Hindawi
Advances in Civil Engineering
Volume 2023, Article ID 3951336, 13 pages
https://doi.org/10.1155/2023/3951336

Research Article

Solutions on Establishing and Utilizing BIM for General Contractors in Key Scenarios of Digital Construction

Yang Peng, Fangqiang Yu, Ming Zhang, Jinglin Xu, and Shang Gao

Shanghai Construction No. 4 (Group) Limited Company, Shanghai 201103, China

Correspondence should be addressed to Fangqiang Yu; fqyu007@163.com

Received 31 May 2022; Revised 19 October 2022; Accepted 30 June 2023; Published 27 July 2023

Academic Editor: Donato Ciampa

Nowadays, digital construction has become popular when bringing convenience and efficiency to the traditional building construction industry. The primary tool of digital construction is the building information model (BIM). However, from the perspective of general contractors, unresolved puzzles still hinder obtaining the benefits of digital construction. When establishing a unified BIM from submodels provided by subcontractors, there will possibly be incomplete or inconsistent data during model merging. But extracting submodels from unified BIM often includes redundant data, thus making models less usable for subcontractors. It is also difficult for general contractors to effectively and accurately utilize resource information and submodel changes. This paper proposed solutions that depend on the widely adopted industry foundation classes standard to ensure the universality of our methods. First, a model merging algorithm is proposed to support the continuous merge of submodels created by different subcontractors. Second, an instance-level model extraction method based on strongly related entities is proposed, which extracts model instances to the minimum submodel and meets the subcontractor requirements at the same time. Third, the new model storage and indexing method are designed to reduce the complexity of model data and support rapid data retrieval, and a new BIM change detection method based on object metadata is provided. The proposed methods were applied by the general contractor of a large airport project during the construction stage. The application results proved that the proposed methods could ensure the quality of established deepening design models and extracted submodels and significantly reduce human labor by improving efficiency when utilizing BIM, which in turn supported key scenarios throughout the digital construction workflow.

1. Introduction

Digital construction is characterized by the use of data and information technologies, which improves efficiency and reduce cost for the application scenarios in the whole lifecycle of buildings [1], including design, construction, maintenance, and demolition phases [2]. The main work of digital construction is currently based on the building information model (BIM). The core purpose is to build a digital virtual building containing complete information about the construction process and provide the desired information for each application scenario [3]. Currently, BIM-based methods have been fully developed, but academic researchers still focus on some key scenarios in digital construction.

The three essential benefits of BIM-based digital construction are accuracy, consistency, and integrity. However, they have also featured the main difficulties in the establishment and application of BIM in the building lifecycle [4]. A general contractor is responsible for managing a unified BIM and providing usable model information to subcontractors. Therefore, ensuring accurate, consistent, and complete construction information in the whole process, from the establishment to the use of BIM, is the core issue that directly decides the success of the practice of digital construction scenario.

From the perspective of the digital department of general contractor companies, this paper studies the difficulties of establishing and utilizing BIM in key scenarios during the digital construction process. To ensure the generality of the method, BIMs in this paper are defined under the framework of industry foundation classes (IFC) because it is now the only de facto standard for BIM data exchange. Therefore, the following sections will define three key digital construction scenarios and analyze existing problems that general contractors will face.

1.1. Merging Design Models from Subcontractors and Departments. The most crucial digital work for general contractors is establishing a unified BIM and maintaining its integrity from the design phase to final delivery. From the architectural design phase, subcontractors and departments create different BIM submodels of the building. For the multidisciplinary coordination of digital construction and subsequent construction organization, merging these submodels into one final BIM model is necessary. For example, the designer's architectural and structural design models are merged into a full component model for subsequent electromechanical professional design. When merging submodels, it is necessary to ensure the integrity and avoid repetition of information as well, including (1) maintaining consistent modification information of components and (2) correctly reconstructing the association relationship, both essential.

Problem: During merging operations of designing models, the "relationship entities" in the To-merge model are likely to have the problem of missing secondary entity sets and cannot be directly added. Otherwise, the final information will be incomplete. Therefore, to ensure the consistency of the model, the algorithm must carefully analyze the changes of instances in the model extraction, modification, and merging process. Currently, some merge methods do not consider these problems, resulting in difficulties in applying them in practice.

1.2. Extraction of Usable Submodels. After establishing the unified BIM, the general contractors are responsible for providing submodels for various on-site management use. For example, the model of the first floor in this month's schedule is extracted for progress analysis. Often the model of all steel materials is extracted and delivered to the steel structure subcontractor. Submodel extraction has two levels: entity-level and instance-level. Instance-level extraction includes filtering conditions for a particular component instance, which require attributes of instances to meet the filter condition. For example, an entity-level extraction requires "material-steel" and "type-beam" so that only steel beams are extracted while concrete beams are eliminated. In the digital construction scenario, the requirement for instance-level model extraction is considerably large [5].

Problem: Difficulties arise during the processing of related model entities. The extracted model becomes very large and useless if too many associated objects are extracted directly. For example, the progress department requires a workflow submodel containing "structural construction." If the subtasks, pretasks, and posttasks associated with the "structural construction" task are all extracted, redundant task nodes will also be included. It is difficult to specify the extraction rules of relational entities in the submodel to ensure the extracted submodel meets the requirements and retains minimum at the same time.

1.3. Fast and Accurate Information Acquisition from BIM. Establishing proper BIM is not the end of digital construction. Instead, acquiring information from BIM is the final goal. A general contractor often needs to obtain various information during project management. After merging submodels, the general contractor expects to quickly summarize the general information on labor, materials, machinery, and other resources. This requires suitable information storage and indexing strategies for rapid search. Most backends use a relational database to store BIM and additional application information. The representative format of IFC is object-oriented. If the IFC objects are stored directly according to their structure, information exchange becomes convenient, but real-time search and query from BIM software will become dubious [6].

After subcontractors submit their submodels, there are massive model changes in the deepening design, subsequent construction cost, material procurement, and construction plan [7]. Therefore, the general contractor hopes to obtain the changes between old and new models, including the contents and the reasons for changes. Among them, the reasons for changes are more critical, and the general contractor will screen and classify accordingly and send the changes to the department for further processing. In addition, the entities of the IFC model have well-defined metadata structures [8]. Therefore, more meaningful detection results will be given if taking full advantage of this structure.

Problem: Referring to existing model storage and indexing methods, time consumption originates from searching and joining many tables. This leads to low efficiency when retrieving entities in an IFC-based database. For example, data from geometry and resource tables must be queried to extract the information of a beam instance. In addition, existing methods cannot classify model changes by the cause of changes, which makes it difficult for users to obtain helpful information [9]. Many insignificant small changes make it hard for users to locate the changes they care about. For example, hundreds of air conditioning pipes are connected. If one pipe changes, it will change other related pipes through relational entities. But these relationship changes should not be reported when doing quantity takeoffs.

2. Related Works

Similar studies on merging, extraction, and information acquisition of IFC or BIM exist. Some related works are compared in Table 1.

The most significant gap in the existing research is that they focus on software engineering but do not emphasize the application of general contracting. More papers discussed merging external information into BIM [16, 17], dealing with various data formats like JSON, images, and OBJ [18]. Merging one submodel into another model seems less concerning. The general contractors hope to solve inconsistency from subcontractors' submodels. The merging problem is caused by management naturally, not by software barriers. The same gap exists in model extraction methods, mainly geometric or topological data extracted for other calculation software [19, 20], but few focus on an entirely usable model for subcontractors or department users. As for information

Table 1: A nonexhaustive list of related works.

