Int. J. Production Economics 240 (2021) 108223

Contents lists available at ScienceDirect

International Journal of Production Economics

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

Blockchain-based digital twin sharing platform for reconfigurable socialized manufacturing resource integration

Ming Li[a,b], Zhi Li[c,d,*], Xidian Huang[e], Ting Qu[a,b]

[a] School of Intelligent Systems Science and Engineering, Jinan University (Zhuhai Campus), Zhuhai, Guangdong, PR China
[b] Institute of Physical Internet, Jinan University (Zhuhai Campus), Zhuhai, Guangdong, PR China
[c] Guangdong Provincial Key Laboratory of Computer Integrated Manufacturing Systems, School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, Guangdong, PR China
[d] Department of Mechanical and Energy Engineering, Southern University of Science and Technology, Shenzhen, Guangdong, PR China
[e] Macau Institute of Systems Engineering, Macau University of Science and Technology, Macau SAR, PR China

ARTICLE INFO

Keywords:
Digital twin
Social manufacturing
Blockchain
Knowledge sharing

ABSTRACT

The emergence of social manufacturing has facilitated the evolution of resource organization in the manufacturing industry, with a trend towards a flat management structure. The mass absorption of socialized idle resources that lack a digitalized and intelligent basis has influenced the efficiency of collaborative production. The digital twin has been widely applied to promote cyber collaboration through the multidimensional digitalization of physical resources. As a knowledge-intensive software entity, it is impractical for resource owners, especially individuals and small and micro businesses, to implement digital twins. To make sufficient use of the sharing convenience in the social manufacturing community (SMC), this paper proposes a blockchain-based digital twin sharing platform to enable software copyright protection and simplify heterogeneous manufacturing resource integration in decentralized and distributed environments. Domain-driven design is adopted as the major methodology to recognize the problem domain and partition the business subdomains. Blockchain technology is introduced to construct a novel sharing schema to preserve and retrieve the copyrights of digital twins under a distributed and decentralized SMC network. Furthermore, to enhance the reusability of digital twins, a reconfigurable digital twin architecture is proposed to enable UPnP implementation of shared digital twins under heterogeneous deployment environments. Finally, a demonstrative case study motivated by a real-life 3D printing scenario is conducted to verify and evaluate the proposed platform.

1. Introduction

The concept of social manufacturing has emerged with the transformation and upgrading of the manufacturing industry in recent years. Social manufacturing is regarded as a novel kind of networked manufacturing paradigm that leverages the concept of crowdsourcing and the sharing economy. Social manufacturing integrates and organizes socialized manufacturing resources (SMRs) in the form of a social manufacturing community (SMC), which is taken as a flexible host to meet the increasing demand for small-scale, personalized and customized production (Leng et al., 2019). The benefits of social manufacturing have gradually been felt. First, it promotes the use of flat management structures instead of traditional vertical structures, which improves the flexibility of resource organization and assignment (Jiang et al., 2016). Second, social manufacturing absorbs and makes sufficient use of idle social resources, similar to crowdsourcing. The organizational unit, SMC, is beneficial, as it enhances the efficiency, reliability and specialty for manufacturing service delivery (Hirscher et al., 2018). Third, social manufacturing facilitates the servitization extension of products, which supports sustainable product-service systems (Zheng et al., 2019b).

The collaboration problem has already complicated networked manufacturing (Moghaddam and Nof, 2018). The problem is even more serious in social manufacturing due to the distributed socialized resources and social media (Tao et al., 2017). Many studies from academia and industry have focused on this problem. Current studies started from the social media perspective to analyze and address collaboration. For example, Jiang et al. (2016) discussed the promotion of social interaction for business collaboration and depicted the conceptual framework

* Corresponding author. Guangdong Provincial Key Laboratory of Computer Integrated Manufacturing Systems, School of Electromechanical Engineering, Guangdong University of Technology, Guangzhou, Guangdong, 510006, PR China.
E-mail address: piersli@foxmail.com (Z. Li).

https://doi.org/10.1016/j.ijpe.2021.108223
Received 4 April 2021; Received in revised form 29 May 2021; Accepted 5 July 2021
Available online 7 July 2021

for a cyber-physical-social system (CPSS) based on a cyber-physical system (CPS) for social manufacturing. Zhang et al. (2018) enriched the conceptual framework into a realizable CPSS with consideration of the information sharing, physical interaction and social behavior of participants in social manufacturing collaboration and analyzed the influence of social context on manufacturing collaboration. CPSS has been a promising solution to address multidimensional collaboration in social manufacturing from a technical aspect.

The emergence of digital twins has contributed to the construction of smart objects for implementing cyber-physical-related systems (Zheng and Sivabalan, 2020). This approach integrates multiphysics, multiscale, and probabilistic simulation of an as-built system using the best available physical models, sensor updates, behavior history, etc. to mirror the life of a corresponding twin (Litt et al., 2000). In social manufacturing, increasingly deeper participation in social interactions increases the possibilities for collaborative innovation and improvement in product design, production, assembly and other aspects (Xiong et al., 2017), which accordingly increases the cost for product verification and evaluation in physical space. Hence, a digital twin is essential to provide simulation, optimization, computation, and experimentation in cyber space based on a physical SMR to further control the cost and achieve three-dimensional collaboration among physical, cyber and social space. However, the research and development of digital twins, especially for manufacturing equipment, e.g., CNC machines, is usually knowledge intensive and expensive. Only a few leading companies have the ability to develop digital twins and the desire to afford their development costs. Thus, the application of the digital twin in social manufacturing presents a polarization trend. On the one hand, some social manufacturing communities that are spinoffs from large companies have a better basis for digital twin application in order to fully collaborate to serve their inner product innovation needs. On the other hand, the social manufacturing market is also flooded with both manufacturing demands and resources from small and medium enterprises (SMEs). Collaboration still depends on verbal- or text-based communication without enjoying any dividend of a digital twin. Hence, bridging this application gap is a major challenge for intelligent collaboration in social manufacturing.

Intellectual property sharing is a key benefit of self-organized SMCs in social manufacturing (Hirscher et al., 2018). Intellectual property sharing was originally proposed for the open innovation of personalized products (Zheng et al., 2019a) and involves absorbing and sharing knowledge related to product design, production and improvement among SMCs. Leveraging the sharing characteristics of social manufacturing, the digital twin, as a special kind of intellectual property, could also be considered to be shared among social manufacturers. The benefits of digital twin sharing are obvious. First, it could promote the communication and exchange of knowledge, which could narrow the gap of infrastructure intelligence between leading companies and SMEs in terms of cyber-physical-social collaboration. Second, it could facilitate the integration of SMRs in a UPnP manner to improve their interconnectivity and interoperability. Third, swarm experience, knowledge and intelligence could be used to update, improve and upgrade digital twins sustainably. Therefore, through the sharing of digital twins, traditional SMRs could be rapidly integrated to support intelligent collaboration in social manufacturing.

However, concerning the sharing of digital twins, several research questions remain to be addressed:

- How can a digital twin be shared in a distributed, decentralized and dynamic social manufacturing network with consideration of protection and transactions related to its knowledge during joint development and collaboration?
- How can sufficient reconfigurability be enabled for digital twins so that they can be easily configured in the above network and flexible to adapt to heterogeneous runtimes?

To address these questions, this paper proposes a blockchain-based digital twin sharing platform as an integrated solution to enable sharing and protection of the knowledge of the digital twin in decentralized and distributed environments to facilitate the integration of SMRs in a UPnP manner. Domain-driven design (DDD) is adopted as the major methodology to conduct this research. Blockchain technology (BCT) is introduced to construct a novel sharing schema to preserve and retrieve digital twin copyrights under a distributed and decentralized SMC network. Furthermore, to enhance the reusability of digital twin copyrights, a reconfigurable digital twin architecture is proposed to enable UPnP implementation of the shared digital twin copyrights under heterogeneous deployment environments.

The rest of this paper is organized as follows. Section 2 reviews the relevant research on social manufacturing and blockchain-based intellectual property protection. Section 3 illustrates the DDD methodology used to conduct this research. Section 4 illustrates the overall system framework for the sharing platform based on the tactical design of DDD. Section 5 specifies the blockchain-based digital twin sharing schema. Section 6 proposes the reconfigurable digital twin architecture from the object-oriented structure and encapsulation perspective. A demonstrative case study is conducted in Section 7 to verify and evaluate the platform under a scenario of 3D printer digital twin sharing. Finally, Section 8 summarizes the major contributions of this paper and directions for future research.

2. Literature review

2.1. Social manufacturing

Social manufacturing is conceptualized as a promising networked manufacturing paradigm established based on the self-organization and self-configuration of distributed and socialized manufacturing resources and enabled by social media-driven interaction, large-scale collaboration and sharing (Z. Li et al., 2018). Users, enterprises, suppliers, service providers and other participants with different roles establish relationships through social media and form different types of SMCs (Jiang et al., 2016). As the basic organization unit in social manufacturing, the evolution mechanisms of SMC have long been a concern. Xue et al. (2018) proposed a computational model based on a social learning evolution method to quantitatively analyze the dynamics and complexity of social manufacturing in terms of the involved participants. This study revealed the underlying relationship between the micro factors in social manufacturing and the macro evolution of SMC. Concerning the distributed and decentralized nature of SMCs, Leng et al. designed a decentralized construction method for SMC organization based on BCT and developed a prototype system based on Ethereum for self-organizing SMCs of producers and consumers (Leng et al., 2019). To further enhance the intelligence and connectivity of SMC, the concept of CPS is introduced to build smart objects involving man, machine and material, aiming to deliver innovative applications through collaborative interaction (Ding and Jiang, 2016). As a core method for smart object construction, digital twins have been widely applied in CPS (Negri et al., 2017). The multidimensional heterogeneities of large-scale SMRs hinder digitalized transformation during digital twin implementation, which must be explored aggressively.

