How to model and implement connections between physical and virtual models for digital twin application

Author: Haifan Jiang[1], Shengfeng Qin[2], Jianlin Fu[1], Jian Zhang [1]*, Guofu Ding [1]*
[1]Institute of Advanced Design and Manufacturing, Southwest Jiaotong University, Chengdu 610031, People's Republic of China
[2]School of Design, Northumbria University, Newcastle upon Tyne NE1 8ST, UK

Corresponding author: Guofu Ding, Jian Zhang
E-mail address: dingguofu@swjtu.edu.cn
E-mail address: jerrysmail@263.net

Abstract: Digital twin (DT) is a virtual mirror (representation) of a physical world or a system along its lifecycle. As for a complex discrete manufacturing system (DMS), it is a digital model for emulating or reproducing the functions or actions of a real manufacturing system by giving the system simulation information or directly driven by a real system with proper connections between the DT model and the real-world system. It is a key building block for smart factory and manufacturing under the Industry 4.0 paradigm. The key research question is how to effectively create a DT model during the design stage of a complex manufacturing system and to make it usable throughout the system's lifecycle such as the production stage. Given that there are some existing discussions on DT framework development, this paper focuses on the modeling methods for rapidly creating a virtual model and the connection implementation mechanism between a physical world production system at a workshop level and its mirrored virtual model. To reach above goals, in this paper, the discrete event system (DES) modeling theory is applied to the three-dimension DT model. First, for formally representing a manufacturing system and creating its virtual model, seven basic elements: controller, executor, processor, buffer, flowing entity, virtual service node and logistics path of a DMS have been identified and the concept of the logistics path network and the service cell is introduced to uniformly describe a manufacturing system. Second, for implementing interconnection and interaction, a new interconnection and data interaction mechanism between the physical system and its virtual model for through-life applications has been designed. With them, each service cell consists of seven elements and encapsulates input/output information and control logic. All the discrete cells are constructed and mapped onto different production-process-oriented digital manufacturing modules by integrating logical, geometric and data models. As a result, the virtual-physical connection is realized to form a DT model. The proposed virtual modeling method and the associated connection mechanism have been applied to a real-world workshop DT to demonstrate its practicality and usefulness.

Keywords: Digital twin; Cyber-physical system; Discrete event system modeling theory; Virtual-physical connection

1. Introduction

Under the trend of economic globalization and mass customization, manufacturing enterprises face a fierce competitive environment. The market and user needs/requirements are changing fast, therefore more and more customized products are needed on the market in a reasonable shorter time [1]. This requires fast product design and its manufacturing system design and commissioning in a concurrent and smart fashion. This paper assumes that the product design and process information are available for a manufacturing system design, therefore we focus our research on how to rapidly and effectively design, commission, and optimize a manufacturing system at a shop floor level. As an important manifestation of manufacturing systems, discrete manufacturing systems (DMS) are widely used with flexibility. When information technology, cyber system and physical manufacturing system are interconnected for smart manufacturing, the material flow, information flow and control flow of the manufacturing system are intertwined in shop floors [2]. It brings great difficulties and challenges for manufacturers to design and analyze a complex DMS.

In this context, manufacturers need a digital (or virtual) model created during the design stage to mirror and interconnect a complex physical manufacturing system for the system simulation, prediction, analysis, and optimization throughout the system's lifecycle. Nowadays, this connected model is known as digital twin (DT) which emphasizes on bidirectional data-exchanges and controls between the physical model in a real world and its virtual model [3]. With internet of thing (IoT) [4], sensor technology and big data [5] as enablers, the real-time data and status of the physical system (physical twin) could be updated to its virtual model (digital twin), and the simulation and analysis results in cyber space could act on the physical system in turn to form closed-loop control, making DT hold the potential for simulation, monitoring and scheduling applications. DT enables optimizations and decisions making with the virtual system relying on the same data that is updated in real-time within the physical system, through synchronization enabled by sensors and controls.

The logical modeling and implementing the virtual-physical connections are prerequisites for simulation applications throughout production lifecycle, while they are also challenging tasks. Firstly, DMSs comprise various heterogeneous devices and organizational structures, and couple the products, production, and operation together. In order to comprehensively and intuitively represent heterogeneous DMSs, a generic logical model with unified formal and graphical definitions, and high-level model abstraction and encapsulation, must be developed. While the discrete event systems (DES) modeling theory [6-9] is relatively mature, providing a theoretical ground for logical modeling of DMSs. Secondly, in order to make the logical model available throughout the system's lifecycle such as the production stage, a virtual-physical connection needs to be established between a physical system and its logical model. However, the coupling of them leads to a large number of interactions (i.e. physical-physical, virtual-virtual, and virtual-physical) and data (e.g. real-time data of equipment, production orders, execution feedback, simulation results, etc.) during production, which in turn poses a great challenge for data interaction mechanism. Recently, big data processing technology has developed rapidly [10], and many message middleware and databases have been developed, such as Kafka [11] and Redis [12], which provide technical support for heterogeneous device/system communication and real-time data interaction.

The contributions of this article are twofold. In theory, the DES modeling theory is applied to a three-dimensional digital twin model [13] to develop the formal and graphical modeling specifications for the key elements and relationships identified in heterogeneous DMSs. In practice, firstly, an interconnection scheme is designed for linking the physical systems and their representation models to make them DTs via integrating logical, geometric and data models. Secondly, a data interaction mechanism is proposed to realize the interoperability among physical systems, service systems and simulation systems; and on this basis, the simulation mechanism towards DT application is discussed. Finally, a virtual model construction system is developed for rapidly prototyping a DT. The proposed method enables logical modeling, virtual-physical connection, and simulation applications, as demonstrated by a real-world workshop DT.

The remainder of this paper is organized as follows. After reviewing the related work in Section 2, how to model and implement connections between physical and virtual models for digital twin application is presented in Section 3. Implementation of the proposed method is illustrated by an application case in Section 4, followed by the discussion and conclusion section.

2. Related work

This article focuses on the logical modeling at a concept level and the implementation of virtual-physical connections in practice towards DT applications. This section provides the state-of-the-art review on DT modeling and application related research, and identifies the research gaps in this field.

2.1 Digital twin and CPS related modeling framework, specifications, and standards

DT was originally derived from product lifecycle management (PLM) [14], evolved from digital product definition (DPD) and digital mock-up (DMU) [15]. In the narrow sense, DT refers to a high-fidelity model or a digital equivalent of physical object [3,16], which integrates data from all phases of product lifecycle. The above ideas spawned the concept of products DT. While a cyber-physical system (CPS) is characterized by a physical asset and its DT, where the DT refers to a virtual model that can be connected to the physical asset in bi-directions [10]. That is, DT is a prerequisite for implementing a CPS. The cyber-physical production system (CPPS) could be viewed as a network of CPS-powered physical assets [17], which relies on the production DT, e.g. shop floor digital twin (DTS) [3]. And this article focuses on the modeling and implementation of production DT.

In order to realize the vision of DT, scholars have proposed a unique modeling framework from different perspectives. Grieves first proposed a general and standard DT modeling reference framework, known as three-dimensional DT model [13], including three dimensions namely physical entities, virtual models, and their connections. Tao [13,18] extended it to a five-dimensional DT model by adding the services and DT data dimensions. Above models point the way for the creation of products DT or productions DT, but they are still abstract and lack of pertinence in the implementation process.

To implement collaborative manufacturing and assembly of micro devices, an IoT-based cyber physical framework was proposed by Cecil [4], which consists of five collaborative entities, namely cyber physical manager, cloud, the cyber components, the cyber physical interactions and the physical equipment. Based on this framework, an assembly plan and command could be generated, and the process of assembly could be simulated and monitored. Liu et al. [17] introduced the term CPS node (CPSN) which integrates manufacturing resources and corresponding DT model. On this basis, a systematic framework of DT-based CPPS was proposed. This framework presented how to configure a CPSN through the manufacturing resource virtualization, and how to orchestrate various CPSNs as a CPPS. From the shop floor manufacturing process perspective, the product DT, process DT, and operation DT were set up by Bao [19], and the interoperation among above DTs were also explored. Using DT technology can enhance the performance of a structural parts machining cell.

Several information-centric modeling specifications and standards contribute to DT or CPS modeling. IDEF-0 is an information centric system engineering methodology [4], which was used to design the interactions among cyber physical components. In [17], DT model was divided into information model, geometric model, and function model. And the information model was uniformly modeled by ontology to facilitate the data exchange of heterogeneous manufacturing resources. Asset Administration Shell (AAS) is a logical representation of an asset; the combination of the asset and its AAS forms the core component of Industry 4.0 [20]. AAS describes the technical functions of assets and their relationship with other assets, which can be considered as a bridge between tangible assets and the IoT world, or a data model of DT [21]. Liu et al. [22] suggested using OPC unified architecture (OPC UA) and MTConnect for information modeling of DT machine tools. Automation Markup Language (AutomationML) is useful to describe the DT-related attributes for data exchange between DT and other systems [23]. For instance, in [19], AutomationML was used to model a machining cell to describe the connection and interaction relationships among resource, process, and product.

From the literatures above, it can be observed that various modeling specifications and standards (e.g. ontology, AAS, OPC UA, MTConnect, AutomationML, etc.) could be applied to DT or CPS modeling. These researches mainly focused on data structure and semantics [17], information modeling [22] and the data exchange [19,23] of manufacturing assets; few studies were on the logical representation of elements and relationships of a manufacturing system. Both logical model and data model are the pillars of DTs. From the perspective of simulation modeling, the proper identification and representation of system elements and relationships are the basis for logical modeling and simulation. DES modeling theory has advantages in unified modeling specifications. Based on DES modeling theory, a large number of modeling methodologies have been proposed (e.g. ACD [7], Petri nets [8], FSM [9] and DEVS [6,8]), which could be used to model the internal logics of physical entities and interaction mechanisms among physical entities.

2.2 Existing ways of connecting physical objects and their virtual models in practice for industry application

Nowadays, applications of DTs could be found in various stages of product and production, e.g. product/production designing [1,5], job shop scheduling [24], production management and control [2], prognostics and health management (PHM) [13] and so forth. Based on PLM ideas, Zhuang [25] expounded the product DT in term of connotations, implementation approach, architecture, and trends. Tao et al. [5,26] discussed DT-driven product design, manufacturing, and service. And the Siemens explored the potential of DT across the whole value chain including product, production, and performance [27]. Regarding to CPS, digital twin machine tools (DTMT) [22], digital twin production lines [1,28] and DTS [3] were discussed. Shafiq [29] modeled the virtual engineering object and virtual engineering process to realize CPS for industry 4.0. Ding [30] defined a DT-based CPPS, which combines the DT product and DTS, providing the conceptual basis for DT-based operations and applications.

Although most applications are still in their infancy, several DT applications are full-fledged and representative, e.g. cyber-physical machine tool (CPMT) [22,31] and DT-driven rapid designing of production line [1,28].

Liu et al. [22] proposed a generic system architecture for CPMT, in which the physical devices (including machine tool, cutting tools, workpieces, CNC controller and data acquisition devices) and DTMT are connected via networks (e.g. Ethernet internet, Profinet, etc.). The DTMT is enabled by four development methodologies, i.e. data acquisition, data fusion, information modeling based on MTConnect and OPC UA, database and intelligent algorithms. With the interoperability of

MTConnect and OPC UA as an enabler, a CPMT platform was developed, supporting data interaction and interoperation among machine tools and different software applications [31].

