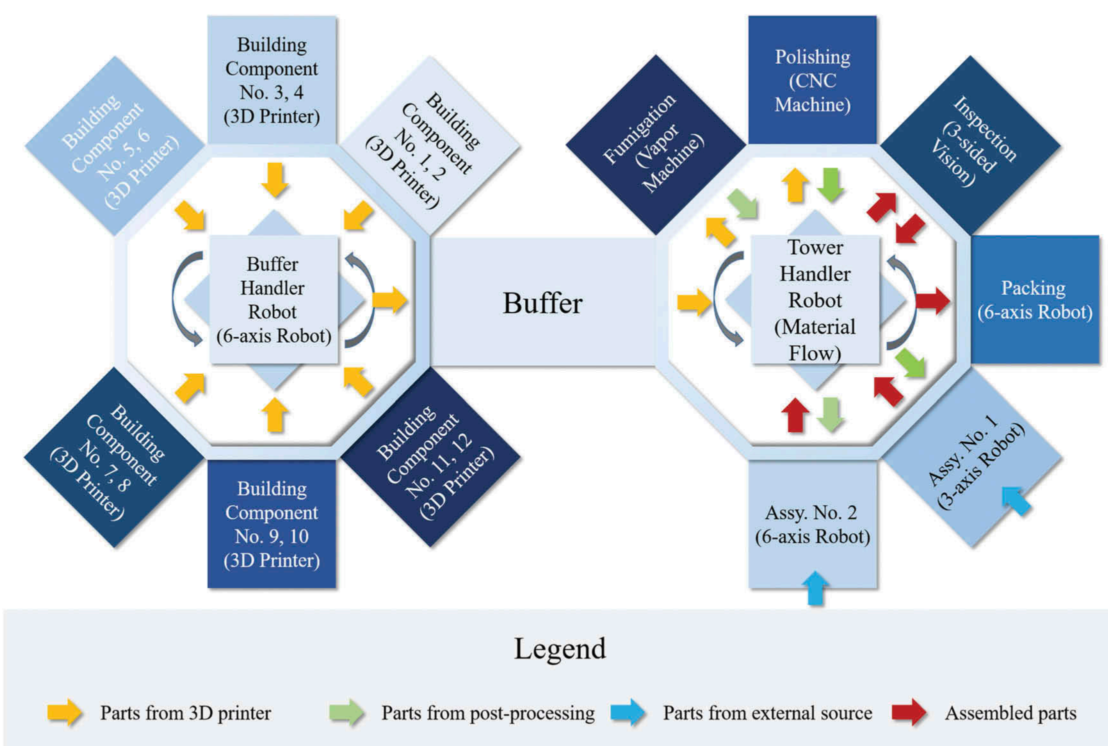

Figure 2. Schematic diagram of a micro smart factory (MSF) layout (Kang et al.).

interconnectivity through the integration of sensors and devices to record, transmit, and share collected data (Ngai et al. 2008). In other words, IIoT is defined as IoT utilized in industry and applied to the entire product lifecycle (Liao, Loures, and Deschamps 2018).

IIoT-based manufacturing applications enable adding value and creating new services through the exchange of data among the manufacturers, operators, and manufacturing devices connected to IIoT by utilizing electronic equipment, software, sensors, and connectivity (Lu and Cecil 2015). IIoT-based manufacturing applications also utilize connectivity, data exchange, and interaction among the aforementioned manufacturing elements, is being studied and applied to a variety of manufacturing issues, including additive manufacturing (Mai et al. 2016; Durão et al. 2017), energy efficiency (Lu and Cecil 2015), manufacturing service (Spring and Araujo 2013), human resources, robots (Schirner et al. 2013), and industrial automation (Morgan and O'Donnell 2017). The target process to with IIoT-based manufacturing is applied varies as well, such as monitoring (Durão et al. 2017; Morgan and O'Donnell 2017), optimization (Fang et al. 2016), control (Morgan and O'Donnell 2017), support for decision-making (Lu and Cecil 2015; Schirner et al. 2013), and management (Lu and Cecil 2015).

Due to the advancement of the IIoT, the concept of a digital twin has continued to change and evolve over time since its introduction by Grieves (Grieves 2014),. A digital twin can be defined as an integrated virtual model of a real-world system containing all of its physical information and functional units. In the manufacturing field, it can be used as a virtual factory to synchronize information and functional units related to the design, operation, and production of actual factories (Tao et al. 2017; Alam and Saddik 2017). The digital twin is model the components of the physical world before synchronizing the information and functional units with the created components via an IIoT network (Alam and Saddik 2017).

The digital twin has the following features. First, the physical world is reflected in the cyber world in real-time, suggesting that digital twin applications in the cyber world can perform real-time monitoring and tracking of various situations in the physical world. Second, it fosters development through repeated interactions and convergence at high levels. It can interact between various informational and functional elements in both the physical world and network, as well as various integrations and opportunities to add value based on these elements. In addition, it can derive new analysis results and accumulate knowledge based on past and real-time data (Gabor et al. 2016). Third, although the digital twin plays a role similar to that of the existing simulations, as it adopts simulation as a core functionality, it can also play relatively diverse roles through integrated analysis by supporting quick decision-making via the IIoT network and by securing interoperability with other manufacturing applications (Boschert and Rosen 2016).

Owing to the growing interest in digital twin, it has been developed, and applied to various domains. Tammaro et al. (2017) succeeded in fabricating a handler robot, equipped with image-based sensors to track objects in a virtual environment by using a digital twin, thereby allowing the end effector to perform tracking. Bielefeldt, Hochhalter, and Hartl (2015) employed a digital twin to monitor and detect damages to an aircraft structure. Wang et al. (2017) attempted to teach robots by capturing the gesture of the operator by using a digital twin. Studies on the digital twin have also been conducted to achieve greater efficiency for bigger-unit physical elements by extending the scope of the physical world and integrating data. Some other studies have attempted to apply the digital twin to a single physical element. Alam and Saddik (2017) defined the number of complex interactions between the physical world and the cyber world to create a reference model for the application of the digital twin.

Lee, Beghari, and Kao (2015) proposed a five-level cyber-physical system structure for using a digital twin, which they called the 5C architecture, for the five keywords of the 5C architecture (connection, conversion, cyber, cognition, and configure), thereby showing various directions in which the digital twin can be developed, As an example of the conversion/cyber approach, they proposed a time-machine approach that utilizes a digital twin to support monitoring and decision-making. This approach synchronizes the past history, current operation, and future plan, like a time machine for the digital twin. Furthermore, peer-to-peer comparison monitoring compares snapshots, which provides point-of-time information from the past to future, to support advanced engineering services.

In summary, many studies related to digital twin have been conceptually proposed, but research with practical results remains insufficient thus far. From the practical application point of view, some highly sophisticated research has concentrated on a single machine and its monitoring, prognosis, or simulation. Studies on increasing the efficiency of a single machine are also valuable, but using a digital twin for production requires the integrated construction of an entire manufacturing system. In particular, in order to improve the productivity of the distributed manufacturing system, a digital twin that can support the decision-making of the entire manufacturing system must be developed.

Design of a digital twin application for connected micro smart factory

This study was conducted within the FaaS service scenario introduced in the background section, and the research scope was set ensuring the operational efficiency of the 'manufacture products' stage, which is the fourth stage of the scenario. The system adopted in this study utilizes the time-machine approach of the digital twin application method to achieve operational efficiency at the 'manufacture products' stage. Toward this end, the adopted system synchronizes various information on the past, present, and future states of MSF to provide the monitoring and tracking as well as decision-making support functions to the users.

This section introduces the design of the digital twin application. First, the operational scenario executed by the present system is described. This scenario defines the two worlds contained in the system as well as the layers within each world, and describes the activity performed by each layer. Second, the interoperability-context system architecture for utilization of digital twin is described. This system architecture consists of several elements according to the scenarios, which interoperate with one another to exchange information. Third, the application of the digital twin is explained: this is an element that directly performs synchronization in the entire system, interacts with the other applications in the manufacturing application layer as well as with the databases in the IIoT network layer. The configuration for this interaction is then described along with the configuration of the contents of the operation module within the configuration. Finally, the actual operation process of the digital twin application, including an operation module, is described.

Scenario of the digital twin application

This section discusses the scenario that the adopted system should provide for the system design. This system provides the tracking, monitoring, and decision-making support functions through the utilization of the digital twin at the 'manufacture products' stage. In addition, as a subsystem of the FaaS system, the adopted system utilizes the time-machine approach based on creation and synchronization of information to the digital twin. To do this, the system needs information on the past production history, current operation status, and future production plan in MSF. The proposed system consists of four layers: two layers each in the physical and cyber worlds.

Figure 3 shows a conceptual map of the digital twin application proposed in this study, which aims to resolve issues with personalized production, which produces various products and creates many dynamic situations. The digital twin application also performs integrated management roles for multiple distributed manufacturing systems within FaaS. To accomplish these roles, it generates and synchronizes an IIoT-based digital twin. According to the synchronized information, the digital twin monitors the present in real-time, tracks the past, and makes predictions to support decision-making for the future.

Figure 4 shows the scenario of the proposed concept and digital twin through an activity diagram. The physical and cyber worlds are each divided into two layers, as labeled on the diagram. Namely, the physical world involves the CMSF and the IIoT network, and the cyber world involves manufacturing application and a cloud application. The exchange of data or information with other cells is indicated by a solid line, while dotted arrows indicate that the first activity requires other activities before proceeding to the next step. The scenario starts from the physical world's MSF and concludes at the manufacturing application layer of the cyber world.

The first layer of the physical world is the CMSF. The manufacturing elements of the CMSF are connected to the IIoT network via the IIoT sensors or middleware. When the CMSF manufactures products according to the production schedule, the manufacturing data are collected per unit time, such as information on the product currently under production, the process status, plant information from the factory cell, and operation information in actual plant. This information flows to the IIoT network, which is the second layer of the physical world. The manufacturing data that are sent via the

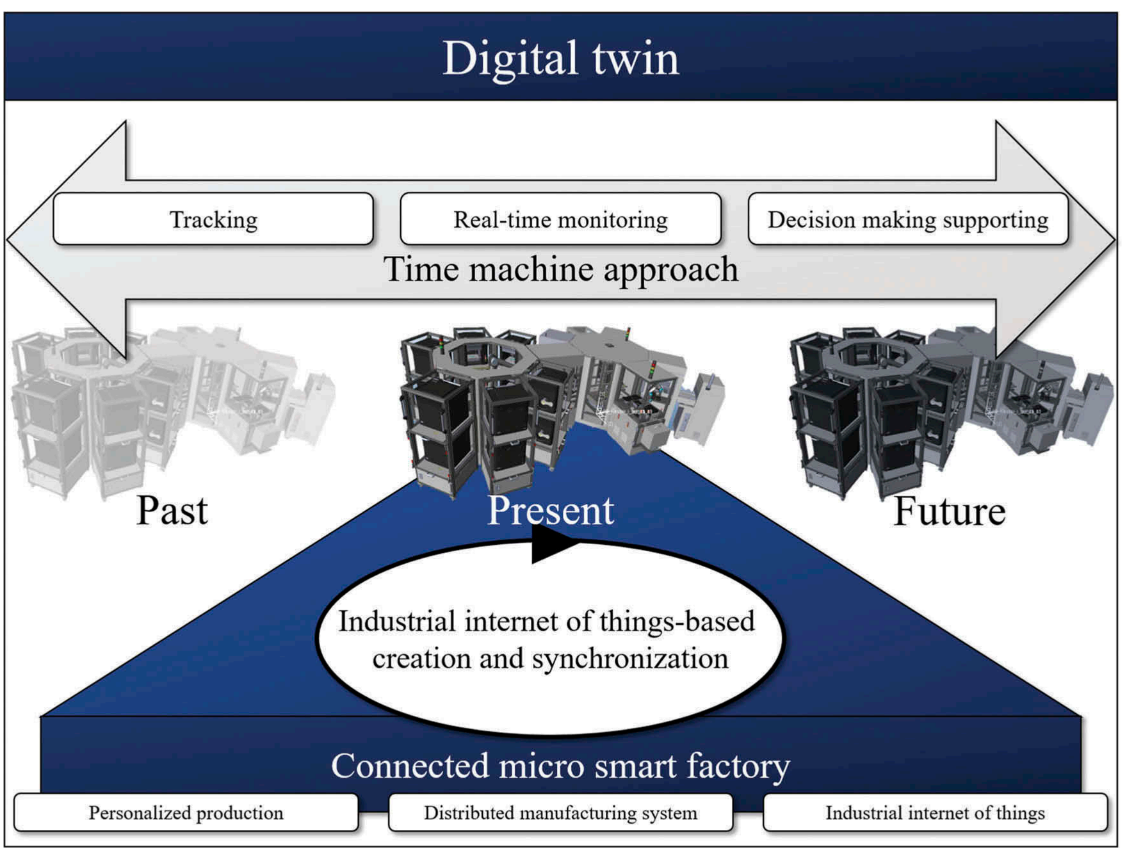

Figure 3. Conceptual map of the proposed digital twin application.

