A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration

Fan Xue and Weisheng Lu
xuef@hku.hk ; wilsonlu@hku.hk
The University of Hong Kong

This is the peer-reviewed post-print version of the paper:
Xue, F., & Lu, W. (2020). A semantic differential transaction approach to minimizing information redundancy for BIM and blockchain integration. *Automation in Construction*, 117, 103270. Doi: 10.1016/j.autcon.2020.103270
The final version of this paper is available at: https://doi.org/10.1016/j.autcon.2020.103270.

Highlights

- A novel semantic differential transaction (SDT) approach for BIM and blockchain integration was proposed.
- The SDT core identifies the incremental semantic changes in BIM development cycle.
- The SDT approach was implemented in Python with state-of-the-art algorithms and JSON data structures.
- The SDT approach has a smart contract-like change consensus protocol, which is ready for blockchain.
- The SDT approach was validated on two BIM cases.
- BIM changes in the tests were captured with minimum information redundancy, e.g., the SDT results were as small as 0.02% of the BIM file size.
- The tests confirmed the bi-directional operations between BIM and SDT results in near real-time.

Abstract

Those attempting to integrate building information modeling (BIM) and blockchain soon encounter the enormous challenge of information redundancy. Storage of duplicated building information in decentralized ledgers already creates redundancy, and this is exacerbated as the BIM model develops and is utilized. This paper presents a novel semantic differential transaction (SDT) approach to minimizing information redundancy in the nascent field of BIM and blockchain integration. Whereas the conventional thinking is to store an entire BIM model or its signature code in blockchain, SDT captures local model changes as SDT records and assembles them into a BIM change contract (BCC). In this way, the version history of a BIM project becomes a chain of timestamped BCCs, and stakeholders can promptly synchronize

BIM changes in blockchain. We test our approach in two pilot cases. The results show that SDT captures, in near real time, sequential and simultaneous BIM changes at less than 0.02% of the Industry Foundation Classes file size. We also prove model restoration from the lightweight BCCs in a small-scale BIM project. In addressing the fundamental issue of information redundancy in BIM and blockchain integration, this research can help the industry advance beyond the rhetoric to develop operable blockchain BIM systems.

Keyword: Building information modeling, Semantics, Blockchain, Industry foundation classes, Interoperability, Information redundancy.

1 Introduction

Various researchers have articulated the challenges of construction. Every building is a unique prototype developed by a team of stakeholders that may never have worked together before and may never again (ICE 2019). Construction processes such as design, manufacturing, transportation, and site work suffer discontinuity and are deeply fragmented, distributed, and specialized (Egan 1998). This situation is made worse by the long construction supply chain for design for manufacturing and assembly (DFMA) and industrialized construction (Molloy et al. 2012; Larsson et al. 2014). The fragmentation and distribution features cause widespread and chronic problems, such as inferior quality, escalating cost, severe delay, and lackluster productivity. Successful delivery of any construction project requires seamless collaboration among stakeholders and efficient information exchange, and a broad spectrum of model specifications and software tools for specialized construction tasks have been adopted to this end. In addition, interoperability of building information is critical (Eastman et al. 2011). Building information modeling (BIM) provides this interoperability through a trustworthy, shared information platform. As the “digital representation of physical and functional characteristics of a facility and a shared knowledge resource for information about a facility, forming a reliable basis for decision during its life-cycle” (NIBS 2015), BIM is a game-changing technology that has been successfully mainstreamed across the global construction industry.

Recently emerging from the technology sphere, blockchain is potentially an alternative means of building trustworthy collaboration in construction. A blockchain is a cryptographically secured distributed ledger within a decentralized consensus mechanism (Risius & Spohrer 2017). It keeps an immutable, secure, and transparent database through which users can transact valuable assets in a public and pseudonymous setup without the presence of an intermediary or central authority (Beck et al. 2016; Xia et al. 2017). Traditional exhortations of trust building have a strong root of normativism. According to this school, trust is a quintessence to business success, an intrinsic value of human being, and a social norm (Laan et al. 2011). Therefore, we do anything positive to build it. Blockchain-based trust building, in contrast, has a root of naturalism. Untrusting behavior in construction transactions is a state that is accepted as natural, like it or not. However, blockchain adopts an alternative approach by keeping custody of immutable, cryptographic, and verifiable information in decentralized ledgers that construction stakeholders cannot deny or falsify but choose to trust each other. Blockchain is not based on a single centralized server or company's cloud. Rather, it is supported by a network of computers (peers), each holding all duplicated transactions in a blockchain. The duplicated transaction histories introduce information redundancy for the sake of credibility (e.g., by safeguarding immutable, decentralized, and distributed information) but sacrifice time, storage, and access efficiency in comparison with native computer storage.

Interest in BIM and blockchain integration is growing. For example, Li et al. (2019) review blockchain technology in the built environment and construction industry, presenting conceptual models and practical use cases. Zheng et al. (2019) propose a blockchain-based big data model for BIM modification audit and provenance. According to Penzes (2018), "the fundamental concept that can enable the combination of BIM and blockchain technology is their shared ability to serve as a single source of truth." He distinguishes two ways of utilizing BIM and blockchain: (1) BIM can take information from the blockchain, such as supply chain, provenance, installation, and payment; and (2) building information can be assigned to a blockchain to be used later, e.g., for smart payment or procurement. Through integration, therefore, BIM and blockchain can offer more value-added applications than either can separately.

However, those who aim to develop an operable blockchain BIM system face massive challenges. One is information redundancy. The file-based data exchange in BIM (e.g., information delivery manual) leads to massive data volume. A typical model can be of tens to hundreds of megabytes (MB), while block sizes are typically at kilobyte (KB) levels. As mentioned above, to ensure information accountability transactions in a blockchain are duplicated and safeguarded in a decentralized ledger distributed among peers. This process will increase the BIM data volume exponentially, and it will be "sticky" to maneuver it. Even more challenging is that information in BIM is continuously being changed and updated by stakeholders. The archived history of a model is redundant in current practice because saving a small change can lead to a new BIM file. Although it is technically feasible to blockchain an entire model and its history, e.g., using the MD5 hash value of a model, users have to spend considerable time and Internet bandwidth to synchronize a new BIM file. Managing changes, especially those made simultaneously by different stakeholders, is notoriously difficult using existing centralized and cloud BIM platforms (e.g., BIM 360), let alone in decentralized, widely distributed ledgers. Finding a novel way to minimize information redundancy is a fundamental challenge to harnessing the power of BIM and blockchain integration.

This paper aims to develop an innovative semantic differential transaction (SDT) approach to minimizing information redundancy. This approach is applicable to Industry Foundation Classes (IFC), the de facto open information standard ensuring interoperability across different BIM platforms, and is based on capturing BIM changes, safeguarding them in a blockchain, and restoring them when needed. The remainder of the paper is organized as follows. Section 2 reviews the literature on information change management in a BIM context, and Section 3 reviews blockchain technologies and their promise in construction. Section 4 presents the SDT approach with its three components: a semantic interoperability method, an SDT model, and a BIM change contract (BCC). The SDT approach is further illustrated and validated in two pilot studies in Section 5. The novelties and shortcomings of the approach are discussed in Section 6, and conclusions drawn in Section 7.

2 BIM interoperability and IFC

The kernel of BIM is information (Lu et al. 2018), and the product is a 3D or *n*D digital model of physical and functional characteristics of a facility. This model contains various digital components or objects. In the back end, BIM consists of clustered arrays of information, e.g., organized in a BIM file or a database. The information comprises geometric and non-geometric semantics (Jung & Joo 2011; Xue et al. 2018b). The geometric semantics describe the sizes, volumes, shapes, and textures of individual BIM objects, while the non-geometric semantics describe less visible but arguably more meaningful attributes such as functions, behavior, cost, and maintenance history (Pratt 2004). BIM was developed with a view to providing a one-truth information source facilitating communication amongst stakeholders such as clients, designers, engineers, contractors, and suppliers. However, the models can be developed or enriched by different stakeholders using BIM authoring tools, and neither the digital models nor the back-end databases lend themselves to easy communication among these stakeholders. Therefore, interoperability of different stakeholders' models is highly desired (Eastman et al. 2011) to provide the data foundation for BIM-based project collaboration and decision-making (Taylor & Bernstein 2009). While the industry is reinforcing proprietary BIM platforms and solutions, an open BIM standard is the key to interoperability.

IFC is an open data exchange schema that facilitates BIM interoperability in the architecture, engineering, and construction (AEC) industry, with ISO certifications such as 16739:2013 and 16739-1:2018. Developed by buildingSMART International, IFC defines BIM objects using an EXPRESS (ISO 10303-11)-based entity-relationship model and saves the BIM model in the STEP (Standard for the Exchange of Product model, ISO 10303-21) file format with the *.ifc* file extension. The latest version of IFC now consists of more than 600 entities organized into an object-based inheritance hierarchy (buildingSMART 2019). Figure 1 illustrates parts of the IFC schema (Version 4, Addendum 2), which is the meta-model of how the standardized IFC data (e.g., objects identities, semantics, relations, and concepts) are organized (buildingSMART 2019). *IfcRoot* is at the top-most abstract level. Derived from it are three fundamental IFC model entity types: *IfcObjectDefinition* capturing semantically treated tangible object items (e.g., products, processes, and resources); *IfcPropertyDefinition*, which defines the characteristics of both general object types and specific object occurrences; and *IfcRelationship* assigning property information to the corresponding BIM objects while specifying the relationships among objects. IFC has been widely adopted as a general standard and is supported by many

BIM software vendors (Ali & Mohamed 2017; Gao et al. 2017), and is thus the focus of this paper in integrating open BIM and blockchain.

