

Lingming KONG, Rui ZHAO, Chimay J. ANUMBA, Weisheng LU, Fan XUE

Open BIM exchange on Blockchain 3.0 virtual disk: A traceable semantic differential transaction approach

© The Author(s) 2024. This article is published with open access at link.springer.com and journal.hep.com.cn

Abstract Building information modeling (BIM) has become prevalent in construction engineering management. However, the efficiency of traditional file-based BIM exchange between multidisciplinary teams remains low due to the transmission of redundant data from mostly unchanged BIM objects. Additionally, the traceability of changes between BIM files is limited. This paper proposes a traceable semantic differential transaction (tSDT) approach for open BIM exchange, which minimizes data redundancy and enables semantic-level traceability of BIM changes. Furthermore, openBIMdisk implements the tSDT and provides a Blockchain 3.0 virtual disk to support efficient, traceable, and secure BIM exchanges across multiple blockchain services. A pilot study of a modular construction project demonstrated the effectiveness of tSDT and openBIMdisk. Experimental results indicated that tSDT achieved minimal BIM redundancy for storing and restoring all BIM changes, using a sheer 0.007% of disk space on average. openBIMdisk facilitated BIM version management and object-level semantic traceability with a response time of 5.3 ms. The contributions of this paper are twofold. First, tSDT offers a novel and efficient approach for semantic BIM change traceability. Second, openBIMdisk provides BIM practitioners with a Blockchain 3.0 application featuring intuitive and user-friendly interfaces for BIM exchange.

Keywords building information modeling, traceable semantic differential transaction, Blockchain 3.0, Open BIM exchange, information redundancy

1 Introduction

Building information modeling (BIM) has emerged as a milestone and is progressively mandated in conceptualizing, generating, communicating, and storing information for construction engineering management (Sacks et al., 2022). BIM is defined by Eastman et al. (2008) as a digital integration of information throughout the life cycle of a building, from concept to design through construction and deconstruction. However, BIM collaboration has been challenged by poor BIM interoperability, which has resulted in inefficient data exchange (Oraee et al., 2019; Liu et al., 2017; Zhao et al., 2017).

Industrial Foundation Classes (IFC) was developed to support seamless data exchange across heterogeneous professional software for BIM collaboration (BSI, 2023). IFC presents a digital building model via an inheritance hierarchy that defines inheritable attributes and relationships of objects (Borrmann et al., 2018). IFC is gradually growing into a *de-facto* standard of open BIM (Lai and Deng, 2018). However, exchange with IFC files encounters three prevalent challenges, i.e., redundancy (Zhang et al., 2014), traceability (Hastig and Sodhi, 2020), and security (Zhao et al., 2021; Berlo et al., 2021; Lou et al., 2021; Das et al., 2021).

Researchers employed cloud computing and blockchain technologies to ease the challenges to IFC exchange in the literature. Cloud computing and graph models partially overcame the redundancy and traceability challenges associated with the centralized file-based exchange paradigm (Sacks et al., 2022; Wang et al., 2023). Blockchain has the potential to complement BIM's traceability and security; Still, it is incapable of storing IFC files directly, as an IFC project file size is tens of, or even several hundred of megabytes, which exceeds

Received Jan. 7, 2024; revised Jul. 3, 2024; accepted Jul. 25, 2024

Lingming KONG, Rui ZHAO, Weisheng LU, Fan XUE (✉)
Department of Real Estate and Construction, The University of Hong Kong, Hong Kong 999077, China
E-mail: xuef@hku.hk

Chimay J. ANUMBA
College of Design, Construction and Planning, University of Florida, Gainesville, FL 32611, USA

The work presented in this paper was supported by Hong Kong Research Grants Council (RGC) (Grant No.17200221) and in part by Hong Kong Innovation and Technology Commission (ITC) (No. ITP/029/20LP).

blockchain's storage capability. Studies in the literature addressed the challenges using two architectures. One architecture employs two-tier BIM storage, i.e., off-chain BIM files and an on-chain storage for hashed file records (Tao et al., 2021; 2022; Celik et al., 2023b). The two-tier architecture compromises on blockchain's limited storage capacity that only metadata of BIM files, e.g., object version, issuing discipline, etc., are stored on the blockchain. Thus, the two-tier architecture supports only file-level traceability and circumvents data redundancy issues in BIM exchange. However, the semantic-level traceability is becoming increasingly important in real-time applications such as digital twins (Boje et al., 2020; Liang et al., 2024; Hakimi et al., 2024) and artificial intelligence (Chen et al., 2024). The other architecture records all semantic BIM changes, computed by algorithms like semantic differential transaction (SDT) (Xue and Lu, 2020), on a permissioned Blockchain 2.0 backbone. The second architecture aims to respond to all three challenges but is confined by the IFC processing algorithm's leading time and semantic-level BIM traceability. As a consequence, the proposed implementation of Blockchain technologies in BIM exchange is still in its infancy and far from reality (Celik et al., 2023a; Figueiredo et al., 2022).

This paper aims to present a novel approach for achieving efficient, traceable, and secure open BIM exchange in project-level application. First, a novel traceable SDT (tSDT) approach is developed as a significant extension of semantic traceability capability and performance to the SDT (Xue and Lu, 2020). Furthermore, OpenBIMdisk realizes the tSDT and provides a Blockchain 3.0 virtual disk for supporting efficient, traceable, and secure BIM exchanges in construction engineering management.

The remainder of this paper is organized as follows. Section 2 provides a systematic review of IFC-based BIM exchange, existing open BIM exchange paradigms, and blockchain applications. Section 3 describes the conceptual framework of this paper. Section 4 introduces the tSDT approach with its two components: semantic interoperability as pre-processing and tSDT processing. Section 5 presents the openBIMdisk with a Blockchain 3.0 design and graphical user interfaces. A case study validated the feasibility and efficiency of the tSDT approach and openBIMdisk in Section 6. Section 7

discusses the contributions and limitations of this paper. The final section identifies areas for further improvement and research.

2 Literature review

2.1 BIM exchange and IFC

“No BIM is an island” (Chen, 2018), BIM exchange stands as an essential yet costly objective of collaboration in construction engineering management. This is because numerous off-the-shelf BIM vendors provide an array of BIM software programs for BIM users to choose from (Wu and Zhang, 2019), not to mention the specialized disciplines and professional inputs inherent in construction projects (Deng et al., 2019). To promote the exchange efficiency of BIM, IFC is proposed as the data exchange standard (Laakso and Kiviniemi, 2012). As an open BIM standard, IFC presents a digital building model via an inheritance hierarchy that defines the inheritable attributes and relationships of objects (Borrmann et al., 2018; Guo et al., 2020). The IFC schema is also open to accommodating new extensions, such as nine actions in three categories of typical BIM semantic changes (Kong and Xue, 2023). Researchers have utilized the IFC inheritance hierarchy for BIM exchange (Krijnen and Beetz, 2020; Tang et al., 2020).

IFC data can be encoded in various file formats, each having benefits and trade-offs pertaining to software support, scalability, and readability. Table 1 outlines the benefits and tradeoffs of three primary IFC formats, namely STEP Physical File (SPF), Extensible Markup Language (XML), and JavaScript Object Notation (JSON). SPF is the most widely used format in IFC practice and the most compact of the three formats (BSI, 2023). Nevertheless, SPF grapples with issues of poor readability and redundant data arising from inconsistent entity names and obscure parent-child relationships (Krijnen and Beetz, 2020). Alternatively, the XML format is established through an automatic mapping from EXPRESS to XML as presented in ISO-10303 Part 28 Edition 2 (ISO, 2016). While the XML format ensures data consistency, it exhibits drawbacks in terms of data redundancy, processing

Table 1 List of mainstream IFC formats

Format	Extension	Size	Pros	Cons
STEP Physical File (SPF)	.ifc	100%	<ul style="list-style-type: none"> • Most widely used • Most compact 	<ul style="list-style-type: none"> • Data redundancy (↓) • Low flexibility (↓↓) • Poor readability (↓↓)
Extensible Markup Language (XML)	.ifcxml	113%	<ul style="list-style-type: none"> • Enhanced readability (↑) • Higher computational efficiency (↑) • Higher compact (↑) 	<ul style="list-style-type: none"> • Data redundancy (↓↓) • Low flexibility (↓) • Byte-level inconsistency (↓)
JavaScript Object Notation (JSON)	.ifcjson	148%	<ul style="list-style-type: none"> • Enhanced readability (↑) • Higher computational efficiency (↑↑) • Enhanced flexibility (↑) 	<ul style="list-style-type: none"> • Data redundancy (↓↓↓) • Low compact (↓) • Preliminary application

efficiency and inflexibility, thus restricting its application (Afsari et al., 2017; Xue and Lu, 2020; Xu et al., 2022). Multiple previous studies have substantiated that parsing JSON data is more efficient with fewer computing resources, ultimately enhancing the operational efficiency of web service applications (Wang, 2011; Lin et al., 2012; Sun et al., 2020). Consequently, a JSON serialization of IFC data, as proposed by Afsari et al. (2017), is employed in this research to bolster web-based BIM exchange.

2.2 Comparison of existing open BIM exchange paradigms

Data redundancy, traceability, and security are prevalent and significant concerns of open BIM exchange (Zhang et al., 2014; Hastig and Sodhi, 2020; Zhao et al., 2021). With the development of information technology, different types of open BIM exchange paradigms were developed to resolve the concerns. Table 2 lists three types of existing open BIM exchange paradigms: (1) centralized, (2) decentralized, and (3) distributed. The term “centralized” means BIMs are exchanged through centralized files, the term “decentralized” means BIM sources are available in multiple centrally controlled data centers, and the term “distributed” means the centralized control of BIM sources is further eliminated and can be accessed through any peers in the network.

Centralized open BIM exchange paradigm is the most straightforward way for BIM communication between stakeholders. For the improved efficiency of centralized open BIM exchange, the BIM collaboration format (BCF) and the Model View Definition (MVD) (Sacks et al., 2018) were included in the IFC standard. However, BCF requires definitions before use and lacks generalization, and the MVD contains random contents (e.g., GUIDs). As a result, centralized open BIM exchanges encounter inconsistencies and are time-consuming to revise (Lee et al., 2021b). Moreover, the risk to security, encompassing potential issues such as data loss, data manipulation, and data privacy, prevents traceable centralized BIM exchange (Berlo et al., 2021; Lou et al., 2020).

Decentralized open BIM exchange paradigm is a new trend to support efficient and simultaneous BIM collaboration. Cloud computing is the most widely adopted ICT for decentralized open BIM exchange (Zhao and Taib, 2022). The advent of cloud BIM is considered to realize

real-time data exchange (Wong et al., 2014). Several BIM software vendors already provide cloud services, such as Autodesk BIM 360, Graphisoft BIM Explorer (BIMx), and BIMcloud (Afsari et al., 2016). Furthermore, the decentralized exchange of open BIM provides enhanced traceability. For instance, Sacks et al. (2022) proposed a ‘Cloud BIM’ approach to building modeling, which maintains consistency across federated discipline-specific models. However, issues of security still present challenges to cloud BIM integration (Mutis and Paramashivam, 2019).

The recent emergence of the blockchain offers a fresh perspective on recognizing the reliable and efficient distributed open BIM exchange of the construction industry (Hijazi et al., 2021). Blockchain is a distributed database that facilitates peer-to-peer communications through cryptographic transactions and distributed consensus protocols (Belotti et al., 2019). Thus, a secure, transparent, and distributed data exchange is achieved by eliminating the requirement for centralized authorities (Li et al., 2019). Many researchers have discussed the possible application of blockchain for open BIM exchange to promote security and bitwise traceability (Das et al., 2021). According to Table 2, blockchain’s distributed exchange performance is the best among the listed BIM exchange paradigms. In conclusion, the distributed nature of Blockchain responds to the challenges of basic traceability and security issues in BIM data exchange. Therefore, adopting blockchain for BIM data exchange can be a promising solution. The only remaining problems that prevent the integration of BIM and blockchain are BIM data redundancy and BIM object-level traceability.