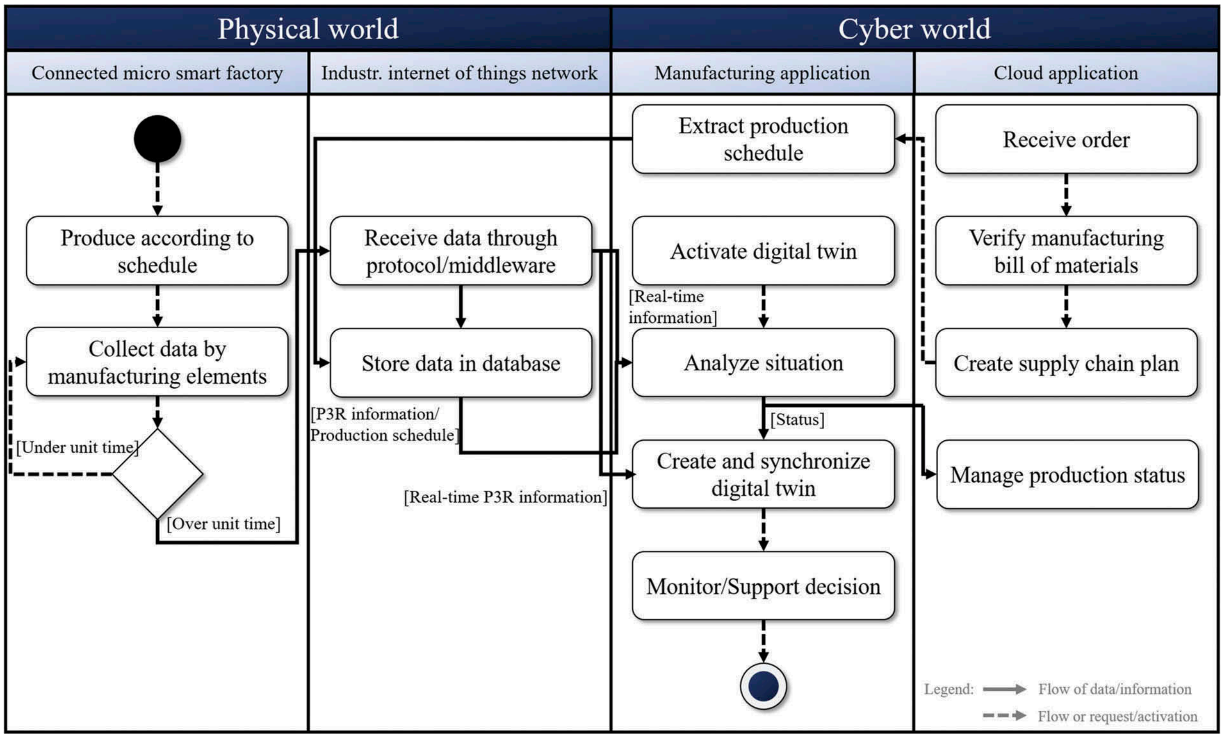

Figure 4. Interoperability-context scenario of the digital twin application.

connection with the IIoT protocol or the middleware itself are then classified and stored.

The cyber world consists of a manufacturing application layer designed to overcome the performance hurdle of personalized production, and a cloud application layer, which works as a contact point between the customer and the system and performs management roles. The cloud application layer also helps overcome the access hurdle by receiving customer orders along with the FaaS scenario. Moreover, it confirms the manufacturing bill of materials (MBOM) to accurately grasp the customer needs. After the MBOM is confirmed, a supply-chain plan is prepared for the CMSF, which is a distributed manufacturing system, and a production schedule is requested from the manufacturing application layer. The extracted production schedule is then moved to the IIoT network layer and stored in the database.

The manufacturing application layer synchronizes the past, present, and future operations based on the manufacturing information received from the databases or protocols of the IIoT network, depending on the time that the user wants to monitor. If users want to track the past, they can ask for past manufacturing information from the database, identify the status based on the obtained information, and utilize integrated monitoring, tracking, and decision-making support functions through synchronization based on the identified status. If users want to know about the current operation, they may receive data from the direct protocol, and if they are interested in the future production plan, they need to obtain a schedule plan from the database before synchronizing it and performing the system role in the digital twin.

Design of the digital twin application

This section discusses the design for implementing the above scenario of the system. Similar to the scenario, this digital twin architecture has four layers (i.e. CMSF, IIoT network, manufacturing application, and cloud application), and the elements within each layer are explained later in the paper. The information flow between each layer or between applications in the layer will also be explained. This information flow varies depending on the three perspectives provided by the time-machine approach.

Next, the composition of and interoperability between the elements in the four layers to implement the system scenario are discussed. These elements include interface, model, module, engine, rule, and algorithm components according to their respective roles, while each layer is classified by its main role. To create and synchronize the digital twin, the MSF elements provide manufacturing data from the sensors, and the elements of the IIoT network layer transfer and store data. The cloud application plans production based on the customer order, and the creation and synchronization are finalized by the manufacturing application by using the digital twin in operation.

Figure 5 schematically shows the interoperability-context architecture, complete with each layer's elements and given role. As in the diagram of the scenario above in the previous section, the solid arrows indicate the flow of information or data, and the dotted arrows indicate requests or calls between the elements. Data are collected on the products assembled in the MSF of the physical world and the manufacturing processes, such as the part production, post-processing, assembly,

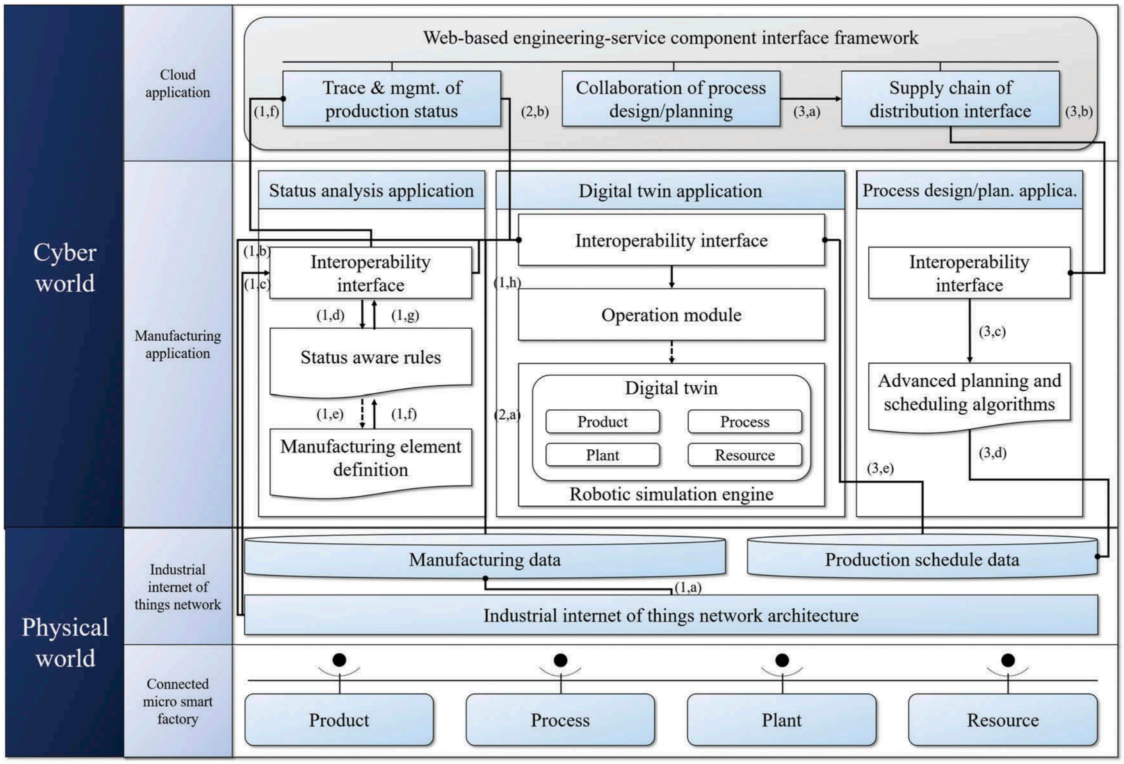

Figure 5. Interoperability-context architecture of the digital twin application.

inspection, and packaging processes, through which these products are manufactured, and these data are moved via the IIoT protocol or middleware in network architecture. In addition, the cell performing such production processes sends out plant data, while the robotic devices transferring the assembled products transmit real-time data via the communication medium. The IIoT network layer thus consists of the communication interfaces that transmit the manufacturing data and the databases that store the data for synchronization. There are two databases: one stores the manufacturing data, and the other contains the production plans and schedules.

The manufacturing application layer consists of a status analysis application, digital twin application, and process-design/planning application. Each application has an interface for sending and receiving data. The status analysis application contains the definitions of the characteristics of each manufacturing element within the defined rules, and the MSF to identify the status from the data. The digital twin application has a robotics simulation engine capable of performing three-dimensional (3D) visualization functions and a digital twin containing the elements of the MSF and definitions of the functional units. This digital twin is defined to contain the manufacturing information, while the operation module creates and synchronizes this digital twin based on the information received from the interoperability interface. The process-design/planning application extracts the schedule according to the production plan based on the advanced planning and scheduling algorithm. The cloud application consists of an integrated management application for the Web-based status, an application for confirming the MBOM after receiving an order, and an application for supply-chain planning based on the confirmed MBOM.

Table 1 illustrates the information flow during the synchronization of the MSF. In the case of synchronization for the current operation, the manufacturing data collected in the MSF is not stored in the database as (1, b) and (1, c) but flows directly from the IIoT protocol/middleware to the digital twin application and the status-aware application. This is to increase the efficiency of the processing time. The manufacturing data are instead stored separately, as in (1, a), and are used when performing the tracking role. (1, d) indicates the process of sending the current manufacturing data to a status-aware rule defined to identify the actual status in the interoperability interface. On this basis, the status information is identified through mutual exchange of information between the status-aware rule and manufacturing-element definition. The information identified as such is then transmitted to the

Table 1. Description of the information flow of real-time data synchronization.