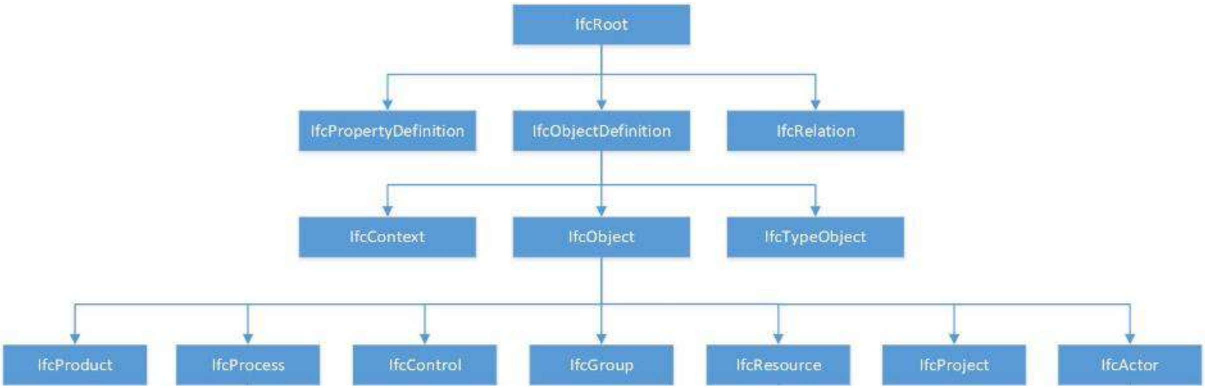


Figure 1. Part of the IFC schema (Version 4, Addendum 2)

Information redundancy is a problem of continuous BIM data exchange using IFC. The redundancy is rooted in two aspects: STEP format's sequential identifiers (STEP #-Ids) in each line, and the cross-referencing of IFC objects' generated globally unique identifiers (GUIDs). The STEP #-Ids are sometimes randomly generated for IFC objects, which leads to considerable byte-level inconsistency in the *.ifc* files. An IFC object's GUID ought to be unique and consistent through the BIM lifecycle. However, many GUIDs, regardless of the complex references and relations anchored between them, are randomized on the mainstream BIM platforms. For example, Autodesk Revit retains the GUIDs of IFC objects that are associated with a unique "ElementID," such as doors (*IfcDoor*), but randomizes the GUIDs of other objects such as a door's properties (*IfcPropertySet*). As a result, one small change in a BIM model, or even no change at all, can result in a considerably different IFC file. With these randomly assigned GUIDs, together with the complex hierarchical structures, BIM objects become very difficult to trace and compare when massive files are exchanged. In contrast to the line-by-line STEP structure, modern tree-like data structures, e.g., in JavaScript Object Notation (JSON) and eXtensible Markup Language (XML), have higher computational efficiency and explainability. Thus, buildingSMART (2020) has developed other IFC formats such as IFCXML based on STEP-XML standard (ISO 10303-28), IFC-ZIP, IFC-JSON, and IFC-SQLite. Some new IFC formats, such as IFCXML, have eliminated the inconsistency from STEP #-Ids, though introducing some other types of byte-level inconsistency; E.g., "<Tag></Tag>" and "<Tag />" are equivalent in XML but different in the byte level.

The global AEC community has endeavored to minimize information redundancy by comparing BIM changes in IFC files. Lee et al. (2011) used a "flattening" method, decoding the relations and nesting all the referenced definitions to form a full description for an IFC instance. Oraskari & Törmä (2015) developed a Short Paths Crossings Algorithm (SPCA) to detect the changes between IFC-derived graphs. Afsari et al. (2017) confirmed the possibility of serializing IFC objects in the JSON format, which is better supported by modern programming languages. Shi et al. (2018) investigated the content rather than flattening and developed similarity index software; Shafiq & Lockley (2018) suggested looking into the 'signature' of IFC objects; Lin & Zhou (2020) implemented a hash code for quick detection of BIM changes in Autodesk Revit; and Li et al. (2020) presented a Tversky similarity-based method for querying IFC objects based on their semantic attributes. Froese (2003) pinpointed another research direction as the GUID-based transactional IFC exchange on distributed systems, beyond the file-based exchange. Later, buildingSMART started to develop the BIM Collaboration Format (BCF) standard of IFC model servers. Jørgensen et al. (2008) demonstrated an IFC model server with code version-control functions such as "check out" and "check in" for editing a subset of the IFC objects with GUIDs; Lee et al. (2014) confirmed object-relational databases could improve the querying performance of such servers. Such GUID-based transactional exchanges of IFC semantics are becoming increasingly important in real-time applications such as virtual reality (Du et al. 2018). In short, BIM objects should be assigned their semantic meanings and associated with specific GUIDs rather than random ones to reduce redundancy and improve interpretability.

Two essential characteristics of BIM change management inspired this study: (a) the incremental nature of BIM changes, and (b) the systematic nature of BIM semantics. Similar to a Lego stacking process, BIM is developed element by element and phase by phase (Figure 2a). This presents an opportunity to distinguish and blockchain the model development cycle as incremental changes rather than recording the entire model every time a change is made. BIM files are organized in a meaningful way (Figure 2b), and the task of comparing and capturing model changes should focus not on the byte level but the semantic level: the meanings, systematic relations, and their hierarchies. Wang and Meng (2019) regard semantics as the key to managing not only BIM but also other construction processes and knowledge. However, how to identify the incremental semantic changes automatically in BIM, especially for IFC, is yet to be satisfactorily explored by the literature.

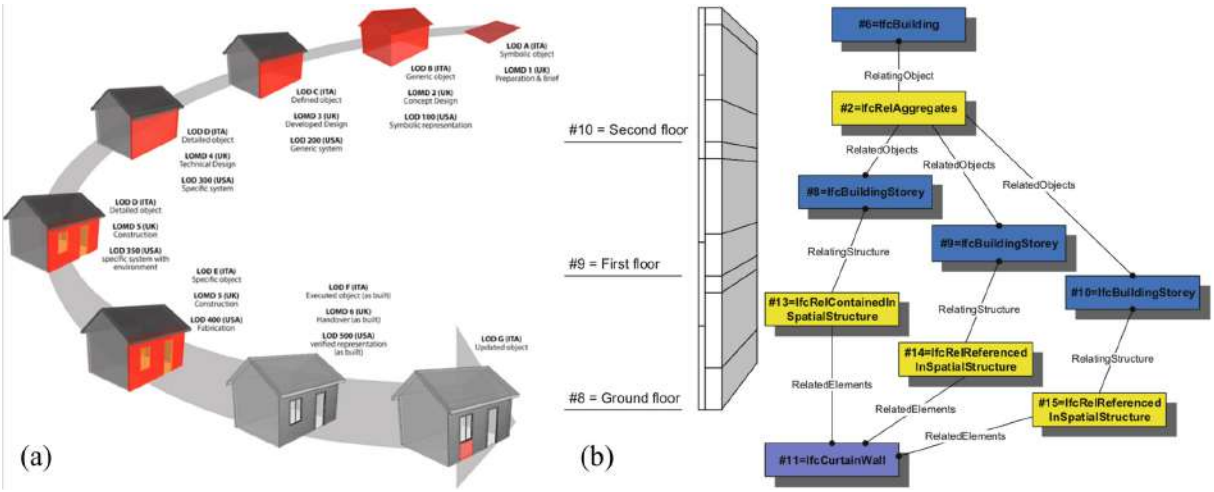


Figure 2. The incremental and systematic nature of BIM. (a) Incremental development (Ellis 2019); (b) Example relation system between IFC instances (Borrmann et al. 2018)

3 Blockchain in construction

Blockchain has recently received construction industry attention for its payment, procurement, supply chain, BIM, and smart asset management potential. For example, Dakhli et al. (2019) propose that blockchain could help achieve a saving of 8.3% of the total cost of residential construction. Allam and Jones (2019) have investigated blockchain potential for air rights development as an urban sprawl prevention measure, and Li et al. (2019) and Wang et al. (2020) establish technical frameworks for blockchain in the construction industry. Nevertheless, empirical blockchain studies for construction have been limited, with Perera et al. (2020) finding barriers such as digital asset privacy and scalability in construction and the 50% vulnerability in blockchain technology. Industrial reports such as Kinnaird et al. (2017) and Penzes (2018) focus more on the potential value-added applications of BIM, blockchain and their integration for smart contracts and quality assurance. Recent construction scandals, e.g., fake concrete tests in the Hong Kong-Zhuhai-Macau bridge (SCMP 2017) and corner-cutting in the Hung Hom MTR Station construction (SCMP 2019), have led to calls for the use of blockchain to safeguard building information for provenance and forensic investigation purposes. Whether BIM and blockchain integration should occur seems to be no longer debatable, and now the industry should move beyond envisioning such an integrated system to actually constructing one that is genuinely operable.

Unlike conventional file systems or relational databases, blockchain adopts a distributed data architecture. Three components support its function, namely, cryptographic algorithms, a distributed database, and a decentralized consensus mechanism (Hawlitschek et al. 2018). Cryptographic algorithms, e.g., Secure Hash Algorithms (SHA), are used to encrypt transactional data based on the agreed blockchain protocol (Beck et al. 2016). The algorithms promise that it is practically impossible to derive the original data from the generated ciphertext. The data is then appended to a chain of data blocks with cryptographic inter-connections (Gipp & Breitinger 2016). The distributed database and decentralized consensus mechanism are rooted in early work on homogeneous distributed database systems (Breitbart et al. 1986). These systems, such as cloud services and distributed database engines, are now widely available (Özsu & Valduriez 2020). Due to the distributed nature of the data, no third party is entrusted with responsibility for its validation and management. Instead, all nodes collect the transactions into a new block and work on the consensus protocols, such as proof of work (PoW) and proof of stake (PoS), to validate the transaction systems (Notheisen et al. 2017).

Blockchain is built on an information redundancy mechanism that deliberately sacrifices efficiency and speed to achieve its designated merits of immutability and decentralization (Wüst & Gervais 2018). Although to the best of our knowledge there is no literature investigating its exact extent, one can easily imagine duplication in a blockchain as it encrypts pieces of information chained with hash codes and distributes them to decentralized ledgers in different peers. While computer storage space and Internet speed are increasingly affordable, one must consider information efficiency and speed when it comes to blockchaining BIM models. Our industrial engagements have shown that these models, depending on project complexity and Level of Development (LoD), are often too "sticky" to be maneuvered using remote Internet computers. This explains why previous studies such as Zheng et al. (2019) only store BIM files' hashing signatures on chain and do not handle information redundancy in the models, with the result that BIM interoperability still creates a massive amount of network traffic.

