IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 June 2024, accepted 4 July 2024, date of publication 10 July 2024, date of current version 18 July 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3425898

RESEARCH ARTICLE

Research on Partial Model Extraction of Railway Infrastructure Based on the Industry Foundation Classes Files

YUNSHUI ZHENG[1,2], YIMIN SHI[1,2], AND XINKAI WANG[2]

[1]Key Laboratory of Railway Industry of BIM Engineering and Intelligent for Electric Power, Traction Power Supply, Communication and Signaling, Lanzhou Jiaotong University, Lanzhou 730070, China
[2]School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

Corresponding author: Yimin Shi (12221578@stu.lzjtu.edu.cn)

ABSTRACT To effectively cope with the ever-growing intricacy of railway Industry Foundation Classes (IFC) file sizes, the adoption of a partial model approach has consistently been recommended as a highly effective strategy. This approach facilitates the efficient handling of increasing data volumes, enhancing the management capabilities of IFC files for purposes such as data storage, exchange, and transmission. This paper proposes an algorithm that adopts an iterative extraction methodology, grounded on the hierarchical IFC model's tree structure to produce the desired partial models. Its primary objective is to minimize the size of large IFC files by selectively extracting the indispensable models. By leveraging solely the data structure of the IFC files, this approach circumvents the necessity for file format conversion or reliance on Model View Definitions (MVD). During the extraction of the required models, two types of related attributes are simultaneously extracted, and the extraction of relationship entity properties is optimized to enhance the extraction efficiency. To assess the effectiveness of our algorithmic approach, we conducted an in-depth case study centered around a contemporary high-speed railway project located in the southwestern region of China. The extracted models' integrity was verified using BIMvision, while semantic syntax verification was performed using Express Engine and IfcObjectCounter. The results demonstrate that our algorithm not only accurately extracts the desired model segments from the IFC file, but also significantly improves model loading efficiency, minimizes memory usage, achieves model miniaturization, and shows promising performance and application prospects.

INDEX TERMS Industry foundation classes (IFC), partial model, extraction, building information modeling (BIM), intelligent railway.

I. INTRODUCTION

During the past ten years, Building Information Modeling (BIM) technology has emerged as a crucial milestone in the evolution of the architecture, engineering, and construction (AEC) industry, progressively evolving into a digital technology (DT) that profoundly influences various aspects of the field [1]. BIM technology boasts a wide range of applications, encompassing numerous infrastructure domains, including building construction [2], [3], [4], railway systems [5], [6], [7], [8], Heating, Ventilation and Air Conditioning (HVAC) systems [9], fire safety measures [10], and bridge and tunnel engineering [11], [12]. Compared to the limitations of traditional CAD technology, BIM's visualization and collaborative capabilities align more closely with the vision of intelligent railway planning [13], making the integration of BIM with railway engineering a global trend [14]. Railway BIM models not only comprehensively depict the geometric features of construction elements and objects, but also contain comprehensive semantic and functional details [15], [16], providing robust support for various stages of the building lifecycle. However, despite its immense

The associate editor coordinating the review of this manuscript and approving it for publication was Diego Bellan.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

potential, implementing BIM in railway engineering still faces numerous challenges.

To enhance the cross-platform interoperability of BIM technology, buildingSMART has introduced the Industry Foundation Classes (IFC) as a neutral and open standard for data exchange. This standard aims to facilitate the efficient exchange of both geometric and semantic data among various project stakeholders [17], [18], [19], thus strengthening collaboration and communication in the construction industry. IFC 4, the latest iteration of this standard, has been continuously updated and expanded since its release in 2013. The latest specification, IFC 4×3, extends the scope of the IFC model to cover railways, highways, ports, and waterways, and includes dynamic and static extensions tailored to the specific requirements of the railway engineering domain [20]. However, large IFC files challenge storage, exchange, management, and transmission. As projects grow complex, so do the demands for diverse skills and extensive information exchange, especially in intricate undertakings like railway projects [21], [22].

The paper primarily focuses on the critical components of railway infrastructure, electric power, traction power supply, communication, and signaling. It introduces an algorithm tailored to railway engineering, efficiently selecting data models from intricate IFC files. This reduces costs related to object creation, storage, management, and access. Accumulating reusable components lays a solid foundation for BIM libraries. Furthermore, the partial model extraction function enhances security by minimizing data breaches and erroneous modifications in railway models.

The subsequent sections will provide an in-depth analysis of these components in the following structure. Section II delves into the underlying causes of oversized IFC files and provides a review of pertinent research, synthesizing the existing issues. Section III presents fundamental terminology and the hierarchical organization of the IFC. Section IV presents a comprehensive exposition of our proposed algorithm. Section V presents a case study and a thorough analysis of the obtained results. Section VI provides a summary of our key contributions, bringing the discussion to a conclusion.

II. RELATED WORK

The substantial size of IFC files can be attributed primarily to two factors. Firstly, the sheer scale of buildings described in IFC files, along with their intricate structural details, significantly contributes to the bulkiness of these files. Secondly, a significant quantity of redundant data instances exported by BIM software, often manifesting as duplicate information, further contributes to the large magnitude of IFC files [23]. There exist numerous methods for mitigating excessive file sizes, and one commonly employed approach is the lightweight BIM model, which effectively addresses the issue of overly large files [24], [25]. For IFC files, a direct and effective strategy for size reduction involves compressing the text. One standardized compression format that is endorsed by buildingSMART is ifcZIP, offering an effective means to achieve this objective. The utilization of the ifcZIP compression format can effectively compress the resulting file size by a significant amount, ranging between 60% and 80%. However, this approach necessitates a decompression step prior to utilization, compromising the ease of comprehension and readability of the file content [26]. Another approach involves the deletion of redundant instances within IFC files. This method aims to eliminate redundant information arising from the import and export processes across various software platforms, minimizing the file size and achieving a more concise and compact IFC file format [22], [23], [27], The widely utilized commercial software, Solibri IFC Optimizer, falls under the category of methods that involve deletion of redundant instances. This software effectively removes redundant information from IFC files, contributing to a more compact and efficient file format [28]. The final methodology involves the partial extraction of models, a process where designers of diverse specialties, throughout the lifecycle, concentrate solely on the model information pertinent to their respective domains. Extracting the partial model approach related to specific requirements from the original IFC model, to decrease the data size and reconstruct the entire IFC file [29], [30]. This method was also used in this paper. Table 1 summarizes some of the recent works concerning problems about mitigating excessive IFC file sizes.

TABLE 1. Summary of works concerning problems related to the realm of IFC partial model extraction.

Extraction method	centering References
Compressing the text	[24], [25], [26]
Deletion of redundant instances	[22], [23], [27], [28]
Partial extraction of models	[29], [30]

In the realm of IFC partial model extraction, three principal methodologies currently exist extraction from a database derived from the transformation of the IFC file, extraction utilizing an ontology transformed from the IFC file, and direct extraction from the IFC file itself.

A. DATABASE-BASED MODEL EXTRACTION

IFC standards provide substantial support for BIM-integrated applications and facilitate the secondary development of associated software. However, due to the inherent limitations of the IFC file format, it cannot serve as the primary means of data storage for information systems [31]. Consequently, the utilization of an IFC model server is necessary for effective data storage. The prevailing types of IFC databases encompass relational databases, non-relational databases, and BIMServer servers. Solihin et al. [32] successfully transformed IFC data into a BIMRL schema, utilizing a star-like schema as its foundation. This schema enables the utilization of standardized SQL queries, which can be tailored to retrieve data based on specific component attributes and spatial locations. However, this approach offers read-only access, and spatially related retrieval remains relatively inefficient. Lee et al. [33] presented an

object-relational database (ORDB) framework that leverages relational databases (RDBs) to enhance the translation of inheritance structures and aggregation relationships inherent in the IFC standard, thus optimizing the mapping process. This approach harnesses the object-oriented capabilities of ORDB, enhancing the query efficiency of IFC servers. However, the study failed to consider the potential impact on the performance of other database operations. BIMServer, the foremost IFC model server, provides a Java-based interface that enables the construction of partial BIM model queries. It employs Oracle Berkeley DB as its backend database [34]. Despite its popularity, BIMServer is constrained by the absence of 3D spatial operators necessary for spatial analysis, limiting its usage primarily to attribute query operations.

