Automation in Construction 67 (2016) 1–21

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison

Yichuan Deng [a], Jack C.P. Cheng [a,*], Chimay Anumba [b]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong, China
[b] Department of Architectural Engineering, The Pennsylvania State University, 104 Engineering Unit A, University Park, PA 1680, USA

ARTICLE INFO

Article history:
Received 2 June 2014
Received in revised form 22 February 2016
Accepted 13 March 2016
Available online 6 April 2016

Keywords:
CityGML
IFC
Interoperability
Levels of detail
Reference ontology
Schema mapping

ABSTRACT

The Building Information Modeling (BIM) domain and the Geographic Information System (GIS) domain share a mutual need for information from each other. Information from GIS can facilitate BIM applications such as site selection and onsite material layout, while BIM models could help generate detailed models in GIS and achieve better utility management. The mapping between the key schemas in the BIM domain and the GIS domain is the most critical step towards interoperability between the two domains. In this study, Industry Foundation Classes (IFC) and City Geography Markup Language (CityGML) were chosen as the key schemas due to their wide applications in the BIM domain and the GIS domain, respectively. We used an instance-based method to generate the mapping rules between IFC and CityGML based on the inspection of entities representing the same component in the same model. It ensures accurate mapping between the two schemas. The transformation of coordinate systems and geometry are two major issues addressed in the instance-based method. Considering the difference in schema structure and information richness between the two schemas, a reference ontology called Semantic City Model was developed and an instance-based method was adopted. The Semantic City Model captures all the relevant information from BIM models and GIS models during the mapping process. Since CityGML is defined in five levels of detail (LoD), the harmonization among LoDs in CityGML was also developed in order to complete the mapping. The test results show that the developed framework can achieve automatic data mapping between IFC and CityGML in different LoDs. Furthermore, the developed Semantic City Model is extensible and can be the basis for other schema mappings between the BIM domain and the GIS domain.

1. Introduction

The architecture, engineering, and construction (AEC) industry is fragmented and yet information intensive [1]. Information sharing between stakeholders in the AEC industry is critical for collaboration during the construction process. Building Information Modeling (BIM) provides a solution for the interoperability in the AEC industry by providing an information backbone throughout the building life cycle. BIM is the process to create, store, and manage the information related to buildings throughout their whole life cycle [2]. Using BIM, different parties involved in the building process can work on a common platform, where the cost of information sharing is much less. While BIM aims to solve the problem of interoperability between stakeholders within the AEC industry, the integration of BIM with other systems, such as Geographic Information System (GIS), is becoming increasingly important. In the AEC industry, it has been reported that more than 80% of information could be referenced from geographical information [3]. Thus, the integration between BIM and GIS could further enhance information sharing.

GIS is a system to capture, store, manipulate, analyze, manage, and present all types of geographical data [4]. Traditionally, GIS is based on 2D maps, in which objects are assigned 2D references such as longitude and latitude. Currently, 3D GIS is also emerging. 3D GIS schemas such as KML [5], COLLADA [6] and Geography Markup Language (GML) [7] are able to store 3D attributes of objects in GIS, which enhances the functionality of GIS. There have been several studies concerning the application of GIS in the AEC industry. Su et al. [8] reported a GIS-based dynamic construction site material layout evaluation for building projects. Simão et al. [9] used a web-based GIS for collaborative planning and public participation in the planing of wind farm sites. Isikdag et al. [10] investigated the application of GIS to support site selection and fire response for BIM models. Anumba [11] developed a GIS-based approach to labor market planning in consruction. A comprehensive review about the application of GIS in construction activities was presented in [12]. BIM models are also rich information sources for GIS for certain applications. Benner et al. [13] presents a framework to generate

* Corresponding author at: Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. Tel.: +852 2358 8186; fax: +852 2358 1534.
E-mail address: cejcheng@ust.hk (J.C.P. Cheng).

http://dx.doi.org/10.1016/j.autcon.2016.03.006
0926-5805/

semantic 3D buildings using BIM models. Hijazi et al. [14] introduced their initial investigations for modeling interior utilities in 3D GIS buildings from BIM models. Some applications also require collaboration between the GIS and BIM models. For example, Strzalka et al. [15] presented an urban scale heating energy demand forecasting system by combining information from GIS and BIM models. Thiis and Hjelseth [16] tried to use BIM and GIS to enable climatic adaptations of buildings.

It has been shown in previous studies that the BIM domain and the GIS domain have mutual need of information from each other. For example, when designing a building in a crowded urban environment, noise could be a concern for designers. By using data from GIS on neighborhood buildings and surrounding infrastructure, designers could have a much better understanding about the noise sources and take necessary measures in the design process in BIM. This kind of design process uses data from both BIM and GIS, and a seamless data integration between BIM and GIS should be achieved. The automatic data mapping between data schemas in the BIM models and GIS models must be achieved first in order to exchange data seamlessly.

In this study, the Industry Foundation Classes (IFC) and City Geography Markup Language (CityGML) are chosen as the representative schemas for the BIM domain and the GIS domain, respectively. IFC is an EXPRESS-based open data standard initiated by buildingSMART (formerly the International Alliance for Interoperability) in 1994. Supported by most of the BIM software in the AEC industry, IFC is believed to be the most popular BIM standard. On the other hand, CityGML is a GIS standard developed by SIG3D (Special Interest Group 3D). It was adopted as an official OGC (Open Geospatial Consortium) standard in 2008 by OGC members. It is a semantic-rich data standard which supports five levels of detail (multi-resolution) modeling of city objects. The data mapping between IFC and CityGML could be challenging due to the fact that the two schemas are proposed for completely different purposes. IFC tries to capture all the information relevant to a building, such as detailed geometry of building components and semantic information such as cost, scheduling, and utility information. CityGML models are usually used to capture demographic information with reference to a map or the geometry of buildings. IFC is EXPRESS-based, in which the entities are refered to each other by line number, while CityGML is an XML-based schema which uses the XML Schema Definition (XSD) to define the relationships between entities. Since there are a large number of entities in both schemas and the stucture of the two schemas are different, an instance-based inspection of entities in both schemas is desirable. Furthermore, due to the fact that IFC contains much more information than CityGML, a seamless information exchange will require the extension of the CityGML schema, for which a reference ontology is proposed to store all the relavant information from IFC models and CityGML models.

In this paper, we propose a mapping framework between IFC and CityGML which consists of four components:

- Transformation of geometry among BRep, Swept Solid, and Constructive Solid Geometry
- Transformation of coordinate system
- Schema mediation using reference ontology for different sets of terminology from IFC and CityGML, and
- Transformation among different LoDs in CityGML

Different components in IFC and CityGML from the building models will be extracted and compared to generate the mapping rules. The reference ontology is developed based on the attributes in these two schemas. Special attention is given to the inverse-attributes in the schemas of IFC and CityGML, which have not been covered by previous studies.

The remainder of the paper is structured as follows: Section 2 introduces the two data schemas and reviews previous methodology proposed to achieve the mapping between IFC and CityGML. Section 3 presents our proposed methodology framework utilized for the mapping. Section 4 discusses the transformation between LoDs in

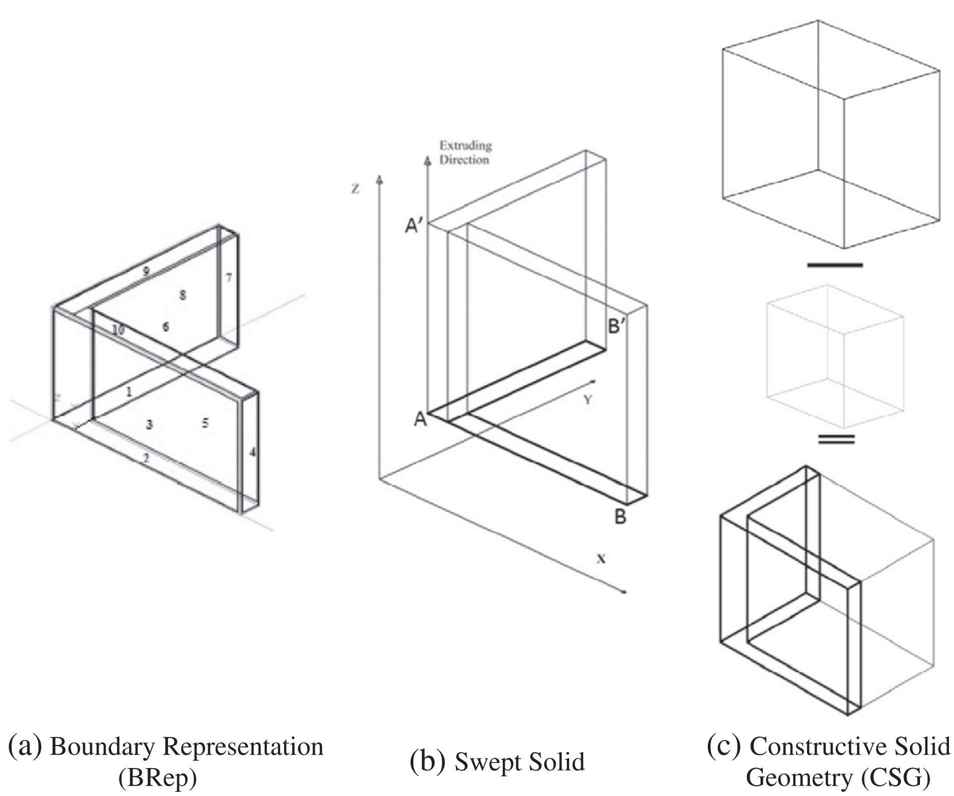


Fig. 1. BRep, Swept Solid. and CSG.

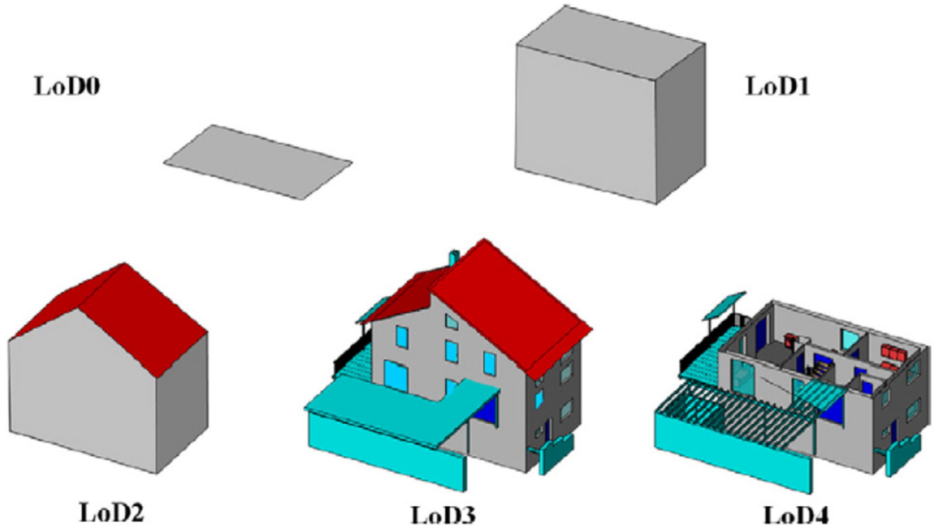


Fig. 2. LoDs in CityGML (adapted from Gröger and Plümer, 2012).

CityGML in order to complete the mapping. The results are discussed in Section 5, which is followed by a number of conclusions in Section 6.

2. Research background and related works

2.1. Introduction to IFC and CityGML

IFC is an object-oriented open standard initiated by buildingSMART in 1994. It has now become a formally registered international standard as ISO/PAS 16739. IFC supports a wide range of geometric representations as well as rich semantic information. The objects in IFC format could be represented by one of, or the combinations of Boundary Representation (BRep), Swept Solid, and Constructive Solid Geometry (CSG). As shown in Fig. 1, BRep uses the boundary surfaces to define the shape of objects. In Swept Solid, the cross-section of objects will be defined first and then extruded in certain directions to a defined length in order to show the solid shape. CSG is the result of Boolean operations of objects. For instance, as shown in Fig. 1, a big box minus a smaller box will generate an L-shaped object. In addition, IFC could store various kinds of semantic information, such as owner information, modification history of model, and cost and schedule of building components. BIM models based on IFC could be used in various phases of the construction, such as in feasibility studies, tendering [17], code checking [18], and operation management [19].

CityGML was developed by Special Interest Group 3D (SIG 3D) of the initiative Geodata Infrastructure North Rhine-Westphalia in Germany in 2002. It was accepted as an official Open Geospatial Consortium (OGC) standard in 2008. CityGML is the first GIS schema to support rich semantic information [20]. It supports component-based modeling in which different building components in the GIS are assigned unique IDs, names, and descriptions. CityGML also supports five levels of detail (LoDs) that vary from LoD0, which is basically a regional 2D map, to

```
#719=IFCWALLSTANDARDCASE('3uLk5Utsz5exvK686zwv8C',#33,'Basic Wall:Exposed - 115mm Brk:157936',$,'Basic Wall:Exposed - 115mm Brk:110171',#703,#718,'157936');
  #33=IFCOWNERHISTORY(#32,#2,$,.NOCHANGE.,$,$,$,0);
    #32=IFCPERSONANDORGANIZATION(#30,#31,$);
      #30=IFCPERSON($,$,'Allen',$,$,$,$,$);
      #31=IFCORGANIZATION($,'','',$,$);
    #2=IFCAPPLICATION(#1,'2011','Autodesk Revit Architecture 2011','Revit');
      #1=IFCORGANIZATION($,'Autodesk Revit Architecture 2011',$,$,$);
  #703=IFCLOCALPLACEMENT(#46,#702);
    #46=IFCLOCALPLACEMENT(#25,#45);
      #25=IFCLOCALPLACEMENT(#4184,#24);
        #4184=IFCLOCALPLACEMENT($,#4183);
          #4183=IFCAXIS2PLACEMENT3D(#3,$,$);
            #3=IFCCARTESIANPOINT((0.,0.,0.));
        #24=IFCAXIS2PLACEMENT3D(#3,$,$);
          #3=IFCCARTESIANPOINT((0.,0.,0.));
      #45=IFCAXIS2PLACEMENT3D(#44,$,$);
        #44=IFCCARTESIANPOINT((0.,0.,900.));
    #702=IFCAXIS2PLACEMENT3D(#701,$,$);
      #701=IFCCARTESIANPOINT((-11573.326807107,-6075.139430185804,0.));
  #718=IFCPRODUCTDEFINITIONSHAPE($,$,(#706,#717));
    #706=IFCSHAPEREPRESENTATION(#27,'Axis','Curve2D',(#705));
    #717=IFCSHAPEREPRESENTATION(#27,'Body','SweptSolid',(#711));
      #27=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.E-006,#26,$);
        #26=IFCAXIS2PLACEMENT3D(#3,$,$);
      #711=IFCEXTRUDEDAREASOLID(#709,#710,#9,3000.);
        #709=IFCRECTANGLEPROFILEDEF(.AREA.,$,#708,19300.0000000003,114.9999999999998);
        #710=IFCAXIS2PLACEMENT3D(#3,$,$);
        #9=IFCDIRECTION((0.,0.,1.));
```

Fig. 3. Information about a wall in IFC.

