

Article

Generic Language for Partial Model Extraction from an IFC Model Based on Selection Set

Xueyuan Deng ¹, Huahui Lai ^{2,*}, Jiayi Xu ¹ and Yunfan Zhao ¹

¹ Department of Civil Engineering, School of Naval Architecture, Ocean and Civil Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; dengxy@sjtu.edu.cn (X.D.); xiaoyi1213@sjtu.edu.cn (J.X.); zhaoyunfan@sjtu.edu.cn (Y.Z.)

² Shenzhen Municipal Design & Research Institute Co., Ltd., Shenzhen 518029, China

* Correspondence: laihuahui81665@alumni.sjtu.edu.cn

Received: 31 January 2020; Accepted: 11 March 2020; Published: 13 March 2020



Abstract: During data sharing and exchange of building projects, the particular business task generally requires a part of the complete model. This paper adopted XML schema to develop a generic language to extract the partial model from an Industry Foundation Classes (IFC) model based on the proposed Selection Set (called PMESS). In this method, the Selection Set was used to integrate users' requirements, which could be mapped into IFC data. To ensure the validity of the generated partial IFC models in syntax and semantics, seven rules—including three basic rules for a valid IFC file, three extraction rules based on the Selection Set, and a processing rule for redundant information—were defined. Through defining PMESS-based configuration files, the required data can be extracted and formed as a partial IFC model. Compared with the existing methods, the proposed PMESS method can flexibly extract the user-defined required information. In addition, these PMESS-based configuration files can be stored as templates and reused in other tasks, which prevents duplicated work for defining extraction requirements. Finally, a practical project was used to illustrate the utility of the proposed method.

Keywords: Building Information Modeling (BIM); Industry Foundation Classes (IFC); partial model extraction; query language; selection set

1. Introduction

In traditional computer-aided design (CAD) practices, most interdisciplinary data exchanges take place with two-dimensional (2D) drawings, documents, or reports. As unstructured forms, these data files are hardly used by other software tools, resulting in remodeling work for their own tasks. Due to the lack of semantic meanings of elements (e.g., points, lines, and planes) in electronic documents, designers have to manually interpret these elements based on their experience in order to identify and extract the required data [1]. The CAD-based process limits the reusability of project data through the whole building life cycle. Compared with the traditional CAD-based method, Building Information Modeling (BIM) technology is able to represent the geometry, properties, and relations of building objects based on the object-oriented method [2]. In the early stage of BIM technology, its purpose is to make a complete model available for every participant. Nowadays, the use of BIM technology in the architecture, engineering, construction, and facility management (AEC/FM) industry is becoming more widespread [3], and it results in numerous structured data in various domains, which can be interpreted by different software tools and used for different business tasks. The business task means that participants use some project data to carry out some activities for a professional application, and it can take place during the building lifecycle. It is useful for participants to use the complete model because of its rich and precise information.

However, as projects become larger and more complicated, project information increases dramatically [4], resulting in the huge file size of the BIM model. It is difficult for participants to use such a large model, and it inevitably takes much time to process the complete model for their required information. In general, the designer/engineer requires a part of model data for their own business tasks, rather than a complete one. For example, the structural engineer mainly focuses on the structural objects (such as columns, walls, and slabs) from the architectural model for structural design and analysis, rather than the overall architectural model. The method or system to automatically extract the required information from BIM models can improve the quality and productivity by preventing unnecessary work. Due to the lack of effective tools to extract the required data, the designer/engineer generally deals with the original model in manual way. Such a large body of information makes it difficult for designers/engineers to directly process, leading to the inefficiencies in data sharing and exchange between software tools. Extracting the required data from the original BIM model has become one of the problems that must be addressed in BIM uses [5].

In general, commercial software tools in the AEC/FM industry have the functions of querying or extracting objects. However, only specific partial model data which have been defined in software tools can be exported. Or designers/engineers can use embedded filtering in BIM authoring tools to select the components of interest (e.g., only make the structural components visible, only make the beams visible), and then save this model for use. In this way, it is different to extract other model data according to specific requirements or purposes (e.g., only extract the concrete components). Consequently, a number of plug-ins in designated software tools were developed to extract the required information. It is noted that their functions are unavailable for other software tools, only can be used in specific software tools. Although the objects can be selected one by one manually for extraction, this process is cumbersome and prone to error. In addition, the manual extraction method cannot be stored as templates for reuse. Hence, extracting partial models based on a public data schema is inevitable in order to meet requirements from diverse business tasks.

Industry Foundation Classes (IFC) was developed to support the full range of data exchange [6]. Many studies related to partial model extraction have been carried out, and more details about these studies will be presented in the following section. However, most methods were developed to extract specific model data for specific business tasks. The users hardly extract model data by their intents, and sometimes manually select the required objects. It is necessary to develop an innovative method to extract the required model data based on the users' requirements. In this paper, a generic language was designed to extract partial models from IFC models based on the eXtensible Markup Language (XML) format. By using the proposed language, users can design a configuration file to define extraction requirements, and then the required model data is automatically extracted and formed as a partial model. The proposed method supports diverse definitions of extraction requirements, including object types, attributes, and relations. To make the definitions of extraction requirements more rigorous and standardized, the Selection Set was proposed to represent extraction requirements. Furthermore, mathematical logic and set theory [7] were used to describe rules for partial model extraction, so IFC data with multiple representations could be processed to form valid partial IFC models.

The rest of the paper is organized as follows. A review of related work for partial model extraction from two aspects (that is, task-specific and user-defined methods) was first introduced. Second, according to comparative analysis, the concept of the Selection Set was proposed to integrate users' requirements. Third, seven rules in the extraction of syntactically and semantically valid partial IFC models were designed. Subsequently, with the adoption of XML schema, a generic language was developed for partial model extraction from an IFC model based on the Selection Set (PMESS), and then the proposed method was validated through a test case. The final section summarizes the most important conclusions.

2. Related Work

The methods of extracting partial models could be classified into different areas [8,9]. This section mainly presents these methods according to the task-specific and user-defined requirements.

2.1. Partial Model Extraction According to Specific Tasks

At present, many commercial software tools have developed data interfaces for specific tasks, such as Revit, ArchiCAD, and Tekla Structures. The required objects and attributes can be extracted from the original model by using these tools. However, these data interfaces could not be applied to other software tools. As commercial software tools, their algorithms or codes are not public, so it is difficult to modify these algorithms for extracting different model data by users' intention. Besides the native model filtering for export, some software products provide the functionality for exporting IFC files. For example, IFC Translator in ArchiCAD [10] can be used to export different IFC models according to options: (1) selected elements only; (2) visible elements, on either all stories or the current story; (3) entire project. The first and second options are used for partial model extraction, and the third option is for the complete one. Similarly, a functionality called 'IFC Export Setup Options' exists in Revit [11], which can export only elements visible in the view. The aforementioned functionality is mainly used to extract physical objects according to the type or view. However, it can hardly extract the partial model according to other requirements, such as relationships and the required attributes.

IFC is a de facto standard to support a full range of business tasks in the AEC/FM industry [12]. It is a rich schema for representing diverse information throughout the building life cycle. Rather than exchange requirements (ERs) for specific tasks, IFC schema focuses on the complete building information among all actors at every stage of a building. Consequently, Information Delivery Manual (IDM) and Model View Definition (MVD) were proposed by buildingSMART International (bSI). The main purpose of IDM is to define exchange requirements of specific tasks in a non-technical term, and as a subset of IFC schema, MVD is a technical standard that translates IDM-based exchange requirements to Model Views for software implementation. Hence, software tools are able to export required partial models by defining different MVDs. So far, bSI has published several MVDs, such as IFC4 Reference View and IFC4 Design Transfer View [13]. Based on the approach of the Georgia Tech Process to Product Modeling (GTPPM) [14], Lee et al. [15] developed an eXtended Process to Product Modeling (xPPM) tool to automatically generate process maps (PMs), ERs, functional parts (FPs), and MVDs. The xPPM promotes consistent and reliable implementations of IDM and MVD. The implementation of IDM and MVD standardizes exchanged information in domain-specific tasks. For example, the recent version of software Revit is capable of exporting IFC models according to a set of MVDs, such as Ifc2 × 2 Singapore BCA e-Plan Check, Ifc2 × 3 Coordination View 2.0, Ifc2 × 3 Basic FM Handover View, and IFC4 Reference View. However, except these mentioned MVDs, other published ones have not been widely supported by BIM software tools yet. There are several tools to define and document new MVDs, such as IfcDoc [16] and ViewEdit [17]. It is useful for users to generate their own MVDs according to the requirements of business tasks. However, it still needs additional work to develop corresponding software tools to realize these MVDs.