Notation	Contents
(1, a)	Manufacturing data collected from the MSF for each unit time stored in the manufacturing database
(1, b)	Manufacturing data collected from the MSF for each unit time utilized in real-time by the digital twin application
(1, c)	Manufacturing data collected from the MSF for each unit time utilized in real-time by the status analysis application
(1, d)	Manufacturing data parsed to enable handling by the interoperability interface in the status analysis application
(1, e)	Request for function information by the manufacturing element
(1, f)	Functional information of the manufacturing element
(1, g)	Status information per manufacturing element
(1, h)	Status information per manufacturing element

other applications through the interoperability interface, as shown in (1, g) and (1, h).

Table 2 illustrates the information flow during the synchronization of the factory operation previously performed by the MSF. Unlike using real-time data, synchronization is performed by receiving the information stored and managed in the database according to the time of need. All the data for a specific time are provided at once instead of being provided at a short time allowance and with a limited amount per unit time when performing tracking, unlike when utilizing real-time data, which provides advantages in terms of the allowed time. Therefore, the manufacturing data flows from the manufacturing database according to the desired event period, as in (2, a), and the status information flows as in (2, b), thereby moving the information, similar to that in Table 1.

Table 3 illustrates the information flow when the MSF synchronizes the production schedule to be performed later. MBOM reflecting the customer value through interaction between the customer and engineers is transferred to extract the supply-chain plan information, as shown in (3, a), while the supply chain information derived from this MBOM is provided to the process-design/planning application, as shown in (3, b). Further, the advanced planning and scheduling algorithm in the application was used to save the schedule in the process-plan/schedule database. This stored schedule has a Gantt-chart-type information structure and schema. Based on this information, synchronization was performed for the decision-making support of the production plan for a specific event in the future among the three kinds of events of the time-machine approach.

Configuration of the digital twin application

The digital twin application implements the final synchronization in the system adopted in this study, with the interoperability-context system architecture and information flow described in the previous section. This section illustrates the detailed configuration of each of the elements used in this application. In addition, the function of each element in the composition is explained, along with its function configuration based on the class diagram of the operation module.

The digital twin application in the manufacturing application layer consists of the interoperability interface, robotics simulation engine, background data, and operation module, as shown in Figure 6. Table 4 shows the flow of the information from Figure 6 and describes the calls and requests between the different components. Although the notation numbers in Figure 6 and Table 4 do not directly contain the sequence of procedures, the corresponding procedure elements work in combination according to the creation target and synchronization event.

The interoperability interface consists of a streamer, which is activated when real-time data are input, and a parser, which imports and processes all types of received information. This interface serves as a channel for transferring information from the manufacturing elements, applications, and databases to the digital twin application.

The robotics simulation engine is an engine of the digital twin application; it utilizes the robot library, an entity provided by the engine itself, and other functions, such as 3D visualization and simulation execution/result analysis. The robot library contains the model of the robot used in the MSF. The MSF uses multiple robots per plant for transportation and assembly while their kinematics definitions have their own motion information. The digital twin importing function implements the digital twin environment in the robotics simulation engine by importing the background data, component information, and configuration unit information in the robotics simulation engine. The result is shown to the user through a 3D visualization function. Virtual object mapping directly synchronizes the manufacturing and status information with the digital twin components according to the synchronization request. For future cases, synchronization is performed by conducting a simulation based on the simulation sequence, and the results are obtained after the simulation is completed.

Background data contain predefined information and elements for utilization of the digital twin by using the aforementioned engines. The base model has a 3D CAD model containing digital twin components, and each model has one sub-model. The plant-setting library needs to set information, such as the operating range and speed, to operate the device and robot in the plant. The set values of each MSF are presented in the form of a library. The Device/Path/Logic library has a CAD model and the basic information of devices, such as of the machine that performs the actual work (excluding the robot in the plant), working path, and logic. The background data contain definitions of the kinematics information of the robot for the operation of devices.

The robot kinematics information and device definition are separated in space because commercial robots can be manipulated by utilizing the robot kinematics definition in the simulation engine, for example, the universal robot 10 (UR10) shown in the left image in Figure 7. Therefore, a separate device definition, such as that of the tower handler shown in the right image in Figure 7, must be configured and stored in library form to operate devices, such as a grinder, fumigator, transfer robot, and granulator, which are directly used in the MSF. In fact, users can control, import, and use the configuration values of the devices and robot operation in an actual factory by storing them in the plant/robot-setting library. The numbers in the two images in Figure 7 show the concept of the kinematics of the two handlers. Such robots

Table 2. Description of the information flow of tracking information synchronization.

Notation	Contents
(2, a)	Manufacturing data stored in the manufacturing database corresponding to the target tracking time
(2, b)	Status information per manufacturing element corresponding to the target tracking time

Table 3. Description of the information flow of the production schedule plan information synchronization.

Notation	Contents
(3, a)	MBOM deduced from the interactions with customers
(3, b)	MBOM-based supply-chain plan
(3, c)	Parsed supply-chain plan
(3, d)	Schedule information of the target planning time
(3, e)	Status information per manufacturing element

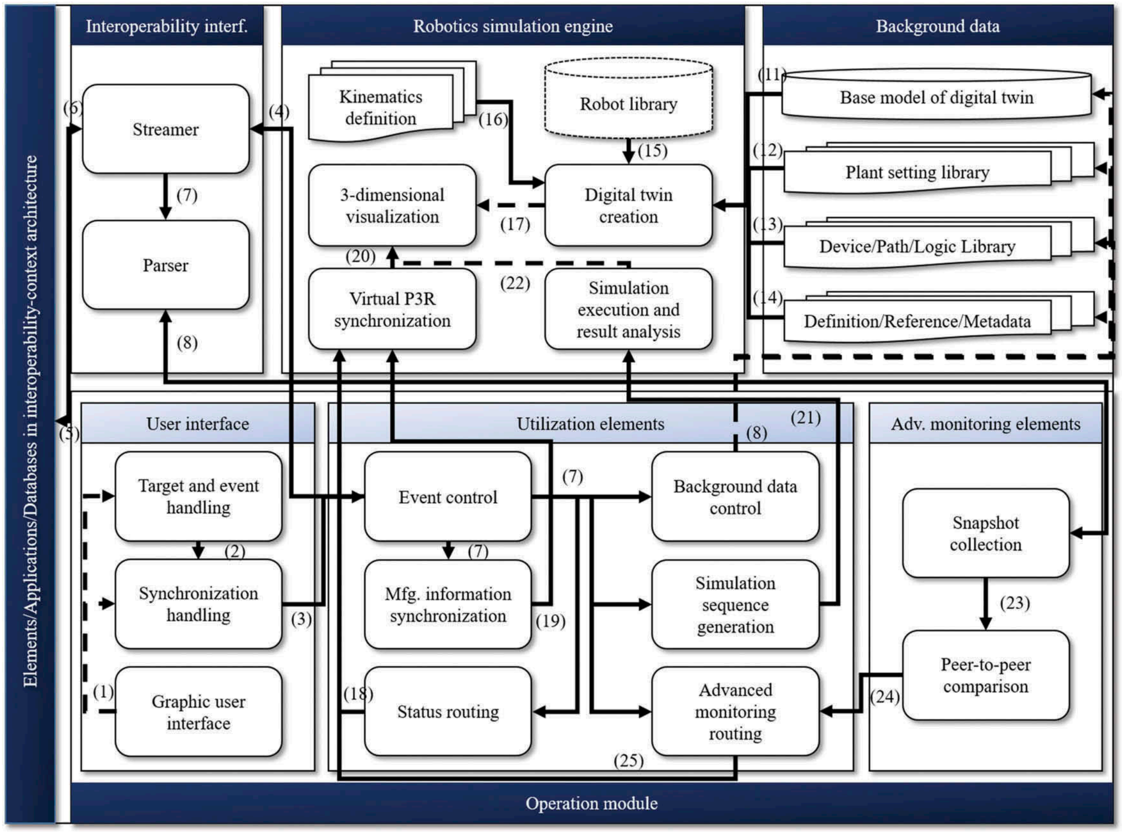

Figure 6. Configuration diagram of the digital twin application.

and devices implement operations with the above-mentioned kinematics combinations.

The user interface of the operation module sets up the event that the user wants to monitor/track through synchronization and handles the synchronization accordingly. The *Manufacturing Information Synchronization* class set (Figure 6) synchronizes the information provided to the other applications or elements in the operation module. It can also interact with the background data based on different information types, such as the status information, manufacturing information, and analysis result, according to the set event and perform synchronization to enable monitoring/tracking. In addition, it also generates the sequences required to facilitate the actual control of the information necessary for the event, the activation of the digital twin in the engine, and the simulation of the production schedule plan. The snapshot collection and peer-to-peer comparison, which are the two analyses performed in the operation module, provide information to enable advanced monitoring routing by analyzing the functions to be provided by the time-machine approach that relies on the digital twin.

Figure 8 shows a class diagram with only the main contents of the operation module. *Event control* is an abstract class that utilizes only the data-request and calling functions according to the event called in the user interface. Other classes contain functions, which are defined for interacting with the interoperability interface, robotics simulation engine, and background data.

The *background data control* class imports a digital twin by importing the base model, plant, device, path, logic, definition, and robotics simulation engine in the background data. The *Manufacturing Information Synchronization* class includes a function that can visualize what process a particular product or part is performing in a particular plant, and it can display the device/robot behavior. The *status* and *advanced monitoring routing* classes serve to project the information from each application or module to the operation module through a digital twin, while the *simulation sequence creating* class serves to create a sequence chart to synchronize the production schedule and plan and promote an atmosphere conducive to simulation.

Procedures of the digital twin application

This section explains how the function sets and libraries are actually called and operated according to the event and creation object. This procedure works by organically linking the function and library in the digital twin application, and it comprises different procedures according to the set event and creation object.

The digital twin should be created first before synchronization according to the point-of-event. For this, a 14-step procedure should be performed (Figure 9). First, run the digital twin application to complete the initial settings, and input the MSF that is the target of creation and the event, which is the desired timing for synchronization, through the user interface

Table 4. Description of the configuration diagram of the digital twin application.

Notation	Contents
(1)	Determination of and request for a user event and the creation and synchronization of target via the graphic user interface
(2)	Request for creation and synchronization of the confirmed event
(3)	Request for creation for the confirmed target
(4)	Request for data for the confirmed event and synchronization
(5)	Request for data for the confirmed event and synchronization
(6)	Provision of data by other elements in interoperability-context architecture
(7)	Move the real-time information received via the streamer
(8)	Move the information preprocessed for use in the operation module
(9)	Call function tailored for the confirmed event and synchronization
(10)	Call background data for extracting a digital twin
(11)	Move the information on the base model of the digital twin components into the robotics simulation engine
(12)	Move the configuration value information of the plant/resource
(13)	Move the device model and its path/logic information among the actual task elements excluding the robot in the plant of the digital twin
(14)	Move the information on the behavior definition/constraints/criteria information for actual operation in the digital twin
(15)	Move the robot model information, among the actual task elements, in the plant of the digital twin
(16)	Move the kinematics information, which is a definition of the behavior of the robot model of the digital twin
(17)	Request for 3D visualization of the imported digital twin components and functional unit
(18)	Move the past and present status information
(19)	Move the past/present/future manufacturing information
(20)	Request for 3D visualization of the status and manufacturing information
(21)	Move the simulation sequence information in the future
(22)	Request for 3D visualization of the simulation
(23)	Move the snapshot for peer-to-peer comparison monitoring
(24)	Move the result of the peer-to-peer comparison monitoring
(25)	Request for 3D visualization of peer-to-peer comparison monitoring

to check if the event is feasible. If it is not feasible, reset the creation target and event to check if it is feasible, and then prepare a digital twin to be loaded onto the robotics simulation engine through the background control function. The main role of this input and preparation process is to confirm if the target, which is selected to be created, and the event, which is requested to be synchronized, are appropriate and to check if the background data for this procedure is fully ready. The base model, robot, and device are then imported from the background data and built-in libraries of the robotics simulation engine. As these base model, robot, and devices are CAD models and do not include functional units, they should import functional units, such as the kinematics, definitions, path, logic, and configuration values, for digital twin configuration. Finally, the importing of the digital twin is completed by updating the view through 3D visualization.