2.3 Blockchain in construction

Blockchain is a recent focal point of research. Blockchain has rapidly gained recognition as a potentially disruptive technology poised to propel the construction industry into the fourth industrial revolution (Perera et al. 2020; Wu et al. 2024) and achieve sustainability (Lu et al., 2024). Blockchain can be categorized into two types based on privacy: permissioned blockchain and permission-less or public blockchain (Nawari and Ravindran, 2019). Permissioned blockchains restrict the actors who can contribute to the consensus of the system state, whereas permissionless blockchains allow any participant to validate transactions or create smart contracts. Hyperledger Fabric and Ethereum are two common examples of the

Table 2 Comparison of three open BIM exchange paradigms

Exchange paradigms	Redundancy	Traceability	Security	References
Centralized	High	Lack	Low	Sacks et al. (2018); Pinheiro et al. (2018)
Decentralized	Medium	Medium	Low	Mutis and Paramashivam (2019); Sacks et al. (2022)
Distributed	Very high	High	High	Xue and Lu (2020); Tao et al. (2021)

types, respectively. Construction projects are typically sensitive to copyrights, budgets, and accountability of stakeholders. Concerns about data manipulation, misuse of information, and other security issues arising from third-party involvement in construction projects necessitate tight management of BIM accessibility. Some previous research explored how to promote the security of permissioned blockchain for BIM through information technologies, such as Zero-knowledge proof (Kong et al., 2024). Therefore, permissioned blockchain is a better choice for most BIM projects (Kong et al., 2022) and adopted in this research. The word “permissioned” is ignored in the following part of this paper for readability.

The milestone Blockchain 2.0, such as *Hyperledger Fabric*, includes noteworthy attributes such as decentralization to avert single point of failure (Monrat et al., 2019), identity anonymization (Sun et al., 2016), immutable data storage (Li et al., 2022), and transparent transaction broadcasting (Bodkhe et al. 2020). In the literature, Blockchain 2.0 offers off-the-shelf domain-independent transactional functionalities for consensus mechanism and membership services, and thus holds significant promise for facilitating construction data exchange (Swan, 2015).

A Blockchain 3.0 computational paradigm ports Blockchain 2.0 transactional functionalities to construction domain-specific systems and applications; so that decentralized apps (DApps) can respond well to the needs of construction project stakeholders (Angelis and Ribeiro de Silva, 2019; Zhao et al., 2023). For example, Zhao et al. (2023) designed a Blockchain 3.0 with a two-step workflow with a locally cached digital twin of data and confirmed a savings of 99.2%–99.8% of the blockchain’s response time and off-line structured BIM objects query functions for three modular construction applications.

Some researchers have theoretically discussed the integration of blockchain with BIM for BIM data storage and validation. For example, a preliminary case study presented by Pradeep et al. (2020) applied blockchain to exchange project files (e.g., drawings and schedules in DWG and PDF formats). Similarly, Tao et al. (2021) introduced a distributed BIM file storage framework supported by blockchain and IPFS technologies. Celik et al. (2023a) utilized blockchain to record metadata of BIM objects and establish the information priority and provenance. Wang et al. (2020) implemented blockchain to improve information scheduling and traceability for supply chain information management. Notably, existing studies in this domain have primarily operated at the framework level, often neglecting the challenge of data redundancy. These studies chiefly store BIM file metadata (e.g., hash signatures) on the blockchain, sidestepping the challenge of redundant BIM data. The root of data redundancy lies in the inherent conflict between the centralized nature of BIM and the distributed nature of blockchain. A multitude of redundant data is prevalent within a BIM

model, and the replicated data storage mechanism inherent in blockchain further exacerbates this issue.

Semantic changes in BIM recently emerged as a new trend in the field of open BIM exchange, rather than concentrating solely on compressing individual IFC models by eliminating duplicated instances through dynamic STEP #-IDs (Gui et al., 2019). Xue and Lu (2020) proposed the SDT model to capture incremental semantic changes in a series of IFC models. The advantages of documenting semantic changes rest upon two aspects: (1) BIM inherently follows an incremental model that lends itself to representation through semantic changes, aligning more closely with industrial practices; (2) recording differences circumvents the impact of inconsequential byte-level changes stemming from repetitive import and export of IFC models across BIM software programs. Researchers, such as Lee et al. (2021a) and Pradeep et al. (2021), also discussed the potential usage of this conceptual model. Subsequently, Wang et al. (2022a) have advanced the proof-of-concept work for multi-person collaborative design. However, the existing methods for BIM incremental changes encounter several limitations: (i) inefficient computation due to the $O(n^2)$ complexity in solving the longest common subsequence problem between IFC entities across versions (Xue and Lu, 2020); (ii) unsatisfactory redundancy minimization due to IFC entities (such as *IfcOpeningElement* and *IfcShapeRepresentation*) generated by the black-box conversion by commercial BIM software (Solihin et al. 2015); and (iii) lack of semantics traceability for BIM objects for construction engineering management context. The three theoretical defects hinder exchanging semantic changes of BIM on the real projects’ application.

In summary, the distributed exchange paradigm offers the best traceability and security, and higher data redundancy among the three exchange paradigms. However, current research on utilizing blockchain for BIM data exchange remains at the theoretical or proof-of-concept stage due to data redundancy and limited semantic traceability. Thus, the minimization of data redundancy and maximization of semantic traceability, which are the critical enablers to the distributed BIM data exchange paradigm, can be a crucial breakthrough to advance open BIM exchange in theory. Furthermore, Blockchain 3.0 is a promising and user-friendly technology for BIM practitioners in construction engineering management.

3 The conceptual framework

The conceptual framework of this paper is shown in Fig. 1. The framework consists of three layers from top to bottom: (1) the application layer, (2) the tSDT approach, and (3) the Blockchain 3.0 framework. The uppermost application layer, which hosts a web-based DApp named openBIMdisk, connects to the distributed BIM exchange.

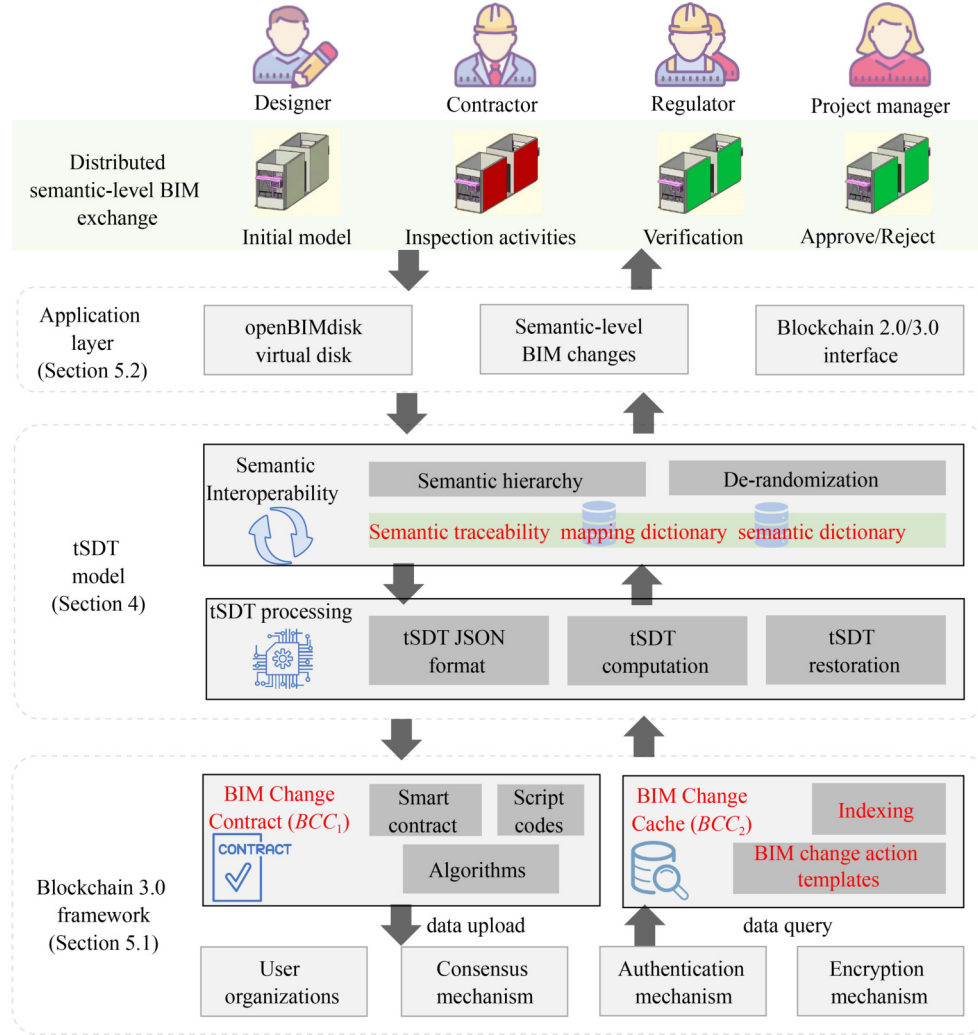


Fig. 1 Conceptual framework of this paper (newly proposed concept and processes highlighted in red).

In this paper, the second layer is the most sophisticated. The tSDT model layer captures and indexes all the semantically changed BIM objects for blockchain-ready storage and traceability. The third layer plugs in the blockchain backbone with a novel BIM change contract (BCC_1) and BIM change cache (BCC_2).

The proposed framework of this paper revolves the SDT model in two aspects, i.e., BIM change model and blockchain-based exchange. Regarding the BIM change model, our tSDT model resolves, wholly or partially, the three theoretical defects of its precedent SDT model described in Section 2.3. The key innovation of tSDT includes two new mapping dictionaries between IFC-GUIDs, IFCJSON-UUIDs, and tSDT-UUIDs that generated through semantic traceability, and a novel two-step semantic traceability operation to systematize changed semantic identifiers. On the blockchain-based exchange, our novel blockchain 3.0 framework partially response to the traceability defects of its precedent SDT model. The novel BCC_2 contains a locally cached database for BIM change model and communicates with the general

blockchain backbone. Through integrating the novel BCC_1 and BCC_2 with blockchain backbone, the Blockchain 3.0 framework enables swift traceability of both on-chain and off-chain records of the BIM change model.

4 The tSDT model

4.1 The semantic interoperability

The semantic interoperability involves pre-processing IFC files for in-depth semantic analysis and computation. The IFC files undergo a systematic hierarchy to streamline IFC semantics. Two dictionaries are then constructed to map IFC elements among IFC-GUIDs, IFCJSON-UUIDs, and tSDT-UUIDs. Three main operations, namely semantic hierarchy, de-randomization, and semantic traceability, are employed, as shown in Fig. 1. This section focuses on the newly introduced semantic traceability, while briefly discussing the other two operations

inherited from SDT to ensure model consistency and clarity.

4.1.1 Semantic hierarchy

The semantic hierarchy operation processes STEP or JSON encoded IFC files with '.ifc' or '.ifcjson' extensions. The output is a structural tree-like objects dictionary that illustrates the relationships between rooted objects (such as *IfcWindow*) and their non-rooted properties and materials. This organization and ordering of the original cluttered one-to-many relationships between IFC objects enhance clarity and understanding.

4.1.2 De-randomization

The de-randomization operation aims to replace all randomly generated IFCJSON-UUIDs with constant tSDT-UUIDs based on attributes or content to ensure consistency. First, a group of entities that have specific attributes (such as names, tags, and object types) are selected, and their UUIDs are replaced by integrating the attributes string. If the required attributes cannot be found in object instances, such as *IfcPropertySets* that describe the non-geometric properties of building components, a hashing function is used to create a short content-based string. This hashing function, commonly used in blockchain technology, efficiently generates constant tSDT-UUIDs. Additionally, the references for the de-randomized objects are updated.

4.1.3 Semantic traceability

The semantic traceability feature is designed to track the changes of IFC-UUIDs during the tSDT processing. As depicted in Fig. 2, semantic traceability consists of two separate steps. The first step creates a one-to-one mapping dictionary (D_1) between IFC-UUIDs and IFCJ-

SON-UUIDs, while it also converts the IFC file into an IFCJSON format. Then, during the streamlining of the semantic hierarchy of IFCJSON objects, the semantic traceability operation records the one-to-multiple mapping (D_2) between IFCJSON-UUIDs and tSDT-UUIDs. This mapping relationship between original IFC objects and semantic hierarchy objects enables semantic traceability. Furthermore, the one-to-multiple mapping dictionary between IFCJSON-UUIDs and tSDT-UUIDs allows for the elimination of redundant semantics.

Redundant semantics in IFC refer to objects that have identical contents but different IFC-UUIDs. In an IFC file, object instances of the same type often have repetitive geometry presentations, properties, and relationships. For example, in Autodesk Revit, families represent object types of IFC and can be reused to create multiple physical building components. Although the family instances have identical shapes, materials, and other properties, the exported IFC files contain similar contents with different IFC-UUIDs. In an actual construction project, such similar objects can occupy up to 30% of the size of IFC files, reducing the efficiency of information exchange. Through the de-randomization and semantic traceability processing, IFC objects are updated to have identical IFC-UUIDs, and tSDT-UUIDs are updated based on the content. The mapping dictionary stores a one-to-multiple structure as '{tSDT-UUID: [IFC-UUID 1, IFC-UUID 2, ...]}'. Based on this mapping relationship, only the first IFC-UUID with index 0 is retained, and object instances containing the remaining listed IFC-UUIDs are removed. The complexity of this algorithm is $O(n \log n)$ due to binary research.