Following an idea of "iterative optimization between static design and dynamic execution", Zhang [1] and Liu [28] discussed the DT-driven rapid designing of production line, where the cyber-physical synchronization was guaranteed by the binging and mapping among PLC I/O point on the simulation model and I/O address on equipment. It has been successfully applied in a hollow glass production line [1] and a sheet material production shop floor [28].

From the literatures above, it can be observed that the DT applications have attracted good attentions, spawning a lot of valuable research results. However, few studies have attempted the virtual-physical connection and interaction in the workshop-level, which are the prerequisites for simulation applications throughout production lifecycle. On the other hand, the development of DT practical applications is still at a very early stage [10]. The physical devices and DTMT were connected to a CPMT in [22], but mainly focusing on information modeling and data applications, rather than logical modeling and simulation applications. Although the virtual-physical synchronization was addressed in [1] and [28], it is mainly suitable for semi-physical simulation at equipment-level. Therefore, from an application viewpoint of production simulation, researches on the workshop-level connection scheme and data interaction mechanism are still needed urgently.

3. Proposed method

Three key concepts are extracted from the three-dimensional DT model [13]: physical model, virtual model, and the connections of them. The proposed methodology consists of three parts, each targeting one of these concepts, namely identification of elements and relationships of physical systems, representation of physical systems in virtual space, and connection between physical systems and their representation models to make digital twins.

3.1 Identification of physical systems elements and relationships

DMS has various manifestations and diverse structure. It can be divided into one-piece small-batch production, mass production and mass customization according to the production mode. It can also be divided into serial, parallel, assembly, disassembly, and flexible manufacturing units according to the production organization relationship [32]. In order to study the essential properties of DMSs, the first step is to identify the basic elements and relationships in DMSs. The components of a manufacturing system can be summarized as 4M1E (i.e. man, machine, material, method, and environments). Its relationships can be described in a hierarchical production structure with networked logistics relationship from the viewpoint of production organization. Hierarchical structure is the essential attribute of DMSs. The production elements include subsystems or units, which in turn act as constituent elements of a higher-level system. As for logistics relationships, they portray the material exchange relationship among logistics equipment, processing equipment and storage equipment. They can be classified into the logistics in the units and the logistics among the units. The former is relatively simple, while the latter is more complex because of multiple equipment participation, long logistics distances and complicated logistics routes. Fig. 1 depicts the basic elements and relationships identified in DMSs.

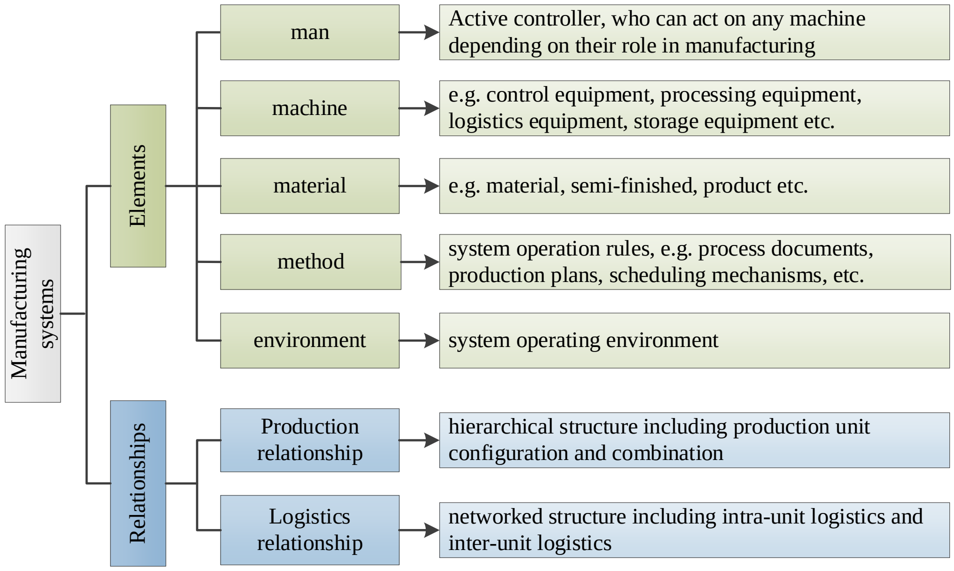


Fig. 1 Basic elements and relationships identified in discrete manufacturing systems

3.2 Representation of physical systems in virtual spaces

As mentioned earlier, DES modeling theory is relatively mature and can be used to describe the internal logic of entities and the interaction between entities. However, the lack of encapsulation of details makes it difficult to describe complex systems directly. Therefore, the identified elements and relationships could be encapsulated into corresponding representation models according to their functions and behaviors. The internal operation logic of these representation models could be described via DES modeling theory, e.g. finite state machine [9]. The interactions between models could be described by defining their input/output interfaces. In order to realize the unified expression of heterogeneous DMSs, a set of graphical and formal models are provided.

Fig. 2 outlines these representations models from three levels of composition, structure, and operation. In composition level, seven elements describe the basic elements of DMS. In the structure level, the service cells (*SC*) and the logistics path networks (*LPN*) are proposed to represent the production relationship and logistics relationship. Finally, multi-virtual service nodes (*VSNs*) based logical model is outlined in operation level, which combines production processes, service cells, and logistics path networks into an organic whole to achieve an orderly logical flow of parts.

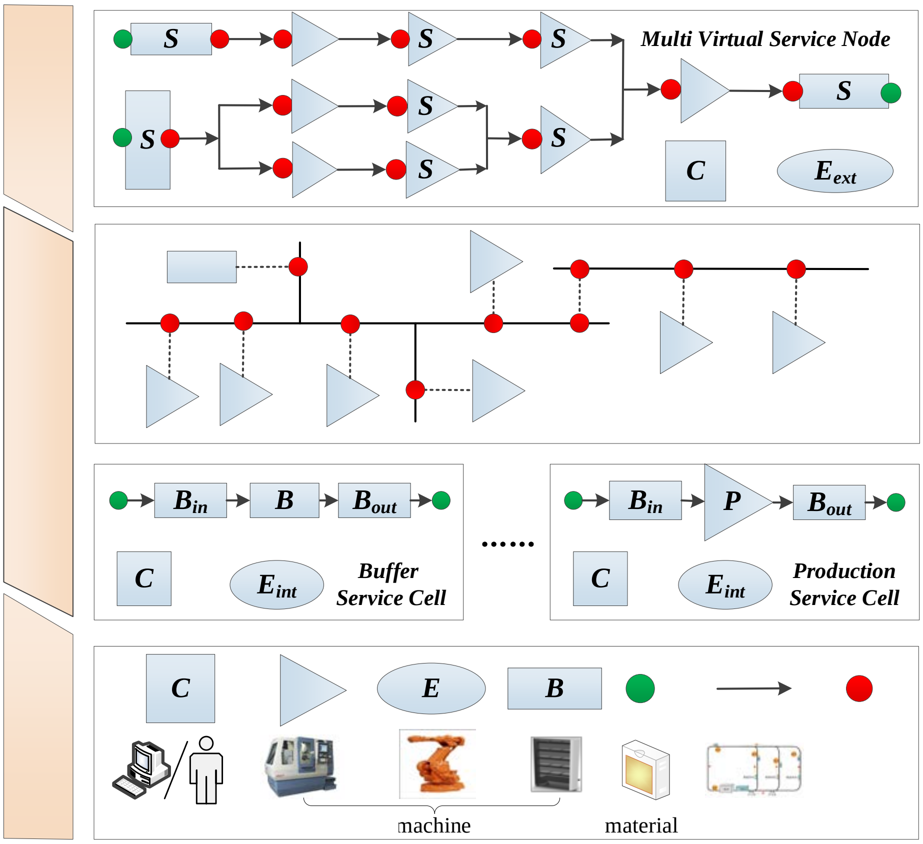
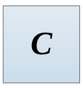
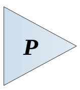


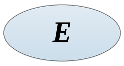
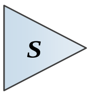
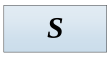
Fig.2 Representations of manufacturing systems

3.2.1 Representation of elements

In order to unify the elements of DMS identified in the previous section, the Seven-element model is proposed (see the bottom of Fig.2). Each element is an individual object acting as a special role in the system, which encapsulates internal logic and interfaces with external environment. The Controller (*C*) determines the functions and operating logic. The Processor (*P*), Executor (*E*), and Buffer (*B*) form the basic physical structure, which represents the "machine" in physical systems. The Flowing entity (*F*) - a green dot in Fig. 2 represents the material; the Logistics path (*L*)-an arrow line in Fig.2 indicates the flowing direction and represents the material flow rule; and the *VSN* - a red dot in Fig 2, associates the above elements to build a virtual operating environment. These seven basic elements are associated with other secondary elements. The graphical and algebraic specification of the key element representation models is given in Table 1 and explained next. In this article, the symbol "<>" represents the composition and characteristics of a logical model; and the symbol "×" represents the logical relationship between logical models.

Table 1 Graphical and algebraic specifications of representation models

Logical model	Graphical specification	Algebraic specification
Controller (*C*)		*C*=<*IM*, *OM*, *MR*> *IM:* input message set, *OM:* output message set, *MR:* message processing rule set
Processor (*P*)		*P*=<*IE*, *OE*, *T*> *IE:* input entity set, *OE:* output entity set,

		T: service time
Executor (*E*)		*E*=<*IT*, *OM*, *LR*> *IT:* input task set, *OM:* output message set, *LR:* logistics task execution rule set
Buffer (*B*)		*B*=<*IE*, *OE*, *V*> *IE:* input entity set, *OE:* output entity set, *V:* capacity
Flowing entity (*F*)		*F*=<*GUID*, *Inf*> *GUID:* Global Unique Number, *Inf:* Information set
Virtual Service Node (*VSN*)		$VSN = LPN \times SC \times F$ *LPN:* Logistics Path Network, *SC:* Service Cell, ×: *LPN*, *SC* and *F* are logically connected by *VSN*
Logistics path (*L*)	→ (directed graph)	$L=<O_i, Oj>$ O_i: originating object, O_j: destination object
Logistics Path Network (*LPN*)	— (undirected graph)	*LPN*=< *G*, *E_Set*, *VSN_Set*> *G:* undirected graph, *E_Set:* a set of Executors, *VSN_Set:* a set of *VSN*
Production Service Cell (*PSC*)		*SC*=<*C*, *P*, *E*, *B*, *F*, *L*, *VSN*>
Buffer Service Cell (*BSC*)		

Formally, the Seven-element model is algebraically specified in the following.

$$Seven\ Elements =< C, P, E, B, F, L, VSN > \quad (1)$$

- *C*: Controller, a mapping of various control devices or decision makers, provides decision services for system operations. On the one hand, it converts input message (*IM*) from the upper-level controller, such as an operation, into an execution order, and then sends it to other logical models for execution according to a predefined message processing rule (*MR*). On the other hand, the execution status and result are fed back to the higher-level controller in the form of output messages (*OM*).
- *P*: Processor, a mapping of various processing devices, provides processing services (a specific operation, e.g. milling) for parts. Its state (*P_State*) can be denoted by the current service time (*t*), and the corresponding relationship is

$$P\_State = \begin{cases} 0 & (t = 0) \\ 1 & (0 < t < T) \\ 2 & (t = T) \end{cases}$$

Where *T* is the service time of a certain operation, the three numbers refer to waiting for loading, processing, and waiting for unloading, respectively.

- *E*: Executor, a mapping of various logistics devices, is responsible for parts transfer driven by logistics tasks under the execution rules. A logistics task as input can be defined as $IT = (F{:}\ O_i \rightarrow O_j)$, meaning that the flowing entity (*F*) is transferred from the object O_i to O_j. Logistics task execution rules (*LR*) include manipulator motion control programs, automated guided vehicle (AGV) path planning algorithms, and scheduling mechanisms. The execution results of the logistics tasks are fed back to the controller as output messages (*OM*), including 4 types of tasks namely waiting to be executed, tasks started, tasks executed successfully, and tasks executed failed.

