Journal of Computing in Civil Engineering

IFC-based Partial Data Model Retrieval for Distributed Collaborative Design

--Manuscript Draft--

Manuscript Number:	CPENG-2508R2
Full Title:	IFC-based Partial Data Model Retrieval for Distributed Collaborative Design
Article Type:	Technical Paper
Abstract:	<p>To deal with the growing complexity of IFC models, partial model has always been proposed as an effective way to satisfy users' specific requirements by tailoring the central models. The traditional approaches with IFC-validated partial models have the problem of the high structural complexity and the low data density. This paper proposes an IFC-property based model generation method to support domain-specific data extraction with a property-based model view definition (MVD). This approach removes data-irrelevant IFC definitions while keeping data links between the view and the IFC central model. Based on the MVD, an algorithm is designed to extract the required properties from the central IFC model by recursively iterating through IFC data instances for two types of properties. The proposed approach has been implemented on the top of the open source BIMserver platform. Its feasibility and efficiency have been validated through a set of practical examples. Compared to central IFC model and IFC-validated partial model approach, the experiment results show that our proposed approach achieves the smallest model size, the fastest speed for the model loading, the least memory usage and the lowest network transmission overhead.</p>

IFC-based Partial Data Model Retrieval for Distributed Collaborative Design

Ning Gui¹, Chuhan Wang², Zhifeng Qiu³, Weihua Gui⁴, Geert Deconinck⁵

¹ Associated Prof., School of Software, Central South University, China, ninggui@gmail.com

² Post-graduate student, College of Information, Zhejiang Sci-Tech. University, Hangzhou, China, 404833948@qq.com

³ Associated Prof., College of Information Science and Technology, Central South University, China, zhifeng.qiu@csu.edu.cn

⁴ Professor, College of Information Science and Technology, Central South University, China, guiweihua@csu.edu.cn

⁵ Professor, LECTA/ESAT, KU Leuven, Heverlee, Belgium, geert.deconinck@esat.kuleuven.be

Abstract: To deal with the growing complexity of IFC models, partial model has always been proposed as an effective way to satisfy users' specific requirements by tailoring central model. The traditional approaches with IFC-validated partial models have the problems of the high structural complexity and the low data density. This paper proposes an IFC-property based model generation method to support domain-specific data extraction with a property-based model view definition (MVD). This approach removes data-irrelevant IFC definitions while keep data links between the view and the IFC central model. Based on the MVD, an algorithm is designed to extract the required properties from the central IFC model by recursively iterating through IFC data instances for two types of properties. The proposed approach has been implemented on the top of the open source BIMserver platform. Its feasibility and efficiency have been validated through a set of practical examples. Compared to central IFC model and IFC-validated partial model approach, the experiment results show that our proposed approach achieves the smallest model size, the fastest speed for model loading, the least memory usage and the lowest network transmission overhead.

Key words: building information modeling (BIM); industry foundation classes (IFC) standard; model view definition (MVD); collaborative design; partial data model.

Introduction

The Building Information Model (BIM) aims at constructing a shared information foundation by digitalizing all related information of building projects (Eastman, Eastman et al. 2011). By accurately modeling different building-related elements, it is possible to simulate the building performance from different perspectives. Industry Foundation Classes (IFC) – an international standard for exchanging BIM models among different stakeholders (Schein 2007) – has evolved to IFC 4 with much increased domain supports, e.g., for structural steel and timber, GIS, 5D support (Liebich 2013). Other researchers, also enrich IFC schema with supports for energy simulation (Yan, Clayton et al. 2013, Gupta, Cemesova et al. 2014, Cemesova, Hopfe et al. 2015, Kim, Shen et al. 2016), virtual construction (Zhang, Yu et al. 2014), asset management (Motamedi, Soltani et al. 2016) and infrastructure domains (Lee and Kim 2011). Thus, BIMs have been exerted to the whole concept-to-occupation time-span and consist of data and resources in the different stages of building's lifecycle. Consequently, BIMs inevitably have been used in growing scale in practice, with refined granularity and by more and more stakeholders with both geometry and domain diversities. In order to allow collaborations between different stakeholders, a set of BIM platforms has also been developed: open source approaches like IFCServer (Chen, Cui et al. 2005), BIMServer (Beetz, van Berlo et al. 2010) and commercial platforms such as Autodesk Revit® Cloud (Zhang and Issa 2012). Those platforms normally provide a management structure with a uniform central BIM database. In this structure, different stakeholders make their collaborative revisions on a central BIM model. With the growing volume of data size and the growing number of stakeholders, a typical BIM model, together with its related documents, normally is at the multitude of GB size. Thus, this central collaboration structure will become very inefficient as one stakeholder might be overwhelmed by the volume of data since most of them is unrelated to his/her interests. To avoid the deluge of information and facilitate the distributed collaborative design, a distributed platform is proposed as shown in Figure 1. In this platform, each stakeholder only works on

his/her specific partial models rather than a central BIM model. Those partial models are dynamically retrieved when necessary. A distributed BIM-based collaborative design platform involves work for many different parts, e.g., partial model retrieval, distributed version control and the model synchronization.

The focus of this paper is on the partial data retrieval and representation. Three different major design concerns are identified:

- Data consistence: As the revisions made by different designers will be ultimately integrated into the central model, it is important to keep the multi-direction map between the partial models and the central model in order to facilitate both partial model retrieval and revision integration;
- Representation capabilities: For different designers, the partial model should be able to represent data with geometry, e.g., for architects, as well as data without geometry like COBie ([East 2007](#)), e.g., for financial officers;
- Efficiency: In this collaborative design environment, respective stakeholder normally need to frequently retrieve and update their domain-specific partial models. Thus, the partial model distilling should be very efficient.