Related work	IFC-based	Submodel merging	Submodel extraction	Information acquisition	
				Model indexing	Change detection
Doukari et. al. [10]	√	▲			
Ren et. al. [11]	√		▲	△	
Xu et. al. [12]	√		▲	▲	
Lin et. al. [13]	√		▲	△	
Chen et. al. [14]	×		△	▲	
Shi et. al. [9]	√				▲
Lin et. al. [15]	×		△		▲
This paper	√	▲	▲	△	△

Note. ▲: as the main proposed method; △: as one of the supplementary methods.

acquisition, the need for general contractors was only partially satisfied.

There are reports of various cloud-based approaches to address the storage and search of large-scale IFC models [14, 21]. Based on the cloud storage framework, it is easier to manipulate BIM entities among different remote stakeholders and execute intelligent queries on the server side. However, little research has been done to optimize resource information storage (labor, materials, machinery), which is more concerned by general contractors. Some new change detection algorithms were proposed. For example, IFCdiff [9] and the IFC hash acceleration [15] both reported significant improvement in time complexity and accuracy. However, again, these researchers focused on optimizing computing algorithms, not digging into the root cause of changed contents. Therefore, it remains a problem for general contractors to classify changes that matter to different subcontractors.

3. Methodology

3.1. IFC, Model View Definition (MVD), and Other Related Concepts. IFC standard, particularly the IFC4 version, is an open and standardized data storage format for building lifecycle modeling. Popular BIM modeling software support exporting IFC-format files for storage and exchange. IFC represents data in an object-oriented manner and stores various building components and their relationships with instances of various types of IFC entities. Each IFC entity has many attributes for storing information about building components.

The submodel is a subset of the IFC model. In digital construction scenes, it is necessary to extract a simplified submodel for a specific construction process or an area of particular concern. A standard format defines the submodel–MVD, which is a combination of some entity restrains and some instance filtering conditions [22], usually appearing as an independent eXtensible Markup Language text file [23].

Next, some technical terms of IFC are briefly introduced. IFC entities can be divided into independent exchangeable entities and nonexchangeable (resource) entities. With the globally unique identifier "GlobalID," exchangeable entities can be retrieved, extracted, and shared separately. Resource entities only have temporary IDs and cannot be independently extracted and shared between different software. Independent exchangeable entities have complex inheritance trees and form the central part of the IFC scheme.

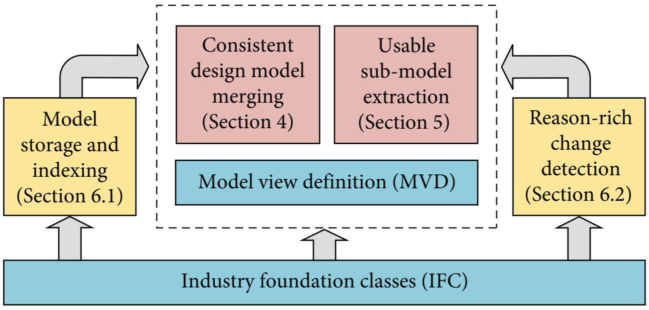
Consistent design model merging (Section 4)
Usable sub-model extraction (Section 5)
Model storage and indexing (Section 6.1)
Reason-rich change detection (Section 6.2)
Model view definition (MVD)
Industry foundation classes (IFC)

Figure 1: The framework of the proposed method.

"Process entity" and "product entity" belong to the basic entities in the independently exchangeable entities. Process entities describe the process, task definition, and schedule information in construction management. Product entities describe physical components in buildings, such as beams, columns, walls, etc., including geometric and nongeometric attributes [24].

Another kind of "relational entity" amongst independent exchangeable entities describes the relationship between basic entities. The relationship between basic entities cannot be established through direct mutual reference but must be connected through relational entities. This makes the entity-relationship clearer and can be adequately used. A relationship entity records the relationship between an entity E (called "primary entity") and a set of entities Bs (called "secondary entity set"). Accordingly, entity E owns a "reverse attribute" pointing to all related entities related to itself.

3.2. Method Framework and Novelty. As shown in Figure 1, this paper proposes solutions to address the problems mentioned above that general contractors will face in digital construction. First, the IFC standard lays the foundation, and BIM data are represented by IFC. Then two core problems about establishing BIM are solved based on MVD: submodel merging and extraction. After that, two information acquisition methods are provided, as complementary approaches, to help better utilize BIM for general contractor companies. The model indexing method helps to search and summarize project

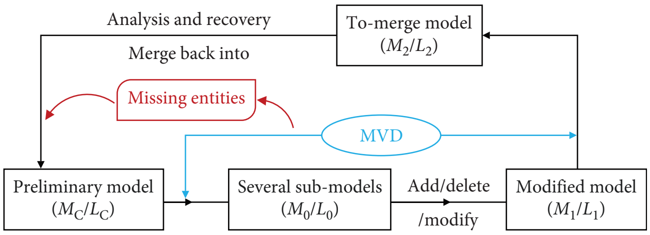
Analysis and recovery
Merge back into
To-merge model (M_2/L_2)
Missing entities
MVD
Preliminary model (M_C/L_C)
Several sub-models (M_0/L_0)
Add/delete /modify
Modified model (M_1/L_1)

Figure 2: The processing logic of submodel merging.

resources sufficiently, and the change detection method helps compare models before and after submodel merging.

There are some novelties achieved from the proposed method: (1) a submodel merging method is proposed to support the continuous merge of submodels created by different users and help establish a complete digital construction model. The method also solves the problem of possibly incomplete or inconsistent data. (2) An instance-level model extraction method based on the characteristics and strong correlation of various entities in the model is proposed to ensure that the extracted instance set is the minimum submodel to meet the use requirements. (3) A model storage method is designed to meet the needs of rapid data retrieval. Then a new IFC model change detection method based on object metadata is provided, which can give the causes of changes and improve efficiency.

4. Consistent Merge of Design Models

This section describes the proposed BIM model merging method based on IFC and MVD. The overall logic of this method is illustrated in Figure 2. The proposed method can maintain the model's integrity by merging models from different stakeholders with the original design model.

Step 1: Make an entity list L_0 of the submodel: from the IFC preliminary design model (the starting point in Figure 2), extract the required IFC submodel M_0, including the IFC instance list $L_0 = \{e_t, \{o_{tj}\}\}$, where e_t is the IFC entity name, and $o = \{o_{tj}\}$ is the collection of IFC instances with entity type e_t.

Step 2: Read the modified IFC submodel and create instance list L_1: read in the IFC submodel M_1 to be merged, and similarly establish the IFC instance list $L_1 = \{(e_i, \{o_{ij}\})\}$; at the same time, read the MVD as $V = \{e_m\}$, where V describes those IFC entities to be merged in L_1.

Step 3: Calculate the IFC instance list L_2 to be merged: traverse each class e_m in V. Then extract all instances $\{o_{mj}\}$ under the class e_m from the instance list L_1 of M_1, and compose the instance list $L_2 = \{(e_m, \{o_{mj}\})\}$. If L_1 does not contain the element under e_m, an empty pair $(e_m, \{\})$ will be added to L_2. The IFC submodel M_2, a subset of M1, has been established and is ready to merge with MC.