4 The proposed approach

The SDT approach to minimizing information redundancy developed in this paper is a computational model of BIM changes over time. Calculating all the essential semantic changes with minimized redundancy, it is an innovative means of mapping BIM onto blockchain, and vice versa. The overall framework is shown in Figure 3. Three layers of the SDT approach connect the distributed BIM systems to the Internet-based blockchain shell: (i) semantic interoperability, (ii) the SDT model, and (iii) BIM change contract (BCC). The first layer connects to the BIM, while the third plugs in blockchain's distributed implementation. SDT ignores all the semantically unchanged BIM objects and focuses on the changes only, handling not only sequential changes but also distributed simultaneous changes for multi-stakeholder BIM uses.

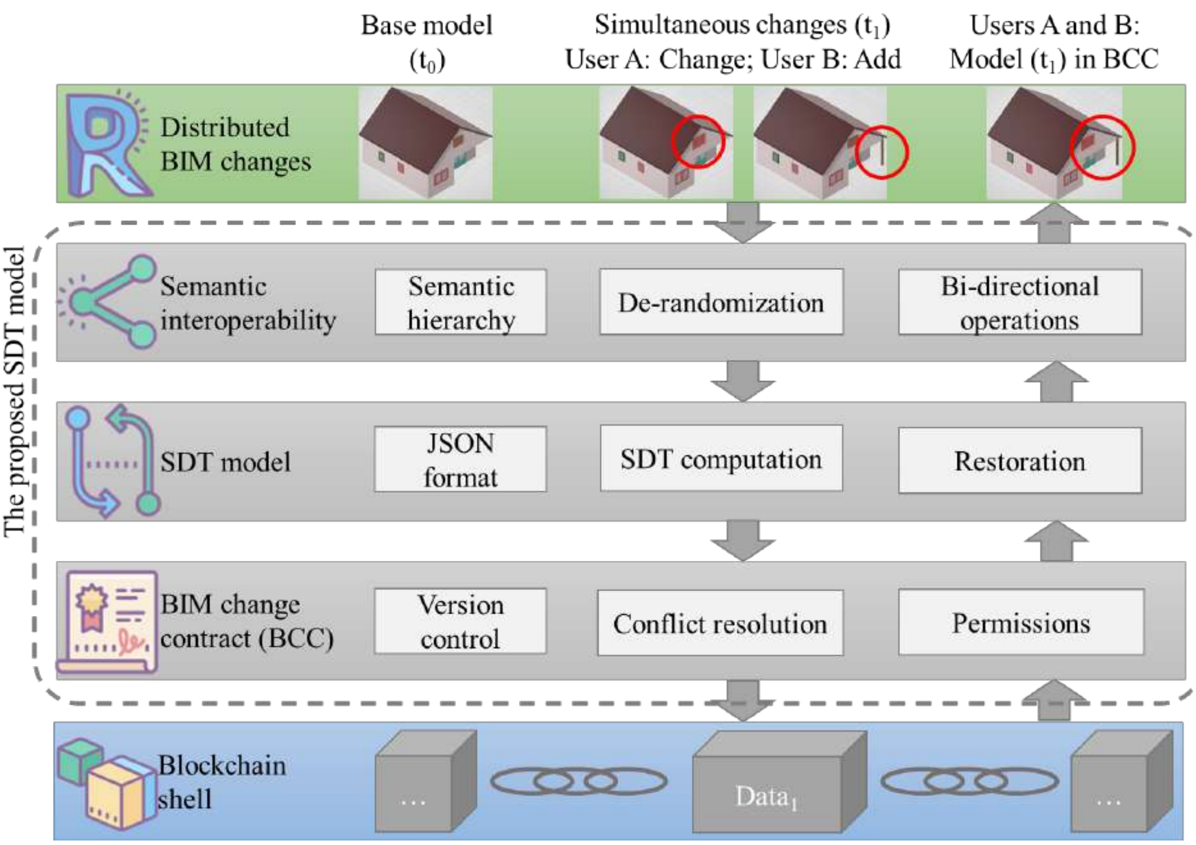


Figure 3. Framework of the proposed SDT approach for integrating blockchain and BIM

4.1 Semantic interoperability

This paper employs IFC as the target BIM format due to its openness and wide recognition. As shown in Figure 3, the semantic interoperability layer focuses on three functions: semantic hierarchy, de-randomization, and bi-directional operations between IFC and blockchain.

The semantic hierarchy function processes the STEP expressions, representing all the IFC objects and their geometric and non-geometric properties, into systematic tree-like hierarchies. For example, the type and style expression (e.g., of *IfcWallType* and *IfcDoorStyle*) can be embedded into the physical BIM objects (e.g., *IfcWall* and *IfcBuilding*). The hierarchy generation process removes partial randomized contents, such as the expressions' line numbers and some ad hoc relations. The embedding results are tree-like efficient data structures compatible with IFC's non-STEP formats such as IFCXML and Afsari et al.'s (2017) IFCJSON. However, there is a trade-off between full explanatory power and computational efficiency. For example, a material definition referred by twenty structural elements is better attached to a "materials" hierarchy independent of the main hierarchy of building elements.

The de-randomization function aims to eliminate the remaining random contents to streamline the semantic hierarchy. First, a selected list of attributes of software oracles, i.e., potential names, are examined for each IFC object. For instance, Autodesk Revit can export its internal object IDs into the *Tag* descriptors in IFC. Another example is the unique names such as *Width* and *Height* defined in certain geometric property sets. In addition, the hashing function, which is well known in blockchain, is a baseline method for mapping the semantic expression of an object to a short, semantic content-only code if ultimately the expected attributes cannot be found. By using such a priori identifier or the hashing function, an IFC object can be recognized by a semantic identifier rather than the random GUID. Meanwhile, the references to the de-randomized objects can also be updated.

Bi-directional operability focuses on reconstructing IFC from the de-randomized semantic hierarchy. In order to maintain reconstructability, there should be no semantic (excluding the random contents) losses in the semantic hierarchy function, while auxiliary properties or relations are allowed. The de-randomized semantic hierarchy can be re-randomized with new standard STEP #-Ids to fit the IFC standard, though byte-level accuracy is not guaranteed. The re-randomized IFC model should be semantically identical to the real one, e.g., the same geometries and relations, though the byte-level contents can be considerably different. The bi-directional operability is thus more straightforward in the IFCXML format than the IFC STEP format because there is less involvement of randomized contents.

4.2 Semantic differential transaction (SDT) model

The SDT model translates between the BIM changes in IFC and the SDT records on chain. So, for example, if the BIM model's semantic hierarchies were information "bank accounts," an IFC version history of the BIM semantics would be a long list of "bank transactions" of "deposits/withdrawals." Figure 4 shows the pseudo-code for computing the SDT from two consecutive (i.e., slightly changed) models, i.e., ifc_0 and ifc_1, of a BIM project. First, the input IFC models are read into two tree objects (i.e., σ_0 and σ_1 on Lines 1–2) of semantic hierarchies through the semantic interoperability functions, so that the σ_0 and σ_1 are free from random contents (both STEP #-Ids and GUIDs). Then, a quick comparison on Lines 3–5 removes the unchanged IFC instances as the intersection tree from σ_0 and σ_1. The removal can considerably expedite the M to N comparison of σ_{0c} and σ_{1c}, where M is the maximum branching size in the changed semantic hierarchies σ_{0c}, and N is that of σ_{1c}. Finally, the SDT from σ_{0c} to σ_{1c} can be computed as the difference between the two tree objects, through up-to-date tree comparison algorithms (Line 6). Line 1 in Figure 4, i.e., $\sigma_0 \leftarrow \sigma_{1\_previous}$, indicates the possible reuse of previous semantic hierarchy to save time from IFC loading, parsing, and de-randomization.

```
procedure compute_SDT
input: ifc0, ifc1                                   // IFC changed between t0 and t1
1   σ0 ← semantic_interoperability ( ifc0 );         // To call "semantic interoperability"
2   σ1 ← semantic_interoperability ( ifc1 );
3   σ* ← σ0 ∩ σ1;                                    // The intersection (unchanged) tree
4   σ0c ← σ0 − σ*;                                   // To purge the unchanged instances
5   σ1c ← σ1 − σ*;
6   Δσ ← tree_diff (σ0c, σ1c);                       // Difference between changed objects
7   return Δσ
```

Figure 4. Pseudo code of the SDT computation algorithm

As shown in Figure 5, SDT results consist of three types of changes: addition, change, and deletion. An oracle ID is assigned to recognize the BIM object from multiple instances of the same type. Two keywords "insert" and "delete" are preserved for indications, while a value pair such as the item "Property3" stands for a changed property. If the property is an array of values, all types of changes are in value pairs, with possible involvements of the empty JSON object "{}," as shown in Figure 5.

```
{
  { "IfcObject#OracleID": {                              // A changed IFC object with an Oracle ID
      insert:       { "Property1": New_value },          // A new "Property1" is added
      delete:       { "Property2": Old_value },          // The "Property2" is deleted
      "Property3":  [ Old_value,    New_value ],         // The "Property3" is changed
      "Array1":     [ {},           New_values ],        // Added to "Array1"
      "Array2":     [ Old_values,  {} ],                 // Deleted from "Array2"
      "Array3":     { i: [ Old_value, New_value ] }      // The i-th entry of "Array3" is changed
  } },
  …
}
```

Figure 5. JSON example of SDT records of BIM changes

The BIM semantic hierarchy can be restored at any time by adding up all the transactions to the base model, i.e., $\sigma_k = \sigma_0 + \Sigma_{i=1, 2, \ldots, k}\, \Delta_{\sigma i}$, based on the bi-directional operability function in Sect. 4.1. The restoration is an inverse operation of the differential in Figure 4. With such data structure, the restored BIM semantic hierarchy is computable for many BIM applications. Because of the small sizes of the SDT records, the proposed approach can achieve minimal information redundancy for BIM data exchange.