Partially due to the facts that IFC is the standard format for the BIM data exchange and it is open and extensible, most existing researches on the partial BIM model support are based on IFC. The organization IAI proposed the Information Delivery Manual (IDM) ([Wix and Karlshoej 2010](#)) to define what information is required for specific exchanges in the different life-cycle stages and between different stakeholders. The introduction of the Model View Definition (MVD) provides a filter to extract the required information from a large IFC model which allows for automated partial model processing and extraction ([Boukamp and Akinci 2007](#)). The buildingSMART alliance has been working in defining standard MVDs for typical information exchange tasks, e.g., the Coordination View 2.0 have been defined. Chen et al. ([2005](#)) developed a filter that can extract a structural partial model only for structural engineers. To simplify the data management of IFC model, [Solihin, Eastman et al. \(2016\)](#) proposed a framework in which an IFC model is constructed with federated partial models with custom-defined IFC extensions. However, those solutions are normally limited to pre-defined domains. In order to mitigate this problem, researchers have been working on the flexible definition of MVDs. There exist many partial instance models according to the defined MVD. [Weise, Katranuschkov et al. \(2003\)](#) developed a model subset definition schema as a meta-schema for defining MVDs. [Yang and Eastman \(2007\)](#) summarized the rules for MVD generation based on the IFC schema and proposed a rule-based MVD definition method for the fast MVD generation. In order to simplify the definition of MVD, the natural-language-based data view generation mechanisms are introduced for facilitating view generation processes. Instead of defining MVD based on IFC schema, another stream of researches focuses on the extraction of partial models directly from the IFC instance model. [Zhang \(2009\)](#) introduced an IFC-based partial model generation method which supports the IFC instance-based extraction. [Won, Lee et al. \(2013\)](#) proposed a no-schema algorithm for efficient partial model extraction of an IFC instance model without the support of IFC schema. In all those approaches, the generated partial models syntactically and semantically satisfy IFC-specification (denoted as the *IFC-validated* partial model in this paper). However, in order to be IFC-validated, many IFC-entities and types that might not be directly related to the required data are also included in the partial model, no matter whether they are of the stakeholders' interest. The resulting IFC-validated partial model is of low value density towards its users and requires additional parsing time. Thus, it is very hard for the resulting partial model to be effectively used in the collaboration contexts featured with high frequency of data retrieval and modification. The introductions of COBie, by providing the transformation between IFC model, spreadsheet and other formats, to some extent, alleviate this problem as it removes many IFC schema related information. However, its major targeted domains are for the facility management and lack the supports for geometric data, which is important for many other stakeholders, e.g., architects and civil engineers. In summary, the existing

approaches for partial model generation mainly provide supports for the data consistency, while a few provide supports for efficiency, e.g., COBie. COBie focuses on delivering asset data by introducing a Microsoft Excel-based data format. It is light weighted and achieves the parsing time. However, no existing approach provides partial model generation meeting all three requirements, i.e. consistence, representation capabilities and efficiency, at the same time, which are all necessary for the effective distributed collaborative designs. The traditional IFC schema based MVD is designed for one-time data delivery scenes which demands one IFC file be self-explainable. Thus, there are many structural related redundancies. However, in the collaborative design environments, the central model exists and can be referred at any time.

The major contributions of this paper are to design an IFC property-based partial model extraction method that can generate partial model with the support for all three requirements. An IFC property-based partial model view is defined to keep properties in demand (for consistency) while removes IFC schema related information (for efficiency). Then, based on the partial model view definition, a partial model generation algorithm is proposed for the retrieval of the properties related to the required objects. The proposed partial model view supports data representation for both geometric data and non-geometric data. This approach is realized on the top of the open source IFC platform – BIMserver – and tested with three open-shared IFC projects. Test results show that, for the same IFC model, our approach has significant advantages in terms of data size, processing time and network transmission speeds, over both IFC central model and IFC-validated partial model for all studied projects with slightly increased parsing time.

Problem Analysis

In the IFC2x3 and later specifications, there exist two major different types of entities: *independent exchangeable entities* (IEE, inherited from *IfcRoot* class) and *no-independent exchangeable entities* (not inherited from *IfcRoot*, often denoted as resource entities). Each IEE has a global unique id (GUID) and history information. IEE can independently exist and be exchanged. The resource entities, in contrast, cannot independently exist and are only referenced by the IEE.

The IEE includes five major types: *IfcObject*, *IfcRelationship*, *IfcPropertySetDefinition*, *IfcProject* and *IfcTypeObject* as well as their child classes (in order to simplify our discussion, in the following sections, without ambiguity, those types are used to indicate the objects of this class and its child classes). An *IfcObject* is the generalization of any semantically treated thing or process. The *IfcObjects* are the major information carrier of an IFC model, constitute the major parts of an IFC model. Thus, they are often denoted as the *major entities*. Examples of *major entities* include physically tangible items, e.g., wall, beam and conceptual items, for instance, grids or virtual boundaries. It also stands for processes for control and actors for involved persons. *IfcRelationship* is used to describe the reference relationship between two or more entities. There are two different types of relationships, 1-to-1 relationship and 1-to-many relationship. The *IfcPropertySetDefinition* is a generalization of property sets. They are either: dynamically extendable properties or statically defined properties. The *IfcProject* describes the project basic information and the *IfcTypeObject* describes the types of an entity.

Figure 2 shows the description of an *IFCColumn* object and its attributes and properties. This IFC description includes the *IFCCOLUMN* object definition, geometry description and material properties. The existing partial model extraction approaches normally are based on the extraction of IFC selected entities. In order to guarantee the validity of generated partial model, all selected objects and their associated information, e.g., relations and property sets, should be included in the model. However, for a particular stakeholder, for instance, in the asset management domain, information of major interest is the object name, original/current value and owner rather than its geometric data and the relation entities. The indiscriminate incorporation of all the object information, especially for the relationship entities, would lead to the problem of significant data redundancy and performance overhead. Furthermore, the IFC-

validated partial model has a complex internal structure, resulting in low querying performance. For instance, in order to get the value of the “unit_cost” property, the program needs to firstly get the *IfcPropertySet* through the *IfcRelDefinesByProperties*. Then, this *IfcPropertySet* has to be traversed to locate the *IfcPropertySingleValue* with this property name. For each property, several steps might be needed to retrieve its value.

In order to support the high efficient model usage, the extracted partial model, it is argued that, should focus only on the data of interest with comparably simple data structure. To increase the information value density and reduce the model complexity, the IFC-specification related description should be eliminated as much as possible. In the meantime, as IFC is the international standard for the BIM data exchange, this partial model should contain enough information regarding the IFC structure. When necessary, this partial model can be converted back into the IFC -validated partial model with the help of the IFC central model (here, we denoted the feature as the IFC-traceability). Under these two design principles, this paper proposes a partial model view which only contains the properties of selected major objects.

Property-based Partial MVD

In order to effectively extract the domain-specific partial model from the IFC global model, the definition of the partial model view must be constructed to reflect the abstraction requirements. In this **partial model view**, data requirements required by different stakeholders are represented by the list of selected properties of selected IFC elements. As discussed in the previous sections, the view is defined with the above-mentioned two principles: high data density for the domain-specific values and enough data for the IFC-traceability.

Based on those two principles, a property-based IFC partial model view is defined. As shown in the Figure 3, six major elements are included:

Model name, which is used for the unique reference of a partial model. 2) Model description, this element provides the further descriptions for the partial model. 3) Domain classification, which is used to indicate the targeted domain of the view definition. This information can help view users to locate their interested views. 4) The GlobalID of the IFC central model, this unique id keeps track of the IFC central model from which the partial model is generated. 5) The set of IfcObjects GlobalID, these ids are used to indicate which IFC entities are involved in the information exchange. 6) The set of properties involved with data exchanges. As shown in Figure 3, there are five major types of properties: graphic description, geometric description, material description, display property and property sets. The first three types of properties belong to the geometric aspect, through the relationship of entity association, describes the entity properties from the 3D perspective.