In social manufacturing, participants need to establish a series of interactive relationships (e.g., social relations, service relations, supply relations) to achieve efficient collaboration. However, interactive relations are sophisticated, dynamic and diverse, so the analysis and management of these relations has become key to the effective implementation of social manufacturing, which has attracted increased attention from academia. Hence, from the perspective of relations, Xiong et al. (2014) adopted AHP and fuzzy evaluation methods to integrate qualitative and quantitative indicators to explore supplier relationship selection in social manufacturing. Based on the social history and interaction context of the participants in a social manufacturing network, Leng and Jiang (2016) proposed a semisupervised learning method to extract the implicit patterns and relationships of these participants to optimize demand forecasting and matching with suppliers. Moreover, the interactive relations in social manufacturing have also brought about a gradual enhancement of the breadth of production collaboration, which accordingly has resulted in more potential collaboration issues. Regarding the decision-making problem among multiple participants in social manufacturing, Andreadis (2015) proposed a technical framework for collaborative design based on social media and real-time streaming media to improve collaboration efficiency. Frazzon et al. (2013) designed a cyber-physical social system (CPSS) considering the factors of information sharing, physical interaction and social behavior of participants in a social manufacturing network and analyzed the influence of social context on production collaboration. As a key enabler of CPS implementation, the digital twin is a complicated software entity and integrates a knowledge-intensive digital model that virtualizes and simulates the physical asset (Lu et al., 2020). The huge input requirement of digital twin development accordingly limits its support of implementing CPSs in social manufacturing (Magomadov, 2020). On the basis of interactive relationships, SMC has the natural advantage of knowledge sharing (Ayala et al., 2017). However, little attention has been given to investigating knowledge sharing, especially digital twin sharing, as an enabling kind of knowledge to promote the implementation of related CPSs.

2.2. Blockchain-enabled intellectual property protection

Blockchain is a novel decentralized infrastructure and distributed computing paradigm that takes a chained data structure for verification and storage and uses distributed consensus algorithms to generate and update data (Wang et al., 2019). Blockchain adapts cryptology methods to protect data transmission and access and applies automated script-based smart contracts to operate rules and data processes (Kouhizadeh et al., 2021). Blockchain was proposed by (Nakamoto, 2008) as a decentralized ledger widely used in initial coin offerings, e.g., Bitcoin. The promising BCT has been widely adopted in supply chain management to address decision-making issues with multiple stakeholders (De Giovanni, 2020; Lohmer et al., 2020). These benefits of BCT adoption also stimulate and migrate its application to more complicated scenarios with multiple stakeholders. Recently, intellectual property protection has become a hot research topic for applying BCT to resolve the traditionally associated problems, with this application especially benefiting from the tamper-resistant nature of BCT. Integrating blockchain with copyright and content protection for traditional media (Savelyev, 2018), such as films (Tsai et al., 2017; Poujol, 2019), music (Cai, 2020) and books (Nizamuddin et al., 2018), has recently been favored. Blockchain is usually taken to maintain the media files in these studies and is combined with other technologies to support business operations, e.g., property modification and transactions. On this basis, researchers have extended the protection scope to more kinds of digital assets. To enable a secure and trustful environment for mold design knowledge licensing and transfer, Li et al. (2019) proposed a blockchain-based knowledge-sharing platform integrated with a private cloud to record mold design knowledge. Regarding circuit copyright protection and transactions, Liang et al. (2020) developed a homomorphic encryption-based blockchain to reduce the transmission cost and improve the efficiency of circuit design data storage and supervision. Focusing on knowledge sharing in the edge computing environment, Li et al. (2020) designed a novel user-centric blockchain to preserve edge knowledge sharing among decentralized intelligent network edges. Lin et al. (2020) also presented a permission edge blockchain to manage peer-to-peer energy data and a knowledge-sharing process for their novel energy-knowledge-trading incentive mechanism. However, the high cohesion and low coupling of SMC have complicated internal knowledge sharing, which should be discussed.

3. Research method

DDD, which centers the development on programming a domain model that has a rich understanding of the processes, rules and knowledge of a domain, has gradually become a promising method for complicated software design, development and implementation (Evans and Evans, 2004). DDD starts the design process from the problem domain following the perspective of the business logic, rather than that of technique (Vernon, 2013). In addition, DDD also contributes to the accumulation and transmission of domain knowledge. Three factors resulted in the adoption of DDD for this research. First, the development and management of digital twins involves cross-disciplinary knowledge and techniques. Second, the research questions belong to different problem domains and should be addressed collaboratively. Third, the research output will be delivered in the form of a software system. DDD can facilitate the fusion of multiple domains through a uniform domain model and ubiquitous language and specialize in this kind of complex system development and implementation.

Generally, DDD involves two stages. The first stage is strategic design. According to the business domain analysis, this stage defines the boundaries and relationships of the solution/system after breaking down the domain problem into several subdomains. Three types of subdomains may be involved in the strategic design. The most important subdomain is the core subdomain, where the greatest focus is placed. As a digital twin is knowledge- and technology-intensive, it is clearly the key value object, so its basic management problem should be of particular concern. The second type is the supporting subdomain, which complements the core subdomain. It wraps the knowledge, business and technique that belong to other subdomains but are related to the core subdomain. Two subdomains are involved in this research to address the research questions separately. The third type is the generic subdomain, which refers to auxiliary issues that do not provide any specific rules to the core business. Generic subdomains can typically be satisfied by ready-made solutions, i.e., the user management issue.

Based on the strategic design, the second stage focuses on tactical design to determine specific solutions for each subdomain. Six kinds of elements are adopted as building blocks of the model for tactical design. Each of their features is briefly described as follows:

3.1. Entity

Many objects are not fundamentally defined by their attributes. They represent a thread of continuity and identity throughout a lifecycle. Sometimes, an object must be matched with another object even though their attributes differ. Meanwhile, an object should also be able to be distinguished from other objects even when they have the same attributes. This kind of object could be extracted and defined as an entity.

3.2. Value object

Many objects have no conceptual identity. They are typically used to describe or compute characteristics of a thing. Hence, when we are concerned only about the attributes and logic of an object and not its identity, we could consider making the concept a value object and giving it related functionality.

3.3. Aggregate

An aggregate refers to a cluster of connected entities that can be treated as a single unit, with a boundary drawn around entities. One entity should be selected as the root of each aggregate, and external objects are allowed to hold references to the root only. Properties and invariants should be defined for an aggregate as a whole, and enforcement responsibility is assigned to the root.

3.4. Repository

A repository is responsible for retrieving and storing all entities and value objects within a particular aggregate collection through a global interface. Generally, creation, modification, configuration and deletion of objects within the aggregate should be defined and exposed by a repository.

3.5. Domain event

A domain event is the object that is used to mark and record a discrete event related to model activity within the system. Domain events are generated and tracked only for event types that domain experts are concerned about.

3.6. Domain service

A domain service is an operation or form of business logic that does not naturally fit within the realm of domain objects. Such a service is responsible for separating the domain functionality from any entity or value object in case of distortion of the definition of a model-based object or the addition of meaningless artificial objects.

Driven by the two general stages of DDD, this paper is constructed in a more specific way, as follows: (1) analyze and partition the problem domain, (2) identify the specific types of subdomains, (3) design each subdomain with domain elements, (4) specify and address the research issues within some subdomains, (5) wrap domain elements into micro-services and combine them into a prototype system, and (6) perform system verification and evaluation.

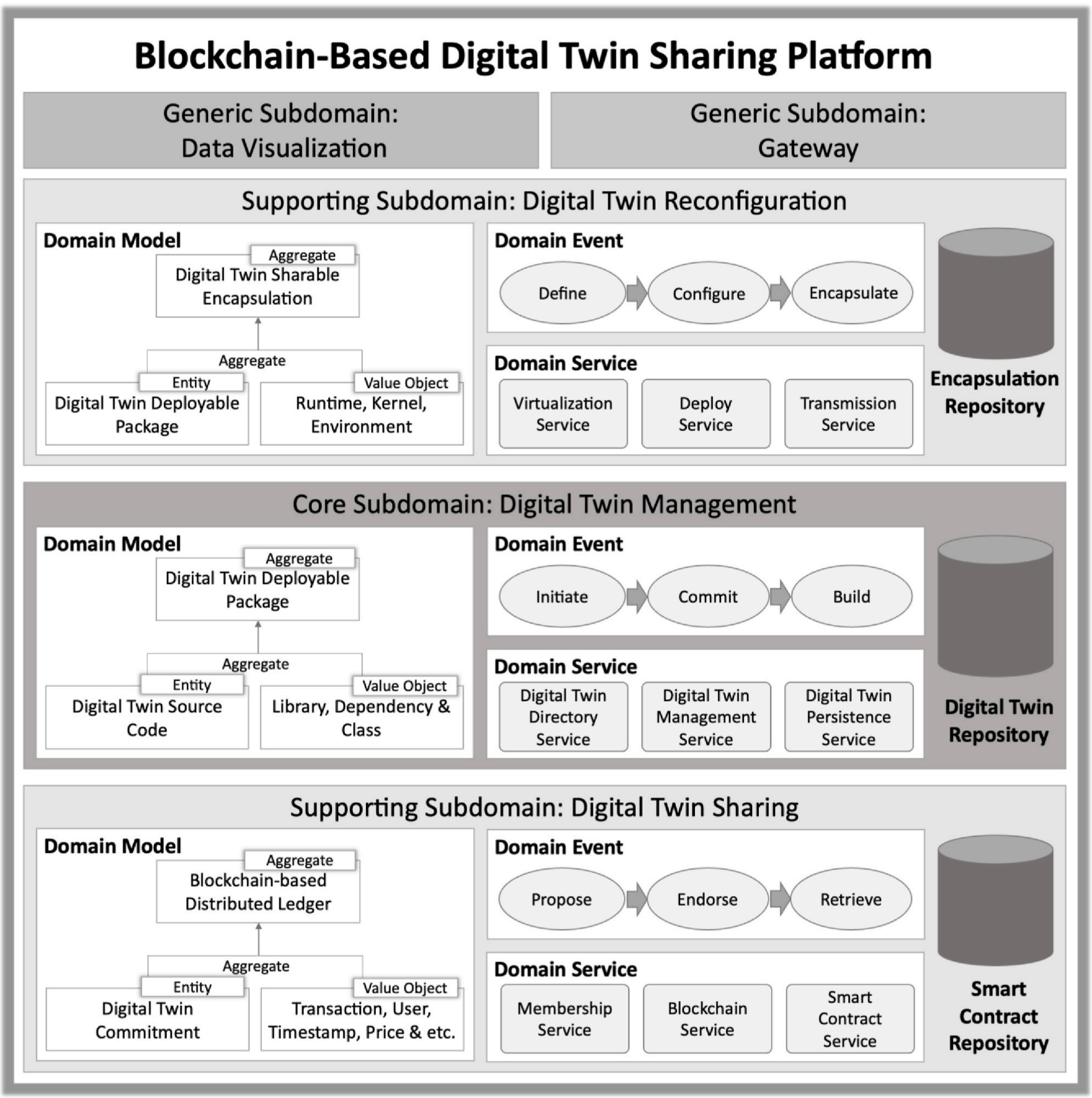

Fig. 1. Overall system framework for BcDTSP.

4. Overall system framework

Based on the strategic design for problem domain partitioning, the system framework is proposed in Fig. 1, following the tactical design of DDD. The framework consists of one core subdomain, two supporting subdomains and two generic subdomains.