The exploration of partial model extraction from IFC databases remains in its nascent stages, posing significant challenges in generating novel geometric models from existing geometries and resolving spatial 3D complexities [35]. Addressing these challenges holds the potential to offer innovative approaches for partial model extraction, thus advancing the field significantly.

B. ONTOLOGY-BASED MODEL EXTRACTION

Ontologies serve as a means to articulate domain-specific terminology and offer a comprehensive portrayal of the intricate relationships among terms and terminology across various hierarchical levels of the model [36]. Both OWL (Web Ontology Language) and RDF (Resource Description Framework) possess the capability to articulate and describe ontologies effectively. Beetz et al. [37] introduced a methodological framework for deriving BIM partial models from a graph database, subsequent to the transformation of IFC text data into an ontology representation language. However, this method is limited to the extraction of geometric and topological information. Zhang and Issa [38] developed a methodology for extracting partial IFC models from comprehensive IFC files, with the implementation executed utilizing Java. This approach incorporates an ontology-based framework that performs two traversals to retrieve specific information from the IFC model, enabling the extraction of partial models. Venugopal et al. [39] introduced an ontology-centered methodology for facilitating the exchange of information models. This approach involves the transformation of IFC instances into ontologies, followed by the extraction of partial models utilizing an inference engine. Nevertheless, there exists a potential risk of information loss during the conversion process and lacks the capability to eliminate redundant instances within the partial model. Farias et al. [40] proposed a methodology for extracting building model views by leveraging Semantic Web technologies to construct a knowledge graph. Initially, this approach converts the IFC standard model into an OWL ontology. Then, it reformulates the model view using DL-safe Horn rules. Finally, it accomplishes the extraction of a segment of the model within the newly defined model view.

Ontology-based extraction methods exhibit sensitivity to the IFC version, necessitating timely adaptations in the event of updates to the IFC file version. Furthermore, these methods are limited to extracting specific entities and attributes within the model, potentially leading to data loss, and the conversion of files to instances results in an increase in file size, extending the time required for the extraction process.

Both aforementioned methods do not directly operate on the original IFC file but rather involve a conversion of the file format. Firstly, the IFC format file is transformed into ontology or database, upon which the extraction of a partial model is subsequently performed. This approach is susceptible to potential information loss and alterations during the conversion process. It is more time-consuming and less efficient compared to directly extracting the desired model portion from the original IFC model. Therefore, this methodology is not considered an optimal solution for addressing the pertinent challenges associated with such problems.

C. PARTIAL MODEL EXTRACTION

The IFC framework utilizes Express as its foundation, with data information being described through STEP physical files. Won et al. [41] introduced a schema-free algorithm that eliminates the requirement for Model View Definition (MVD) considerations and exclusively uses the data structure of the IFC file for extracting partial models. This algorithm initiates with the data instances of the designated IFC entity, subsequently traverses referencing relationships iteratively, and terminates once all data instances referenced by the extracted instance have been extracted. This approach incorporates the consideration of the inherent data structure of the IFC, yet the initial point of the algorithmic recursion focuses on the physical entity of the IFC that is being queried. This approach may result in inefficient retrieval operations. Furthermore, certain extracted models incorporate a segment of irrelevant instances due to the absence of consideration for the specificity of IfcOwnerHistory. Gui et al. [42] proposed an MVD-based extraction method that utilizes attribute-based local MVD to define the required elements to directly generate the desired partial models. Deng et al. [30] suggests a BIM partial model extraction approach that is founded on the utilization of selection sets, leveraging the extensible Markup Language (XML) as a common linguistic tool. This approach designs various extraction rules, tailored to the needs of building users, enabling them to extract the desired data from the original model. However, IFC XML files are typically three to four times larger in size compared to EXPRESS-based files, and their usage is generally limited. Their utilization is primarily necessitated in scenarios where interoperability with XML tools is an imperative requirement. Du et al. [29] introduced a methodology titled NR-RSB for the extraction of non-redundant BIM partial models from IFC files. This approach specifically targets physical and relational entities, integrating partial model extraction with

redundant instance elimination to optimize the size of the resulting model. The methodology employed for extracting partial models is analogous to the approach described in the literature [41] and incorporates considerations for the specificity of IfcOwnerHistory. Nevertheless, it concurrently exhibits limitations in terms of inefficient lookup operations.

The primary focus of this paper's research lies in extracting IFC models specifically pertaining to the realm of railway engineering. In contrast to the traditional construction industry, there have been comparably fewer attempts to implement BIM in the realm of railway engineering. Consequently, the extraction algorithms employed must possess enhanced precision, rendering the extraction process more challenging. It is imperative to guarantee that each device is disassembled at the component level in strict adherence to the Guidance on Railway Engineering Breakdown Structure and other pertinent standards and norms, ensuring the completeness of the extracted minimal model unit.

The key contributions of this paper involve devising an instance-based partial model extraction algorithm called CRF-IFCExtractor that effectively extracts the requisite model along with its related attributes. It leverages the hierarchical structure of IFC files to its fullest extent, reduces the time required for partial model extraction, and reassigns the extracted instance. The realization of this approach takes place on the Visual Studio 2022 platform, and it has been tested using three distinct categories of IFC models pertaining to railway engineering. The test outcomes highlight the significant advantages of our approach, particularly in reducing data size, enhancing processing time efficiency, and optimizing network loading speeds. Furthermore, the algorithm has been optimized in accordance with the specific demands of railway engineering, enhancing the accuracy of extraction.

III. BASIC TERMS AND IFC HIERARCHICAL STRUCTURE

A. BASIC TERMS USED IN THE IFC FILE

The IFC is described in EXPRESS language and belongs to an official International Standard ISO 16739:2013. Schema selection is based on choosing the most mature rather than the latest schema specifically favoring the IFC 2×3 specification, as it provides a more stable schema to support our research, which version encompasses 653 Entities, 327 Types, and 312 Property Sets. The IFC physical file serves as the direct medium for information exchange among BIM platforms. It exists in three distinct formats, with the STEP physical file format, typically identified by the ''*.ifc'' extension, being the most prevalent. STEP is rigorously defined by ISO 10303-21, which precisely outlines the composition of the IFC physical file (hereinafter referred to as the IFC file). This composition comprises two integral parts: the header section and the data section, as depicted in Figure 1.

The header section of the file starts with ''Header'' and ends with the first ''ENDSEC'', which mainly describes the

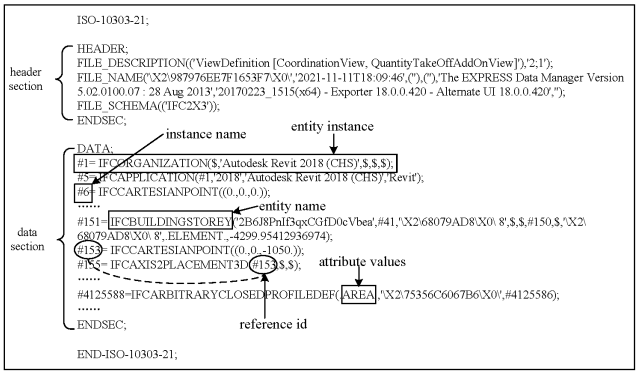

FIGURE 1. The fundamental terminology employed in the IFC file.

time of the creation of the IFC file, the modeling software, and the schema of the IFC used. The file header does not contain specific semantic information in it and exhibits a considerable brevity relative to the data section, thus so only the information in the original file needs to be preserved in the partial model extraction phase.