After the digital twin is imported, the digital twin application can synchronize the information according to the creation target MSF and the selected point in time, as shown in Figure 10. The information synchronization procedure is divided into the past tracking of the factory operations, real-time monitoring of the present status, and verification of the future plan. In cases of the real-time monitoring of the current point in time and the tracking of the past plant operation, the procedure is performed by utilizing a similar function as a different event flow.

In the case of the real-time monitoring of the factory-operation information, information is fetched via the status analysis application and streamer before being parsed for handling purposes. In addition, if users want to track the past plant operation, they can retrieve the information from the status management application and manufacturing database to parse so that the operation module can use this parsed information. Information synchronization of the past and present viewpoints works similarly afterward. The procedures are first performed to synchronize the status information and manufacturing data. During synchronization for the real-time monitoring of manufacturing data, the loop is turned on during the unit time to collect information, before retrieving the information again by inputting the information to the digital twin and updating the digital twin view. However, when tracking the past factory operation, all the manufacturing data of the input event is retrieved at once, thereby ensuring that the manufacturing data are projected without receiving additional

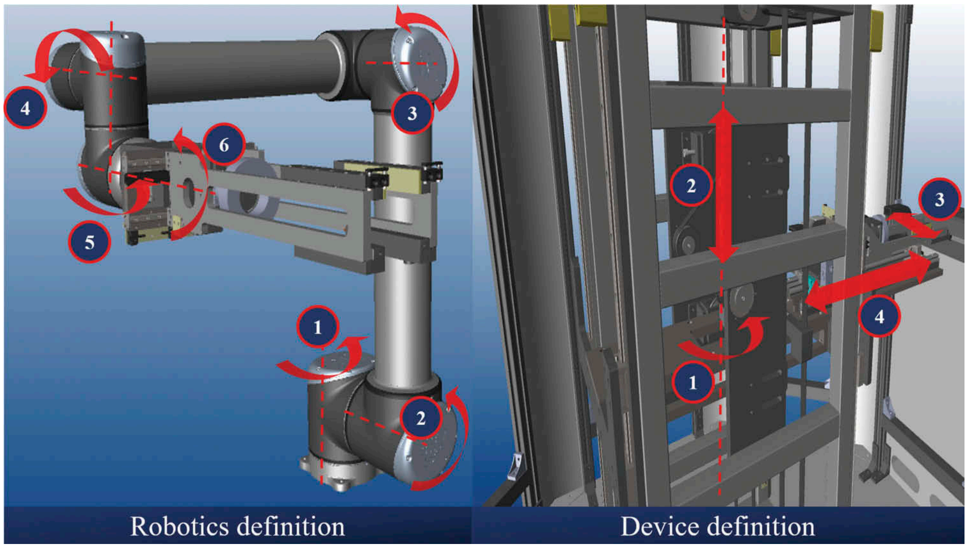

Figure 7. Snapshots of examples: kinematics definition of universal robot 10 (left) and device definition of the tower handler (right).

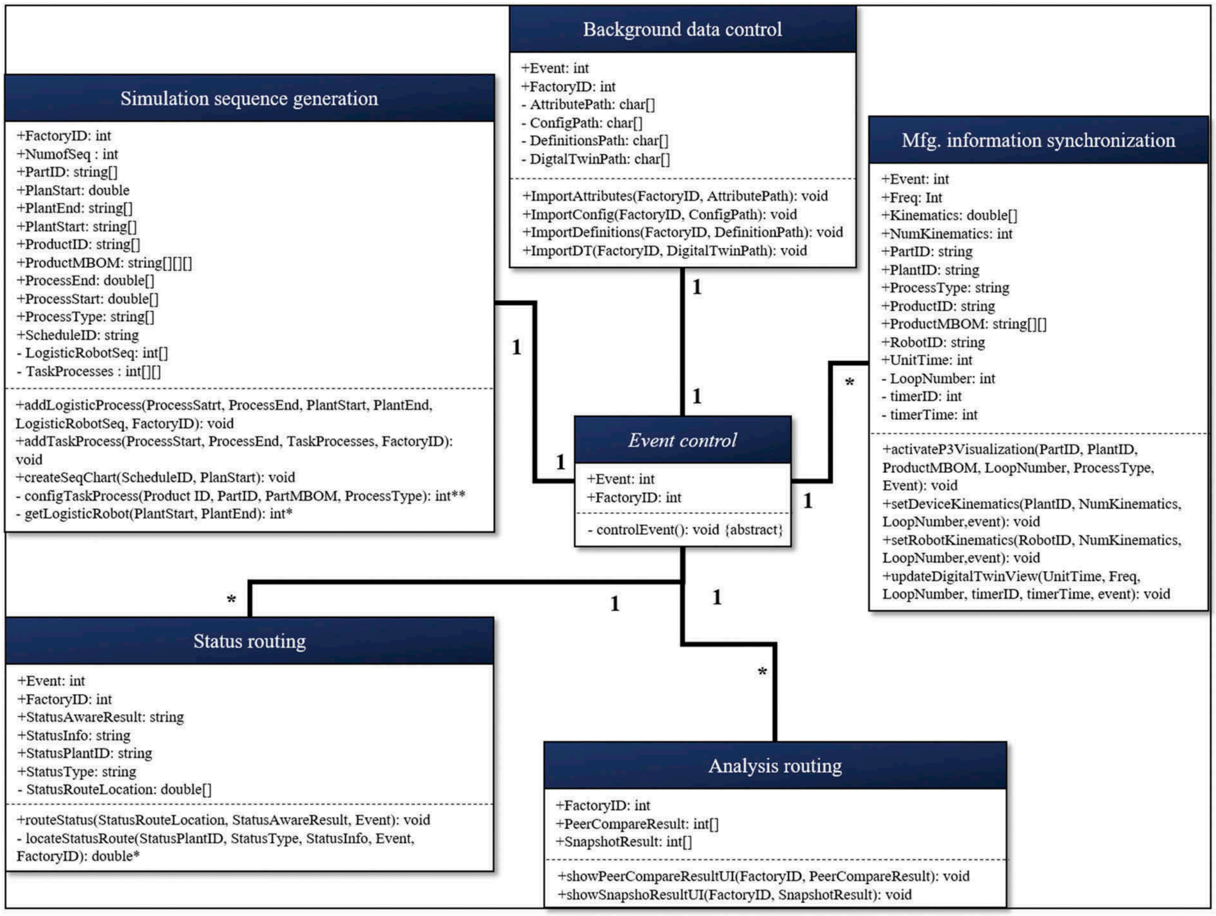

Figure 8. Class diagram of utilization of elements in the operation module.

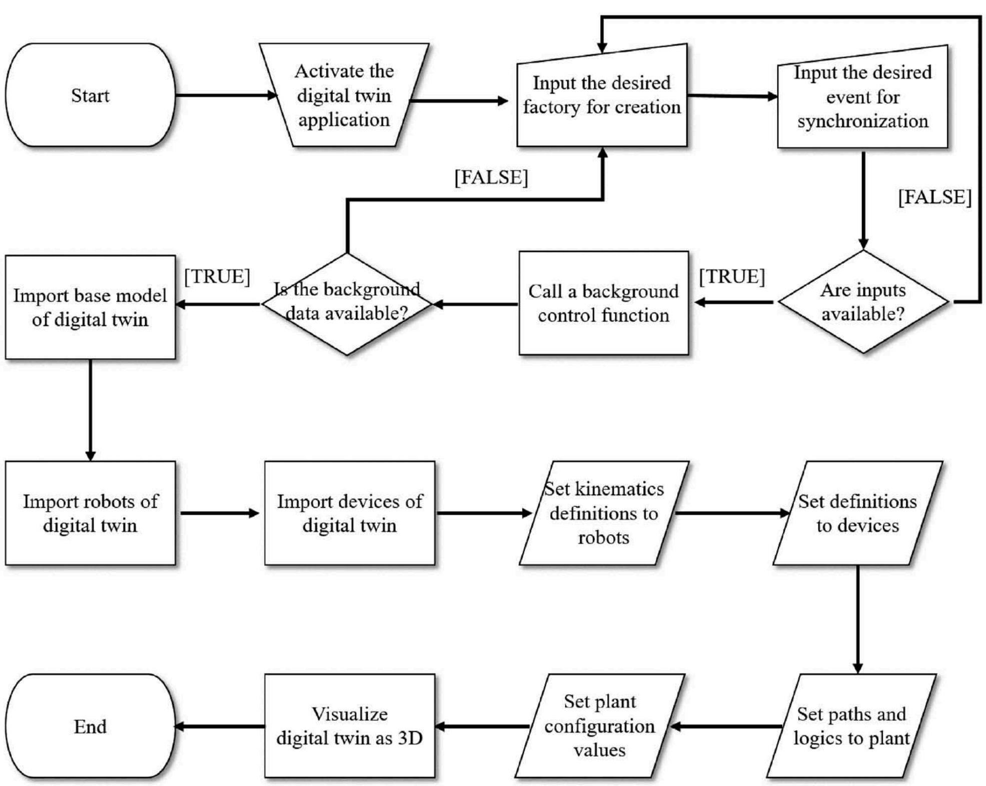

Figure 9. The digital twin creation procedure of the digital twin application.

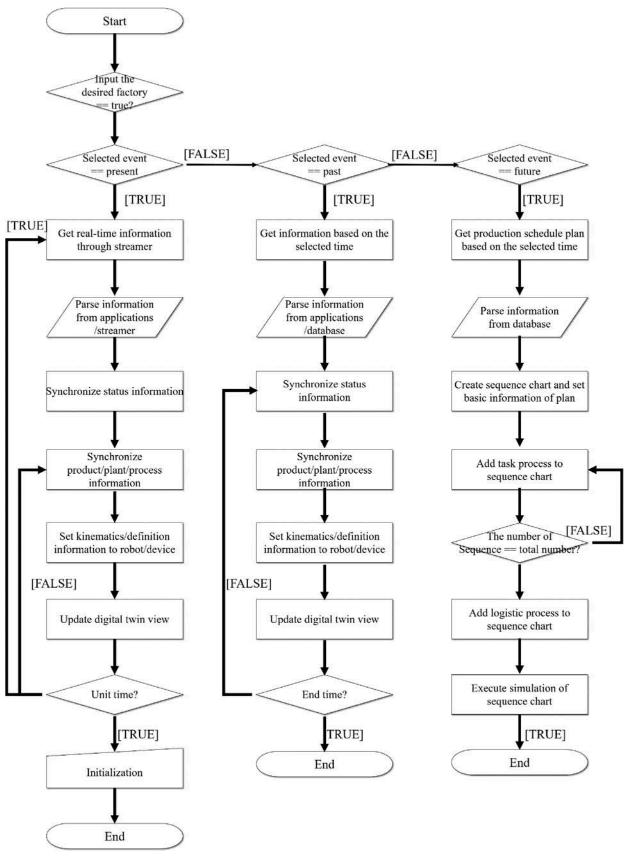

Figure 10. The event-driven information-synchronization procedure of the digital twin application.

information, the digital twin view is updated, and the procedure is terminated.