4.1. Core subdomain

The core subdomain is critical and fundamental to the business as it represents the foundational concept behind the business and the competitive advantage of a platform. Digital twin management is recognized as a core subdomain because it is the basis for constructing the whole system and it is associated with all business lifecycles. Meanwhile, the digital twin encapsulates core knowledge related to a specific physical object, which could help the platform gain a competitive edge by continuously gathering and organizing this knowledge.

First, the domain model is constructed. The source code for a digital twin is recognized as the entity because it is unique for generating instances of a specific kind of physical resource. The library, dependency and compiler are value objects because they have no identifies but are essential for building the source code. The digital twin deployable package is the aggregate result obtained via the aggregate operation applied to the digital twin source code and its associated libraries, dependencies and compilers. Second, three events are extracted as domain events. (1) The initiate event is triggered when the digital twin source code is created or imported. (2) The commit event is used to tag changes in the source code if new developments have been contributed and confirmed, for example, the addition or revision of source code. (3) The build event is activated when the source code is compiled into a deployable package. Third, three domain services are essential for domain operation. (1) The digital twin management service provides the mechanisms needed to implement full lifecycle management of the digital twin in terms of an integrated development environment, version control and compiling. (2) The digital twin directory service enables the indexing of both digital twin development projects and executable packages and provides retrieval service for external invokes. (3) The digital twin persistence service provides interfaces to store digital twins in the repository.

4.2. Supporting subdomain

The supporting subdomain refers to the contexts that could help perform ancillary or supporting functions related directly to what the business does. Two supporting subdomains are separated based on domain problem partitioning. The digital twin reconfiguration subdomain supports the customization and (re)configuration of digital twins under heterogeneous deployment scenarios. First, the entity of the domain model in this subdomain is the digital twin executable package from the core subdomain as it takes effect on a certain model or type of physical resource. Then, through the aggregation of specific configurations, runtimes and environments, a sharable encapsulation of digital twins can be constructed. A universally unique identifier is granted for the encapsulation because it has already bound with a definite SMR under a target deployment scenario. Second, three domain events are critical for recognizing the key milestones of business logic. (1) The define event signifies the instantiation of a digital twin with specific definitions and information for a physical SMR. (2) The configure event is responsible for adjusting and configuring the necessary parameters and settings for the digital twin. (3) The encapsulate event is used to trigger the construction of a digital twin with its essential runtime for the target deployment environment. Third, this subdomain involves three services. (1) The virtualization service provides virtualized computing environments for digital twin encapsulation. (2) The deploy service enables a series of scripts to assist in the implementation and deployment of digital twin executable packages in a defined environment. (3) The transmission service is responsible for providing multiple transport protocols to connect and send encapsulated digital twins to end-users.

The other supporting subdomain focuses on the business of digital twin sharing. First, the domain model is established based on BCT considering the tamper-proof persistence of digital twins. The commitment is regarded as the entity because each commitment describes a current status of the digital twin that needs to be versioned. The transaction, user, timestamp, profiles and other related attributes should be attached to a specific commitment to express the business implications so that they are categorized as value objects. Then, a blockchain-based distributed ledger can be generated through the merging of commitments of multiple digital twins to maintain the unique states of version and sharing across the network. Second, propose, endorse and retrieve are three domain events in digital twin sharing logic. (1) The propose event is used to respond to calls for digital twin sharing. (2) The endorse event is activated when the sharing proposals need to be verified in the distributed network. (3) All digital twin shares are indexed and obtained through the retrieve event. Third, three domain services are promoted to perform the business logic. (1) The membership service offers an abstraction of membership operations, including all cryptographic mechanisms and protocols behind issuing certificates, validating certificates and user authentication. (2) The blockchain service is responsible for constructing the peer-to-peer network and providing the basic functional components to maintain the distributed ledger. (3) The smart contract service achieves contract management in terms of formulation, development, deployment and testing dependent on the blockchain service.

4.3. Generic subdomain

The generic subdomain refers to a necessary part of the system that facilitates the business but is not core to the business. In general, generic subdomains can be adopted from mature solutions or purchased from a vender instead of requiring complete development. Two subdomains are partitioned based on the general business requirements of the platform. The first is data visualization, which enables a series of analysis charts to illustrate the diversified data involved in the platform. Apache ECharts, Ant G2 and other mature frameworks can be used directly for this subdomain. The second subdomain focuses on the gateway business to facilitate the establishment of connection and communication between the physical SMR and its digital twin. Existing solutions from our research team can be adopted to fulfil this subdomain (Fang et al., 2013; M. Li et al., 2018, 2019).

5. Blockchain-based digital twin sharing schema

Benefiting from SMC, crowdsourcing is common for digital twin development. In addition, traditional digital twins could also be customized or improved by community members to facilitate their further application. Hence, intellectual property protection is complicated, with multiple stakeholders who contribute and maintain the same digital twin. In this section, a blockchain-based digital twin sharing schema is proposed to enable the overall traceability of the shared rights of digital twins among multiple stakeholders. To maintain a fine-grained record of contributions, the schema depends on a source code management paradigm to realize digital twin codevelopment. In addition, BCT is taken to maintain the development records as effective evidence to partition the shared rights.

A Merkel tree is adopted to construct and maintain the unique state for the source code of a digital twin project. Three objects are designed for source code management, as shown in Fig. 2. A tree object is responsible for describing the directory structure of the source code. Each directory node is represented by a tree object, and the hierarchical relationships among directories and directory files are reflected in the contents of the tree objects. A blob object is a leaf node of a tree object that realizes the encapsulation of the content of source code files, such as

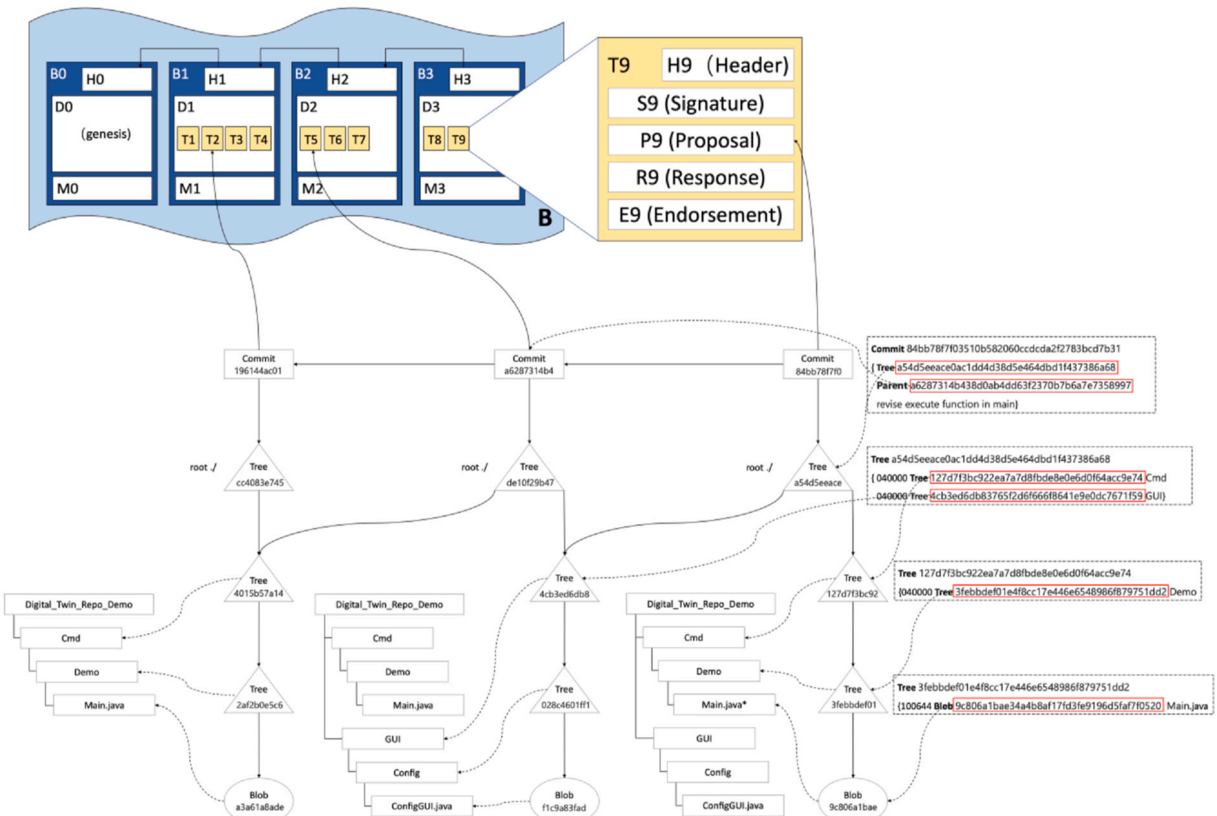

Fig. 2. Blockchain-based digital twin sharing schema.

the class files or configuration files. A commit object is generated by the commit event and is the root of all tree objects. The commit object is also responsible for versioning the current digital twin development project and works as the entry to index and analyze the object relationship to differentiate branches and versions.

The values of these three kinds of objects are generated through a hash algorithm, e.g., SHA256. The hash value of a blob object is calculated based on its content. The tree and commit object merge the values of their adjacent leaf nodes into a string, which is then used as the basis to compute their hash values. According to the hash value generation method, any change in the source code will result in a change in the hash value of its blob object. Subsequently, the hash value of the corresponding tree object and commit object will also be influenced. For example, the change on Main.java (Denoted with *) in Fig. 2 will result in reconstructing its blob object with a new hash value. This update is propagated to all its parent objects to recalculate their hash values.