4.2 tSDT processing

The tSDT processing facilitates the bi-directional computation of semantic hierarchies to capture and restore semantic differences to IFC. Figure 3 demonstrates that this bi-directional computation is accomplished through

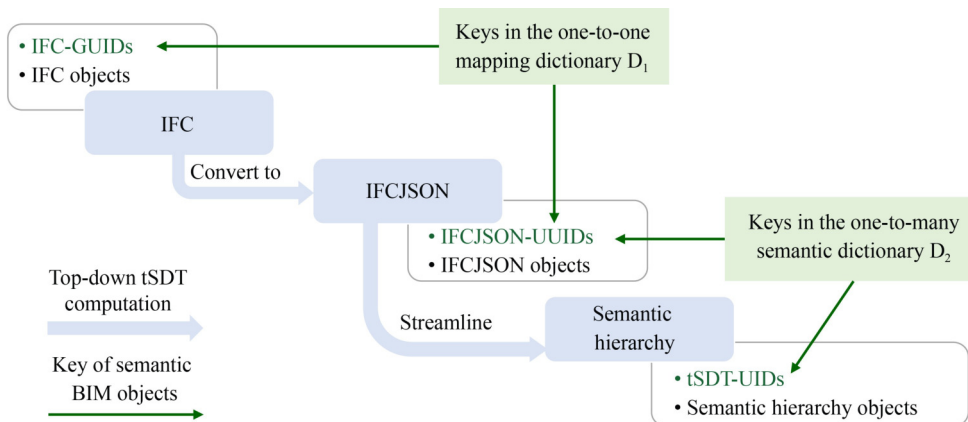


Fig. 2 Two steps of the semantic traceability operation.

the tSDT computation function and the tSDT restoration function. Initially, the trimmed semantic hierarchy (i.e., ifc_0 and ifc_1) is inputted into two tree objects (i.e., θ_0 and θ_1) through the semantic interoperability pre-processing. As a result, both θ_0 and θ_1 are devoid of extraneous data such as STEP line numbers and IFC-GUIDs. Subsequently, the common sections of the two objects (θ^*) are computed, and the modified instances (i.e., θ_{0c} and θ_{1c}) are retained after removing the intersecting parts from the entire semantic hierarchy. Finally, the tree-comparison algorithms calculate the difference between the modified parts (i.e., Δ_θ). As the BIM evolves incrementally, a series of tSDT results are stored in the proposed system in a chain structure as $(ifc_0, \Delta_{\theta_1}, \Delta_{\theta_2}, \dots, \Delta_{\theta_t})$, where ifc_0 denotes the base model and Δ_{θ_t} represents the disparity between two semantic hierarchies at time t .

The tSDT restoration function ensures that the computed semantic differences are restored back to IFC. Initially, the missing reference objects in the tSDT results are added based on the mapping dictionary of IFCJSON-UUIDs and tSDT-UUIDs, which is generated by the semantic traceability operation in Section 4.1.3 (Semantic traceability). Subsequently, the semantic hierarchy at a specific time can be retrieved by “adding” the integrated tSDT results (Δ^*). Finally, the IFC in STEP format is generated by converting the IFCJSON to IFC.

Moreover, the tSDT processing indexes two mapping dictionaries for semantic traceability. To realize the dictionaries, a local relational database is designed. Figure 4 illustrates how example traceability-related attributes are stored in the relational model. The relational model keeps a record of BIM changes with keys such as the transaction ID ($transID$), $tSDT-UID$, and transaction time ($transTime$). One attribute of particular importance is the ‘ $isRVToObj$ ’, which is a Boolean data type used to determine if an IFC entity is a building component within Revit, commonly known as a ‘Revit Object’. This distinction is crucial, as a ‘Revit Object’ possesses an internal ID that enables easy querying of building components. When the value of ‘ $isRVToObj$ ’ is false, the ‘ $relationGUID$ ’ represents the reference Revit Object. For example, for an *IfcPropertySet*, the ‘ $relationGUID$ ’ refers to its related building component, such as an *IfcDoor*.

The indexed data are housed in the local database, as depicted in Fig. 4(a), and are connected to the blockchain network via the transaction ID ($transID$), as shown in Fig. 4(b). This seamless integration enables efficient access to and utilization of the indexed data within the blockchain context.

5 The openBIMdisk virtual disk

The proposed openBIMdisk implements the tSDT approach based on the Blockchain 3.0 framework. As depicted in Fig. 1, our Blockchain 3.0 framework enhances the conventional Blockchain 2.0 backbone with the BCC_2 . The cache BCC_2 enables offline data storage, synchronization, and query. Additionally, the openBIMdisk aims to provide user-friendly interfaces for BIM project stakeholders to exchange BIM files and semantic-level changes over indirect interactions with the Blockchain 3.0. Thus, the Blockchain 3.0 framework can supplement effective and secure BIM exchange functions, by resolving the third challenges summarized in Section 2.

5.1 A Blockchain 3.0 framework

This paper extends the Blockchain 3.0 framework proposed by Zhao et al. (2023), to address issues related to Internet connectivity and response time of querying large-scale BIM files. The primary objective is to enhance the synchronization of open BIM changes computed by tSDT. Figure 5 conceptualizes the components in the Blockchain 3.0 framework. The framework consists of two layers, i.e., the cached layer and the data adaptor layer. The cached layer facilitates offline data storage and offers indexing, querying, and analysis, for front-end user functions and back-end data synchronization. In comparison, the data adaptor layer aims to support the cached layer to Blockchain 2.0 (*Hyperledger Fabric* ver. 2.5), such as automatic off-chain data synchronization when an active Internet connection is re-established. The Blockchain 3.0 framework ensures

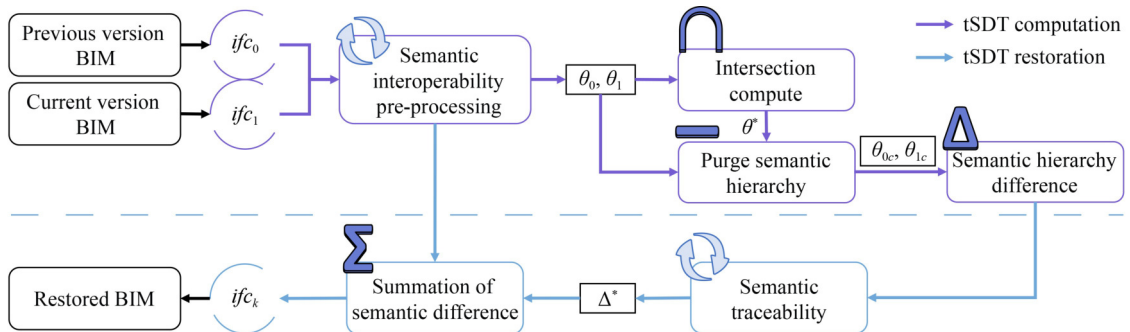


Fig. 3 Procedures of the bi-directional computation of semantic hierarchy.

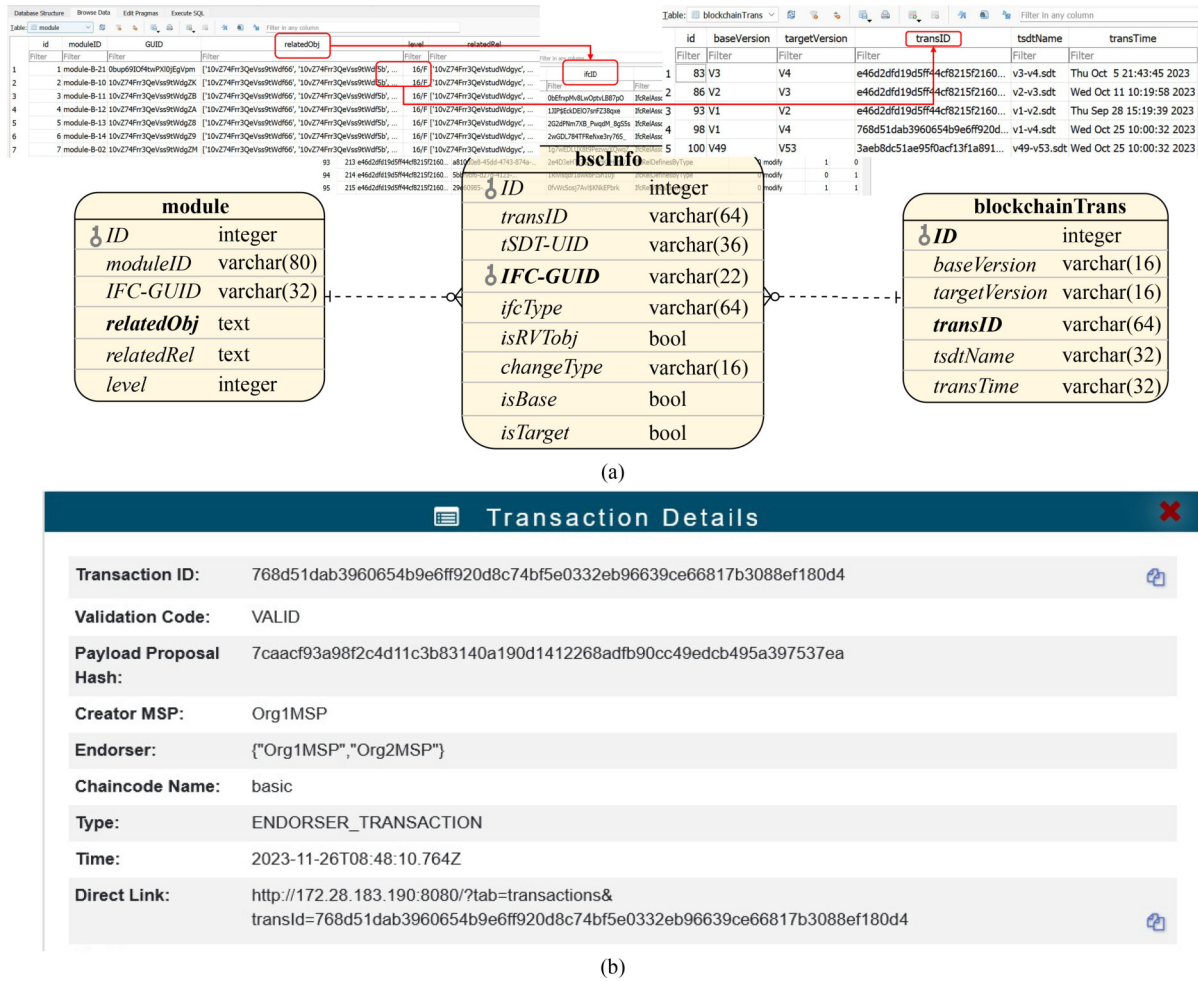


Fig. 4 An example of the relational model between the indexed offline data and on-chain transactions: (a) local indexing of BIM changes for exchange; (b) transactions in the blockchain network.

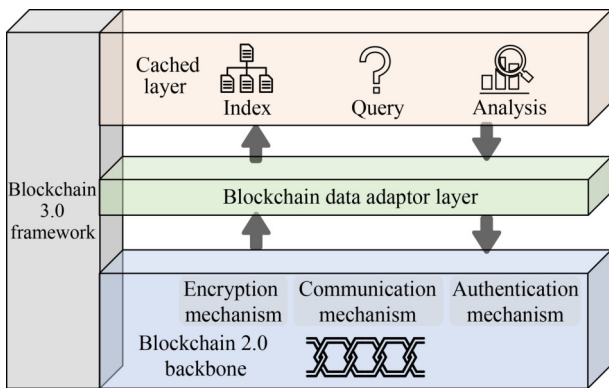


Fig. 5 Structure of the adopted Blockchain 3.0 backbone (Zhao et al., 2023).

smooth data transition between offline and online environments.

With the cached layer, BIM exchange activities are cached as transactions in the BIM change contract (BCC_1). Each transaction in BCC_1 , besides containing a timestamp, consists of the following six components:

- **BIM exchange channel:** A permissioned channel for project BIM access control,
- **User ID:** A unique ID for BIM stakeholders who submit the transaction,
- **Change action:** Boolean value, either `tSDT_compute` or `tSDT_restore`,
- **BIM base version:** The version of the base (or a milestone version) BIM,
- **BIM target version:** Target BIM version for semantic differentials or restorations,
- **File information:** Semi-structured data of IFC file details, including encoded tSDT results, file path, size, and type, and
- **Timestamp.**

Table 3 provides two sample transactions, illustrating the information contained in BCC_1 . When the change action is `tSDT_compute`, the file information includes the Base64 encoded tSDT results. However, if the action is `tSDT_restore`, only the metadata of the resulting IFC files is included, given the large size of IFC files that exceeds the block capacity.