Commencing with ''DATA'' and concluding with the subsequent ''ENDSEC'', the data section presents the core content of the file, specifies the geometric and semantic information contained in the model, as well as the relationships between them. According to the STEP standard, the data section consists of multiple data instances (referred to as instances), with each entity starting with ''#'' and ending with '';'', which contains the instance name, the entity name, and a comprehensive array of attribute values. The instance name (e.g. ''#6'' and ''#151'') typically consists of numerically encoded identifiers. The segment preceding the initial bracket following the ''='' within the instance number represents the entity name (e.g. '' IFCBUILDINGSTOREY''). The instance attribute item consists of a number of instance attributes (e.g., ''AREA''). Within an IFC file, each entity may correspond to numerous instances, the instance name (e.g., ''#151'') ensures uniqueness and functions as a reference identifier that can be referenced by other entity instances. As shown in Figure, ''#153'' represents an attribute belonging to ''#155''.

B. HIERARCHICAL STRUCTURE OF THE IFC FILE

IFC categorizes entities into two distinct groups: rooted and non-rooted. Rooted entities are derived from the fundamental IfcRoot class, each endowed with a unique GUID and an array of diverse attributes. On the contrary, non-rooted entities do not possess a GUID. Their data instances exist solely when referenced, either directly or indirectly, from a rooted data instance. This hierarchical classification ensures a structured representation of entities within the IFC framework. The hierarchical structure in the IFC data model serves as its fundamental framework, typically with IfcProject, a rooted entity, serving as the root node. Herein, we refer to this structure as the IFC hierarchical structure. Each IFC entity contains attributes and properties.

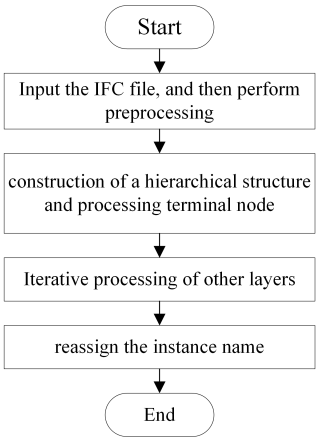

FIGURE 2. IFC partial model extraction algorithm.

The IfcObject instance directly stores its attributes, making them immediately accessible. Among these attributes are GlobalID, Name, Geometry, and Position. Properties are linked to IfcObject entities via relational entities and are not directly accessible from the IfcObject itself, encompassing elements such as material sets and property sets.

Data instances that do not possess any reference IDs within their attributes are designated as terminal nodes and assigned a level of 0. Meanwhile, those data instances that make direct references to level 0 nodes are classified as their parent nodes and assigned a level of 1. Similarly, data instances that function as parent nodes to level n-1 nodes are organized as level n nodes.

IV. THE INSTANCE-BASED IFC PARTIAL MODEL EXTRACTION ALGORITHM

To facilitate precise extraction of partial IFC models, this section delves into an automated, instance -based extraction methodology. This approach involves an iterative traversal of the IFC model's tree structure which is a data structure, for the model extraction. The method first preprocesses the file, subjecting it to a filtering process to obtain a comprehensive list of the railway's devices. Then, the hierarchical structure of the IFC model is constructed. Commencing from the terminal node, it iteratively traverses the file hierarchy employing a depth-first search algorithm. Figure 2 illustrates the primary process, which encompasses the following four key steps.

- **Step 1:** Preprocess the IFC file. This step encompasses the preprocessing of the IFC file, culminating in the procurement of a list of railway engineering needs. This process involves the elimination of redundant information, including unnecessary spaces and duplicate lines, within each individual data instance. At the end of the processing, three basic terms (i.e., instance name, entity name, and attribute value) are extracted from each data instance to determine the desired entity type and associated attributes in preparation for subsequent algorithms.

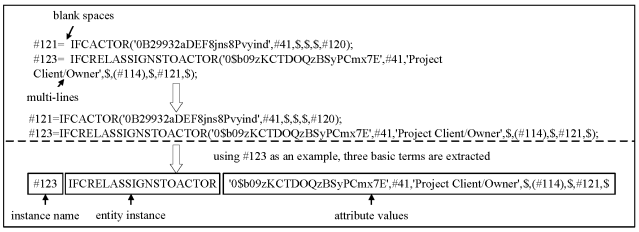

FIGURE 3. IFC file preprocessing.

- **Step 2:** Construct the IFC hierarchical structure and categorize all terminal nodes of the IFC model's tree structure. After completing the preprocessing of STEP 1, the IFC model's tree structure is constructed based on the referential relationships between the instances, and the information on the terminal nodes is collected. Next, update the terminal node to extract the required data.
- **Step 3:** Comprises the iterative and sequential extraction of the remaining data instances by reiterating Step 2. This process, being both recursive and iterative, ceases once the nodes have attained the root node, marking its conclusion. The required instances and associated attributes are recorded in PartialInstances.
- **Step 4:** Lastly, reassign the numerical identifier, serving as the instance name. This rearrangement prioritizes data instances based on their citation frequency, allocating smaller instance names to those that are most frequently referenced.

A. STEP 1: PREPROCESSING DATA INSTANCE SENTENCES IN THE INPUT IFC FILE

As the initial phase of this algorithm, we commence by preprocessing the data instances contained within the input IFC file. This preprocessing aims to achieve a more concise and structured format for each data instance sentence, while also extracting fundamental terminology for subsequent utilization. The principal steps involved in this process are outlined below.

1) Firstly, the input IFC file is traversed line by line, and each instance of data in it is examined and formatted, removing any spaces that may be present in its first and last positions, and converting multiple instances of data into a single line, which is illustrated in the portion above the dotted line in Figure 3.
2) Then, the segmentation of each entity into three parts, instance name, entity name, and attribute values, is stored in the collection in a unified structure, which is illustrated in the section below the dashed line in Figure 3.
3) Next, analyze all the instances to get a list of railway engineering needs, and categorize the diverse instances based on their respective demands. Here the demand data contains geometric and non-geometric information which are attribute and property information, for different demands different sets are needed to be stored.

```
.....
#6= IFCCARTESIANPOINT((0.,0.,0.));
.....
#31= IFCAXIS2PLACEMENT3D(#6,$,$);
#32= IFCLOCALPLACEMENT(#3425134,#31);
.....
#100= IFCAXIS2PLACEMENT3D(#6,$,$);
#101= IFCDIRECTION((6.12303176911189E-17,1.));
#103= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,0.01,#100,#101);
#108= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,*,#103,$,.MODEL_VIEW.,$);

.....
#153= IFCCARTESIANPOINT((0.,0.,-1050.));
#155= IFCAXIS2PLACEMENT3D(#153,$,$);
#156= IFCLOCALPLACEMENT(#32,#155);
.....
#1229= IFCCARTESIANTRANSFORMATIONOPERATOR3D($,$,#6,1.,$);
.....
#1509484= IFCAXIS2PLACEMENT3D(#6,$,$);
#1509485= IFCREPRESENTATIONMAP(#1509484,#1509482);
.....
#1509613= IFCMAPPEDITEM(#1509485,#1229);
#1509614= IFCSHAPEREPRESENTATION(#108,'Body','MappedRepresentation',(#1509613));
.....
#1509621= IFCPRODUCTDEFINITIONSHAPE($,$,(#1509614,#1509619));
.....
#1509628= IFCLOCALPLACEMENT(#156,#1509627);
#1509629= IFCBUILDINGELEMENTPROXY('0lTrLVc595Hfle9Qnk$PSY',#41,'\X2\67956728\X0\1:\X2\67956728\X0\
1:1981734',$,'\X2\67956728\X0\1',#1509628,#1509621,'1981734',$);

.....
#3425133= IFCAXIS2PLACEMENT3D(#6,$,$);
#3425134= IFCLOCALPLACEMENT($,#3425133);
.....
```


FIGURE 4. Partial fragment of the input original IFC file.