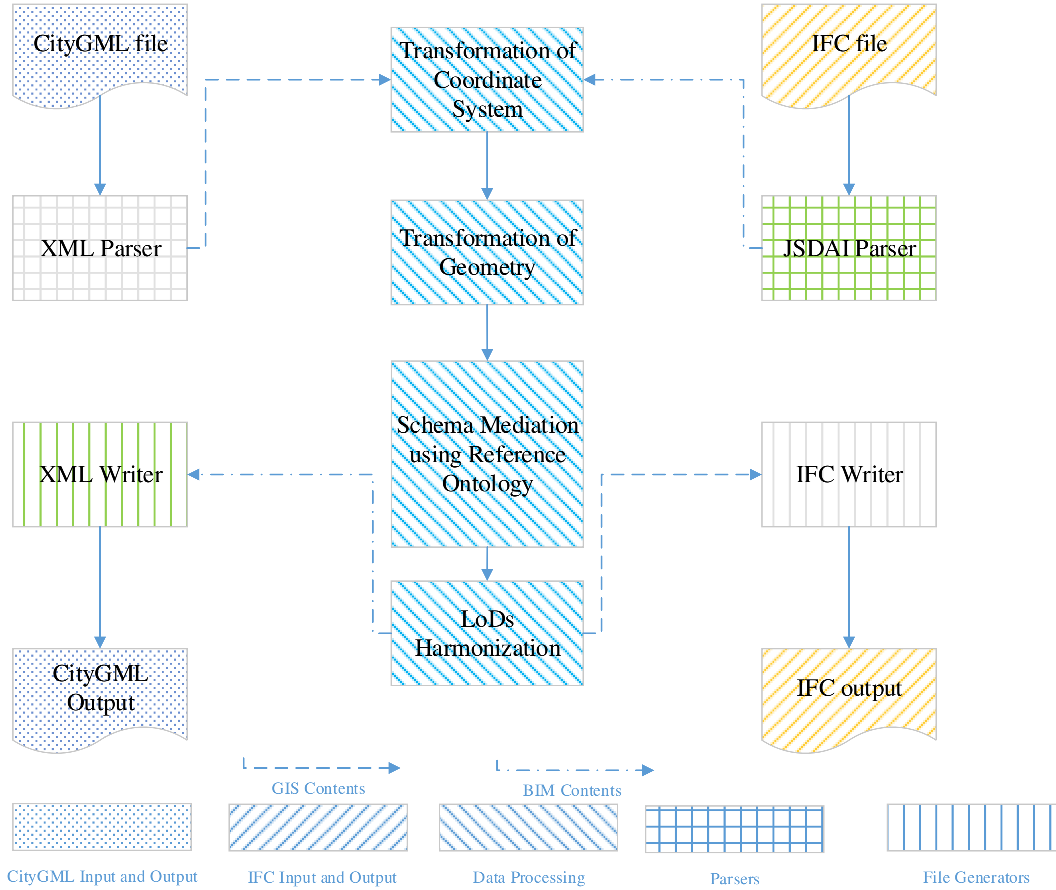


Fig. 4. Mapping process with medium ontology.

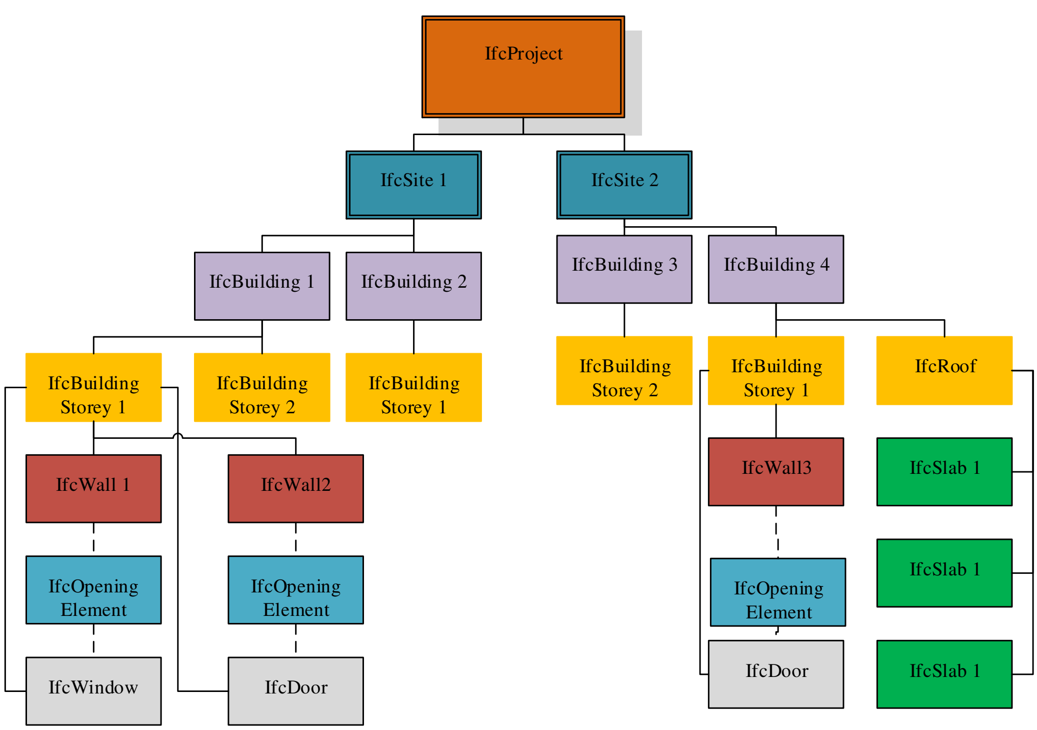


Fig. 5. The tree-like structure of components in BIM models.

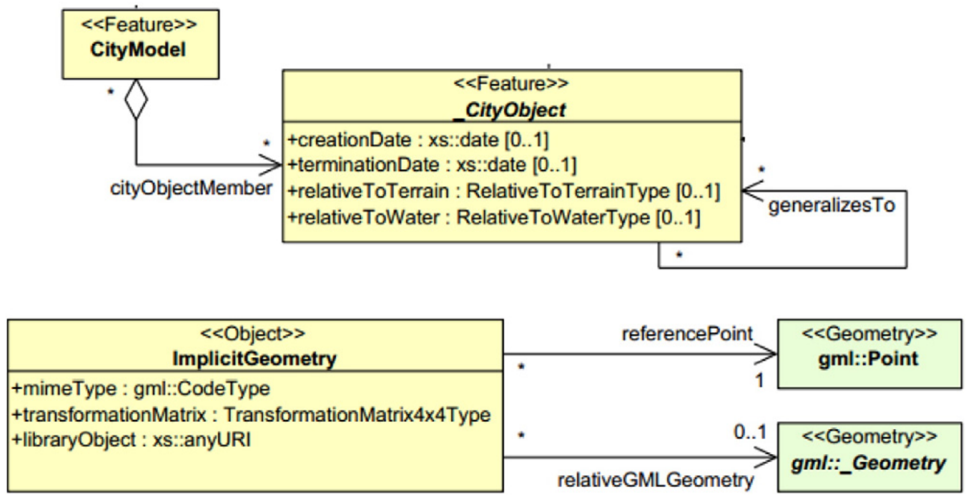


Fig. 6. Core module of CityGML [37].

LoD4, which models the inside details of buildings, as shown in Fig. 2. The LoD definitions provide more options for users. They provide different resolutions for different applications. According to Gröger et al. [21], a LoD0 model is simply a 2.5-dimensional digital terrain model, which is a two-dimensional map with a 3D terrain. LoD1 models are simple box models. LoD2 models add roof structure to LoD1 models. LoD3 models have detailed exterior features, such as openings and wall structures. LoD4 models are the most complex models with building interior features. Furthermore, CityGML supports application domain extensions (ADE) in which users can create their own extensions for their particular applications. Different LoDs and ADEs could broaden the application area of CityGML.

Both IFC and CityGML are component-based, in which the information about a single building component could be extracted separately, as demonstrated in Fig. 3. For example, the *IfcWallStandardCase* in Fig. 3 represents a wall in a BIM model in IFC. The related information about this wall, such as its location (*IfcLocalPlacement*), shape (*IfcProductDefinitionShape*), and other semantic information could be parsed and extracted and processed. This component-based mapping is the basis for the proposed instance-based mapping.

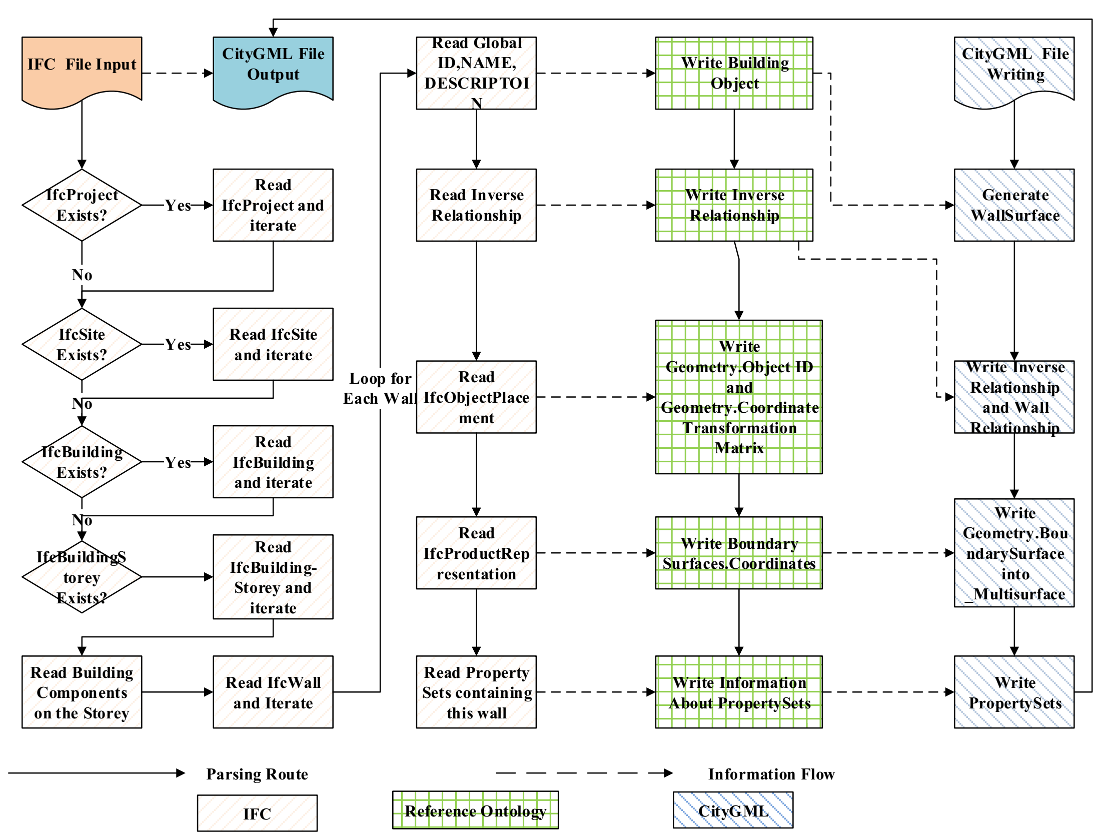


Fig. 7. Parsing route for mapping from IFC to CityGML.

```
IfcWindow(ID,  #owner_history  (a),  name,  description,   #local_placement  (b),
#representation (c), Height, width)
        #IfcOwnerHistory() (a)
        #IfcLocalPlacement(#place_related_to, #relative_placement) (b)
                #Ifcproduct()
                #ifcAxis2Placement3d(#location,#axis,#reference_direction)
                        #(x',y',z')
                        #(x,y,z)
                        #(a,b,c)
        #IfcRepresentation (c)
                #IfcPolyloop(P1,P2,P3,P4)
                        #(a1,a2,a3)
                        #(b1,b2,b3)
                        #(c1,c2,c3)
                        #(d1,d2,d3)
```

Fig. 8. Representation of a window in IFC.

2.2. Challenges of data integration between BIM and GIS

Since IFC is a schema in the BIM domain whereas CityGML is a schema in the GIS domain, the two schemas are designed for different purposes. The contents and data structure inside these two schemas are different. For instance, in IFC files, users could define contents such as construction scheduling, cost, and labor, while in CityGML files, these contents are unavailable. The IFC uses an EXPRESS schema, while CityGML uses an XML-based schema. Although buildingSMART also released the XML version of IFC, which is called the IFCXML, it is not as widely used as the EXPRESS-based IFC. In our research, the dynamic information such as scheduling or operation data in IFC files is neglected. However, the differences of contents and data structures of these two schemas make it hard to achieve a complete mapping between the two schemas. Four major challenges are identified in the early stage of this research.

The first challenge is the transformation of local placement system to world coordinate system. IFC uses a local placement system in *IfcLocalPlacement* to define the location of objects. The location of objects does not refer to the world coordinate system; instead, it points to a local placement system of a relevant object. For instance, the local placement system of a wall may refer to the placement system of a building, while the building refers to the site. In CityGML, however, all the objects are defined in a world coordinate system with absolute coordinates. The transformation of coordinate systems is needed during the mapping process.

The second challenge is the transformation of Swept Solid/CSG to BRep. IFC supports all three kinds of common geometric representations. Particularly, for cuboid shapes, such as walls and slabs, Swept Solid is commonly used since this may require smaller data storage. For complex shapes in IFC files, such as rectangular shapes, CSG is an efficient way to represent them. In CityGML, all the shapes are represented using BRep, so the transformation of Swept Solid/CSG to BRep should be developed for the mapping between IFC and CityGML.

The third challenge is the level of detail harmonization during the mapping process. CityGML supports five levels of detail (LoDs) to provide different resolution options for users. LoDs are also an efficient way to reduce rendering power and data storage. A complete mapping from IFC to CityGML requires the mapping of IFC to all the LoDs in CityGML. However, there are no complete transformation frameworks between LoDs in CityGML, so the harmonization of LoDs is another challenge for the complete mapping.

The final challenge is the data loss during the data integration process. As a BIM data standard, IFC is designed to capture almost all the information of the building, including owners, construction process, cost information, and operation and management. However, the focus of CityGML is to define the common GIS properties, such as location, coordinates, height, and construction time. In general, the information contained in BIM is much richer than that of GIS. Many entities in IFC could not find the matching entities in CityGML. For instance, the entity *IfcApplication*, which defines the information of the software for creating and managing the information of the building, could not map to any of the entities in CityGML. On the other hand, since CityGML also stores information related to cartography, such as world coordinate system for locating points. This information could not be stored in current BIM standards. The data loss is inevitable in the data integration process, so extensions of CityGML schemas must be defined in order to incorporate the additional information from IFC files.