In order to track all the BIM changes in the development, the SDT computation in Figure 4 can be regularly executed, e.g., every minute, or triggered by the task when the BIM project is saved. In terms of disk (and memory) space, the saving will be considerable for large BIM projects; one only needs an initial base model plus a time series of SDT records of the incremental changes to represent the whole development history. Nevertheless, one has to spend time on BIM restoration for the up-to-date or a historical version. Major version checkpoints, like keyframes for video coding, can limit the extra time to a certain amount. Therefore, SDT computation can offer a spectrum of trade-off options between the computational space and time.

4.3 BIM change contract

The BIM change contract (BCC) in the SDT approach aims to provide a smart contract-like protocol for integrating multiple BIM editors' distributed SDT records for blockchain. Figure 6 shows the BCCs on a permissioned blockchain structure, i.e., with restricted access. Generally, permissioned blockchain architectures are slightly preferred over permissionless ones for management purposes, according to a PwC (2018) global survey. The BCC is a smart contract protocol that involves three groups of elements in Figure 6: base models in the middle, the interconnected blockchain nodes, and the stakeholders' current BIM models with software and hardware oracles.

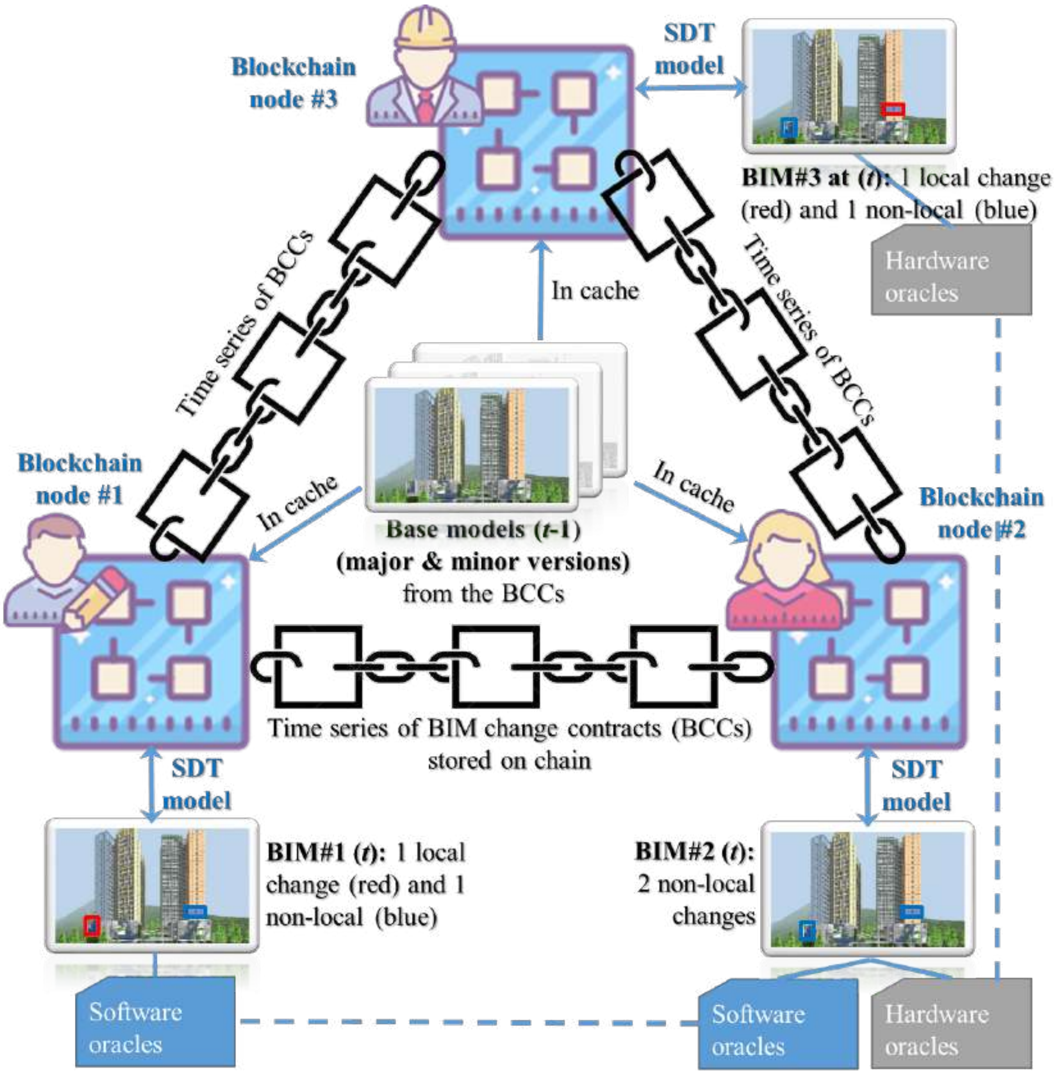


Figure 6. Example blockchain architecture for BIM change contract over SDTs

A BCC concluded at time t, noted as BCC_t, represents the overall BIM changes by all the stakeholders between time $t - 1$ and t. Therefore, at time t, the base model (as shown in the middle of Figure 6) is the initial BIM model with accumulated historical BCCs up to time $t - 1$, i.e., $ifc_{t-1} = ifc_0 + \Sigma_{t-1} BCC_i$. A special case is that the base model at $t = 1$ is the initial model (ifc_0), when no BCCs are stored in the blockchain. The base model is identical but is not centralized or shared. Instead, it is computed, trusted, and cached by every stakeholder individually on top of the trusted initial model (ifc_0) and the trusted historical BCCs on the chain.

Each BIM stakeholder runs a blockchain node for conflict resolution and version control in the permissioned architecture in Figure 6. Each blockchain node has the base model in its local cache, a reserved memory space, and monitors the local changes regularly, as described in Section 4.2. The local SDT records computed by the algorithm in Figure 4 only reflect the stakeholder's local BIM change. In a distributed BIM context, there can be conflicts in SDT records submitted by different stakeholders simultaneously. The conflict resolution mechanisms are thus necessary to conclude a contract on the overall changes. Conflict resolving methods can be as complicated as Jäger's (2018) directed acyclic graph (DAG) model for Turing completeness, or simple divide-and-conquer of all BIM objects' editorships to designated stakeholders, e.g., all the air ducts to one sub-contractor. The latter mechanism leads to a single version of the base BIM model, while the DAG approach may generate a major and several minor versions.

Each stakeholder works on its current BIM model independently. For example, Stakeholder 1 updates the glass curtain wall of the lobby in BIM#1 in Figure 6, while Stakeholder 3 changes a facade on the third floor in BIM#3. Both changes are tracked as local SDT records (indicated in red boxes) and integrated into the BCC at time *t*. Due to the bi-directional operability of the SDT model, other stakeholder's SDT records can be restored immediately to updated BIM objects based on the cached identical base BIM model. As a result, each stakeholder, including Stakeholder 2 who makes no changes, can be aware of the non-local changes (indicated in blue boxes) in the meantime. Software and hardware oracles in Figure 6 can automate the identification of BIM objects in the construction processes and local SDT records. For example, a software oracle is a good naming convention based on the hierarchy of BIM objects such as the "function/type/vertical-location/horizontal-location/description" format (Chen et al. 2017). An example of a hardware oracle is the Internet of Things attached to the construction elements (Xue et al. 2018a).

4.4 Software implementation

We implement the SDT approach in Python (Ver. 3.7). Three classes, namely, *Interop*, *SdtModel*, and *BCContract*, are created to realize the three layers in Figure 3, respectively. The *Interop* class employs the *ifcconvert* tool (ver. 0.6, available at: http://ifcopenshell.org/) to convert the IFC files to XML contents, and accepts IFCXML inputs as well. The difference between the two is that IFCXML is lossless from IFC but redundant, while *ifcconvert*'s XML export is concise but lossy. Then, the XML contents are reformatted to tree-like JSON objects using the *xmltodict* library (ver. 0.12). The Python native hashing function is used as the software oracle to represent an IFC instance's semantic "signature" if no other oracles are identified. The *SdtModel* class employs the *jsondiff* library (ver. 1.2, available at: https://github.com/xlwings/jsondiff) to compare the differences between the JSON objects. The *BCContract* class integrates local SDTs to homogeneous BCC. We implement a simplistic BCC mechanism by ticking out all the conflicting SDT records from the major version BIM. This simplistic BCC mechanism, rather than the complex contracts based on DAG, serves our proof-of-concept purpose.

5 Pilot study

5.1 Experimental settings

We employ two pilot cases to verify the proposed SDT approach. The first case, shown in Figure 7a, involves the architect as the only stakeholder. An IFC wall with a 750mm x 1400mm window at t_0 is changed to a 1400mm x 1400mm window (circled) at t_1 in this case. The GUIDs in the IFC files were de-randomized by pre-processing to mitigate the randomization in the computational tests. The second case, in Figure 7b, has two stakeholders, i.e., an architect and a client, co-editing roof windows in a sample project in Autodesk Revit 2018. First, at t_1 one window on the roof was moved towards the living room to capture daylight. The window was reverted to its original position by the client at t_2. The client noted “Please keep this” in the property “Comments” of the BIM object at t_3. Also at t_3, the architect added a new roof window for the living room. Clearly, this case is more sophisticated than the first because it creates new instances, changes non-geometric properties (e.g., text comments), and handles simultaneous changes. The IFC versions of both cases were IFC 2x Edition 3 (2x3). The models in the second case were exported to IFC immediately using Revit 2018’s native exporter once changed. We also conducted auxiliary tests on IFCXML formats exported from the same IFC models via xBIM Xplorer (ver. 4.0, https://docs.xbim.net /) Export function on a desktop computer with a 4-core Intel i5-6500 3.2GHz CPU and 8 GB memory. To avoid hard disk operation latency, a 500MB virtual hard disk was emulated in the memory.