The procedure for synchronizing the future production schedule begins with the task of retrieving the schedule information from the production schedule database and parsing it. Next, a sequence chart is created based on the basic information of the schedule, and then a task process is created and added to the chart based on the logic of the plant information of the schedule information. After all the task processes are added, logistic processes are added to the chart and generated for each required interval. At this time, the plants responsible for the transfer are added to the chart by creating a corresponding logistic process so that they can be moved according to the path library. The completed sequence chart is used to support decision-making by continuously synchronizing the future events that have been set and executed.

Case study

The research results were validated by implementing the digital twin application concept by using the digital twin adopted in this study and performing the integrated monitoring, tracking, and decision-making support functions by using the time-machine approach. For this, the Daejeon MSF, from among the CMSFs selected as the factories where the model was to be tested, was examined. Next, the figure and components of the digital twin that describe the information and functional unit of the Daejeon MSF is explained. The test environment details for the implementation and experimentation of the digital twin application are then detailed. They illustrate a series of processes for collecting and synchronizing data based on the IIoT sensors and middleware from the Daejeon MSF. Finally, the provision of the tracking, monitoring, and decision-making support functions was illustrated through the operation of the proposed digital twin application and time-machine approach. Lastly, the effectiveness of this study is discussed.

Environments of micro smart factory for validation

For the case study, an actual MSF plant environment is described. The factory to which this study was applied was the MSF test bed in Daejeon, South Korea. Similar to other MSFs, the Daejeon MSF uses the IIoT protocol and middleware to collect data on products, processes, plants, and resources.

The two images in Figure 11 show the MSF and its digital twin in Daejeon. The MSF consists of the manufacturing elements shown in Figure 2, with the same layout. This plant corresponds to the CMSF layer in the physical world of the interoperability-context system architecture of the proposed digital twin application. Among the four layers, the digital twin

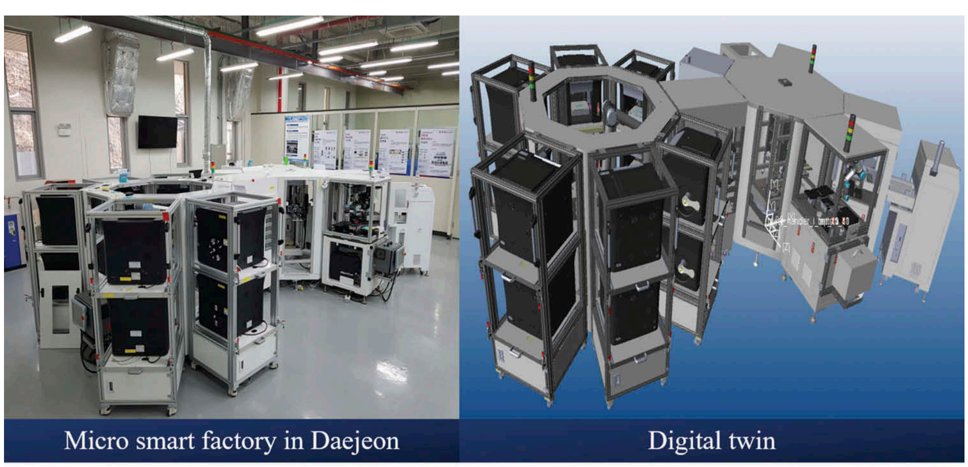

Figure 11. Micro smart factory in Daejeon (left) and digital twin of the micro smart factory (right).

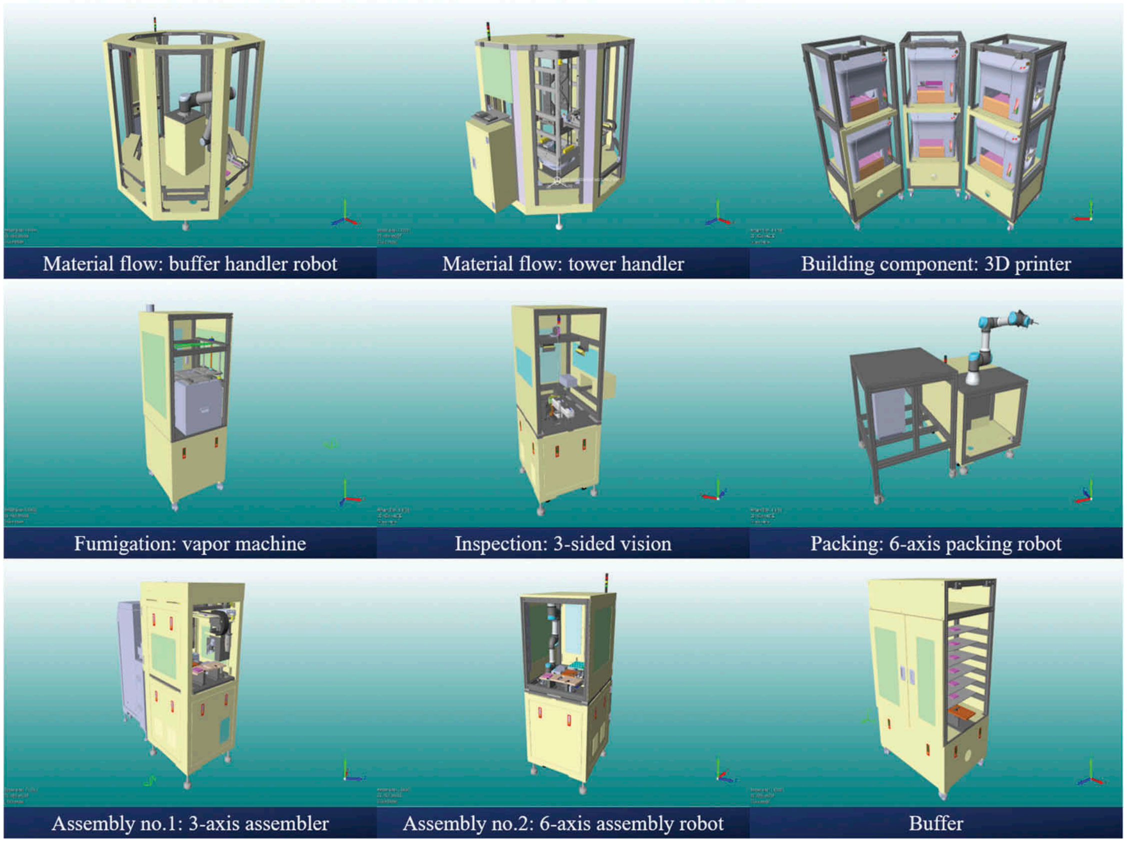

Figure 12. Base models of the digital twin of the micro smart factory in Daejeon.

Table 5. Implementation environment for the digital twin application.

Item	Contents
Operating system	Windows 10
Processor	Intel® Core™ i7-6500U CPU @ 2.50 GHZ
Platform toolset	Visual Studio 2015 (v140)
Programming language	C++/Microsoft Foundation Class
Robotics simulation engine	ezRobotics DMWorks ×64 2.3.14452.0

Table 6. Experimental environment for the digital twin application.

Item	Contents
Data object format	JavaScript Object Notation
Installed memory	64.0 GB
Network architecture	RESTful API
Operating system	Window 7
Processor	Intel® Xeon® CPU E5-2640 v2 @ 2.00GHz

included in the manufacturing application layer was predefined to synchronize the information and functional units on the MSF components.

Figure 12 illustrates the base model of the plants in the digital twin, which depicts the components of the MSF. The digital twin application can provide 3D visualization services by utilizing these 3D models. From top to bottom and from left to right, the first screen shows the material flow plant containing a buffer handler robot that transfers the printed parts from the building-component process. The MSF, the subject of this case study, uses UR10, which is classified as a handler robot in the MSF. The second screen shows a material flow plant using a tower handler, which is classified as a device designed and used separately for the construction of a MSF. The building-component plant involves a process of printing components by using a 3D printer, which is classified as a device, and the MSF has 12 corresponding plants. The fourth screen shows a fumigation plant that performs post-processing of parts, work-in-processes, and assemblies by using a vapor machine. The inspection plant uses the three-sided vision machine to inspect the final product, and the packing plant performs packaging of the inspection-approved products by using universal robot 5 (UR5). The assembly plant no. 1 performs the assembly process by using a three-axis assembler, and the assembly plant no. 2 performs the assembly process by using a six-axis assembly robot.

Plant no. 2 utilizes a buffer handler robot for material flow and packing assembly. It stores a robot base model in the robot library of a robotics simulation engine and fetches and synchronizes the kinematics definition as data for the production operation. In contrast, plant no. 1, which performs the material flow, component building, fumigation, inspection, and assembly by using a tower handler, is a plant that contains a device that is separately designed and used to construct the MSF. It is not included in the robotics simulation engine, and it fetches related information from the background data. Synchronization is enabled by defining devices in advance to operate these devices.

Table 5 shows the implementation environment using Visual Studio 2015 and Windows 10 for the system development. C++ was used as the development language of the

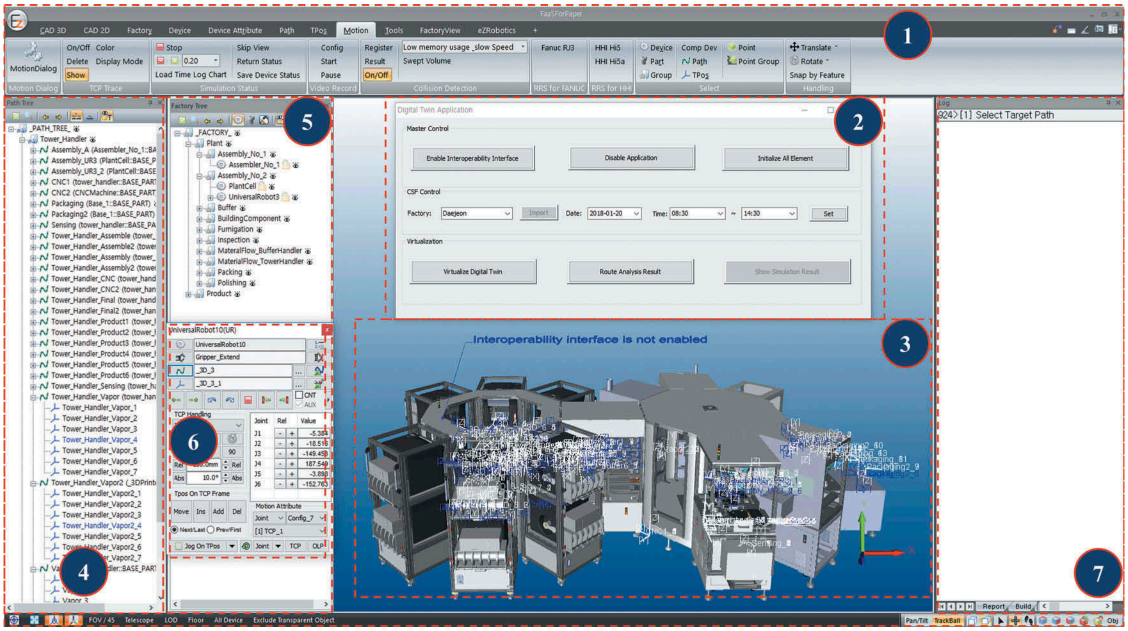

Figure 13. Implementation of the digital twin application.

Table 7. Description of the implementation of the digital twin application.