Step 4: Analysis of the modification status of IFC instances: in this step, L_0 and L_2 are compared to classify instances by four modification statuses: deleted, added, refreshed, and unchanged. First, for each $(e_i, \{o_{ij}\})$ in L_0, find $(e_m, \{o_{mj}\})$ in L_2, where $e_i = e_m$. If o_{ij} has no GlobalID match in $\{o_{mj}\}$, it is marked "deleted" and put in $\{o_{mj}\}$. Then, for each $(e_m, \{o_{mj}\})$ pair in L_2, find $(e_i, \{o_{ij}\})$ pair in L_0 where $e_m = e_i$. For each o_{mj}, if it has no GlobalID match in $\{o_{ij}\}$, it is marked "added." But when a GlobalID match can be found, the OwnerHistory is compared. If they have inconsistent OwnerHistory, o_{mj} is marked "refreshed;" otherwise, "unchanged." Finally, if e_i cannot be found in the first place, $\{o_{mj}\}$ will all be marked "added."

Step 5: Recover information of entities in the submodel: this is the most critical step to avoid incomplete data and maintain the model's integrity. Referring to Figure 2, some entities are probably missed from M_C to M_0, which means M_2/L_2 is incomplete. Therefore each $(e_m, \{o_{mj}\})$ in L_2 is recovered as follows. If e_m is an IFC basic entity, its relationships should be recalculated to recover its reversed attributes. If e_m is an IFC relationship entity, its related entities should be recalculated to recover its secondary entity set. Details of this process are shown in Figure 3.

Step 6: Merge M_2 into M_C: M_2/L_2 is now ready to be merged into the preliminary M_C model. For every $(e_m, \{o_{mj}\})$ pair in L_2, find $(e_k, \{o_{kj}\})$ in L_C where $e_k = \text{em}$. If one cannot find any e_k, $(e_m, \{o_{mj}\})$ should be directly added into L_C. After that, each o_{mj} is processed by its modification status. If o_{mj} is marked "added," it will be directly added into $\{o_{kj}\}$. If o_{mj} is marked "refreshed," it will replace the instance with the same GlobalID in $\{o_{kj}\}$. If o_{mj} is marked "deleted," the instance with the same GlobalID in $\{o_{kj}\}$ will also be deleted. Those "deleted" instances are invisible in BIM software.

By recording the instance list of extracted submodel, the instance modification status of the IFC submodel is analyzed, especially to merge the "added," "deleted," and "refreshed" status into the preliminary model.

5. Minimum Usable Submodel Extraction

Theoretically, the elements in the BIM model are all interrelated, so additional relevant instances must be added when extracting the submodel. But the amount of additional instances is challenging to determine. The final result's size will be too large to use if too many related entities are extracted. On the contrary, there may be missed but valuable instances. As mentioned, the result of class-level extraction always contains redundant objects. This section proposes a new extraction strategy by the "core instance" concept to obtain a minimum usable instance-level model.

The submodel M is extracted from the original model M_0.

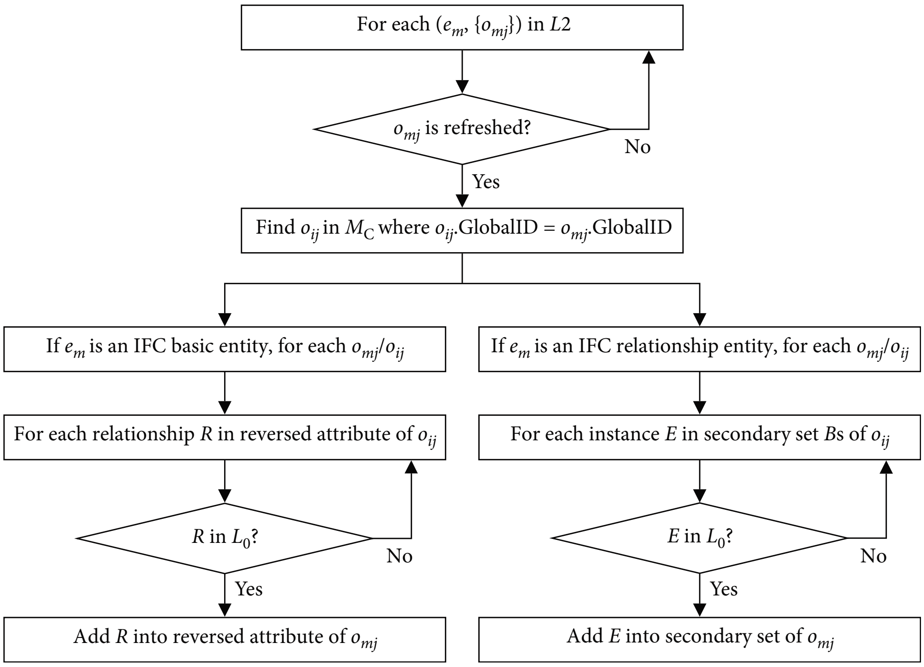
For each (e_m, {o_{mj}}) in $L2$
o_{mj} is refreshed?
No
Yes
Find o_{ij} in M_C where o_{ij}.GlobalID = o_{mj}.GlobalID
If e_m is an IFC basic entity, for each o_{mj}/o_{ij}
If e_m is an IFC relationship entity, for each o_{mj}/o_{ij}
For each relationship R in reversed attribute of o_{ij}
For each instance E in secondary set Bs of o_{ij}
R in L_0?
No
Yes
E in L_0?
No
Yes
Add R into reversed attribute of o_{mj}
Add E into secondary set of o_{mj}

Figure 3: The flow of recovering information of entities in submodel.

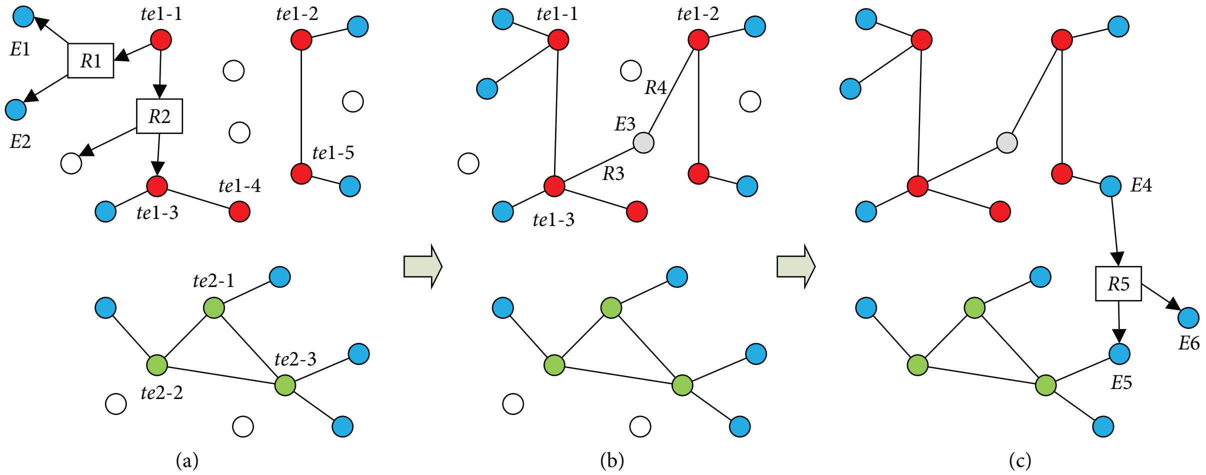

Figure 4: Main steps of finding extracted instances.

Step 1: Read the MVD as $V=\{e_m\}$, where V describes IFC entities to be extracted. Then read the filter conditions of instances as $T=\{\boldsymbol{t}\}$, which gives instance-level rules. A filter condition t is expressed as (te, tp, tv), where te, tp, and tv are the entity name, property name, and expected property value, respectively.