- *B*: Buffer, a mapping of various storage devices, provides temporary or long-term storage services for parts. Its state (*B_State*) can be denoted by the current capacity (v), and the corresponding relationship is

$$B\_State = \begin{cases} 0 & (v=0) \\ 1 & (0<v<V) \\ 2 & (v=V) \end{cases}$$

Where V is the maximum capacity of the buffer, the three numbers refer to prohibiting output, allowing input and output, and prohibiting input, respectively.

- *F*: Flowing entity, a mapping of parts including materials or semi-finished products, receives services from *P*, *E* and *B*. It is uniquely identified by the global unique number (GUID) and can be identified and traced. Also, it acts as a carrier of information, including production process information, production plans, historical track information, and simulation information.
- *L*: Logistics path, a mapping of logistics relationships, presents the direction of the *F* between the two objects (elements or cells).
- *VSN*: Virtual Service Node, a core concept specially proposed in this article, is a high-level abstraction and mapping of production organization relations, logistics logic, and production logic. Its specific connotations will be elaborated in Section 3.2.6.

Further, the relationships identified in the previous section can be formulated as:

$$System\ Structure = < SC, LPN > \tag{2}$$

- *SC:* service cell, a subsystem consisting of seven elements with a specific function. It is divided into production service cell (*PSC*) and buffer service cell (*BSC*).
- *LPN:* logistics path network, the mapping of the actual logistics layout and relationships.

3.2.2 Representation of production relationships

Production activities can be viewed as a series of "services" that are carried out in an orderly manner. Therefore, all production-related activities such as processing, assembly, test, and buffering can be abstracted into services. A set of elements with specific service content are associated into an organic whole, called the service cell. Like the seven elements, *SC* also encapsulates its internal logic and interfaces with its external environment. At the *SC* level, the Executor is called the internal Executor (E_{int}) and the logistics path refers to the directional path between seven elements. The Buffer is subdivided into the input Buffer (B_{in}) and the output Buffer (B_{out}) depending on its relationship with the outside of a cell. Correspondingly, at the system level, the Executor is called the external Executor (E_{ext}) and the logistics path refers to the direction between *SCs*. The reference models and logical models of *SC* are shown in Fig. 3.

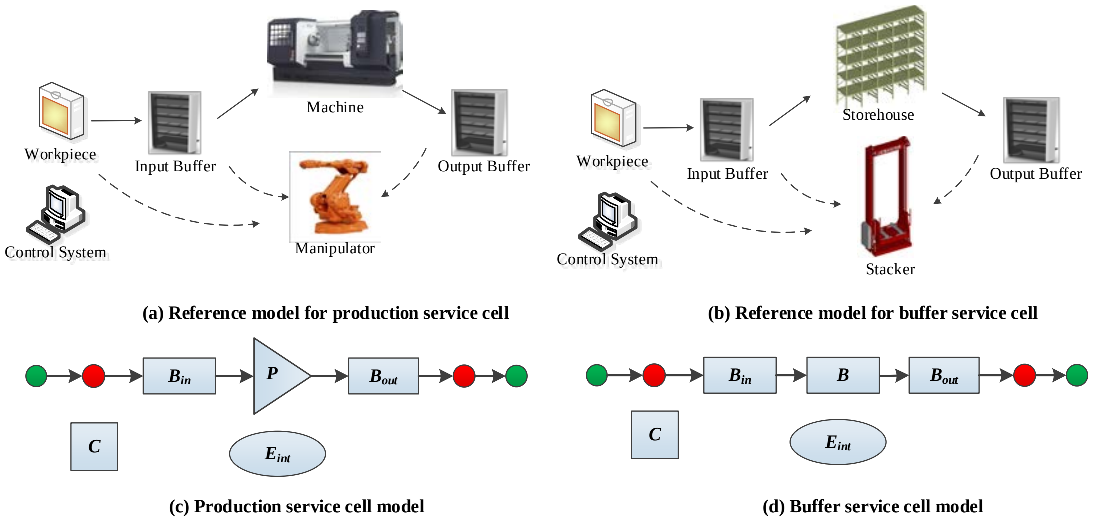


Fig. 3 Typical service cell models

As depicted in Fig. 3(a) (c), in production service cell (*PSC*), the processor is the principal part which provides processing services such as machining, inspection, decomposition or assembly. The *F* enters the *PSC* through the *VSN*, successively passes through the B_{in}, the *P* and the B_{out} receiving services, and finally leaves the *PSC* through the *VSN*. The *C* controls the E_{int} to assist the *F* flow among above elements. The above logic is predefined as a logic template and encapsulated into the *C* of the *SC*.

As illustrated in Fig. 3(b) (d), in buffer service cell (*PBC*), the Buffer is the principal part which provides buffer services such as transfers or storages for the *F*. Besides for the difference of service contents, there are no essential distinct between the *BSC* and the *PSC* in structure and logic.

3.2.3 Configuration and combination of service cell

Based on typical service cell models, different manifestations of service cell can be derived by the configuration of elements. For instance, three basic forms of *PSC* are derived depending on different configurations of the buffer (Fig. 4(a) (b) (c)), and a type of *PSC* is derived from different configurations of E_{int} (Fig. 4(d)). More forms can be derived from the combination of *B*'s and E_{int}'s in different configurations. Likewise, the *BSC* also has multiple different manifestations.

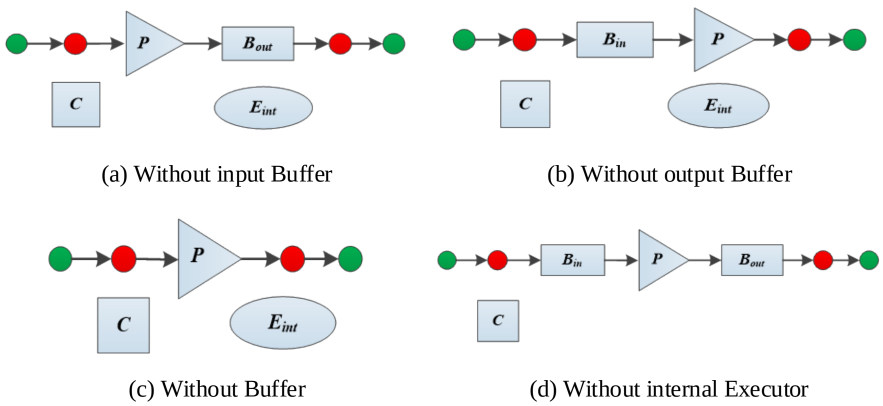


Fig. 4 Different forms of service cell

A complex production organization relationship could be expressed via the combinations of *SCs*. Fig. 5 demonstrates four major types of *SC* combinations: "serial", "parallel", "assembly" and "expansion".

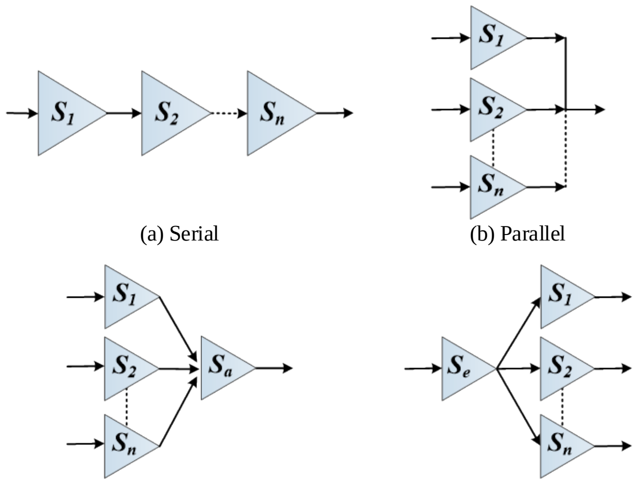


Fig. 5 Four major combinations of service cell models

3.2.4 Properties of the service cell

Properties of the *SCs* can be summarized below:

- *SC is a loose, autonomous agent with an independent function.* It mainly indicates that: (a) there is no distance constraint on the spatial distribution for the elements in the *SC*; (b) the *SC* has no exclusive constraint on its constituent objects, as if an E_{int} could belong to multiple *SC* at the same time; and (c) the *SC* can complete an independent task without external help and only keep contact with the outside via the *VSN*. Hence, the *SC* shows more flexible and reconfigurable features than the traditional functional unit (e.g. functional unit in [33] and unit production process in [34]).
- *SC consists of seven elements or a subset of them and its internal logic is determined by the different configurations of elements.* The service provider (a *P* in *PSC* or a *B* in *BSC*), *C* and *VSN* are the basic elements and indispensable. The B_{in}, B_{out} and E_{int} can be flexibly configured depending on different demands (see Fig. 4). The operating logic of a typical *SC* is predefined as a template. The custom *SC* is derived from the typical *SC*, and its operation logic could be created based on the template, which depends only on the configuration of the custom *SC*. Thus, once the configuration is completed, the logic of the *C* is generated automatically.
- *SC is a middle level of the system. SC* is composed of basic elements, and it in turn acts as a basic element in a higher-level system. A complex system structure can be further simplified by the combination of *SCs* (see Fig. 5). Therefore, the complexity of modeling a system can be reduced by the proposed hierarchical features.

3.2.5 Representation of logistics relationships

In a manufacturing system, there exists various logistics equipment (e.g. AGV, stackers, manipulators etc.), various layout forms (e.g. linear layout, circular layout, and network layout) and various control logics (e.g. automatic tracking and manual control). In order to truly reflect the logistics relationships, the logistics path network (*LPN*) model is put forward, which is defined below.

$$LPN =< G, E\_Set, VSN\_Set > \tag{3}$$

- $G = (V, E', W)$ is a finite, simple and undirected graph with a set of vertices (V), a set of edges (E'), and a weight W_{ij} = Distance(V_i, V_j). It describes the geometric properties of the logistics layout. Let $G(V_i \times V_j)$ denotes the adjacency matrix of G.

$$G(V_i \times V_j) = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, a_{ij} = \begin{cases} 1, \text{ if there is a path } V_i \text{ to } V_j \\ 0, \text{ otherwise} \end{cases} \quad (4)$$

  Where: V_i and V_j are the vertices of the undirected graph; a_{ij} represents the adjacency relationship of the V_i and V_j.
- $E\_Set = \{E_1, E_2 \ldots E_n\}$ is a finite and non-empty set of executors (*Es*), which unifies the various logistics equipment.
- $VSN\_Set = \{VSN_1, VSN_2 \ldots VSN_n\}$ is a finite and non-empty set of *VSNs*, which are control points where the executors and service cell exchange material.

Multiple *LPNs* are interconnected through specific *VSNs* to form an “electronic map” in a virtual environment, which fully express the logistics structure of a complex DMS. And they provide a ground for shortest path planning and logistics scheduling [35].

3.2.6 Representation of operation

Further, how to implement a dynamic operation of a virtual system is discussed. A production logical model is first set forth. Next, the connotations of *VSNs* are highlighted. Finally, an operation based on the production logical model is illustrated.

Production activities could be viewed as a series of alternating operations and logistics activities. Its operation logic could be described as the interactions among parts, operations, and logistics. Therefore, a production logical model of DMS is described in Fig. 6, and it is denoted with a triple:

$$Production\ Logical\ Model = \langle F\_Set, SC\_Set, LPN\_Set \rangle \quad (5)$$

- *F_Set* is a finite, non-empty set of *Fs*, which defines the input and output of a production system.
- *SC_Set* is a finite, non-empty set of *SCs*, which describes the hierarchical production structure.
- *LPN_Set* is a finite, non-empty set of *LPNs*, which describes the network logistics structure.