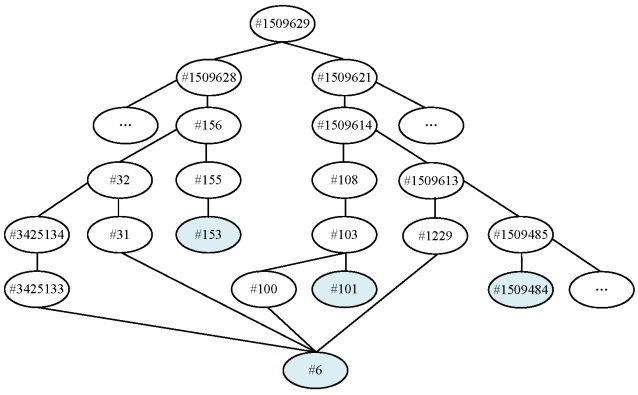

FIGURE 5. Tree structure corresponding to IFC file fragments.

4) Finally, all reference numbers are extracted in the attribute entries and the mapping relationships between them are established in preparation for the subsequent creation of a hierarchical reference structure. Figure 4 a sample of the IFC file fragment is used to describe the extraction algorithm.

B. STEP 2: CONSTRUCT AN IFC HIERARCHICAL STRUCTURE AND CLASSIFY AND EXTRACT ALL TERMINAL NODES OF THE IFC MODEL'S TREE

Firstly, leveraging the established association between instance numbers and reference numbers from Step 1, the hierarchical tree structure of the IFC model is constructed. A data instance like "#6" is identified as a terminal node (level 0), with "#3425133" serving as its parent node (level 1). "#3425134," which is the parent of "#3425133," is categorized as level 2. Figure 5 shows the IFC model's tree structure corresponding to the IFC file fragments, within this structure, the data instances "#6", "#101", "#153", and "#1509484" are identified as terminal nodes.

Secondly, we employ the bottom-to-top iterative traversal approach to access the nodes. We gather all data instances residing on the terminal nodes. Then, we undertake a comparative analysis of the collected terminal nodes and categorize all instances sharing identical entity types into the same MatchedIDMap. This categorization streamlines subsequent attribute value invocations on the instances.

C. STEP 3 UPDATE THE END NODES, ITERATIVELY PROCESS THE REMAINING DATA INSTANCES STEP BY STEP AND EXTRACT PART OF THE MODEL AND RELATED ATTRIBUTE INFORMATION BASED ON REQUIREMENTS

Once the processing tasks of all the terminal nodes have been completed, it is imperative to consider the upper-level parent nodes that make references to these terminal nodes. These parents can only be included in the DealtInstances set after all the instances they reference have been processed, and the next iteration can then begin. The iterative processing process consists of three main sub-steps as described below:

1) Firstly, all instances that have not yet been processed (i.e., instances that are not included in the DealtInstances set) need to be identified first. Then, for the current instance, it is necessary to verify one by one whether all the instances referenced by it have been processed and add them to the DealtInstances set after passing the verification.
2) Secondly, making a judgment about the type of the instance. When an instance is determined to be a physical instance or a relational entity, given the essential differences in attribute extraction between the two, we must adopt a targeted extraction strategy for each. When encountering the physical entity type "IfcOwnerHistory," particular caution should be exercised in selecting instances, avoiding the inclusion of irrelevant ones to maintain the authenticity and precision of the data. Conversely, if the instance does not belong to the above three types, it should be processed according to the standard processing flow of the terminal node.
3) Thirdly, to ensure that the iterative processing of the previous level can proceed smoothly, after each iteration, we must empty all the data in the DealtInstances set. This step is critical to avoid redundancy and confusion and to ensure the consistency and accuracy of the entire process.

1) EXTRACTION OF IFC PHYSICAL INSTANCES AND ASSOCIATED ATTRIBUTES

In this sub-step, the main task is to extract the IFC physical entities and their associated attributes through a hierarchical iterative approach. The iteration commences with physical instances, employing a depth-first traversal strategy to retrieve and enumerate all physical entities within the current layer, which are subsequently accumulated in the DealtInstances set. Subsequently, each physical entity

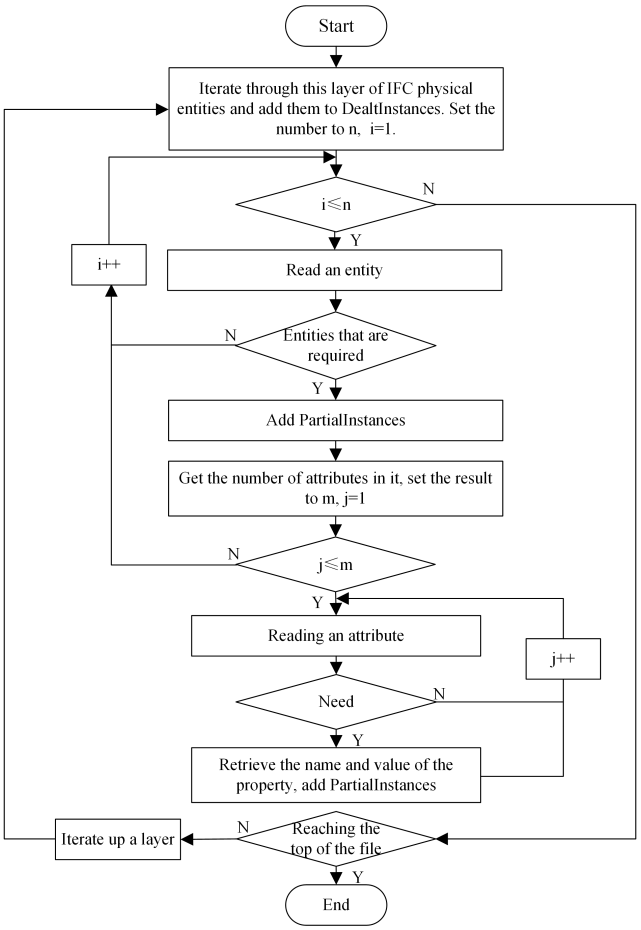

FIGURE 6. Physical entities and related attributes extraction.

undergoes an exhaustive extraction process, involving the retrieval of both its directly and indirectly referenced instances. Attributes that meet the established criteria for desirability are then selectively identified, and their names and values are extracted, appended to the PartialInstances set. Once all entities within the current layer have been thoroughly processed, the DealtInstances set is purged, preparing the system for the subsequent iteration at the next hierarchical level. This recursive sequence of steps is reiterated until the topmost layer is attained, ensuring a comprehensive extraction of IFC physical entities and their pertinent attributes as depicted in Figure 6.