2.3. Related works on IFC and CityGML mapping

The applications of mapping between CityGML and IFC have been discussed by several scholars. Hijazi et al. [14] used the mapping between CityGML utility extensions and IFC utility representations for geo-analysis. The mapping in the utility level focused more on the relationship between components in the utility network. The geometric representation of the network was not covered in detail. In an OGC test, Lapierre and Cote [22] reported that the integration between IFC

```
<bldg:opening>
<bldg:Window>
  <xbuilding:GlobalId value="D"/>
  <gen:doubleAttribute name="OverallWidth">
    <gen:value>Width</gen:value>
  </gen:doubleAttribute>
  <gen:doubleAttribute name="OverallHeight">
    <gen:value>Height</gen:value>
  </gen:doubleAttribute>
...
</bldg:opening>
```

```
<bldg:lod4MultiSurface>
<gml:MultiSurface>
  <gml:surfaceMember>
  <gml:Polygon>
    <gml:exterior>
      <gml:LinearRing>
        <gml:posList srsDimension="3">
        s1, s2, s3, t1, t2, t3, u1,
        u2, u3, s1, s2, s3
        </gml:posList>
      </gml:LinearRing>
    </gml:exterior>
...
</bldg.:lod4MultiSurface>
```

Fig. 9. Semantic (left) and geometric (right) representation of a window in CityGML.

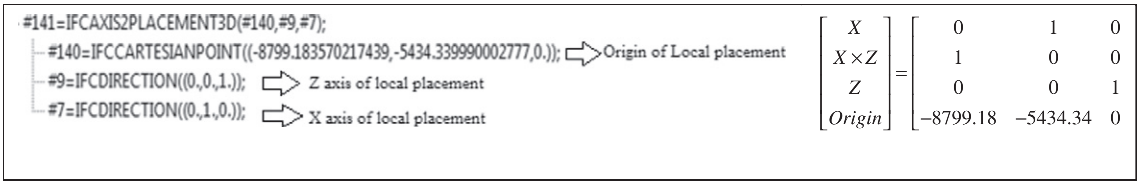


Fig. 10. Generating the transformation matrix M.

and CityGML can facilitate the emergency response in an urban scenario. The mapping between IFC and CityGML was also discussed in the report. However, only the room space was converted and the detailed building components were not discussed [23]. Döllner and Hagedorn [24] presented a 3D viewer for urban data in which the information from GIS, CAD (Computer Aided Design), and BIM were integrated. Irizarry et al. [25] developed a BIM–GIS integrated system to improve the visual monitoring of construction supply chain management. In the developed system, material demands were calculated from information retrieved from BIM, and the material delivery status was visualized in GIS. Semantic information such as time-of-services of subcontractors and construction schedule was used in the developed system. Liu and Issa [26] developed a system to integrate BIM and GIS for managing utilities. The 3D geometry of utilities from BIM and 2D GIS system were integrated to better visualize utility network. Moreover, the facilities maintenance records were also incorporated in the system to allow better data management. Park et al. [27] tried to conduct preliminary feasibility studies for national road projects using information from BIM and GIS. Information from BIM, such as repair cost or rehabilitation cost provides basis to estimate construction costs, land acquisition costs, and O&M costs. In these approaches, the mapping development was performed on a problem-driven basis, in which no completed mapping was achieved.

There have been several attempts to achieve complete mapping between IFC and GIS standards. Some of these mappings were done in the BIM domain, in which people tried to extend BIM data standards with the contents of GML, which is the basis for CityGML. One example of this approach is the IFC for GIS (IFG) project initiated by the Norwegian Strate Planning Authority Statens Bygningstekniske Etat. The IFG project aimed to provide geographic information within the framework of IFC, in order to provide more efficient planning and building plan submission [28]. The other approach focuses on extending GIS standards in order to contain information from BIM models. The GeoBIM extension in CityGML from the BIMServer project was a representative of this approach. The GeoBIM extension allows translation from IFC files to CityGML files by defining additional information in CityGML entities [29]. However, the two approaches only support unidirectional mapping. They can only allow translation from one format to another. This unidirectional way of mapping is not helpful for the integration between BIM and GIS systems. A bi-directional approach is needed. Moreover, data loss and huge data file generated are also reported in these approaches [29].

The integration between BIM and GIS systems has gained increasing attention these days. There are also several conceptual frameworks proposed to map the data standards in the two domains. EI-Mekawy et al. [30] merged IFC and CityGML in a Unified Building Model (UBM), in which entity definitions from the two schemas were extended according to the entity definitions in the two schemas. However, EI-Mekawy et al. [30] did not provide details for geometry transformation. The mapping of UBM to IFC and CityGML was not discussed. It seems that although UMB contained rich information, much of the information cannot be transformed to GIS models defined in CityGML. Isikdag and Zlatanova [31] proposed a framework to generate 3D CityGML models using IFC models. In the framework, the transformation of different LoDs and semantic information were discussed. However, in the proposed framework, the authors did not consider mapping rules for geometric and semantic information in IFC and CityGML. They also failed to mention the details of LoD transformation in CityGML. For example, given a building model in LoD4 in CityGML, their framework cannot translate it to lower LoDs. Nagel et al. [32] proposed conceptual requirements for generating 3D building models from un-interpreted 3D models using CityGML as the transformation medium. The authors developed mapping rules to allow transforming CityGML models to IFC model. However, the authors did not consider the semantic information mapping in the process. The developed mapping is also unidirectional which only allows transforming from GIS model to BIM. Xu et al. [33] developed a unified model schema called City Information Modeling (CIM), which contained five categories of entities, namely, the building, transportation, city furniture, MEP, and water body. The authors developed mapping rules between IFC, CityGML, and CIM to allow information exchange between IFC and CityGML. However, Xu et al. [33] did not provide details for the mapping; for example, the software engineering part is missing. Kang and Hong [34] proposed a software architecture for the effective integration of BIM into a GIS-based facilities management (FM) system using an Extract, Transform, and Load (BG-ETL) method. The authors also proposed transformation from BIM to different LoDs in CityGML, but the process is semi-automatic and manual input is required.

In summary, the limitations of previous studies are: firstly, there were no bidirectional translators between IFC and CityGML developed. Secondly, the mapping between IFC and different LoDs of CityGML was also not completed. Thirdly, the semantic information in IFC was partially translated during the mapping process as well. The objective of this study is to develop a framework that allows complete mapping between BIM and GIS models in different levels of detail. Since IFC and CityGML have limited overlapping in terms of information stored, the

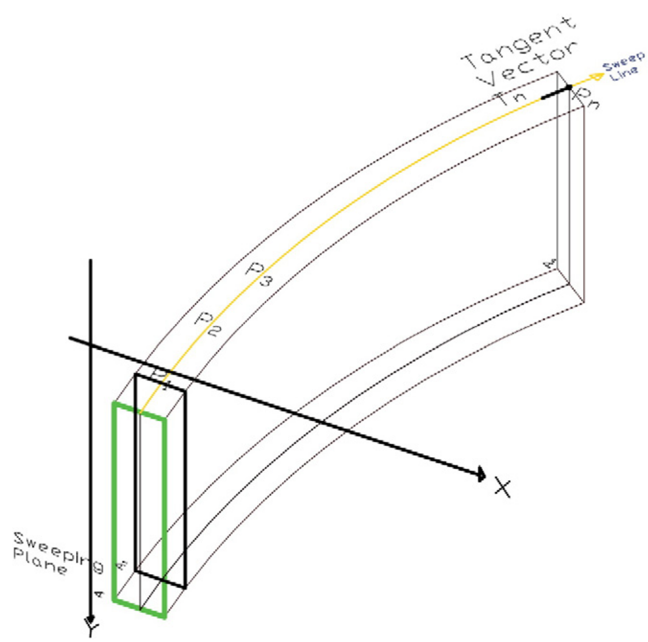


Fig. 11. The case of rotating sweeping plane.

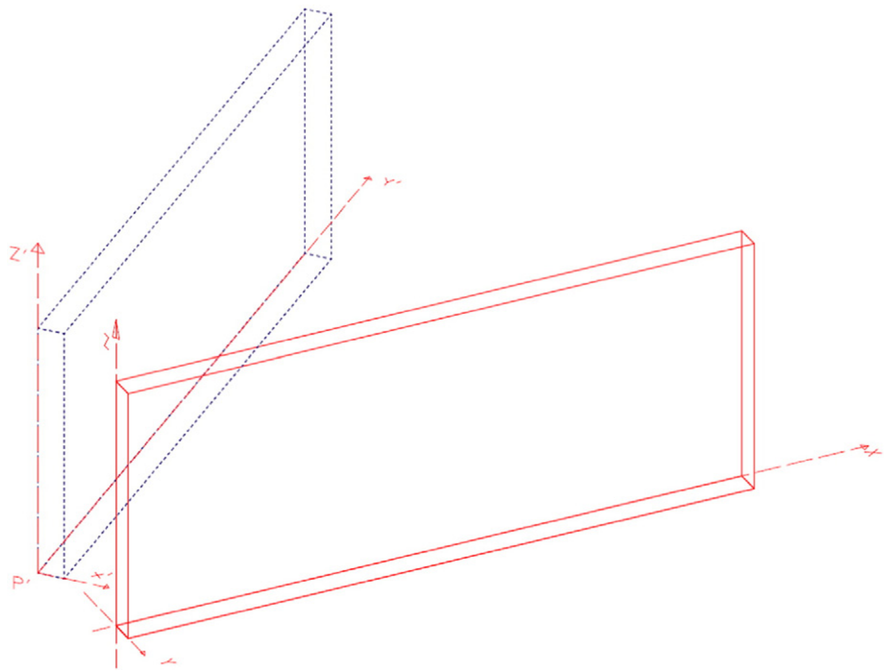


Fig. 12. Rotation of swept solids in different coordinate systems.

extensions to IFC and CityGML should be developed by implementing a reference ontology. The mapping rules between entities and semantic information in IFC and CityGML should be developed by an instance-based approach.

2.4. Related works of schema mapping using reference ontology

In computer science and information science, an ontology is the representation of knowledge by using a set of pre-defined concepts within a domain and the relationship between concepts. In theory, an ontology is a "formal, explicit specification of a shared conceptualization" [35]. Ontologies are widely used in information retrieval [36], schema development, and development of search engines [37,38]. When dealing with knowledge from multiple domains, a reference ontology could serve as a medium for carrying knowledge and thus facilitate interoperability between domains [39].

In the field of biomedical informatics, business intelligence and functional design, reference ontologies have been designed for achieving interoperability or data mapping between schemas. Rosse and Mejino Jr. [40] presented a reference ontology for biomedical informatics in anatomy. Anatomy taxonomy was divided into physical anatomical and non-physical anatomical entities and all the anatomy taxonomy terms could be mapped to this reference ontology. Andersson et al. [41] designed a reference ontology for business models based on three established business model ontologies: the REA, e3-value, and BMO. The authors stated that the developed ontology widened the scope of knowledge in the business models. Moreover, the relationship between entities in the original could be better understood by using the reference ontology. Kitamura et al. [39] tried to design a reference ontology for

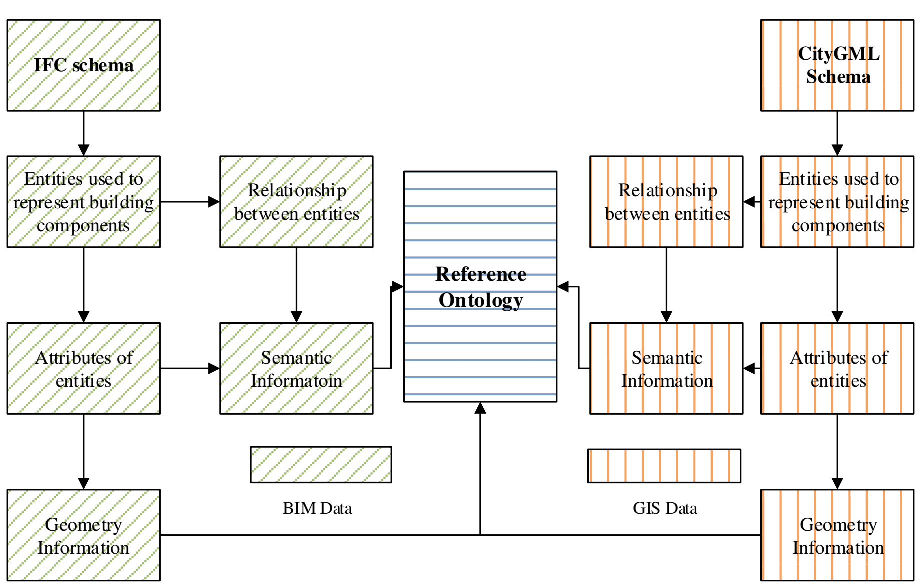


Fig. 13. The design of reference ontology.

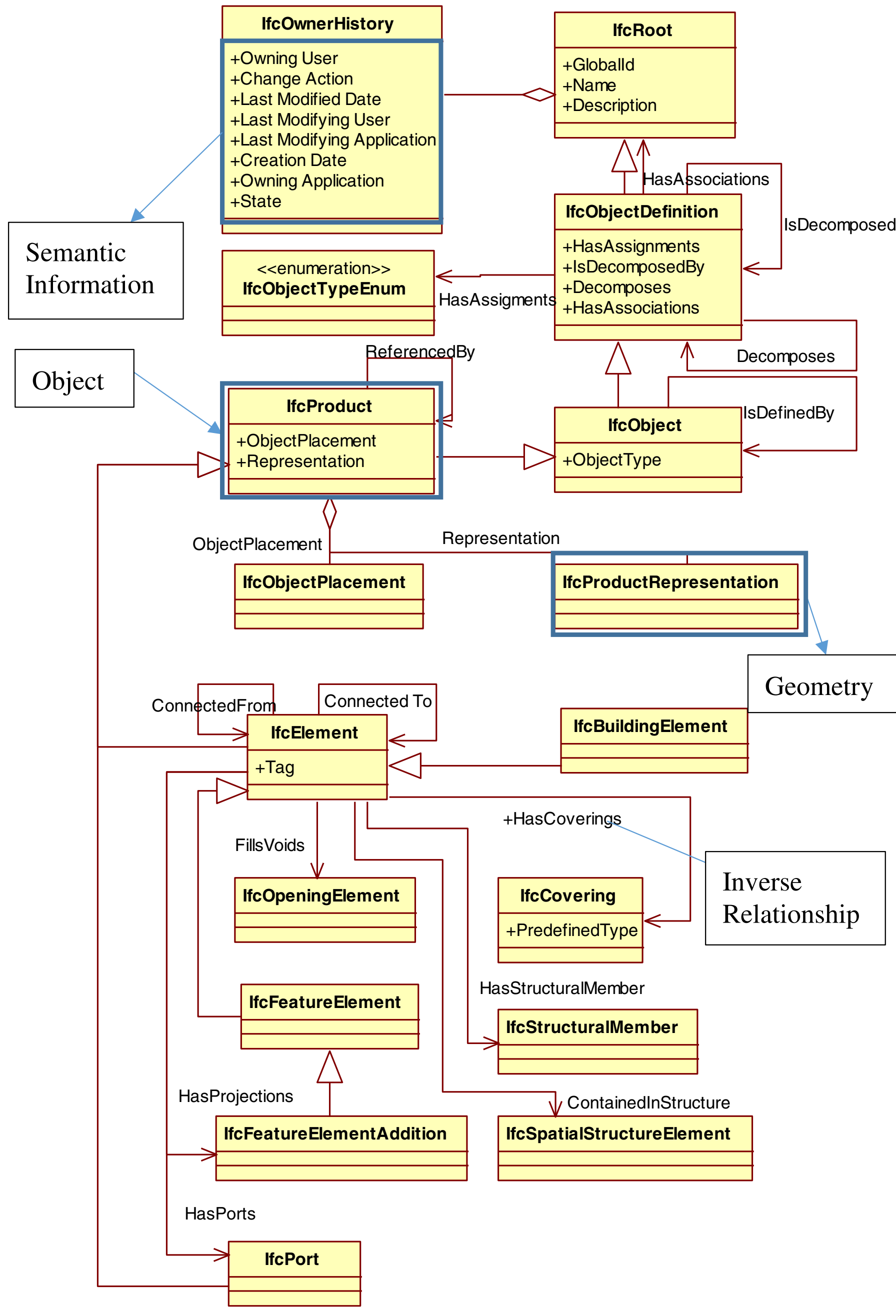


Fig. 14. UML diagram of IFC building elements.

functional knowldege interoperability. They studied different behaviors, functions, and contexts for the functional desgin and came up with a reference ontology that could better describe the functional design of people. These studies illustrated the use of reference ontologies to expand the scope of knowledge and its potential for cross-domain schema mappings.