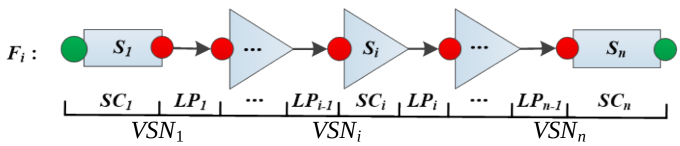


Fig. 6 Production logical model of F_i based on multi-*VSNs*

As illustrated in Fig. 7, the *VSNs* play a significant role in the production logical model. Three ideas central to *VSNs* are:

- *The organizer and manager of the elements in SCs.* The *VSN* associates with a group of objects related to a specific service to form a *SC*. It defines the cell’s composition, service content, and external interfaces. Through the *VSN*, it is possible to traverse any elements of the *SC* and obtain its attributes and status.
- *The link between the SCs and LPNs.* The *VSNs* represent the key geometric points discretely distributed on logistics paths, which associate production and logistics via linking of the *SCs* and *LPNs*. In addition, by traversing status of the *Es* bound on the *LPNs*, appropriate executors can be

selected. By dynamically adding two *VSNs* to *LPNs*, a shortest path between two corresponding *SCs* can be found quickly based on Dijkstra's algorithm [35].

- *The mapping of the part routes*. By mapping production processes associated with *Fs* to a sequence of ordered *VSNs*, a production logic model can be rapidly and dynamically generated. This allows dynamic organization of manufacturing routes during production, rather than a priori linking processes to specific resources in traditional simulation software [36].

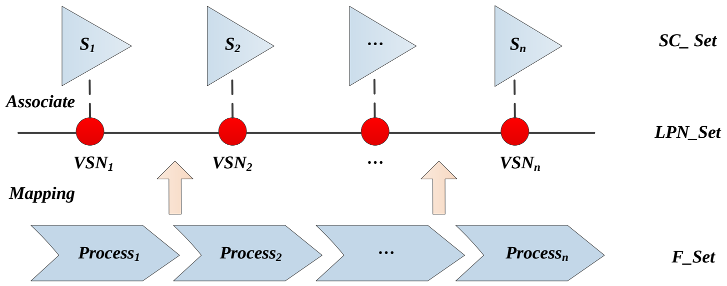


Fig. 7 Relationships among *SCs*, *LPNs* and *Fs*

Depending on its production logical model, the operation processes of a complex DMS are described below.

Input stage. Let F_i be a Flowing entity, and *inf_process* be the process information obtained from it. Assume that the F_i has n processes which are sequentially ordered from 1 to n, and each process corresponds to 1st to nth *SC*. Therefore, *inf_process* can be described as $\{SC_1, SC_2, \cdots, SC_n\}$. Because the *SC* and the *VSN* have a one-to-one correspondence, the *inf_process* can also be described as a sequence of *VSNs* (from VSN_1 to VSN_n).

Execution stage. According to the input, a production logical model of F_i is generated dynamically (see Fig. 6). Initially, F_i is placed at the S_1, waiting to be sent to the next process $S_{i\text{-}1}$ to accept services. There is a demand for logistics services, that is, F_i is required to transfer from S_1 to $S_{i\text{-}1}$. Assume that S_1 and $S_{i\text{-}1}$ are connected through LPN_1, and an executor set E_Set_1 is bound to LPN_1. As a result, an appropriate executor E is selected from E_Set_1 and is assigned to transport F_i from S_1 to $S_{i\text{-}1}$, based on the following two rules. One is that the current state of E is idle. The other is that the total transportation distance (a sum of the distance from E's current position to S_1 and the distance from S_1 to $S_{i\text{-}1}$) is the shortest. Then, by adding the VSN_1 and $VSN_{i\text{-}1}$ to the adjacent matrix of the LPN_1, a shortest logistics path LP_1 would be found based on Dijkstra's algorithm [35]. Therefore, F_i arrives at $S_{i\text{-}1}$, where the service logic is controlled by the *SC* model. After it receives the service in $S_{i\text{-}1}$, a new logistics demand is created. In this way, the materials flow in an orderly manner under the logical model.

Output stage. As shown in Fig. 6, the service process alternates with the logistics process. F_i leaves the systems until accepting all services. The logistics, buffer and processing information generated during the simulation are collected into F_i's simulation information. This simulation information will be applied to verification of production plans and system performance evaluation.

3.3 Connection and interaction design between physical systems and their representation models

The representation models provide high level abstraction and encapsulation for the elements and relationships of a manufacturing system, which are the cornerstones of simulation. In this section, a connection scheme and a data interaction mechanism are designed to achieve the virtual-physical interaction in practice, which enable the simulation applicable throughout the production lifecycle.

3.3.1 Virtual-physical connection scheme

The lack of the virtual-physical connection and interaction leads to a non-real-time simulation, which limits the industrial application of simulation based on DES modeling theory.

Although most simulation software currently provides a 3D simulation environment, they are more concerned about the user experience and ease of modeling than the actual mapping of the physical system. This makes it difficult to consider the actual logistics layout and logistics relationship in traditional DES simulation. For example, in traditional DES simulation, the completion time of each logistics task is usually set to a constant or modelled by a certain statistical distribution, and it is difficult to accurately match to the actual situation with possible disturbance. Therefore, a geometric model that mirrors physical system is needed to realistically reproduce the production scene. On the other hand, in traditional DES simulation, simulation parameters cannot be obtained automatically from physical objects, and can only be derived from statistical values and estimates based on actual production experience. For example, in traditional DES simulation, the operating time of a process on a machine tool is usually assumed to be a fixed constant or set by a certain statistical distribution. In actual production, the operating time of a machine tool is closely related to the specific process. Therefore, a data model is needed to connect the physical object and its virtual model, so that the state and operating parameters of the physical object can be automatically updated to its virtual model.

A connection scheme between physical and representation models is depicted in Fig. 8(a). As shown in Fig. 8(a), a logical model is first set up based on its representation model, which enables virtual models to simulate. Its simulation properties and behavior are described by discrete events and activities based on the DES modeling theory. A geometric model is a mirror image of the physical object in virtual space, which describes the geometric properties (e.g. shape, size, color, assembly relationships, etc.) and kinematic structures (e.g. kinematic relationship of parts, range of movement, degree of freedom, etc.) of the physical object. The former can be expressed by a CAD model, while the latter can be described by a topological model. It also provides a 3D interactive environment for the simulation and monitoring of physical objects. Finally, a data model is responsible for data sensing, processing, transmission, storage, and analysis, enabled by IoT [4] and big data [5]. It is a bridge that connects physical objects and their virtual models. On the one hand, geometric models and logical models can be kept consistent with the physical objects through model reconfiguration and data updates, so that logical models can simulate throughout production lifecycle. On the other hand, with the improvement of the accuracy and real-time performance of simulation input, physical systems could be on-line simulated and continuously optimized. By above-mentioned connection scheme, the physical devices and their representation models are connected, as shown in Table 2.

Fig. 8(a) also shows the relationship between simulation applications and various dimensional models at different production stages, i.e. offline simulation before production, online simulation during production, and predictive simulation after a certain period of stable operation. It will be discussed in Section 3.3.3.

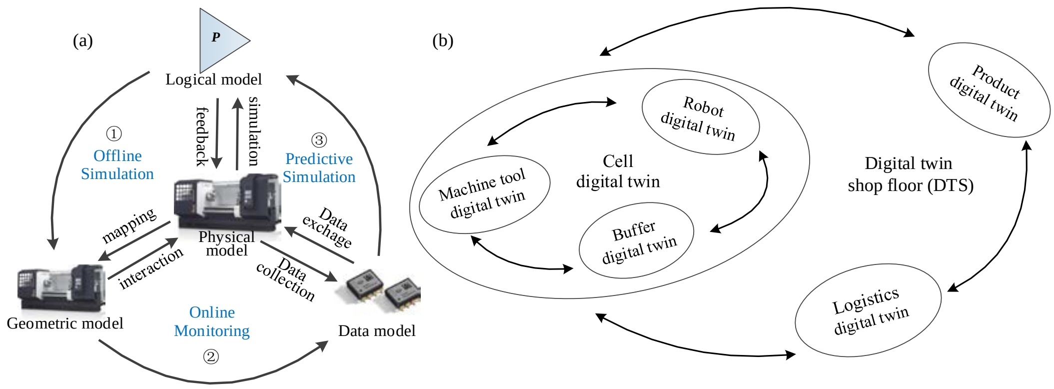


Fig. 8 Connection scheme: (a) digital twin devices; and (b) digital twin shop floor

Table 2 Connection of physical devices and their representation models

Reference models	Connections			Representation models
	Geometry model	Data model	Logical model	
Control equipment	3D CAD models or not	Running status, and production orders and execution results	Message arrival event and message processing activity	Controller (*C*)
Processing equipment	3D CAD models, topological models	Running status, operating time, motion axis real-time position, etc., and production orders and execution results	Part arrival/processing start/processing end/ departure events and operation activity	Processor (*P*)
Logistics equipment	3D CAD models, topological models	Running status, moving speed, carrying capacity, loading and unloading time, etc., and production orders and execution results	Receive / start / complete logistics task event and material handling activity	Executor (*E*)
Storage Equipment	3D CAD models	Capacity	Part arrival / departure events and storage activity	Buffer (*B*)

By Specification (5) and Fig. 7, the logical model of a production system is described by the interactions among *Fs*, *SCs*, and *LPNs*. Thus, a DTS could be modeled by the interaction and integration among products DT, cells DT and logistics DT. According to the connection scheme provided in Table 2, machine tools DT, robots DT and buffers DT can be constructed. These DT models can further build higher-level DT models such as cell DT and logistics DT. Finally, products DT, cells DT and logistics DT interact to form a DTS [3], as shown in Fig. 8(b).

Many scholars have established product DT from a PLM perspective [19,25]. In this research, product DT refers to a fusion model of CAD model, data models and flowing entity. Its data models include production process information, production plans, historical track information, and simulation information. These data are mapped one-to-one with the information set of the flowing entity. The production process and planning information can be extracted from the product data management (PDM) and manufacturing execution system (MES). Above information provides basic data for simulation. Real-time locations of the product can be identified and tracked by RFID or barcode technology to form historical track information, which provides a reference for production plan execution monitoring and real-time scheduling. On the other hand, the flowing entity runs through the entire simulation process and collects simulation information such as processing, storage, and logistics information, which is used for system performance evaluation and production plan verification.

The cell DT is a combination of digital twin devices, including machine tools DT, robots DT and buffers DT. Similarly, it can be described as a fusion of geometric models, data models and logical models. Its geometric model is a three-dimensional layout drawing, including the CAD model and relative position relationship of its constituent devices. Its data model includes the real-time status and operating parameters of each component device, as well as production orders and execution results. And its operation logic is described by a *C* in *SC* model.

The logistics DT is modeled by a logistics layout, a *LPN* model, and data connecting them. As mentioned earlier, *LPN* is defined as an undirected graph *G*, a set of *Es*, and a set of *VSNs*. The *G* is a mirror of a logistics layout, which provides data support for shortest path planning. Based on designed connection scheme and methods, the logistics devices and their corresponding logical models (*Es*) can be connected into a digital twin model. In an actual logistics layout, the two-dimensional code is used to identify where AGVs turn and interact. These two-dimensional codes can correspond one-to-one with *VSNs* in a *LPN*. Through the correspondence between the two-dimensional code and the *VSNs*, logistics devices can be synchronized with *Es* at key points. Furthermore, logistics scheduling methods and operating mechanisms under different production modes (such as push and pull production) are mapped into corresponding rules and simulation logic. Therefore, the *LPN* model can always be synchronized and consistent with the actual logistics situation on the shop floor.