In practice, the involved IfcObject entities and their involved properties can be determined several different methods: list-like approaches, e.g., IDM for specific use cases, SQL-like query languages ([Mazairac and Beetz 2013](#)), topological query language ([Daum and Borrmann 2014](#)) and natural-language based query ([Lin, Hu et al. 2016](#)). The properties to be exchanged are from the major types of properties. Normally, the geometric and spatial property sets are mainly used for building elements geometric description. Those data are bundled together as one geometric property to simplify property abstraction. From the definition of partial model view, only properties to be exchanged are included, which is a fraction of all properties. For the column object illustrated in Figure 2, when the 3D display is needed, only the graphic properties are included in the MVD. While for the asset management domain, only asset-management related properties are included. This is against most entity-based abstraction methods in which most properties and relation entities are abstracted into the partial model to keep the validity of the generated IFC partial model. In the meantime, in order to support the second design principle, this view definition keeps the IFC central model information (GUID) and each object’s GlobalID. Thus, when necessary, this information can be used to regenerate the IFC-validated partial model.

It is worth to note that the term of MVD is originally used to define a subset of the IFC schema. However, COBie

172 is also an IFC MVD with the name “Basic FM Handover View” in which this term is used to indicate the subset of the
173 IFC data model, rather than the schema (COBie adopts a spreadsheet based data-collection/ delivery format). Thus,
174 the MVD used in in paper mainly follows the later scene.

175 **Partial Model Generation**

176 In this section, the IFC property description mechanisms are firstly analyzed in order to generate the property-based
177 partial model. A supporting partial model generation algorithm is then introduced according to different description
178 methods.

179 **Property Description Mechanisms**

180 In the IFC specification, as *IfcObjects* represents the physical building elements, they are the core entities used for
181 partial model extraction. For respective stakeholders, they only are interested in domain-specific properties for the
182 selected *IfcObject* entities. Thus, the *IfcObject* entities are the starting points of the extraction process for the property
183 extraction. From the access perspective, the properties of an *IfcObject* can be further classified into two major types:
184 attributes and properties(Liebich 2013). The attributes are directly stored in the instance of *IfcObject* class (subclass)
185 and is not presented by the relationship entities. Examples include the GlobalID, name, geometric and location
186 properties. Those attributes can be directly retrieved from the *IfcObject* instances. In contrast, properties are the
187 properties related to an *IfcObject* entity through relation entities and cannot be directly retrieved from the *IfcObject*.
188 Examples include material sets, property sets. The *IfcObject* entities use the *IfcRelDefines* class to express their own
189 properties by referring to certain resource entities. The referred entities can be located by the querying the reverse
190 properties. As shown in the Figure 4, the relation entity of *IfcRelDefines* can be further classified into two major types:

191 1) Associations through relationship entities: In this type of association, an *IfcObject* associates with *IfcPropertySet*
192 through the *IfcRelDefinedByProperties* entities. This relation entity stores a dual-reference between the *IfcPropertySet*
193 and the *IfcObject*. Several *IfcObject* entities can use one relation entity to build associations towards several
194 *IfcPropertySet*. Each *IfcPropertySet* entity might contain multiple properties.

195 2) Associations through type entities: Similar to the relationship between class and objects, one *IfcObject* entity is
196 normally also associated with a type entity, which is defined with *IfcObjectType*. The *IfcObjectType* can also define a
197 set of properties with the *IfcRelDefinesByType* relation entities. For objects of the same type, they share the same
198 properties defined in the *IfcObjectType*.

199 In order to generate the property-based model, it is important to acquire all selected properties from all three types
200 of properties definitions.

201 **Partial Model Generation**

202 Based on the definition of partial model view and the IFC property description mechanisms, a partial model
203 generating algorithm is proposed to achieve both the attribute and property abstraction. It traverses through the IFC
204 central model for the selected properties. The entities to be checked are *IfcObjects*, *IfcRelDefinedByProperties* relation
205 entities and *IfcObjectType* entities. The attribute/property abstraction algorithm is illustrated as Figure 5. The attribute
206 extraction process including nine major steps and the textbox with dotted lines are additional steps demands by the
207 properties extraction process:

208 **Extraction of attributes**

209 As discussed in the previous section, the attributes can be retrieved by starting from the *IfcObject* entities. If an
210 entity visited is within the partial model view definition, all its attributes are traversed for the matching ones. If any
211 match were found, this attribute would be extracted into the partial model.

212 Step 1: According to the domain-specific requirements described in the partial model view, determine the types
 213 of required entities and their required attributes and properties;
 214 Step 2 and Step 3: Initializes the data set and traverses the project entity and other major entities. The data set
 215 stores all extracted data records with format as GlobalID –{attribute-name, value; attribute-value, value...}. Here the
 216 GlobalID indicates the unique id of required major entity;
 217 Step 4: If the entity with the given GlobalID is found, traverses through its attribute and initializes an attribute set
 218 for the temporal data storage;
 219 Step 5: Read one attribute, if it is the type of attribute that is needed in the model view, go to step 6; otherwise,
 220 read next attribute;
 221 Step 6: Check the type of the attribute, if it is reference, go to step 5, read the next attribute of this reference;
 222 recursively calling the attribute extraction algorithms until the retrieved attribute is non-reference type;
 223 Step 7: Retrieve the name and value of this attribute;
 224 Step 8: If this attribute is needed, store the attribute into the attribute set; if the object is firstly visited, store its
 225 GlobalID as the key for the attribute set;
 226 Step 9: If all the required attributes are found, return the attribute set result to the model, the attribute set is indexed
 227 with the global id of entities. Then, i++, to find another required major entity.
 228 When all the entities are found, returns the partial model with attributes.

229 **Extraction of properties**

230 For the properties defined by the relationship entities, their extraction process slightly differs from the approach for
 231 attributes. For properties, the starting points of retrieval are relation entities, as shown in the Figure 5 by the textbox
 232 with dotted line. For each relation entity, its associated major entities are queried by the definition of association or its
 233 reverse property. When its associated major entities matching the keys defined in the partial model view, the properties
 234 can be retrieved recursively similar to the method for attribute extraction. The only difference is the starting point is
 235 from relation entities rather than *IfcObject* entities. A similar process also works for properties defined with the
 236 *IfcObjectType* entities. Thus, in order to retrieve all the properties for all selected major entities, all relation entities are
 237 checked.

238 Here, one relation entity can be referenced by multiple *IfcObject* entities and might be extracted multiple times. To
 239 avoid duplicated extraction and speed up extraction process, an optimization method is adopted. As shown in the dotted
 240 textbox, the extracted properties, with its GlobalID and values, are cached in a dictionary. When an object or a property
 241 needs to be parsed, its GlobalID is checked first in the dictionary. If found, the existing record is directly reused. For
 242 simplicity, Figure 5 only provides the caching operations for parsed main entities and properties. This process
 243 optimizes the property parsing process at the cost of additional memory overhead. For instance, for the relation entity
 244 (#52) described in Figure 2, Its attributes include the following properties: the GlobalID of relation entity, the
 245 references to both main entities(*IfcObject*) with id #50 and #71 and reference to the property (#51). There exists two
 246 *IfcObject* references (#50 and #71). They share the same set of properties. If the property extraction process visits this
 247 property set at second time, existing records for the retrieved properties is be directly reused.