Based on this feature of the three kinds of objects, a shared rights preservation mechanism is implemented, as illustrated in Fig. 3. Each contributor can obtain the original source code copy of a digital twin from the community repository and create its own local repository, and the development of the digital twin is conducted on the local repository. Then, the confirmed versions are pushed back to the community repository, which results in two situations. First, the development addresses the general requirements of a digital twin, such as fixing current

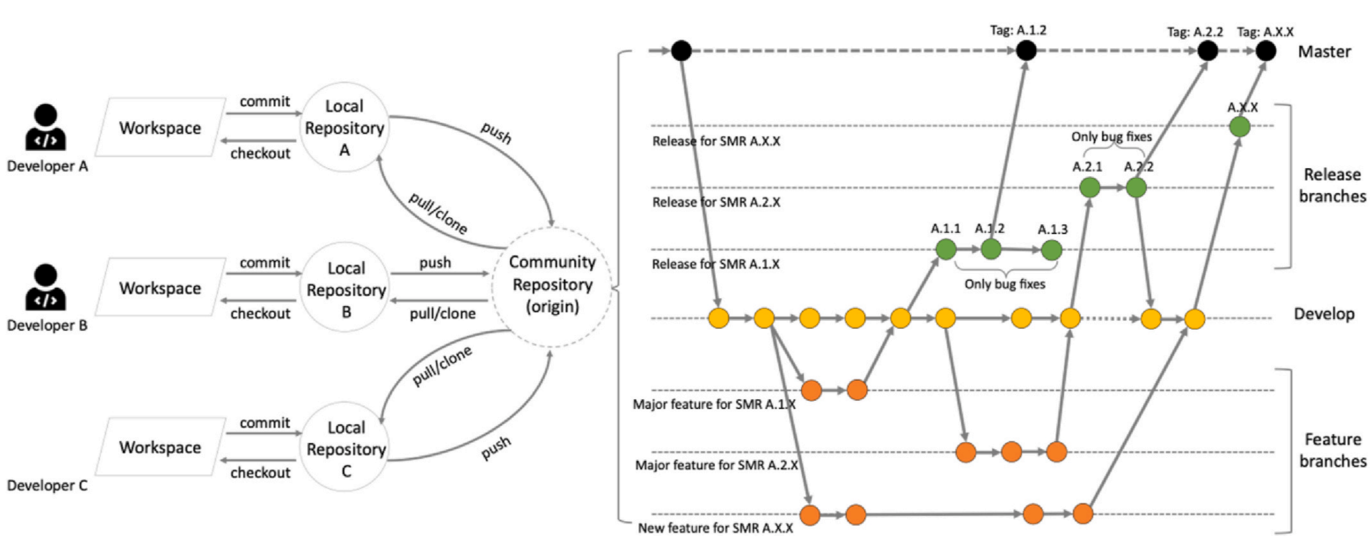

Fig. 3. Shared rights preservation mechanism for digital twin codevelopment.

bugs or improving the performance. In this situation, the commit objects can be further forwarded and attached to the original community repository in a time sequence to form a master chain. Second, some developments may be performed only for the purpose of specific customization or adaptation. The commit object for this kind of development will link to the previous commit object that it is based on and construct a new branch chain that is parallel to the master chain.

BCT acts as the underlying technology to support the sharing scheme. Two smart contracts of BCT are designed to address existing management issues. As the contributors are distributed and their organization is based on a weakly centralized architecture (e.g., community repository) or even a decentralized architecture (e.g., single repository), it is challenging and inefficient to synchronize all repositories into a global repository. A smart commit contract is then developed to maintain the global consistency of all shared digital twin projects across multiple repositories in social manufacturing. Specifically, the community repository packs commit objects in the form of transactions and request the persistence of transactions into a distributed ledger that is maintained by related members of the SMC, such as developers and users. Hence, a transaction in the distributed ledger acts as a credible depository receipt to prove a historical version of a digital twin project, which could be used as the token to retrieve the source code back from the community repository for shared rights auditing.

In addition, concerning credible visibility and traceability for digital twin trading behavior, a smart trading contract is designed to execute the transactions of digital twins on the blockchain.

In a transaction, any commit object of a digital twin can be taken as the commodity. Three key steps are followed to complete a transaction. First, a transaction proposal is initiated with the necessary parameters. Second, the proposal is endorsed based on a defined consensus mechanism. Third, the smart trading contract extracts the source code from the community repository and invokes the digital twin reconfiguration subdomain to deliver a runnable digital twin entity for the buyer.

6. Reconfigurable digital twin

Two types of variations exist in the target implementation environments of digital twins. One is the changes in business logic for digital twin execution. The other results from the different runtimes for digital twin deployments. To enhance the adaptability of digital twins served in different environments, a reconfigurable digital twin framework is proposed in Fig. 4. Two types of reconfigurability are embodied to address application uncertainty: one aims to improve the suitability of the shared digital twin for its corresponding kinds of heterogeneous SMRs in terms of structure and capability, whereas the other focuses on the runnable encapsulation of the digital twin to enhance its self-regulation.

6.1. GRMVC-based digital twin core

Digital twins commonly exist in the form of applications in cyber space. To promote the adaptability of the digital twin in differentiated business scenarios, a GRMVC-based digital twin core is proposed to separate the business logic and presentation/interaction. The GRMVC-based digital twin core can be formulized as a five-tuple $\chi_{DTC} = \{\boldsymbol{G}, \boldsymbol{R}, \boldsymbol{M}, \boldsymbol{V}, \boldsymbol{C}\}$, where.

- *G is the gateway component that connects and communicates with the physical entity.*
- *R is the repository component that provides data persistence and query service.*
- *M is the model component that manages and manipulates the data, logic and rules of a digital twin.*
- *V is the view component that presents digital twin data to the user and handles user interaction.*
- *C is the controller component that links the view with model components through event-driven processes.*

The normal working logic between these components follows seven steps, as shown in Fig. 4.

(1) The user interacts with a view to operate the digital twin.
(2) View alerts the controller of a particular event.
(3) Controller triggers the model to handle the event.

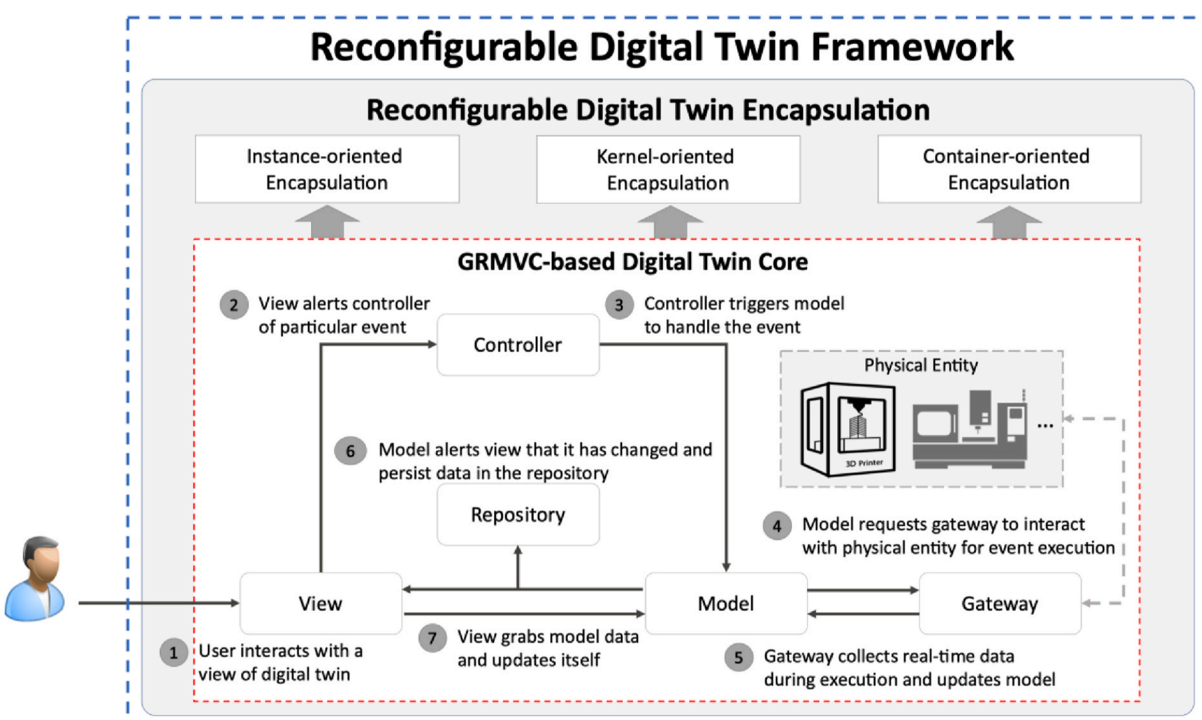

Fig. 4. Reconfigurable digital twin framework.

(4) Model requests the gateway to interact with physical entities for event execution.
(5) The gateway collects real-time data during execution and updates the model.
(6) Model alerts view that it has changed and persists data in the repository.
(7) View grabs the model data and updates itself.

Due to the separation of components, the reconfigurability of the GRMVC-based digital twin core is embodied in three aspects. First, the separation of view facilitates multiple views and enables the user interface to reconfigure multiple views of the same digital twin data simultaneously according to business needs. Second, the model abstracts data interoperability and is independent of the view and controller, which improves the reconfigurability of digital twins across different platforms because the model can easily be migrated. Third, the separation of view and controller makes them work in a pluggable manner such that they can be reconfigured dynamically and even changed at run time as business logic requires.

6.2. Reconfigurable digital twin encapsulation

The reconfigurability feature of digital twin encapsulation can be formally described by a Mealy finite-state machine, which is denoted as a six-tuple $\Gamma = \{S, s_0, \Sigma, \Lambda, T, R\}$ consisting of the following:

- *S is a finite, nonempty set of encapsulation states for a digital twin.*
- *s_0 is the initial encapsulation state of a digital twin.*
- *Σ is a finite, nonempty set of reconfigurations.*
- *Λ is a finite, nonempty set of reconfigured encapsulation instances of a digital twin.*
- *T is a transition function $T: S \times \Sigma \rightarrow S$ mapping pairs of digital twin encapsulation and reconfiguration to the corresponding reconfigured encapsulation state.*
- *R is a reconfiguration execution function $R: S \times \Sigma \rightarrow \Lambda$ mapping pairs of a digital twin encapsulation and a reconfiguration to generate the corresponding reconfigured encapsulation instance.*

Based on the Mealy finite-state machine, three kinds of encapsulation granules are defined with different levels of reconfigurability.

Definition 1. If tuple S is expanded as a triple $\Phi = \{CI, OP, EF\}$, where:

- CI is a finite set of computing infrastructure $CI = \{CI_0, CI_1, CI_2, \ldots, CI_m\}$, m is the total number of kinds of computing infrastructure, and CI_0 denotes the nonencapsulated state for the computing infrastructure.
- OP is a finite set of operating systems $OP = \{OP_1, OP_2, \ldots OP_n\}$, and n is the total number of eligible operating systems for a digital twin.
- EF is a finite set of encapsulation formats, $EF = \{EF_1, EF_2, \ldots, EF_k\}$, and k is the total number of types of encapsulation formats.

This type of digital twin is defined as instance-oriented encapsulation. $\forall s = \{CI_a, OP_b, EF_c\} \in S$ is reconfigured to $s' = \{CI_{a'}, OP_{b'}, EF_{c'}\} \in S$, $\exists (a == a') \wedge (b == b') \wedge (c == c') = False$.