3.3.2 Data interaction mechanism

A data interaction mechanism, which integrates message middleware, memory database and relational database, is designed to implement virtual-physical interaction. The message middleware, e.g. Kafka [11] and MQTT [17], is used to send production instructions and feedback execution results. The memory database, e.g. Redis [12], is used to store real-time operating data during production. And the relational database, e.g. MySQL, Oracle, SQL Server and so on, is used to data management for simulation systems.

Designed data interaction mechanism enables four types of data interaction as indicated by the arrows in Fig. 9: (1) the production orders sent to physical and virtual shop floor for execution via a message middleware, i.e. ex-warehouse/in-warehouse orders of storage units, operation orders of production units, and logistics orders; (2) the results of the orders execution fed back from the physical workshop via a message middleware, i.e. execution failed, execution succeeded, and execution started; (3) the real-time status and operating parameters synchronized from the physical devices and workshop via a memory database, e.g. machine tool motion axis real-time position, speed of AGVs, sensor signals, etc., which makes the virtual and physical workshops to stay synchronized at critical points in time; and (4) the simulation analysis results and verified scheduling fed back from the virtual workshop via a relational database, with real-time data as enablers.

With above data interaction mechanism, workshop can operate based on both practical situation and simulation. On the one hand, the workshop service system sends orders to the physical workshop, then schedules and adjusts according to the results of execution feedback. On the other hand, the physical workshop and the virtual workshop are driven by the same data source and remain consistent at critical moments. Abnormal data and events of the physical workshop will be reflected to the virtual workshop in real time. They are captured, simulated，analyzed, and evaluated to aid decision-making. And new production orders could be generated to adjust and optimize the physical workshop to form a closed-loop.

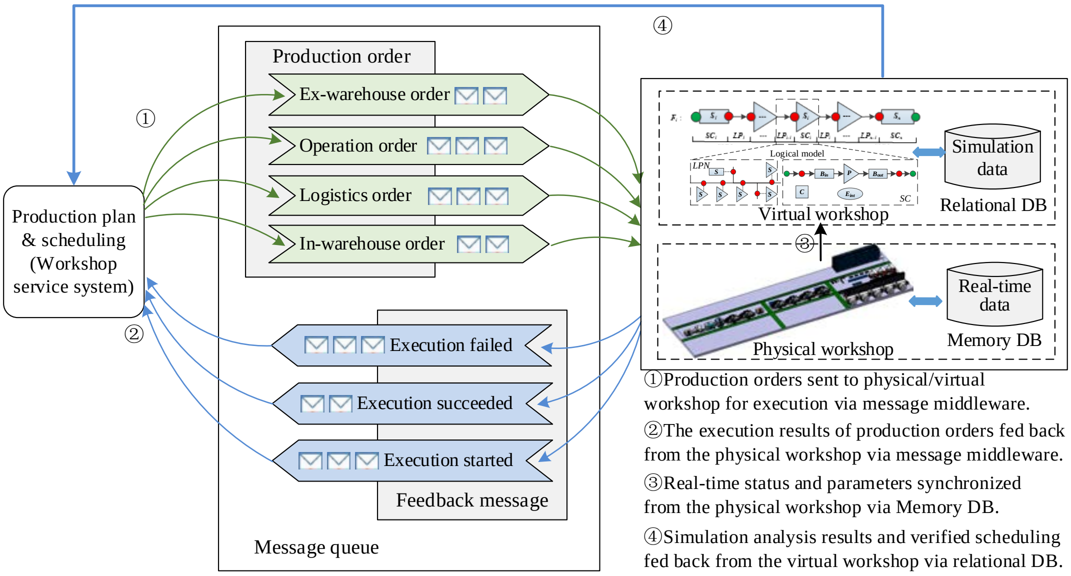


Fig. 9 Data interaction mechanism

3.3.3 Simulation mechanisms throughout the production lifecycle

Based on the designed connection scheme and data interaction mechanism, the virtual model has the potential for full production cycle simulation [37]. A simulation mechanism is described in Fig. 10(a). First, a DMS is represented via the concepts, i.e. Seven-elements, *SC*, *LPN* and their configurations. Next, two parallel lifelines (i.e. physical system and virtual system) exist in physical space and virtual space respectively, which are interweaved via the geometric, logical and data models. On the one hand, the model-driven offline simulation and data-driven online monitoring run in parallel. On the other hand, when the physical system is disturbed, the current statuses of the system as simulation inputs invoke the simulation model to evaluate the impact of this disturbance on production. Finally, a predictive simulation can be achieved by the fusion of real-time data and simulated data. On that basis, the scenarios of simulation applications along different production phases are described in Fig. 10(b).

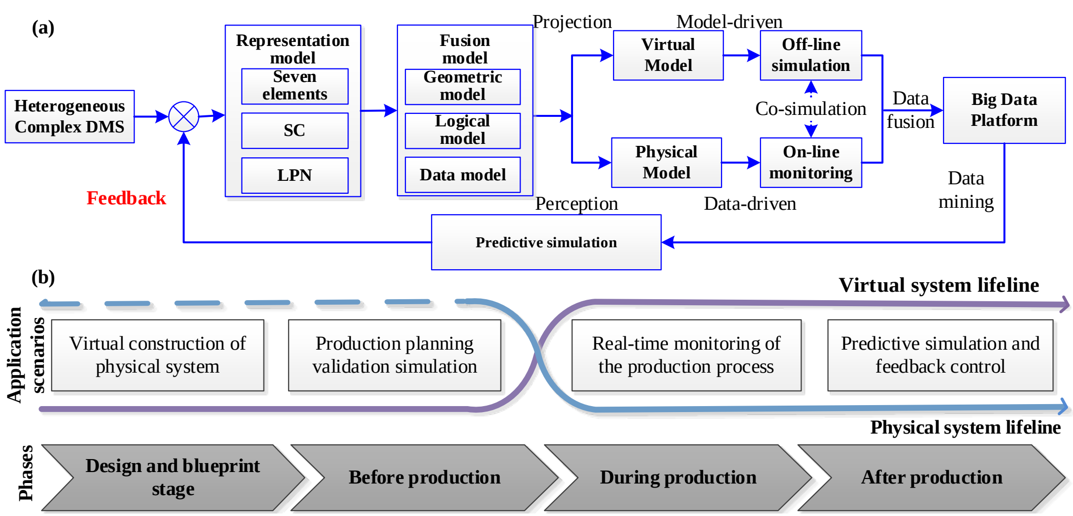


Fig. 10 (a) Simulation mechanism; and (b) applications scenarios along different phases

In the design and blueprint stage, the lifeline of the physical workshop has not yet begun (see the blue dotted line in Fig. 10(b)); various design schemes can be generated quickly in virtual space through the configuration and combination of the Seven-elements, *SC*, and *LPN*. By comparing and analyzing these schemes, the design parameters can be optimized, e.g. the number and capacity of the buffer stations, the number of logistics equipment, logistics path settings, etc.

Before production, the production plans from a schedule system are sent to the virtual shop floor where various working conditions are simulated to find potential conflicts in the plans. After verification and revision, the plans are dispatched to the physical workshop for execution.

During production, the physical shop floor and the virtual shop floor run in parallel. The real-time status of resources is synchronously reflected to virtual shop floor; therefore, the production process could be monitored, and the real-time data could be recorded. On the other hand, the consequences of different scheduling schemes under various disturbances (e.g. equipment failures, plan changes and material shortages) could be assessed via online simulation, which is the decision-making foundation for dynamical scheduling.

After the system runs stably for a period, massive operational data and simulation data are generated, which provide a data foundation for subsequent data mining, knowledge engineering, and model iterative optimization. Production scenarios, especially the scene before the failure, can be reproduced driven by historical data to gain valuable experience and find potential problems. What is more, based on massive historical data, predictive simulation becomes possible; and the possible state of device/workshop at the next moment would be acquired, which is of great significance for improving the accuracy of decision-making and the effectiveness of control.

4 Application case for verification

The proposed DT implementation method has been applied to a real-world digital twin shop-floor mainly producing various types of complex aerospace structural parts. In order to improve production efficiency and shorten time-to-market, the transformation and upgrades toward DTS were carried out, including workshop automation and information construction, virtual workshop construction, production plan simulation and verification, and production process real-time monitoring and scheduling.

4.1 Identification and representation of the workshop

As depicted in Fig. 11, this shop floor consists of six functional areas that are distributed on both sides of the logistics route. Further, these functional areas are classified into 23 units, and units with similar function and structure are in the same functional area. For instance, the warehouse area (A_1) is divided into two units, where each unit consists of a storage, a stacker, and an input and output buffer station, and operates with the same logic.

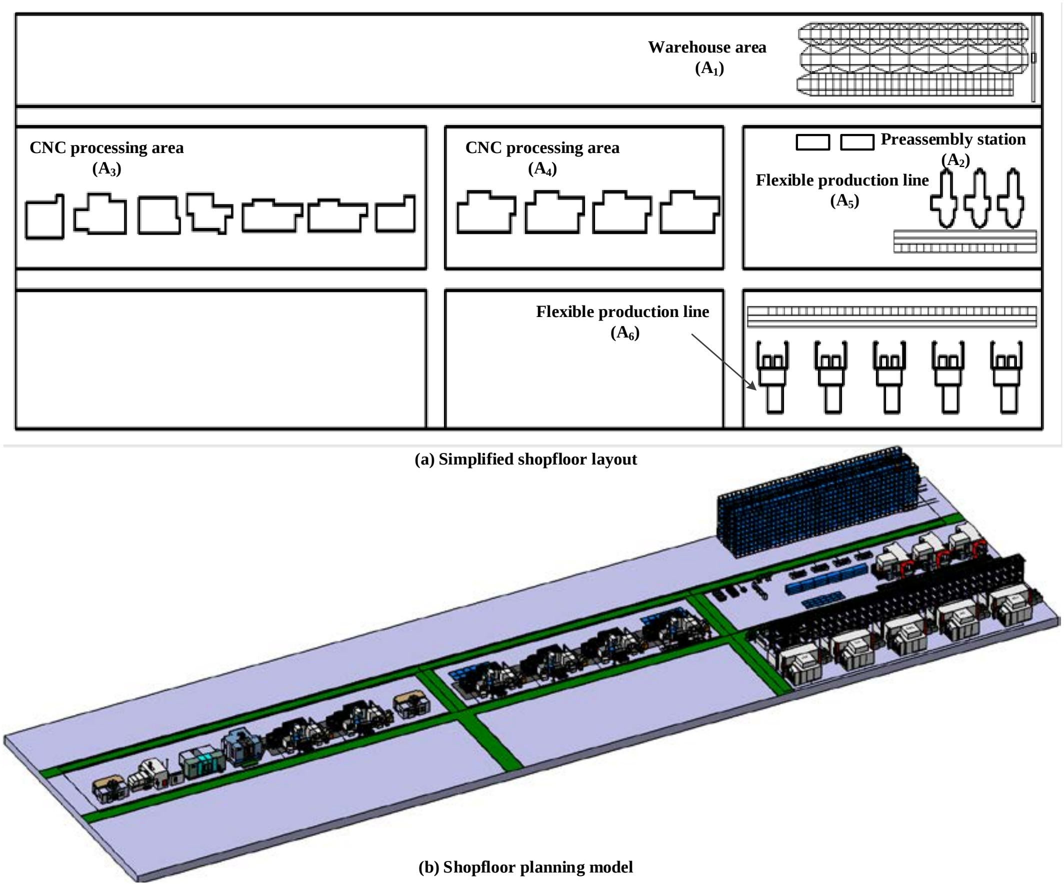


Fig. 11 A structural parts processing workshop

Through a comprehensive analysis of this workshop, its specific composition and hierarchical structure are provided in Table 3 and the logistics structure is shown in Table 4.