In the AEC field, there have been several studies concerning the design of a reference ontology for schema mapping. The Unified Building Model (UBM) designed by El-Mekawy et al. [30] could be seen as one of the representative efforts in designing reference ontology for mapping between BIM models and GIS models. In the UBM, the authors defined the relationships between building components, such as linking walls to building storey, with reference to the BIM schema and the GIS schema. Although their approach only focused on the component level and not to the geometry and semantic information, the UBM could

Table 1
The entities for representing a window in IFC and CityGML.

Level	IFC Entities	CityGML Entities
Object	IfcWindow	Opening Window
Middle	IfcOwnerHistory IfcLocalPlacement IfcProductDefinitionShape IfcShapeRepresentation IfcGeometricRepresentationSubcontext IfcMappedItem IfcRepresentationMap IfcAxis2Placement3d IfcShapeRepresentation IfcGeometricSet	MultiSurface SurfaceMember Polygon
Value	IfcPolyline IfcCartesianPoint	LinearRing Exterior

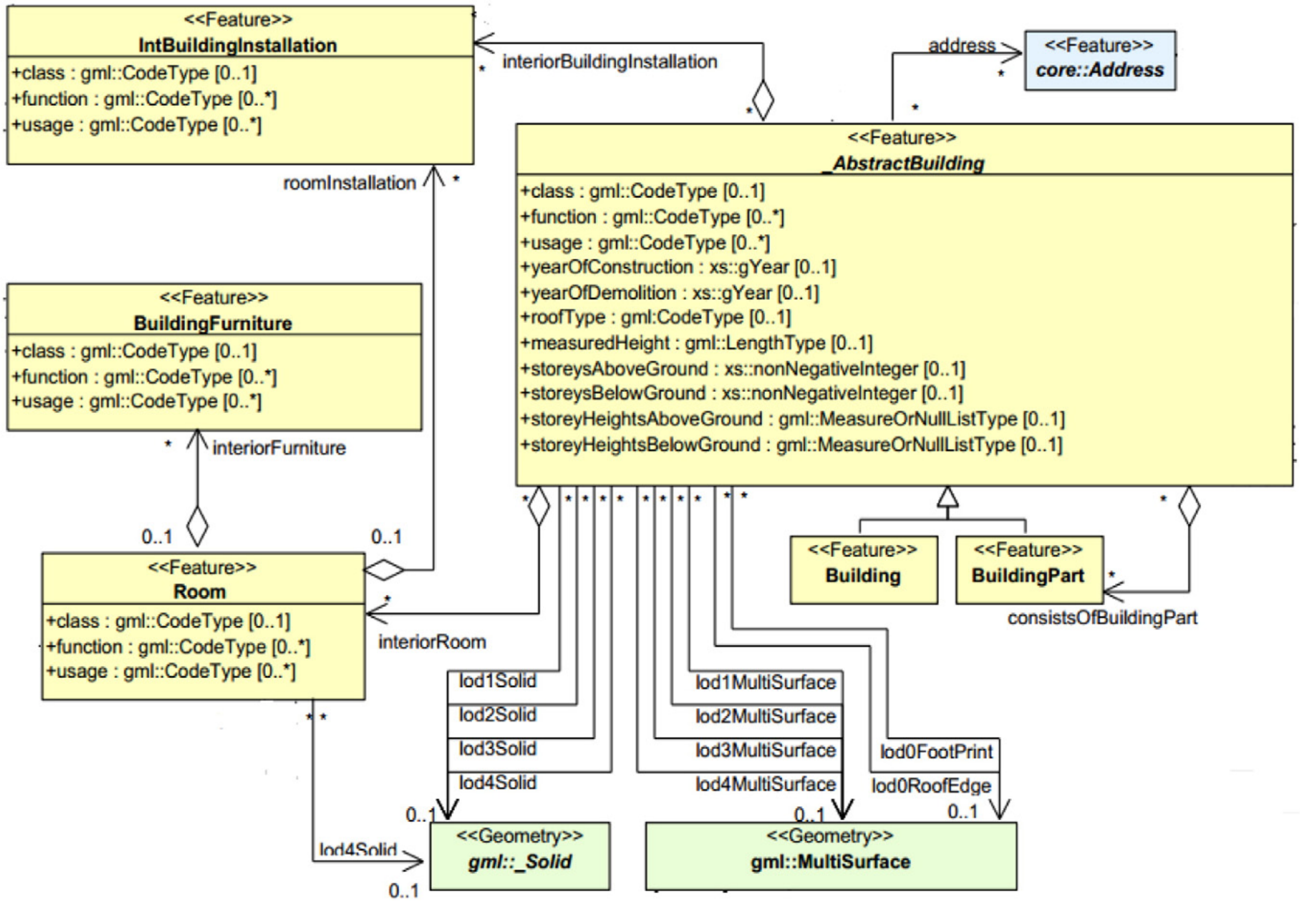


Fig. 15. UML diagram of CityGML building model [38].

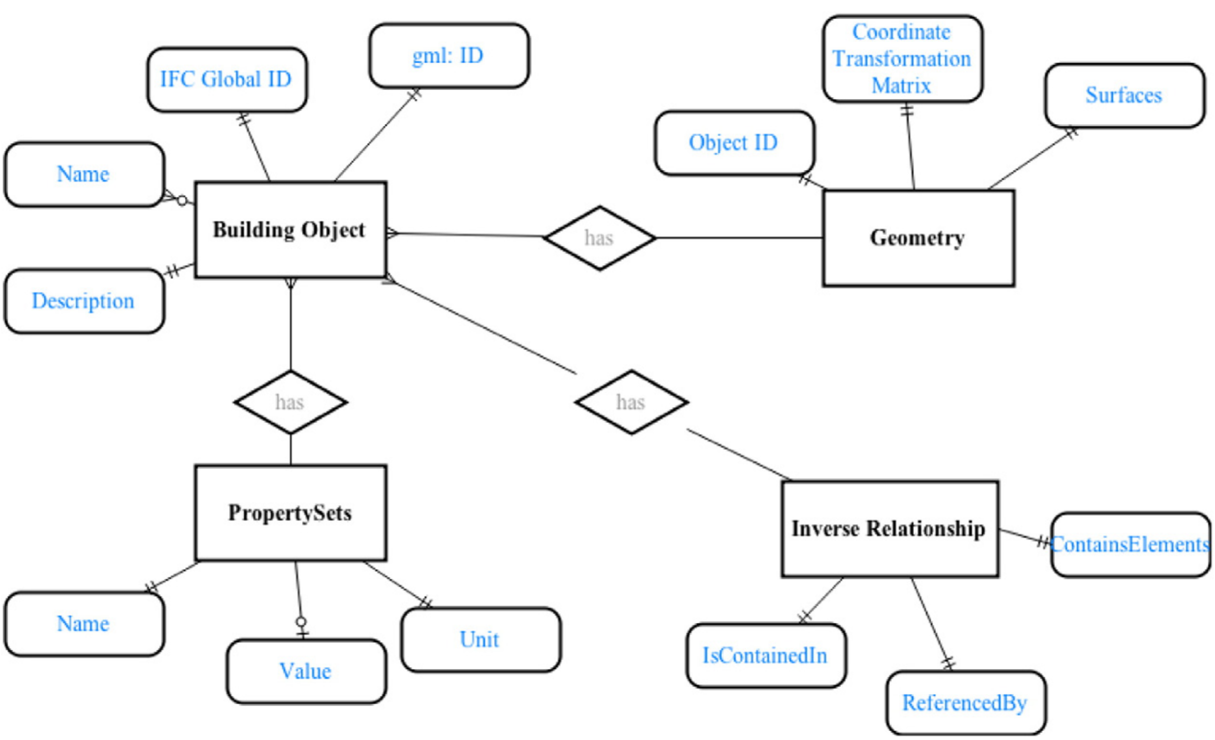


Fig. 16. The generic core of the developed reference ontology.

provide a clear relationship for the mapping between BIM models and GIS models.

2.5. Related works of schema mapping using instance-based method

While designing the reference ontology could achieve the mapping of entities in the schema level, the mapping rules between the values of entities should be developed by an instance-based method. For example, Herrlich et al. [42] tried to map CityGML with COLLADA, which is an XML-based schema to transport 3D assets between applications with purely geometric reprensentation of objects and excludes semantic information. During the process, as reported by Herrlich et al. [42], they must transfer the coordinates and values representing texture into the COLLADA standard with certain mapping rules.

The instance-based method could generate promising results for schema mapping with complex schema structures. It compares instances from both the origin and target schema and generates mapping rules between schemas accordingly. One instance-based mapping in the AEC field was reported by Lipman [43]. In this study, Lipman tried to map IFC with the steel production standard CIS/2 using component-based manual inspection. Both IFC and CIS/2 have complex schema structure, so Lipman attempted different entities and generated several mapping rules accordingly. Despite guaranteeing the mapping accuacy, the instance-based method could be time consuming. It also requires domain knowledge to achieve an accurate mapping. However, it is by far the most effective way for schema mapping between complex schemas.

3. Methodology

In this section, we proposed a methodology for complete mapping between IFC and CityGML. The proposed methodology will achieve the goal of bidirectional mapping between IFC and CityGML. A reference ontology called Semantic City Model, which carries and transforms the information from the two data standards, was developed as the core mapping module. With the reference ontology and the instance-based mapping rule generation, the complete and bidirectional mapping between IFC and CityGML could be achieved. The mapping process is shown in Fig. 4.

The parsers for both CityGML file and IFC file were built based on the JAVA platform. The parser for CityGML was CityGML4j developed by Nagel [21]. CityGML4j can parse CityGML files and generate an object-tree in the JAVA platform. Data inside the CityGML files can thus be accessed. The parser for IFC was the JSDAI, which is an application programming interface (API) for reading, writing, and runtime manipulation of object-oriented data defined by an EXPRESS-based data model [44]. By building IFC entity libraries using JSDAI, JAVA can also parse the IFC files into an object-tree structure. The CityGML4j and JSDAI were also used as the writer for CityGML and IFC files, respectively.

3.1. Parser strategy

In BIM models, building components are related to each other. For example, a window may be related to an opening in a wall; in this case, a “contained in” relationship may be assigned between the

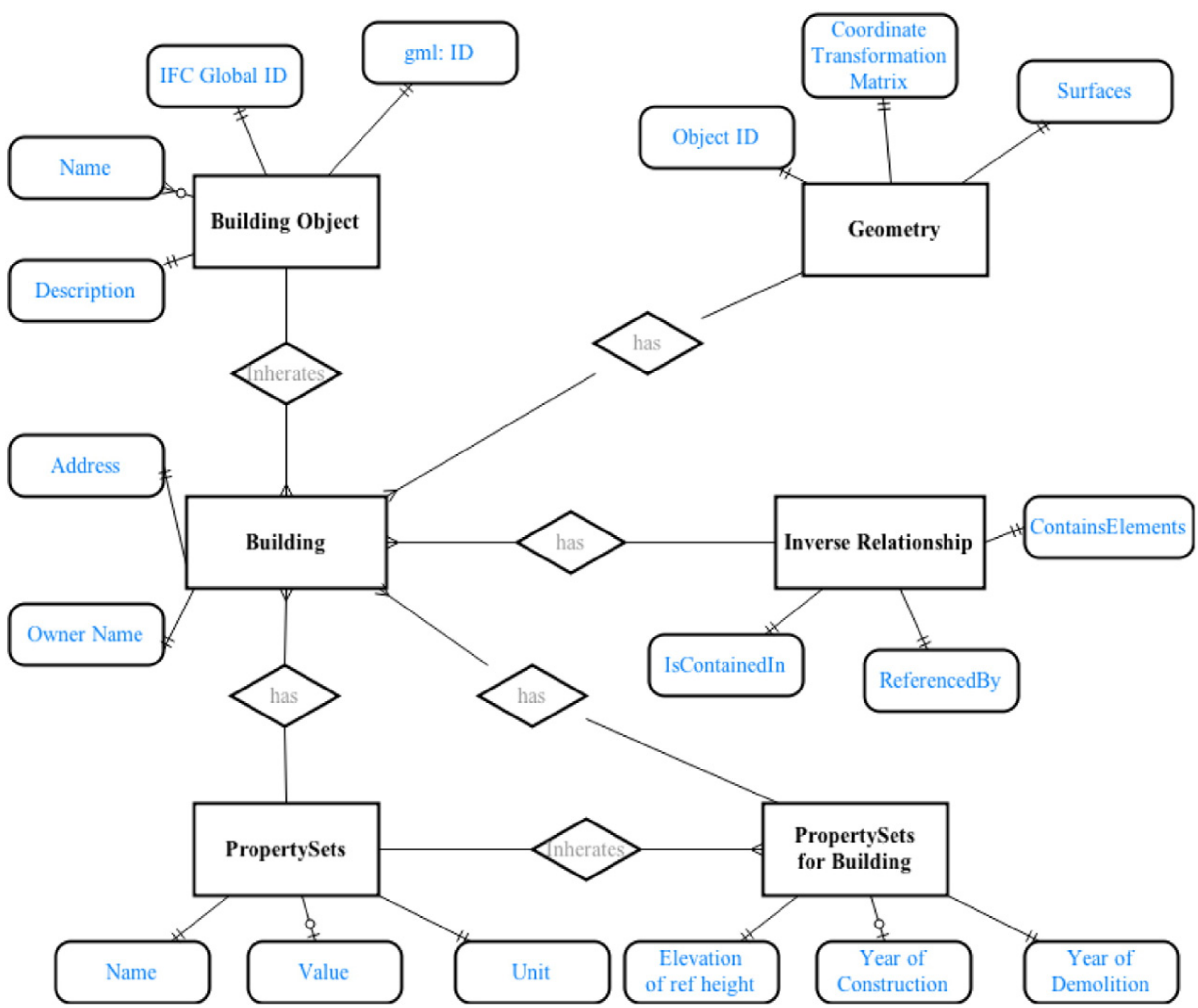


Fig. 17. The reference ontology developed for buildings (corresponding to *IfcBuilding* and *AbstractBuildingType*).