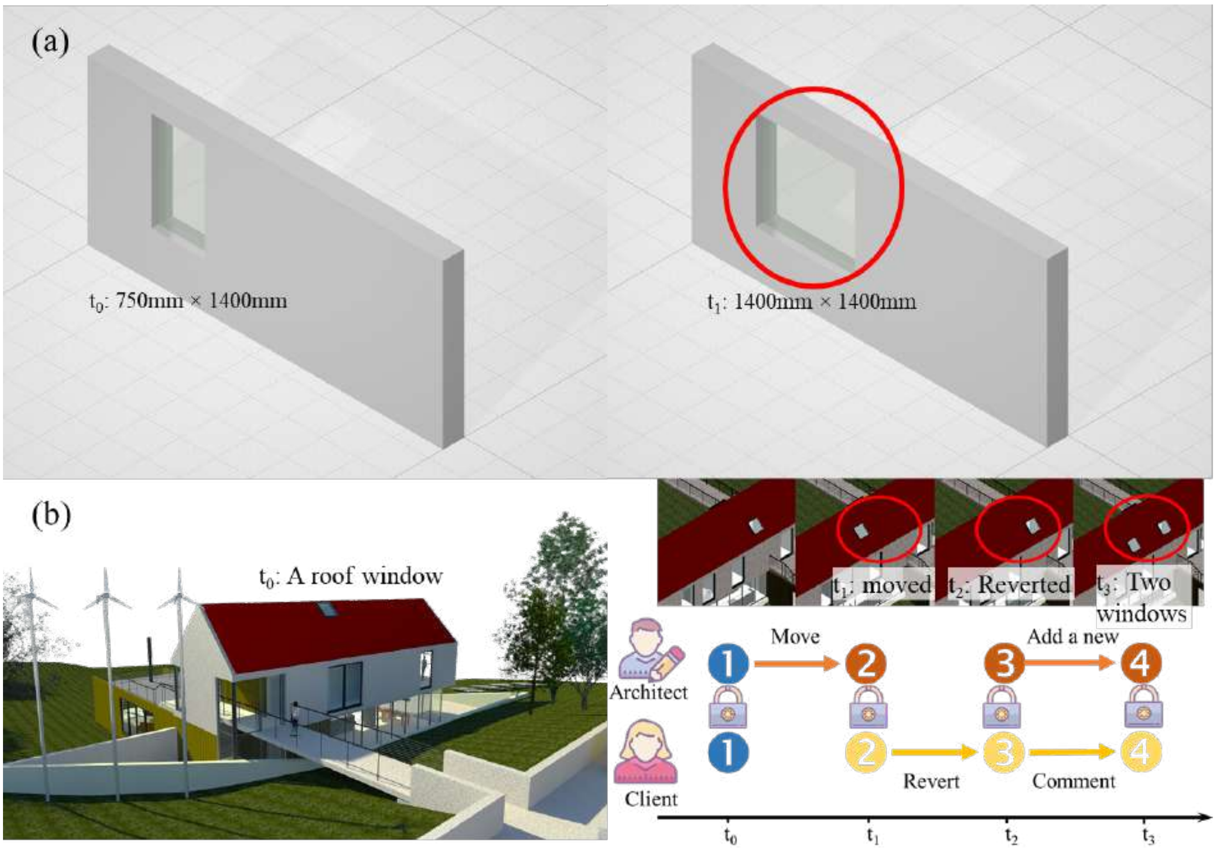


Figure 7. Two pilot IFC cases. (a) A wall model with a changed window; (b) Collaborative roof window design on a sample BIM project using Autodesk Revit 2018

5.2 Experimental results

Figure 8 shows the results of file difference and the SDT in the first case, already de-randomized. We tested two formats of IFC inputs. The first is IFC, in which each file is 7.4KB and has a *.ifc* extension. The SDT result was computed as a 0.36KB JSON object in 0.003s, as shown in Table

1. The JSON object correctly notes four semantic changes in the IFC, including file save time, two changes of lengths in *IfcElementQuantity* (i.e., one for the window and the other for the opening), and the *OverallWidth* of the only window. We compared the SDT results to the file comparison method, which has a 1.00KB result of 6 changed lines in the IFC files in 0.041s. In contrast, the IFCXML files with the "*.ifcxml*" extension are about four times larger than IFC on disk. The SDT result contains six changed values, as shown in Table 1. The result is 0.89KB in JSON, and the computational time 0.012s, four times that of the IFC test. The file comparison cost 0.042s for a 0.56KB difference of six changed lines. To sum up, the proposed SDT can correctly extract the semantic changes in IFC files, as well as IFCXML files, and achieve the first directional interoperability (i.e., from IFC to SDT).

Table 1. Comparison of the IFC file difference and SDT results in the first de-randomized case

Input	Item	Line-by-line file comparison	The proposed SDT
IFC	Size (KB)	1.00	0.36
(7.4KB	Time (s)	0.041	0.003
each)	SH?*	✗	✓

	Output	6 changed lines:	4 changed properties:
			{'
			header': {'file_name': {
			'time_stamp': ['2019-11-01T11:53:56', → '2019-11-0
			'quantities': {'IfcElementQuantity': {
			0: {'IfcQuantityLength': {
			1: {'@LengthValue': ['0.75', → '1.4']}}},
			1: {'IfcQuantityLength': {
			2: {'@LengthValue': ['0.75', → '1.4']}}}}},
			'decomposition': {'IfcProject': {'IfcSite': {
			'IfcBuilding': {'IfcBuildingStorey': {
			'IfcWindow': {
			'@OverallWidth': ['0.75', → '1.4']}}}}}}
			}

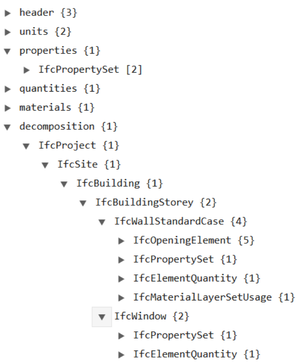
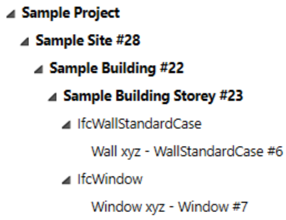
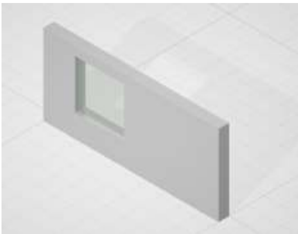
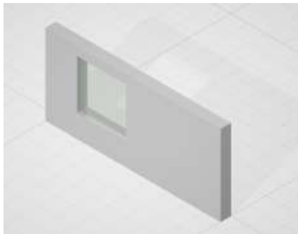
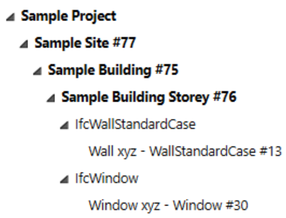
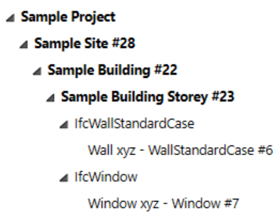
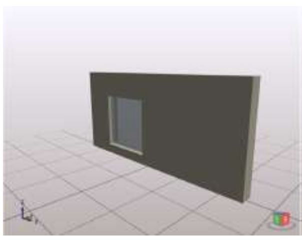
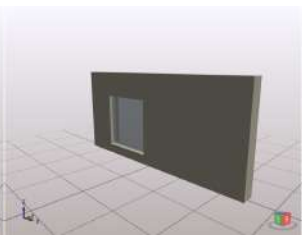
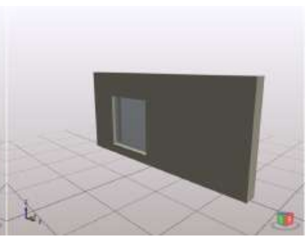
IFCXML	Size (KB)	0.56	0.89
(32.9KB	Time (s)	0.042	0.012
each)	SH?*	✗	✓

	Output	6 changed lines:
		5c5
		< <ex:time_stamp>2019-11-01T11:53:56</ex:time_star
		--- 	
		> <ex:time_stamp>2019-11-01T11:57:59</ex:time_star
		290c290
		< <IfcLengthMeasure pos="0">0.75</IfcLengthMeasure
		--- 	
		> <IfcLengthMeasure pos="0">1.4</IfcLengthMeasure>
		⋮
		421c421
		< <LengthValue>0.75</LengthValue>
		--- 	
		> <LengthValue>1.4</LengthValue>
			6 changed properties:
			{
			'ex:iso_10303_28_header':{
			'ex:time_stamp':['2019-11-01T11:53:56',→'2019-11-0
			'uos':{'IfcWindow':{'Representation':{'IfcProductDefinitic
			'Items':{'IfcExtrudedAreaSolid':{'SweptArea':{'IfcArbitra
			2:{'Coordinates':{'IfcLengthMeasure':{0:{
			'#text':['0.75',→'1.4']}}}},
			3:{'Coordinates':{'IfcLengthMeasure':{0:{
			'#text':['0.75',→'1.4']}}}}}}}}}}}}}}}},
			'OverallWidth':['0.75',→'1.4']},
			'IfcOpeningElement':{'Representation':{'IfcProductDefir
			'Items':{'IfcExtrudedAreaSolid':{'SweptArea':{'IfcArbitra
			2:{'Coordinates':{'IfcLengthMeasure':{0:{
			'#text':['0.75',→'1.4']}}}}}}}}}}}}}}}}},
			'IfcQuantityLength':{
			1:{'LengthValue':['0.75',→'1.4']}}}
			}

*: With semantic hierarchies?

Table 2 shows the results of the second directional interoperability, i.e., IFC restoration from SDT records, in the first case. The restoration utilizes the semantic hierarchy in SDT. In the semantic hierarchy, the extract positions of all the changes are recorded in a tree-like data structure. The restoration process is almost instant because of the small size of SDT records. All restoration tests were completed in less than 0.0001s, which was much faster than the SDT computation. With the IFC inputs, the restored XML of BIM semantics was 100% identical to the expected values, though the semantic hierarchy was reformatted. However, the conversion from XML to STEP failed due to lack of support from the *ifcconvert* library. With the IFCXML inputs, the semantic hierarchies are more consistent, and the restoration resulted in 100% correct IFC files in IFCXML and STEP formats. However, the correct files are not identical to the expected IFC files at the byte level. The restored XML output has 86.0% lines identical to those expected, while the restored STEP file has a mere 4.8%. The differences come from alternative expressions in XML syntax and the re-randomization of IFC instances' numbers (i.e., the STEP #-Ids). In short, the changed IFC files can be restored from small SDT records and a base model, particularly for IFCXML formats.