Table 3 The production structure and major equipment

Function area	Function and structure	Main equipment (numbers)
Warehouse area (A_1)	Storing workpieces and product, including 2 buffer units	Storehouse (2), stacker (2), input and output buffer (2)
Preassembly station (A_2)	Preassembly and disassembly, including 2 preassembly units	Manual preassembly station (2), manipulator (2), worker (2)
CNC processing area (A_3)	Processing, including 7 processing units	Machine center with auto transmitting mechanism (7)
CNC processing area (A_4)	Processing, including 4 processing units	Machine center with auto transmitting mechanism (4)
Flexible production line (A_5)	Processing, including 3 processing units	NC machine (3), manipulator with guide (1), transfer station (1)
Flexible production line (A_6)	Processing, including 5 processing units	NC machine (5), manipulator with guide (1), transfer station (1)

Table 4 The logistics structure

Logistics paths	Logistics equipment (numbers)	Tasks

AGV routes(P#1)	AGV (1)	Material handling between A_1 and A_2
AGV routes (P#2)	AGV (3)	Material handling among A_2, A_3, A_4, A_5, and A_6
Guideway(P#3)	manipulator with guide (1)	Material handling among 3 processing units in A_5
Guideway (P#4)	manipulator with guide (1)	Material handling among 5 processing units in A_6

By Specification (5), the production system's logical model is described by the interactions among *Fs*, *SCs*, and *LPNs*. Thus, in the production model of this workshop depicted in Fig.11, the three combined parts are specified as follows:

$$SC\_Set = \{SC_i \mid 1 < i < 23\}$$

$$LPN\_Set = \{LPN_i \mid 1 < i < 4\}$$

$$F\_Set = \{F_i \mid 0 < i < 4\}$$

Where, 23 service cells (*SCs*) include 2 buffer service cells (*BSCs*) and 21 production service cells (*PSCs*); their algebraic representations are outlined in Table 5; and hierarchical production relationships are illustrated in Fig. 12. Note that *SCs* with same composition and structure are grouped into one class, therefore only one *SC* is described for each class. Also note that the logistics path (*L*) and the *VSN* are the abstract models in a *SC*, thus there are no physical equipment corresponding to them. In addition, the serial numbers of these *SCs* are the same as that of the *VSNs* (see Fig. 13 for the serial numbers).

Four logistics path networks (*LPNs*) are two AGV routes (P#1, P#2) and two guideways (P#3, P#4) (see Fig. 13). Taking the AGV routes (P#2) as the example, it has eight vertices and three AGVs running on it. Meanwhile, 14 *VSNs* are defined on it; 11 of which are interaction points where the AGVs and *SCs* exchange material; and 3 of which (VSN_3, VSN_{22}, VSN_{23}) are connection points which respectively connect P#2 with P#1, P#4, P#3. From Specifications (3) and (4), the algebraic representations of the LPN_2 based on P#2 is:

$$LPN_2 = < G_2, E_{ext}\_Set_2, VSN\_Set_2 >$$

Where

$$G_2(V_8 \times V_8) = \begin{pmatrix} a_{11} & \cdots & a_{18} \\ \vdots & \ddots & \vdots \\ a_{81} & \cdots & a_{88} \end{pmatrix}, \quad a_{ij} = \begin{cases} 1, \text{ if there is a path } V_i \text{ to } V_j \\ 0, \text{ otherwise} \end{cases}$$

$$E_{ext}\_Set_2 = \{E_{ext}^i \mid 1 \le i \le 3\}$$
$$mapping: \ \mathrm{AGV}_i \to E_{ext}^i$$

$$VSN\_Set_2 = \{\{VSN_3\}, \{VSN_i \mid 11 \le i \le 23\}\}$$

Table 5 Algebraic representations of *SCs* in Fig. 11

Function Area	Service Cell	Algebraic representation

A_1	SC_1, SC_2	$SC_1 = \{C, B, B_{in}, B_{out}, E_{int}, F, L, VSN_1\}$ *Mapping:* storehouse control system → C storehouse → B input and output buffer → B_{in}, B_{out} stacker → E_{int} workpieces → F
A_2	SC_5, SC_6	$SC_5 = \{C, P, E_{int}, F, L, VSN_5\}$ *Mapping:* worker → C, P manipulator → E_{int} workpieces → F
A_3, A_4	SC_{11}-SC_{17}, SC_{18}-SC_{21}	$SC_{11} = \{C, B_{in}, B_{out}, P, E_{int}, F, L, VSN_{11}\}$ *Mapping:* machine center control system → C machine center → B_{in}, B_{out}, P, E_{int} workpieces → F
A_5, A_6	SC_8-SC_{10}, SC_{25}-SC_{29}	$SC_8 = \{C, B_{in}, B_{out}, P, E_{int}, F, L, VSN_8\}$ *Mapping:* production line control system → C transfer station → B_{in}, B_{out} NC machine → P manipulator with guide → E_{int} workpieces → F

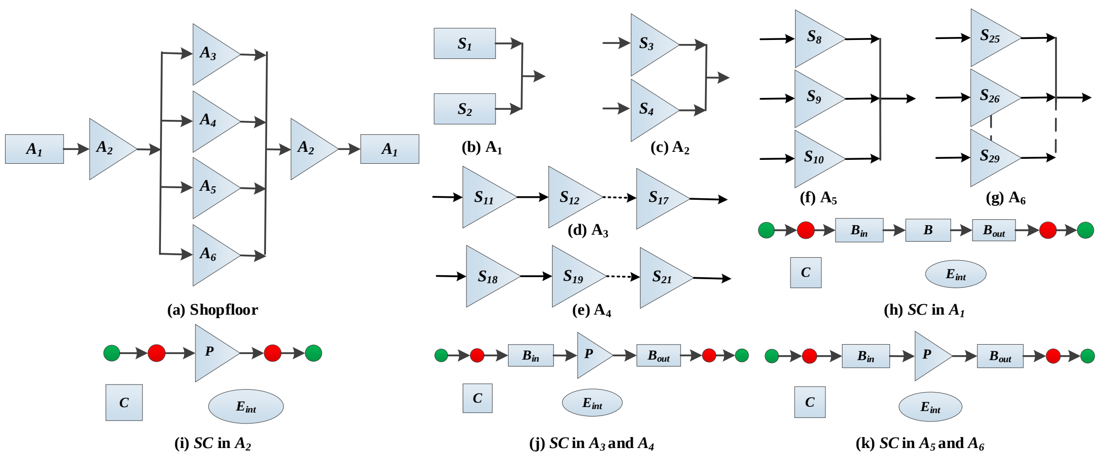


Fig. 12 Representations of production relationship of workshop in Fig. 11

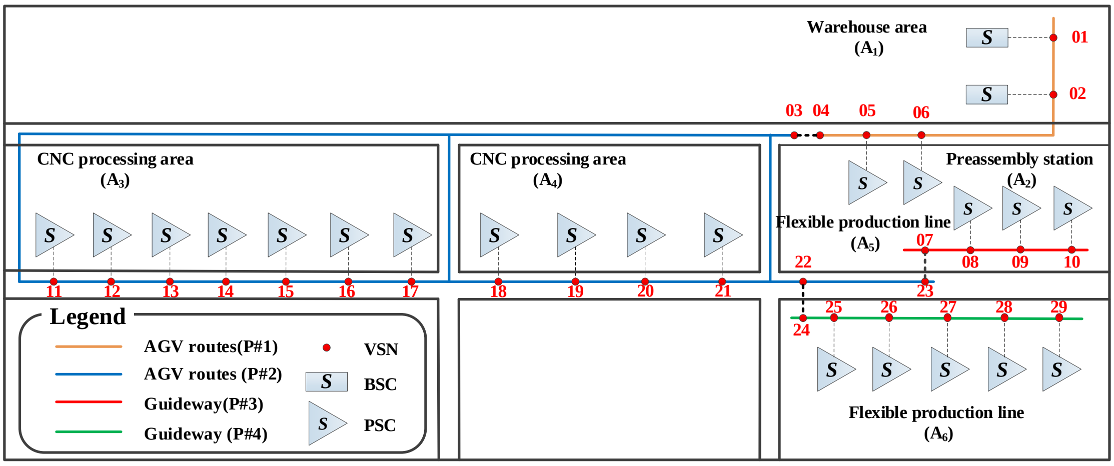


Fig. 13 Representations of logistics relationships of workshop in Fig. 11

4.2 Virtual model construction system and its applications

A virtual model construction system is developed to support virtual workshop construction and simulation applications. It consists of four main modules, i.e. virtual construction of elements, virtual construction of (sub) systems, production plan simulation and verification, and production processes real-time monitoring and scheduling.

4.2.1 Virtual construction of elements

Based on the connection scheme mentioned earlier (Fig. 8(a)), virtual models of physical equipment (e.g. processing equipment, logistics equipment, and storage equipment) are constructed by fusion of their geometric models, logical models, and data models. Element virtual construction modules provide templates and interactive interfaces to set up above models. As a result, the virtual components were encapsulated and stored. For instance, machine tool was encapsulated as a Processor (*P*), and stored in an element library.

Fig. 14 shows how to construct a virtual machine tool and establish a data connection with the physical machine tool. Firstly, based on multi-body kinematics, the kinematic relationship of a machine tool is defined by a topological chain which is composed of a series of topology nodes and connecting arrows; the former describes the key components of a machine tool, and the latter describes the parent-child relationship between these components. And, a 3D CAD model of each part of a machine tool is imported and linked to the corresponding topological nodes. Secondly, the functions of a machine tool, e.g. clamping, door opening and closing, and operating, are defined by associating CAD models and their motion parameters. And the logical relationships are modeled by setting the sequence and function trigger conditions. For instance, the function of “door opening” was defined by entering the CAD model name, the speed and stroke of the door. And its triggering condition was set to the start of a new operation or the end of a previous operation. The logical sequence between these functions of a machine tool was defined as the door opening, clamping, door closing, and then operating. Thirdly, the real-time running status and parameters of the machine tool were collected by OPC UA. A total of 17 items of data including feed speed, feed rate, spindle power and speed, real-time positions of the five motion axes and so on, were collected, which were synchronized to geometric and logical models via Kafka and Redis.