Index	Item	Contents
1	Robotics simulation engine	Robotics simulation engine in the digital twin application
2	User interface	Performs event handling and synchronization handling with the user interface of the operation module in the digital twin application
3	Digital twin	Virtual factory capable of reflecting MSF components as well as their matching information and configuration unit
4	Path/logic tree	Structure that imports the path/logic library among the background data in the digital twin application
5	Device tree	Structure that imports the device library among the background data in the digital twin application
6	Plant/robot setting	Structure that imports the plant/robot-setting library among the background data in the digital twin application
7	Log window	Outputs the log data after the robotics simulation engine starts to run

operation module and interoperability interface, and Microsoft Foundation Class was used as the user interface of the operation module. The robotics simulation engine of the digital twin application was built using ezRobotics' DMWorks version ×64 2.3.14452.0.

Table 6 shows the base environment for the experiment. The data object format for moving information was implemented using JavaScript Object Notation (JSON), thereby facilitating the reliable operation of the interoperability interface. The RESTful API was used as the network architecture for the IIoT network layer of the interoperability architecture to implement the Web-based communication environment.

Results of implementation

This section details the implementation results of the digital twin application, beginning with the implementation description of the whole digital twin application along with its components.

The initial environment of the digital twin application is shown in Figure 13, and the description of each component is shown in Table 7. The robotics simulation engine is a component of the digital twin application. In the case study, the DMWorks software was used, as described earlier. This engine has a built-in kinematics definition and robot library and can perform functions such as simulation of execution/result analysis. The user interface acts as a controller to operate the digital twin application. It can handle the desired event and operate the synchronization according to the event. As described earlier, the digital twin is a virtual factory that resembles the MSF in Daejeon. Nos. 4–6 in Table 7 describe the background datasets for supporting creation in the digital twin application, which can be viewed by the user, as shown in Figure 12, after the importing is completed.

The user interface can activate the interoperability interface through buttons on the master control tab. When the interoperability interface is activated, information can be shared and utilized, and then the event can be defined by setting the desired point of creation in the CMSF control tab. Each button on the Synchronization tab allows users to view the manufacturing information, status, and simulation results based on the events set in the CMSF control tab.

It is necessary to move, parse, process, and utilize information to operate the digital twin application. Figures 14 and 15 show implemented examples using the JSON message object format. Each JSON message is received through an interoperability interface, which is parsed according to an

```
{
  "header":
  {
    "sourceDeviceId":"oa_",
    "targetDeviceId":"12345",
    "command":"rStream"
  },
  "data":
  {
    "ProcessNumber":1,
    "ProcessList":
    {
      "ProductId":"4af502aa",
      "PartInfo":[{
        "PartId":"9663bde3-f3b6-401e-b7f0-9021ale22646",
        "PartJobKey":1,
        "ResourceId":"CNC"},
        {
        "PartId":"4af502aa-13cc-44c2-8f2c-1cbe9f0b589f",
        "PartJobKey":2,
        "ResourceId":"Vapor"
        },
        {
        "PartId":"4eb7ac2f-2c43-490d-8f07-556f18895178C",
        "PartJobKey":3,
        "ResourceId":"Assembly_No_2"
        }],
      "ProductMBOM":{
        "BOM":
        {
          "Part01":"['o0p0', 'o0p1', ['o0p0', 'o0p1']]",
          "Part02":"['o1p0', 'o1p1', ['o1p0', 'o1p1']]",
          "Part03":"['o2p0', 'o2p1', ['o2p0', 'o2p1']]",
          "Part04":"['o3p0', 'o3p1', 'o3p2', ['o3p0', 'o3p1'], [['o3p0', 'o3p1'], 'o3p2']]]"
        },
        "MaterialType":
        {
          "PartType01":"['a', 'b']",
          "PartType02":"['a', 'a']",
          "PartType03":"['a', 'b']",
          "PartType04":"['a', 'b', 'b']"
        },
        "PartTaskFlow":
        {
          "PartTask01":"[['printer', 'buffer', 'assembly1'], ['printer', 'buffer', 'cnc', 'assembly1'], ['assembly1', 'vision']]",
          "PartTask02":"[['printer', 'buffer', 'cnc', 'assembly0'], ['printer', 'buffer', 'assembly0'], ['assembly0', 'cnc', 'vision']]",
          "PartTask03":"[['printer', 'buffer', 'assembly0'], ['printer', 'buffer', 'assembly0'], ['assembly0', 'cnc', 'vision']]",
          "PartTask04":"[['printer', 'buffer', 'cnc', 'assembly1'], ['printer', 'buffer', 'assembly1'],
                        ['printer', 'buffer', 'cnc', 'assembly0'], ['assembly1', 'cnc', 'assembly0'], ['assembly0', 'vapor', 'vision']]"
        },
        "PartTaskTime":
        {
          "PartTime01":"[[4405, 0, 150], [3216, 0, 600, 150], [150, 60]]",
          "PartTime02":"[[4492, 0, 600, 150], [5024, 0, 150], [150, 600, 60]]",
          "PartTime03":"[[4894, 0, 150], [5930, 0, 150], [150, 600, 60]]",
          "PartTime04":"[[5772, 0, 600, 150], [3284, 0, 150], [5664, 0, 600, 150], [150, 600, 150], [150, 1200, 60]]"
        }
      }
    }
  }
}
```


Product, Process, Resource

Figure 14. Examples of product, process, resource information object for the validation of the digital twin application.


```
{
  "Type": "Schedule",
  "ScheduleID": "FaaS_Schedule_2018_01_20",
  "NumofSeq": 10,
  "PlanStart":3021,
  "Sequence":
  {
    "Process:1":
      {
        "PartId": "o1p1_0",
        "PartName": "o1p1",
        "ProcessName": "loading/transporting/unloading",
        "ProcessType": "bufferHandler",
        "ProcessStart": 3052,
        "PlantStart": "printer1",
        "PlantEnd": "buffer",
        "ProcessEnd": 3067
      },
    "Process:2":
      {
        "PartId": "o1p1_0",
        "PartName": "o1p1",
        "ProcessName": "loading/transporting/unloading",
        "ProcessType": "handler",
        "ProcessStart": 3067,
        "PlantStart": "buffer",
        "PlantEnd": "vapor",
        "ProcessEnd": 3082
      },
    "Process:3":
      {
        "PartId": "o0p1_0",
        "PartName": "o0p1",
        "ProcessName": "loading/transporting/unloading",
        "ProcessType": "bufferHandler",
        "ProcessStart": 3188,
        "PlantStart": "printer5",
        "PlantEnd": "buffer",
        "ProcessEnd": 3203
      }
  }
}
```


Schedule


```
{
  "header":
    {
    "sourceDeviceId":"BH_22345",
    "targetDeviceId":"12345"
    },
  "data":
    {
    "Kinematics":
      {
      "t01":"-90.011,-53.760,-138.987,-167.252,-89.920,44.993",
      "t02":"-90.008,-53.761,-138.985,-167.251,-89.920,44.993",
      "t03":"-90.012,-53.759,-138.986,-167.253,-89.918,44.991",
      "t04":"-90.010,-53.761,-138.988,-167.251,-89.916,44.991",
      "t05":"-90.006,-53.762,-138.985,-167.251,-89.918,44.995",
      "t06":"-90.009,-53.761,-138.986,-167.251,-89.916,44.993",
      "t07":"-90.010,-53.759,-138.989,-167.251,-89.919,44.993",
      "t08":"-90.010,-53.761,-138.987,-167.253,-89.918,44.994",
      "t09":"-90.013,-53.759,-138.987,-167.251,-89.918,44.994",
      "t10":"-90.010,-53.766,-138.988,-167.254,-89.920,44.994",
      "t11":"-90.012,-53.760,-138.990,-167.251,-89.920,44.994",
      "t12":"-90.009,-53.761,-138.987,-167.252,-89.919,44.994",
      "t13":"-90.008,-53.757,-138.988,-167.251,-89.919,44.994",
      "t14":"-90.009,-53.760,-138.989,-167.254,-89.919,44.994",
      "t15":"-90.009,-53.763,-138.989,-167.252,-89.918,44.994",
      "t16":"-90.013,-53.761,-138.987,-167.253,-89.918,44.994",
      "t17":"-90.008,-53.762,-138.987,-167.253,-89.923,44.993",
      "t18":"-90.010,-53.761,-138.989,-167.255,-89.920,44.994",
      "t19":"-90.010,-53.762,-138.987,-167.253,-89.920,44.994",
      "t20":"-90.007,-53.761,-138.987,-167.253,-89.919,44.994",
      "t21":"-90.008,-53.760,-138.987,-167.253,-89.920,44.993",
      "t22":"-90.010,-53.763,-138.989,-167.253,-89.918,44.993",
      "t23":"-90.011,-53.759,-138.985,-167.253,-89.919,44.995",
      "t24":"-90.010,-53.762,-138.989,-167.253,-89.921,44.994",
      "t25":"-90.008,-53.759,-138.986,-167.252,-89.919,44.994"
      },
    "Freq":5,
    "NumKinematics":25
    }
}
```


Plant


```
{
  "header":
   {
    "sourceDeviceId":"R_22342",
    "targetDeviceId":"12345",
    "command":"abStatus"
   },
  "data":
   {
     "StatusPlantID":"A_22344#99",
     "StatusType":2,
     "StatusInfo":99,
     "StatusAnalysisResult":"TrayError"
   }
}
```


Status

Figure 15. Examples of plant, schedule, status information object for the validation of the digital twin application.

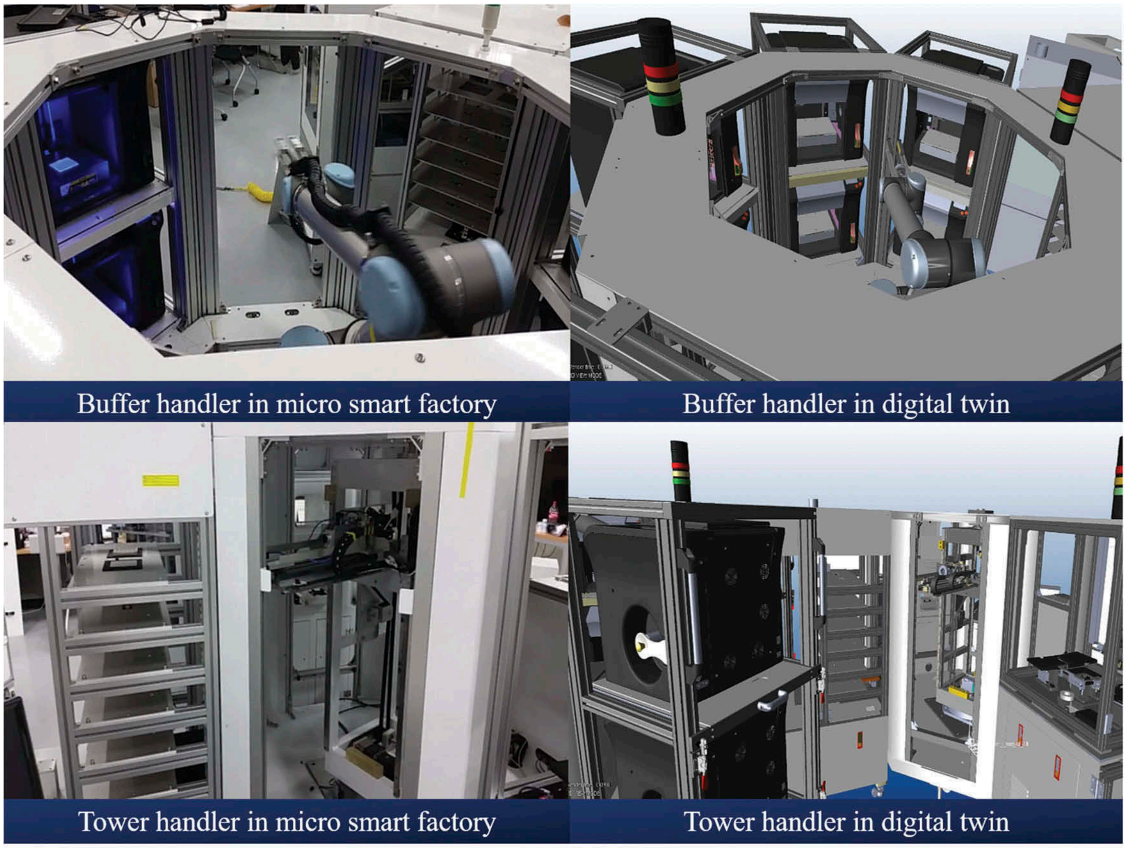

Figure 16. Synchronization of manufacturing information.