248 **Implementations and Case Studies**

249 In this section, a prototypic implementation of this MVD and extraction algorithm is introduced. Then, this
 250 prototype is evaluated with two case studies.

251 **Implementation and validation**

252 The partial model view and corresponding algorithm is realized on the BIMserver platform, an open source BIM

platform built as a plugin framework for easy extension. BIMServer also provides a B/S model that is important for implementing the proposed distributed collaborative framework. This work is implemented on BIMServer with three major extensions. These three extensions allow our proposed models to directly use the off-the-shelf feature provided by BIMServer.

- The deserializer: The deserializer of IFC 2X3 is extended so it can generate three different models: the IFC central model used for later partial model (here, the term IFC central model is used to show it is the central model for the later partial model distilling), an IFC-validated partial model and our partial model.
- The data storage: The parsed data is stored together with the **central IFC model** as the sub-project provided by BIMServer. Similar to the data storage method used for IFC central model, our MVDs are stored into the BerkleyDB database with the key-value pairs. Compared to the IFC central/partial models, our storage format is comparable simple as our proposed algorithm maps all properties into direct properties.
- The rendering engine from BIMServer. In order to demonstrate the stored partial model, the rendering engine is revised to support our proposed model rendering. The original rendering logic is to retrieve a major object, then gradually retrieve all relationship entities related to this object, including the 3D properties. It differs from our approach in which the geometric properties are directly associated with the major objects.

In order to make a fair comparison towards IFC central and **IFC-validated partial model** and directly reuse the WebGL-based rendering service provided by BIMServer, all three types of approaches are stored as BIMServer subprojects in the BerkleyDB database rather than the IFC files. All these models are stored in the BIMServer with the sub-project feature and can be individually accessed. **For the IFC-validated partial model extraction**, the extraction algorithm proposed in reference ([Won, Lee et al. 2013](#)) is implemented for comparison. All these models are stored in the BIMServer with the sub-project feature provided by BIMServer and can be individually accessed.

The validity of the extracted model is also tested. Compared to the IFC-validity, our proposed partial MVD is not IFC-valid as it removes many IFC structure-related information, especially the relationship entities which compose a major set of all entities. Here, the validity is defined as whether all the objects and properties to be exchanged objects and properties are in deeded retrieved. More specifically, the validity of extracted model is checked with the following two way.

a) The comparison of our generated model and IFC-valid partial model. The main entities ids and property ids from both models are compared. Results show both model has identical set of those ids. This shows that our model has all the required information. The relation objects, and other IFC-related objects are not compared.

b) The rendering engine from the BIMServer. We use the rendering service from BIMServer to display our partial model. BIMServer does not log any object-missing errors and the displayed model is identical to the corresponding peers. [Won, Lee et al. \(2013\)](#) used this method in validating extracted IFC models.

Experiments

In order to test the performance of our approach, the IFC models of three open shared IFC projects shown in the Figure 6 are adopted. The Office MEP project is shared by buildingSMART alliance ([BuildingSMart Alliance 2016](#)). The Hospital and Conference center projects are shared by the Open IFC Model repository ([The University of AUCKLAND 2015](#)). The size of the IFC models are 6.26MB of the Hospital project, 40.9MB of Office_MEP and 68.7MB of Conference Center respectively. The Server configuration is with Windows 2008 Server 64bit, i5 4570, 8G RAM, 2T disk. The client side configuration runs windows 10 64bit, i7 3770, 8G RAM. The local network is 100Mbps. A revised BIMServer v1.40 with partial model support is used for model parsing and model storage and a Chrome v 53.0 running in the client-side is used for the model loading and the 3D and 2D data representation.

Several metrics are collected to evaluate the performance of difference approaches. The metric of *the transmission*

overhead measures the size of transmitted data between client and server. The loading and rendering time includes both the model transmission time and the time for 3D rendering. These two metrics are measured together as BIMServer adopts the progressive rendering techniques, data are represented while they are loading. The *memory usage* is collected from the memory used by the Chrome browser instance in the client. The *parsing time* specifies the time needed to generate the targeted models when the IFC file is deserialized into the BIMServer. The *storage time* measures the times needed to store a generated model (central or partial) into the database.

Those three approaches are compared with two different scenarios. In the first scenario, the partial model is generated with spatial constraints with geometric data for 3D display. In particular, the first floor division is adopted as guideline for partial model view definition for the three different projects. The second scenario adopts the requirements from the asset management perspective. This partial model view contains only information related to the asset management, e.g., manufacture, date of delivery. Only non-geometric data are needed.

Partial model with geometric data

The partial model view, as it is used for spatial division, is to select major entities related to the first floor. In order to do so, an *IFCBUILDINGSTOREY* major entity is found with its name equals “level 1”. Then, the relation entity of *IfcRelContainedInSpatialStructure* is found with reference to the first floor *IFCBUILDINGSTOREY* major entity. Then, this *IfcRelContainedInSpatialStructure* relation entity contains the reference of all the *IfcObject* entities that exist in the first floor. For all the extracted *IfcObject* entities, the geometric properties, spatial properties, material properties are selected for extraction. With the defined partial data view, the IFC central model is parsed and extracted. The visual representation of the extracted model is shown in the right side of Fig. 5. Table 1 shows the basic factors of those three projects and comparison results for the partial model with geometric data.

As shown from the table, the number of *IfcObject* entities for all three projects are significantly less than that of the complete model approach and equals to the IFC-validated partial model. Furthermore, the geometry related properties, instead of the complete entity, are included in our partial model. In contrast, the IFC-validated partial model contains complete entities with all related relation and property entities. The asterisk of our approach marks this difference. Our partial model contains far less unnecessary data for display. The structure of the model is also much simpler than that of the IFC-validate partial model as IFC-related structure-related data isn’t needed in our model.

As shown in the Table 1, the *transmission overhead* for the three projects are all considerably lower than the IFC central model and the IFC-validated model. The size of our model is 29.5%, 52.3% and 37.3% for their central model peers. The transmission overhead is only needs 78.9%, 74.4% and 71.3% of the IFC-validated partial model. Due to the simplified model complexity and model size, the model *loading and rendering time* for our model is also about 25% less than the IFC-validated partial model and far less than the IFC central model.

To generate the partial model (denoted as the *model generation time*), two major processes are needed. One process is to parse the IFC central model (measured with the *parsing time*) and the other process (measured with the *storage time*) is to store the generated model into the BIMServer database. Compared to the IFC-validated approach, our approach needs to recursively traverse the model for the selected properties. It results in increased parsing time, as shown in Table 1. However, on the other hand, our model contains less data for storage. It results in the decrease of storage time. Thus, the total overhead of the model generation process, displays complex behaviors: less time for the Hospital and Office MEP project and more time for the Conference Center project. However, as the parsing process only appears once for the whole lifecycle of the partial model and can be performed during system idle time, its impacts towards the overall system performance can be largely mitigated.