Figure 6 illustrates the workflows of the proposed Blockchain 3.0 framework for semantic BIM exchange. The workflow begins with a user initiating a request to upload or query data for semantic-level BIM exchanges. The request is then validated and preprocessed by BCC_1 . If the request aligns with the specified data format in the BCC_1 and is authorized, it must be signed and submitted by the authorized user. This initiates a two-step process involving interactions with the Blockchain 3.0 cache and the Blockchain 2.0 backbone. Initially, BCC_1 autonomously forwards the request data to BCC_2 . Once the request is received, BCC_2 confirms the request and indexes the data in preparation for communication with the Blockchain 2.0 backbone. If the network is accessible, BCC_2 passes on the request, impacting the state of the Blockchain 2.0 network. Subsequently, the resulting

outcome is transmitted back as a second response, accompanied by the updated transaction hash.

5.2 Application layer

A web-based DApp openBIMdisk is designed with GUI to facilitate BIM authors and practitioners. Figure 7 displays the sketch designs of the GUI of openBIMdisk. Figure 7(a) presents the front page of the DApp, guiding users to three essential pages: tSDT-powered BIM change management (including computation and restoration), tSDT-powered BIM change searching, and blockchain backbone information.

Figure 7(b) provides an overview of the conceptual GUI for the tSDT processing functions. The main functions for BIM stakeholders include uploading IFC or IFCJSON

Table 3 Example transactions of BIM change contract (BCC_1)

BIM exchange channel	User ID	Change action	BIM base version	BIM target version	File information (semi-structured)
Channel A	ID001	tSDT_compute	V0	V1	{"sdt": "[Base64_text]", "url": "..."} {"url": "...", "fileName": "V1.ifc", "fileSize": "135MB"}
Channel A	ID002	tSDT_restore	V4	V1	

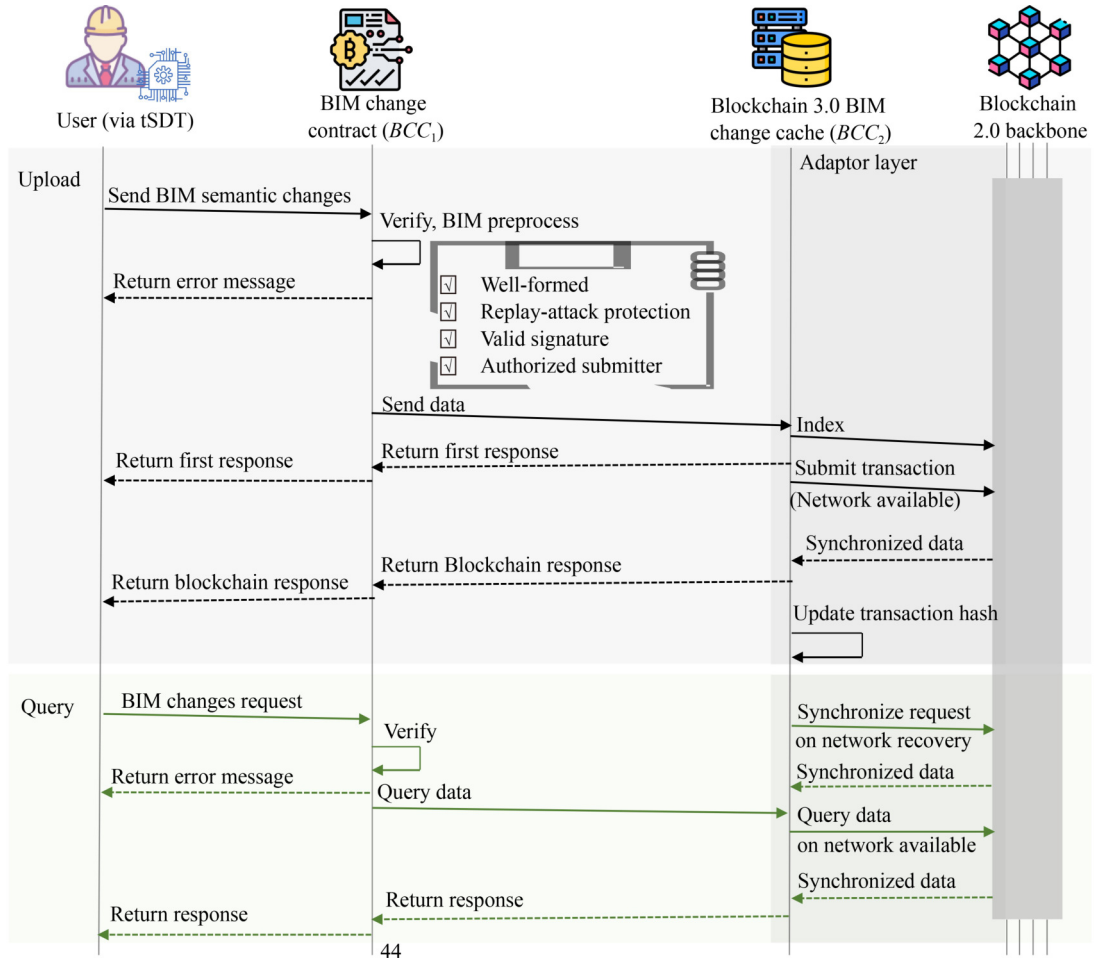


Fig. 6 Workflow of the adopted Blockchain 3.0 framework.

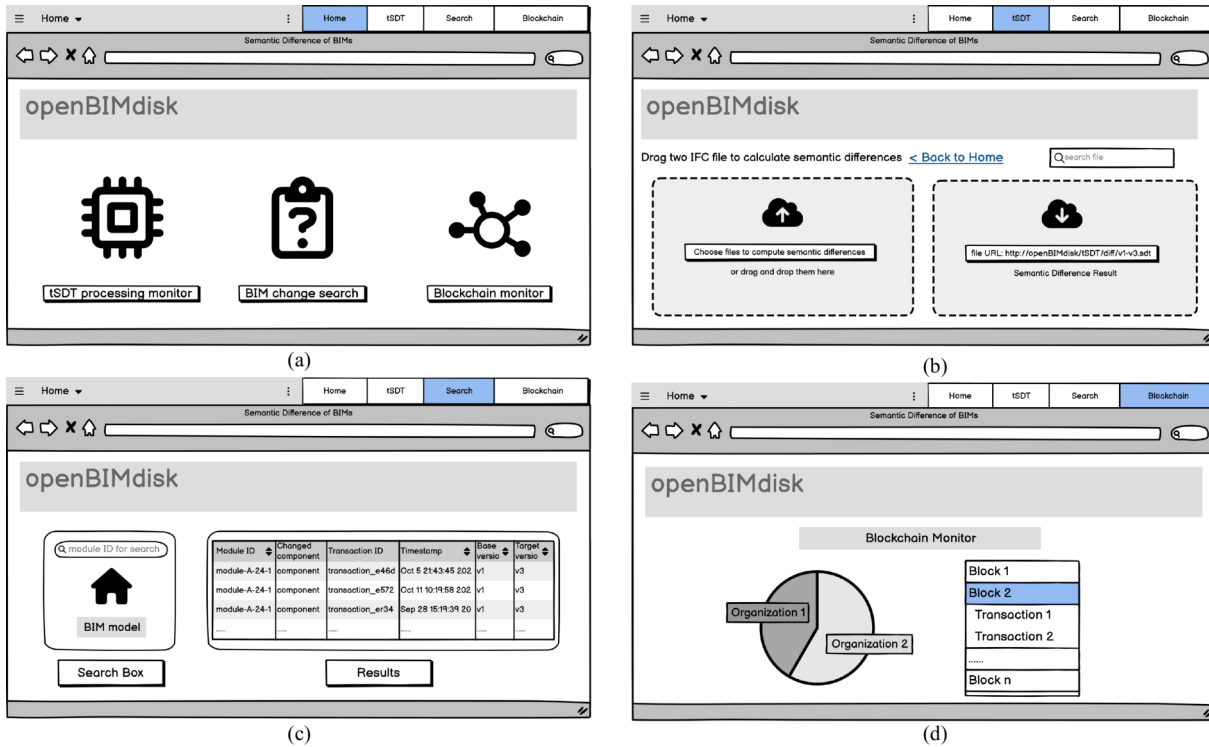


Fig. 7 Sketch design of graphical interfaces of the openBIMdisk: (a) Front page; (b) tSDT-powered BIM changes calculation and storage; (c) tSDT-powered BIM changes searching; (d) blockchain backbone information.

files, automating the tSDT process to capture semantic differences, and restoring the historical BIM version given a specific date. Users can also download the BIM changes as tSDT files. Figure 4 illustrates that the outcomes of the processing steps are initially stored in the local virtual disk. Subsequently, the indexed data is saved into the cached database from the virtual disk. Ultimately, the semantic-level BIM changes are automatically decoded and uploaded as transactions to the blockchain, contingent upon Internet availability.

Figure 7(c) displays the functionality provided for users to trace semantic-level BIM changes. Within the BIM model, a search box allows users to query the semantic changes for specific building modules by either selecting the module in the BIM model or entering the module's name. The search results are presented in a table. Furthermore, BIM stakeholders can review on-chain records through the Blockchain monitor page, as depicted in Fig. 7(d). The trace function of openBIMdisk enables users to query semantic-level BIM changes independently from the Internet, ensuring unaffected by information delays.

5.3 Software implementation

The development of openBIMdisk adhered to the proposed framework presented in Fig. 1. The bottom Blockchain 3.0 framework was internally developed based on *Hyperledger Fabric* (ver. 2.5). BCC_1 was

deployed in JavaScript, and BCC_2 was realized through a local relational database (SQLite, ver. 3.43). The tSDT model was implemented using Python3, compatible with IFCJSON standards up to ver. 4. Example screenshots of the four interfaces of openBIMdisk developed on the Flask web application framework (ver. 2.2) and Python3 are shown in Fig. 8.

A benchmark test was conducted to compare the performance of two popular blockchain frameworks: Hyperledger Fabric and FISCO BCOS (ver. 3.6). The test configuration is presented in Fig. 9, where Fig. 9(a) demonstrates the Hyperledger Caliper configuration for Fabric testing, and Fig. 9(b) illustrates the Java SDK testing toolkit configuration for FISCO BCOS. As Hyperledger Caliper does not support benchmark testing for FISCO BCOS version 3.6, the official Java SDK testing toolkit for FISCO BCOS was utilized. The transaction count was set to 300, and the number of worker processes was set to 3. The comparison results are listed in Table 4. Notably, BCOS only required 27.6% of the time taken by Fabric to process 300 transactions. However, Fabric outperforms in terms of usability, as it supports multiple languages, such as Go, Java, Node.js, for compiling smart contracts. Additionally, Fabric was initially designed as a cross-industry application, aiming to build a common framework for different domains. FISCO BCOS performs worse than Fabric in terms of usability and cross-industry applications. FISCO BCOS primarily supports the Solidity language for smart contracts and is mainly utilized in the