Table 2. Comparison of IFC restoration from SDT at t_1 in the first case

Input	Item	Restored BIM semantics (XML)	Restored IFC (STEP)	Ground truth IFC-STEP file
IFC	Time (s)	< 0.001	(Not supported by *ifcconvert*)	–
	Byte-level	100% identical		–
	Semantic level	100% identical		–
	Semantic hierarchy			
	3D view			
IFCXML	Time (s)	< 0.001	< 0.001	–
	Byte-level	86.0%* identical	4.8%# identical	–
	Semantic level	100% identical	100% identical	–
	Semantic hierarchy			
	3D view			

*: Due to flexible XML syntax, e.g., “<Tag></Tag>” and “<Tag />” are equivalent but different in bytes.
#: The STEP #-Ids in the “.ifc” files were re-randomized, e.g., Sample Site’s #28 was restored as #77.

The second case is very close to a real-world BIM project. Tests of the four local changes were conducted first. As listed in Table 3, each input IFC file exported from Autodesk Revit becomes about 27.4MB. The SDT approach spent around 6.66–7.00s (over 90% of the time) converting the input IFC models to JSON, i.e., algorithm Lines 1–2 in Figure 4. The results showed the SDT time consumption increased almost linearly from Case 1 to Case 2, i.e., from 0.003s for 7.4KB to 7.00s for 27.4MB, for IFC files based on *ifcconvert* function. The SDT computational time (algorithm Lines 3–6) is less than 0.5s, comparable with the file comparison method. The SDT results win in several aspects. First, there is minimal redundancy. For instance, local changes to the roof window (moving, reverting, and writing comments) were extracted as small (0.34–0.47KB) SDT outputs in JSON, while the addition of a new window was concluded as a 3.37KB output. All the SDT outputs were less than 0.02% of the IFC models, and small enough for blockchain systems. It is worth noting that the SDT outputs, even though small, incorporate the IFC semantic hierarchies. In contrast, the comparison of IFC files resulted in an unnecessary amount of changed lines and huge files without pre-processing for de-randomization. The sizes were almost twice the input file size in three out of four changes, indicating failures of meaningful change detection. We also tested Shi et al.'s (2018) *IFCdiff* method in the second case, with no result in any local changes in three hours. In summary, the SDT approach can effectively (correctly) and efficiently (in small file sizes and short time) detect local IFC changes.

Table 3. Results of IFC file difference and the proposed SDT in the second case

		Line-by-line file comparison			The proposed SDT			
Input	Change	Size (KB) (lines)	Time (s)*	SH?#	Size (KB)	Interop. time (s)*	SDT time (s)*	SH?#
IFC (27.4MB each)	$t_0 \rightarrow t_1$	11,400 (99,369)	0.398	✗	0.47	6.664	0.435	✓
	$t_1 \rightarrow t_2$	55,000 (538,443)	0.784	✗	0.47	6.641	0.463	✓
	$t_2 \rightarrow t_3$ (Arch.)	54,700 (533,923)	0.789	✗	3.37	6.681	0.414	✓
	$t_2 \rightarrow t_3$ (Client)	53,900 (514,192)	0.756	✗	0.34	7.004	0.411	✓
IFCXML (141.7MB each)	All$	(*Exceeded memory limit*)			(*Program halted by authors after waiting for three-hour execution*)			

*: Average of 10 runs; #: With semantic hierarchy or not?; $: All changes failed in the tests.

Similar crashes and failures were observed in the IFCXML tests for the second case. Neither the SDT approach nor the plain comparison method returned results in comparing the four pairs of 141MB IFCXML files. One key reason is that IFCXML is scrupulous but too lengthy. For example, an *IfcWindow*'s *ObjectPlacement* property is a 4x4 transformation matrix. That property can be a pre-computed finalized 4x4 matrix, such as "[-0.798636 -0.601815 0 0 … -18094.7 -16609.2 4610.17 1]" (111 bytes) in *IfcOpenShell*; in contrast, the same property in IFCXML included 106 XML lines (4,231 bytes) by referring to 4 instances of

IfcLocalPlacement, 4 instances of *IfcAxis2Placement3D*, 4 instances of *IfcCartesianPoint*, 3 instances of *IfcDirection*, and 12 instances of *IfcLengthMeasure*. By tracing the iterations of the failed SDT tests on the IFCXML inputs, we found the problem was an unexpectedly lengthy comparison task, which involved solving a longest common subsequence (LCS) problem between two lists of 140,833 *IfcCartesianPoints*. The complexity of the problem exceeded the classical algorithm's capacity, which has an $O(n^2)$ time complexity (billions of comparisons in this case). To sum up, the SDT approach using *ifcconvert* works for industrial-level IFC cases, while IFCXML inputs are appropriate for blockchaining small-scale BIM cases, but inappropriate for large-scale cases until novel comparison algorithms are developed.

Figure 8 shows the result of the BCC test for the second case. Between t_2 and t_3, the blockchain nodes of the architect and the client computed local SDT records. The architect's SDT records mainly involve four parts. The first is the changed time of file save; the next two are about the properties of the changed roof elements and the new roof window; and the final part describes the semantics of the new roof window instance, including all the properties and references. The client's SDT record, as shown in Figure 8, contains a short section of the newly added comment beside the changed time block. The final BCC is a 3.45KB JSON expression, integrating the blue and green parts into the IFC semantic hierarchy and excluding the conflicted date changes. The BCC on the IFC semantic hierarchy can be applied to compute the BIM model in consensus for all the stakeholders based on the IFC model on t_2.

Figure 8. Results of BIM change contract test for the second case ($t_2 \rightarrow t_3$).

5.3 Simulation on a minimal blockchain

We uploaded the experimental results in the second case on a minimalized blockchain for proof-of-concept validation of the compatibility of the SDT approach. The blockchain structure is a distributed blockchain with the essential functions run on a webpage (https://andersbrownworth.com/blockchain/distributed). As shown in Figure 9a, each blockchain peer independently stores the three BCCs in three blocks at t_1, t_2, and t_3. Each block refers to the previous one by including the previous hashing value, as indicated by the blue arrows in Figure 9a. As a result, the BIM changes, including the moving, reverting, addition, and comments, can be recorded with timestamps and managed in a distributed manner with minimal redundancy. The time series of BIM changes are fundamental for managing BIM versions. In addition, the blockchained BCCs become immutable. For example, Figure 9b shows that a falsification of BIM change can be detected at t_2 in the mismatch between the block content and its hashing value (underlined in red). Such BIM falsifications should be rare but possible, e.g., for claiming false authorships, destroying evidence, or being hacked. Nonetheless, the correct SDT blocks and the blockchain continued working among other peers in the consortium blockchain while the problematic peer was identified and refused by the consortium network.

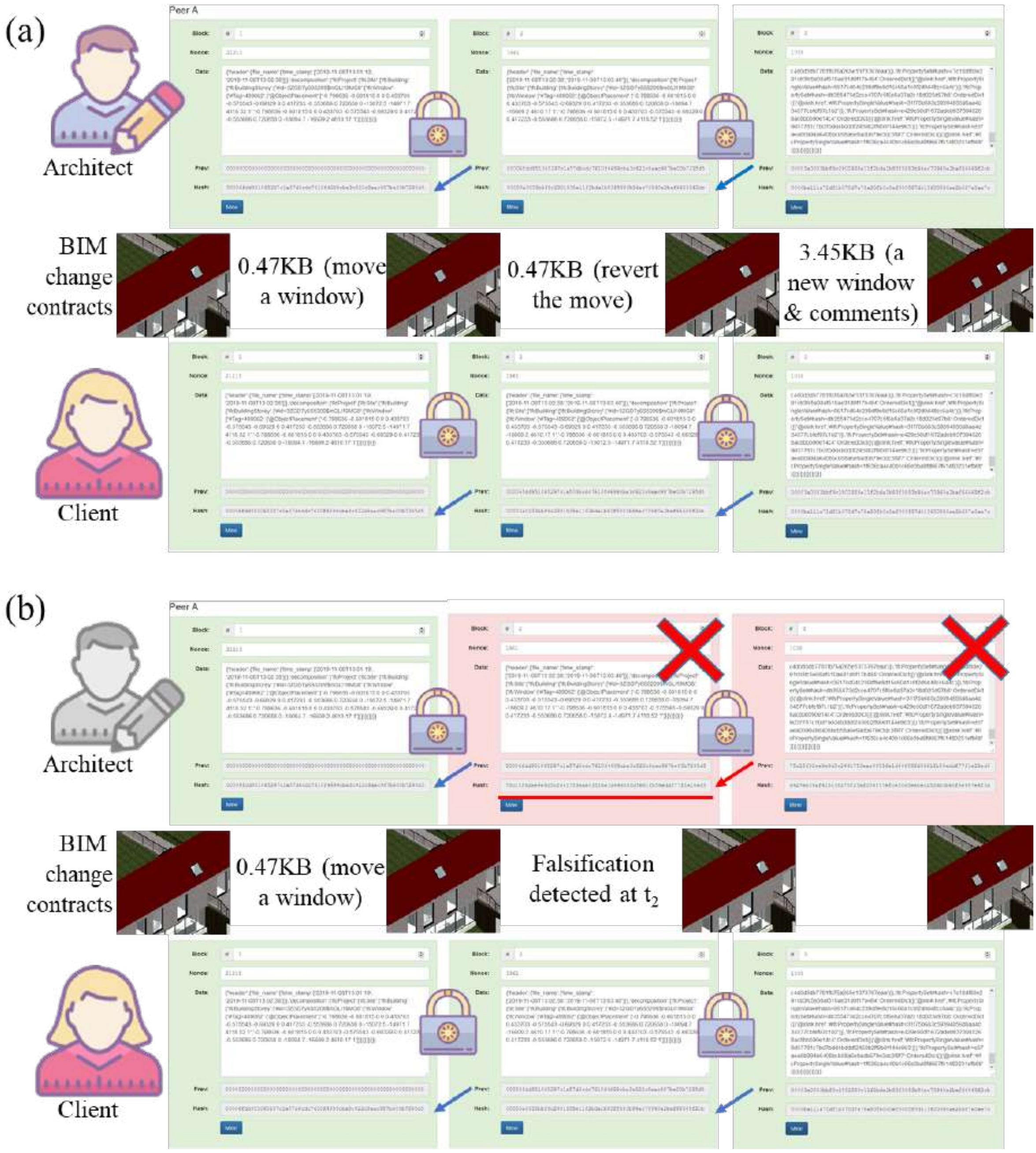


Figure 9. Simulation of SDT results in the second case on a minimal blockchain. (a) Distributed blockchain storage of BCCs; (b) Falsification detection

6 Discussion

There are five aspects to the novelty of our SDT approach, as follows.