Partial model for the asset management

In addition to the requirements of geometric data, there exist the significant requirements for the non-visual and

data-oriented partial models. For instance, the asset management staff might need asset ID, production date, price etc. According to the asset management requirements, the partial model view for the asset management related to the first floor is defined and the corresponding partial model is generated. Here, due to the fact that the original shared project does not provides those properties, we insert those properties into corresponding major entities through the user-defined property supported by the IFC schema. In addition to the entity ids, three types of asset related properties are inserte: originalValue, owner and user. In this experiment, the values of those properties are arbitrary set due to the lack of related information. However, it should not influence the major conclusion of this comparison as both IFC partial model and our model contains the same asset information.

As can been seen from Table 2, for the partial model of the asset management, our approaches can achieve even higher improvements compared to the IFC-validated partial model. The final transmission overhead is only 45.35, 3763.4 and 1021.6 KB which is 1.92%, 4.79% and 3.82% of the IFC-partial model approach, due to the fact that both geometric data and IFC specification data are removed. For the model loading & rendering time metric, our approach only requires about 26.5%, 14.9% and 17.0% of its IFC-validated partial model peer. The data storage time also display similar improvements. Of course, our property-based partial model generation needs slightly more parsing time due to the introduced property retrieval overhead.

In summary, the simulation results show that for both scenarios, our approaches provide identical data required by the stakeholders with simpler model structure, less transmission overhead, less memory usage, and model loading and rendering time with slightly increased parsing time.

Conclusions

This paper studies the demands for the partial IFC models and points out the complexity and low data density problems in the existing IFC-validated based approaches. This paper proposes a partial model view defined with the property-based partial model extraction. This view is designed with the goal to eliminate the IFC-specification related and the data-unrelated structure while keep enough information for reversing mapping. Correspondingly, an extraction algorithm is also designed and implemented on the top of the BIMserver platform. The simulation results show that our approach is much better in terms of the data accessing time, the model loading time, the memory usage and the network transmission overhead than the central IFC model and compared IFC-validated partial model extracted based on the extraction algorithm in reference (Won, Lee et al. 2013). However, due to the facts of the extracted partial models by different algorithms shares the same set of independently exchangeable entities while with slightly difference on the sets of resource entities to be included. Thus, our solution, it is argued, should have better performance towards other similar IFC-validated partial model solutions.

In our current prototype, the partial model view is arbitrarily defined. We are working to integrate the existing BIM query language, e.g., BIM QL (Mazairac and Beetz 2013) or natural-language based query language (Lin, Hu et al. 2016), in defining the partial model view. Another limitation of our current partial model is the lack of model modification support as all the modifications have to be mapped back to the central IFC model. We have already made some progresses on this perspective. Parts of this work will appear in another paper. Currently, we are also working to provide a distributed version control system in supporting the IFC-based collaborative design. As different designers might inquiry, update their respective partial models, how to effectively manage different version of partial models is another important domain to be investigated.

References

- Beetz, J., L. van Berlo, R. de Laat and P. van den Helm (2010). BIMserver. org—An open source IFC model server. Proceedings of the CIP W78 conference.
- Boukamp, F. and B. Akinici (2007). "Automated processing of construction specifications to support inspection and

quality control." Automation In Construction **17**(1): 90-106.

BuildingSMart Alliance. (2016). "BuildingSMart Alliance Common Building Information Model Files and Tools." from https://www.nibs.org/?page=bsa_commonbimfiles.

Cemesova, A., C. J. Hopfe and R. S. Mcleod (2015). "PassivBIM: Enhancing interoperability between BIM and low energy design software." Automation in Construction **57**: 17-32.

Chen, P.-H., L. Cui, C. Wan, Q. Yang, S. K. Ting and R. L. Tiong (2005). "Implementation of IFC-based web server for collaborative building design between architects and structural engineers." Automation in construction **14**(1): 115-128.

Daum, S. and A. Borrmann (2014). "Processing of topological BIM queries using boundary representation based methods." Advanced Engineering Informatics **28**(4): 272-286.

East, E. W. (2007). Construction operations building information exchange (COBie), DTIC Document.

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

Gupta, A., A. Cemesova, C. J. Hopfe, Y. Rezgui and T. Sweet (2014). "A conceptual framework to support solar PV simulation using an open-BIM data exchange standard." Automation in Construction **37**: 166-181.

Kim, H., Z. Shen, I. Kim, K. Kim, A. Stumpf and J. Yu (2016). "BIM IFC information mapping to building energy analysis (BEA) model with manually extended material information." Automation in Construction **68**: 183-193.

Lee, S. H. and B. G. Kim (2011). "IFC Extension for Road Structures and Digital Modeling." Procedia Engineering **14**: 1037-1042.

Liebich, T. (2013). "IFC4—the new buildingSMART Standard." Retrieved Dec 20, 2016, from http://www.buildingsmart-tech.org/specifications/ifc-releases/ifc4-release/buildingSMART_IFC4_WhatisNew.pdf.

Lin, J. R., Z. Z. Hu, J. P. Zhang and F. Q. Yu (2016). "A Natural-Language-Based Approach to Intelligent Data Retrieval and Representation for Cloud BIM." Computer-Aided Civil And Infrastructure Engineering **31**(1): 18-33.

Mazairac, W. and J. Beetz (2013). "BIMQL—An open query language for building information models." Advanced Engineering Informatics **27**(4): 444-456.

Motamedi, A., M. M. Soltani, S. Setayeshgar and A. Hammad (2016). "Extending IFC to incorporate information of RFID tags attached to building elements." Advanced Engineering Informatics **30**(1): 39-53.

Solihin, W., C. Eastman and Y. C. Lee (2016). "A framework for fully integrated building information models in a federated environment." Advanced Engineering Informatics **30**(2): 168-189.

The University of AUCKLAND. (2015). "Open IFC Model Repository." from <http://openifcmodel.cs.auckland.ac.nz/>.

Weise, M., P. Katranuschkov and R. J. Scherer (2003). "Generalised model subset definition schema." CIB REPORT **284**: 440.

Wix, J. and J. Karlshoej (2010). "Information delivery manual: Guide to components and development methods." BuildingSMART International.

Won, J., G. Lee and C. Cho (2013). "No-Schema Algorithm for Extracting a Partial Model from an IFC Instance Model." Journal Of Computing In Civil Engineering **27**(6): 585-592.

Yan, W., M. Clayton, J. Haberl, W. Jeong, J. B. Kim, S. Kota, J. L. B. Alcocer and M. Dixit (2013). Interfacing BIM with building thermal and daylighting modeling. 13th International Conference of the InternationalBuildingPerformanceSimulationAssociation (IBPSA).