Table 8. Kinematics comparison between the physical world and digital twin.

	Buffer handler in micro smart factory						Buffer handler in digital twin					
Time	rx	ry	rz	yaw	pitch	roll	rx	ry	rz	yaw	pitch	roll
1022652	−90.002	−53.751	−138.979	−167.256	−89.998	45.005	−90.002	−53.751	−138.979	−167.256	−89.998	45.005
1023294	5.178	−54.003	−138.894	−167.089	−84.698	44.936	5.178	−54.003	−138.894	−167.089	−84.698	44.936
1023306	8.316	−54.148	−138.84	−166.977	−81.56	44.939	8.316	−54.148	−138.84	−166.977	−81.56	44.939
1023319	12.551	−54.343	−138.773	−166.83	−77.326	44.936	12.551	−54.343	−138.773	−166.83	−77.326	44.936
1023331	17.22	−54.557	−138.695	−166.655	−72.674	44.931	17.22	−54.557	−138.695	−166.655	−72.674	44.931
1023344	22.708	−54.816	−138.596	−166.493	−67.141	44.921	22.708	−54.816	−138.596	−166.493	−67.141	44.921
1023356	27.118	−55.02	−138.523	−166.367	−62.768	44.912	27.118	−55.02	−138.523	−166.367	−62.768	44.912
1023369	31.029	−55.196	−138.466	−166.25	−58.841	44.905	31.029	−55.196	−138.466	−166.25	−58.841	44.905
1023382	34.081	−55.347	−138.418	−166.136	−55.822	44.901	34.081	−55.347	−138.418	−166.136	−55.822	44.901
1023394	36.124	−55.448	−138.383	−166.069	−53.772	44.897	36.124	−55.448	−138.383	−166.069	−53.772	44.897
1023407	37.506	−55.505	−138.359	−166.029	−52.387	44.895	37.506	−55.505	−138.359	−166.029	−52.387	44.895
1023419	38.014	−55.518	−138.348	−166.019	−51.856	44.895	38.014	−55.518	−138.348	−166.019	−51.856	44.895
1023432	38.015	−55.518	−138.348	−166.015	−51.853	45.181	38.015	−55.518	−138.348	−166.015	−51.853	45.181
1023445	38.01	−55.521	−138.343	−166.011	−51.868	46.314	38.01	−55.521	−138.343	−166.011	−51.868	46.314
1023457	38.007	−55.521	−138.344	−166.006	−51.875	48.139	38.007	−55.521	−138.344	−166.006	−51.875	48.139
1023470	38.006	−55.523	−138.345	−166.005	−51.883	50.949	38.006	−55.523	−138.345	−166.005	−51.883	50.949
1023483	38.006	−55.526	−138.343	−166.003	−51.883	54.613	38.006	−55.526	−138.343	−166.003	−51.883	54.613

event so that it can be shared and utilized within the digital twin application.

Figure 16 illustrates the synchronization of the manufacturing information, in which the buffer and tower handlers in the MSF work simultaneously with the buffer and tower handlers of the digital twin, respectively. This way, the users are provided with a high-level visualization of how the actual manufacturing information is being managed.

To verify that the information synchronization was actually performed, the kinematics of the buffer handler in the digital twin was collected, as shown in Table 8. It illustrates accurate information synchronization between a manufacturing element in the physical world and digital twin, and the same results were obtained for the same experiments performed on other plants and robots – 100% match rate.

Figure 17 illustrates the synchronization status of each plant according to its StatusPlantID. At this time, status routings are not provided for the facilities that are not performing any other task for visual efficiency. Instead, the status is reported in red text for any operational problem arising from the problems experienced during repair or communication connection.

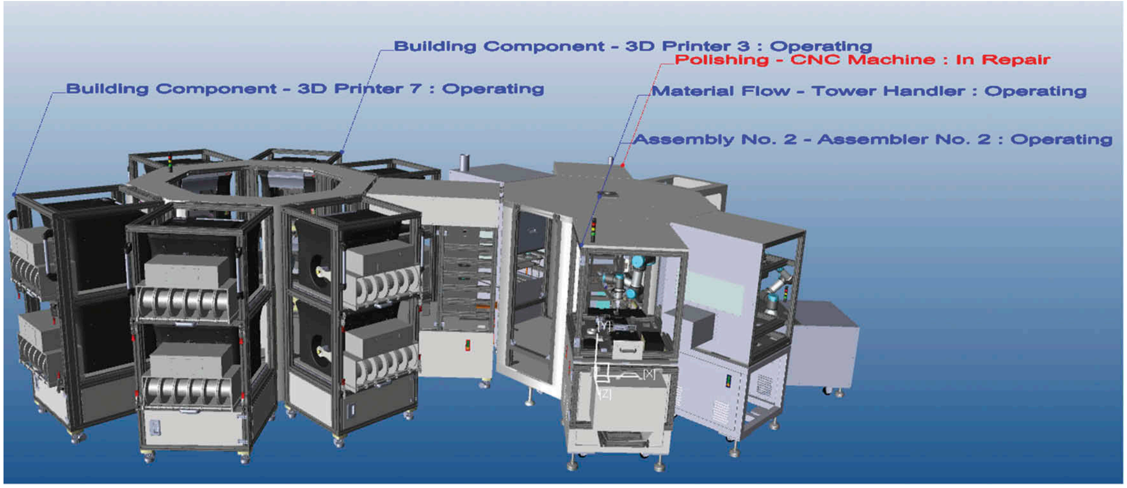

Figure 17. Implementation of status routing.

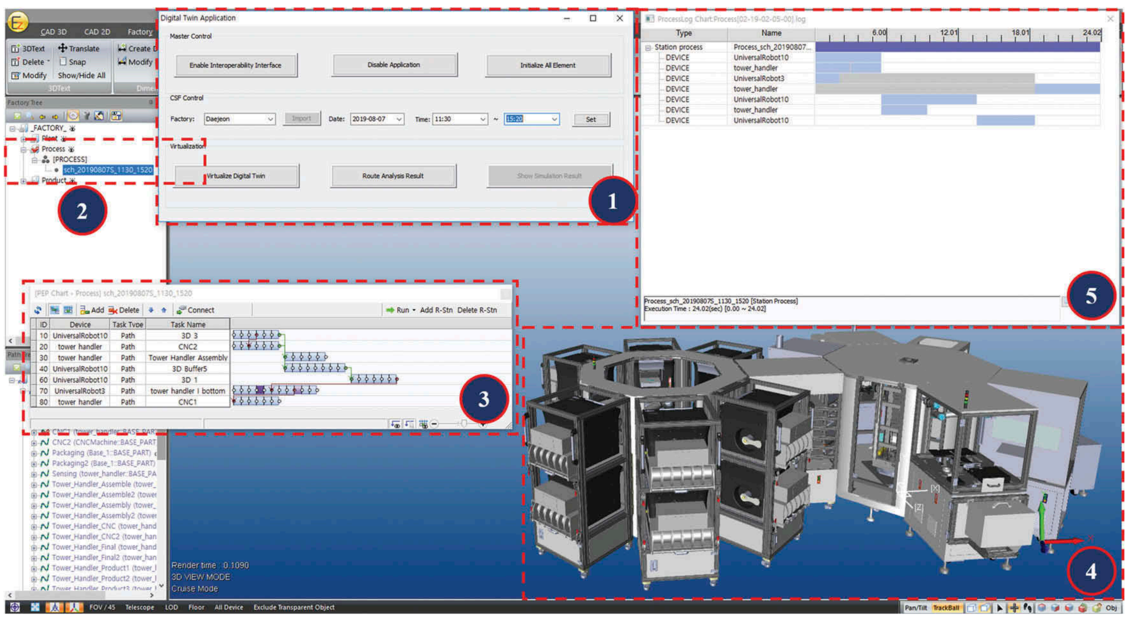

Figure 18. Creation of a simulation sequence.

Figure 18 illustrates the screen for the monitoring and decision-making support of the future production schedule through the creation of simulation sequences. First, a sequence is created by setting the future viewpoint in the user interface, and the production schedule at that point is added to the digital twin application, as in (2). In the DMWorks software, the production schedule is generated from the same chart as in (3) before being synchronized in the digital twin. If users click the 'Show simulation result' button of the user interface, they could view the simulation result, such as in (5), because the robotics simulation engine is utilized. Based on this result, users can perform decision-making.

Figure 19 illustrates peer-to-peer comparison of conceptually same objects based on past, present, and future information. A relatively opaque object shows the actual operation of the selected point-of-event. It provides users with an advanced monitoring function by comparing with the plan information at the event, which is represented by a relatively transparent object in Figure 19. This is implemented by projecting three point-in-time information-synchronization procedures to multiple objects according to the defined events. The difference between the planned and actual operation can be used to confirm delayed operation, inappropriate manufacturing execution, the cause of machine failure, and the cause of the poor quality.

Conclusions

This study was conducted by setting the CMSF in the FaaS system and incorporating the concepts of personalized production and distributed manufacturing as the research domains. This study designed and implemented a digital

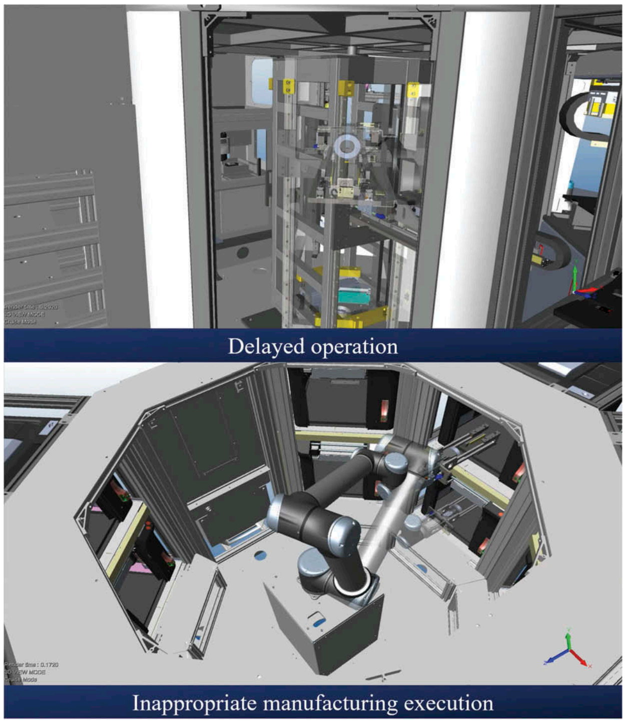

Figure 19. Implementation of advanced monitoring routing.

twin to solve the problems of various product groups and dynamic situations of the personalized production, as well as the distributed manufacturing system having heterogeneous manufacturing systems. The digital twin is designed to monitor the present in real-time, track the past, support decision-making by predicting the future to handle a variety of product lines, so it can handle dynamic situations and the multiple manufacturing systems of CMSF.