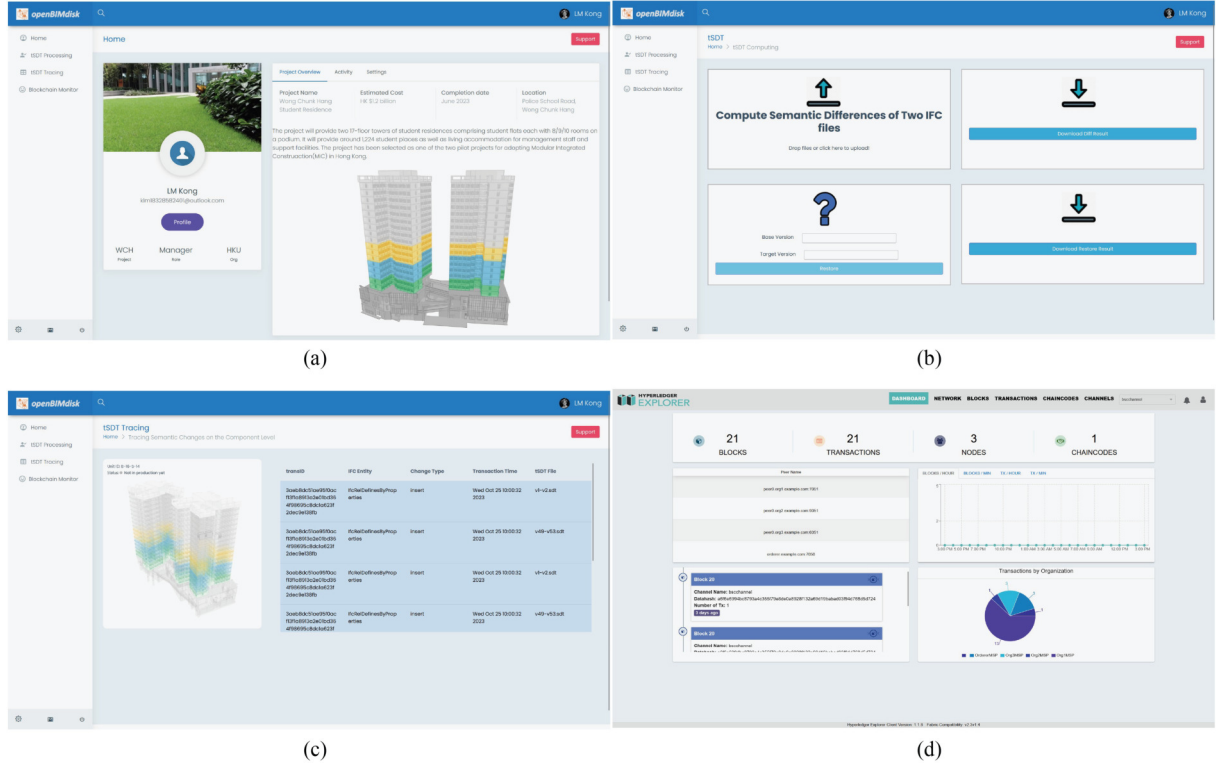


Fig. 8 Screenshots of the implemented openBIMdisk virtual disk: (a) Home page with project brief; (b) tSDT-powered BIM changes page; (c) Semantic BIM change query results and (d) Information of blockchain storage.

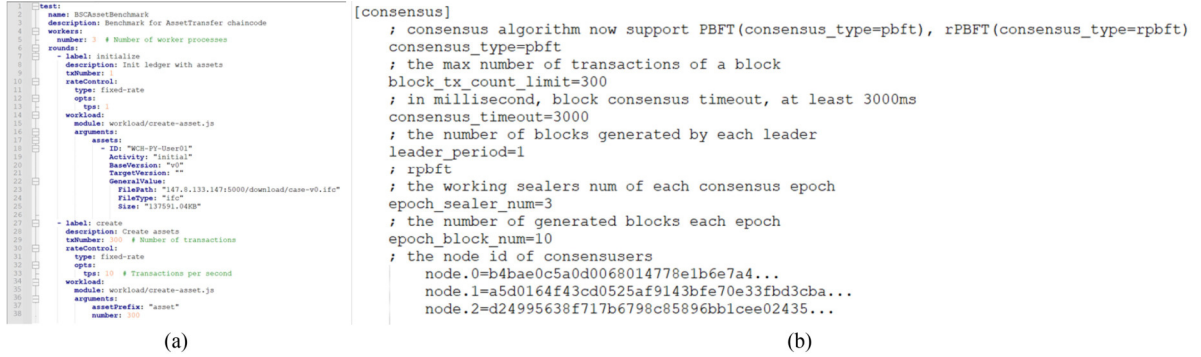


Fig. 9 Configurations of the benchmark test for (a) Fabric by Caliper, (b) FISCO-BCOS by Java SDK.

Table 4 Comparison between Fabric and FISCO-BCOS (best value in each row in bold).

Metric	Fabric	FISCO BCOS
Time cost (ms)	115.38	31.89
Usability	High	Middle
Application scenarios	Cross-industry applications	Mainly for financial industry

financial industry. Since BIM data exchanges are not as frequent as financial transactions, the time cost of Fabric is deemed acceptable. As a result, Hyperledger Fabric is employed as the foundation of the Blockchain 2.0 network in this study.

6 A case study

6.1 Experimental settings

A case study was conducted to demonstrate the functionality of tSDT and openBIMdisk using a modular construction project consisting of two 19-level buildings, as shown in Fig. 10. The BIM model in Fig. 10(a) was created using Autodesk Revit 2019 and exported as the IFC2x3 Coordination View 3.0 schema. The decision to use IFC 2x3 instead of the more recent IFC4 was due to limited support for IFC4 in Revit 2019. Additionally, IFC4 partitions the full BIM semantics into two

components, the Reference View and Design Transfer View, which could introduce additional variables to control in the experiments. The BIM model went through daily versioning, resulting in a total of 129 versions ranging from ver. 1 to ver. 313, with skips on non-progressive days. The size of the input IFC files varied from 134 MB to 136 MB across the 129 different versions.

In Fig. 10(b), the BIM exchange was required to synchronize daily inspections of modular building components manufactured in distributed factories. This collaborative environment involved three key BIM stakeholders: a contractor, a regulator, and a project manager. The contractor initiated the process by uploading daily inspection data, including inspection tasks and photographic records. Subsequently, the regulator and project manager sequentially reviewed and authorized the data. The interactive manipulations resulted in local semantic changes to the BIM model before reaching blockchain consensus. Finally, the validated results were sent back to the contractor, while the global semantic changes were updated in the openBIMdisk with the blockchain consensus. The collaboration process involved multiple interactive stages among the BIM stakeholders, making the determination of BIM versions crucial. In the tests, a new BIM version was defined daily, indicating that the BIM version changed after daytime inspection and authorization work.

Figure 11 depicts two pilot tests conducted in this paper. The first test, a comparison test against the SDT,

was designed to evaluate the progress achieved by the tSDT approach. This test involved 30 different versions of BIM. The process began by applying both the tSDT and SDT methods to the IFC files. The results were then compared based on various factors, including computation time, file size, and the level of BIM semantic redundancy. The second test, a scale-up test of openBIMdisk, served two purposes. The first purpose is to assess the feasibility of semantic-level BIM exchange and historical version recovery using 129 versions of the BIM from the project record. The other purpose is to evaluate the efficiency and traceability of semantic-level BIM changes in the openBIMdisk virtual disk.

The deployed consortium blockchain network consists of the following components: (1) one channel named “bscchannel” (short for BIM semantic changes channel); (2) three organizations - *Org1* representing the contractor company, *Org2* representing regulator representatives, and *Org3* representing owner organizations; (3) three peer nodes, with one peer node assigned to each organization. The configuration of the blockchain is illustrated in Fig. 12. Figure 12(a) shows the three organizations and their corresponding peer nodes. Figure 12(b) outlines the network settings, including the number of messages batched into a block and the endorsement policy. Figure 12(c) provides details on the channel configuration, including the organizations and profiles configuration.

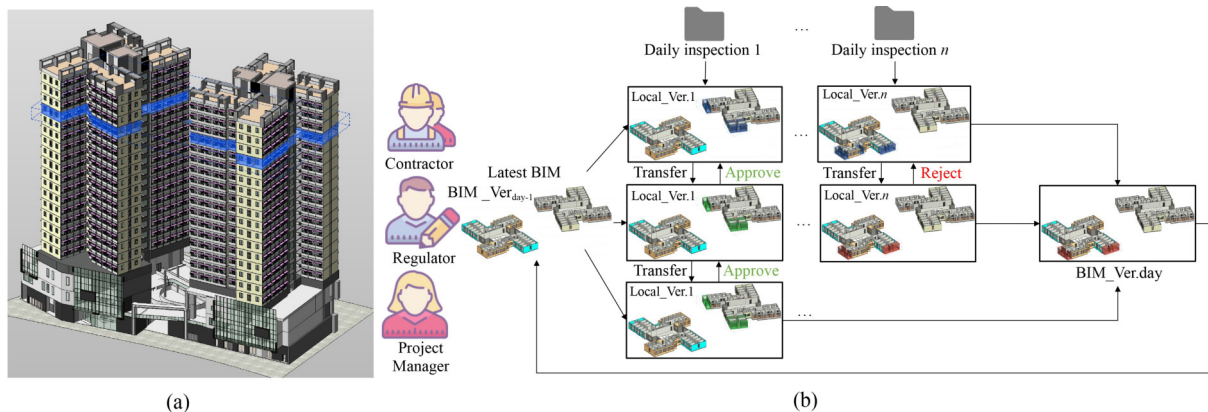


Fig. 10 A case BIM project for a modular construction: (a) 3D view of the BIM; (b) daily BIM exchange scenario for modular components inspection.

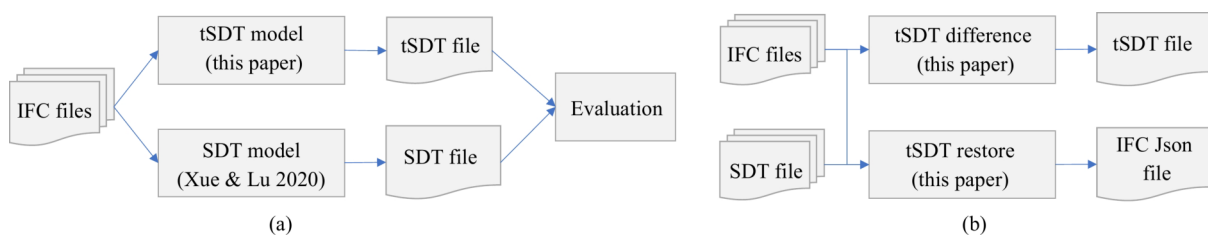


Fig. 11 Two test cases: (a) a performance test of tSDT model; (b) A scale-up test of openBIMdisk.

Ledger Height						
Peer Name	Request Uri	Peer Type	MSPID	High	Low	Unsigned
peer0.org1.example.com...	peer0.org1.example.com...	PEER	Org1MSP	0	106	true
peer0.org2.example.com...	peer0.org2.example.com...	PEER	Org2MSP	0	106	true
peer0.org3.example.com...	peer0.org3.example.com...	PEER	Org3MSP	0	106	true
orderer.example.com:7050	orderer.example.com:7050	ORDERER	OrdererMSP	-	-	-

(a)

```

1 Orderer: &OrdererDefaults
2 OrdererType: etcdraft
3 Addresses:
4   - orderer.example.com:7050
5
6 EtcdRaft:
7   Consenters:
8     - Host: orderer.example.com
9       Port: 7050
10    ClientTLSCert: ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.crt
11    ServerTLSKey: ../organizations/ordererOrganizations/example.com/orderers/orderer.example.com/tls/server.key
12
13 BatchTimeout: 2s
14
15 BatchSize:
16   MaxMessageCount: 10
17   AbsoluteMaxBytes: 99 MB
18   PreferredMaxBytes: 512 KB
19
20 Organizations:
21
22 Policies:
23   Readers:
24     Type: ImplicitMeta
25     Rule: "ANY Readers"
26   Writers:
27     Type: ImplicitMeta
28     Rule: "ANY Writers"
29   Admins:
30     Type: ImplicitMeta
31     Rule: "MAJORITY Admins"
32
33 BlockValidation:
34   Type: ImplicitMeta
35   Rule: "ANY Writers"

```

(b)

```

1 version: '3.0'
2
3 volumes:
4   - orderer.example.com:
5     peer0.org1.example.com:
6     peer0.org2.example.com:
7     peer0.org3.example.com:
8
9 networks:
10   - orderer.example.com:
11     peer0.org1.example.com:
12     container_name: peer0.org1.example.com
13     image: hyperledger/fabric-peer:latest
14     labels:
15       - service: hyperledger-fabric
16     environment:
17       - FABRIC_CFG_PATH=/etc/hyperledger/peerconfig
18       - FABRIC_LOGGING_SPEC=INFO
19       - FABRIC_LOGGING_SPEC=DEBUG
20       - CORE_PEER_TLS_ENABLED=true
21       - CORE_PEER_PROFILE_ENABLED=false
22       - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt
23       - CORE_PEER_TLS_KEY_FILE=/etc/hyperledger/fabric/tls/server.key
24       - CORE_PEER_TLS_ROOTCERT_FILE=/etc/hyperledger/fabric/tls/ca.crt
25     # Peer specific variables
26     - CORE_PEER_ID=peer0.org1.example.com
27     - CORE_PEER_ADDRESS=peer0.org1.example.com:7051
28     - CORE_PEER_LISTENADDRESS=0.0.0.0:7051
29     - CORE_PEER_CHAINCODEADDRESS=peer0.org1.example.com:7052
30     - CORE_PEER_CHAINCODELISTENADDRESS=0.0.0.0:7052
31     - CORE_PEER_GOSSIP_BOOTSTRAP=peer0.org1.example.com:7051
32     - CORE_PEER_GOSSIP_EXTERNALENDPOINT=peer0.org1.example.com:7051
33     - CORE_PEER_LOCALMSPID=Org1MSP
34     - CORE_PEER_MSPCONFIGPATH=/etc/hyperledger/fabric/msp
35     - CORE_OPERATIONS_LISTENADDRESS=peer0.org1.example.com:9444
36     - CORE_METRICS_PROVIDER=prometheus
37     - CHAINCODE_A_A_SERVICE_BUILDER_CONFIG={"peername":
38       "peer0.org1"}
39     - CORE_CHAINCODE_EXECUTETIMEOUT=300s
40
41 volumes:
42   - ../organizations/peerOrganizations/org1.example.com/peers/peer0.org1.example.com:/etc/hyperledger/fabric
43   - peer0.org1.example.com:/var/hyperledger/production
44
45 working_dir: /root
46 command: peer node start
47
48 ports:
49   - 7051:7051
50   - 9444:9444
51
52 networks:
53   - test
54
55 peer0.org2.example.com:
56   container_name: peer0.org2.example.com
57   image: hyperledger/fabric-peer:latest
58   labels:
59     - service: hyperledger-fabric
60   environment:
61     - FABRIC_CFG_PATH=/etc/hyperledger/peerconfig
62     - FABRIC_LOGGING_SPEC=INFO
63     - FABRIC_LOGGING_SPEC=DEBUG
64     - CORE_PEER_TLS_ENABLED=true
65     - CORE_PEER_PROFILE_ENABLED=false
66     - CORE_PEER_TLS_CERT_FILE=/etc/hyperledger/fabric/tls/server.crt

```

(c)

Fig. 12 Configuration of the developed consortium blockchain network: (a) Organizations and peer nodes; (b) settings of the network; (c) channel configuration.