Yang, D. and C. M. Eastman (2007). "A Rule - Based Subset Generation Method for Product Data Models." Computer - Aided Civil and Infrastructure Engineering **22**(2): 133-148.

Zhang, J., F. Yu, D. Li and Z. Hu (2014). "Development and Implementation of an Industry Foundation Classes - Based Graphic Information Model for Virtual Construction." Computer - Aided Civil and Infrastructure Engineering **29**(1): 60-74.

Zhang, L. and R. Issa (2012). "Comparison of BIM cloud computing frameworks." Computing in Civil Engineering

2012: 389-396.

Zhang, Y. (2009). Research on BIM-based Building Information Integration and Management PhD Thesis, Tsinghua University.

Figure List:

- Figure 1. Distributed BIM-Based Collaborative Design
- Figure 2. IFC description for a column object and its properties
- Figure 3. Definition of Partial Model View
- Figure 4. IFC Property Description （Express-G）
- Figure 5. Algorithm for the attribute extraction
- Figure 6. IFC central models and their partial models for three projects
- a.1 Hopstial_Parking Garage a.2 The partial model of a.1
- b.1. Office MEP b.2. The partial model of b.1
- c.1. OTC-Conference Center c.2 The partial model of c.1

Table 1. The performance comparisons with three different approaches

	Hospital			Office_MEP			Conference Center		
Metrics	IFC Central Model	IFC-validated Partial.Model	Our Appr.	IFC Central Model	IFC-validated Partial Model	Our Appr.	IFC Central Model	IFC-validated Partial Model	Our Appr.
Major entities(num.)	1,209	49	49	5,697	3,964	3,964	6,021	2,677	2,677
Relation entities(num)	5,054	1,277	0	42,197	29,749	0	21,478	15,055	0
Property entities(num)	785	218	0	15,663	12,941	0	1,909	1,432	0
Load & Rendering(s)	5.15	1.17	0.73	24.1	15.5	9.238	19.7	12.3	8.9
Trans. overhead (kB)	6,165	2359.9	1817.1	111,734	78,546.4	58,435.	50,349.5	26,735.6	19,062.6
Memory usage (GB)	0.317	0.03	0.018	0.6--01	0.443	0.255	0.866	0.385	0.217
Data storage time(s)	7.9	3.2	2.6	41.3	31.5	21.3	62.3	27.5	14.6
Parsing time (s)	34.8	35.6	36.1	95.5	100.3	117.8	206.9	216.3	219.5

Table 2. The performance of our approach for the asset-management

	Hospital		Office MEP		Conference Center	
	Our Appr.	IFC Partial model	Our Appr.	IFC Partial model	Our Appr.	IFC partial model
loading & rendering time	0.43	1.17	2.31	15.5	2.09	12.3
transmission overhead (kb)	45.35	2359.9	3763.4	78,546.4	1021.6	26,735.6
storage time	0.31	3.2	1.93	31.5	1.46	27.5
parsing time (s)	34.4	35.6	119.1	100.3	213.7	216.3

Figure2. IFC description for a column object and its properties

[Click here to access/download;Figure;figure2-](#)

```
#50=
IFCCOLUMN('3zDh5EEab3N9JwS9XH8dhw',#41,'X2\
57FA672C5899\X0\:\X2\5E3889C4\X0\ - 200mm:154889',$,\
X2\57FA672C5899\X0\:\X2\5E3889C4\X0\ -
200mm:1740',#133,#160,'154889');

#51=
IFCPROPERTYSINGLEVALUE
('originalValue',$,IFCREAL(1000.),$);

#52=
IFCRELDEFINESBYPROPERTIES('2qf77y4kPCxvr2m1kpzV_',
#41,$,$,(#50,#71),#51)

#53=
IFCPROPERTYSET('0kpMDU0vX1xgVPXgFzO1_V',#41,
'Text(Type)',$,(#51));

#133=
IFCLOCALPLACEMENT(#121,#132);

#160=
IFCPRODUCTDEFINITIONSHAPE($,$,(#139,#157));

#618=
IFCRELASSOCIATESMATERIAL('3cEGiCHjX2HwKbmfrwBeyF',
#41,$,$,(#50),#250);

#636=
IFCRELDEFINESBYTYPE('2CN0azncjCXhSE2ltWyQoc',#41,$,$,(#
50),#251);
```

choose

choose

Figure1. Distributed BIM-Based Collaborative Design

Figure3. Definition of Partial Model View

[Click here to access/download;Figure;figure3.pdf](#)

Figure4. IFC
Property

(ABS) IfcObject

[Click here to
access/download](#)

Figure5. Algorithm for the attribute extraction

Figure6a1.
Hopstia Parking

Click here to
[access/download,Fi](#)

Figure6a2. The
partial model of a.1

Click here to
[access/download;Fi](#)

Figure6b1. Office
MEP

Click here to
[access/download;Fi](#)

Figure6b2. The
partial model of b: 1

Click here to
[access/download;Fi](#)

Figure6c1. OTC-
Conference Center

[Click here to
access/download;Fi](#)

Figure6c2. c.2 The
partial model of c.1

[Click here to
access/download;Fi](#)

Response Letter

Dear Editor,

We would like to sincerely thank you for giving us an opportunity to improve the manuscript based on the valuable comments and constructive suggestions made by you and the reviewers. In what follows we answer the questions raised by the reviewers and the changes for grammar errors.

Yours sincerely,
Ning Gui on behalf of all the authors

Question 1.0.1 Associate Editor: The reviewers and I concur that the authors have not yet addressed all of the issues that were raised in the first round of reviews - in fact, this revision has also raised a few additional concerns, as listed in the reviews. Please make sure to cover all of the issues thoroughly in your next revision - if you fail to do so, we will not progress to another review. Also, please pay special attention to the grammar and writing quality, typos, etc. Please also remove "The Research of" from the title. In addition, please respond point-by-point to each review comment. In responding to a comment, please note any changes made in the manuscript and refer to the respective line numbers.

Answer 1.0.1: *Thank you very much for your kind suggestions. We double-checked this paper and correct grammar and typos. All the questions raised by the two reviewers are responded point-by-point. The questions and answers are numbered with 1.X.Y, 1 represents the second time Q&A and X represents the reviewer X and Y represents the Yth question from reviewer X.*

Following your suggestion. The "The Research of" is removed from the title.

Reviewer #1: The authors made some revisions to address the comments. However, the authors failed to adequately address some of the comments.

Question 1.1.1 The authors explain properties are domain-specific. They also try to illustrate the examples in Figure 2 for asset management domain. However, the properties that are needed for asset management are still not clear. In the text, they mentioned only asset-management related properties are included. What are these properties? Also, Figure 2 is not very clear.