Step 2: Extract the core instance set of M as E_c. For each instance E in M_0, when $E\notin V$, it will not be extracted as E_c. If $E\in V$, search all filter conditions. When there exists a filter $t=(te,\ tp,\ tv)$ such that te is the entity name of E and $E.tp=tv$, E is extracted to E_c. If no filter is found, E is directly added in E_c. For example, as shown in Figure 4(a), the group of te1-1–te1–5 are filtered instances under t1 and another condition t2 filters the group of te2-1–te2–3.

Step 3: Expand the core instance set. For each core instance C, traverse all C's reverse attributes and find product entities directly related to C. Put relationship entities and product entities that pass MVD into E_c. For example, as shown in Figure 4 (a), E_1 and E_2 are related to te1-1 through R_1; therefore, E_1, E_2, and R_1 are added into submodel M.

Step 4: Consider each group of instances filtered by the same condition, such as te1-1–te1–5 in Figure 4 (b). Some core instances are linked by relationship entities, thus forming a graph structure. Between each pair of core instances in the group, if a path does not exist, find the shortest path and add all instances on the path to M. For example, there is no path between $te1$–2 and $te1$–3, but E_3 can link them through related entities. Therefore E_3, R_3, and R_4 are extracted into M.

Step 5: Extract possible relationships between groups. For every pair of groups P and Q, if an instance from P can connect to an instance from Q with only one relationship entity r, the secondary entity set of r is extracted into M. As shown in Figure 4(c), since R_5 directly connects E_4 and E_5, R_5 and its secondary set member E_6 are extracted.

Step 6: Fill necessary resource entities into instances in M.

Now M is the minimum submodel from M_0 defined by MVD and filter conditions T.

This method focuses on MVD, filter conditions, and core instances, proposing clear rules to extract an appropriate number of relevant instances, which brings two apparent benefits: (1) unnecessary instances are filtered according to IFC metadata definitions to ensure that the final result is minimal and (2) it can extract qualified instances and other strongly related instances as well. This ensures the submodel information is complete and rich in meeting on-site use needs.

6. Fast and Accurate Information Acquisition

6.1. Relational BIM Storage and Indexing Method for Rapid Search. To address the problems mentioned in Section 1, a model storage method is designed to efficiently store model data from IFC and improve efficiency during data search. First, according to the metadata of entities in IFC models, different kinds of storage tables are established respectively for a basic entity, relationship entity, and resource entity. Then some indexing tables are precalculated for search requirements. The detailed algorithm is listed in Appendix A, which features the following advantages:

(1) The attribute information of each independently exchangeable instance is stored in the same table, so there is no need to retrieve data from many resource entity tables during data extraction.

(2) For relational entities and reverse attributes of the base entity, only the entity name and GlobalID of the base entity in the attribute value are stored, which enables a fast query of the relationship between the base entities and avoids duplicate storage of the base entity data.

(3) Toward the need for rapid search to the attributes of resource entities, indexing tables are established in advance, avoiding traversing multiple tables. On average, indexing tables will occupy 5% of space, but its query is 10 times faster than table join.

6.2. Reason-Rich Model Change Detection. A change detection method for IFC models based on object metadata is proposed to solve the unreliability and low efficiency of component change detection between different versions of BIM models. The detailed algorithm is listed in Appendix B. The proposed change detection method can detect changes such as component, type, property set, as well as the relationship between components. In addition to changed contents, the profound reason for changes can be reported, which enriches the semantics of detection results. The reason for changes helps the general contractor to classify numerous changes and focus on changes that matter. This method can also avoid the error result caused by circular references.

7. Case Study

7.1. Case Overview. The proposed IFC-based solutions were tested in a large airport terminal project in South China.

This airport terminal consists of five floors above the ground and three floors underground, with more than 210,000 square meters of the construction area. In this case, the application scenarios of digital construction are distributed from the deepening design phase to the early stage of the construction phase.

The digital construction model involved was the first and second floors of the main terminal building. Since the project's steel structure, mechanical system, and construction process were quite complex, digital design technology was adopted in the preliminary design stage. After a detailed preliminary design model was delivered, the solution package proposed in this paper was used in several key scenarios to establish deepening design BIM and better utilize models in subsequent stages.

The application process is shown in Figure 5, which takes the construction section of the check-in island on the second floor as an example. First, the preliminary design model was divided into several professional submodels through several MVDs and delivered to each deepening design group for design work, including structure, mechanical pipeline, equipment installation, water supply and drainage, architecture, and other majors. In this process, the general contractor required important changes between the final version of the structural deepening and a previous version, so the change detection algorithm was adopted. Next, the professional models are merged into the primary model by the BIM consulting team, which involves a consistent integration scenario. Then, after collision detection, verification, and approval, the BIM database was established through the proposed model storage method and several indexing tables. Finally, several submodels are extracted from the BIM database to use the digital model during construction, such as steel columns with working schedule information for the steel installation team. And all mechanical pipes and equipment for the mechanical subcontractor on the second floor. The following section describes the case application with several examples and a performance analysis.

7.2. Digital Construction Scenarios and Applications

7.2.1. Design Submodel Merge. During the model merge scenario of deepening design submodels, the IFC instances were analyzed in steps according to Section 3.1. The original primary design model MC describes electric components and a

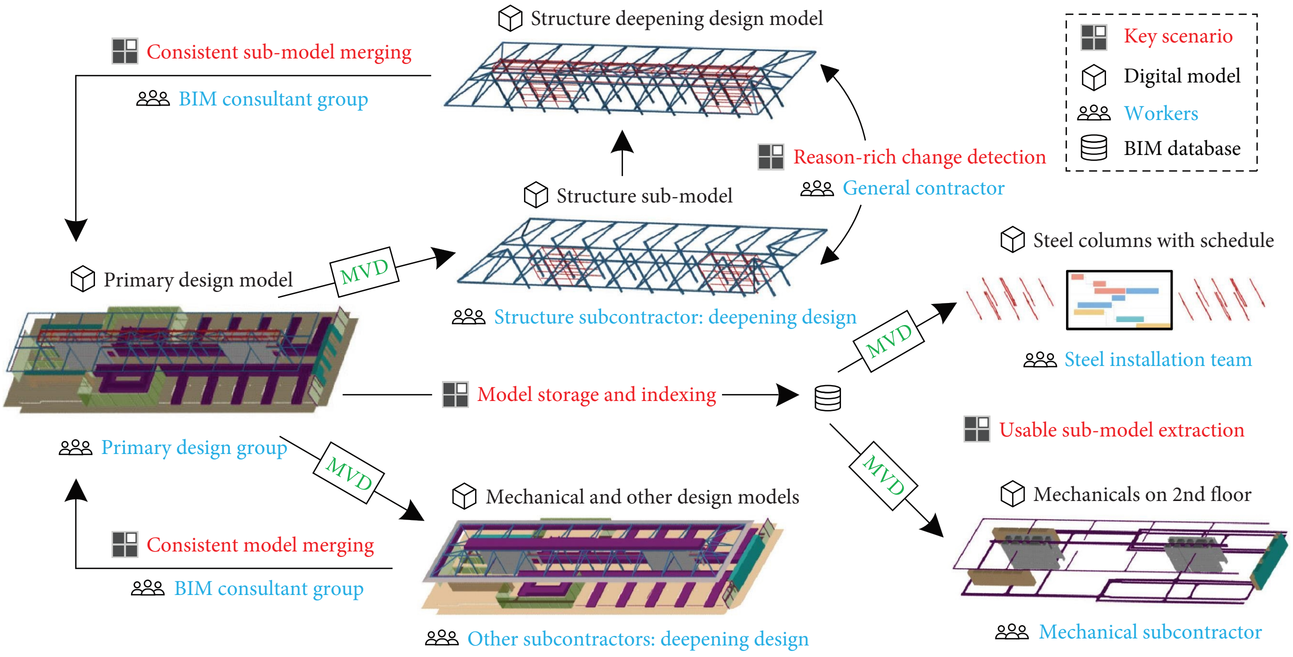
Structure deepening design model
Consistent sub-model merging
BIM consultant group
Key scenario
Digital model
Workers
BIM database
Reason-rich change detection
General contractor
Structure sub-model
Steel columns with schedule
Primary design model
MVD
Structure subcontractor: deepening design
MVD
Steel installation team
Model storage and indexing
Usable sub-model extraction
Primary design group
MVD
Mechanical and other design models
MVD
Mechanicals on 2nd floor
Consistent model merging
BIM consultant group
Other subcontractors: deepening design
Mechanical subcontractor