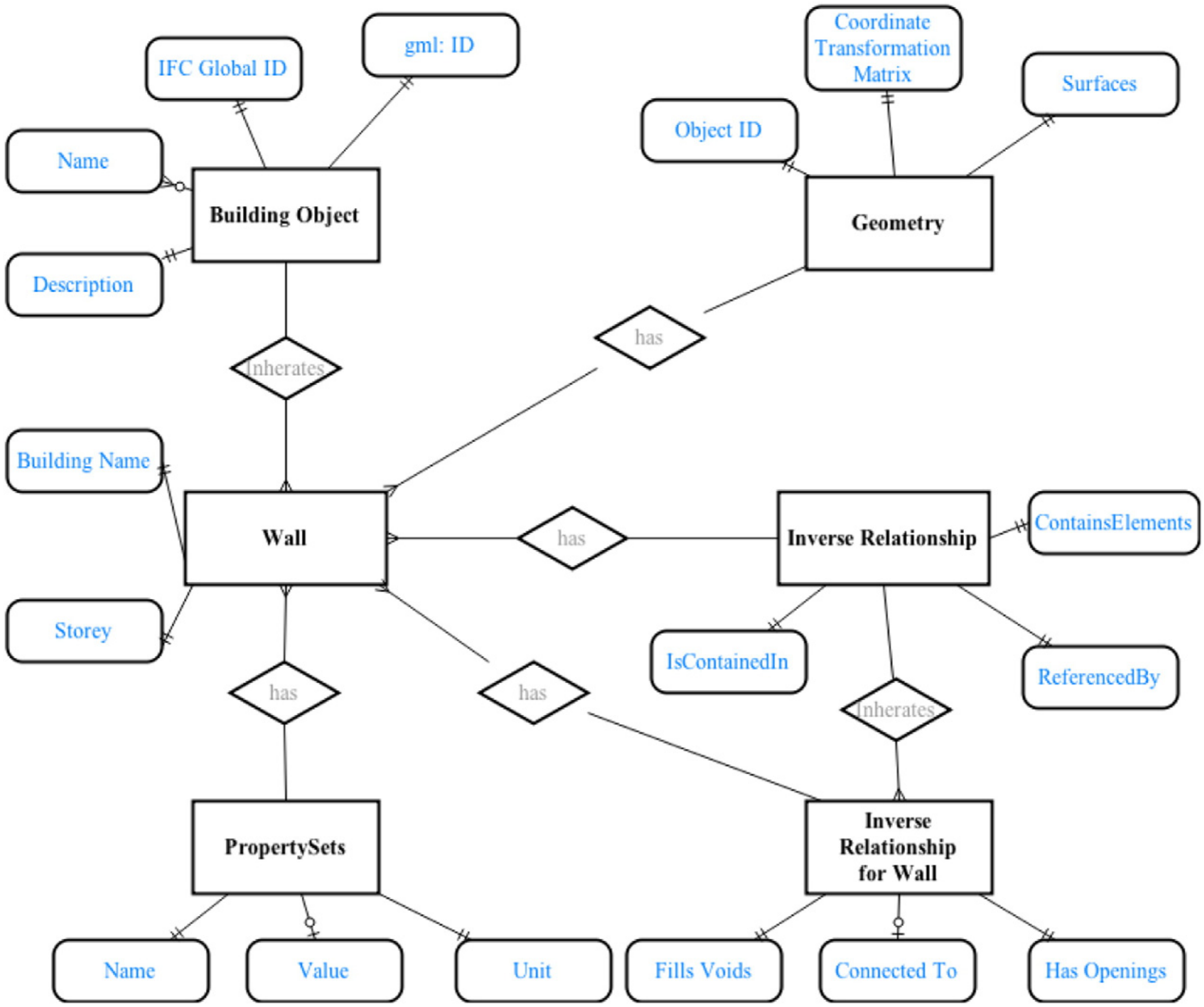


Fig. 18. The reference ontology developed for walls (corresponding to *IfcWall* and *WallSurface*).

window and the wall. As well as the geometric information, the BIM models also define the semantic information for building components. The relationship between building components is an important part of the semantic information. By reading these relationship definitions, a tree structure of the BIM models which defines the relationship between components can be generated. Fig. 5 shows a simple example of the relationship between components in a building project defined in IFC. For CityGML, the structure of entities is similar to that of IFC models. Fig. 6 shows the core module of CityGML. While parsing the CityGML files, the top level will be *_CityObject* entity.

The tree-like structures of IFC models and CityGML models indicate two possibilities for developing the parser: a top-down approach and a bottom-up approach. The top-down approach takes the root of the IFC and CityGML file first and finds its child entities while the bottom-up approach searches for the indivisible objects and starts building the tree from these leaves. The bottom-up approach is efficient for BIM models and GIS models with small amount of components as it neglects some unnecessary searches. However, for complex building models, the reconstruction time of the whole file tree may be too long because more than one relationship may exist between objects. So the parser strategy adopted in our framework is the top-down approach. The parser will find *IfcProject* and the related *IfcBuilding* by looking at these inverse-attributes, and then continue to go down until all the building components are visited. The process is illustrated by Fig. 7, taking transformation of wall components as an example. The solid lines show the parsing route and the dashed lines show the information flow.

Table 2
Mapping of information between IFC, reference ontology, and CityGML.

	IFC		Reference ontology		CityGML	
Component information	GlobeUID	1yl03kehn1dOa1Teyk6xq0	UID	IFC_1yl03kehn1dOa1Teyk6xq0	gml:ID	IFC_1yl03kehn1dOa1Teyk6xq0
	Name	Basic wall:exposed—115 mm Brk:155,276	Name	Basic wall:exposed—115 mm Brk:155,276	ADEName	Basic wall:exposed—115 mm Brk:155,276
Inverse relationship	Contained In	Ground floor level	Inverse ContainedIn	Ground floor level	ADEInverse ContainedIn	Ground floor level
Semantic information	Built In	Phase I	PropertySet	Built In Phase I	ADEPropertySet	Built In Phase I
	Component type	New construction	PropertySet	Component type new construction	ADEPropertySet	Component type new construction

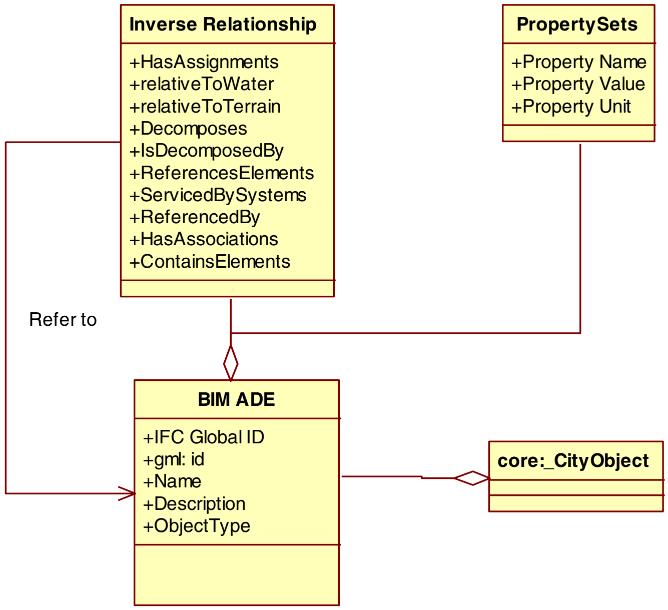


Fig. 19. CityGML semantic city model ADE core module.

Table 3
Definitions of each LoD in CityGML.

Classifications	Attributes	LoD1	LoD2	LoD3	LoD4
Building interiors	Room	×	×	×	√
	Opening • Window • Doors	×	×	√	√
	IntBuildingInstallation	×	×	×	√
	Building furniture	×	×	×	√
	Ceiling surface	×	×	×	√
	Interior wall surface	×	×	×	√
Building storey	Floor surface	×	×	√	√
	Roof surface	×	√	√	√
	Building installation	×	√	√	√
Exterior shell of buildings	Wall surface	×	√	√	√
	Ground surface	×	√	√	√
	Closure surface	×	√	√	√
	OuterCeilingSurface	×	√	√	√
	OuterFloorSurface	×	√	√	√
Massing	Geometry (GML)	×	√	√	√
	Solid (GML)	√	√	√	√
	MultiSurface (GML) • FootPrint/RoofEdge	√	√	√	√

3.2. Schema-based mapping and instance-based rule generation

The mapping between IFC and CityGML is performed on a schema level. The mapping of entities is generated based on the schemas of IFC and CityGML. For instance, the entity "*IfcWallStandardCase*" was mapped to "*WallSurface*" on the schema level. The data transformation rules are generated manually using instance-based component inspection. IFC and CityGML use different methods to represent objects, with a different schema structure. For example, in an IFC file segment representing a window in Fig. 8, the coordinates of the window is stored in an *IfcPolyLoop*, while in CityGML, the same window might be represented by a list of coordinates, as shown in Fig. 9. By inspecting different instances of components in IFC and CityGML, the mapping rules between data values in the schemas could be generated. Typically, two major transformations of data from IFC to CityGML are critical: (1) transformation of the local placement system in IFC to the world coordinate system in CityGML, and (2) conversion of geometry to boundary surfaces.

IFC employs a local placement system in which objects are defined in local coordinate systems. The local placement system is further referenced to other local placement systems. For example, the local placement system of a wall may refer to the local placement system of the building, while the building may refer to the world coordinate system. The local placement system ensures that each building component is uniquely defined and is easy for copying of information. For scenarios such as multiple similar walls in the same building storey, one might just have to change their local placement system and retain the other information. However, CityGML uses a world coordinate system in which

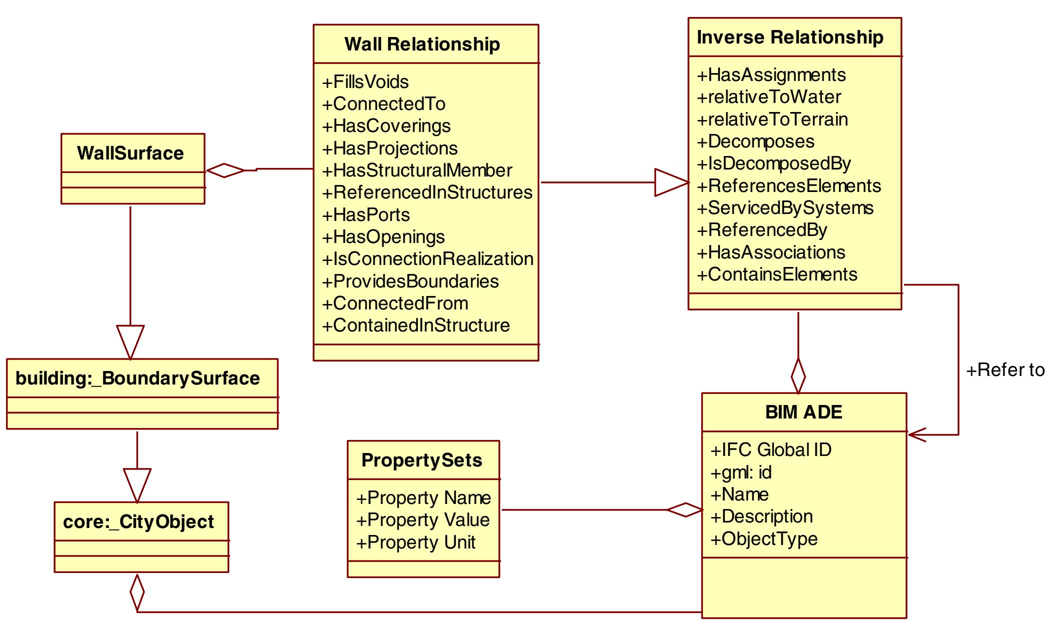


Fig. 20. CityGML semantic city model ADE for WallSurface.

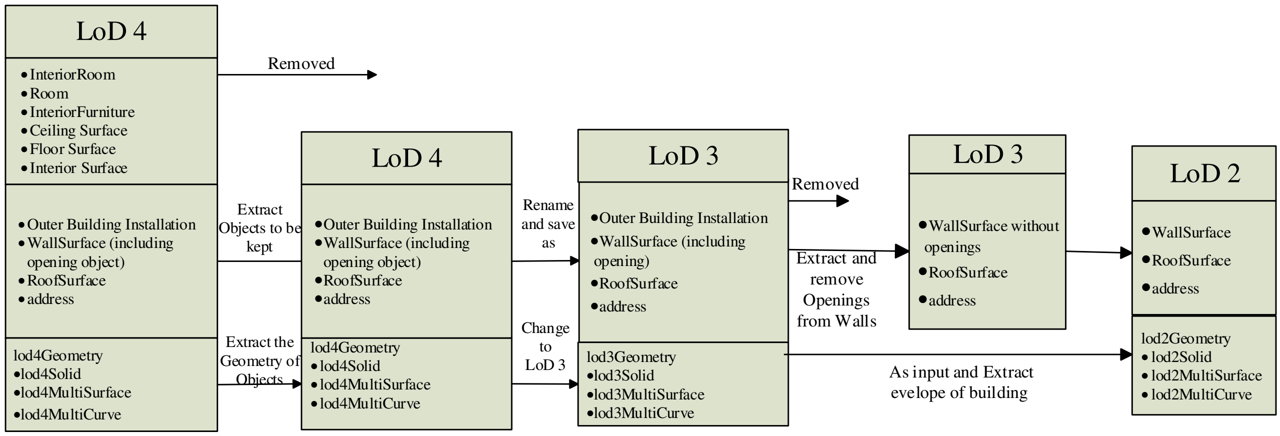


Fig. 21. The developed method of LoD harmonization for CityGML.

all the coordinate values of objects are absolute and do not refer to other objects. The transformation from a local placement system to a world coordinate system may be achieved by a method proposed by [45]:

$$\begin{bmatrix} C_x \\ C_y \\ C_z \end{bmatrix} = \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} \times M + \Delta \tag{1}$$

where vector C represents the coordinate in CityGML and vector I is the coordinate in IFC. Given the origin and axis orientation of the local coordinate system and its referencing system, the coordinate system transformation matrix M and the origin difference Δ can be calculated. The transformation matrix could be generated using methods specified in Fig. 10.