6.2 Experimental results

6.2.1 Results of the performance test

For each IFC model, two mapping dictionaries of semantic identifiers are generated: (1) the one-to-one mapping dictionary of IFC-GUIDs and IFCJSON-UUIDs; and (2) the one-to-multiple semantic dictionary of IFCJSON-UUIDs and tSDT-UUIDs. The average size of the two mapping dictionaries is 2.13 MB and 6.3 MB, respectively. The average number of semantic items in the IFC models is 30,322. Partial examples of the two dictionaries are displayed in Fig. 13.

Table 5 presents the average results of the comparative tests conducted to assess the effects of the two new dictionaries. The complete results can be found in the Supplementary material. Over the 30 versions of BIM files, the SDT method yielded an average of 12.3 kB of minimum incremental BIM changes in 325.8 s. The proposed tSDT approach in this paper resulted in 9.7 kB of minimum incremental BIM changes, saving 21.4% more disk space and data transmission than SDT. Furthermore, the tSDT method required an average of 94.0 s, which was 71.1% faster than SDT.

The quantities of *IfcShapeRepresentation* and *IfcOpeningElement* in Table 5 explain the technical details for the considerable improvement done by tSDT. The two classes of IFC entities represent ‘black box’ reported entities. The experimental results demonstrate

that tSDT reduced the average quantity of *IfcShapeRepresentation* by 43.4% and the quantity of *IfcOpeningElements* by 82.1%. As a result, tSDT outperformed SDT in terms of minimal incremental BIM changes.

The effect of the two-step semantic traceability operation was also verified. Figure 14 illustrates the redundancy in SDT’s output using an example. SDT captured two shape representations that describe two window opening elements, where all properties are identical except for their respective global IDs. The tSDT approach eliminates such redundant semantic changes through the semantic traceability operation.

6.2.2 Results of the scale-up test

Table 6 presents the overall results from the scale-up test, which involved 129 versions of BIM files. The average computation and restoration time costs for tSDT were both under 100 s, and the average size of tSDT files was $9.64 < 10$ kB. Considering each BIM file was over 130 MB, the captured tSDT results comprised only 0.007% of the IFC file size. Hyperledger Fabric’s officially recommended block size of 512 kB can accommodate around 50 BIM versions per day. Therefore, the tSDT approach scales effectively from laboratory tests to real-world construction projects. Additionally, the results were uploaded and synchronized on the Blockchain 3.0 network, obtaining automatic approval from BIM stakeholders through the BIM change contract.

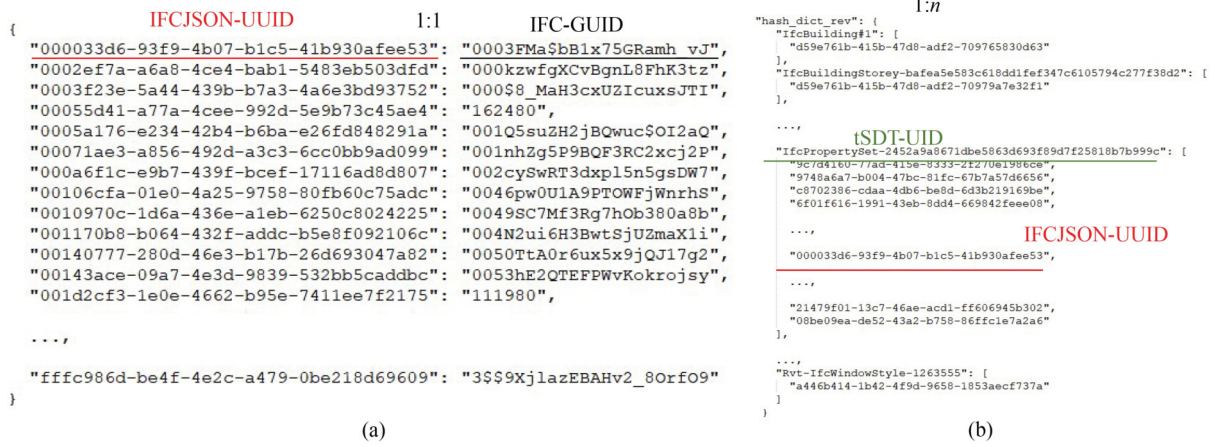


Fig. 13 Partial example of (a) 1-to-1 mapping dictionary of IFC-GUIDs and IFCJSON-UUIDs, and (b) 1-to-multiple semantic dictionary of IFCJSON-UUIDs -UUIDs and tSDT-UUIDs.

Table 5 Comparison of SDT and tSDT on computing the differentials between the 30 versions of BIM files

Performance metric	Average of SDT	Average of tSDT	Improvement
Incremental BIM change (kB)	12.3	9.7	21.4%
Computational time (s)	325.8	94.0	71.1%
No. of <i>IfcShapeRepresentation</i>	39.6	22.4	43.4%
No. of <i>IfcOpeningElements</i>	30.7	5.5	82.1%

6.2.3 Results of object-level BIM semantics traceability

Another experiment conducted queries of semantic BIM changes at the object level, following customary practices of BIM stakeholders. Figure 15 provides an example of the query results for Module 'B-16-S-14', highlighting the target BIM object in the interactive 3D view. The test was based on the observation that inspectors typically focus on changes in prefinished modules rather than individual properties for the pilot project. The average query response time was 5.3 ms, indicating swift response due to the local cached database. Furthermore, the query consistently responded to users' queries irrespective of whether the system was operating offline or online. However, the most recent changes may not have been synchronized to the interface when Blockchain 3.0 was offline. The millisecond-level response time demonstrates the efficiency, traceability, and security of openBIMdisk.

7 Discussion

The findings in this paper makes a twofold contribution, to both theoretical and practical aspects. First, tSDT is a state-of-the-art, distributed BIM change model, which contributes to the knowledge of blockchain BIM and BIM exchange. The key innovation of tSDT lies in two new mapping dictionaries, i.e., IFC-GUIDs, IFCJSON-UUIDs, and tSDT-UUIDs, which are generated through

semantic traceability. The novel two-step semantic traceability operation is introduced to systemize changed semantic identifiers. Experimental results demonstrate that tSDT outperforms its predecessor, SDT, regarding computation efficiency and semantic redundancy elimination. Notably, tSDT facilitates object-level BIM semantic changes traceability through the two mapping dictionaries, thus enabling the industrial application of BIM exchange via blockchain. In the case of blockchain-based exchange, our blockchain 3.0 framework addresses the traceability issues of the preceding SDT model. To be specific, the novel BCC_2 incorporates a locally cached database for the BIM change model and communicates with the general blockchain backbone. Consequently, our Blockchain 3.0 framework ensures swift traceability of both on-chain and off-chain records of the BIM change model. By minimizing BIM semantic changes and maximizing traceability at the object level, our tSDT propels the proof-of-concept of Blockchain BIM data exchange toward realistic project applications.

The second practical contribution is openBIMdisk, a web-based Blockchain 3.0 DApp virtual disk that facilitates intuitive and traceable semantics-level BIM exchange for practitioners. openBIMdisk introduces three significant advancements for industrial practices. First, the Blockchain 3.0 backbone ensures fast traceability of both on-chain and off-chain records of BIM semantic changes through the BIM change cache. Secondly, the BIM change contract deployed on the blockchain network guarantees automatic uploading and concurrent

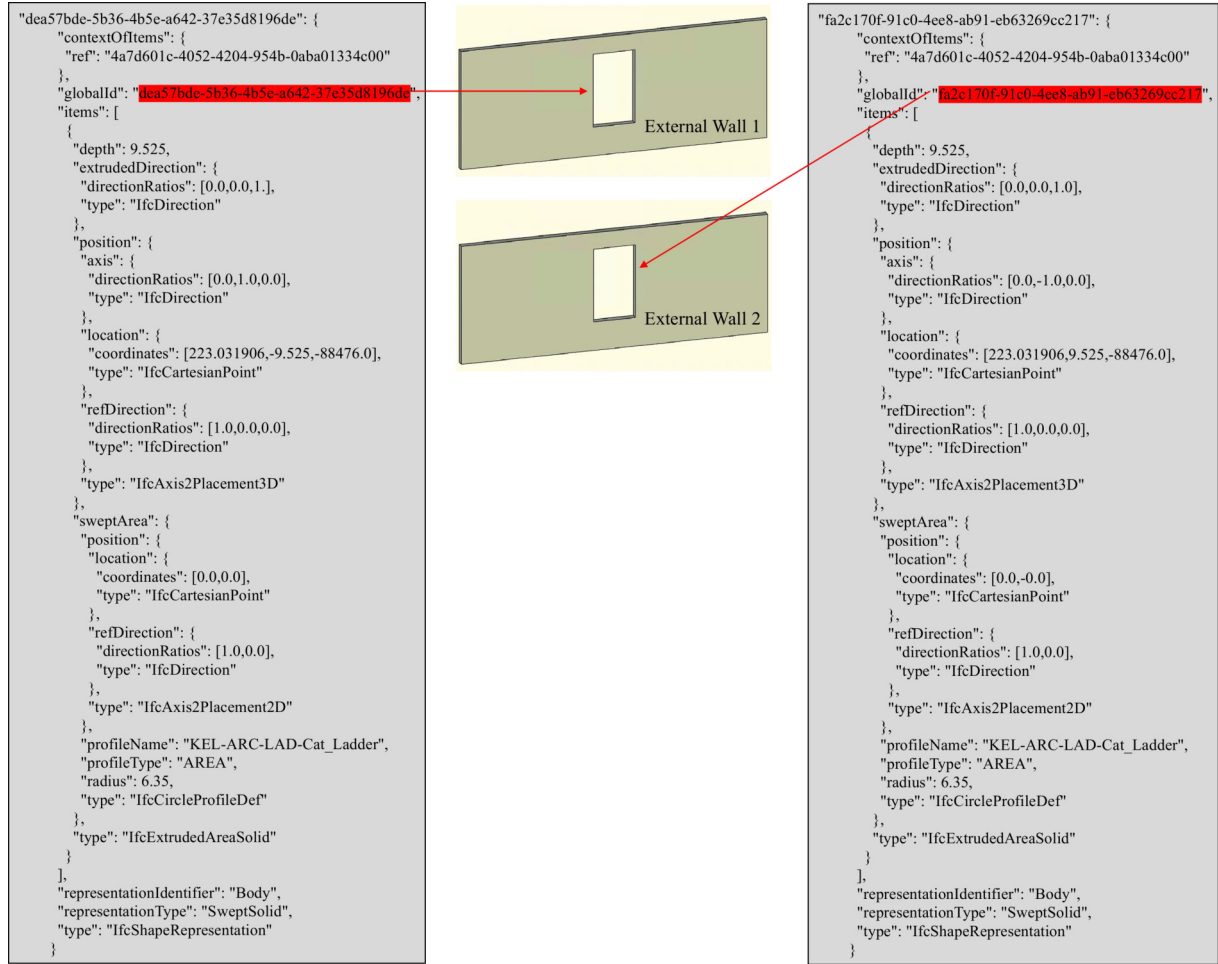


Fig. 14 An example of redundant semantic entities in SDT's results, which is addressed by the proposed tSDT

Table 6 Scale-up test of tSDT on all the versions of BIM files

Item	tSDT computational time (s)	Incremental BIM change (kB on disk)	BIM restoration time (s)
Average	97.32	9.64	89.81
Min	86.13	3.51	75.83
Max	168.48	206	153.34
Std. dev.	7.41	17.54	9.32

approval of BIM changes among all relevant parties. Finally, openBIMdisk offers a concise and user-friendly interface, empowering BIM stakeholders to efficiently engage in semantics-level BIM exchange.