Figure 5: Application flow in the airport terminal project.

Table 2: Some examples of modifications and analysis.

Modifications in M_1/L_1	Analysis for changes in IFC	Instances to be merged in L_2 by MVD and modification status
Add two child tasks for the "cable tray" task: "cable tray-underground" and "cable tray-ceiling"	Add two new task entities: IfcTask1#, IfcTask2#, and one relational entity, IfcRelAggregates1#. Modify the reverse attribute of IfcTask12#	IfcTask1# (added) IfcTask2# (added) IfcRelAggregates1# (added) ★IfcTask12# (modified)
Relate the model components of the ceiling cable tray to the "cable tray-ceiling" task node. Associate the model components of the underground cable tray to the "cable tray- underground" task node	Add two relational entities: IfcRelAssignsToTask3#, IfcRelAssignsToTask4#. Modify the reverse attributes of the underground cable tray (IfcDuctSegment1#) and ceiling cable tray(IfcDuctSegment2#). Modify IfcRelAssignsToTask34# to relate to IfcTask34# and corresponding physical components	IfcRelAssignsToTask3# (added) IfcRelAssignsToTask4# (added) ★IfcRelAssignsToTask34# (modified)
Delete the "fire hydrant box" task	Delete IfcTask7# instance and IfcRelAssignsToTask7#. Modify an aggregate	IfcTask7# (deleted) IfcRelAssignsToTask7# (deleted) IfcRelAggregates7# (modified)

Note: "★" denotes instances that should be recovered after merging.

sequence of working steps to install them, including 21 IfcTask instances, 374 IfcProduct and its subclasses, 42 IfcTask instances, and their IfcRelationship instances. For example, the model includes IfcTask instances "compass box air duct," "cable tray," "fire hydrant box installation," and their related components. Parts of the extracted instances in M0/L0 are as follows:

(**IfcTask**, {IfcTask12# (compass box air duct), IfcTask34# (cable tray), cabinet), IfcTask7# (fire hydrant box), …})

(**IfcDuctSegment**, {IfcDuctSegment1# (cable tray-ceiling), IfcDuctSegment2# (cable tray-underground), …})

There are some relationship instances under **IfcRelAggregates**, **IfcRelAssignsToTask**, and **IfcRelSequence**.

After the deepening design phase, the mechanical submodel with schedule (progress information) was about to merge into M_0. Read in the IFC submodel M_1 as list $L_1 = \{(e_i, \{o_{ij}\})\}$. Table 2 lists some typical modifications in M_1. The corresponding MVD is $V = \{e_m\} = \{$IfcTask, IfcDuctSegment, IfcRelAggregates, IfcRelSequence, IfcRelAssignsToTask$\}$. According to the proposed method, L_2 with To-merge instances and their modification status was calculated in Table 2 as well.

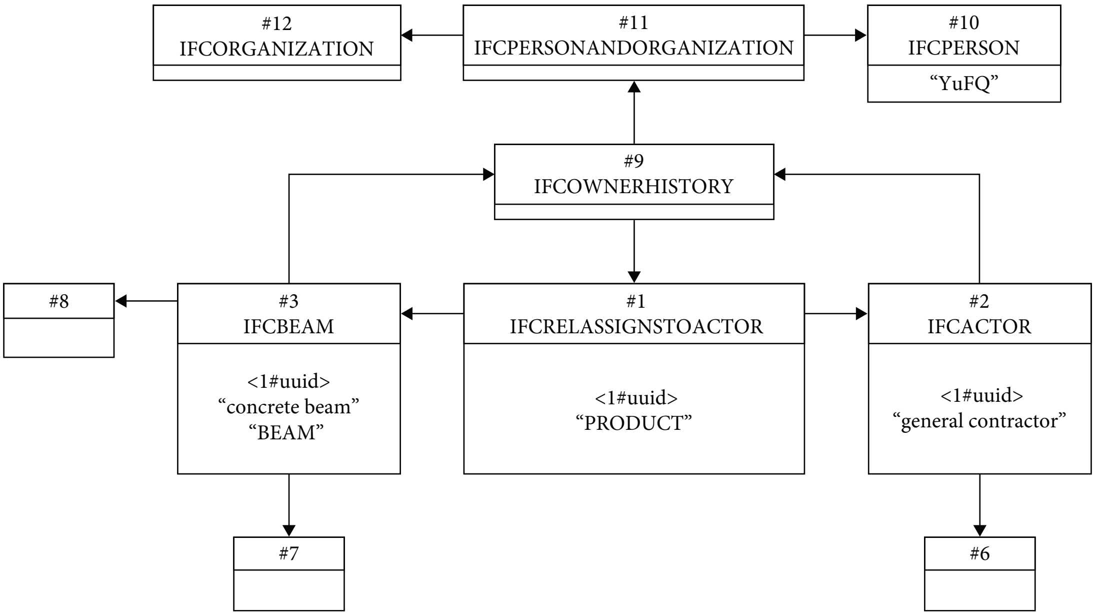

Figure 6: The partial structure of an IFC file.

Table 3: Column structure and values of beam table.

Data column name	Attribute type	Value
GlobalID	Basic/simple	"0ZgN…egZq3Dq"
Owner history	Basic/complex	Binary series of resource 9#
Name	Basic/simple	"Concrete beam"
Predefined type	Basic/simple	"BEAM"
⋮	⋮	⋮
Object placement	Geometry/complex	Binary series of an IFC object placement
Representation	Geometry/complex	Binary series of an IFC product representation

For those added instances, such as IfcTask1# and IfcRelAssignsToTask3#, there are no related instances in M_0; thus no need to recover. For deleted instances, their information would be ignored. However, some modified instances should be recovered on their attributes. For example, there are some relationship instances from the primary model but not in M_0/L_0 relating to IfcTask12#, so three IfcRelAssignToControl are added to reversed attribute HasAssignments, and one IfcRelNests was added to reversed attribute Nests. Another recovered instance was IfcRelAssignsToTask34#. In addition, a slab in M_C was not extracted to M_0, and an IfcSlab was added to the secondary entity set.

Finally, merge M_2 to M_C: (1) add all new instances to M_0; (2) use modified instances to replace original instances in M_0; and (3) find deleted instances by GlobalID and set their OwnerHistory to "deleted." After any submodel merge event, M_C was able to remain consistent and complete.