Definition 2. If tuple S is expanded as a triple $\Psi = \{K^T, K^F\}$, where:

- K^T is a finite set of kernel types $K^T = \{K_1^T, K_2^T, K_3^T, \ldots, K_n^T\}$, and n is the total number of kernel types.
- K^F is a finite set of kernel features $K^F = \{K_1^F, K_2^F, K_3^F, \ldots, K_m^F\}$, and m is the total number of kernel features.

This type of digital twin is defined as kernel-oriented encapsulation. $\forall s = \{K_a^T, K_b^F\} \in S$ is reconfigured to $s' = \{K_{a'}^T, K_{b'}^F\} \in S$, $\exists (a == a') \wedge (b == b') = False$.

Definition 3. If tuple S is expanded as a triple $\Theta = \{C^I, C^F\}$, where:

- C^I is a finite set of container infrastructure $C^I = \{C_1^I, C_2^I, C_3^I, \ldots, C_n^I\}$, and n is the total number of kernel types.
- C^F is a finite set of kernel features $C^F = \{C_1^F, C_2^F, C_3^F, \ldots, C_m^F\}$, and m is the total number of kernel features.

This type of digital twin is defined as container-oriented encapsulation. $\forall s = \{C_a^I, C_b^F\} \in S$ is reconfigured to $s' = \{C_{a'}^I, C_{b'}^F\} \in S$, $\exists (a == a') \wedge (b == b') = False$.

7. A demonstrative case

To verify the proposed solution, a 3D printing scenario in a laboratory environment is considered to demonstrate the sharing of digital twins to facilitate the decentralized integration of heterogeneous social manufacturing resources. In this section, the scenario is first introduced. Then, heterogeneous social manufacturing resources are rapidly integrated through the sharing of digital twins. Finally, specific issues and potential merits of the proposed BcDTSP are summarized.

7.1. Case scenario

Motivated by small- and medium-sized 3D printing providers, two business phases are included in this case scenario. The first phase is the manufacturing phase that uses 3D printers to fulfill the customers' printing demands. The second phase focuses on warehousing of the printed components so that they can easily be consolidated into a package for delivery. The specific business logic is given in Fig. 5.

Based on this scenario, three 3D printers have some structural heterogeneities. The Makerpi K5 is regarded as an ordinary 3D printer. The SHDM 3DDP-500 provides more control features, such as separate nozzle fan on-off control, fan speed control and wider range of print speeds. Bluemaker 345 is considerably different, as it has dual printer nozzles and more optional settings. Hence, a digital twin for each 3D printer can be reconfigured for these printers. Meanwhile, two types of warehousing units with different business logic types that can be executed by the same warehousing infrastructure are also applied. A digital twin of the warehousing unit is developed and shared on BcDTSP to demonstrative the rapid integration of these resources. Table 1 lists the specifications of all the hardware infrastructure in this case.

7.2. System implementation

The prototype of BcDTCP is developed based on Spring Boot Version 2.0.1. The development environment is based on JDK8, Lombok, IDEA and Gradle. Netflix-eureka is selected for service registration and discovery. Openfeign is enabled for interservice invocation. Netflix-zuul is integrated as the web gateway. RabbitMQ is used as the advanced message queuing protocol for message transportation.

The core domain is established using GitLab community edition. GitLab provides the underlying mechanism for source code management. The domain service is developed under the scheme of FIPA (Bellifemine et al., 1999) in the form of microservices. The domain event is realized through the publish-subscribe pattern of RabbitMQ. The generated domain events are wrapped into messages and inserted into the queue for consumption by related services.

The digital twin sharing supporting domain is founded based on the Hyperledger Fabric V2.1 framework. The network consists of two organizations with three peers in each and two orderers. Raft is chosen as the consensus mechanism, and CoachDB is adopted because the commitment object is structured in the form of JSON (JavaScript Object Notation). The digital twin reconfiguration supporting domain is built

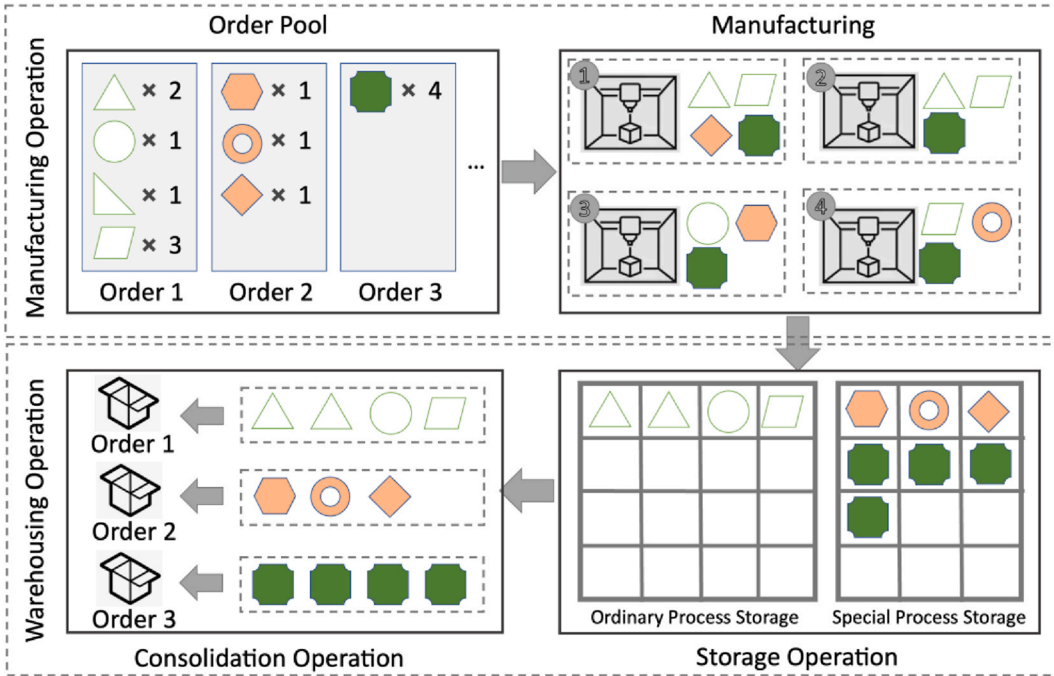

Fig. 5. General business logic for small- and medium-sized 3D printing providers.

Table 1
Hardware infrastructure specifications.

Hardware Type	Model	Specifications
3D Printer	SHDM FDM 3DDP-500	Print Volume: 500*500*600 mm Printing Speed: 60–120 mm/s Layer Resolution: 0.05 mm–0.4 mm Nozzle Diameter: 0.4–0.8 mm
3D Printer	Makerpi K5	Print Volume: 200*200*280 mm Printing speed: 40–60 mm/s Layer Resolution: 0.05 mm–0.3 mm Nozzle Diameter: 0.4 mm
3D Printer	Bluemaker 345	Print Volume: 300*350*450 mm Printing Speed: 30–400 mm/s Layer Resolution: 0.02 mm–0.4 mm Nozzle Diameter: 0.1–1.0 mm Dual Printer Nozzles
Smart Warehousing Unit (12 Locations)	Self-designed	Silicon Labs EFR32BG Bluetooth SoC starter kit Light tags*12 Huawei M5 smart tablet*1 Wearable barcode scanner*1 Customized shelves with 12 locations
Smart Warehousing Unit (16 Locations)	Self-designed	Silicon Labs EFR32BG Bluetooth SoC starter kit Light tags*16 Huawei M5 smart tablet*1 Customized shelves with 16 locations

upon a series of technologies. The compiling service depends on GNU (Stallman, 1985) compiler collection, Maven and Gradle. The encapsulation service is implemented via Docker, Remastersys and VMWare Workstation for container-oriented, kernel-oriented and instance-oriented encapsulation, respectively. The instantiation service is delivered by scripts for instantiation methods of different encapsulation formats under different target deployment environments.

The digital twin of 3D printers is developed based on an open-source project, OctoPrint release 1.4. The digital twin of the smart warehouse units comes from our previous study (Kong et al., 2020). It is built as an Android application that can be run on a smart tablet associated with an EFR32BG development board for lighting tag control and a wearable scanner for barcode identification if necessary.

7.3. Demonstration

The above digital twins have been committed to the community repository. Fig. 6 illustrates the current state of the "ipark_pda" source code in the community repository and the distributed ledger.

The demonstration is divided into three phases. The first phase is leveraging the shared digital twins to achieve the rapid integration of heterogeneous resources. Generally, the integration is completed in three steps. The Octoprint for the Bluemaker 345 3D printer is taken as an example in Fig. 7. First, basic information on the resources should be provided to define a new digital twin using the existing sharing in the community repository. Second, a specific configuration must be settled upon for the digital twin in terms of its features and heterogeneities so that the related components of the digital twin core can be reconfigured to these heterogeneities. Fig. 8 illustrates the reconfigurations of two 3D printers and the same warehousing units with different execution logic types. Third, the basic settings and parameters for the target deployment environment must be provided for digital twin encapsulation. The generated digital twin encapsulation (OVF file) is downloaded by the user to import through the VMWare vCenter to instantiate a virtual machine that has a built-in digital twin model of the Bluemaker 345 3D printer.

The second phase is verifying the reconfigurability of the digital twin under different deployment environments. The current deployment target is a virtual machine created by VMWare vCenter via the instance-oriented encapsulation of the OVF file ESXi hypervisor. A new deployment is provided based on the Docker installed on Ubuntu 18.04LTS in a Dell T7920 workstation. Hence, the current instance-oriented encapsulation should be reconfigured as a container-oriented encapsulation, as illustrated in Fig. 9. The downloaded reconfigured digital twin can be