Answer 1.1.1: *In our previous submission, we stated that "...for a particular stakeholder, for instance, in the asset management domain, information of major interest is the object name, and prices and location rather than its geometric data and the relation entities.". Thus, the demonstrative properties related to asset-management, used in this paper, is orginalValue, user and owner. Of course, real properties related to asset-management might be more complex and might have different sets for different practical scenes.*

In Figure 2, we are sorry that the property names are originally in Chinese that is why you might not see the string of those properties. The string name start with \X2 are the Unicode of the Chinese

characters. We provided a new version of the ifc file and change the properties into English (Here, only the originalValue is demonstrated in new Figure 2 due to space limitation). We hope this time it is much clearer to understand.

Question 1.1.2 The authors added information on testing the validity of the extracted model. They define validity as "whether all the objects and properties to be exchanged objects and properties are in deeded retrieved." However, the method that is used to test such validity is questionable. For example, how to make sure the extracted asset-related information is actually the relevant and specific information used for asset management? Do you adopt any gold standards? Do you involve any experts in developing the gold standards?

Answer 1.1.2: Thank you for your question. However, your question is "how to make sure the extracted asset-related information is actually the relevant and specific information used for asset management? However, we think your question is more related to the **correct definition of the MVD (it is a view) rather than the validity of the extracted model**. Thus, your question, from my understanding, is how to make sure the definition of the MVD is correct. Many researchers agree that it is very difficult to find golden standards to define a general domain-specific MVD. The complexity in the definition of a MVD, to some extent, explains the slow progress of "the standard MVDs for typical information exchange tasks" initialized via buildingSMART alliance. That is why many approaches now study how to have fast and flexible MVD generation according to some requirements. This fact has been explained in the introduction section of our submission (see line. 65 and 66).

Furthermore, for the collaborative design scenes in which our paper most focused on, **a general MVD for a certain domain-specific domain is not so useful, it is argued, even if it is available**. At different design stages, a user might use different partial models. For instance, for the asset management, in the early stage, the user might focus more on the originalValue and owner, etc. While, during the maintenance stage, user might focus or work more on the TotalReplacement Cost and user, etc. Thus, the flexible definition of MVDs (typically via Query languages) are more appeal to our scenes. This paper focuses on how to extract information efficiently and correctly **according to the MVD rather than for certain general domain**, e.g. the asset management domain. In the design of MVD, we have worked with several designers. The collaborative design actually evolves from our experience on the BIPV (Building Integrated Photovoltaic) design process, which involves experts from several domains, e.g. design, structure, electrical, PV and green building design... Our work on the BIPV and the BIM-based collaborative design process have been published in the Energy and Buildings[1] and Automation in Construction[2].

Question 1.1.3 The terminology in describing the three models in Table 1 is still not consistent, which could cause confusion. For example, is "IFC Central Model" the same with "Complete IFC model"? If yes, why use two different terms? If no, the difference should be explained in text.

Answer 1.1.3: Thank you very much for your suggestion. For this paper, the "IFC Central Model" and "Complete IFC model" are two interchangeable term due to the fact that our paper focuses on the

collaborative design environments. Yes, we do agree these two terms creates ambiguity, so we change the “Complete IFC model” to the “IFC Central model” throughout the paper.

Question 1.1.4. Line 305: The properties that are manually inserted into the original model should be explained and described in the paper. Also, how to ensure the information on these properties is correct? Are there any experts involved in inserting or evaluating these properties?

Answer 1.1.4 : The open shared IFC models contain no information related to the asset management.

In order to extract partial model for asset management for experiments, we have to introduce some properties for this domain. Here, the manually inserted properties are referred to the following process (some are operated manually, some are operated via Revit).

- Open IFC model in Revit;
- Create the sharable parameters in the Revit for the current projects, three properties are created: original price (value), owner (text) and user (text)
- Assign parameters to the major types of IFC entities.

- Save and export the IFC model (with the following settings).

Those processes are manually performed. We do not elaborate those process as we thought the detailed descriptions of this process might result in the deviation from the objectives of our paper.

Actually, we can also define a Revit plugin or use the IFC parser to automate this process, which is not necessary as Revit allows the batching assignments of properties.

As for the “how to ensure the information on these properties is correct?”

We do not need to guarantee the inserted information be correct. We only need to ensure that the format of those properties are correct. In the experiment, the same property value is used for originalValue(1000), User(Smith) and owner(Smith). For our algorithm, as long as their formats are correct, it generate a light-weight partial model based on the MVD and the central IFC model. The insurance of the correct information is the responsibility of the model designers. It is hard to have a general guideline for those values. For instance, the original values of the entities might vary greatly and we cannot imagine having a general guideline for entering those values. However, it would not influence the major conclusions of this comparison as both IFC partial model and our model contain the same asset information.

In the revised manuscript, the descriptions on the inserted properties and the value of those properties are illustrated. One sentence on the impacts of the different value of the properties has also been added.

Question 1.1.5. Line 307-308: The style of Table 2 is confusing. I suggest changing the table style.

Answer 1.1.5. *The original design is to reduce the size of the table. We agree it might create some ambiguity. According to your suggestion, Table 2 is revised to the current style.*

Table 2. The performance of our approach for the asset-management

	Hospital			Office MEP			Conference Center		
	Our	IFC	Partial	Our	IFC	Partial	Our	IFC	partial
	model	model		model	model		model	model	
loading & rendering time	0.43	1.17		2.31	15.5		2.09	12.3	
transmission overhead (kb)	45.35	2359.9		3763.4	78,546.4		1021.6	26,735.6	
storage time	0.31	3.2		1.93	31.5		1.46	27.5	
parsing time (s)	34.4	35.6		119.1	100.3		213.7	216.3	

Question 1.1.6. Line 323-325: The authors revised the sentence a little to conclude that the proposed method is "much better" than other approaches. However, the authors only compare this method with one previous method (Won et al. 2013). The authors need to further revise this sentence to be more conservative. Its current narrative seems to imply the proposed method is better than all other existing extraction methods, which is not justified by the experiments. The authors also need to clearly define IFC-validated approach in the paper, does this only mean the approach used by Won et al. (2013) or does it include other existing approaches?

Answer 1.1.6: *Thank you very much for your suggestion. In the experiment, we only compared with the central IFC model and IFC-validated partial model. There are many different extraction methods to generate*

the partial models. The resulting models might be various from different extracting methods, as pointed out in [3], and might have slight difference in the set of resource entities which are not independently exchangeable. Different extraction algorithms might take different standards in retrieving the resource entities. **However, the major parts of the IFC model keep the same (major entities, type entities, relation entities, and property sets entities).** For all the IFC-partial models, there do exist slightly differences in how to retrieve the resource entities. From reference [3], there are less than 4% differences in respect to the number of entities. The major differences in those extraction algorithms are the granularity of the resource entities. The more detailed definition of MVD, the less unnecessary data while it create complexity in definition.

That is why we conclude our partial model should be better than general IFC-validated partial models, including the approach via Won et al (2013). Yes, we agree this conclusion is too strong as we do not compared to those approaches.