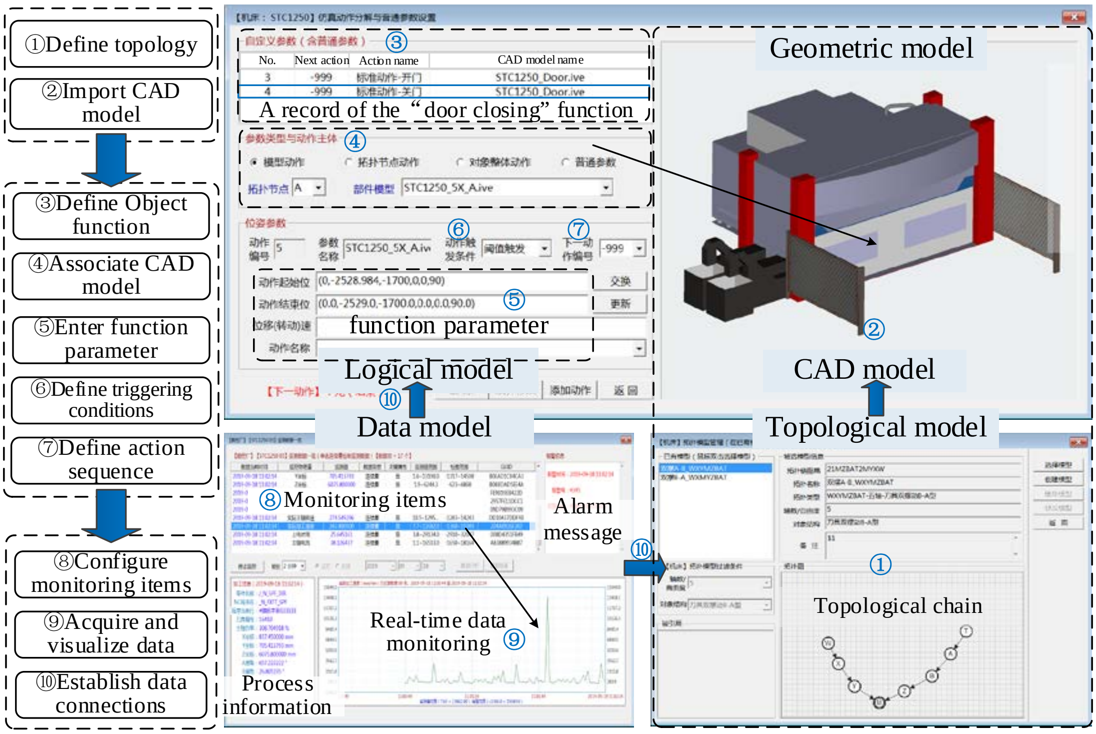


Fig. 14 Process of a virtual machine tool modeling

4.2.2 Virtual construction of complex systems

The system virtual construction modules provide the functions of 2D layout, logistics path network configuration, and service cells configuration. On the one hand, it can quickly build a virtual model of an existing workshop and truly reflect its equipment layout and logistics relationships. When the physical workshop layout or organizational relationship is adjusted, the virtual model can also be quickly updated and adjusted through logistics path modeling and service cell reconfiguration. On the other hand, it can also be used to simulate and evaluate the facilities and logistics layout of a workshop in the blueprint.

Fig. 15 illustrates how to construct a virtual model of an existing workshop. The workshop 2D layout was first created via dragging and dropping models from the element library built in the previous section. Through the human-computer interaction interface, the logistics layout of the actual workshop was mirrored into a virtual space. Logistics equipment, i.e. AGVs, stackers, and manipulators with guide, were bound to the corresponding logistics paths. And logistics control points (see *VSNs* in Fig. 15) were defined on the logistics paths, which associate logistics paths with service cells. As a result, the logistics path networks described in Fig. 13 was modeled, which would support shortest path planning and logistics scheduling. As shown in Fig. 15, one can configure the input buffer (B_{in}), output buffer (B_{out}) and Executor (E) of a service cell. Different configuration schemes will generate different service cell models, which determine the simulation logics of the service cells. In this way, different production service cells and buffer service cells represented in Table 5, were set up by configuring its Seven-element models.

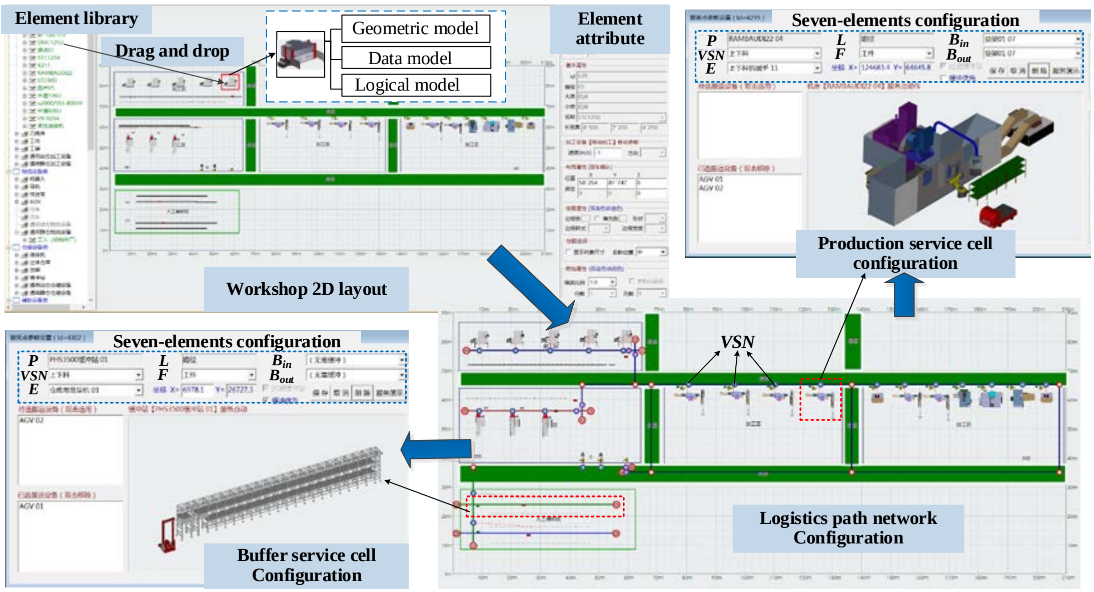


Fig. 15 Virtual workshop modeling

4.2.3 Production plan simulation and verification

A production plan is generated based on scheduling models under resource constraints and optimization goals. In the production scheduling, the actual facilities and logistics layout of the workshop as well as the buffer station capacity, logistics transportation time, auxiliary operation time, etc. are difficult to consider or less considered. Such scheduling plan is rough. The simulation overcomes above shortcomings and provides more details for the scheduling plan.

Fig. 16 illustrates the process of production plan simulation and verification. Taking production tasks, process information and production plan as input, the logical model for each part is generated, which drives the virtual shop floor established in Section 4.2.2. The simulation data (e.g. the start time and end time of each process and logistics tasks) is collected and plotted as a production execution Gantt chart, which is compared with the planned Gantt chart drawn by the production planning. Based on the simulation results, the production plan is adjusted and optimized. And the adjusted plan can be verified by re-simulation.

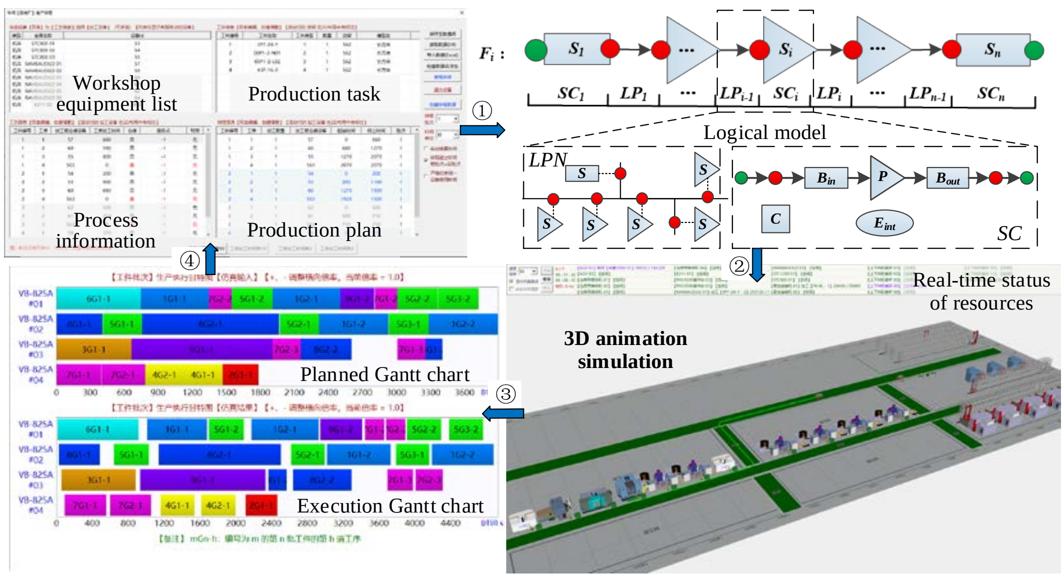


Fig. 16 Process of production plan simulation and verification

4.2.4 Production process real-time monitoring and scheduling

The physical devices and their virtual models were connected via the data model in Section 4.2.1; a virtual workshop was established in Section 4.2.2; and a virtual workshop-based production plan simulation and verification was implemented in Section 4.2.3. On this basis, this section demonstrates the implementation of production process real-time monitoring and scheduling.

As shown in Fig 17(c), a verified plan is decomposed into operation and logistics orders, which are sent to the physical workshop and devices for execution through Kafka. During execution, the execution results are feedback to production plan and scheduling system via Kafka. And the operating parameters and real-time status of the physical devices are obtained from Redis, driving the virtual devices and shop floor to run synchronously (Fig. 17(a) and (b)). As shown in Fig. 17(b), these data are monitored and analyzed, e.g. real-time status analysis and overall equipment effectiveness (OEE) analysis. On the other hand, if the execution deviates from the planned, the simulation modules would be triggered and invoked to evaluate and analyze possible effects under disturbance; and the simulation results aid scheduling decisions. The new scheduling scheme could be verified again by the simulation modules. The verified scheduling plan is fed back to physical shop floor for execution. Therefore, the physical shop floor is continually simulated and optimized via virtual-physical interaction.

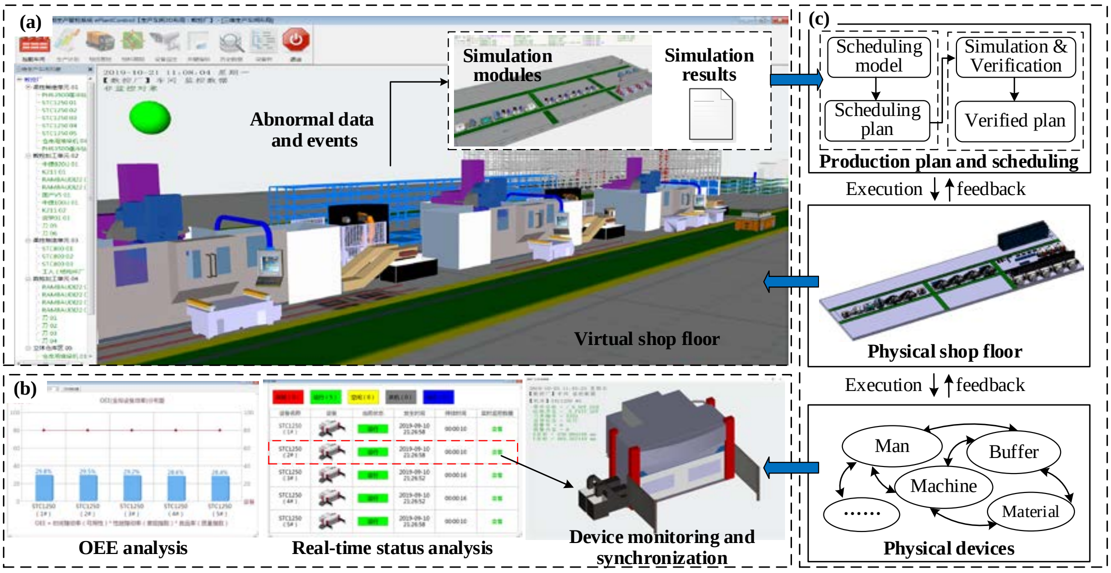


Fig. 17 Production process real-time monitoring and scheduling. (a) Process visualization and simulation based on virtual shop floor; (b) device status monitoring and analysis; and (c) physical shop floor operation scenario.