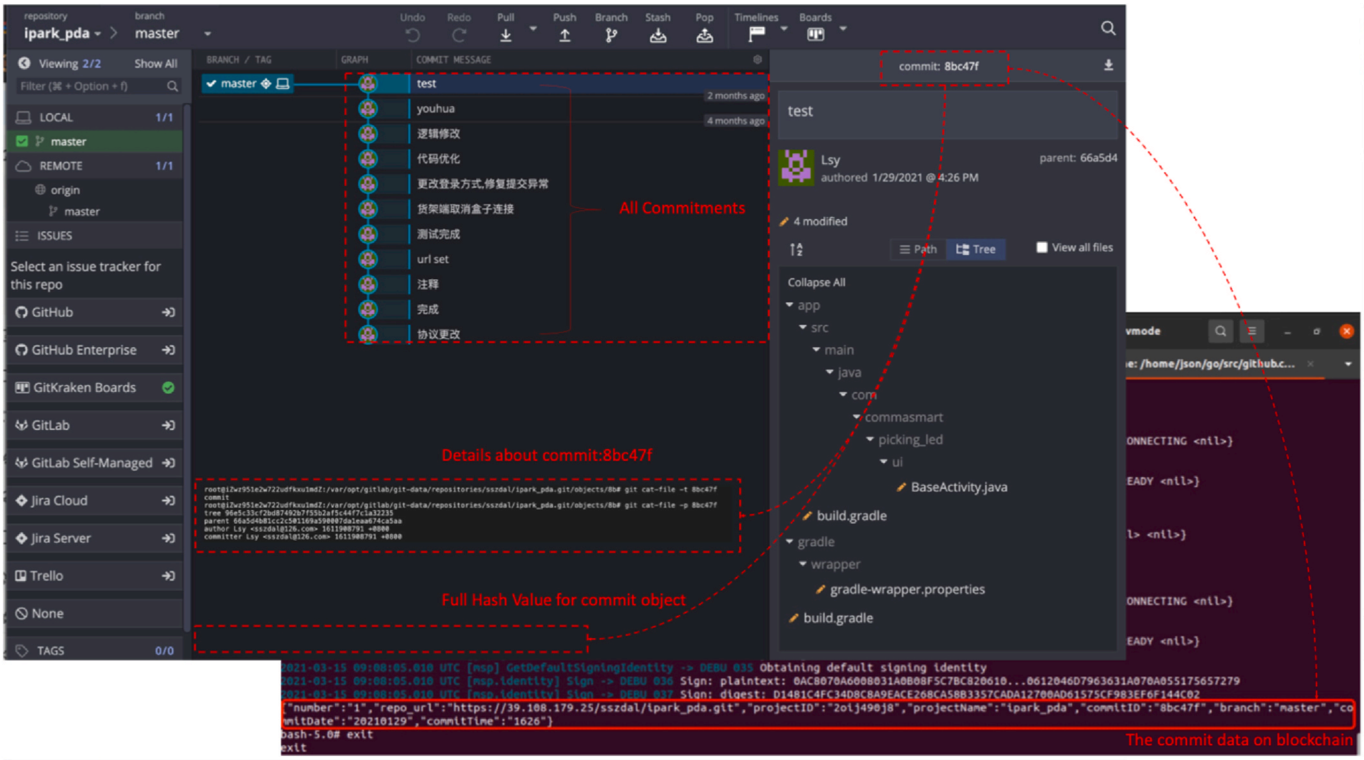

Fig. 6. Current state of the ipark_pda source code in the repository and ledger.

loaded by Docker through the command line or script and connected with the original 3D printer for task execution.

The third phase is executing the digital twins for real-life production, as illustrated in Fig. 10. One printing demand is issued to the 3D printers (Bluemaker 345 & SHDM FDM 3DDP-500) through their digital twins. The SHDM printer fulfills this demand using one extruder, while the Bluemaker printer executes with two extruders loaded with different filaments. After printing, the printed components (replaced by demonstrative building blocks) will be temporarily stored on the warehousing units that are configured for storage so that each cell of the warehousing unit maintains the same type of printed components. Finally, order consolidation must be performed on a warehousing unit reconfigured as a picking station, and each cell of this warehousing unit is used for a custom order to consolidate the produced components of multiple warehousing units (storage).

7.4. Discussion

This demonstrative case study represents a specific verification of the proposed platform, and a comparison is conducted to evaluate its effectiveness and efficiency, as illustrated in Table 2.

The smart contract performance of the BcDTSP is tested in terms of the operation of storing commits, as shown in Fig. 11. Hyperledger Caliper is adopted as the testing tool. The basic settings and testing results are illustrated in Table 3. The presented throughput and transaction latency are the averages of multiple runs.

BatchSize is a very important parameter that must be configured carefully to optimize the performance. BatchSize determines the number of transactions the orderer will collect before constructing a block. If the system load is lower than the convergence value, increasing BatchSize will generally increase the latency because more transactions must be collected for a block. Conversely, if the transaction receiving rate is close to the convergence value, increasing BatchSize will result in only a small throughput increase and latency decreases. Hence, the sensitivity for throughout and latency determines the selection of BatchSize.

Meanwhile, some potential merits of the proposed BcDTSP are also summarized from the perspective of the software system (Kleppmann, 2017). The first merit is the high reliability, which is a benefit of adopting blockchain. The decentralized working scheme combined with the distributed sharing scheme of the digital twin makes it possible to separate the original dependency on the central sharing platform from any eligible node of the sharing network to enhance the fault-tolerant ability. In addition, the consensus mechanism of blockchain also guarantees high consistency among nodes, even when a failure node recovers to normal.

The second merit reflects the scalability in terms of two aspects. First, the platform architecture can easily be extended according to business needs, due to the domain-driven design, to decouple the business domain. New functions or features can be added or removed in the form of microservices. Second, the digital twins in the platform could be continuously improved and upgraded with the sharing mechanism based on the source code. More plug-ins may also be added to expand the current functionalities of digital twins. In particular, more developers will be attracted to contribute once an appropriate incentive mechanism is designed.

The third merit results from maintainability. Through domain partitioning, each domain can clearly be considered and assigned more professional operation and maintenance engineers to its domain to guarantee operability. Meanwhile, the complexity of understanding each domain is also reduced, so the maintenance response and efficiency can be improved accordingly. Furthermore, the platform also enables better evolvability. As microservices are the smallest system granules, current microservices could be upgraded or evolved individually or recombined to create a new domain on demand.

However, the implementation of the BcDTSP has some limitations. First, the current prototype system is used only for verification and evaluation of this study. Follow-up research to address various factors, such as privacy, security, and authority issues, should be conducted to enhance the applicability of the BcDTSP in a production environment. Second, the sharing mechanism can only protect and prove the

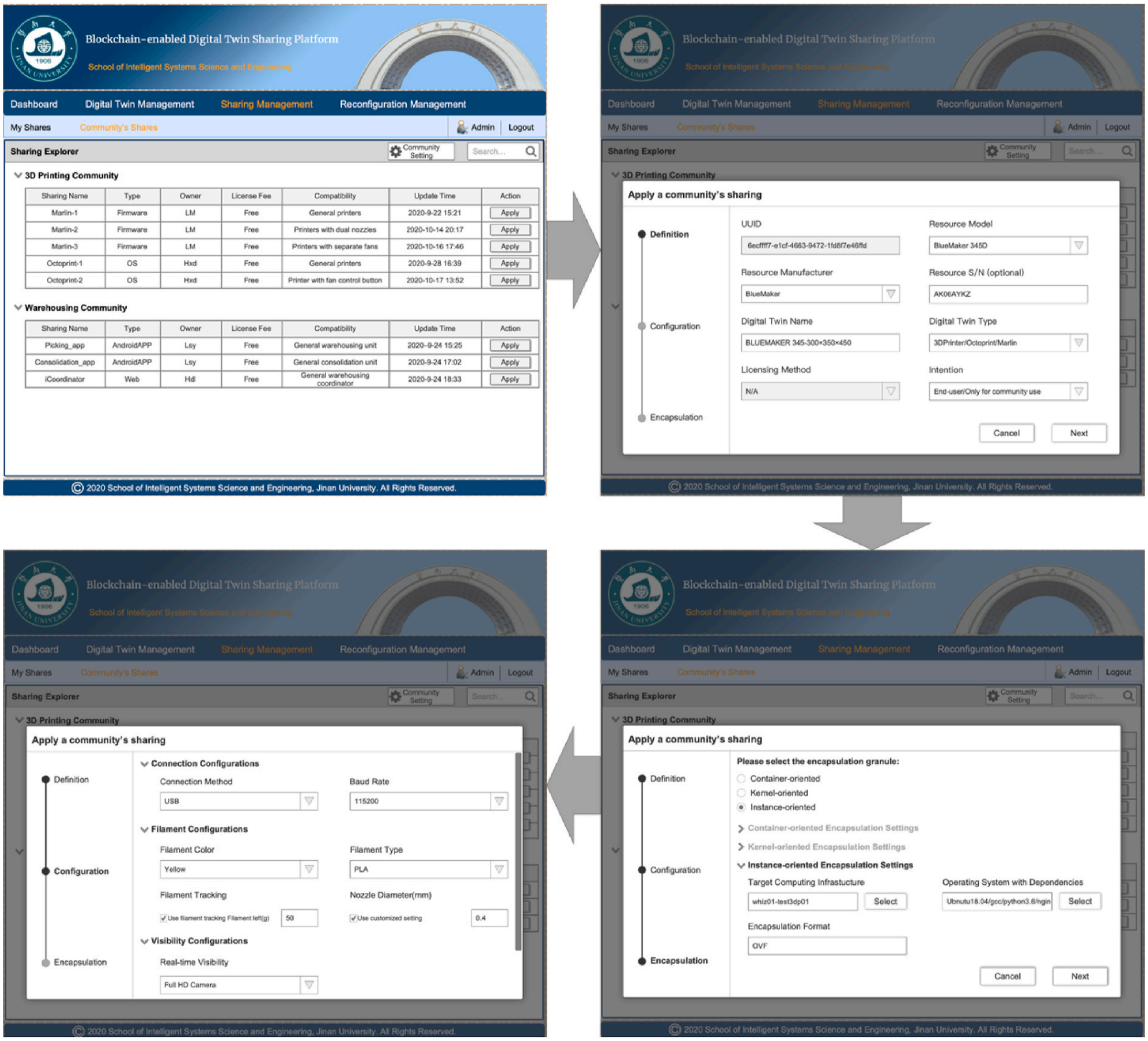

Fig. 7. Demonstration of 3D printer integration.

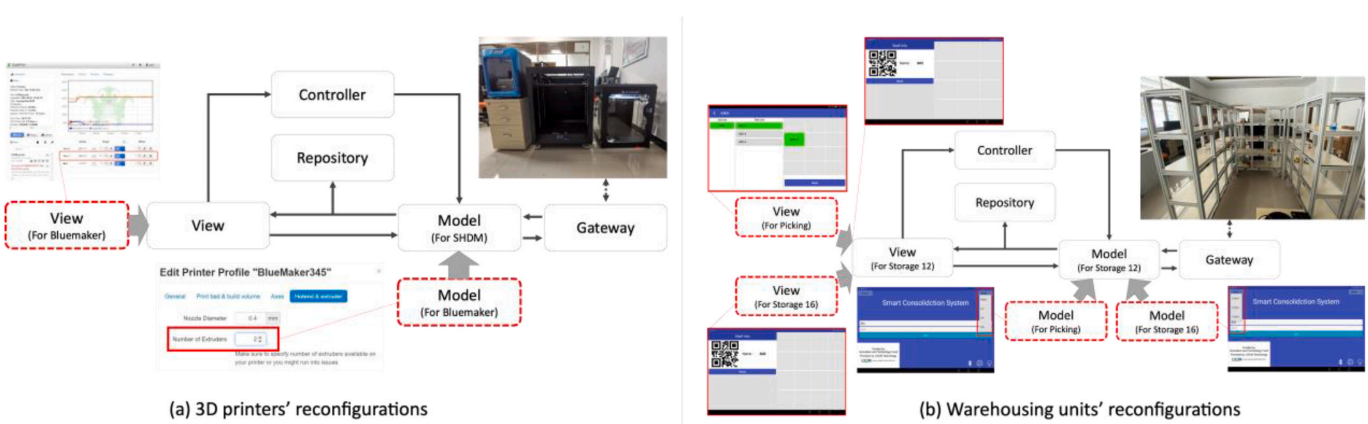

Fig. 8. Reconfigurations of the digital twin core for heterogeneous SMRs.

ownership and contribution of a digital twin; it cannot limit unauthorized copying of encapsulated digital twins. Hence, more technological means (e.g., activation policy of Microsoft products) should be applied to restrict the use of encapsulated digital twins, especially if fees are charged for digital twin licenses. Third, the performance testing was conducted in an ideal network environment: a quantitative analysis of network influence should be performed in the following real-life application with our collaborated company.