Currently, there are numerous research projects on how to efficiently extract geometric information from data models [18–20]. However, business tasks require different kinds of information, not just geometric information. Hence, different frameworks and methods were developed to extract the required information for specific business tasks. For example, in the structural domain, some information—such as the type, location, geometry, and material of objects—should be extracted from architectural models. Qin et al. [21] proposed the Structural General Format (SGF) based on XML and developed an algorithm to automatically extract structural information from IFC-based architectural models to generate SGF-based models. Besides IFC data format, the SGF-based model could be translated into different finite element models for structural analysis. Hu et al. [22] proposed an IFC-based Unified Information Model with conversion algorithms between the architectural and structural models, and among various structural analysis models. Besides the structural domain,

other business tasks also need to extract information from upstream models, e.g., energy simulation [23], construction schedule [24], etc. The aforementioned studies mainly extract the required information for specific business tasks.

2.2. Partial Model Extraction According to User-Defined Requirements

Due to the multidisciplinary, multi-stage and multi-party nature of building projects, business tasks require different information. The studies related to partial model extraction can be divided into two types: the schema level-based and the instance level-based methods. The former method is to develop a definition format with various mappings for data exchange, and the latter directly deals with the data within the original model. The schema level-based method refers to extract model data according to a predefined model data structure. Consequently, it is necessary to define all possible transformations in advance. The instance level-based method focuses on the specific information of project objects, and this method enables the extraction of some designated model data according to the user-defined requirements. In the IFC file, the specific information of objects means corresponding IFC instances. This method gives users enough flexibility and well-understanding, but more complex querying algorithms are required than the schema level-based method.

2.2.1. Methods at the Schema Level

Partial Model Query Language (PMQL) [25] and Generalized Model Subset Definition schema (GMSD) [26] are two early partial model extraction methods based on the schema level. PMQL was developed based on XML, Structured Query Language (SQL), and Simple Object Access Protocol (SOAP). It aims to extract a partial instance model from the original IFC model through the select, update, and delete operations. Inspired by PMQL and SPARQL Protocol and RDF Query Language (SPARQL), Mazairac and Beetz [9] adopted the BIM Server to develop an open query language (BIMQL, Building Information Model Query Language). The purpose of this language is to select, set, create, and delete IFC model data for managing BIM models, and the functions ‘select’ and ‘set’ have been developed. However, IFC schema includes numerous logical relations. When extracting specific relations using the PMQL, it requires many iterative cycles of request and response [26]. Furthermore, the PMQL method has room for improvement, such as path expression, nested queries, and inheritance hierarchies. As a result, GMSD was developed to support the dynamic selection of object instances and the filtering of a building model through predefined model view definitions. GMSD was designed to support EXPRESS-based models for consistency with IFC. Users need to define or edit MVDs within the GMSD method, but the MVD definition is a challenge for users. To solve this problem, bSI developed an official standardization specification data format for capturing MVDs based on the XML format, called mvdXML [27]. The mvdXML is a machine-interpretable representation for information exchange in IFC schema, and can be easily processed by software tools. More and more software tools are expected to support mvdXML. Inspired by this method, the proposed method in this paper was also developed based on the XML.

The new model view definitions generated based on the schema level need to be validated in the syntax and semantics, so Yang and Eastman [28] defined a serial of rules for subset generation by using set theory, aiming at supporting specific exchanges through the generation of valid model views. Furthermore, Lee [29] proposed the ‘minimal set’, the smallest complete subset of a schema related to a concept. Several conditions for extracting valid subsets from EXPRESS schema were defined to match the concepts. A tool called ‘IFC Model View Extractor’ (alpha version) was developed for generating subschema from IFC schema. According to subset generation rules [28], Yu et al. [30] proposed a semi-automatic generation method for MVDs, which could extract partial models according to core concepts of specific tasks. However, these core concepts need to be accurately predefined by users. In this study, with reference to the rule-based subset generation method [28], the rules for Selection Set-based partial model extraction were designed based on mathematical logic and set theory.

Some other researchers attempted to convert IFC models into a generic data schema. Given the requirement on spatial analysis, Daum and Borrmann [31,32] carried out a topological analysis of BIM models and proposed Query Language for Building Information Models (QL4BIM). This method enables users to extract the partial model by defining boundary representation. Fuchs and Scherer [33] developed a language called Multi-Model Query Language (MMQL), which required homogeneous data access to link and filter multi-model information, such as the bill of quantities, building, and schedule. Nevertheless, the export results are documented in textual format, rather than IFC-based data format or other model formats. Zhang and El-Gohary [34,35] developed an automated BIM Information Extraction method to extract the required information from IFC models with semantic Natural Language Processing techniques and Java Data Access Interface. Some limitations still exist in this method. For example, IFC relations in the extracted model are not yet fully aligned with the proposed semantic logic-based representation. Pauwels and Terkaj [36] proposed a procedure to convert IFC EXPRESS schema to an IfcOWL ontology for construction industry.

2.2.2. Methods at the Instance Level

The extraction method at the schema level always needs to be defined in a formal language for the data schema [15]. It is a complex and difficult task for users in the AEC/FM industry. Methods at the instance level (e.g., the object, property, etc.) were proposed to meet user-defined exchange requirements. Katranuschkov et al. [8] adopted the semantic query method to extend the GMSD definition at the instance level and developed Multi-model View Generator to extract partial models from BIM and non-BIM data. Based on the GMSD work, Windisch et al. [37] proposed a generic framework for consistent generation of BIM-based model views, which aims to provide the filtering at class and object level, and the generation of ad-hoc and multi-model views. However, the relations between these levels need to be further studied and implemented. To avoid the definition of data schema, Won et al. [38] proposed a no-schema algorithm to extract a partial model from an IFC model depending on user-defined object types or predefined ERs. The current version could not extract the partial model under combinatorial conditional expressions.

Currently, there are some good open source software libraries that help users and software developers to work with BIM IFC files, such as IfcOpenShell [39], xBIM [40], and IfcPlusPlus [41]. The user can use one of these IFC libraries to read IFC files according to the requirement, and IfcPlusPlus was selected as the IFC library in this study.

2.3. Summary on Related Research Works

The partial model extraction methods in the first subsection are mainly used in domain-specific tasks. The second subsection presents two types of generic partial model extraction methods related to user-defined requirements. One method is to extract partial models through definitions at the schema level. Even though it is general enough to meet various requirements of business tasks, users are required to be familiar with definitions within these methods. The other provides the selection function to extract partial models at the instance level. Nevertheless, the current tools mainly extract some common physical elements, but not fully support the extraction under some restrictions.

To support the extraction with user-defined requirements, the Selection Set was proposed to integrate user-defined requirements, and the XML format was used to design a generic language to automatically extract partial models. By using the proposed method, the required objects and their attributes can be extracted from the original model according to the user-defined requirements, and other objects that are not required can be filtered out. The key characteristics of the proposed method compared to related studies are summarized as follows:

- (1) The mathematical logic and set theory were used to define Selection Set and extraction rules for partial model extraction. The mappings between IFC data and user-defined requirements were developed by using the mathematical method, ensuring the stability of the proposed algorithm. When the version of IFC schema is updated, only some definitions in the Selection Set or extraction

rules need to be updated or revised, rather than the entire extraction algorithm. The structure of the proposed language is stable and independent of IFC versions.

- (2) A generic language for partial model extraction was designed based on Selection Set (PMESS). In order to extract different model data, data extraction requirements were analyzed and classified according to IFC schema. Subsequently, the technical structure of the partial model extraction language was developed by using the software-independent XML schema. The purpose of Selection Set is to standardize and integrate the users' extraction requirements, and its elements can be used to map into different user-defined requirements. By using the proposed language, the partial model can be extracted according to the user-defined extraction requirements, including objects, properties, and relations.