Another critical problem for the geometry transformation is the transformation between BRep and CSG/Swept Solid. IFC utilizes different methods to represent solids, such as storing their boundary surfaces (BRep), storing the cross-section of solids and the swept direction, or using combination of solids (CSG). CityGML uses only the BRep, in which all the objects are represented by surfaces. In this study, the IFC BRep to CityGML BRep and IFC Swept Solid to CityGML BRep were developed for data mapping. The IFC BRep to CityGML BRep function extracts the coordinates of the IFC BRep from the parser and translates the coordinates to world coordinate system in CityGML using the coordinate system transformation function. The IFC Swept Solid to CityGML BRep transformation is the most commonly seen geometric transformation. A function is developed to use the coordinates from the sweeping planning and the sweeping line to generate new BRep surfaces. For instance, in Fig. 1 (b), given coordinates $A = (X_A, Y_A, Z_A)$ and $B = (X_B, Y_B, Z_B)$ and the sweeping vector $V = (X_V, Y_V, Z_V)$, the corresponding points after sweeping are

$$\begin{aligned} A' &= A + V \\ B' &= B + V \end{aligned} \tag{2}$$

and a new BRep surface $AA'B'B$ induced from the sweeping would be generated accordingly. The new BRep surface could be transformed to CityGML BRep directly. The rotation of objects in the extrusion process is also considered. Both the rotation of the sweeping plane and the swept solid are considered in this transformation algorithm. In the first case, the sweeping line is a curve, which is represented as a series of connected points in IFC, as shown in Fig. 11.

$$CURVE = \{P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), P_3(X_3, Y_3, Z_3) \ldots P_n(X_n, Y_n, Z_n)\}. \tag{3}$$

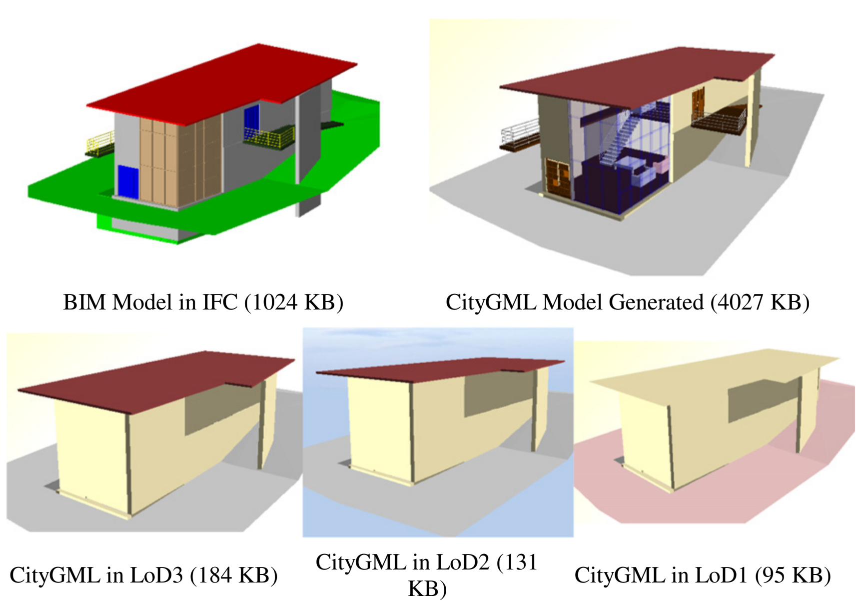


Fig. 22. Test results of the mapping framework.

Define T_n as the tangent vector of curve on point P_n, as shown in Fig. 11.

$$\mathrm{T_n} = (\mathrm{X}_n - \mathrm{X}_{n\text{-}1}, \mathrm{Y}_n - \mathrm{Y}_{n\text{-}1}, \mathrm{Z}_n - \mathrm{Z}_{n\text{-}1}) \tag{4}$$

According to the IFC2x3 documentation about swept solids, given a coordinate $A = (X_A, Y_A, Z_A, 0)$ on the boundary of the sweeping plane, the corresponding coordinate on the approximated curved surface for A is calculated as

$$\mathrm{A_n} = \mathrm{A} \times \mathrm{M_n} \tag{5}$$

where M_n is constructed from T_n. Define N as the normal of the surface containing the curve,

$$\mathrm{M_n} = \begin{bmatrix} N & 0 \\ N \times T_n & 0 \\ T_n & 0 \\ P_n & 1 \end{bmatrix}^{-1} \tag{6}$$

In this way, the swept solid to BRep transformation algorithm can transform swept solids with curved sweeping lines.

In the second case, the rotations of coordinate systems are needed, as shown in Fig. 12. In IFC swept solids are commonly defined in local placement coordinate systems. The rotation matrix, as defined in Eq. (7), can be used to transform coordinates in one local placement coordinate system to coordinates in the global coordinate system.

$$M_r = \begin{bmatrix} X' & 0 \\ Y' & 0 \\ Z' & 0 \\ P' & 1 \end{bmatrix} \tag{7}$$

where X' Y' $Z'$are vectors defining the local placement coordinate system and P' is the origin. Any point $B(X_B, Y_B, Z_B, 0)$ in the rotated object has a mapping with the corresponding point $B'(X_{B'}, Y_{B'}, Z_{B'}, 0)$ on the original object, as follows

$$B = B' \times M_r^{-1} \tag{8}$$

For complex shape in CSG, open source computational geometry library VTK is used to transform CSG to BRep.

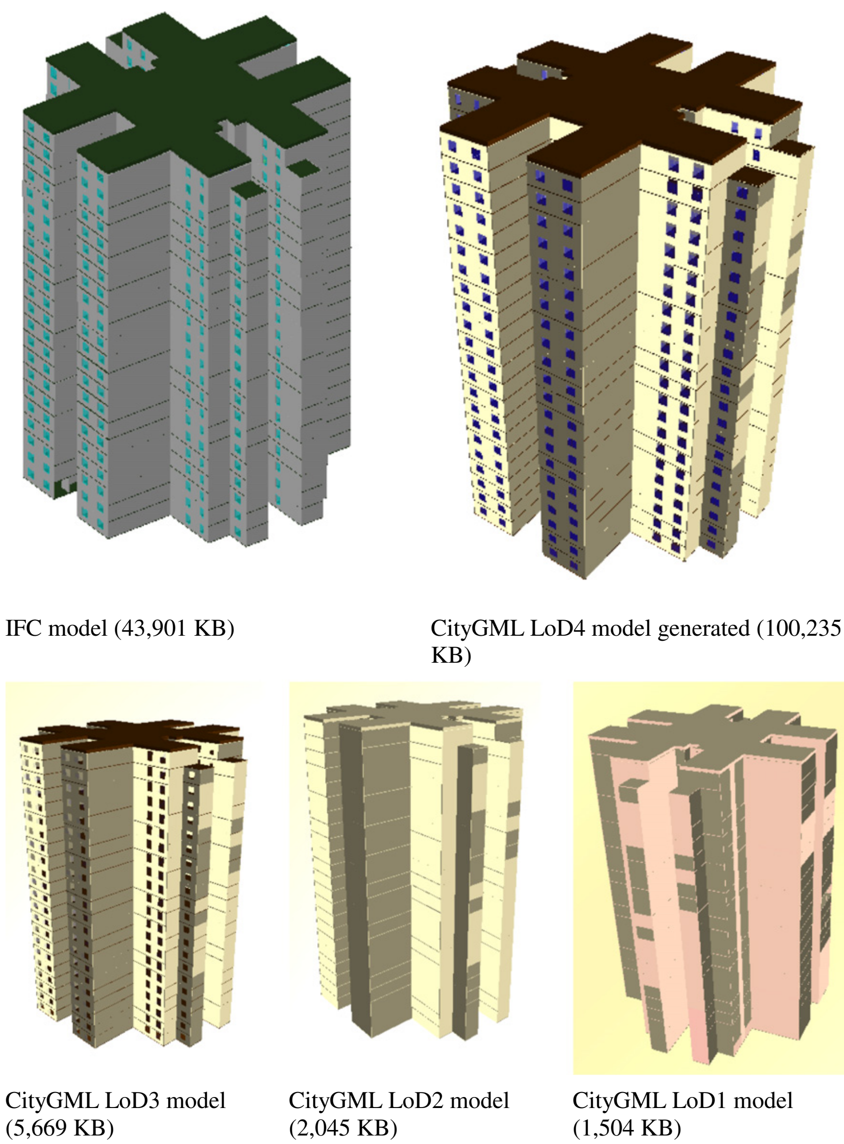


Fig. 23. Results of transforming a huge building model.

3.3. Design of reference ontology and CityGML ADE for scheme mediation

A reference ontology for mapping between IFC and CityGML models is defined as a superset ontology that contains all the entities and attributes from building models of IFC and CityGML. A superset ontology contains all the information from the schemas of IFC and CityGML. The mapping process is divided into two phases: (1) mapping from origin schema (IFC or CityGML) to the reference ontology and (2) mapping from the reference ontology to the target schema. Herein, the reference ontology serves as a medium for information exchange.

The Unified Modeling Language (UML) is chosen to be the modeling language for the reference ontology due to its wide acceptance. The reference ontology tries to capture all the information carried in the IFC and CityGML. First, the schemas of IFC and CityGML were studied in order to establish which entities will be involved in the mapping process; second, the attributes of these entities will be studied in order to extract the information needed to be captured in the reference ontology. Then the reference ontology was designed according to entities involved and their relationships. In the extraction of semantic information from IFC schema, special attention was paid to the inverse relationships between entities. These inverse attributes in IFC, such as "contained in structure" or "fills voids", defines the relationship between building components. They are important information contained in the BIM models which has not been studied by previous research efforts on the mapping between BIM schemas and other data schema. The design process of the reference ontology is illustrated in Fig. 13. To better understand the design of the reference ontology, studies about the schemas of IFC and CityGML will be firstly discussed.

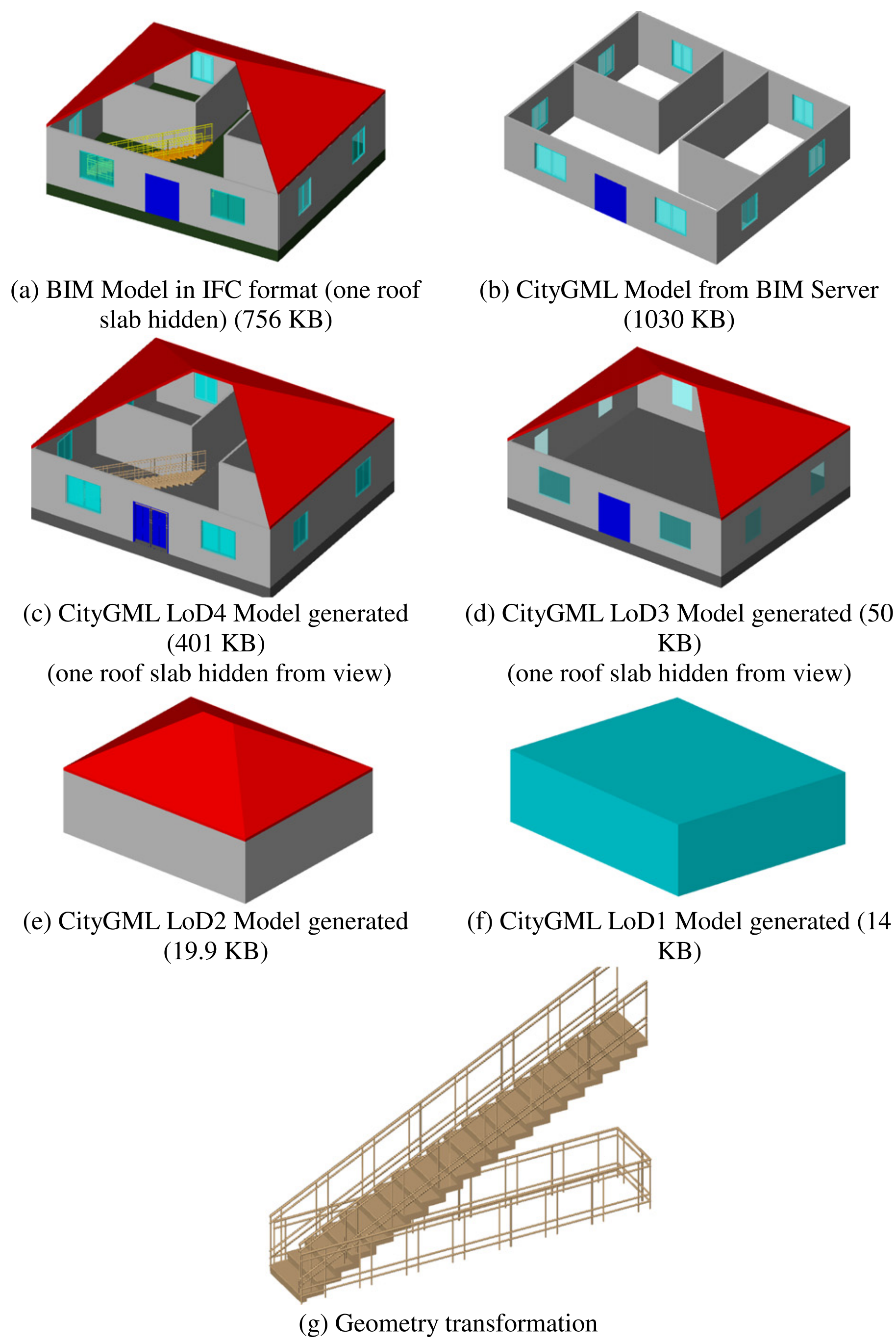


Fig. 24. Comparison with results from BIM Server, as well as demonstration of LoDs transformation.

Table 4
Components in each LoD.

Components	LoD1	LoD2	LoD3	LoD4
Room	0	0	0	3
Opening				
• Window	0	0	9	9
• Doors				
IntBuildingInstallation	0	0	0	2
Interior wall surface	0	0	0	8
Floor surface	0	0	0	1
Roof surface	0	4	4	4
Wall surface	0	4	4	8
OuterFloorSurface	0	1	1	1

3.3.1. Schema of IFC for building models

Ontologies represent knowledge by defining concepts and their relationships. For the purpose of designing the reference ontology, the IFC schema was firstly studied in order to understand what concepts are involved and how to define the relationship between these concepts.

As shown in Fig. 3, the IFC entities concerning a building component could be expanded to a tree-like structure in which the object values are linked to these objects. For example, the geometry shape of the *IfcWall* "*IfcRepresentation*" is linked to the wall in the IFC file. The inverse attributes in IFC links the building components to each other. For instance, an *IfcWall* may be linked to the *IfcBuildingStorey* using the inverse attribute "contained in structure". The *IfcWall* itself may be referred to by the *IfcWindow* by the inverse attribute "fills voids". To better understand the structure of the IFC schema, a UML diagram of the general IFC building entity is shown in Fig. 14. It is clear that the relationship between entities is more complex than the XML-based schemas. Entities in IFC could be linked to other entities by not only inheritance relationship, but also inverse relationships. One entity could also be linked to itself. For example, one *IfcObjectDefinition* could also be linked to itself by the inverse relationship "is decomposed by". For each building component, two concepts are important for representation—the associated building objects and the inverse relationships.

In the attributes defining the building components, the entities involved in IFC could be divided into three levels: the object level, the middle level, and the value level as shown in Table 1. The object level defines the nature of building components, such as *IfcWall* and *IfcCurtainWall*. For example, in Fig. 8, the entity *IfcWindow* is on the object level to define a real world window. The middle level links the real-world objects to its values, such as the lines to define its shape and the semantic information of the objects. For example, in Fig. 8, the *IfcRepresentation* entity links the coordinates (the value level *IfcPolyLoop*) to the real world window. In terms of attributes of IFC entities, four concepts could be generated: the objects, the semantic information, inverse relationships, and the geometry information (see Fig. 14).