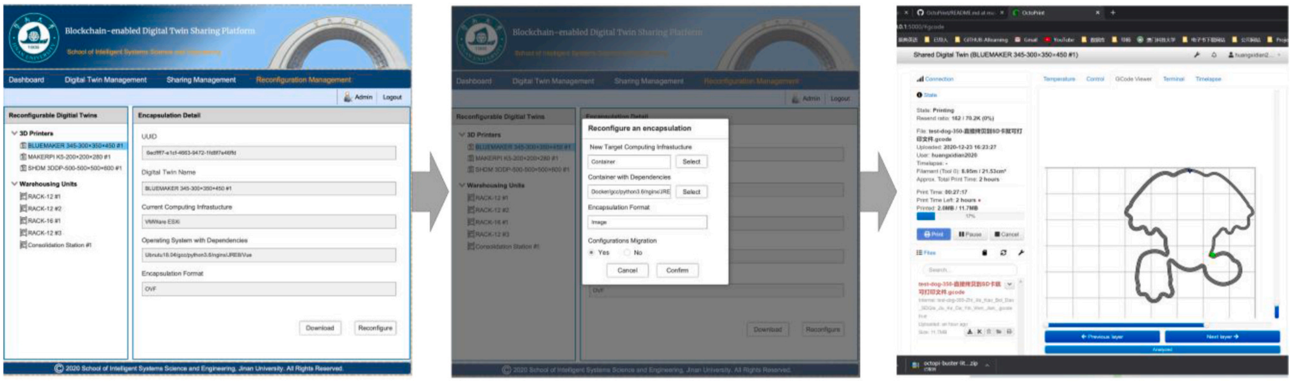

Fig. 9. Digital twin encapsulation reconfiguration.

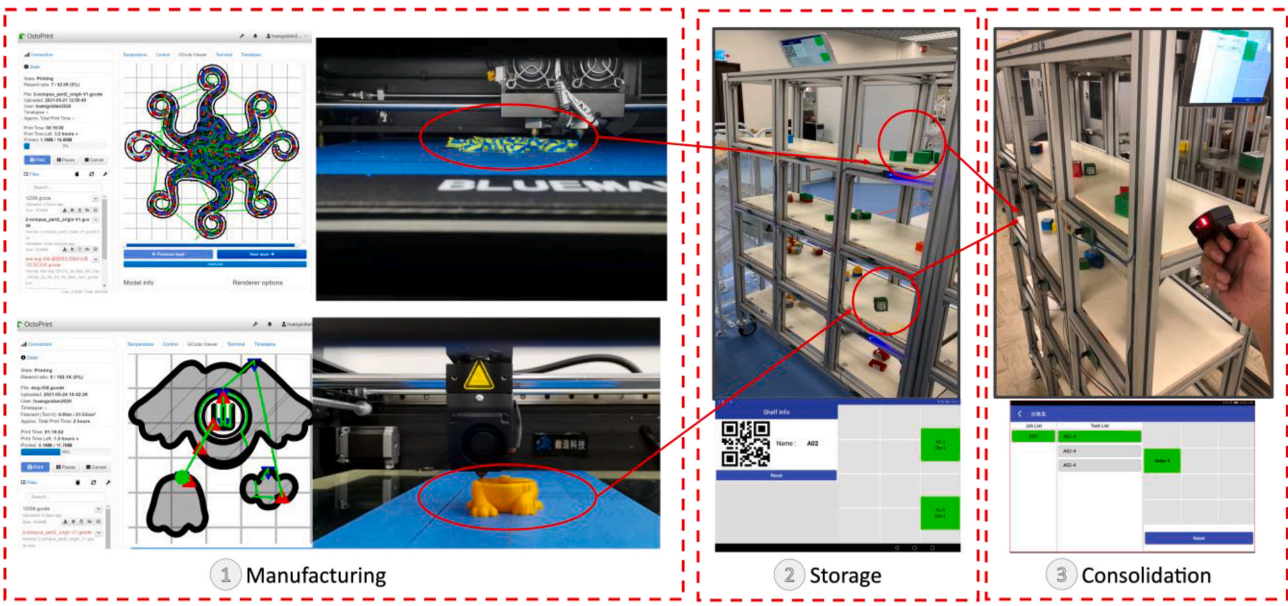

Fig. 10. Demonstrative execution for the case scenario.

Table 2
Comparison between BcDTSP and general sharing platforms.

	BcDTSP	General sharing platforms (e.g., GitHub)
Resource Integration Usability	Easy, by importing digital twins	Difficult, by further compiling and deploying of digital twins
Resource Integration Time	Minute level	Hour or above level
Digital Twin Mobility	High, by encapsulation	Low, by source code
Digital Twin Protectiveness	Supported, blockchain-enabled evidence traceability	Not supported
Digital Twin Source Code Visibility	Only for developer	All people
Digital Twin Diversity	High level, all branches driven	Low level, master branch driven
Digital Twin Reconfigurability	Source code, configuration, encapsulation	Only source code

8. Conclusion

The evolution of social manufacturing has promoted the exploitation of mass idle social manufacturing resources. These resources usually lack sufficient digitization capacity to be integrated with current manufacturing schemas. Motivated by this real-life problem, this paper has explored the underlying issues hindering social manufacturing resource integration. To leverage the strength of SMC in sharing, a knowledge sharing platform, named BcDTSP, is proposed to facilitate sharing of digital twin models among SMCs. A blockchain-based digital twin sharing scheme is designed to protect the copyright and knowledge of a digital twin during sharing. In addition, a reconfigurable digital twin architecture that considers heterogeneous digital twin deployment environments is presented. Finally, the BcDTSP is verified and evaluated by means of a demonstrative 3D printing case motivated by a real-life printing business.

The contributions of this paper are reflected in three primary aspects. First, the framework of the BcDTSP uses the concept of divide and conquer to partition a problem domain from the domain perspective and associate knowledge with experiences to design the subdomains. It provides a reference to inspire the design and implementation of inter-disciplinary knowledge-based systems.

Second, the blockchain-based digital twin sharing schema provides a novel approach to integrate BCT for digital twin sharing and protection, making sufficient use of the collaboration and sharing advantages of SMC. It makes a preliminary contribution to exploring and perfecting software copyright sharing, transaction and protection, especially for knowledge-intensive products. Moreover, the approach could be further

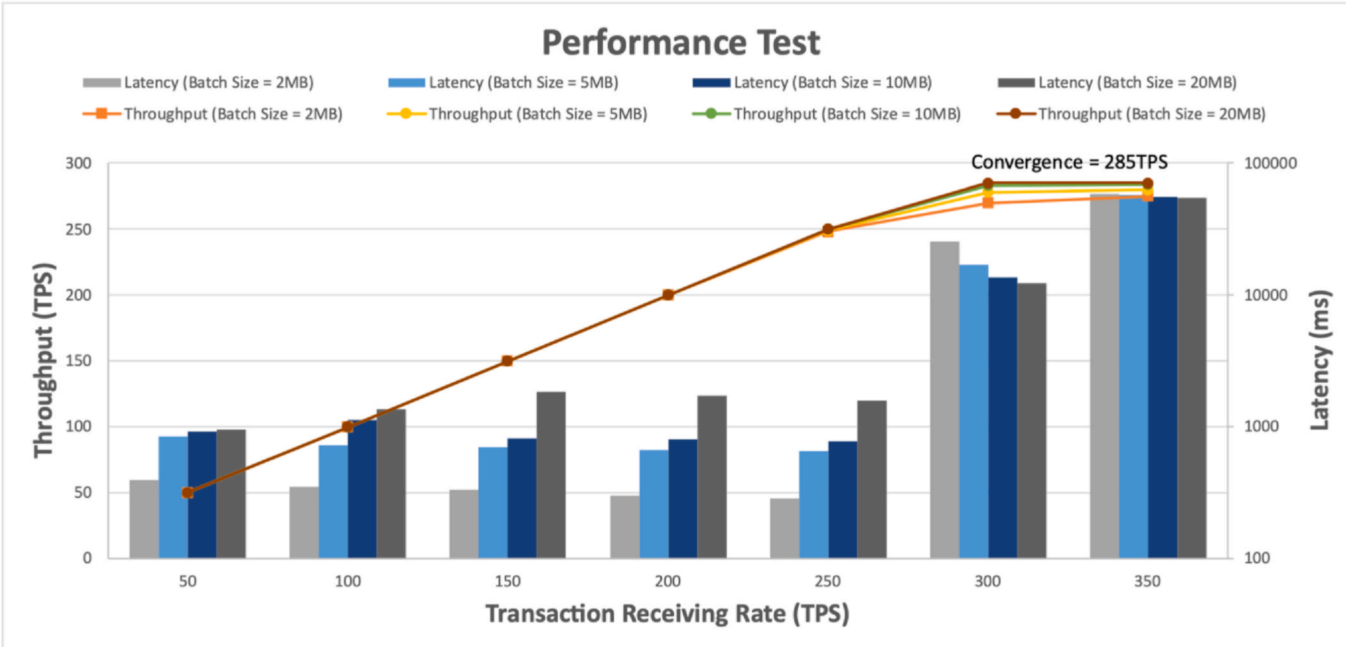

Fig. 11. Performance test result.

Table 3
Basic settings for the performance tests.