The method uses MVD and metadata of secondary entity set and reversed attributes to avoid merging redundant IFC entities. In the project, redundant entities were no more than 5% in the primary model. In addition, the problem of incomplete data caused by modified instances was avoided. The general contractor validated the information integrity rate as greater than 95%.

7.2.2. Model Storage and Indexing. After the final deepening design model was established by merging, it was stored in a database according to the proposed storage and indexing method. Figure 6 shows an example of a fragment of an IFC file.

Exchangeable entities were first stored in separate tables. Then, all beams (including 3#) were stored in the beam table as in Table 3.

Then, some indexing tables are calculated based on management requirements. For example, during schedule management, the executor information of a construction task is important. Therefore some rapid search requirements for all team leaders are proposed, for instance, 10#. The database created an indexing table I_PERSON, storing the ID and name of 10#. Then find all instances that quoted 10#, which is #11. Similarly, find that #9 quoted 11#, and {1#, 2#, 3#} quoted 9#. Finally, a row in the I_PERSON table stored information as Table 4.

Now important exchangeable instances can be obtained in one single query, which is much faster than traditional

Table 4: Column structure and values in the indexing table.

Data column name	Value
GlobalID	#10
Name	“YuFQ”
cze	[(“IFCRELASSIGNSTOACTOR,”<1#uid>), “IFCACTOR,”<2#id =), (“IFCBEAM,”<3#uuid>)]

Table 5: Performance of indexing tables.

Requirement	$t = (te, tp, tv)$	Result size	Time by database search (s)	Time by indexing table (s)	Extra space (%)
Progress information of a team leader	(IFC person, family name, YuFQ)	120	3.0	0.3	1.8
The work content of all subcontractors	(IFC actor role, role, IFC roleEnum. SUBCONTRACTOR)	603	11.1	0.9	1.9
Steel components	(IFC material, name, steel)	5,084	21.0	1.6	5.8

database table joins. The manager of a general contractor can easily search for schedules and jobs of a particular work team and then calculate their progress. In Table 5, the search performance of the database search and our method were compared on three requirements. It is clear that the indexing table saved 90%–92% resource search time and behaved better when the results became larger. Although the index tables for common resource entities occupied extra space, we found that less than 10% of space was needed.

7.2.3. Reason-Rich Change Detection for Different Department Users. As mentioned, the project’s general contractor often checked the difference between the two versions of detailed design models to understand the modifications of the design drawing. The proposed method can detect the changes in components, progress, and attribute set and give the reasons for the differences. This has brought great benefits to the on-site managers. In the past, many changes were detected through the direct comparison of the database, which wasted much labor to manually choose the important differences. Now one can choose the type one care about and check for exciting changes. According to the proposed method, change detection results on the overall model are shown in Figure 7, where *R*, *P*, *T*, and *E* denote relationship, property set, types, and entities, respectively. Arrows indicated the cause of changes. More than 4,000 detected changes from nine kinds of causes, among which modified product entities were important. However, different departments of the general contractor emphasized different aspects of changed entities.

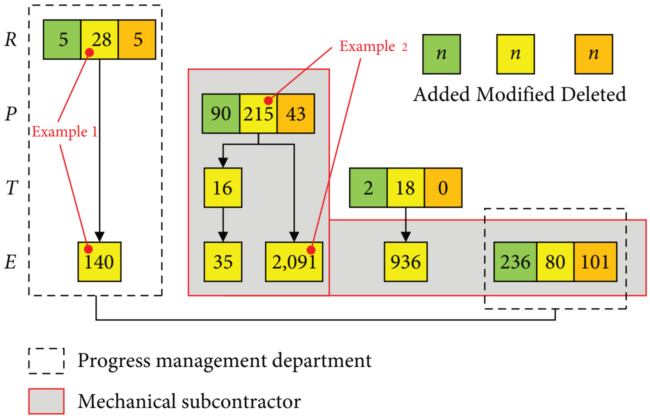

Figure 7: Numbers of detected changes and corresponding departments.

(1) The progress management department: their work was to check all component entities and progress information. Therefore, the changes in association relationships were crucial because they would affect the organization sequence and schedule of construction. Progress managers should focus on reviewing those models changed by relationship changes between components and tasks. As shown in Example 1 in Figure 7, the order of “cable tray installation task” and “fire hydrant installation task” was swapped. So there were two task changes caused by a change of relationship entity. The general contractor would send this change information only to the progress department. Because no components or geometry had been changed, other departments or subcontractors did not need to deal with these changes.

(2) Mechanical construction subcontractor: the relationship between components in the model was less important so that the relationship change will be ignored. They paid more attention to the components that change due to the difference of property set because it will affect the model information of equipment installation, the material of pipes, etc. As shown in Example 2 in Figure 7, the material of cable trays was changed from aluminum alloy to galvanized steel. So there were 15 cable trays changed because of one change of material property set. These changes significantly increased the overall budget of the mechanical equipment. The general contractor

Table 6: Performance of model extraction on requirement 1.

Method	Time (min)	Size (MB)	Comment of usability
IFC text match	0.5	1.2	Most schedule information was missing
Database join	8.1	15.6	Was good enough for the worker's disclosure
Proposed method	15.0	13.9	Was good enough for the worker's disclosure

Table 7: Performance of model extraction on requirement 2.

Method	Time (min)	Size (MB)	Comment of usability
IFC text match	4.8	25.8	Property and relationship instances were missing
Database join	17.8	366	Contained many redundant instances
Proposed method	34.3	195	Was suitable for delivery

would send these changes to the mechanical subcontractor and budget managers but not other irrelevant stakeholders.

7.2.4. Usable Submodel Extraction. There was a great demand for submodel extraction in the construction phase of the airport terminal building. Before a construction task, the work contents of a team were extracted from the original model, including component entity, spatial relationship, size information, workflow breakdown, etc. Then use the extracted model to disclose to the construction team so they can quickly understand the upcoming jobs. Therefore, model extraction requires comprehensive and complete information. In addition, it is also necessary to provide relevant partial models for each subcontractor to carry out cost management or export their drawings. These models should exclude irrelevant entities and avoid including the information of other subcontractors as possible.

According to Section 6, the following two models are used to extract requirements as examples. The first requirement was "all steel components at the ceiling of the check-in zone." The components involved were a small number of structural components but contained complex IfcTask entities. The second requirement was "mechanical units and pipes on the 2nd floor installed by Company X," involving many interrelated electromechanical pipelines and equipment, and it is required to exclude the model information of other subcontractors.

Compared with the general IFC text matching method or cross-table join operation in the database, the proposed method can consider the integrity and result volume by focusing on core entities, expanding and supplementing the strongly related entity information. Tables 6 and 7 show the comparison of the method proposed in this paper with other methods and the comments of field workers about the usability of the model. The proposed method can obtain the submodels in no more than two times the time consumption, acceptable for field use. The advantage of the proposed method became more evident when the extracted model was complex. In addition, the extraction results rarely contain redundant entities, so the size of the result remained about 10%–40% smaller than the database join.