2) EXTRACTION OF IFC RELATIONAL ENTITIES AND RELATED PROPERTIES

This sub-step is focused on extracting IFC relationship entities and their associated properties. This differs from extracting attributes, which can be obtained directly from the IfcObject. In contrast, properties link attributes between relationship entities and IfcObject entities, a process that cannot be accomplished by simply retrieving data from IfcObject. The retrieval should commence from the relational entity and enumerate all relationship entities within the current hierarchy, which are then incorporated into the DealtInstances set. For each relationship instance, three rigorous validation steps are performed: the first step is to verify that all of its referenced instances are present in the PartialInstances set, and to proceed with the extraction operation only if all references in the relationship entity are confirmed; second, the type of the referenced instances is identified, and if the referenced instances belong to the IfcOwnerHistory type, it is regarded as a non-relevant instance and ignored, and only those instances that meet the conditions for further processing are analyzed in-depth; finally, perform the detailed processing flow for the remaining referenced instances, and if the instances meet the pre-set physical entity type extraction criteria, they are not considered at this stage. On the contrary, if the referenced type does not match, its property information is corrected, the physical instance that is not the target of extraction is removed, and a placeholder (“$”) is introduced for compliant substitution if necessary. This substitution process strictly follows the IFC syntax specification, and any deviation from the rules will result in the relational entity being regarded as an irrelevant instance and being excluded. Given that a relational entity may suffer from multiple references from multiple IfcObject entities, this undoubtedly increases the complexity and time-consuming nature of the extraction process. To augment the effectiveness of the extraction procedure, the utilization of HashMap as the fundamental storage for extracted property data is proposed, improving the overall efficiency of the extraction process. During the retrieval phase of properties, HashMap facilitates rapid existence verification, and properties verified for extraction are promptly incorporated into the PartialInstances set. After all relationship instances of the current hierarchy are processed, the DealtInstances set is emptied to prepare for the next round of hierarchical iteration. This recursive iteration continues until the top level, marking the successful completion of the extraction process of relationship entities and their properties. The detailed extraction procedures are outlined in Figure 7, and the resulting partial model file, adhering to these steps, is illustrated in Figure 8.

D. STEP 4: REASSIGN THE INSTANCE NAME

The concluding step of our algorithmic process entails the reassignment of instance names, where a distinct positive integer is assigned to each individual data instance within the IFC file, serving as its unique identifier, such as “#3425133”, with values typically below 263. However, assigning unnecessarily large integers as instance names will lead to an increase in the size of the IFC file, as they generate numerous references, augmenting the overall file size. The instance name holds significance only within the context of the IFC file, and the specific order of these names is immaterial. Therefore, we can optimize the file size by reordering the instance names and reassigning them with smaller integers. Figure 9 illustrates the IFC file after the above steps have been processed.

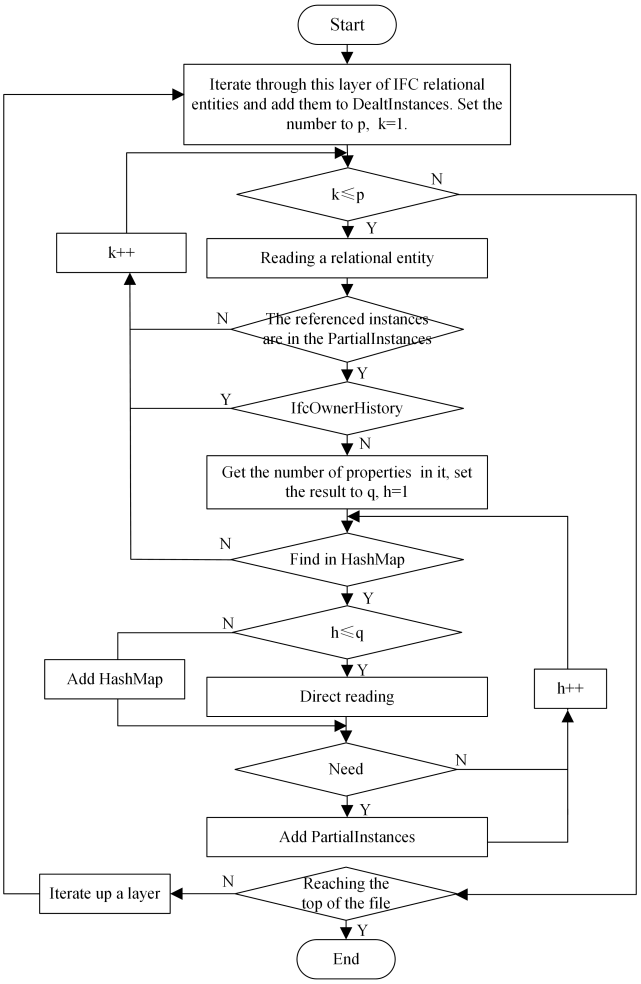

FIGURE 7. Relationship entity and related properties extraction.

```
#1= IFCORGANIZATION($,'Autodesk Revit 2018 (CHS)',$,$,$);
#5= IFCAPPLICATION(#1,'2018','Autodesk Revit 2018 (CHS)','Revit');
#6= IFCCARTESIANPOINT((0.,0.,0.));
#9= IFCCARTESIANPOINT((0.,0.));
#11= IFCDIRECTION((1.,0.,0.));
#13= IFCDIRECTION((-1.,0.,0.));
#15= IFCDIRECTION((0.,1.,0.));
#17= IFCDIRECTION((0.,-1.,0.));
#19= IFCDIRECTION((0.,0.,1.));
#21= IFCDIRECTION((0.,0.,-1.));
#23= IFCDIRECTION((1.,0.));
#25= IFCDIRECTION((-1.,0.));
#27= IFCDIRECTION((0.,1.));
#29= IFCDIRECTION((0.,-1.));
#31= IFCAXIS2PLACEMENT3D(#6,$,$);
#32= IFCLOCALPLACEMENT(#3425134,#31);
#35= IFCPERSON($,'','LL-WORKS',$,$,$,$,$);
#37= IFCORGANIZATION($,'',$,$);
#38= IFCPERSONANDORGANIZATION(#35,#37,$);
#41= IFCOWNERHISTORY(#38,#5,$,.NOCHANGE.,$,$,$,1639726619);
……
```

FIGURE 8. Partial model file of IFC after extraction in step 3.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

To evaluate the performance and efficiency of the CRH-IFCExtractor, the current study selects a portion of the railway engineering IFC models pertaining to a specific railway line in China, designated as Project A. This project encompasses critical elements such as the traction substation,

```
#1= IFCORGANIZATION($,'Autodesk Revit 2018 (CHS)',$,$,$);
#2= IFCAPPLICATION(#1,'2018','Autodesk Revit 2018 (CHS)','Revit');
#3= IFCCARTESIANPOINT((0.,0.,0.));
#4= IFCCARTESIANPOINT((0.,0.));
#5= IFCDIRECTION((1.,0.,0.));
#6= IFCDIRECTION((-1.,0.,0.));
#7= IFCDIRECTION((0.,1.,0.));
#8= IFCDIRECTION((0.,-1.,0.));
#9= IFCDIRECTION((0.,0.,1.));
#10= IFCDIRECTION((0.,0.,-1.));
#11= IFCDIRECTION((1.,0.));
#12= IFCDIRECTION((-1.,0.));
#13= IFCDIRECTION((0.,1.));
#14= IFCDIRECTION((0.,-1.));
#15= IFCAXIS2PLACEMENT3D(#3,$,$);
#16= IFCLOCALPLACEMENT(#86790,#25);
#17= IFCPERSON($,'','LL-WORKS',$,$,$,$,$);
#18= IFCORGANIZATION($,'',$,$);
#19= IFCPERSONANDORGANIZATION(#27,#28,$);
#20= IFCOWNERHISTORY(#29,#2,$,.NOCHANGE.,$,$,$,1639726619);
……
```

FIGURE 9. Result after completing the reassignment of instance names.

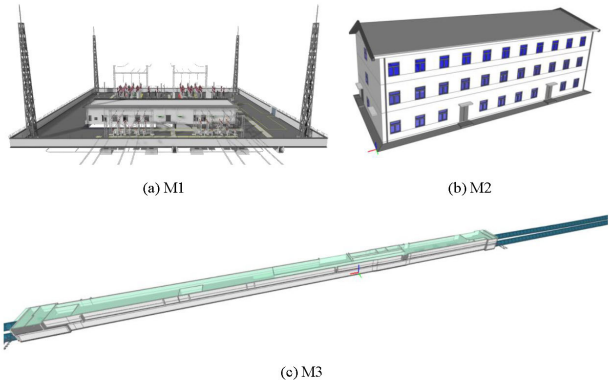

FIGURE 10. The visualization of the IFC file employed for the purposes of experimental testing: (a) traction substation; (b) production complex; (c) station.

production complex, and station. The chosen model comprehensively encompasses the bulk of equipment components pertinent to railway engineering. The BIM model is constructed using Autodesk Revit 2018 and subsequently exported as an IFC file. The original IFC files exhibit sizes of 656.5M, 299.8M, and 139.0M, respectively, encompassing an extensive dataset with over 30.7 million data instances. Figure 10 graphically depicts the corresponding IFC models, designated as M1, M2, and M3. An elaborate summary of the three original IFC files employed in this section is presented in Table 2, including the "Size(MB)" representing the file size in megabytes, "#instances" indicating the number of data instances within each file, and "Time" reflecting the duration required for visualization using BIMVision. The experiments were carried out utilizing a Windows 11

TABLE 2. The details of the three IFC files that were tested in the experiments.