3. Concept of the Selection Set

During the process of partial model extraction, the software tool firstly identifies the extraction requirements defined by users, then extracts information which meets the requirements, and finally forms a valid data model based on the extracted information. Therefore, the extraction requirements can be regarded as input parameters. In this study, user-defined extraction requirements are integrated into the Selection Set, which can be assumed as some basic sets with specific semantic to extract partial models.

The Selection Set is an information set that is formed based on the requirements, such as object types, attributes, relations, and mixed ones. Elements in the Selection Set are used as input parameters of the proposed method. According to referencing relations between IFC data, the proposed method queries IFC data based on input parameters and then exports the required IFC model data, that is, a partial model or sub-model.

Extraction requirements can be classified as different semantics and relationships, such as object types, properties, relations, and mixed cases. In terms of data representation in the IFC schema, the entities and rules can be used to describe these extraction requirements. Hence, the first condition for the Selection Set is defined as follows:

Condition 1: A Selection Set includes a set of Entities and Rules, and has at least one Entity.

$$\exists e \exists r \forall S [[e \in E \wedge r \in R \wedge S = \{E, R\}] \rightarrow |S| \geq 1] \quad (1)$$

where e is ENTITY; r is Rule; S is Selection Set; E is a non-null set of Entities; and R is a set of Rules.

In an IFC model, every IFC instance represents a specific meaning, and it is illegal to have an abstract IFC entity in the IFC model. The proposed method is to query and extract IFC instances from the IFC model according to the Selection Set, so the abstract IFC entities should not be included in the Selection Set. The Condition 2 for Selection Set is listed as follows. This condition is similar to the rule BR02 defined by Yang and Eastman [28].

Condition 2: A Selection Set cannot include an abstract entity.

$$\forall e [e \in S \rightarrow e \notin A_{abs}] \quad (2)$$

where A_{abs} is a set of Abstract entity data types.

Business tasks require diverse information, so a set of exchange requirements may need to be defined for partial model extraction. In some cases, the partial model may only need to meet one of many Selection Sets, while other cases require to meet many Selection Sets. In conclusion, the former relation among Selection Sets is 'union', and the latter one is 'intersection'. Hence, the theorem for forming new Selection Sets is defined as follows:

Theorem 1: Forming new Selection Sets

The union of many Selection Sets is still a Selection Set, and the intersection of many Selection Sets is also a Selection Set. Let s_i denote a Selection Set:

$$(1) \forall s_i [[s_i \subset S \wedge \bigcup_{i \geq 1} s_i = s] \rightarrow s \subset S]; \text{ and}$$

$$(2) \forall s_i [[s_i \subset S \wedge \bigcap_{i \geq 1} s_i = s] \rightarrow s \subset S].$$

Proof:

- (1) Let a be an element of $s = \bigcup_{i \geq 1} s_i$, that is, $a \in s$. There is at least one s_i , so that a belongs to s_i .

Because $a \in s_i \subset S$ and $\forall a \in s, s \subset S$;

- (2) Let b be an element of $s = \bigcap_{i \geq 1} s_i$, that is, $b \in s$. For any s_i , b belongs to s_i .

Because $b \in s_i \subset S$ and $\forall b \in s, s \subset S$. \square

4. Rules for Partial Model Extraction

The output of the partial model extraction method is the IFC file, so the file must comply with the IFC schema. During the extraction process, the proposed method is to process the original IFC model depending on rules for partial model extraction and then export the partial IFC model by integrating required model data. These rules for partial model extraction can be categorized into three types: basic rules for a valid IFC file, extraction rules based on Selection Set, and processing rule for redundant information, as shown in Figure 1.

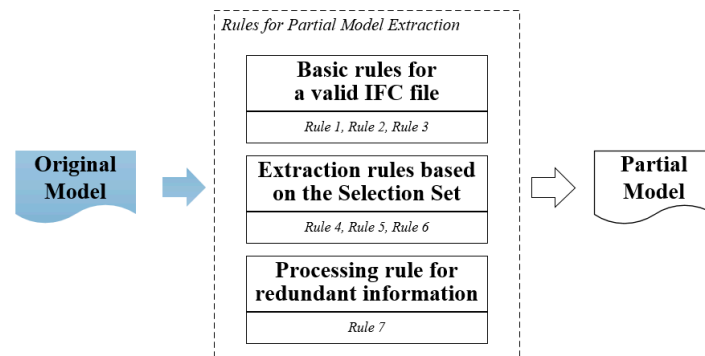


Figure 1. Rules for partial model extraction.

Basic rules for a valid IFC file: The basic rules refer to the fundamental requirements that should be complied with when an IFC file is formed. The proposed method is to export an IFC file, so the extracted partial model should also comply with these basic rules.

Extraction rules based on Selection Set: The extraction requirements were included in the Selection Set. In order to query and extract the matching IFC data, the mapping from elements in the Selection Set to IFC model data was developed.

Processing rule for redundant information: Required data could be identified according to the basic rules and extraction rules. In the final step to export the IFC file, the information that is not required by business tasks should be filtered to ensure the validity of the IFC file.

Traditionally, the existing extraction methods generally process BIM data based on one IFC version. When the IFC version is updated, it is clear that defining a mapping between one version and the updated one is a major undertaking. It may be needed to modify the corresponding algorithm. Through the proposed rules, the data processing of partial model extraction was divided into different steps. It can significantly reduce the modification work of the proposed algorithm because of the

updated IFC version, ensuring the stability of this method. The data processing flow can be concluded as follows. The first step is to query corresponding IFC data according to the user-defined requirements. These requirements (such as object types, properties, and relations) can be obtained from the Selection Set. In Step 2, the target object will be located through the IFC reference relationship, and all of its corresponding attributes will be remained. Finally, all the target objects and their attributes will be extracted to form a new IFC file. The following subsection describes the proposed rules in detail.

4.1. Basic Rules for a Valid Industry Foundation Classes (IFC) File

The *IfcProject* is an important entity in an IFC file. It is not only a foundation of space structure entities (such as *IfcSite*, *IfcBuilding*, and *IfcBuildingStorey*), but also contains unit, owner history, geometric representation and other basic information of a building project. An IFC file has only one *IfcProject* entity [42]. Based on this entity, some basic project information can be queried from the IFC model data, such as site, building, and unit. Hence, a partial model should contain the *IfcProject* entity and related entities which are referenced by attributes of the *IfcProject*.

Rule 1: The partial model has only one *IfcProject* entity and an entity set which consists of other entities referenced by attributes of the *IfcProject*.

$$\forall M_o[[[e_{pro} \in M_o] \wedge |e_{pro}| = 1] \rightarrow R_{pro} \subset M_p] \quad (3)$$

where M_o is the original model; e_{pro} is the *IfcProject* entity; R_{pro} is a set of entities referenced by *IfcProject* entity's attributes; and M_p is the partial model.

In the IFC schema, the IFC entity mainly contains explicit and inverse attributes [43]. The explicit attributes are scalar values or the information computed from other attributes, while the inverse ones are identified relationally through other entities.

To ensure the completeness of the partial model, when extracting one designated entity (called 'target entity'), all entities referenced by the explicit attributes of the target entity should be extracted together. The entity set consisting of these referenced entities is assumed to define as the Essential Set (ES) of the target entity.

Rule 2: The Essential Set of the target entity is included in the partial model.

$$\forall e[e \in S \rightarrow R_e \subset M_p] \quad (4)$$

where R_e is the entity set referenced by the attributes of e .

Besides explicit attributes, some information of the target entity is represented by other IFC instances defined in inverse attributes. Similarly, the entities defined in inverse attributes of the target entity should be extracted. According to the referencing and inheritance structure of the IFC model, the entities in the ES also need to be queried to find out corresponding entities defined in inverse attributes. It is noteworthy that some particular IFC entities are used to represent basic information of a building project, which may be referenced by many IFC instances. A representative entity is the *IfcOwnerHistory*. If these entities were queried to search entities defined in inverse attributes, some entities that were not required would be extracted as well as some repeat entities. To avoid this situation, these entities comprising the Particular Set (PS) were designed as ending points of the query process. Besides the *IfcOwnerHistory*, the basic entities—such as *IfcDirection*, *IfcCartesianPoint*, *IfcAxis2Placement3D*/*IfcAxis2Placement2D*, *IfcLocalPlacement*, and *IfcGeometricRepresentationContext*/*IfcGeometricRepresentationSubContext*—were set as the particular entities in this study. When encountering these particular entities during the query of inverse attributes, the proposed method will stop the running and enter into the next query. After this step, the non-target IFC entities cannot be extracted. The entity set including entities in the ES and entities defined in inverse attributes is assumed to be the Individual Set (IS) of the target entity. The IS contains complete information of the target entity.