8. Discussion and Conclusion

Aiming at the problems and difficulties of three key scenarios for the general contractor in the digital construction process, this paper proposed four solutions to better establish and utilize BIM. The algorithms of these solutions depended on the widely used IFC standard, which ensures the universality of our method. During various stages of digital construction, it is difficult to ensure consistency by integrating models in the preliminary and deepening design stages, while some information about related entities may be lost. The proposed model merge method ensured consistency before and after fusion by analyzing the modification state of individuals and incrementally recovering the information associated with entities. Then, in response to the massive need for instance-level submodels, a minimum model extraction algorithm was proposed to extract strongly related model information required by users. Irrelevant entities will be eliminated at the same time. Therefore, the extracted submodel was accurate and met the user's needs. After establishing consistent BIM and submodels, how to utilize the model information was the third scenario. The proposed new model storage method reduces a lot of cross-table searches via the BIM database, improves the efficiency of obtaining model information, and realizes the rapid search of information through index tables. In addition, the extraction results of the model were simplified under constraints from MVD. In the deepening design phase, model change detection with semantic results enables managers to find important and concerning changes pertinently and avoids retrieving unnecessary information.

This paper's advantage is providing better digital services for all participants of large-scale projects from the perspective of general contracting. The solutions had been applied in a large-scale construction project, and the performance was summarized. As a result, the departments of the general contractor company and subcontractors validated two significant benefits as follows:

(1) By applying the method of establishing BIM, the correctness and consistency of general contractors are improved. Model merging ensures that all entities in the design model are of no repetition or omission. The results obtained through submodel extraction

are also independently suitable for subcontractors' use. Therefore, BIM technology plays a vital role in modern design and construction.

(2) It is more convenient for on-site general contractors to utilize BIM information, reducing manual labor and improving production efficiency. Retrieving resource information from the BIM database became more efficient. When facing many model changes, one can use the change reasons to locate interesting components and filter irrelevant information. Generally speaking, practitioners have introduced digital construction into the traditional workflow and gained benefits, so they are more willing to use new technologies, forming a virtuous circle. This benefit also promoted the digitization transformation of the construction industry.

On-site applications have proved that the proposed method can support key scenarios in digital construction workflow. However, there are still some known limitations of the proposed methods. Future research may include the followings: (1) although IFC is a widely supported data standard format, the IFC exported by current digital construction software will still lack information. Therefore, getting through the integration and change detection from software to software is worth studying. (2) There is a lack of suitable methods to deal with users' custom IFC extension entities. (3) On the computing side, the proposed algorithm still involves many intensive calculations, such as path searching, so the speed performance may be further optimized by the cloud BIM computing framework.

Appendix

Appendix A. The detailed algorithm of model storage and indexing

Traverse each instance ***E*** in the IFC model and decide subsequent steps by ***E***'s metadata.

A.1. Case 1: E Is an Instance of a Basic Entity

Step 1: Find the corresponding data table T_p based on the name of an entity. Add a row to the table and store the value of GlobalID.

Step 2: Iterating through each attribute A of instance E, storing the attribute's value in the A column. If the value of an attribute is a complex IFC instance (not a simple number or string), the binary sequence of this complex instance is directly stored instead of just an ID. When visiting the attribute, the binary sequence will be deserialized into complete information of the instance. This strategy avoids massive cross-table searches and improves efficiency.

Step 3: Iterate through each reverse attribute V of instance E, storing the instance name and GlobaID of the reverse attribute in the V column.

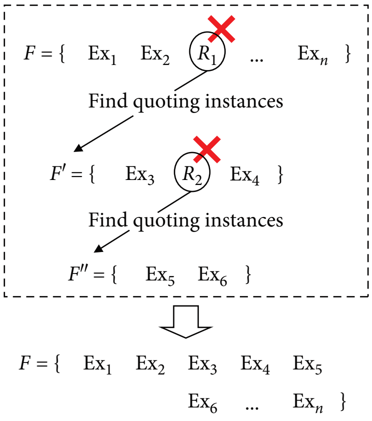
F = { Ex$_1$ Ex$_2$ R_1 ... Ex$_n$ }
Find quoting instances
F' = { Ex$_3$ R_2 Ex$_4$ }
Find quoting instances
F'' = { Ex$_5$ Ex$_6$ }
F = { Ex$_1$ Ex$_2$ Ex$_3$ Ex$_4$ Ex$_5$ Ex$_6$ ... Ex$_n$ }

FIGURE 8: Recursive process of finding related exchangeable entities.

A.2. Case 2: E Is an Instance of a Relationship Entity

Step 1: Find the corresponding data table T_r based on the entity's name. Add a row to the table and store the value of GlobalID.

Step 2: Iterating through each attribute A of instance E, storing the attribute's value in the A column.

Step 3: Store the entity name and GlobalID of E's primary entity (instance of base entity) as two strings in the "*cra*" column.

Step 4: Store entity names and GlobalID of each instance in E's secondary entity set (set of base entities) as a list of string pairs in the "*crb*" column.

A.3. Case 3: E Is an Instance of a Resource Entity. Before data storage, some standard search requirements for resource entities are determined by a list $Zs = \{z_{si}\}$, $z_{si} = (z_{\text{name}}, z_{\text{pname}})$, where z_{name} is the resource entity name and z_{pname} is the index attribute name of the *zname* entity.

Step 1: According to the search requirement list, some "indexing tables" called T_z are established to store index data of necessary resource entities. The name of an indexing table is the resource entity z_{name}. Then create a GlobalID column, a column named by z_{pname}, and a column "*cze*" to store all related instances.

Step 2: Traverse each resource instance E in the IFC model. If any z_{name} in Zs equals E's name, a row is added in the table z_{name} to store E. The temporary ID of E is put into the GlobalID column.

Step 3: Find all instances $F = \{f_i\}$ that quote E. In other words, there exists an attribute of f_i equals E. Most instances in F are independently exchangeable entities, but some are resource entities.

Step 4: Enter a recursive process shown in Figure 8. For each resource instance f_i in F, find all instances F' that quote f_i, and replace f_i with F'. Finally, all instances in F are exchangeable entities.

Step 5: Store the entity names and GlobalID of instances in F as a list of strings in "*cze*" column. This allows for finding related exchangeable entities within one single query.

Appendix B. The detailed algorithm of change detection

IFC models A and B are compared. Take model A as a benchmark. Changes made in model B will be detected.

Step 1: Extract component instance set E_a (under IfcObject), component type instance set T_a (under IfcObjectType), extended attribute instance set P_a (under IfcPropertyDefinition), and relational instance set R_a (under IfcRelation) from IFC model A. Similarly, extract E_b, T_b, P_b, and R_b from IFC model B.

Step 2: Based on the ID and attributes of the instance, calculate added component set E_{add}, deleted component set E_{del}, modified component set E_{mod}, and unchanged component set E_{unc}. First, only compare GlobalID and get $E_{\text{unc}} = E_a \cap E_b$, $E_{\text{add}} = E_b - E_{\text{unc}}$, $E_{\text{del}} = E_a - E_{\text{unc}}$. Then compare simple attributes (only numbers and strings) in E_{unc}. If instances o_a and o_b has the same GlobalID but differ in any attribute, move o_a into E_{mod}.

Step 3: Calculate all component instances E_{type} associated with T_{add}, T_{del}, and T_{mod}. Then calculate modified component instances because of type modification: $E_t = E_{\text{unc}} \cap E_{\text{type}}$. Remove E_t from E_{unc} and add E_t to E_{mod}.

Step 4: For T_a and T_b, use the same method to calculate added type set T_{add}, deleted type set T_{del}, modified type set T_{mod}, and unchanged type set T_{unc}. For P_a and P_b, similarly calculate added property set P_{add}, deleted property set P_{del}, modified property set P_{mod}, and unchanged property set P_{unc}.