Number	Models	Figure	Size (MB)	Instances	Times (S)
1	M1	11 (a)	656.5	17254897	134.02
2	M2	11 (b)	299.8	9102605	52.46
3	M3	11 (c)	139.0	4349689	21.57

operating system environment, equipped with 16GB of RAM and powered by an i9-13900HX processor, and were facilitated by the synergistic utilization of Visual Studio 2022 and the XBIM toolkit.

A. EVALUATION METRICS

To evaluate the effectiveness of the proposed method, the extraction completion rate and correctness rate are utilized as quantitative metrics for assessing the algorithmic performance.

1) EXTRACTION COMPLETION RATE (ECR)

The extraction completion rate is used to quantify the completion of extraction of partial model data and is defined as

$$Extraction\ Completion\ rate(\%) = \frac{n}{m} \times 100\%. \quad (1)$$

where n is the count of demand-related instances in the partial model and m is the quantity of demand-related instances in the original model, the ECR can quantify that the partial model is extracted completely. When the ECR attains a value of 100%, it signifies that the partial model has been extracted accurately, indicating a successful and comprehensive extraction process.

2) CORRECTNESS RATE (CR)

The Correctness Rate serves as a metric to quantify the degree of accurate extraction achieved in the partial model and is defined as

$$Correct\ rate(\%) = 1 - \frac{p-n}{p} \times 100\%. \quad (2)$$

where p represents the total count of instances within the partial model, the Correctness Rate (CR) serves to quantify the percentage of irrelevant instances present in the extracted partial model. An elevated value of the CR signifies a heightened level of efficacy on behalf of the extraction algorithm.

B. TEST CASES

The experimental results obtained through extraction using three distinct IFC files indicate that the extracted files can primarily be categorized into two groups. The first category comprises fundamental architectural components, such as doors, walls, stairs, and windows. The second category includes specialized component-level equipment related to various specialties in railway engineering, including traction transformers, circuit breakers, pantographs, switch machines, balises, signals, unit combined towers, base station BTS, and various boxes and cabinets.

For the subsequent analysis, representative IFC files were selected from each specialty. The extraction results are presented in Figure 11, specifically: M1_P represents the traction transformer, M2_P corresponds to the base station BTS, and M3_P depicts the ZDJ9 electric switch machine.

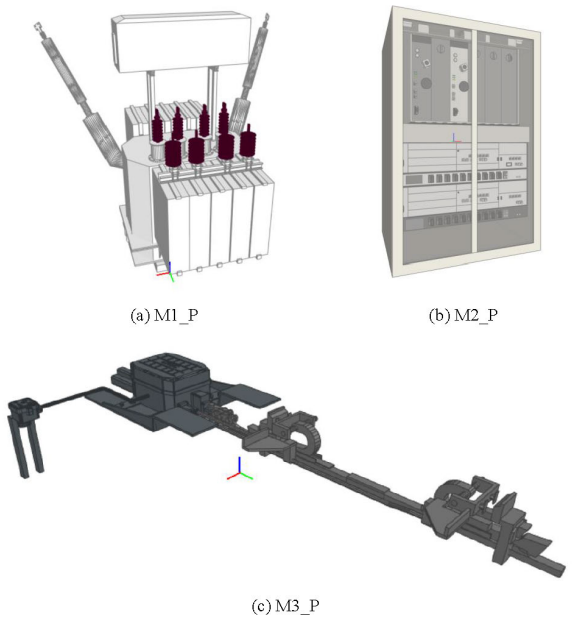

FIGURE 11. The extracted partial model: (a) traction transformer; (b) base station BTS; (c) ZDJ9 electric switch machine.

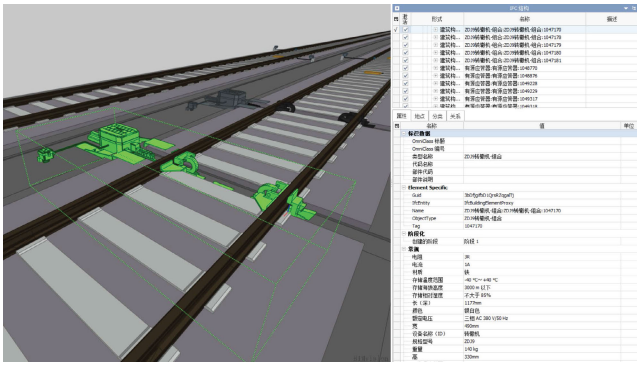

FIGURE 12. Geometric appearance and attributes of the ZDJ9 electric switch machine before extraction.

To ensure the extracted model retains both the integrity of its visual representation and the extraction of crucial attributes, a case study focusing on the ZDJ9 electric switch machine is examined. A comparative analysis is conducted, contrasting the attributes of the original model prior to extraction with the extracted segment, validating the extraction process. Figure 12 shows the geometric appearance of the model prior to extraction, along with its corresponding attributes, whereas Figure 13 demonstrates the geometric appearance of the model following extraction, inclusive of its associated attributes. The comparative analysis reveals that the partial model derived through the algorithm proposed in this study effectively satisfies the extraction criteria, ensuring the complete extraction of the required attributes.

Table 2 provides a comprehensive analysis of our algorithm's extraction performance, encompassing various metrics such as file size, instance count, extraction completion rate, and correctness rate. Among the evaluated files, the M2_P file exhibits the smallest size, whereas the

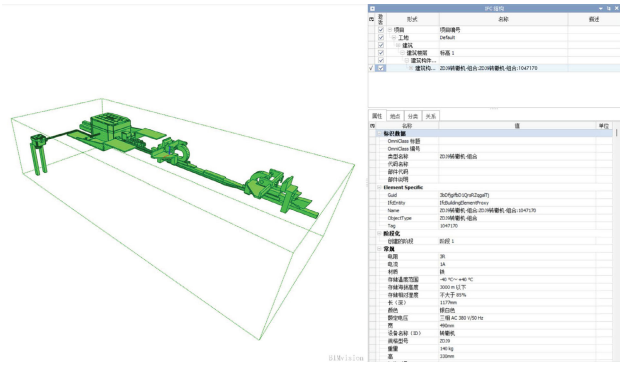

FIGURE 13. Geometric appearance and attributes of the ZDJ9 electric switch machine after extraction.

TABLE 3. The experimental outcomes derived from the application of our proposed algorithm.

Models	Size (MB)	Instances	ECR (%)	CR (%)	Times (S)
M1_P	63.21	2107337	100	99.8	9.57
M2_P	2.46	58643	100	99.6	2.53
M3_P	9.45	321389	100	99.5	5.16

M1_P file is the largest. This discrepancy can be attributed to the diverse content and complexity within each model. The M1_P model incorporates intricate modeling details and extensive texture information, resulting in elaborate geometries that encapsulate nuanced modeling aspects and require a substantial memory allocation. Furthermore, the need to store diverse types of attribute information also contributes significantly to the observed size variance.