3.3.2. Schema of CityGML for building models

The CityGML is an XML-based schema which defines geometry and semantic information of the building objects in 3D GIS. The entities of CityGML could also be divided into three levels, as shown in Table 1. CityGML also defines some inverse relationships. For example, the building entity is bounded by several boundary surfaces (e.g. walls,

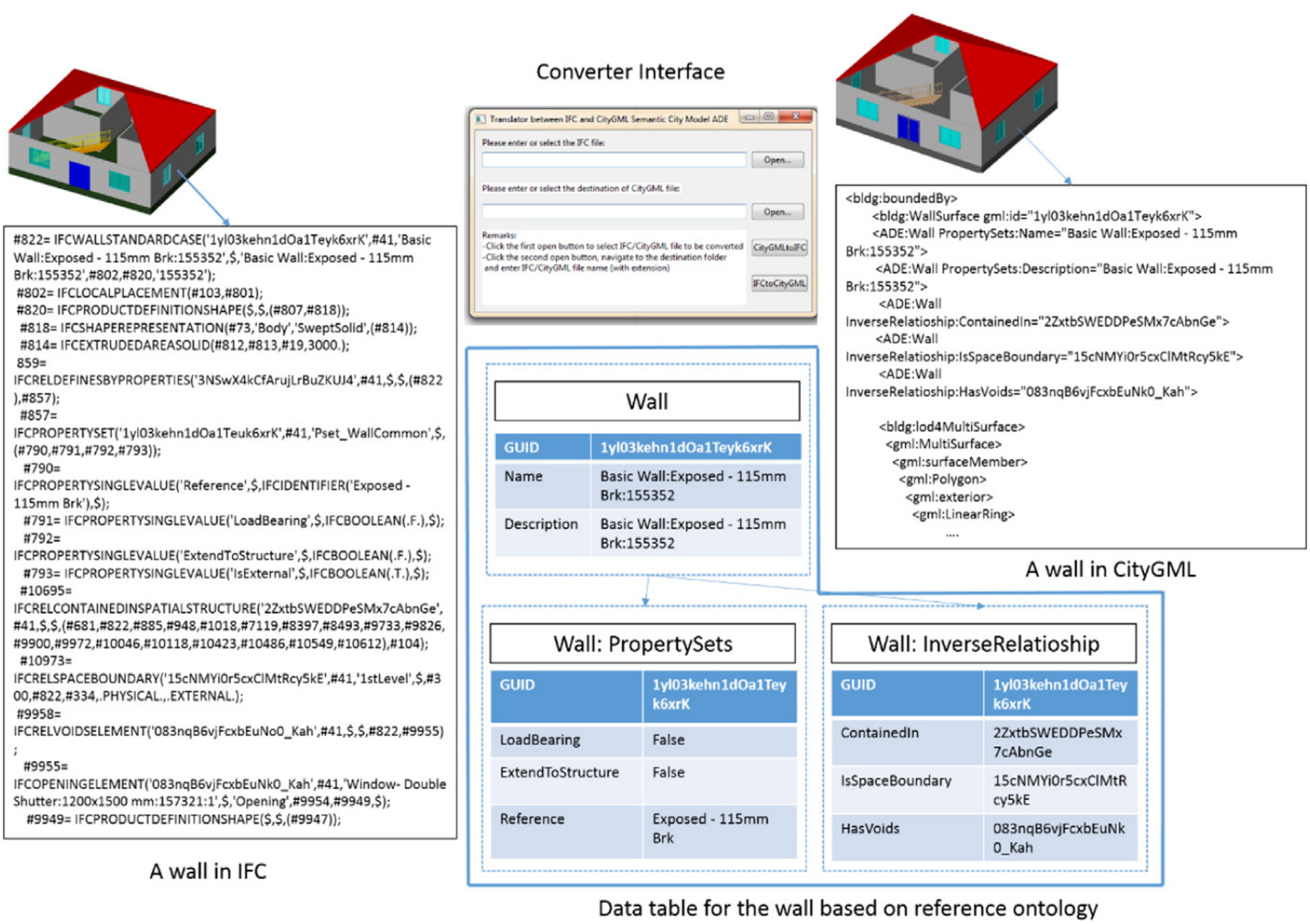


Fig. 25. The storage of semantic information in CityGML and data tables.

Table 5
Comparison with model generated from BIM Server.

Entities	Model from BIM Server	Model from our framework
LandUse	0	1
WallSurface	8	8
RoofSurface	2	4
FloorSurface	0	1
Room	1	3
IntBuildingInstallation	0	2
Stair	0	1
Railing	0	1
Semantic information	×	√
Load bearing	×	√
Structure usage	×	√
Room bounding	×	√

roofs). From the UML diagram of CityGML building model, as shown in Fig. 15, four concepts could be generated: (1) the objects, such as buildings, walls, roofs, and building installations; (2) the geometry of objects, such as surfaces, and coordinates, lines; (3) the semantic information of objects, such as year of construction, address, and location; and (4) the inverse relationship between objects, such as “bounded by” and “consist of building parts”.

Both IFC and CityGML have shared these four similar concepts: the object, geometry, semantic information, and inverse relationships. These concepts could cover all the knowledge in the IFC and CityGML schemas, so they could form the basis for the development of the reference ontology.

3.3.3. Development of a reference ontology

Following the efforts to study both schemas of IFC and CityGML, the major concepts and their relationships can now be extracted. We developed the core representation of the reference ontology and implemented it on different building components.

Fig. 16 shows the core of the reference ontology, which is composed of four basic concepts: (1) building object, (2) geometry, (3) property sets, and (4) inverse relationships.

1. A building object entity is the identifier to the objects in IFC and CityGML. It defines the ID, name, description and object type of building components. For example, both *IfcWall* from IFC and *WallSurface* from CityGML correspond to the same object type “wall”.
2. A geometry entity defines the shape of building objects, referring to them by object IDs. Since CityGML only supports BRep, all the shape representations from IFC and CityGML will be converted to BRep, which is stored in a sub-concept of geometry called boundary surfaces.
3. A property set stores and organizes semantic information in IFC and CityGML. Since there are various kinds of semantic information in IFC, some of which are defined by users and not pre-defined, we only consider the general value of semantic information: its name, its value, and its units. In use cases, this concept could be applied to entities such as “cost” with the value of “1000” and with the unit of “US dollar”.
4. An inverse relationship entity defines the relationship between building components. The root concepts in this inverse relationship are generated from the *IfcRoot*, which is the root entity for the IFC schema. Some examples of these concepts are “referenced by” and “is decomposed of”.

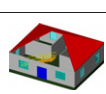
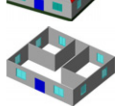
When implementing the reference ontology for schema mapping between IFC and CityGML, the concepts stated in the core of reference ontology could be expanded to meet the information requirements of certain building components. All the building components in the reference ontology instantiate the core ontology. Fig. 17 shows the expanded reference ontology for the “building” concept in IFC and CityGML, corresponding to *IfcBuilding* and *AbstractBuildingType*, respectively. In the semantic information defined in property sets, more concepts are introduced for the building type, such as year of construction. For the concept of walls, corresponding to *IfcWall* in IFC and *WallSurfaces* in CityGML, the reference ontology could be expanded as shown in Fig. 18. More inverse relationships could be added to the reference ontology, which are specified in the IFC schema concerning the wall components. The instantiation of the core ontology to building components is performed to represent all of the building components that are presented in IFC and CityGML. Nineteen of the components directly correspond to the IFC entities and CityGML entities, such as wall, roof, floor, curtain wall, window, and slab. Four of the components correspond to a group of entities in IFC. For example, the element “MEP” corresponds to all the components in the HVAC domain in IFC, such as *IfcDistributionElement*, *IfcFlowController*, and *IfcFlowSegment*. By defining these 23 elements, we could cover all the building components in BIM, except the dynamic information such as scheduling information.

The reference ontology is built within our Java program. A Java class is generated for each of the entities in the reference ontology. The IFC entities and CityGML entities are also transformed to Java classes using JASDAI and JDOM, respectively. Mapping rules between IFC entity and reference ontology entity and reference ontology and CityGML entities are resented in other Java classes that carry mapping rules. The mapping rules are generated from instance inspection. As shown in Table 2, the information in IFC is first mapped to the reference ontology and then translated to CityGML. For instance, the semantic information in IFC “Built in Phase I” is mapped to the reference ontology and then mapped the CityGML ADE we developed.

3.3.4. Development of the semantic city model ADE for CityGML

From the comparison of UML diagrams of buildings in IFC and CityGML shown in Figs. 14 and 15, it could be concluded that IFC schema contains much more information than CityGML, especially in terms of semantic information and inverse relationships. So the data loss from IFC models to CityGML models is inevitable since some of the information is not defined in the schema of CityGML, which indicates that an extension to the CityGML schema is needed. For the seamless data

Table 6
Comparison of the developed framework with other studies.

	Geometry transformation	Bi-directional mapping	Levels of detail	Extensions to schemas	Semantic information	
					Properties	Relationship
Our approach		Allowed	Considered	Reference ontology and extended CityGML	Full sets from IFC and CityGML	Full relationship from IFC and CityGML
BIM Server (van Berlo and de Laat [29])		Only from IFC to CityGML	Not considered	Geo-BIM extension	Only considered “name”	Not considered
El-Mekawy et al. [30]	Not mentioned	Allowed	Considered	Not considered	Not considered	Not considered
Isikdag and Zlatanova [31]	Not mentioned	Only from IFC to CityGML	Considered	Not considered	Not considered	Not considered

integration between IFC and CityGML, the schema of CityGML must be extended using ADE.

The reference ontology developed in this project and described in Section 3.3.3 is the basis for developing the Semantic City Model ADE. The core module of the Semantic City Model ADE is shown in Fig. 19. More information such as inverse relationships and semantic information are added to the CityGML schema. The information to be added follows the concepts of the reference ontology developed. For instance, when deciding which information to be added to *WallSurface* in CityGML, the concepts in the expanded reference ontology for walls are added, as shown in Fig. 20.

4. Transformation of levels of details in 3D GIS models

The concept of level of detail has been studied in both the BIM domain and the GIS domain. The LoD definitions of BIM domain are based either on geometry or information requirements of different stages of building life cycle. Leite et al. [46] defined three geometric levels of detail for BIM, namely, the approximate geometry, precise geometry, and fabrication. The authors evaluated the modeling effort associated with generating BIM at different LoDs and the impact of LoD in a project in supporting mechanical, electrical, and plumbing (MEP) design coordination. The National Guidelines for Digital Modelling in Australia defined five LoDs with respect to the different stages of buildings [47]. While the LoDs in BIM focuses more on the information requirements at different stages of projects and the modeling efforts to achieve certain LoDs, the LoDs in GIS are defined according to geometry details in order to support different applications. In this research, we focus on the transformation of levels of details in GIS models in order to enhance the functionality of GIS models transformed from BIM. CityGML supports five different LoDs in order to provide users with different options of resolution and reduce the computations for rendering. We start from LoD4 CityGML models generated from BIM, and define transformation algorithms from LoD4 to lower LoDs.

4.1. Definitions of each LoD

The current version of CityGML schema allows users to generate models in different LoDs based on their own understanding. However, this flexibility is also a drawback since there are no rules about whether certain objects could exist in certain objects or not [20]. In order to achieve the LoDs harmonization in CityGML, clear definitions of each LoD are first described in this research by referring to various literature and studying the characteristics of CityGML models. Besides studying the official description of LoDs in the CityGML schema, we also considered whether one entity should appear on certain LoD or not. The following sources were analyzed in order to obtain the definitions of each LoD:

A. CityGML specifications and encoding standard from OGC, such as [21,23].
B. Data sets recommended by the official CityGML website. By analyzing different LoDs, whether one entity exists in a certain LoD can be determined. If conflict occurs, we refer to the first source.
C. Papers about CityGML and LoDs in 3D GIS models, such as [20,44,48].

The summarized definitions of each LoD in terms of the participation of certain entities are listed in Table 3.

4.2. Process of LoDs harmonization

After defining each LoD, the harmonization between LoDs in CityGML was developed accordingly. The goal of the harmonization is that given building model in any LoD, the model could be transformed into lower LoDs. The harmonization process starts from LoD4 to LoD3 transformation. The major difference between LoD4 and LoD3 models is that LoD4 models have building interiors, such as interior building installations, furniture, and rooms. So the transformation from LoD4 to LoD3 is simply removing all the interior features and changing the LoD4 geometry to LoD3 geometry.

The LoD3 to LoD2 transformation is the most challenging part for the harmonization between LoDs in CityGML. Two major steps are needed here: the removal of openings and extraction of exterior shell of building which forms the LoD2 model. There have been several studies concerning the extraction of building envelope, but they are not applicable for buildings with complex building shape and wall structures. For instance, Fan et al. [24] proposed an algorithm which could extract the envelope of the building by comparing the distances of surfaces to the center of the building. However, this algorithm is not applicable to buildings with non-convex shapes. In our research, a new scanning algorithm which is inspired by the traditional Ray–Tracing algorithm was developed. The scanning algorithm tests each surface against other surfaces in the building and tries to determine whether this surface is on the exterior shell or not. After the scanning, only the exterior surfaces are kept to form the LoD2 model of the building.

Buildings in LoD1 are blocks that have no detailed roof structure. Since in LoD2 we have already found the exterior shell of the building, the LoD2 to LoD1 transformation is simply removing all the roof structure and building the envelope for the remaining walls. All the remaining surfaces are written into a solid CityGML LoD1 model. The whole process is summarized in Fig. 21.

5. Results and discussions

The proposed methodology for mapping between IFC with CityGML with different LoDs was tested on various IFC and CityGML models. The programming platform for the testing was JDK 7. The results of the tests show that the stand-alone translator developed based on the framework could capture all the geometry information of building components. As shown in Fig. 22, building components such as walls, doors, windows, roof, railing, stairs, covering, slabs, floors, curtain walls, and even building furniture were all converted by the translator from IFC to CityGML and vice versa. With the help of the developed Semantic City Model ADE, the semantic information from IFC could also be kept in the generated CityGML files, which could support more applications using the generated GIS model.

The results from our proposed framework were also compared with the CityGML models generated from the BIM Server [29], which is shown in Fig. 24. In Fig. 24, panel (b) was generated from the BIM Server while panel (c) was generated by the translator developed in this project. While our models kept almost all the geometry and semantic information from IFC models, the CityGML models from BIM Server missed some building components. The CityGML file generated from BIM Server contained eight walls, two roof entities with no geometry, one room entities and no interior building features such as stairs and railings. The models generated from our framework contained eight walls, four roof slabs with correct geometry, one floor surface, three rooms, and interior building installations such as stairs and railings. Given more entities in the models generated from our framework, since the geometry translator from IFC BRep/Swept Solid to CityGML BRep was developed with the principle to generate as few surfaces as possible while still preserving the geometry, the file size of models generated from our framework is much less than those from BIM Server. The geometric transformation of complex shapes is also tested in the complex shapes of stairs, as shown in Fig. 24 (g). Moreover, the model generated from BIM Server did not carry semantic information such as usage, load bearing, or room bounding. The detailed comparison is shown in Table 5.