- Firstly, the information safeguarded in a blockchain is significantly reduced by capturing BIM changes instead of entire BIM files. In our pilot tests, the version history of BIM changes was captured and placed in a blockchain with only around 0.02% of the BIM file size, satisfactorily addressing the challenge of information redundancy in BIM and blockchain integration.
- Secondly, our SDT approach possesses an elegant architecture with three succinct layers: (1) semantic interoperability; (2) SDT model; and (3) BCC mechanism. This

architecture and its included functions represent several original ideas not seen in previous research.

- Thirdly, our research takes IFC as a point of departure. IFC is the de facto open standard ensuring interoperability across different commercial BIM platforms and empowering open BIM. However, IFC has its shortcomings. One is the randomization of its identities, which adds to the difficulty of comparing and identifying BIM changes. The semantic interoperability layer of our SDT approach satisfactorily develops de-randomization functions and adopts modern data structures to allow bi-directional operations between IFC and blockchain. Specifically, SDT computation can be done in near real time, while IFC restoration from SDT is in real time.
- Also novel is the SDT core developed to identify the BIM changes and assemble them in a time series of SDT records. The algorithm of the SDT core is light and lean, suitable for performing heavy computation to identify BIM changes throughout its service life.
- Lastly, our research develops a BCC layer to realize the smart contract-type protocol in blockchain. This layer can deal with simultaneous BIM changes (i.e., SDT records) by different BIM stakeholders and reach a consensus on the global changes before integration into a blockchain.

Despite these innovations, our research is not free from limitations.

- Firstly, some parts of the SDT approach are not perfect, such as the conflict-resolving mechanisms to achieve the BCC. We expect to develop more sophisticated models such as DAG-based reasoning for the BCC in the future.
- Secondly, only limited pilot case studies were conducted. The experiments and the results, therefore, can only be treated as a proof of concept of the SDT approach, rather than a final version for benchmarking performance, or proof of extensibility and compatibility to other construction projects. Future tests should be conducted in more diverse projects.
- Thirdly, the pilot case studies were conducted on a distributed blockchain with basic functions running on a webpage. It is expected that future research should incorporate real blockchain shells, e.g., a permissioned consortium structure. On top of that, a relevant yet unexplored question is the types of blockchain (e.g., public or private) appropriate to a project-based setting such as that of construction.
- Next, the SDT approach is applicable to the IFC format. However, efficiency in computing IFCXML is not satisfactory for large-scale BIM projects. One reason is the $O(n^2)$ optimization of the LCS problem. With proper algorithmic modifications, such as an approximate algorithm returning 1% redundant results with a sheer $O(n \log n)$ time complexity, the approach can be applied to prevailing commercial BIM software platforms. Future research work can be directed to developing efficient IFCXML computation modules and plugins for these commercial BIM platforms as a way to promote BIM and blockchain integration.

- The SDT model in this paper focuses on a whole IFC file. Yet, the time spent for large-scale projects was still unsatisfactory, e.g., over 7 seconds for the tests on Case 2. We noticed that most of the time was consumed by the semantic interoperability layer to understand IFC files. One possible solution is to record the BIM changes directly from BIM software, e.g., Lin and Zhou's (2020) hashing code for Autodesk Revit, with a semantic interoperability add-in that monitors the BIM changes in real time. The de-randomization process in the semantic interoperability layer can be omitted when BIM software can offer a whole lifecycle GUID naming system for all types of IFC objects, including structural elements, materials, and relations.
- Lastly, we would like to stress that the SDT approach is not the only approach for minimizing information redundancy for BIM and blockchain integration. There are other approaches, such as open BIM web service (van Berlo 2015), the BCF standard, and the 'signature' of IFC objects (Shafiq & Lockley 2018) awaiting development.

7 Conclusion

By providing rich semantics of the physical and functional characteristics of a building to facilitate communication and decision-making amongst stakeholders, BIM can alleviate problems related to time, quality, cost, and productivity in construction. Also attractive to the construction industry is blockchain technology, which safeguards important information in immutable, cryptographic, and decentralized ledgers. The integration of BIM and blockchain has enormous potential to enable value-added applications but faces numerous technological hurdles, one of which is information redundancy. The volume of information in a BIM increases dramatically when developed and represented in IFC format, and then reaches an overwhelming level of redundancy when duplicated, encrypted, and distributed in blockchain. Minimizing this information redundancy is a fundamental challenge for BIM and blockchain integration.

This study reports a novel Semantic Differential Transition (SDT) model to capture and blockchain BIM changes instead of entire BIM files, thereby minimizing information redundancy and supporting BIM and blockchain integration. Our SDT approach has three function layers. First, the BIM interoperability layer extracts the BIM semantics from IFC files, applying de-randomization and modern data structures such as JSON objects. The SDT layer then computes the semantic difference, instead of file difference, in a short time and forms a set of local SDTs. The BCC layer offers blockchain a smart contract, e.g., DAG model of versions or designated subsystem editorships, to cope with sequential and simultaneous local SDTs. We demonstrated the proposed model in two IFC cases for blockchain BIM systems. The experimental results confirmed that SDT is effective (correct) and efficient (less than 0.02% BIM file size, in near real-time) for blockchain BIM systems. By following this innovative SDT approach, researchers and practitioners alike can develop truly operable BIM and blockchain integration solutions.

Future research work could improve this SDT approach. For example, the de-randomization and JSON objects are rather innovative but are more tied to IFC and STEP formats, which are involved in relatively ineffective identifier management. Perhaps in the long run, researchers need to work with IFC stakeholders to improve the consistencies for both BIM objects' GUIDs and STEP #-Ids ordering. Directed acyclic graph (DAG)-based reasoning could be a more accurate solution than that reported in this paper to realize BCC. More empirical tests on real-life BIM cases with different LoD and project complexities are expected to gauge the performance of the SDT approach further. Going beyond the SDT, domain-specific blockchain structures for construction projects could also be critical to realizing real-life blockchain BIM systems.

Acknowledgements

The authors would like to acknowledge the financial support by the Hong Kong Research Grant Council (Grant No. 17201717) and partial support by Department of Science and Technology of Guangdong, China (Grant No. 2019B010151001). We wish to express our gratitude to the anonymous reviewers for their constructive comments to improve the quality of the paper.

References

Afsari, K., Eastman, C. M. & Castro-Lacouture, D. (2017). JavaScript Object Notation (JSON) data serialization for IFC schema in web-based BIM data exchange. *Automation in Construction, 77*, pp. 24-51. doi:10.1016/j.autcon.2017.01.011

Ali, M. & Mohamed, Y. (2017). A method for clustering unlabeled BIM objects using entropy and TF-IDF with RDF encoding. *Advanced Engineering Informatics, 33*, pp. 154-163. doi:10.1016/j.aei.2017.06.005

Allam, Z. & Jones, D. S. (2019). The potential of blockchain within air rights development as a prevention measure against urban sprawl. *Urban Science, 3*(1), Article ID 38. doi:10.3390/urbansci3010038

Beck, R., Czepluch, J. S., Lollike, N. & Malone, S. (2016). Blockchain–The gateway to trust-free cryptographic transactions. *Proceedings of 2016 European Conference on Information Systems*. Association for Information Systems. Retrieved March 9, 2020, from https://aisel.aisnet.org/ecis2016_rp/153

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 B. A., K. M., K. C. & B. J., *Building Information Modeling* (pp. 81-126). Cham, Switzerland: Springer. doi:10.1007/978-3-319-92862-3_5

Breitbart, Y., Olson, P. L. & Thompson, G. R. (1986). Database integration in a distributed heterogeneous database system. *1986 IEEE Second International Conference on Data Engineering* (pp. 301-310). IEEE. doi:10.1109/ICDE.1986.7266234

buildingSMART. (2019). *Industry Foundation Classes Version 4.2 bSI Draft Standard*. Retrieved March 9, 2020, from https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/

buildingSMART. (2020). *IFC Formats.* buildingSMART International. Retrieved March 9, 2020, from https://technical.buildingsmart.org/standards/ifc/ifc-formats/

Chen, K., Lu, W., Wang, H., Niu, Y. & Huang, G. G. (2017). Naming objects in BIM: A convention and a semiautomatic approach. *Journal of Construction Engineering and Management, 143*(7), Article ID 06017001. doi:10.1061/(ASCE)CO.1943-7862.0001314

Dakhli, Z., Lafhaj, Z. & Mossman, A. (2019). The Potential of Blockchain in Building Construction. *Buildings, 9*(4), Article ID 77. doi:10.3390/buildings9040077

Du, J., Zou, Z., Shi, Y. & Zhao, D. (2018). Zero latency: Real-time synchronization of BIM data in virtual reality for collaborative decision-making. *Automation in Construction, 85*, pp. 51-64. doi:10.1016/j.autcon.2017.10.009

Eastman, C. M., Eastman, C., Teicholz, P., Sacks, R. & Liston, K. (2011). *BIM Handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors* (2nd ed.). ISBN 9780470541371, John Wiley & Sons.