However, this paper does have some limitations. One limitation is that the tSDT captured certain unexpected semantic changes. For example, a *baseSurface* changed unexpectedly from 'firstOperand' to 'secondOperand' (Fig. 16). While tSDT identified this change as a semantic change due to distinct key values, it may be considered redundant if the operands were defined as swappable. Another limitation is that the hashing function used in blockchain and the tSDT approach is content-based. This

means that even a minor byte-level change, such as the length being saved as 1.7999999523 m instead of 1.8 m (due to float precision on computers) in IFC files, can result in completely different hash values for tSDT-UIDs. Therefore, future research could focus on refining the definitions and conversion of IFC semantics.

Furthermore, the paper also has notable limitations related to the pre-processing of IFC files, which was the most time-consuming part of tSDT. For example, openBIMdisk took an average of 16 min in Python3 scripting language to load and transform a 130MB IFC file in the test from STEP format to IFCJSON format. To address this limitation and substantially improve pre-processing, future solutions could consider either multi-threading IFCJSON conversion programs in C++ language or multi-threading export add-ins for native commercial BIM vendor software.

Another limitation of the case study was the limited type of semantic changes considered. The semantic changes mainly reflected inspection records of updates, approvals, and rejections from three stakeholders during the construction stage of the case project. Future

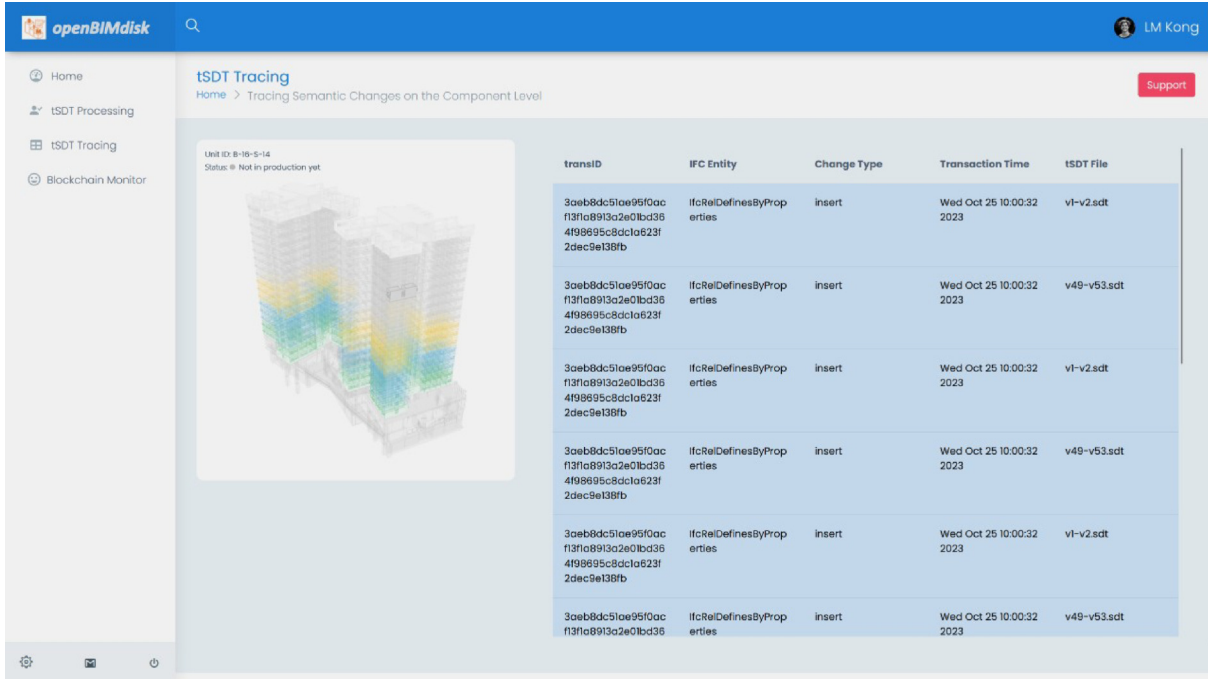


Fig. 15 An example of querying object-level BIM semantic changes of Module “B-16-S-14” (highlighted in the 3D view).

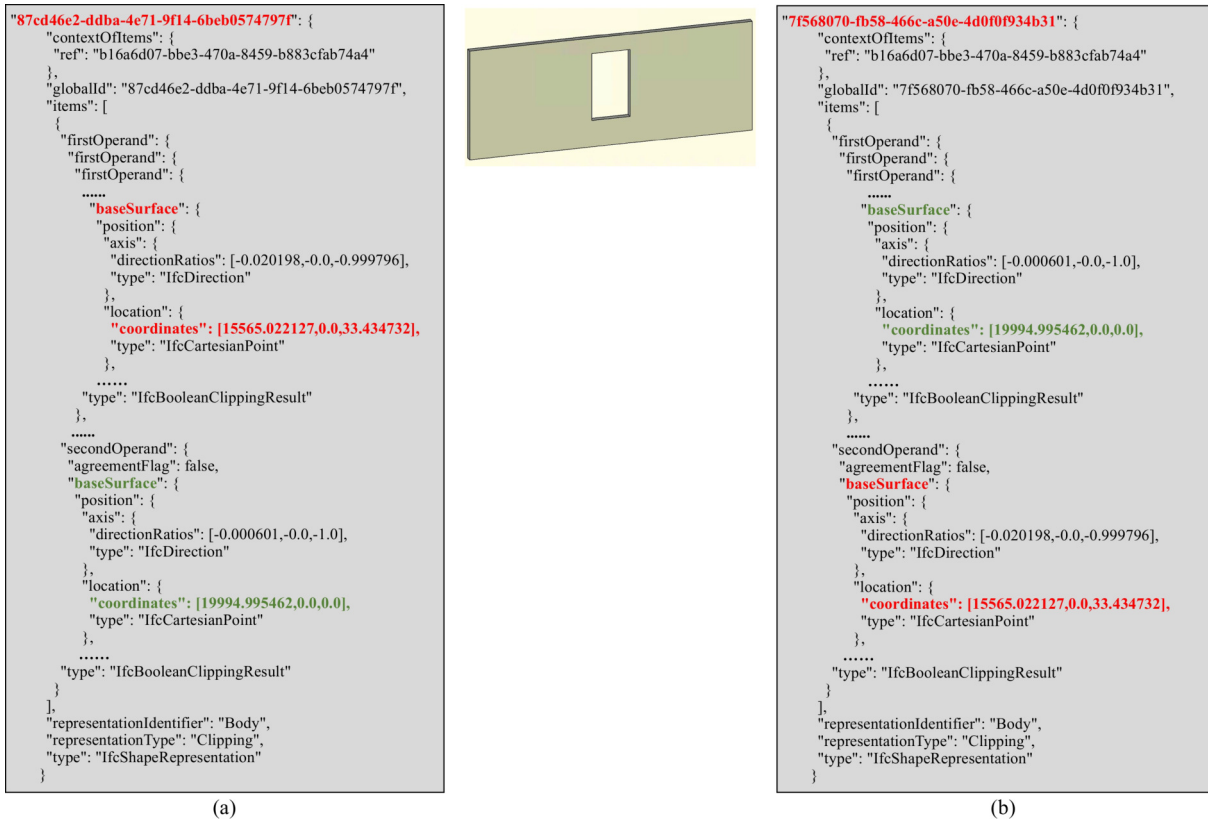


Fig. 16 A redundant semantic changes example: a sequence changes of two base surface in (a) the previous BIM version and (b) the latest BIM version.

researchers may be interested in applying openBIMdisk and tSDT to collaborative BIM design in the design stage (Sacks et al., 2022) and electronic BIM submission in the

vetting stage.

Lastly, openBIMdisk presented in this paper was designed with the essential functionality of a virtual disk

for BIM exchange. Readers and BIM practitioners are suggested to expand and develop new application-specific functions on top of openBIMdisk. Example functions include interpretable machine learning of a project's BIM change patterns (Liang and Xue, 2023) for information extraction (Wang et al., 2022b), BIM project documentation (Singh and Anumba, 2023), BIM automation based on deep learning of semantic changes (Chen and Xue, 2023), and artificial intelligence-based construction management (Yu and Gong, 2024).

8 Conclusions

BIM's importance is increasing in construction engineering management. Integrating BIM and blockchain is a new trend that can benefit efficient, effective, and secure open BIM exchanges for distributed collaboration and BIM versioning. However, two problems of data redundancy and traceability challenges hinder the integration of blockchain and BIM. As the de facto standard for BIM exchange, the IFC format is open to many BIM entities but lacks semantically traceable changes between the versions. The two problems lead to a dilemma when there are overwhelming versions of a BIM project to exchange and track the changes at object level.

This paper proposes a novel tSDT model to resolve the two problems for blockchain BIM integration, by minimizing BIM redundancy and maximizing object-level traceability between versions. The tSDT model utilizes two new mapping dictionaries to enhance traceability and efficiency. To facilitate tSDT applications in industrial settings, this paper develops an open-source DApp named openBIMdisk. The openBIMdisk enables BIM exchange with minimal data redundancy and tracks object-level BIM semantic changes using a Blockchain 3.0 framework and intuitive graphical interfaces.

The experimental results of a case study demonstrated that tSDT successfully stored changes as small as 10 kB (< 0.007% of BIM file size) within a reasonable operation time of 94.01 s between 134 MB BIM files. Additionally, openBIMdisk was validated in real construction projects involving the integration of BIM and blockchain, where object-level BIM semantic changes could be extracted in milliseconds.

This paper also identifies four limitations, which may also highlight future research directions. One future direction is the development of systematic models to address byte-level IFC redundancy and opaque BIM conversion processes. Another notable future research is the creation of multithreading IFC conversion and transformation tools by commercial BIM platforms and the open BIM community. Furthermore, BIM practitioners are suggested to extend the open-sourced tSDT and expand the functionality of openBIMdisk to meet the business demands in design, vetting, and facility

management stages of construction engineering management.

Electronic Supplementary Material Supplementary material is available in the online version of this article at <https://doi.org/10.1007/s42524-024-4006-x> and is accessible for authorized users.

Competing Interests The authors declare that they have no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made.