Rule 3: The partial model includes entities defined in inverse attributes of the target entity, which are not queried from the Particular Set.

$$\forall e[[\exists R_{inv}[e \in E_{es} \wedge e \notin E_{ps} \wedge R_{inv}(e, e_{inv})] \wedge e_{inv} \in E_{inv}] \rightarrow E_{inv} \subset M_p] \quad (5)$$

where e_{inv} is an entity defined in inverse attributes of the target entity e ; R_{inv} is the inverse relation from e to e_{inv} ; E_{ps} is a set of particular entities in the Particular Set; and E_{inv} is a set of required entities e_{inv} .

4.2. Extraction Rules Based on Selection Set

According to Condition 1, the Selection Set has at least one Entity e . In the IFC model, all IFC instances matching the Entity e should be extracted. Furthermore, other IFC instances related to explicit and inverse attributes of the target entity should be extracted according to Rule 2 and Rule 3.

Rule 4: IFC instances matching Entity e in the Selection Set are contained in the partial model.

$$\forall e[e \in S \wedge \exists f_M[f_M(e) = E_e] \rightarrow E_e \subset M_p] \quad (6)$$

where E_e is the set of entities related to Entity e ; and f_M is a function from e to E_e , working on original model M_0 .

Numerous complex relations exist between various objects in a building project. For example, the binary relation can be divided into different types of relations, such as containment, parallel, and crosscutting relations. Consequently, relation entities should be set into Selection Set. All corresponding IFC instances within the user-defined relations in the Selection Set would be extracted to form the partial model. In this study, the object which contains other object(s) or is relied by other object(s) is set as the relating object, and other object(s) are called related object(s).

Rule 5: The corresponding relating entity and related entities are included in the partial model, when a relation entity is included in the Selection Set.

$$\begin{aligned} &\forall e_{rel} \exists e_{relating} \exists E_{related} \\ &[[e_{rel} \in S \wedge IFCREL(e_{rel}, e_{relating}, E_{related})] \\ &\rightarrow [(e_{relating} \cup E_{related}) \subset M_p]] \end{aligned} \quad (7)$$

where e_{rel} is a relation entity; $e_{relating}$ is a relating object entity; $E_{related}$ is the set of related object entities; and $IFCREL$ is a function to test if the relation entity e_{rel} exists between relating object entity $e_{relating}$ and related object entities $E_{related}$.

The attributes of IFC entity represent different essential characteristics from other entities. These attributes can form different rules. As a result, the according model data can be extracted by designing different rules.

Rule 6: The entities included in the partial model satisfy the rules in the Selection Set.

$$\forall r[r \in S \rightarrow r \cap M_p] \quad (8)$$

where r is a Rule.

4.3. Processing Rule for Redundant Information

According to elements in the Selection Set (Rule 4, 5, and 6), the matching IFC instances can be extracted from the original model, while the related necessary IFC instances are extracted based on Rule 1, 2, and 3. The IFC entities which are undefined in the Selection Set or cannot be inferred from the Selection Set are not supposed to be included in the partial model. These entities are called redundant

information in this study. The redundant information, including IFC instances and related attributes, must be filtered before the export of the partial model.

Rule 7: The partial model cannot include entities which are not defined or inferred in the Selection Set.

$$\forall \neg e [[\neg e \in M_o \wedge (\neg e \notin S)] \rightarrow \neg e \notin M_p] \quad (9)$$

where $\neg e$ is an Entity undefined in the Selection Set or cannot be inferred from the Selection Set.

5. Generic Language for Partial Model Extraction Based on the Selection Set

To ensure the proposed method to be interpreted by software tools, the XML schema was adopted to define a generic language for partial model extraction based on the Selection Set (PMESS). The overall architecture of the PMESS is shown in Figure 2.

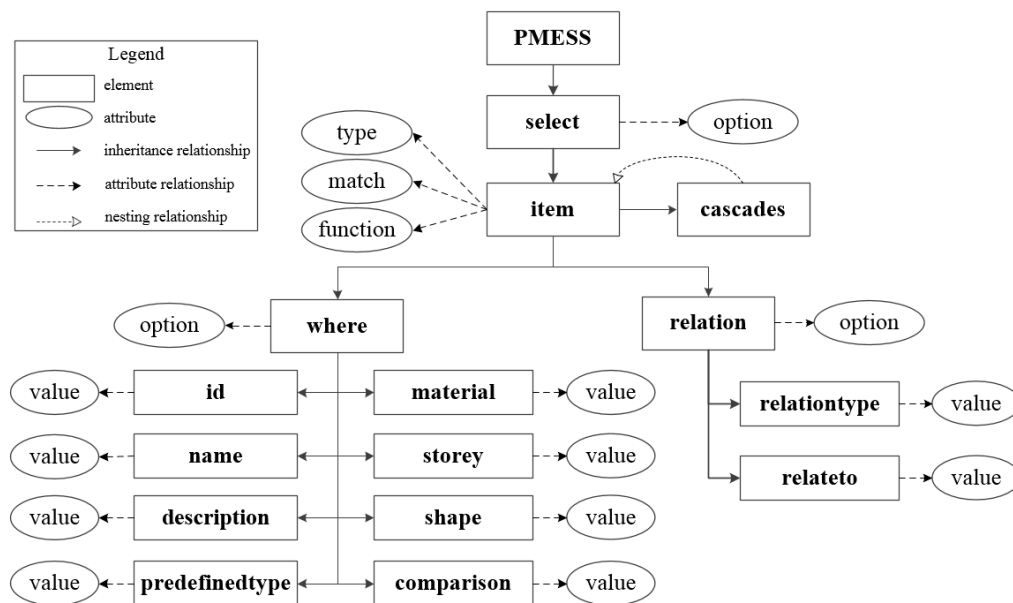


Figure 2. Architecture of the Partial Model Extraction based on the Selection Set (PMESS).

The 'PMESS' element is the root element at the first level of the overall architecture. The element is represented by the box symbol. The element at the second level is the 'select' element, which means the proposed method is to extract the partial model based on the Selection Set. The 'select' is a child element of the 'PMESS', which is connected by an arrow with a solid line. Condition 1 and 2 are mainly prescribed by elements 'item', 'relation', and 'where'. The 'item' element defines the entity type, including object entity and attribute entity, and the 'relation' for the relation entity. The 'item' element has three attributes: type, match, and function. The relationship between the element and the attribute is represented by the dotted arrow. The 'where' is used to represent the rules for extracting partial models. It is noted that the 'item' and the 'cascades' are connected by the hollow arrow with dotted line. This means that the structure of the 'cascades' is the same as the 'item'. More details will be discussed in the following section. An example of a PMESS-based configuration file used for extracting concrete columns is shown in Figure 3.

```

<?xml version="1.0" encoding="UTF-8"?>
<PMESS>
  <select>
    <item type="ELEMENT" match="column" function="extract">
      <where>
        <material value="concrete"/>
      </where>
    </item>
  </select>
</PMESS>

```

Figure 3. PMESS-based configuration file for extracting concrete columns.

5.1. 'Select'-Mechanism

The 'select' element has one unique attribute 'option', which is either 'AND' or 'OR'. The 'option' is designed to comply with the Theorem mentioned above. The use of 'AND' and 'OR' is defined as the intersection and union relations between several items, respectively. While the value of 'option' is 'AND', IFC instances will be extracted from the original model only when they match all the defined items. On the contrary, the 'OR' is required to match any one of the defined items. The default value of 'option' is 'OR'.