Thus, this sentence is revised as follows :

“The simulation results show that our approach is much better in terms of the data accessing time, the model loading time, memory usage and the network transmission overhead than the central IFC model and compared IFC-validated partial model extracted based on the extraction algorithm in reference [4]. However, due to the facts of the extracted partial models by different algorithms shares the same set of independently exchangeable entities while with slightly difference on the sets of resource entities to be included. Thus, our solution, it is argued, should have better performance towards other similar IFC-validated partial model solutions. “

Question 1.1.7 The paper still has language/grammar mistakes and typos. For example, Line 240, incomplete sentence. Line 359, extra "a". Line 279, "results shows". There are other cases.

Answer 1.1.7: Yes, you are right. We revised those mistakes and double-checked other parts for mistakes and errors. Actually, we submitted the wrong version in the last submission process. The “incomplete sentence” and “extra a” result from the wrong file.

Reviewer #2: The authors have addressed all the questions.

Question 1.2.1: Regarding Answer 2.1 It is okay that if the authors clarify in the text that this paper focuses on developing a method for retrieving data from an IFC file. Then it worth adding several sentences to line 49 about why this step is essential in the big picture.

Answer 1.2.1: Thank you very much for your suggestions. Actually, we provided discussions in the previous submission from Line 41 to 43. Here, our discussions do not limit to the IFC model as other formats also face the same partial model extraction, modification and synchronization problems. In the revised manuscript, Line 56~57, we provided discussions on why the IFC is adopted: it is the only standard format for BIM information exchange and it is open and extensible. There are some commercial format, for instance, Revit. However, they are not open and changes rapidly.

Question 1.2.2: I look forward to seeing the authors' new papers. It is arguable if IFC is appropriate

for BIM version control. As you mentioned you finally decide to use Revit API to do this. Some BIM authoring tools might generate different GlobalIDs in IFC files for the same object if you made any changes to the model. In this case, merely using GlobalID may not be able to track the object. As a result, it becomes a question that is it worth developing a new IFC-based data retrieval method for version control? This is another story. The authors don't have to address this in the text.

Answer 1.2.2: *Yes, you are right, some BIM authoring tools, e.g. Revit, might generate different GlobalIDs in IFC files for the same object if you made any changes to the model. However, they keep the internal ids for each object unchanged. In the Revit plugins, we use the project name, version and internal id as the “globalID” to be used. The globalID is not practically used. However, it is more implementation-based practice and it might not influence the general process for mapping and synchronization.*

Question 1.2.3 Regarding Answer 2.2 It is not convincing that the advantage is your method can get the GlobalID, because using others' query engines you can get that as well.

Answer 1.2.3: *Yes, you are right. The advantage is definitely not that our method can get the globalID. The advantage is that we need a light-weight MVD that can best fit the collaborative environments that can extract domain-specific or user-specific partial model from the IFC central model. Most importantly, changes on those partial models should be able to synchronized back to the central model. The design of the property-based MVD is to support those requirements. To get the GlobalID is a necessary step in our approaches. In this paper, we never express, directly or indirectly, this step is our advantage. The previous answer 2.2 just shows that why we cannot simply use the Query language for partial model extraction. Some formats must be enforced to construct a valid view.*

Question 1.2.4 Question, Regarding Answer 0.2.5 According to buildingSmart, "An IFC View Definition, or Model View Definition, MVD, defines a subset of the IFC schema, that is needed to satisfy one or many Exchange Requirements of the AEC industry." An MVD must be domain-specific. An analogy is the View definition in the database design. What the authors propose is another model data representation: object-property-value, which is different from the syntax of IFC but similar to data triples in RDF. As a result, the term MVD is not used correctly in the whole paper.

Answer 1.2.4 : *Yes, our term “MVD” does not fully conform this definition of Model View Definition of buildingSMART. The buildingSMART limit the term MVD to the **subset of IFC schema** which is not the case in our paper.*

*However, buildingSMART also agree that COBie, as an official buildingSMART IFC MVD with the name “Basic FM Handover View”, indicate the **subset of IFC data model**. COBie also does not conform the general IFC schema. From this perspective, those two definitions are not consistent. Thus, we think MVD can be used as a general term to define a subset of BIM data changes.*

In order to clarify this ambiguity, in this new version, we explicitly add discussions on the term of MVD.

It is worth to note that the term of MVD is originally used to define a subset of the IFC schema. However, COBie is also an IFC MVD with the name “Basic FM Handover View” in which this term is

used to indicate the subset of the IFC data model, rather than the schema (COBie adopts a spreadsheet based data-collection/ delivery format). Thus, the MVD used in paper mainly follows the later scene.

Question 1.2.8 Regarding Answer 0.2.5, The term "IFC-validated partial model" is confusing if it was not an IFC file or in IFC format, maybe just use "partial model with GlobalID."

Answer 1.2.8: We are sorry that we cannot find the IFC-validated model in the Answer 0.2.5. We guess it should be Answer 0.2.8. We have to clarify that "the IFC-validated partial model is doing in IFC format and can be stored in IFC file. ONLY in the experiment section, the IFC-validated model are stored in the BIMServer database for comparison as we use the BIMServer as the parsing, manage and display platform to avoid the reinvent of the wheel. This has been answered in the previous Answer 0.2.7.

Question 1.2.9 Answer 0.2.8 and 0.2.9 should be incorporated into the manuscript

Answer 2.9: Thank you very much for your suggestion. The Answer 0.2.8 and 0.2.9 is now incorporated.

Answer 0.2.8 for difference between the one-time delivery and collaborative design scenes are introduced in Line 89-91. Answer 0.2.9 for the efficiency of COBie is introduced in Line 87 and 88.

Question 1.2.10 Answer 0.2.7 says the IFC-validated partial model is not in IFC format, while Answer 2.8 says it is. Please be consistent and be careful about choosing terms that are not confusing

Answer 1.2.10: The IFC-validated partial model, the same as the IFC central model, is doing in IFC format. Only in the experiments, they are parsed and stored in the databased for comparison. Please refer to the Answer 1.2.8 and previous Answer 0.2.7.

- [1] G. Ning *et al.*, "BIM-based PV System Optimization and Deployment," *Energy and Buildings*, 2017.
- [2] G. Ning, H. Kan, Q. Zhifeng, G. Weihua, and D. Geert, "e-BIM: a BIM-centric design and analysis software for Building Integrated Photovoltaics," *Automation in Construction*, vol. 87, pp. 127-137, 2018.
- [3] F. YU, J. Zhang, and Q. LIU, "Semi-automatic generation of the BIM model view definition based on the IFC standard," *J Tsinghua Univ.*, no. 8, pp. 987-992, 2014.
- [4] J. Won, G. Lee, and C. Cho, "No-schema algorithm for extracting a partial model from

an IFC instance model," *Journal of Computing in Civil Engineering*, vol. 27, no. 6, pp.
585-592, 2013.