5 Discussions and conclusions

This article focuses on the virtual modeling method at a conceptual level and the implementation mechanisms of the virtual-physical connections in practice for establishing digital twin shop floor, based on the discrete event system (DES) modeling theory. At the conceptual level, DES modeling theory provides a unified formal and graphical representation of composition, structure, and operation of a manufacturing system. As a result, heterogeneous discrete manufacturing systems could be rapidly modeled and configured in a simple, well-defined, and hierarchical way. In practice, the virtual-physical connection scheme and data interaction mechanism expand the application scope of DES simulation, giving it the potential for digital twin applications. The virtual-physical connection scheme enables virtual devices modeling through integrating geometric models, data models, and logical models. By dragging and dropping predefined virtual devices, personalized workshop and logistics layout could be quickly configured, which could support designing of workshop and production plan verification. In addition, the data interaction mechanism enables the interaction and coordination among virtual workshop, physical workshop, and workshop service system to realize process visualization, real-time monitoring, and scheduling.

It is worth mentioning that there are several limitations of the current work. This article focuses on establishing a generic logical model or simulation model for heterogeneous manufacturing systems, rather than a data model or an information model; the latter could be modeled using AutomationML [23], MTConnct and OPC UA [22]. Integrating existing logical and information modeling specifications could more fully express a digital twin model. Besides, this article does not cover the formal model verification, validation, and accreditation (VV&A) [38]. This article also does not discuss data communication technology issues related to real-time, reliability, and security. It is believed these problems would be better solved with the development of 5G technology and next-generation communication technology. What is more, although the proposed method has been successfully applied in a real-world workshop, it is still at the concept level. Comprehensive evaluation via more applications need to be explored, e.g. predictive simulation, production performance online optimization and feedback control, etc., with big data [5] and machine learning [39] as enablers.

Our conclusions are: (1) A series of graphical and algebraic modeling specifications are determined, which could realize the unified representation of composition, structure, and operation of heterogeneous discrete manufacturing systems. (2) The designed connection scheme and data interaction mechanism enable the connection and interaction between the physical system and its virtual model, which enables digital twin application. (3) An application case study demonstrates the feasibility and effectiveness of the proposed methods. And a virtual model construction system is developed based on the proposed method, which could be used for virtual devices and shop floor modeling, manufacturing system design and optimization, plan simulation and verification, process monitoring and scheduling, and system redesign/reconfiguration.

Future works could be devoted to (1) unified modeling standard of digital twin by integrating existing logical and information modeling specifications and VV&A, and (2) predictive simulation and optimization control based on data fusion.

Acknowledgement

This work is financially supported by the Major State Basic Research Program in Sichuan Province of China under Grant 20YYJC4377. Authors would like to thank the anonymous reviewers' constructive comments for improving this publication.

References

[1] Zhang H, Liu Q, Chen X, Zhang D, Leng JW. A digital twin-based approach for designing and multi-objective optimization of hollow glass production line. IEEE Access 2017;(5): 6901-11. http://doi.org/10.1109/ACCESS.2017.2766453.

[2] Zhuang C, Liu J, Xiong H. Digital twin-based smart production management and control

framework for the complex product assembly shopfloor. Int J Adv Manuf Technol 2018;96:1149-63. https://doi.org/10.1007/s00170-018-1617-6.
[3] Tao F, Zhang M. Digital twin shop-floor: a new shop-floor paradigm towards smart manufacturing. IEEE Access 2017;5:20418-27. https://doi.org/10.1109/ACCESS.2017.2756069.
[4] Cecil J, Albuhamood S, Cecil-xavier A, Ramanathan P. An advanced cyber physical framework for micro devices assembly 2019;49:92–106. http://doi.org/10.1109/TSMC.2017.2733542.
[5] Tao F, Cheng J, Qi Q, Zhang M, Zhang H, Sui F. Digital twin-driven product design, manufacturing and service with big data. Int J Adv Manuf Technol 2018:3563-76, http://doi.org/10.1007/s00170-017-0233-1.
[6] Choi BK, Kang D. Modeling and simulation of discrete-event systems. John Wiley & Sons; 2013. http://dx.doi.org/10.1002/9781118732793.
[7] Kim H, Choi BK. Colored ACD and its application. Simul Model Pract Theory 2016;63:133–48. http://dx.doi.org/10.1016/j.simpat.2016.02.009.
[8] Ryan J, Heavey C. Process modeling for simulation. Comput Ind 2006;57:437–50. http://dx.doi.org/10.1016/j.compind.2006.02.002.
[9] Buyurgan N, Saygin C. Application of the analytical hierarchy process for real-time scheduling and part routing in advanced manufacturing systems. J Manuf Syst 2008; 27:101–10. https://doi.org/10.1016/j.jmsy.2008.08.002.
[10] Lu Y, Liu C, Wang KIK, Huang H, Xu X. Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. Robot Comput Integr Manuf 2020;61. https://doi.org/10.1016/j.rcim.2019.101837.
[11] Apache Software Foundation. Kafka. 2017 (accessed 12 Mar 2020). https://kafka.apache.org/
[12] Redis Labs. Redis. 2015 (accessed 12 Mar 2020). https://redis.io/.
[13] Tao F, Zhang M, Liu Y, Nee AYC. Digital twin driven prognostics and health management of complex equipment. CIRP Ann-Manuf Technol 2018;67(1):169-72. https://doi.org/10.1016/j.cirp.2018.04.055.
[14] Grieves MW. Product lifecycle management: the new paradigm for enterprises, Int J Prod Dev 2005;2 (1-2): 71–84. https://doi.org/10.1504/IJPD.2005.006669.
[15] Dai S, Zhao G, Yu Y, Wang W. Trend of digital product definition: from mock-up to twin. J Comput Aided Des Comput Graph 2018;30(8):1554-62. https://doi.org/10.3724/SP.J.1089.2018.16831.
[16] Zheng Y, Yang S, Cheng H. An application framework of digital twin and its case study. J Ambient Intell Humaniz Comput 2019;10:1141–53. https://doi.org/10.1007/s12652-018-0911-3.
[17] Liu C, Jiang P, Jiang W. Web-based digital twin modeling and remote control of cyber-physical production systems. Robot Comput Integr Manuf 2020;64:101956. https://doi.org/10.1016/j.rcim.2020.101956.
[18] Tao F, Liu W, Zhang M, Hu T, Qi Q, Zhang H, et al. Five-dimension digital twin model and its ten applications. Comput Integr Manuf Syst 2019; 25(1): 1-18. http://www.cims-journal.cn/CN/10.13196/j.cims.2019.01.001.
[19] Bao J, Guo D, Li J, Zhang J. The modeling and operations for the digital twin in the context of manufacturing. Enterp Inf Syst 2019;13:534–56. https://doi.org/10.1080/17517575.2018.1526324.
[20] Bradac Z, Marcon P, Zezulka F, Arm J, Benesl T. Digital Twin and AAS in the Industry 4.0

Framework. IOP Conf Ser Mater Sci Eng 2019;618:0–8. https://doi.org/10.1088/1757-899X/618/1/012001.
[21] Seif A, Toro C, Akhtar H. Implementing industry 4.0 asset administrative shells in mini factories. Procedia Comput Sci 2019;159:495–504. https://doi.org/10.1016/j.procs.2019.09.204.
[22] Liu C, Vengayil H, Zhong RY, Xu X. A systematic development method for cyber-physical machine tools. J Manuf Syst 2018;48:13–24. https://doi.org/10.1016/j.jmsy.2018.02.001.
[23] Schroeder GN, Steinmetz C, Pereira CE. Digital twin data modeling with AutomationML and a communication methodology for data exchange. IFAC-PapersOnLine 2016; 49:12–7. http://doi.org/10.1016/j.ifacol.2016.11.115.
[24] Fang Y, Peng C, Lou P, Zhou Z, Hu J, Yan J. Digital-twin-based job shop scheduling toward smart manufacturing. IEEE Trans Ind Informatics 2019;15:6425–35. https://doi.org/10.1109/TII.2019.2938572.
[25] Zhuang C, Liu J, Xiong H, Ding X, Liu S, Weng G. Connotation, architecture and trends of product digital twin. Comput Integr Manuf Syst 2017;23:753–68. http://www.cims-journal.cn/CN/10.13196/j.cims.2017.04.010.
[26] Tao F, Liu W, Liu J, Liu X, Liu Q, Qu T, et al. Digital twin and its potential application exploration. Comput Integr Manuf Syst 2018;24(1):1-18 http://www.cims-journal.cn/CN/10.13196/j.cims.2018.01.001.
[27] Siemens AG. Digitalization in industry: Twins with potential. 2018 (accessed 28 Oct 2019). https://www.siemens.com/customer-magazine/en/home/industry/digitalization-in-machine-building/the-digital-twin.html.
[28] Liu Q, Zhang H, Leng J, Chen X. Digital twin-driven rapid individualised designing of automated flow-shop manufacturing system. Int J Prod Res 2019;57:3903–19. https://doi.org/10.1080/00207543.2018.147124.
[29] Shafiq SI, Sanin C, Szczerbicki E, Toro C. Virtual engineering object/virtual engineering process: A specialized form of cyber physical system for industrie 4.0. Procedia Comput Sci 2015;60:1146–55. https://doi.org/10.1016/j.procs.2015.08.166.
[30] Ding K, Chan FTS, Zhang X, Zhou G, Zhang F, Ding K, et al. Defining a digital twin-based cyber- physical production system for autonomous manufacturing in smart shop floors. Int J Prod Res 2019;57:6315–34. https://doi.org/10.1080/00207543.2019.1566661.
[31] Liu C, Vengayil H, Lu Y, Xu X. A cyber-physical machine tools platform using OPC UA and MTConnect. J Manuf Syst 2019; 51: 61-74. http://doi.org/10.1016/j.jmsy.2019.04.006.
[32] Cochran DS, Arinez JF, Duda JW, Linck J. A decomposition approach for manufacturing system design. J Manuf Syst 2002;20:371–89. https://doi.org/10.1016/S0278-6125(01)80058-3.
[33] Cicirelli F, Furfaro A, Nigro L. Modelling and simulation of complex manufacturing systems using statechart-based actors. Simul Model Pract Theory 2011;19:685–703. http://doi.org/10.1016/j.simpat.2010.10.010
[34] Huang SH, Dismukes JP, Shi J, Su Q, Wang G, Razzak MA et al. Manufacturing system modeling for productivity improvement. J Manuf Syst 2002;21:249-59. https://doi.org/10.1016/S0278-6125(02)80165-0.
[35] Ding GF et al. Logistics modeling and simulation method for complex discrete manufacturing system based on logistics path network. China Patent No. ZL 201811187173.3.
[36] Ruiz N, Giret A, Botti V, Feria V. An intelligent simulation environment for manufacturing systems. Comput Ind Eng 2014;76:148–68. http://dx.doi.org/10.1016/j.cie.2014.06.013.

[37] Boschert S, Rosen R. Digital twin—the simulation aspect. In: Hehenberger P, Bradley D, editors. Mechatronic Futures. Springer, Cham; 2016, p.59-74. http://doi.org/10.1007/978-3-319-32156-1_5.
[38] Tao F, Zhang H, Liu A, Nee AYC. Digital Twin in Industry: State-of-the-Art. IEEE Trans Ind Informatics 2019;15:2405–15. https://doi.org/10.1109/TII.2018.2873186.
[39] Min Q, Lu Y, Liu Z, Su C, Wang B. Machine learning based digital twin framework for production optimization in petrochemical industry. Int J Inf Manage 2019;49: 502–19. https://doi.org/10.1016/j.ijinfomgt.2019.05.020.