5.2. 'Item'-Classification

The type of entities in the Selection Set can be defined by the attribute 'type' of 'item'. Another two attributes of 'item' are 'match' and 'function'. The 'type' was designed to comply with Condition 1. Its value includes two types: ELEMENT and ATTRIBUTE, and it is required not to include abstract entity types (as mentioned in Condition 2). The 'match' enables users to describe the name of ELEMENT or ATTRIBUTE. The mapping between the value of 'match' and IFC entities/attributes has been established, which can automatically query IFC data according to the user-defined requirements.

When the 'type' is 'ELEMENT', the proposed method will extract IFC object entities which comply with the requirements defined in 'match' and 'where' (as mentioned in Rule 4). Particularly, if the value of 'match' is 'SET', the partial model extraction is required to comply with the rules defined in the 'relation'.

The matching attribute entities in the IFC model will be extracted, if the 'type' is 'ATTRIBUTE' (as mentioned in Rule 4). The proposed method queries and extracts object entities which have the matching attribute entities.

The third attribute of 'item' is 'function', which is currently limited to the 'extract' for partial model extraction. The 'filter', 'modify', and 'add' in the 'function' will be further studied in the next paper.

5.3. 'Relation'-Rule

According to representations of IFC relation entities, objects entities within a relationship can be divided into relating object entity and related object entity (entities), as shown in Figure 4a. In the IFC schema, there are many sub-entities within the *IfcRelationship* entity to represent diverse relations, such as *IfcRelContainedInSpatialStructure*, *IfcRelAggregates*, and *IfcRelAssignsToGroup*. Figure 4b illustrates an example of relating object entity and related object entities defined by the *IfcRelAssignsToGroup* entity.

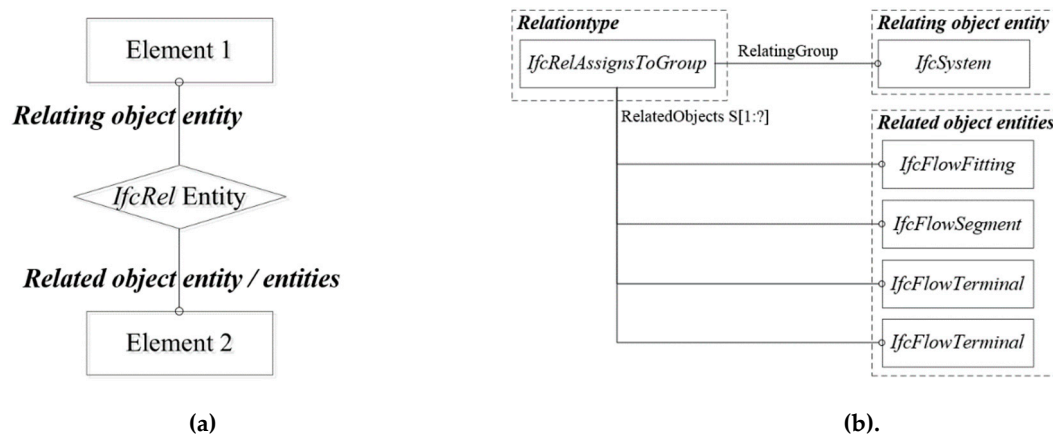


Figure 4. The relation between object entities: (a) Relation between relating object entity and related object entity/entities; (b) Relation using *IfcRelAssignsToGroup* entity.

The relating object entity and related object entity (entities) are defined in the sub-element 'relateto' of 'relation' (as shown in Figure 2), while the sub-element 'relationtype' is for the type of 'relation' (Rule 5). Currently, the proposed method supports the extraction of relations of building storey, group, and element assembly. As mentioned above, in the case of 'type=ELEMENT' and 'match=SET', the proposed method will query the matching relation entity according to the definition in 'relation', and extract IFC object entities referenced by the relation entity.

5.4. 'Where'-Rule

The elements mentioned above are mainly used to extract the objects with a certain type or relation, but not for the objects with some given characteristics. Hence, the 'where' element was designed to define rules for extracting specific objects according to the user-defined semantics (Rule 6). According to the characteristics of objects defined in the IFC schema, the object semantics could be classified as direct and indirect semantics. While direct semantics could be directly attained from IFC instances, indirect ones have to be inferred or computed from other IFC instances. The direct ones include the Identity Document (ID), name, description, and predefined type; and the material, storey, shape, and comparison for the indirect ones, as shown in Figure 2.

When these semantic meanings are defined in the PMESS document, the proposed method can query the target data to form a valid IFC model.

Figure 5 presents an example of all attributes in *IfcBeam* entity and the corresponding IFC entities for 'where' rules. The ID, name, and description are derived from the *IfcRoot* entity, a root entity in the IFC schema storing the most fundamental information. The predefined type, an extension of the IFC4 version to the attribute in the *IfcBeam* entity, is used to define different types of the object (*IfcBeam* in this example). The indirect semantics are required to query other IFC entities, for example, the material. In general, the *IfcMaterial* entity is associated with the IFC object entity through the *IfcRelAssociatesMaterial* entity, a subtype of the *IfcRelAssociates* entity.

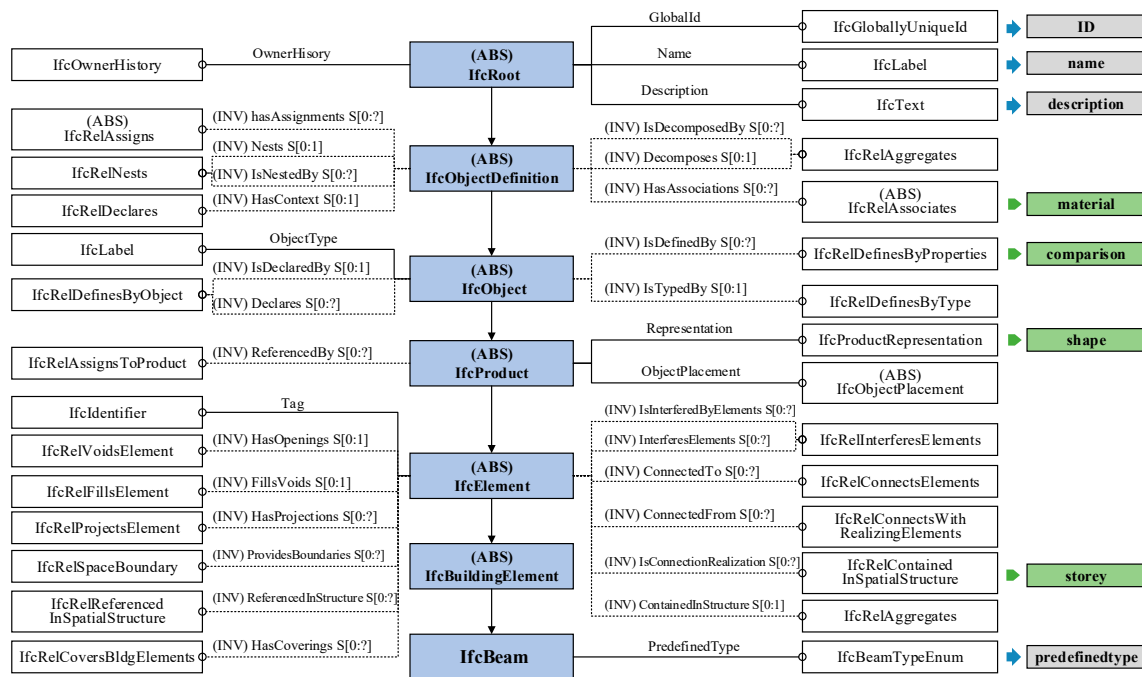


Figure 5. Mapping between *IfcBeam* entity's attributes and 'where' rules. ABS (Abstract): abstract entity of data types.

5.5. 'Cascades'-Rule

The definitions in 'item' are overall requirements for extracting the partial model, and the 'cascades' can be used to further prescribe the requirements. The structure of 'cascades' is the same as 'item' to ensure the uniform definition. Figure 6 shows an example of the PMESS-based configuration file for extracting beams under the 'where'-rule that the construction time is '2019-09-20'. In this case, the proposed method firstly queries all beams in the building project, and then queries the specified beams which match the 'cascades'.

```
<?xml version="1.0" encoding="UTF-8"?>
<PMESS>
  <select>
    <item type="ELEMENT" match="beam" function="extract">
      <cascades>
        <item type="ATTRIBUTE" match="property" function="extract">
          <where>
            <name value="construction time"/>
            <comparison value="2019-09-20"/>
          </where>
        </item>
      </cascades>
    </item>
  </select>
</PMESS>
```

Figure 6. PMESS-based configuration file for extracting beams (construction time is '2019-09-20').