Hardware Specification	Physical Dell T7920 workstation (Xeon 6230R*2, 128 GB RAM, 1 TB SSD)
Virtualization	Hypervisor: VMWare EXSi 6.5u2
Virtual Machine Specification	2vCPU, 4 GB RAM, 60 GB SSD with 18.04LTS
Network	10 Gb vSwitch of VMWare
Parameter Settings	● Number of Channels: 1 ● Number of Orderers: 2 ● BatchTimeout = 1 s ● StateDB Database: CoachDB ● Endorsement Policy: 1 of N ● Number of organizations: 3 ● Number of peers in an organization: 5

improved to enable the potential commercialization of open-source software.

Third, the reconfigurable digital twin framework enables reconfigurability from the structure and encapsulation perspectives to address the heterogeneity of SMRs. The GRMVC-based digital twin core presents a scientific implementation method to promote the reusability and adaptability of digital twins. It can be further applied to design and construct more kinds of smart objects that must be associated with physical entities. Additionally, the encapsulation granules for digital twins are innovatively defined and implemented. They can be migrated to wrap more kinds of software entities that need to be implemented over different runtimes.

This research could be extended from three aspects. First, an incentive mechanism is essential to cover the operation cost and enhance the business sustainability of the BcDTSP. Additionally, a rational measurement method should be designed to quantify digital twin contributions as a basis to distribute the potential benefits from digital twin sharing and transactions. Finally, the current study considers only limited manufacturing resources in the laboratory for verification and evaluation, and more types of resources in the manufacturing and logistics industries should be involved in BcDTSP in the near future.

Declaration of competing interest

None.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No. 52005218), the Guangdong Basic and Applied Basic Research Fund (Guangdong Natural Science Fund, No. 2019A1515110296), the Fundamental Research Funds for the Central Universities (No. 21620359) and the National Key R&D Program of China (No. 2019YFB1705401).

References

Andreadis, G., 2015. A collaborative framework for social media aware manufacturing. Manufacturing Letters 3, 14–17.
Ayala, N.F., Paslauski, C.A., Ghezzi, A., Frank, A.G., 2017. Knowledge sharing dynamics in service suppliers' involvement for servitization of manufacturing companies. Int. J. Prod. Econ. 193, 538–553.
Bellifemine, F., Poggi, A., Rimassa, G., 1999. JADE–A FIPA-Compliant Agent Framework. Presented at the Proceedings of PAAM, London, p. 33.
Cai, Z., 2020. Usage of deep learning and blockchain in compilation and copyright protection of digital music. IEEE Access 8, 164144–164154.
De Giovanni, P., 2020. Blockchain and smart contracts in supply chain management: a game theoretic model. Int. J. Prod. Econ. 228, 107855.
Ding, K., Jiang, P., 2016. Incorporating social sensors and CPS nodes for personalized production under social manufacturing environment. Procedia CIRP 56, 366–371.
Evans, E., Evans, E.J., 2004. Domain-driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley Professional, Hoboken, USA.
Fang, J., Qu, T., Li, Z., Xu, G., Huang, G.Q., 2013. Agent-based gateway operating system for RFID-enabled ubiquitous manufacturing enterprise. Robot. Comput. Integrated Manuf. 29, 222–231.
Frazzon, E.M., Hartmann, J., Makuschewitz, T., Scholz-Reiter, B., 2013. Towards socio-cyber-physical systems in production networks. Procedia Cirp 7, 49–54.
Hirscher, A.-L., Niinimäki, K., Armstrong, C.M.J., 2018. Social manufacturing in the fashion sector: new value creation through alternative design strategies? J. Clean. Prod. 172, 4544–4554.
Jiang, P., Ding, K., Leng, J., 2016. Towards a cyber-physical-social-connected and service-oriented manufacturing paradigm: social Manufacturing. Manufacturing Letters 7, 15–21.
Kleppmann, M., 2017. Designing Data-Intensive Applications: the Big Ideas behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, Inc., Sebastopol, USA.
Kong, X.T., Zhong, R.Y., Zhao, Z., Shao, S., Li, M., Lin, P., Chen, Y., Wu, W., Shen, L., Yu, Y., 2020. Cyber physical ecommerce logistics system: an implementation case in Hong Kong. Comput. Ind. Eng. 139, 106170.

Kouhizadeh, M., Saberi, S., Sarkis, J., 2021. Blockchain technology and the sustainable supply chain: theoretically exploring adoption barriers. Int. J. Prod. Econ. 231, 107831.
Leng, J., Jiang, P., 2016. Mining and matching relationships from interaction contexts in a social manufacturing paradigm. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47, 276–288.
Leng, J., Jiang, P., Xu, K., Liu, Q., Zhao, J.L., Bian, Y., Shi, R., 2019. Makerchain: a blockchain with chemical signature for self-organizing process in social manufacturing. J. Clean. Prod. 234, 767–778.
Li, G., Dong, M., Yang, L.T., Ota, K., Wu, J., Li, J., 2020. Preserving edge knowledge sharing among IoT services: a blockchain-based approach. IEEE Transactions on Emerging Topics in Computational Intelligence 4, 653–665.
Li, M., Lin, P., Xu, G., Huang, G.Q., 2018a. Cloud-based ubiquitous object sharing platform for heterogeneous logistics system integration. Adv. Eng. Inf. 38, 343–356.
Li, M., Xu, G., Lin, P., Huang, G.Q., 2019. Cloud-based mobile gateway operation system for industrial wearables. Robot. Comput. Integrated Manuf. 58, 43–54. https://doi.org/10.1016/j.rcim.2019.02.004.
Li, Z., Liu, X., Wang, W.M., Vatankhah Barenji, A., Huang, G.Q., 2019. CKshare: secured cloud-based knowledge-sharing blockchain for injection mold redesign. Enterprise Inf. Syst. 13, 1–33.
Li, Z., Wang, W.M., Liu, G., Liu, L., He, J., Huang, G.Q., 2018b. Toward Open Manufacturing. Industrial Management & Data Systems.
Liang, W., Zhang, D., Lei, X., Tang, M., Li, K.-C., Zomaya, A., 2020. Circuit copyright blockchain: blockchain-based homomorphic encryption for IP circuit protection. IEEE Transactions on Emerging Topics in Computing. https://doi.org/10.1109/TETC.2020.2993032.
Lin, X., Wu, J., Bashir, A.K., Li, J., Yang, W., Piran, J., 2020. Blockchain-based incentive energy-knowledge trading in IoT: joint power transfer and AI design. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2020.3024246.
Litt, J.S., Simon, D.L., Meyer, C., DePold, H., Curtiss, J.R., Winston, H., Wang, Y., Statler, I., Gawdiak, Y., 2000. NASA aviation safety program: aircraft engine health management data mining tools roadmap. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology II. International Society for Optics and Photonics, pp. 292–298.
Lohmer, J., Bugert, N., Lasch, R., 2020. Analysis of resilience strategies and ripple effect in blockchain-coordinated supply chains: an agent-based simulation study. Int. J. Prod. Econ. 228, 107882.
Lu, Y., Liu, C., Kevin, I., Wang, K., Huang, H., Xu, X., 2020. Digital Twin-driven smart manufacturing: connotation, reference model, applications and research issues. Robot. Comput. Integrated Manuf. 61, 101837.
Magomadov, V., 2020. The digital twin technology and its role in manufacturing. In: Presented at the IOP Conference Series: Materials Science and Engineering. IOP Publishing, 032080.
Moghaddam, M., Nof, S.Y., 2018. Collaborative service-component integration in cloud manufacturing. Int. J. Prod. Res. 56, 677–691.
Nakamoto, S., 2008. Bitcoin: a peer-to-peer electronic cash system [WWW Document]. URL. https://bitcoin.org/bitcoin.pdf.
Negri, E., Fumagalli, L., Macchi, M., 2017. A review of the roles of digital twin in CPS-based production systems. Procedia Manufacturing 11, 939–948.
Nizamuddin, N., Hasan, H.R., Salah, K., 2018. IPFS-blockchain-based authenticity of online publications. In: Presented at the International Conference on Blockchain. Springer, pp. 199–212.
Poujol, P., 2019. Online Film Production in China Using Blockchain and Smart Contracts. Springer, Cham.
Savelyev, A., 2018. Copyright in the blockchain era: promises and challenges. Comput. Law Secur. Rep. 34, 550–561.
Stallman, R., 1985. The GNU manifesto [WWW Document]. URL. http://www.gnu.org/gnu/manifesto.en.html.
Tao, F., Cheng, Y., Zhang, L., Nee, A.Y., 2017. Advanced manufacturing systems: socialization characteristics and trends. J. Intell. Manuf. 28, 1079–1094.
Tsai, W.-T., Feng, L., Zhang, H., You, Y., Wang, L., Zhong, Y., 2017. Intellectual-property blockchain-based protection model for microfilms. In: Presented at the 2017 IEEE Symposium on Service-Oriented System Engineering (SOSE). IEEE, pp. 174–178.
Vernon, V., 2013. Implementing Domain-Driven Design. Addison-Wesley Professional, Hoboken, USA.
Wang, Y., Singgih, M., Wang, J., Rit, M., 2019. Making sense of blockchain technology: How will it transform supply chains? Int. J. Prod. Econ. 211, 221–236.
Xiong, G., Chen, Y., Shang, X., Liu, X., Nyberg, T.R., 2014. AHP fuzzy comprehensive method of supplier evaluation in social manufacturing mode. In: Presented at the Proceeding of the 11th World Congress on Intelligent Control and Automation. IEEE, pp. 3594–3599.
Xiong, G., Wang, F.-Y., Nyberg, T.R., Shang, X., Zhou, M., Shen, Z., Li, S., Guo, C., 2017. From mind to products: towards social manufacturing and service. IEEE/CAA Journal of Automatica Sinica 5, 47–57.
Xue, X., Wang, S., Zhang, L., Feng, Z., Guo, Y., 2018. Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing. IEEE Transactions on Industrial Informatics 15, 3343–3355.
Zhang, J.J., Wang, F.-Y., Wang, X., Xiong, G., Zhu, F., Lv, Y., Hou, J., Han, S., Yuan, Y., Lu, Q., 2018. Cyber-physical-social systems: the state of the art and perspectives. IEEE Transactions on Computational Social Systems 5, 829–840.
Zheng, P., Lin, Y., Chen, C.-H., Xu, X., 2019a. Smart, connected open architecture product: an IT-driven co-creation paradigm with lifecycle personalization concerns. Int. J. Prod. Res. 57, 2571–2584.
Zheng, P., Sivabalan, A.S., 2020. A generic tri-model-based approach for product-level digital twin development in a smart manufacturing environment. Robot. Comput. Integrated Manuf. 64, 101958.
Zheng, P., Wang, Z., Chen, C.-H., Khoo, L.P., 2019b. A survey of smart product-service systems: key aspects, challenges and future perspectives. Adv. Eng. Inf. 42, 100973.