Step 5: Calculate all component instances E_{pro} and all type instances T_{pro} associated with P_{add}, P_{del}, and P_{mod}. Then calculate type instances modified because of property modification: $T_y = T_{\text{unc}} \cap T_{\text{pro}}$. Next, remove T_y from T_{unc} and add T_y to T_{mod}. Then calculate all component instances E_{pt} associated with T_y. Next, calculate component instances modified by property modification: $E_t = E_{\text{unc}} \cap E_{\text{pr}} \cup E_{\text{pt}}$. Finally, remove E_t from E_{unc} and add to E_{mod}.

Step 6: For R_a and R_b, calculate added relation set R_{add}, deleted relation set R_{del}, modified relation set R_{mod}, and unchanged relation set R_{unc}.

Step 7: Let E_r be modified component instances because of relationship modification. Initialize $E_{\text{rm}} = \emptyset$. For each relation, R in R_{add} and R_{del}, add R's primary entity and secondary entity set into E_{rm}. If the primary entity is modified for each relation R in R_{mod}, only add R's primary entity into E_{rm}. But if the secondary entity set is modified, only those different instances in the secondary set are added to E_{rm}.

Finally, calculate component instances modified because of relationship modification: $E_r = E_{\text{unc}} \cap E_{\text{rm}}$. Remove E_r from E_{unc} and add E_r to E_{mod}.

Now all modified instances can be output with detailed information about why they are changed.

Data Availability

The data used to support the case study are available from the corresponding author upon request or download from open data on the product website: https://www.ibuildingsh.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the Shanghai Science and Technology Innovation Action Plan project (22dz1207102), the Shanghai Outstanding Discipline Leader Program (22XD1432000), and the Innovation Project of Shanghai State-owned Assets Supervision and Administration Commission (2022008).

References

[1] S. Durdyev, M. Ashour, S. Connelly, and A. Mahdiyar, "Barriers to the implementation of building information modelling (BIM) for facility management," *Journal of Building Engineering*, vol. 46, Article ID 103736, 2022.

[2] B. Sanchez, C. Rausch, C. Haas, and T. Hartmann, "A framework for BIM-based disassembly models to support reuse of building components," *Resources, Conservation and Recycling*, vol. 175, Article ID 105825, 2021.

[3] C. Merschbrock and B. E. Munkvold, "Effective digital collaboration in the construction industry—a case study of BIM deployment in a hospital construction project," *Computers in Industry*, vol. 73, pp. 1–7, 2015.

[4] O. I. Olanrewaju, A. F. Kineber, N. Chileshe, and D. J. Edwards, "Modelling the relationship between Building Information Modelling (BIM) implementation barriers, usage and awareness on building project lifecycle," *Building and Environment*, vol. 207, Part B, Article ID 108556, 2022.

[5] A. Alattas, E. Kalogianni, T. Alzahrani, S. Zlatanova, and P. van Oosterom, "Mapping private, common, and exclusive common spaces in buildings from BIM/IFC to LADM. A case study from Saudi Arabia," *Land Use Policy*, vol. 104, Article ID 105355, 2021.

[6] H. Li, H. Liu, Y. Liu, and Y. Wang, "An object-relational IFC storage model based on Oracle database," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 41, pp. 625–631, 2016.

[7] A. Pérez-García, N. Martín-Dorta, and J. Á. Aranda, "BIM requirements in the Spanish public tender—analysis of adoption in

construction contracts," *Buildings*, vol. 11, no. 12, Article ID 594, 2021.

[8] L. Liu, B. Li, S. Zlatanova, and P. van Oosterom, "Indoor navigation supported by the Industry Foundation Classes (IFC): a survey," *Automation in Construction*, vol. 121, Article ID 103436, 2021.

[9] X. Shi, Y.-S. Liu, G. Gao, M. Gu, and H. Li, "*IFCdiff*: a content-based automatic comparison approach for IFC files," *Automation in Construction*, vol. 86, pp. 53–68, 2018.

[10] O. Doukari, B. Naudet, and R. Teulier, "Merging IFC-based BIM models: a new paradigm and co-design support tool," *International Journal of 3-D Information Modeling (IJ3DIM)*, vol. 6, no. 1, pp. 51–64, 2017.

[11] G. Ren, H. Li, S. Liu, J. Goonetillake, A. Khudhair, and S. Arthur, "Aligning BIM and ontology for information retrieve and reasoning in value for money assessment," *Automation in Construction*, vol. 124, Article ID 103565, 2021.

[12] Z. Xu, T. Huang, B. Li, H. Li, and Q. Li, "Developing an IFC-based database for construction quality evaluation," *Advances in Civil Engineering*, vol. 2018, Article ID 3946051, 22 pages, 2018.

[13] Y.-H. Lin, Y.-S. Liu, G. Gao, X.-G. Han, C.-Y. Lai, and M. Gu, "The IFC-based path planning for 3D indoor spaces," *Advanced Engineering Informatics*, vol. 27, no. 2, pp. 189–205, 2013.

[14] H.-M. Chen, K.-C. Chang, and T.-H. Lin, "A cloud-based system framework for performing online viewing, storage, and analysis on big data of massive BIMs," *Automation in Construction*, vol. 71, Part 1, pp. 34–48, 2016.

[15] J.-R. Lin and Y.-C. Zhou, "Semantic classification and hash code accelerated detection of design changes in BIM models," *Automation in Construction*, vol. 115, Article ID 103212, 2020.

[16] C. Mignard and C. Nicolle, "Merging BIM and GIS using ontologies application to urban facility management in ACTIVe3D," *Computers in Industry*, vol. 65, no. 9, pp. 1276–1290, 2014.

[17] M. Q. Huang, H. M. Zhu, J. Ninić, and Q. B. Zhang, "Multi-LOD BIM for underground metro station: interoperability and design-to-design enhancement," *Tunnelling and Underground Space Technology*, vol. 119, Article ID 104232, 2022.

[18] Z. Xu, L. Zhang, H. Li, Y.-H. Lin, and S. Yin, "Combining IFC and 3D tiles to create 3D visualization for building information modeling," *Automation in Construction*, vol. 109, Article ID 102995, 2020.

[19] A. A. Siddiqui, J. A. Ewer, P. J. Lawrence, E. R. Galea, and I. R. Frost, "Building Information Modelling for performance-based fire Safety Engineering Analysis—a strategy for data sharing," *Journal of Building Engineering*, vol. 42, Article ID 102794, 2021.

[20] Q. Qiu, X. Zhou, J. Zhao et al., "From sketch BIM to design BIM: an element identification approach using industry foundation classes and object recognition," *Building and Environment*, vol. 188, Article ID 107423, 2021.

[21] J.-R. Lin, Z.-Z. Hu, J.-P. Zhang, and F.-Q. Yu, "A natural-language-based approach to intelligent data retrieval and representation for cloud BIM," *Computer-Aided Civil and Infrastructure Engineering*, vol. 31, no. 1, pp. 18–33, 2016.

[22] S. Tang, D. R. Shelden, C. M. Eastman, P. Pishdad-Bozorgi, and X. Gao, "BIM assisted Building Automation System information exchange using BACnet and IFC," *Automation in Construction*, vol. 110, Article ID 103049, 2020.

[23] G. Lee, Y. H. Park, and S. Ham, "Extended Process to Product Modeling (xPPM) for integrated and seamless IDM and MVD development," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 636–651, 2013.

[24] B. Koo and B. Shin, "Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure Building Information Models," *Journal of Computational Design and Engineering*, vol. 5, no. 4, pp. 391–400, 2018.