The XML schema was adopted to define PMESS elements for complying with Rule 4–Rule 6. Rule 1–Rule 3 are the fundamental rules to form a valid IFC file, while Rule 7 is used to process redundant information within the extracted IFC instances. These four rules (Rule 1, 2, 3, and 7) have been embedded in the data process engine for implementation, not required to be defined by users.

6. Test Case

C++ programming language was used to develop two data interfaces for the implementation of the proposed method (PMESS). One is to read the PMESS-based configuration file, and the other is to extract and export the partial model. For further applications, these data interfaces were embedded into the proposed IFC-based platform. Different IFC models exported from many software tools (such as ArchiCAD, MagiCAD, Revit, and Tekla Structures) have been used to verify the feasibility of PMESS. In this section, a practical project model created by ArchiCAD was used to introduce the utility of the proposed method.

The test case was conducted using a building model of a shopping mall project. The building has eight floors with a construction area of 148,564 m², including six floors above ground and two underground floors. This model was built by ArchiCAD and exported as an IFC file by default settings. The file size of this IFC model was about 101 M, with 1,862,673 IFC instances. Figure 7 shows the visualization of this project in the proposed IFC-based platform. Due to the large IFC file size of this project, it is necessary to extract partial models for different business tasks. Through the proposed method, several partial models were extracted under the following extraction requirements.

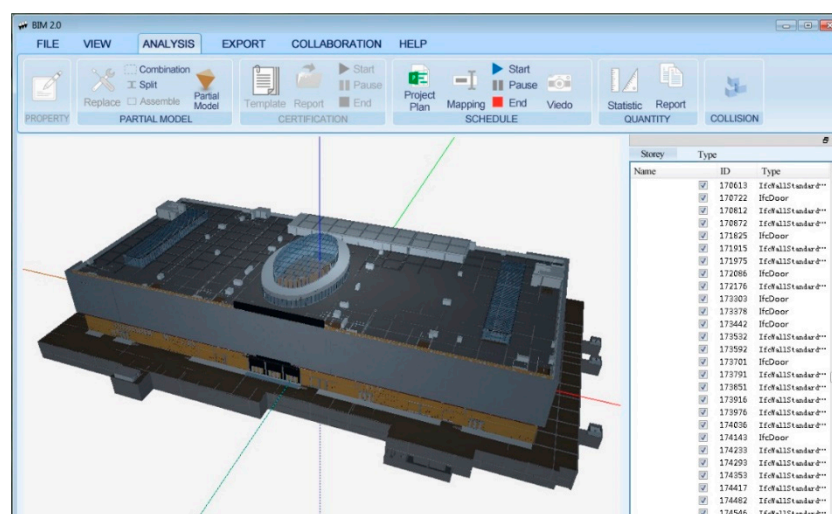
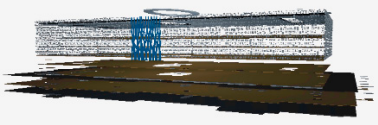


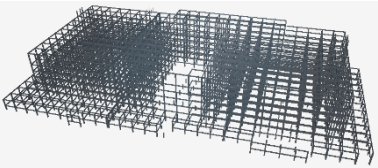


Figure 7. IFC model of the shopping mall project in the proposed IFC-based platform.

6.1. Partial Model Extraction for the User-Defined Extracted Objects

Through setting different 'item' elements, the required objects could be extracted. As examples for types of extracted objects, Table 1 shows four examples of partial models extracted from the original model. The first three partial models extract a certain type of object, while two types of objects are extracted in the fourth one. Figure 8 depicts the PMESS-based configuration file for the fourth partial model in the proposed platform. Through the PMESS, physical objects from different disciplines (architecture, structure, MEP, etc.) can be extracted from the original BIM model, such as the door and window for architecture, the beam and column for structure, the equipment and pipeline for MEP.

Table 1. Some examples of partial model extraction for the required objects by using PMESS.

Extracted Objects	Original Model		Amount	File Size/M	Partial Model Instances	Visualization
	IFC Entity	Amount				
Slab	<i>IfcSlab</i>	5284	5284	19.9 (19.7%)	292,912 (15.7%)	
Door	<i>IfcDoor</i>	911	911	19.2 (19.0%)	309,332 (16.6%)	
Transport element	<i>IfcTransport Element</i>	64	64	15.7 (15.5%)	253,984 (13.6%)	
Beam and column	<i>IfcBeam and IfcColumn</i>	6886 (4002&2884)	6886 (4002&2884)	13.8 (13.7%)	173,763 (9.3%)	

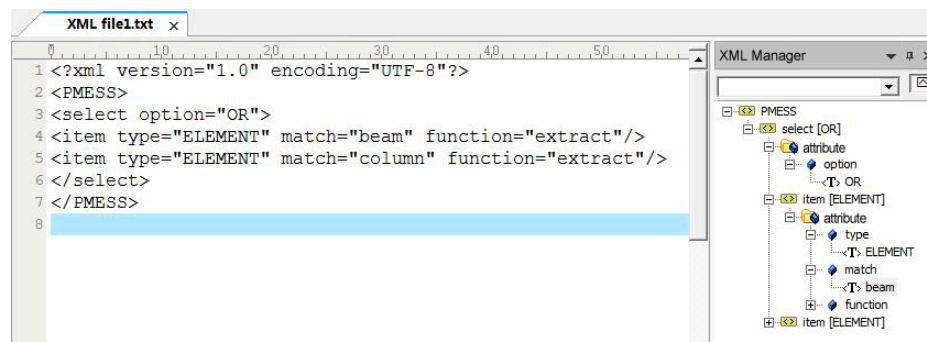


Figure 8. PMESS-based configuration file for extracting beams and columns.

By using the IFC File Analyzer [44], IFC model data can be analyzed in detail. As shown in Table 1, all IFC object entities which match the user-defined requirements were correctly extracted. Moreover, through filtering out other objects, the resulting models only contain the required objects and their attributes. For example, the number of IFC instances in the fourth partial model was 173,763, and 90.7% instances were filtered out. Accordingly, the file size decreased to 13.8 M (only 13.7% of the original model). The results show that the proposed method correctly identifies the user-defined requirements and extracts all the semantically required objects from the original model. On the other hand, the decreasing of the partial models in file size is apparent compared with the original model, which avoids filtering redundant information manually and facilitates the fulfillment of downstream business tasks based on useful building information.

Except the extraction according to the object type, other semantics could be used to extract the required BIM data, such as the relationship and the rules (as shown in the following subsections).

6.2. Partial Model Extraction Based on the User-Defined Relations

This project is composed of underground and overground structures, so it needs to be built by different designers. According to this requirement, the 'relation'-rule was used to extract all objects in the different parts of this building. An example of the PMESS-based configuration file for extracting the underground structure is illustrated in Figure 9, and the resulting partial model is presented in Figure 10. The file size of the extracted partial model is 32.1 M, including 942 columns, 1380 beams, 1253 walls, 351 doors, 69 slabs, and 148 stairs. These extracted objects are consistent with the original model. The proposed method is capable of querying and extracting the required information depending on the user-defined relation rule.

```

<?xml version="1.0" encoding="UTF-8"?>
<PMESS>
  <select>
    <item type="ELEMENT" match="SET" function="extract">
      <relation relationtype="building storey" relateto="related"/>
      <where>
        <name value="Floor B1"/>
      </where>
    </item>
    <item type="ELEMENT" match="SET" function="extract">
      <relation relationtype="building storey" relateto="related"/>
      <where>
        <name value="Floor B2"/>
      </where>
    </item>
  </select>
</PMESS>

```

Figure 9. PMESS-based configuration file for extracting building storeys named Floor B1 and B2.