The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Afsari K, Eastman C, Castro-Lacouture D (2017). JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange. *Automation in Construction*, 77: 24–51
- Afsari K, Eastman C, Shelden D (2016). Cloud-based BIM data transmission: current status and challenges. In: *Proceedings of the 33rd International Symposium on Automation and Robotics in Construction*. IAARC Publications
- Angelis J, Ribeiro da Silva E (2019). Blockchain adoption: A value driver perspective. *Business Horizons*, 62(3): 307–314
- Belotti M, Božić N, Pujolle G, Secci S (2019). A vademecum on blockchain technologies: When, which, and how. *IEEE Communications Surveys and Tutorials*, 21(4): 3796–3838
- Berlo L V, Krijnen T, Tauscher H, Liebich T, Kranenburg A V, Paasiala P (2021). Future of the Industry Foundation Classes: Towards IFC 5. In: *Proceedings of the 38th International Conference of CIB W78*: 123–137. Luxembourg: CIB. Retrieved from the website of scix.net
- Bodkhe U, Tanwar S, Parekh K, Khanpara P, Tyagi S, Kumar N, Alazab M (2020). Blockchain for Industry 4.0: A comprehensive review. *IEEE Access: Practical Innovations, Open Solutions*, 8: 79764–79800
- Boje C, Guerriero A, Kubicki S, Rezgui Y (2020). Towards a semantic Construction Digital Twin: Directions for future research. *Automation in Construction*, 114: 103179
- Borrmann A, Beetz J, Koch C, Liebich T, Muhic S (2018). Industry Foundation Classes: A standardized data model for the vendor-neutral exchange of digital building models. In Borrmann A, König M, Koch C, eds. *Building Information Modeling: Technology Foundations and Industry Practice*. Springer: 81–125
- bSI (2023). *Industry Foundation Classes*. buildingSMART International. Available at the website of buildingsmart.org
- Celik Y, Petri I, Barati M (2023a). Blockchain supported BIM data provenance for construction projects. *Computers in Industry*, 144:

- 103768
- Celik Y, Petri I, Rezgui Y (2023b). Integrating BIM and blockchain across construction lifecycle and supply chains. *Computers in Industry*, 148: 103886
- Chen K (2018). The quest for the holy grail: Bridging BIM and building for information management in construction. HKU Thesis Online (HKUTO)
- Chen K, Zhou X, Bao Z, Skibniewski M J, Fang W (2024). Artificial intelligence in infrastructure construction: A critical review. *Frontiers of Engineering Management*, doi:10.1007/s42524-024-3128-5
- Chen S H, Xue F (2023). Automatic BIM detailing using deep features of 3D views. *Automation in Construction*, 148: 104780
- Das M, Tao X, Cheng J (2021). BIM security: A critical review and recommendations using encryption strategy and blockchain. *Automation in Construction*, 126: 103682
- Deng Y, Gan V J, Das M, Cheng J C, Anumba C (2019). Integrating 4D BIM and GIS for construction supply chain management. *Journal of Construction Engineering and Management*, 145(4): 04019016
- Eastman C, Teicholz P, Sacks R, Liston K (2008). BIM for the construction industry. In: *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*. John Wiley and Sons, Inc
- Erri Pradeep A, Yiu T, Zou Y, Amor R (2021). Blockchain-aided information exchange records for design liability control and improved security. *Automation in Construction*, 126: 103667
- Figueiredo K, Hammad A W, Haddad A, Tam V W (2022). Assessing the usability of blockchain for sustainability: Extending key themes to the construction industry. *Journal of Cleaner Production*, 343: 131047
- Gui N, Wang C, Qiu Z, Gui W, Deconinck G (2019). IFC-Based partial data model retrieval for distributed collaborative design. *Journal of Computing in Civil Engineering*, 33(3): 04019016
- Guo H, Zhou Y, Ye X, Luo Z, Xue F (2020). Automated mapping from an IFC data model to a relational database model. *Journal of Tsinghua University*, 61(2): 152–160 (in Chinese)
- Hakimi O, Liu H, Abudayyeh O (2024). Digital twin-enabled smart facility management: A bibliometric review. *Frontiers of Engineering Management*, 11(1): 32–49
- Hastig G, Sodhi M (2020). Blockchain for supply chain traceability: Business requirements and critical success factors. *Production and Operations Management*, 29(4): 935–954
- Hijazi A, Perera S, Calheiros R, Alashwal A (2021). Rationale for the integration of BIM and blockchain for the construction supply chain data delivery: A systematic literature review and validation through focus group. *Journal of Construction Engineering and Management*, 147(10): 03121005
- ISO (2016). Industrial automation systems and integration — Product data representation and exchange — Part 21: Implementation methods: Clear text encoding of the exchange structure, ISO 10303-21:2016
- Kong L, Chen C, Zhao R, Chen Z, Wu L, Yang Z, Li X, Lu W, Xue F (2022). When permissioned blockchain meets IoT oracles: An on-chain quality assurance system for off-shore modular construction manufacture. In: *2022 IEEE 1st Global Emerging Technology Blockchain Forum: Blockchain and Beyond (iGETblockchain)*, Irvine CA, USA: IEEE
- Kong L, Xue F (2023). Addressing BIM versioning challenges: A blockchain-ready IFC schema extension for semantic change management. In: *Proceedings of the 28th International Symposium on Advancement of Construction Management and Real Estate*, Springer
- Kong L, Zhao R, Xue F (2024). Zero-Knowledge Proof for trusted construction management: A preliminary study of adaptive blockchain BIM identity authentication. In: *Proceedings of the 23rd International Conference on Construction Applications of Virtual Reality*, Florence Italy: University of Florence Press
- Krijnen T, Beetz J (2020). An efficient binary storage format for IFC building models using HDF5 hierarchical data format. *Automation in Construction*, 113: 103134
- Laakso M, Kiviniemi A (2012). The IFC standard: A review of history, development, and standardization, *Information technology. Journal of Information Technology in Construction*, 17(9): 134–161
- Lai H, Deng X (2018). Interoperability analysis of IFC-based data exchange between heterogeneous BIM software. *Journal of Civil Engineering and Management*, 24(7): 537–555
- Lee D, Lee S H, Masoud N, Krishnan M, Li V C (2021a). Integrated digital twin and blockchain framework to support accountable information sharing in construction projects. *Automation in Construction*, 127: 103688
- Lee Y C, Eastman C M, Solihin W (2021b). Rules and validation processes for interoperable BIM data exchange. *Journal of Computational Design and Engineering*, 8(1): 97–114
- Li J, Greenwood D, Kassem M (2019). Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases. *Automation in Construction*, 102: 288–307
- Li X, Lu W, Xue F, Wu L, Zhao R, Lou J, Xu J (2022). Blockchain-enabled IoT-BIM platform for supply chain management in modular construction. *Journal of Construction Engineering and Management*, 148(2): 04021195
- Liang D, Xue F (2023). Integrating automated machine learning and interpretability analysis in architecture, engineering and construction industry: A case of identifying failure modes of reinforced concrete shear walls. *Computers in Industry*, 147: 103883
- Liang D, Chen S-H, Chen Z, Wu Y, Chu L Y, Xue F (2024). 4D point cloud-based spatial-temporal semantic registration for monitoring mobile crane construction activities. *Automation in Construction*, 165: 105576
- Lin B, Chen Y, Chen X, Yu Y (2012). Comparison between JSON and XML in applications based on AJAX. In: *2012 International Conference on Computer Science and Service System*, IEEE
- Liu Y, van Nederveen S, Hertogh M (2017). Understanding effects of BIM on collaborative design and construction: An empirical study in China. *International Journal of Project Management*, 35(4): 686–698
- Lou J, Lu W, Xue F (2021). A Review of BIM Data Exchange Method in BIM Collaboration. In: *Proceedings of the 25th International Symposium on Advancement of Construction Management and Real Estate*, Singapore: Springer
- Lu W, Lou J, Ababio B K, Zhong R Y, Bao Z, Li X, Xue F (2024). Digital technologies for construction sustainability: Status quo, challenges, and future prospects. *npj Materials Sustainability*, 2(10)
- Monrat A, Schelén O, Andersson K (2019). A survey of blockchain from the perspectives of applications, challenges, and opportunities. *IEEE Access: Practical Innovations, Open Solutions*, 7: 117134–117151

- Mutis I, Paramashivam A (2019). Cybersecurity management framework for a cloud-based BIM model, *Advances in Informatics and Computing in Civil and Construction Engineering*. Springer International Publishing
- Nawari N, Ravindran S (2019). Blockchain and the built environment: Potentials and limitations. *Journal of Building Engineering*, 25: 100832
- Orace M, Hosseini M R, Edwards D J, Li H, Papadonikolaki E, Cao D (2019). Collaboration barriers in BIM-based construction networks: A conceptual model. *International Journal of Project Management*, 37(6): 839–854
- Perera S, Nanayakkara S, Rodrigo M, Senaratne S, Weinand R (2020). Blockchain technology: Is it hype or real in the construction industry? *Journal of Industrial Information Integration*, 17: 100125
- Pinheiro S, Wimmer R, O'Donnell J, Muhic S, Bazjanac V, Maile T, Frisch J, van Treeck C (2018). MVD based information exchange between BIM and building energy performance simulation. *Automation in Construction*, 90: 91–103
- Pradeep A S, Amor R, Yiu T W (2020). Blockchain improving trust in BIM data exchange: A case study on BIMCHAIN. In: *Construction Research Congress 2020: Computer Applications ASCE*
- Sacks R, Kedar A, Borrmann A, Mal L, Brilakis I, Huthwohl P, Daum S, Kattel U, Yosef R, Liebich T, Barutcu B E, Muhic S (2018). SeeBridge as next generation bridge inspection: Overview, information delivery manual and model view definition. *Automation in Construction*, 90: 134–145
- Sacks R, Wang Z, Ouyang B, Utkucu D, Chen S (2022). Toward artificially intelligent cloud-based building information modelling for collaborative multidisciplinary design. *Advanced Engineering Informatics*, 53: 101711
- Singh J, Anumba C J (2023). Building commissioning process and documentation: A literature review and directions for future research. *International Journal of Construction Management*, 24(1): 75–85
- Solihin W, Eastman C, Lee Y C (2015). Toward robust and quantifiable automated IFC quality validation. *Advanced Engineering Informatics*, 29(3): 739–756
- Sun C, Zeng X, Sun C, Si H, Li Y (2020). Research and application of data exchange based on JSON. In: *2020 Asia-Pacific Conference on Image Processing, Electronics and Computers*, IEEE
- Sun J, Yan J, Zhang K (2016). Blockchain-based sharing services: What blockchain technology can contribute to smart cities. *Financial Innovation*, 2(1): 26
- Swan M (2015). *Blockchain: Blueprint for a New Economy*. Sebastopol CA, USA: O'Reilly Media, Inc.
- Tang S, Shelden D, Eastman C, Pishdad-Bozorgi P, Gao X (2020). BIM assisted building automation system information exchange using BACnet and IFC. *Automation in Construction*, 110: 103049
- Tao X, Das M, Liu Y, Cheng J (2021). Distributed common data environment using blockchain and interplanetary file system for secure BIM-based collaborative design. *Automation in Construction*, 130: 103851
- Tao X, Liu Y, Wong P Y, Chen K, Das M, Cheng J (2022). Confidentiality-minded framework for blockchain-based BIM design collaboration. *Automation in Construction*, 136: 104172
- Wang G (2011). Improving data transmission in web applications via the translation between XML and JSON. In: *2011 Third International Conference on Communications and Mobile Computing*, IEEE
- Wang J, Shen Y, Xiong X, Wang X, Fang X (2022a). Research on multi-person collaborative design of BIM drawing based on blockchain. *Scientific Reports*, 12(1): 16312
- Wang N, Issa R R, Anumba C J (2022b). NLP-based query-answering system for information extraction from building information models. *Journal of Computing in Civil Engineering*, 36(3): 04022004
- Wang Z, Ouyang B, Sacks R (2023). Graph-based inter-domain consistency maintenance for BIM models. *Automation in Construction*, 154: 104979
- Wang Z, Wang T, Hu H, Gong J, Ren X, Xiao Q (2020). Blockchain-based framework for improving supply chain traceability and information sharing in precast construction. *Automation in Construction*, 111: 103063
- Wong J, Wang X, Li H, Chan G, Li H (2014). A review of cloud-based BIM technology in the construction sector. *Journal of Information Technology in Construction*, 19: 281–291
- Wu J, Zhang J (2019). New automated BIM object classification method to support BIM interoperability. *Journal of Computing in Civil Engineering*, 33(5): 04019033
- Wu L, Lu W, Chu L, Chen C (2024). Visualizing blockchain in construction projects: Status quo, challenges, and a guideline for implementation. *Frontiers of Engineering Management*, doi:10.1007/s42524-024-4034-6
- Xu H, Kim J, Chen J (2022). An iterative reference mapping approach for BIM IFCXML classified content compression. *Advanced Engineering Informatics*, 54: 101788
- Xue F, Lu W (2020). A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Automation in Construction*, 118: 103270
- Yu Z, Gong Y (2024). ChatGPT, AI-generated content, and engineering management. *Frontiers of Engineering Management*, 11(1): 159–166
- Zhang J, Liu Q, Yu F, Hu Z, Zhao W (2014). A framework of cloud-computing-based BIM service for building lifecycle. *Computing in Civil and Building Engineering*, 1514–1521
- Zhao R, Chen Z, Xue F (2023). A Blockchain 3.0 paradigm for digital twins in construction project management. *Automation in Construction*, 145: 104645
- Zhao R, Wu Y, Lou J, Chen Z, Wu L, Xue F (2021). Comparison of BIM collaboration paradigms for digital twin readiness: Centralized files, decentralized clouds, and distributed blockchains. In: *Proceedings of the 38th International Conference of CIB W78*, Luxembourg: CIB press: 315–325
- Zhao X, Feng Y, Pienaar J, O'Brien D (2017). Modelling paths of risks associated with BIM implementation in architectural, engineering and construction projects. *Architectural Science Review*, 60(6): 472–482
- Zhao Y, Taib N (2022). Cloud-based Building Information Modelling (Cloud-BIM): Systematic literature review and bibliometric-qualitative analysis. *Automation in Construction*, 142: 104468