Meanwhile, the chosen methods effectively extract the desired model elements by the experimental requirements, achieving a 100% ECR, and ensuring no information loss. Regarding extraction accuracy, there may still be minor elements unrelated to the extraction criteria, a natural occurrence particularly when considering IfcOwnerHistory. However, the algorithm presented in this study has significantly minimized the impact of IfcOwnerHistory, rendering the count of irrelevant instances negligible compared to the total count of required instances.

Finally, the extracted IFC files undergo analysis with regard to loading time and memory optimization performance. The device model chosen by M2_P, exhibits a remarkable reduction in file size, achieving a decrement of 99.82% in comparison to the original model. Furthermore, the model loading time is significantly optimized, achieving a 95.1% improvement. This particular model stands out as the one with the most significant memory reduction and the most pronounced time optimization among the extracted IFC files. Additionally, the remaining elements of the model also demonstrate substantial size reductions exceeding 90% and loading time improvements surpassing 88%. Experimental findings conclusively demonstrate that our algorithm effectively addresses the challenges associated with large IFC files, particularly those pertaining to excessive memory usage and prolonged loading time.

TABLE 4. Semantic syntax validation results.

Models	BIMvision			IfcObjectCounter			Express Engine		
	S	F	U	S	F	U	S	F	U
M1_P	✓	-	-	✓	-	-	✓	-	-
M2_P	✓	-	-	✓	-	-	✓	-	-
M3_P	✓	-	-	✓	-	-	✓	-	-

Note: S=Success, F=Failure, U=Unknown.

C. RESULT VALIDATION

To finalize the validity of the extracted IFC files, it is imperative to conduct rigorous tests encompassing syntax, semantics, and completeness. These IFC files must adhere to national, industrial, and China Railway BIM Alliance standards, along with other pertinent standards. Firstly, it is imperative to obtain a visual representation of each partial model within BIMvision. This visualization allows for a meticulous examination of potential errors within the model's display, enabling us to visually assess the completeness of the model extraction process. Then, syntactic and semantic checks are conducted. Syntactic errors typically involve deviations from the IFC data format or non-compliance with the specification rules defined by ISO. However, IFC files free from syntactic errors may still fail to accurately construct elements in accordance with the extraction intent. Such issues are addressed by evaluating the validity and completeness of the semantics. For this purpose, we have employed the official BuildingSmart release of the IfcObjectCounter and Express Engine.

The validation results are shown in Table 4, where the three IFC files achieve a 100% pass rate in each check. After testing, we confirm that the IFC files extracted by this method show a high degree of correctness in the three key dimensions of semantics, syntax, and completeness. These IFC files are able to accurately express the information of the IFC files, which ensures the reliability of their subsequent practical utilization in various applications.

VI. CONCLUSION

This research is to devise an algorithm capable of efficiently extracting partial models, tailored for various domains, from IFC files. The proposed algorithm is solely grounded on the data organization of the IFC file, without additional formatting of the IFC file or reliance on MVD. It uses an iterative approach to extract the required partial models directly and efficiently by traversing the tree structure of the IFC file. To validate the effectiveness of the proposed algorithm, this study analyzes a case study focusing on a high-speed railway project in China.To assess the performance of the CRF-IFCExtractor algorithm, two crucial quantitative metrics were chosen: the extraction completion rate and the correct rate. Meanwhile, utilizing the Express Engine, the syntactic validity of the extracted models was confirmed, whereas the IfcObjectCounter functioned as a tool for verifying syntactic accuracy and testing semantic integrity. The empirical findings highlight the efficiency of the extraction algorithm, which directly derives precise

partial models from IFC instance models. This approach not only facilitates the rapid extraction and interchange of BIM data but also offers notable advantages. Specifically, it reduces file sizes, improves load time efficiency, optimizes extraction performance, and enhances extraction accuracy.

Additionally, our algorithm, which has undergone preliminary testing on the IFC 2×3, retains potential for further refinement. Its compatibility with higher versions remains unexplored, and we plan to extend the algorithm's functionality by incorporating an interface utilizing the XBIM toolkit in our subsequent study. Meanwhile the current stage has fully satisfied the necessary conditions for promoting the application of railway intelligent BIM. In the future, our research focus will shift to the deep integration of BIM technology and cloud computing technology, aiming to build a BIM cloud platform, which aims to integrate, store, mine, and analyze heterogeneous data throughout engineering's lifecycle, creating a data-driven decision system. It is anticipated that the integration of BIM and cloud computing technologies will significantly contribute to the digital transformation and intelligent upgrading of railway engineering endeavors.

REFERENCES

[1] Y. Dou, T. Li, L. Li, Y. Zhang, and Z. Li, "Tracking the research on ten emerging digital technologies in the AECO industry," *J. Construct. Eng. Manage.*, vol. 149, no. 3, Mar. 2023, Art. no. 03123003.

[2] M. Urbieta, M. Urbieta, T. Laborde, G. Villarreal, and G. Rossi, "Generating BIM model from structural and architectural plans using artificial intelligence," *J. Building Eng.*, vol. 78, Nov. 2023, Art. no. 107672.

[3] J.-Y. Kim, D. Lee, and G.-H. Kim, "Measurement of work progress using a 3D laser scanner in a structural framework for sustainable construction management," *Sustainability*, vol. 16, no. 3, p. 1215, Jan. 2024.

[4] E. Szafranko and M. Jurczak, "Implementability of BIM technology in light of literature studies and analyses of the construction market," *Sustainability*, vol. 16, no. 3, p. 1083, Jan. 2024.

[5] A. Alqatawna, S. Sánchez-Cambronero, I. Gallego, and A. Rivas, "BIM-centered high-speed railway line design for full infrastructure lifecycle," *Autom. Construct.*, vol. 156, Dec. 2023, Art. no. 105114.

[6] O. A. I. Hussain, R. C. Moehler, S. D. C. Walsh, and D. D. Ahiaga-Dagbui, "Minimizing cost overrun in rail projects through 5D-BIM: A conceptual governance framework," *Buildings*, vol. 14, no. 2, p. 478, Feb. 2024.

[7] Z. Ding, L. Luo, X. Wang, Y. Liu, W. Zhang, and H. Wu, "An artificial intelligence-based method for crack detection in engineering facilities around subways," *Appl. Sci.*, vol. 13, no. 19, p. 11002, Oct. 2023.

[8] Y. Liu, H. Lin, Z. Zhao, W. Bai, and N. Hu, "Research on the visualization of railway signal operation and maintenance based on BIM + GIS," *Sensors*, vol. 23, no. 13, p. 5984, Jun. 2023.

[9] A. H. Gourabpasi and M. Nik-Bakht, "BIM-based automated fault detection and diagnostics of HVAC systems in commercial buildings," *J. Building Eng.*, vol. 87, Jun. 2024, Art. no. 109022.

[10] P. Schönfelder, A. Aziz, F. Bosché, and M. König, "Enriching BIM models with fire safety equipment using keypoint-based symbol detection in escape plans," *Autom. Construct.*, vol. 162, Jun. 2024, Art. no. 105382.

[11] A. Sharafat, M. S. Khan, K. Latif, and J. Seo, "BIM-based tunnel information modeling framework for visualization, management, and simulation of drill-and-blast tunneling projects," *J. Comput. Civil Eng.*, vol. 35, no. 2, Mar. 2021, Art. no. 04020068.

[12] S. Xu, J. Wang, X. Wang, P. Wu, W. Shou, and C. Liu, "A parameter-driven method for modeling bridge defects through IFC," *J. Comput. Civil Eng.*, vol. 36, no. 4, Jul. 2022, Art. no. 04022015.

[13] C. Lu, J. Liu, Y. Liu, and Y. Liu, "Intelligent construction technology of railway engineering in China," *Frontiers Eng. Manage.*, vol. 6, no. 4, pp. 503–516, Dec. 2019.