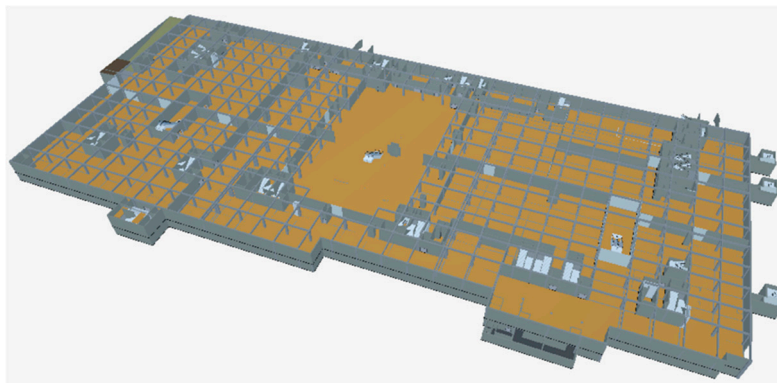


Figure 10. Partial model composed of the underground Floor B1 and B2.

6.3. Partial Model Extraction Based on the User-Defined Rules

Numerous curtain walls were contained in this building project, such as peripheral curtain walls, and the skylight on the sixth floor. Due to the complexity of curtain walls, the models of curtain walls were required to set as separate models, which would be designed by different curtain wall engineers. For this purpose, curtain walls in different placements were extracted from the complete architectural model, and imported back into the original software for further design and analysis, as shown in Figure 11. The main contents of the PMESS-based configuration files were presented in the middle part of Figure 11.

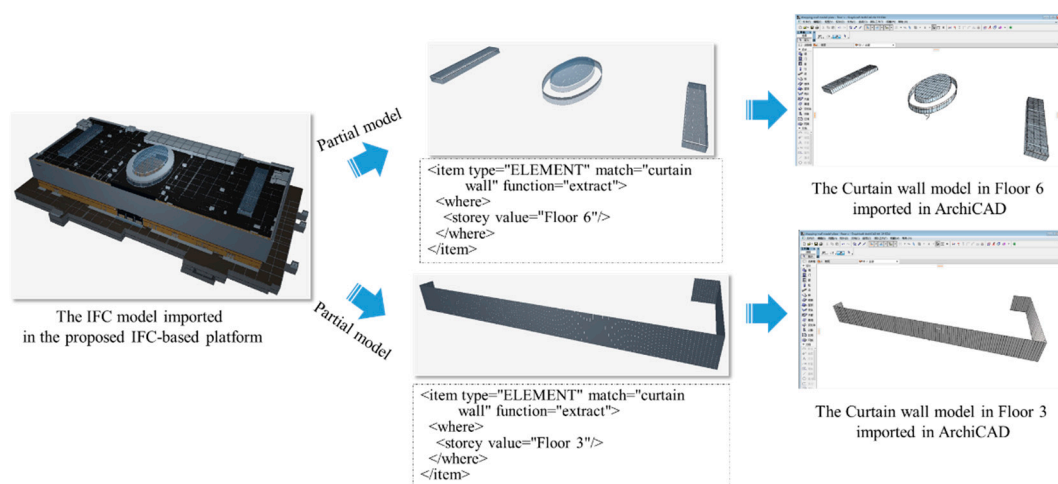


Figure 11. Partial models of curtain walls in different placements.

The file sizes of these two partial models were 2.37 M and 0.59 M, respectively, which were much smaller than the original model's (101 M). It is beneficial for engineers to make a detailed design based on the reduced models. These extracted partial models could be imported back to the original software ArchiCAD for detailed design. In addition, these extracted partial models were IFC compliant models, and could be used to import to other BIM software tools (such as Revit and Tekla) for professional design. It demonstrated that the extracted IFC files were syntactically valid.

7. Conclusions

A building project always consists of different types of information from multiple disciplines. However, business tasks require only a part of the complete building information model. Meanwhile, the required information varies depending on business tasks. A common method for partial model extraction which meets user-defined extraction requirements is necessary. For this purpose, a generic language for partial model extraction based on the Selection Set was proposed to extract a partial model from the IFC model.

The Selection Set was designed to represent extraction requirements. Elements in the Selection Set work as input parameters of the partial model extraction method. Due to the complexity of requirements for business tasks, several extraction requirements could be defined in the intersection or union form.

Furthermore, seven rules were defined to extract the partial model based on mathematical logic and set theory. These proposed rules ensure the syntactical and semantic validity of the partial IFC model during the extraction process. Firstly, the proposed method queries IFC data which matches the requirements defined in the Selection Set, such as IFC entities, properties, and relations. Subsequently, according to seven rules for partial model extraction, these extracted IFC data are defined as the nodes to query other related IFC data, and redundant information is filtered for forming a valid partial model.

Considering the processability of the computer and the readability of users, the XML schema was adopted to design the generic language for partial model extraction. Given the definitions of building information in the IFC schema, this study developed a mapping between IFC data and the elements defined in the PMESS method, which could meet diverse requirements defined by users. Through the PMESS method, users can extract the required information from the original model under different extraction requirements, such as objects, properties, and relations. In addition, the PMESS-based configuration file can be saved as a common template for reuse, which improves the efficiency of the definitions of extraction requirements.

To demonstrate the feasibility of the proposed method, a practical project was used to extract different partial models under three conditions: object types, object relations, and specific rules.

Compared with the original model, the required objects were correctly extracted, which showed the validity of partial models at the semantic level. Furthermore, the extracted partial models could be imported back into the original software tool, which demonstrated the syntactical validity of the extracted IFC file.

Currently, although some commercial software products can be used to extract the required objects, it needs users to manually select the required objects, and the partial model cannot be extracted according to the particular rules. Some researchers have carried out research projects for partial model extraction, and mainly focus on some specific information, such as geometric information. In this study, the proposed PMESS method makes users automatically extract the partial model by requirement definition. Furthermore, users can define different requirements to extract the required partial model based on the PMESS, which can accommodate more applications.

This study is an important step in data sharing and exchange of building projects, and it also has room for improvement. For example, BSI proposed IDM and MVD for exchange requirements, and defined several templates for practical tasks. To extend the applicability of the proposed method, the PMESS should be mapped to IDM and MVD. Given the fact that the PMESS was designed based on the XML schema, the mapping mechanism between the PMESS and mvdXML could be further studied.

Author Contributions: Conceptualization, X.D. and H.L.; Methodology, X.D. and H.L.; Validation, H.L., J.X., and Y.Z.; Writing—original draft preparation, X.D. and H.L.; Writing—review and editing, J.X., and Y.Z.; Visualization, H.L.; Supervision, H.L. and Y.Z.; Project administration, X.D. and J.X.; Funding acquisition, X.D. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research work was supported by the National Key Research and Development Program of China during the 13th Five-year Plan (no. 2016YFC0702001), the Industrial Internet Innovation and Development Project 2019 from the Ministry of Industry and Information Technology of China, and Project Funded by China Postdoctoral Science Foundation (no. 2019M663115).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sanguinetti, P.; Abdelmohsen, S.; Lee, J.M.; Lee, J.K.; Sheward, H.; Eastman, C. General system architecture for BIM: An integrated approach for design and analysis. *Adv. Eng. Inform.* **2012**, *26*, 317–333. [\[CrossRef\]](#)
2. Olawumi, T.; Chan, D. Building information modeling and project information management framework for construction projects. *J. Civ. Eng. Manag.* **2019**, *25*, 53–75. [\[CrossRef\]](#)
3. Juan, Y.K.; Lai, W.Y.; Shih, S.G. Building information modeling acceptance and readiness assessment in Taiwanese architectural firms. *J. Comput. Civ. Eng.* **2016**, *23*, 356–367. [\[CrossRef\]](#)
4. Oh, M.; Lee, J.; Hong, S.W.; Jeong, Y. Integrated system for BIM-based collaborative design. *Automat. Constr.* **2015**, *58*, 196–206. [\[CrossRef\]](#)
5. Belsky, M.; Sacks, R.; Brilakis, I. Semantic Enrichment for Building Information Modeling. *Comput.-Aided Civ. Infrastruct. Eng.* **2016**, *31*, 261–274. [\[CrossRef\]](#)
6. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd ed.; Wiley: New York, NY, USA, 2011.
7. Wilson, P.R. EXPRESS and Set Theory (Draft). Available online: <http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.6106&rank=1> (accessed on 3 August 2018).
8. Katranuschkov, P.; Weise, M.; Windisch, R.; Fuchs, S.; Scherer, R.J. BIM-based generation of multi-model views. In Proceedings of the CIB W78 2010: 27th International Conference—Applications of IT in the AEC Industry, Cairo, Egypt, 16–18 November 2010.
9. Mazairac, W.; Beetz, J. BIMQL—An open query language for building information models. *Adv. Eng. Inform.* **2013**, *27*, 444–456. [\[CrossRef\]](#)
10. Graphisoft. Filter Model at Export. Available online: <https://helpcenter.graphisoft.com/user-guide/65795/> (accessed on 18 February 2020).
11. Autodesk. IFC Export Setup Options. Available online: <http://help.autodesk.com/view/RVT/2020/ENU/?guid=GUID-E029E3AD-1639-4446-A935-C9796BC34C95> (accessed on 18 February 2020).