Egan, J. (1998). *Rethinking construction.* London, UK: Department of the Environment, Transport and Regions. Retrieved April 19, 2020, from https://webarchive.nationalarchives.gov.uk/20040722115001/http://www.dti.gov.uk/construction/rethink/report/index.htm

Ellis, M. (2019, July 12). *Level of Detail or Development: LOD in BIM*. Retrieved November 6, 2019, from REBIM: https://rebim.io/level-of-detail-or-development-lod-in-bim/

Froese, T. (2003). Future directions for IFC-based interoperability. *Journal of Information Technology in Construction, 8*(17), pp. 231-246. Retrieved March 9, 2020, from https://www.itcon.org/2003/17

Gao, G., Liu, Y. S., Lin, P., Wang, M., Gu, M. & Yong, J. H. (2017). BIMTag: Concept-based automatic semantic annotation of online BIM product resources. *Advanced Engineering Informatics, 31*, pp. 48-61. doi:10.1016/j.aei.2015.10.003

Gipp, B. K. & Breitinger, C. (2016). Securing video integrity using decentralized trusted timestamping on the Bitcoin blockchain. *Proceedings of 2016 Mediterranean Conference on Information Systems*, (p. 51). Retrieved March 9, 2020, from https://aisel.aisnet.org/mcis2016/51

Hawlitschek, F., Notheisen, B. & Teubner, T. (2018). The limits of trust-free systems: A literature review on blockchain technology and trust in the sharing economy. *Electronic Commerce Research and Applications, 29*, pp. 50-63. doi:10.1016/j.elerap.2018.03.005

ICE. (2019). *Supply chains in construction.* Institute of Civil Engineers. Retrieved March 9, 2020, from https://www.designingbuildings.co.uk/wiki/Supply_chains_in_construction

Jäger, M. (2018). *VCS 4 CDE–Version control systems as common data environments.* Munich, Germany: Technical University of Munich. Retrieved March 9, 2020, from https://www.cms.bgu.tum.de/images/teaching/abim_seminar/Report_J%C3%A4ger_VCS_as_CDE.pdf

Jørgensen, K., Skauge, J., Christiansson, P., Svidt, K., Sørensen, K. B. & Mitchel, J. (2008). *Use of IFC model servers: Modelling collaboaration possibilities in practice.* Aalborg, Denmark: Aalborg University. Retrieved March 9, 2020, from https://www.kaj.person.aau.dk/digitalAssets/199/199580_12176_reportifcmodelserver-final.pdf

Jung, Y. & Joo, M. (2011). Building information modelling (BIM) framework for practical implementation. *Automation in Construction, 20*(2), pp. 126-133. doi:10.1016/j.autcon.2010.09.010

Kinnaird, C., Geipel, M. & Bew, M. (2017). *Blockchain technology: how the inventions behind bitcoin are enabling a network of trust for the built environment.* London, UK: Arup. Retrieved February 1, 2020, from https://www.arup.com/-/media/arup/files/publications/b/arup--blockchain-technology-report_comp.pdf

Laan, A., Noorderhaven, N., Voordijk, H. & Dewulf, G. (2011). Building trust in construction partnering projects: An exploratory case-study. *Journal of Purchasing and Supply Management, 17*(2), pp. 98-108. doi:10.1016/j.pursup.2010.11.001

Larsson, J., Eriksson, P. E., Olofsson, T. & Simonsson, P. (2014). Industrialized construction in the Swedish infrastructure sector: core elements and barriers. *Construction Management and Economics, 32*(1-2), pp. 83-96. doi:10.1080/01446193.2013.833666

Lee, G., Jeong, J., Won, J., Cho, C., You, S. J., Ham, S. & Kang, H. (2014). Query performance of the IFC model server using an object-relational database approach and a traditional relational database approach. *Journal of Computing in Civil Engineering, 28*(2), pp. 210-222. doi:10.1061/(ASCE)CP.1943-5487.0000256

Lee, G., Won, J., Ham, S. & Shin, Y. (2011). Metrics for quantifying the similarities and differences between IFC files. *Journal of Computing in Civil Engineering, 25*(2), pp. 172-181. doi:10.1061/(ASCE)CP.1943-5487.0000077

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*, pp. 288-307. doi:10.1016/j.autcon.2019.02.005

Li, N., Li, Q., Liu, Y. S., Lu, W. & Wang, W. (2020). BIMSeek++: Retrieving BIM components using similarity measurement of attributes. *Computers in Industry, 116*, Article ID 103186. doi:10.1016/j.compind.2020.103186

Lin, J. R. & Zhou, Y. C. (2020). Semantic classification and hash code accelerated detection of design changes in BIM models. *Automation in Construction, 115*, Article ID 103212. doi:10.1016/j.autcon.2020.103212

Lu, W., Lai, C. C. & Tse, T. (2018). *BIM and big data for construction cost management.* New York, NY, USA: ISBN 9781351172301, Routledge.

Molloy, O., Warman, E. A., Tilley, S. & . (2012). *Design for Manufacturing and Assembly: Concepts, architectures and implementation.* ISBN 9780412781902, Springer. doi:10.1007/978-1-4615-5785-2

NIBS. (2015). *National BIM Standard - United States V3.* National Institute of Building Sciences, USA. Retrieved March 9, 2020, from https://www.nationalbimstandard.org/

Notheisen, B., Cholewa, J. B. & Shanmugam, A. P. (2017). Trading real-world assets on blockchain. *Business & Information Systems Engineering, 59*(6), pp. 425-440.

doi:10.1007/s12599-017-0499-8
Oraskari, J. & Törmä, S. (2015). RDF-based signature algorithms for computing differences of IFC models. *Automation in Construction, 57*, pp. 213-221. doi:10.1016/j.autcon.2015.05.008
Özsu, M. T. & Valduriez, P. (2020). *Principles of distributed database systems.* Cham, Switzerland: ISBN 9781441988331, Springer. doi:10.1007/978-1-4419-8834-8
Penzes, B. (2018). *Blockchain technology in the construction industry: Digital transformation for high productivity.* London, UK: Institute of Civil Engineers. Retrieved March 9, 2020, from https://www.ice.org.uk/ICEDevelopmentWebPortal/media/Documents/News/Blog/Blockchain-technology-in-Construction-2018-12-17.pdf
Perera, S., Nanayakkara, S., Rodrigo, M. N., Senaratne, S. & Weinand, R. (2020). Blockchain technology: Is it hype or real in the construction industry? *Journal of Industrial Information Integration, 17*, Article ID 100125. doi:10.1016/j.jii.2020.100125
Pratt, M. J. (2004). Extension of ISO 10303, the STEP standard, for the exchange of procedural shape models. *Proceedings of Shape Modeling Applications 2004* (pp. 317-326). IEEE. doi:10.1109/SMI.2004.1314519
PwC. (2018). *Global Blockchain Survey 2018: Blockchain is here. What's your next move?* PricewaterhouseCoopers (PwC) China. Retrieved February 1, 2020, from https://pwc.to/37LRuMK
Risius, M. & Spohrer, K. (2017). A blockchain research framework. *Business & Information Systems Engineering, 59*(6), pp. 385-409. doi:10.1007/s12599-017-0506-0
SCMP. (2017, May 23). Officers arrest 21 over faked concrete test results for Hong Kong-Zhuhai-Macau bridge project. *South China Morning Post*. Retrieved March 9, 2020, from https://www.scmp.com/news/hong-kong/law-crime/article/2095389/officers-arrest-21-over-faked-concrete-test-results-hong
SCMP. (2019, May 28). Main contractor at scandal-hit Hung Hom MTR station accused of using wrong rebar, as inquiry of Hong Kong government-appointed commission continues. *South China Morning Post*. Retrieved March 9, 2020, from https://www.scmp.com/news/hong-kong/transport/article/3012015/main-contractor-scandal-hit-hung-hom-mtr-station-accused
Shafiq, M. & Lockley, S. (2018). Signature-based matching of IFC models. *Proceedings of the 35th International Symposium on Automation and Robotics in Construction* (pp. pp. 993-1001). IAARC. doi:10.22260/ISARC2018/0138
Shi, X., Liu, Y. S., Gao, G., Gu, M. & Li, H. (2018). IFCdiff: A content-based automatic comparison approach for IFC files. *Automation in Construction, 86*, pp. 53-68. doi:10.1016/j.autcon.2017.10.013
Taylor, J. & Bernstein, P. (2009). Paradigm Trajectories of Building Information Modeling Practice in Project Networks. *Journal of Management in Engineering, 25*(2), pp. 69-76. doi:10.1061/(ASCE)0742-597X(2009)25:2(69)
van Berlo, L. (2015). *BIM Service interface exchange (BIMSie).* National Institute of Building

Science, USA. Retrieved March 9, 2020, from https://www.nibs.org/page/bsa_bimsie

Wang, H. & Meng, X. (2019). Transformation from IT-based knowledge management into BIM-supported knowledge management: A literature review. *Expert Systems with Applications, 121*, pp. 170-187. doi:10.1016/j.eswa.2018.12.017

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*, Article ID 103063. doi:10.1016/j.autcon.2019.103063

Wüst, K. & Gervais, A. (2018). Do you need a blockchain? *Proceedings of 2018 Crypto Valley Conference on Blockchain Technology* (pp. 45-54). IEEE. doi:10.1109/CVCBT.2018.00011

Xia, Q. I., Sifah, E. B., Asamoah, K. O., G. J., Du, X. & Guizani, M. (2017). MeDShare: Trust-less medical data sharing among cloud service providers via blockchain. *IEEE Access, 5*, pp. 14757-14767. doi:10.1109/ACCESS.2017.2730843

Xue, F., Chen, K., Lu, W., Niu, Y. & Huang, G. Q. (2018a). Linking radio-frequency identification to Building Information Modeling: Status quo, development trajectory and guidelines for practitioners. *Automation in Construction, 93*, pp. 241-251. doi:10.1016/j.autcon.2018.05.023

Xue, F., Lu, W. & Chen, K. (2018b). Automatic generation of semantically rich as-built Building Information Models using 2D images: A Derivative-Free Optimization approach. *Computer-Aided Civil and Infrastructure Engineering, 33*(11), pp. 926-942. doi:10.1111/mice.12378

Zheng, R., Jiang, J., Hao, X., Ren, W., Xiong, F. & Ren, Y. (2019). bcBIM: A blockchain-based big data model for BIM modification audit and provenance in mobile cloud. *Mathematical Problems in Engineering, 25*, Article ID 5349538. doi:10.1155/2019/5349538