Through the aforementioned problem definition, the system scenario for solving the problem was defined. The scenario is summarized, and the respective behaviors of the four layers were defined for the scenario. On this basis, a sketch was drawn of a digital twin application that can perform the roles of final monitoring, tracking, decision-making, and support. The four layers of the components were then defined and summarized in the description of the view and information flow of the interoperability-context system architecture. Accordingly, it was shown how each component of each layer should perform its functions and how data/information should be transferred between the components. Next, the design was implemented for each detailed component to project the information and functional unit to the digital twin application, which plays a key role in the digital twin application. In addition, the components project information to the operation-module functions and the associated parameters. Finally, procedures were provided to illustrate how the entire operation performed by the digital twin application has procedures corresponding to each event.

As a case study, the MSF in Daejeon was selected as the target plant, and the DMWorks software of ezRobotics was selected as the robotics simulation engine. The network was constructed based on the RESTful API, the base model in the MSF was CAD-modeled, and device definition was performed for the plant using the device. In addition, the environment of each layer of the interoperability-context system architecture was built to exchange the information used in the system interoperability through the JSON format. In the experimental environment, the digital twin application was actually implemented along with the implementation of the elements in the operation module. As a result, the system was operated according to the defined procedure, along with the tracking, real-time monitoring, and decision-making support roles of the system.

This study showed a detailed system design by continuously lowering the hierarchy of the interoperability-context system architecture of the digital twin application, which was directly implemented and validated in the case study. In addition, unlike most other studies, an IIoT-based manufacturing application was applied to the MSF in this study, as well as the distributed manufacturing system and CMSF, to perform integrated monitoring, tracking, and decision-making support roles. Toward this end, a system was designed based on the time-machine approach proposed by Lee, Beghari, and Kao (2015), and detailed components and functions were derived to implement the approach. In addition, this study considered the construction and utilization of a factory-level digital twin; this is mainly conceptual, and thus, this study can be considered as a precedent study for other types of factories. Moreover, the study provided example of an advanced solution to the cost and performance hurdles of personalized production and heterogeneous distributed manufacturing systems. In the future, it would be necessary to study functions that can pre-diagnose functions and facilities that can detect collisions before they occur based on the relevant system so that they can notify the system in advance of an impending failure, thereby increasing the added value of the system. Moreover, an advanced architectural framework and a reference virtual representation should be developed to apply to various manufacturing sites.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the ICT R&D program of MSIP/ITTP [B0364-15-1008, Development of an Open FaaS IoT Service Platform for Personalized Production].

References

Alam, K. M., and A. E. Saddik. 2017. "C2PS: A Digital Twin Architecture Reference Model for the Cloud-Based Cyber-Physical Systems." *IEEE Access* 5. doi:10.1109/ACCESS.2017.2657006.

Bielefeldt, B., J. Hochhalter, and D. Hartl. 2015. "Computationally Efficient Analysis of SMA Sensory Particles Embedded in Complex Aerostructures Using a Substructure Approach." *ASME 2015 Conference on Smart Materials, Adaptive Structures and Intelligent Systems*, American Society of Mechanical Engineers, Colorado, pp. V001T02A007.

Boschert, S., and R. Rosen. 2016. "Digital Twin-The Simulation Aspect." In *Mechatronic Futures: Challenges and Solutions for Mechatronic Systems and Their Designers*. Springer, 59–74.

Doukas, M., F. Psarommatis, and D. Mourtzis. 2014. "Planning of Manufacturing Networks Using an Intelligent Probabilistic Approach for Mass Customized Products." *International Journal of Advanced Manufacturing Technology* 74: 1747–1758. doi:10.1007/s00170-014-6121-z.

Du, X., J. Jiao, and M. M. Tseng. 2006. "Understanding Customer Satisfaction in Product Customization." *International Journal of Advanced Manufacturing Technology* 31: 396–406. doi:10.1007/s00170-005-0177-8.

Durão, L., A. Christ, E. Zancul, R. Anderl, and K. Schützer. 2017. "Additive Manufacturing Scenarios for Distributed Production of Spare Parts." *International Journal of Advanced Manufacturing Technology* 93: 869–880. doi:10.1007/s00170-017-0555-z.

Fan, Y. S., and G. Q. Huang. 2007. "Networked Manufacturing and Mass Customization in the Ecommerce Era: The Chinese Perspective." *International Journal of Computer Integrated Manufacturing* 20 (2–3): 107–114. doi:10.1080/09511920601020631.

Fang, C., X. Liu, P. Pardalos, and J. Pei. 2016. "Optimization for a Three-Stage Production System in the Internet of Things: Procurement, Production and Product Recovery, and Acquisition." *International Journal of Advanced Manufacturing Technology* 83: 689–710. doi:10.1007/s00170-015-7593-1.

Gabor, T., L. Belzner, M. Kiermeier, M. T. Beck, and A. Neitz. 2016. "A Simulation-Based Architecture for Smart Cyber-Physical Systems." In *IEEE International Conference on Autonomic Computing (ICAC)*, IEEE, 374–379.

Grieves, M. 2014. "Digital Twin: Manufacturing Excellence Through Virtual Factory Replication." http://www.apriso.com

Han, H., H. Bae, H. Kang, J. Son, and H. Kim. 2016. "Multi Agent 3D Printer and Robot System for Mass Personalization FaaS Platform." In *2016 International Conference on Information and Communication Technology Convergence (ICTC)*, IEEE, 1129–1131. Jeju.

Huang, G. Q., L. Li, T. L. Lau, and X. Chen. 2007. "A Generic and Extensible Information Infrastructure Framework for Mass-Customizing Platform Products." *International Journal of Computer Integrated Manufacturing* 20 (2–3): 292–306. doi:10.1080/09511920601150669.

Kang, H. S., S. D. Noh, J. Y. Son, H. Kim, J. H. Park, and J. Y. Lee. 2018. "The FaaS System Using Additive Manufacturing for Personalized Production." *Rapid Prototyping Journal* 24 (9): 1486–1499. doi:10.1108/RPJ-11-2016-0195.

Kühn, W. 2006. "Digital Factory – Simulation Enhancing the Product and Production Engineering Process." In *Proceedings of the 2006 Winter Simulation Conference*, IEEE, 1899–1906. Monterey, CA.

Kumar, A. 2007. "From Mass Customization to Mass Personalization: A Strategic Transformation." *International Journal of Flexible Manufacturing System* 19: 533–547. doi:10.1007/s10696-008-9048-6.

Lee, J., B. Beghari, and H. A. Kao. 2015. "A Cyber-Physical Systems Architecture for Industry 4.0-Based Manufacturing Systems." *Manufacturing Letters* 3: 18–23. doi:10.1016/j.mfglet.2014.12.001.

Liao, Y., E. F. R. Loures, and F. Deschamps. 2018. "Industrial Internet of Things: A Systematic Literature Review and Insights." *IEEE Internet of Things Journal* 5 (6): 4515–4525. doi:10.1109/JIOT.2018.2834151

Lu, Y., and J. Cecil. 2015. "An Internet of Things (Iot)-Based Collaborative Framework for Advanced Manufacturing." *International Journal of Advanced Manufacturing Technology* 84: 1141–1152.

Mai, J., L. Zhang, F. Tao, and L. Ren. 2016. "Customized Production Based on Distributed 3D Printing Services in Cloud Manufacturing." *International Journal of Advanced Manufacturing Technology* 84: 71–83. doi:10.1007/s00170-015-7871-y.

Morgan, J., and G. D. O'Donnell. 2017. "Multi-Sensor Process Analysis and Performance Characterisation in CNC Turning-A Cyber Physical System Approach." *International Journal of Advanced Manufacturing Technology* 92: 855–868. doi:10.1007/s00170-017-0113-8.

Mourtzis, D., M. Doukas, and C. Vandera. 2017. "Smart Mobile Apps for Supporting Product Design and Decision-Making in the Era of Mass Customisation." *International Journal of Computer Integrated Manufacturing* 30 (7): 690–707. doi:10.1080/0951192X.2016.1187295.

Mulani, T. T., and S. V. Pingle. 2016. "Internet of Things." *International Research Journal of Multidisciplinary Studies* 2 (3): 1–4.

Ngai, E. W. T., K. K. L. Moon, F. J. Riggins, and Y. Yi Candace. 2008. "RFID Research: An Academic Literature Review (1995–2005) and Future Research Directions." *International Journal of Production Economics* 112 (2): 510–520. doi:10.1016/j.ijpe.2007.05.004.

Rosen, R., G. Wichert, G. Lo, and K. D. Bettenhausen. 2015. "About the Importance of Autonomy and Digital Twins for the Future of Manufacturing." *IFAC-PapersOnLine* 48 (3): 567–572. doi:10.1016/j.ifacol.2015.06.141.

Schirner, G., D. Erdogmus, K. Chowdhury, and T. Padir. 2013. "The Future of Human-in-The-Loop Cyber-Physical Systems." *Computer* 46 (1): 36–45. doi:10.1109/MC.2013.31.

Silveira, G. D., D. Borenstein, and F. S. Fogliatto. 2001. "Mass Customization: Literature Review and Research Directions." *International Journal of Production Economics* 72: 1–13. doi:10.1016/S0925-5273(00)00079-7.

Son, J. Y., H. C. Kang, H. C. Bae, E. S. Lee, H. N. Han, J. H. Park, and H. Kim. 2015. "IoT-based Open Manufacturing Service Platform for Mass Personalization." *The Journal of the Korean Institute of Communication Sciences* 33 (1): 42–47.

Spring, M., and L. Araujo. 2013. "Beyond the Service Factory: Service Innovation in Manufacturing Supply Networks." *Industrial Marketing Management* 42: 59–70. doi:10.1016/j.indmarman.2012.11.006.

Tammaro, A., A. Segura, A. Moreno, and J. R. Sanchez. 2017. "Extending Industrial Digital Twins with Optical Object Tracking." *Congreso Español de Informática Gráfica, Sevilla* 2017: 23–26.

Tao, F., J. Cheng, Q. Qi, M. Zhang, H. Zhang, and F. Sui. 2017. "Digital Twin-Driven Product Design, Manufacturing and Service with Big Data." *International Journal of Advanced Manufacturing Technology* 94 (9–12): 1–14.

Tseng, M. M., R. J. Jiao, and C. Wang. 2010. "Design for Mass Personalization." *CIRP Annals – Manufacturing Technology* 59: 175–178. doi:10.1016/j.cirp.2010.03.097.

Wang, X. V., Z. Kemeny, K. J. Váncza, and L. Wang. 2017. "Human-Robot Collaborative Assembly in Cyber-Physical Production: Classification Framework and Implementation." *CIRP Annals – Manufacturing Technology* 66: 5–8. doi:10.1016/j.cirp.2017.04.101.

Yao, S., X. Han, H. Yang, Y. Rong, S. H. Huang, D. W. Yen, and G. Zhang. 2007. "Computer-Aided Manufacturing Planning for Mass Customization: Part 1, Framework." *International Journal of Advanced Manufacturing Technology* 32: 194–204. doi:10.1007/s00170-005-0327-z.

Yao, X., and Y. Lin. 2015. "Emerging Manufacturing Paradigm Shifts for the Incoming Industrial Revolution." *International Journal of Advanced Manufacturing Technology* 85: 1665–1676. doi:10.1007/s00170-015-8076-0.