12. Zhang, C.; Beetz, J.; Vries, B.D. Towards model view definition on semantic level: A state of the art review. In Proceedings of the 20th International Workshop of the European Group for Intelligent Computing in Engineering, Vienna, Austria, 1–3 July 2013.
13. BuildingSMART International. Model View Definition Summary. Available online: <http://www.buildingsmart-tech.org/specifications/ifc-view-definition> (accessed on 12 January 2019).
14. Lee, G.; Sacks, R.; Eastman, C. Product data modeling using GTPPM—A case study. *Automat. Constr.* **2007**, *16*, 392–407. [\[CrossRef\]](#)
15. Lee, G.; Park, Y.H.; Ham, S. Extended process to product modeling (xPPM) for integrated and seamless IDM and MVD development. *Adv. Eng. Inform.* **2013**, *27*, 636–651. [\[CrossRef\]](#)
16. BuildingSMART International. IfcDoc Tool Summary. Available online: <http://www.buildingsmart-tech.org/specifications/specification-tools/ifcdoc-tool> (accessed on 10 February 2019).
17. Scherer, R.J.; Weise, M.; Katranuschkov, P. Adaptable views supporting long transactions in concurrent engineering. In Proceedings of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering, Montreal, QC, Canada, 14–16 June 2006; pp. 3677–3686.
18. Nepal, M.P.; Staub-French, S.; Pottinger, R.; Webster, A. Querying a building information model for construction-specific spatial information. *Adv. Eng. Inform.* **2012**, *26*, 904–923. [\[CrossRef\]](#)
19. Teo, T.A.; Yu, S.C. The extraction of indoor building information from BIM to OGC IndoorGml. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, 167–170. [\[CrossRef\]](#)
20. Zhou, Y.; Hu, Z.Z.; Zhang, W.Z. Development and application of an Industry Foundation Classes-based Metro protection information model. *Math. Probl. Eng.* **2018**, 1820631. [\[CrossRef\]](#)
21. Qin, L.; Deng, X.Y.; Liu, X.L. Industry foundation classes based integration of architectural design and structural analysis. *J. Shanghai Jiaotong Univ. (Sci.)* **2011**, *16*, 83–90. [\[CrossRef\]](#)
22. Hu, Z.Z.; Zhang, X.Y.; Wang, H.W.; Kassem, M. Improving interoperability between architectural and structural design models: An industry foundation classes-based approach with web-based tools. *Automat. Constr.* **2016**, *66*, 29–42. [\[CrossRef\]](#)
23. Kim, H.; Shen, Z.; Kim, I.; Kim, K.; Stumpf, A.; Yu, J. BIM IFC information mapping to building energy analysis (BEA) model with manually extended material information. *Automat. Constr.* **2016**, *68*, 183–193. [\[CrossRef\]](#)
24. Kim, H.; Anderson, K.; Lee, S.; Hildreth, J. Generating construction schedules through automatic data extraction using open BIM (building information modeling) technology. *Automat. Constr.* **2013**, *35*, 285–295. [\[CrossRef\]](#)
25. Adachi, Y. Overview of Partial Model Query Language. Available online: <http://cic.vtt.fi/projects/ifcsvr/> (accessed on 3 December 2017).
26. Weise, M.; Katranuschkov, P.; Scherer, R.J. Generalised Model Subset Definition Schema. In Proceedings of the 20th CIB W78 International Conference on Information Technology in Construction, Waiheke Island, Auckland, New Zealand, 23–25 April 2003; pp. 440–447.
27. Chipman, T.; Liebich, T.; Weise, M. mvdXML Specification 1.1. Available online: https://standards.buildingsmart.org/MVD/RELEASE/mvdXML/v1-1/mvdXML_V1-1-Final.pdf (accessed on 12 March 2019).
28. Yang, D.; Eastman, C. A rule-based subset generation method for product data models. *Comput.-Aided Civ. Infrastruct. Eng.* **2007**, *22*, 133–148. [\[CrossRef\]](#)
29. Lee, G. Concept-based method for extracting valid subsets from an EXPRESS schema. *J. Comput. Civ. Eng.* **2009**, *23*, 128–135. [\[CrossRef\]](#)
30. Yu, F.Q.; Zhang, J.P.; Liu, Q. Semi-automatic generation of the BIM model view definition based on the IFC standard. *J. Tsinghua Univ. (Sci. Technol.)* **2014**, *54*, 987–992. [\[CrossRef\]](#)
31. Daum, S.; Borrmann, A. Boundary representation-based implementation of spatial BIM queries. In Proceedings of the EG-ICE Workshop on Intelligent Computing in Engineering, Vienna, Austria, 1–3 July 2013; pp. 1–10.
32. Daum, S.; Borrmann, A. Processing of topological BIM queries using boundary representation based methods. *Adv. Eng. Inform.* **2014**, *28*, 272–286. [\[CrossRef\]](#)
33. Fuchs, S.; Scherer, R.J. MMQL—A language for multi-model linking and filtering. In Proceedings of the 10th European Conference on Product and Process Modeling, Vienna, Austria, 17–19 September 2014; pp. 273–280.
34. Zhang, J.; El-Gohary, N.M. Automated information transformation for automated regulatory compliance checking in construction. *J. Comput. Civ. Eng.* **2015**, *29*, B4015001. [\[CrossRef\]](#)

35. Zhang, J.; El-Gohary, N.M. Automated extraction of information from building information models into a semantic logic-based representation. In Proceedings of the 2015 International Workshop on Computing in Civil Engineering, Austin, TX, USA, 21–23 June 2015; pp. 173–180.
36. Pauwels, P.; Terkaj, W. EXPRESS to OWL for construction industry: Towards a recommendable and usable IfcOWL ontology. *Automat. Constr.* **2016**, *63*, 100–133. [\[CrossRef\]](#)
37. Windisch, R.; Katranuschkov, P.; Scherer, R.J. A generic filter framework for consistent generation of BIM-based model views. In Proceedings of the 19th International Workshop of the European Group for Intelligent Computing in Engineering, Munich, Germany, 4–6 July 2012.
38. Won, J.; Lee, G.; Cho, C. No-schema algorithm for extracting a partial model from an IFC instance model. *J. Comput. Civ. Eng.* **2013**, *27*, 585–592. [\[CrossRef\]](#)
39. IfcOpenShell. Available online: <http://www.ifcopenshell.org/> (accessed on 20 February 2020).
40. XBIM. Available online: <https://github.com/xBimTeam/XbimWebUI> (accessed on 20 February 2020).
41. IfcPlusPlus. Available online: <https://github.com/ifcquery/ifcplusplus> (accessed on 20 February 2020).
42. Liebich, T. IFC 2x Edition 3 Model Implementation Guide. Available online: https://standards.buildingsmart.org/documents/Implementation/IFC2x_Model_Implementation_Guide_V2-0b.pdf (accessed on 15 January 2018).
43. Lee, Y.C.; Eastman, C.M.; Solihin, W. An ontology-based approach for developing data exchange requirements and model views of building information modeling. *Adv. Eng. Inform.* **2016**, *30*, 354–367. [\[CrossRef\]](#)
44. National Institute of Standards and Technology. IFC File Analyzer. Available online: <https://www.nist.gov/services-resources/software/ifc-file-analyzer> (accessed on 8 March 2019).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).