[14] M. Bensalah, A. Elouadi, and H. Mharzi, "Overview: The opportunity of BIM in railway," *Smart Sustain. Built Environ.*, vol. 8, no. 2, pp. 103–116, May 2019.

[15] E. Frías, J. Pinto, R. Sousa, H. Lorenzo, and L. Díaz-Vilariño, "Exploiting BIM objects for synthetic data generation toward indoor point cloud classification using deep learning," *J. Comput. Civil Eng.*, vol. 36, no. 6, Nov. 2022, Art. no. 04022032.

[16] H. Gao, B. Zhong, H. Luo, and W. Chen, "Computational geometric approach for BIM semantic enrichment to support automated underground garage compliance checking," *J. Construct. Eng. Manage.*, vol. 148, no. 1, Jan. 2022, Art. no. 05021013.

[17] *Industry Foundation Classes (IFC)*. BuildingSMART, Hertfordshire, U.K. Accessed: May 28, 2024. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/

[18] M. Laakso and A. O. Kiviniemi, "The IFC standard: A review of history, development, and standardization, information technology," *J. Inf. Technol. Construct.*, vol. 17, pp. 134–161, May 2012.

[19] J. Zhu, P. Wu, and X. Lei, "IFC-graph for facilitating building information access and query," *Autom. Construct.*, vol. 148, Apr. 2023, Art. no. 104778.

[20] G. Gao, Y.-S. Liu, J.-X. Wu, M. Gu, X.-K. Yang, and H.-L. Li, "IFC railway: A semantic and geometric modeling approach for railways based on IFC," in *Proc. 16th Int. Conf. Comput. Civil Building Eng.*, Jul. 2016.

[21] S. Gerbino, L. Cieri, C. Rainieri, and G. Fabbrocino, "On BIM interoperability via the IFC standard: An assessment from the structural engineering and design viewpoint," *Appl. Sci.*, vol. 11, no. 23, p. 11430, Dec. 2021.

[22] J. Sun, Y.-S. Liu, G. Gao, and X.-G. Han, "IFCCompressor: A content-based compression algorithm for optimizing industry foundation classes files," *Autom. Construct.*, vol. 50, pp. 1–15, Feb. 2015.

[23] X. Du, Y. Gu, N. Yang, and F. Yang, "IFC file content compression based on reference relationships," *J. Comput. Civil Eng.*, vol. 34, no. 3, May 2020, Art. no. 04020012.

[24] J. Huo, J. Liu, G. Pei, and T. Wang, "Research on LOD lightweight method of railway four electric BIM model," in *Proc. 6th Int. Conf. Electron. Inf. Technol. Comput. Eng.*, New York, NY, USA, 2022, pp. 492–497.

[25] J. Huo, G. Pei, T. Wang, and J. Liu, "The research of lightweight method for the electric and electronic systems' BIM model," *Proc. SPIE*, vol. 12593, Mar. 2023, Art. no. 125930V.

[26] *IFC Formats. BuildingSMART*. Hertfordshire, U.K. Accessed: May 28, 2024. [Online]. Available: https://technical.buildingsmart.org/standards/ifc/ifc-formats/

[27] H. Xu, J. I. Kim, and J. Chen, "An iterative reference mapping approach for BIM IFCXML classified content compression," *Adv. Eng. Informat.*, vol. 54, Oct. 2022, Art. no. 101788.

[28] *Solibri 9.13.8 Release Notes*. Solibri, Helsinki, Finland. Accessed: May 28, 2024. [Online]. Available: https://www.solibri.com/news/solibri-9-13-8-release-notes

[29] X. Du, F. Zhang, and L. Dong, "A non-redundant BIM sub-model extraction method for IFC files," in *Proc. Int. Conf. Intell. Automat. Soft Comput.*, May 2021, pp. 563–578.

[30] X. Deng, H. Lai, J. Xu, and Y. Zhao, "Generic language for partial model extraction from an IFC model based on selection set," *Appl. Sci.*, vol. 10, no. 6, p. 1968, Mar. 2020.

[31] H. Gou, Y. Zhou, X. Ye, Z. Luo, and F. Xue, "Automated mapping from an IFC data model to a relational database model," *J. Tsinghua Univ. Sci. Technol.*, vol. 61, no. 2, pp. 152–160, Dec. 2020.

[32] W. Solihin, C. Eastman, Y.-C. Lee, and D.-H. Yang, "A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database," *Autom. Construct.*, vol. 84, pp. 367–383, Dec. 2017.

[33] G. Lee, J. Jeong, J. Won, C. Cho, S.-J. You, S. Ham, and H. Kang, "Query performance of the IFC model server using an object-relational database approach and a traditional relational database approach," *J. Comput. Civil Eng.*, vol. 28, no. 2, pp. 210–222, Mar. 2014.

[34] H. Ying, S. H. Lee, and Q. Lu, "Comparative analysis of the applicability of BIM query languages for energy analysis," in *Proc. CIB W78*, Brisbane, QLD, Australia, Oct. 2016, pp. 1–10.

[35] M. Barzegar, A. Rajabifard, M. Kalantari, and B. Atazadeh, "An IFC-based database schema for mapping BIM data into a 3D spatially enabled land administration database," *Int. J. Digit. Earth*, vol. 14, no. 6, pp. 736–765, Jan. 2021.

[36] S. Wu, Q. Shen, Y. Deng, and J. Cheng, "Natural-language-based intelligent retrieval engine for BIM object database," *Comput. Ind.*, vol. 108, pp. 73–88, Jun. 2019.
[37] J. Beetz, J. van Leeuwen, and B. de Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies," *Artif. Intell. Eng. Design, Anal. Manuf.*, vol. 23, no. 1, pp. 89–101, Dec. 2008.
[38] L. Zhang and R. R. A. Issa, "Ontology-based partial building information model extraction," *J. Comput. Civil Eng.*, vol. 27, no. 6, pp. 576–584, Nov. 2013.
[39] M. Venugopal, C. M. Eastman, and J. Teizer, "An ontology-based analysis of the industry foundation class schema for building information model exchanges," *Adv. Eng. Informat.*, vol. 29, no. 4, pp. 940–957, Oct. 2015.
[40] T. M. D. Farias, A. Roxin, and C. Nicolle, "A rule-based methodology to extract building model views," *Autom. Construct.*, vol. 92, pp. 214–229, Aug. 2018.
[41] J. Won, G. Lee, and C. Cho, "No-schema algorithm for extracting a partial model from an IFC instance model," *J. Comput. Civil Eng.*, vol. 27, no. 6, pp. 585–592, Nov. 2013.
[42] N. Gui, C. Wang, Z. Qiu, W. Gui, and G. Deconinck, "IFC-based partial data model retrieval for distributed collaborative design," *J. Comput. Civil Eng.*, vol. 33, no. 3, May 2019, Art. no. 04019016.

YUNSHUI ZHENG received the B.S. degree from the School of Automation and Electrical Engineering, Lanzhou Jiaotong University, China, in 1994. He is currently an Associate Professor with the School of Automation and Electrical Engineering, Lanzhou Jiaotong University, Lanzhou, China. He is the Deputy Director of the Key Laboratory of Railway Industry of BIM Engineering and Intelligent for Electric Power, Traction Power Supply, Communication and Signaling, Lanzhou Jiaotong University. His current research interests include railway automatic control, reliability of railway systems, and railway BIM technology.

YIMIN SHI received the B.E. degree in rail transit signal and control from Lanzhou Jiaotong University, Lanzhou, China, where she is currently pursuing the M.E. degree in transportation engineering. Her current research interests include building information modeling, railway transportation informatization, and intelligent railway.

XINKAI WANG received the B.E. degree in electronic science and technology from Hunan Institute of Engineering, Xiangtan, China, and the M.E. degree in transportation engineering from Lanzhou Jiaotong University, Lanzhou, China. His research interests include rail circuits and rail transportation automation.