The processing and storage of semantic information is shown in Fig. 25. Mapping of semantic information is limited in previous studies, which either failed to mention the semantic mapping methods, or did not provide a methodology to achieve complete semantic integration between BIM and GIS. In this study, we used a reference ontology to help capture all the information from BIM and GIS. The information exchange happens between the reference ontology and BIM schema or GIS schema. Moreover, extension to the GIS schema was developed based on the reference ontology to store semantic information from BIM. The extension is called the Semantic City Model ADE for CityGML. In Fig. 25, we showed how the semantic information stored in a wall in IFC format is transformed to a data table based on the reference ontology, and then transformed to CityGML with ADE. A user interface was developed to allow users better utilize the proposed integration framework. As shown in Fig. 25, the semantic information in BIM such as relationships of components and structural properties can be stored in CityGML as well. It is noticeable that the process of generating the reference ontology and CityGML ADE is extensible, in which users may later use this methodology to facilitate certain applications involving the use of data from BIM and GIS.

The efforts of the LoDs harmonization in CityGML were also tested using building models in different shapes in 3D GIS. The translator developed by our framework could translate models in higher LoDs to lower LoDs. Users are able to decide whether they want to keep the semantic information from higher LoDs, even if some of the components were deleted (e.g. interior walls) during the LoD harmonization process. Users also have the option of whether to simplify openings in order to further reduce the file size. Two testing models are shown in Figs. 23 and 24. Fig. 23 shows a 40-storey residential building in Hong Kong. The IFC model contains the exterior and interior walls, windows and doors, slabs, stairs, and furniture on each floor. The model was transformed into LoD4 CityGML model, which has more than 150,000 surfaces. The generated LoD4 CityGML model has a bigger file size (100 MB) than the original IFC model (44 MB) because some components are represented in Swept Solid in the IFC model while all components are represented in BRep in the CityGML model. The huge LoD4 model was then used to test the robustness of the LoDs harmonization framework. The results indicate that the LoDs transformation framework is capable of handling large models, and the generated LoD3, LoD2, and LoD1 models have no information loss or wrong geometry. A summary of components in each LoD is shown in Table 4 based on Fig. 24. Fig. 23 also shows the effect of our method to deal with buildings with complex building shapes. Buildings in non-convex shapes could also be simplified and translated into lower LoDs, which has not been achieved by other research approaches.

The generated GIS and BIM models were compared to models transformed in other approaches. Three representative mapping approaches between BIM and GIS were selected for comparison. The selected approaches were reported in van Berlo and de Laat [29], EI-Mekawy et al. [30], and Isikdag and Zlatanova [31]. The comparison was based on five dimensions, namely, the geometry transformation, bi-directional mapping, levels of detail, extensions to schemas, and semantic information transformation. The results of the comparisons are shown in Table 6.

In terms of geometry transformation, the Geo-BIM extension for CityGML was implemented on BIMServer [29] and we have shown the comparison in previous discussions. The frameworks reported in [30, 31] were not engineered to translators, nor did they talk about the transformation of geometry. Our framework, on the other hand, considered the algorithms for geometry transformation and a translator with user interface was developed. Our approach as well as the one reported in EI-Mekawy et al. [30] have considered the bidirectional mapping between IFC and CityGML. In terms of LoD transformation, we have proposed and verified processes to transform CityGML models in higher LoDs to models in lower LoDs. The LoD transformation had been mentioned in [30,31], but they did not provide details to each step of transformation. And they also failed to verify their approaches with real models. In our approach, we have proposed a reference ontology and proposed a CityGML ADE to support semantic information mapping from BIM. The other approaches, however, did not consider extending CityGML schema, or their mapping frameworks were only limited to existing entities in the schemas. Finally, we have studied the development and implementation of a reference ontology and its use in semantic information mapping, while in others' studies, they either touched on limited semantics or did not consider the semantic mapping.

6. Conclusions and future work

This paper presents a mapping framework between IFC and CityGML, the representative schemas in the BIM domain and the GIS domain respectively. Different levels of detail (LoDs) of CityGML are also considered in the mapping framework. Reference ontology and instance-based mapping rule generation were used in this study in order to achieve complete and accurate mapping. The reference ontology ensures that all the relevant information from the IFC and CityGML models are captured and mapped. Four basic concepts were introduced in the reference ontology: building object, geometry, property sets, and inverse relationship. The relationships between these four concepts were also developed by studying the schemas of IFC and CityGML. The reference ontology could be developed for mapping various building components, such as buildings and walls, as illustrated in this paper. The instance-based method compares instances from the two schemas and generates mapping rules for data value in both schemas. Special attention was paid to the transformation of local placement system to world coordinate systems, and transformation from CSG/Swept Solid to BRep, which is utilized in CityGML for geometry representation. Based on the developed reference ontology, a CityGML schema extension called the Semantic City Model ADE was developed in order to store the rich information from BIM models in CityGML models, especially for the semantic information and inverse relationships.

The harmonization between LoDs in CityGML was also developed in CityGML, which completes the mapping between IFC and CityGML. Clear definitions of each LoD in CityGML were given by literature review and review of data sets. Then, the transformation framework among LoDs in CityGML was developed.

The test results of the proposed transformation framework between IFC and CityGML showed that our proposed framework could achieve accurate and complete mapping between the two schemas. Moreover, the semantic information and inverse relationships were also captured by the Semantic City Model ADE for CityGML, which made our mapping complete. The test results of the LoDs harmonization also indicate that the framework for transformation among LoDs could generate models according to the specifications of LoDs in CityGML. The goal of mapping between IFC and CityGML in different LoDs is thus achieved. However, since we created our own extensions for CityGML, the common CityGML viewers do not support the extension yet, which means we have to develop our own viewer for the additional information from the Semantic City Model ADE for CityGML.

This study has several limitations. Firstly, the developed reference ontology, level of detail transformation framework and the translator were only applicable to IFC and CityGML. Secondly, this study only considered building models, and other kinds of models, such as roads, bridges, and tunnels that have used 3D, were not considered. Finally, in terms of solid geometry transformation, we only considered BRep, swept solid, and constructive solid geometries. The clipping geometry in IFC, which is constructed from Boolean differences between swept area solids, half spaces, and Boolean results were not considered. In the future, we will study more schemas in the BIM and GIS domain, such as gbXML and GML to enrich the reference ontology. The geometry transformation can be strengthened by including clipping geometry to BRep transformation.

Acknowledgements

The authors would like to acknowledge the support by the Hong Kong Research Grants Council, Grant No. 622,812. Any opinions and findings are those of the authors, and do not necessarily reflect the views of the Hong Kong Research Grants Council.

References

[1] Z. Aziz, C. Anumba, D. Ruikar, P. Carrillo, D. Bouchlaghem, Intelligent wireless web services for construction—a review of the enabling technologies, Autom. Constr. 15 (2) (2006) 113–123.
[2] C. Eastman, P. Teicholz, R. Sacks, K. Liston, BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors, Wiley Publishing, 2008.
[3] S.A. Kim, D. Shin, Y. Choe, T. Seibert, S.P. Walz, Integrated energy monitoring and visualization system for Smart Green City development: designing a spatial information integrated energy monitoring model in the context of massive data management on a web based platform, Autom. Constr. 22 (2012) 51–59.
[4] M.W. Sweeney, Geographic information systems, Water Environ. Res. 71 (5) (1999) 551–556.
[5] T. Wilson, OGC Keyhole Markup Language, 2.2. 0, Open GIS Consortium, 2008.
[6] ISO (International Organization for Standardization), ISO/PAS 17506:2012 Industrial Automation Systems and Integration—COLLADA Digital Asset Schema Specification for 3D Visualization of Industrial Data, 2012.
[7] Open Geospatial Consortium, OGC® Geography Markup Language (GML)—Extended Schemas and Encoding Rules, 2012.
[8] X. Su, A.R. Andoh, H. Cai, J. Pan, A. Kandil, H.M. Said, GIS-based dynamic construction site material layout evaluation for building renovation projects, Autom. Constr. 27 (2012) 40–49.
[9] A. Simão, P.J. Densham, M. Haklay, Web-based GIS for collaborative planning and public participation: an application to the strategic planning of wind farm sites, J. Environ. Manag. 90 (6) (2009) 2027–2040.
[10] U. Isikdag, J. Underwood, G. Aouad, An investigation into the applicability of building information models in geospatial environment in support of site selection and fire response management processes, Adv. Eng. Inform. 22 (4) (2008) 504–519.
[11] C. Anumba, Application of GIS to Labour Market Planning in Construction, Loughborough University, 2006.
[12] V. Bansal, Potential of GIS to find solutions to space related problems in construction industry, World Academy of Science, Eng. Technol. 32 (2007) 307.
[13] J. Benner, A. Geiger, K. Leinemann, Flexible generation of semantic 3D building models, Proceedings of the 1st International Workshop on Next Generation 3D City Models, EuroSDR Publication, Bonn, German 2005, pp. 27–31.
[14] I. Hijazi, M. Ehlers, S. Zlatanova, T. Becker, L. Berlo, Initial Investigations for Modeling Interior Utilities Within 3D Geo Context: Transforming IFC-Interior Utility to CityGML/UtilityNetworkADE, Advances in 3D Geo-Information Sciences, Springer 2011, pp. 95–113.
[15] A. Strzalka, J. Bogdahn, V. Coors, U. Eicker, 3D city modeling for urban scale heating energy demand forecasting, HVAC&R Res. 4 (2011) 526–539.
[16] T.K. Thiis, E. Hjelseth, Use of BIM and GIS to Enable Climatic Adaptations of Buildings, eWork and eBusiness in Architecture, Engineering and Construction, Taylor & Francis, 2008 409–417.
[17] Z. Ma, Z. Wei, W. Song, Z. Lou, Application and extension of the IFC standard in construction cost estimating for tendering in China, Autom. Constr. 20 (2) (2011) 196–204.
[18] P. Pauwels, D. Van Deursen, R. Verstraeten, J. De Roo, R. De Meyer, R. Van De Walle, J. Van Campenhout, A semantic rule checking environment for building performance checking, Autom. Constr. 20 (5) (2011) 506–518.
[19] M. Hassanain, T. Froese, D. Vanier, Development of a maintenance management model based on IAI standards, Artif. Intell. Eng. 15 (2) (2001) 177–193.
[20] G. Gröger, L. Plümer, CityGML—interoperable semantic 3D city models, ISPRS J. Photogramm. Remote Sens. (2012) 12–33.
[21] G. Gröger, T.H. Kolbe, C. Nagel, K.-H. Häfele, OpenGIS City Geography Markup Language Encoding Standard 2.0, Open Geospatial Consortium Inc., 2012
[22] A. Lapierre, P. Cote, Using Open Web Services for Urban Data Management: A Testbed Resulting from an OGC Initiative for Offering Standard CAD/GIS/BIM Services, in: V. Coors, M. Rumor, E.M. Fendel, S. Zlatanova (Eds.), Urban and Regional Data Management, Annual Symposium of the Urban Data Management SocietyCRC Press, Taylor & Francis Group 2007, pp. 381–393.
[23] G. Gröger, T.H. Kolbe, A. Czerwinski, C. Nagel, OpenGIS City Geography Markup Language (CityGML) Encoding Standard, Open Geospatial Consortium Inc., 2008
[24] H. Fan, L. Meng, M. Jahnke, Generalization of 3D Buildings Modelled by CityGML, Advances in GIScience, Springer, Berlin Heidelberg, 2009 387–405.
[25] J. Irizarry, E.P. Karan, F. Jalaei, Integrating BIM and GIS to improve the visual monitoring of construction supply chain management, Autom. Constr. 31 (2013) 241–254.
[26] R. Liu, R.R.A. Issa, 3D Visualization of Sub-Surface Pipelines in Connection with the Building Utilities: Integrating GIS and BIM for Facility Management, in: R.R. Issa, I. Flood (Eds.),Computing in Civil Engineering (2012) 2012, pp. 341–348.
[27] T. Park, T. Kang, Y. Lee, K. Seo, Project Cost Estimation of National Road in Preliminary Feasibility Stage Using BIM/GIS Platform, in: R.R. Issa, I. Flood (Eds.), Computing in Civil and Building Engineering (2014), American Society of Civil Engineers 2014, pp. 423–430.
[28] T.H. Kolbe, G. Gröger, L. Plümer, CityGML: Interoperable Access to 3D City Models, in: P.V. Oosterom, S. Zlatanova, E.M. Fendel (Eds.), Geo-Information for Disaster Management, Springer, Berlin Heidelberg 2005, pp. 883–899.
[29] L. Van Berlo, R. De Laat, Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension, in: T.H. Kolbe, G. König, C. Nagel (Eds.), Advances in 3D Geo-Information Sciences, Springer, Berlin Heidelberg 2011, pp. 211–227.
[30] M. El-Mekawy, A. Stman, K. Shahzad, Towards Interoperating CityGML and IFC Building Models: A Unified Model Based Approach, in: T.H. Kolbe, G. König, C. Nagel (Eds.), Advances in 3D Geo-Information Sciences, Springer, Berlin Heidelberg 2011, pp. 73–93.
[31] U. Isikdag, S. Zlatanova, Towards Defining a Framework for Automatic Generation of Buildings in CityGML Using Building Information Models, 3D Geo-Information Sciences, Springer, Berlin Heidelberg, 2009 79–96.
[32] C. Nagel, A. Stadler, T.H. Kolbe, Conceptual Requirements for the Automatic Reconstruction of Building Information Models from Uninterpreted 3D Models, in: T.H. Kolbe, H. Zhang, S. Zlatanova (Eds.),Proceedings of the International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vancouver 2009, pp. 46–53.
[33] X. Xu, L. Ding, H. Luo, L. Ma, From building information modeling to city information modeling, J. Inf. Technol. Constr. (2014) 292–307.
[34] T.W. Kang, C.H. Hong, A study on software architecture for effective BIM/GIS-based facility management data integration, Autom. Constr. 54 (2015) 25–38.
[35] T.R. Gruber, A translation approach to portable ontology specifications, Knowl. Acquis. 5 (2) (1993) 199–220.
[36] T.R. Gruber, Toward principles for the design of ontologies used for knowledge sharing, Int. J. Hum. Comput. Stud. 43 (5–6) (1995) 907–928.
[37] M. Fernandez-Lopez, O. Corcho, Ontological Engineering: With Examples from the Areas of Knowledge Management, E-Commerce and the Semantic Web, first ed. Springer-Verlag, London, 2004.
[38] S. Gauch, J. Chaffee, A. Pretschner, Ontology-based personalized search and browsing, Web Intell. Agent Syst. 1 (3) (2003) 219–234.
[39] Y. Kitamura, S. Takafuji, R. Mizoguchi, Towards a reference ontology for functional knowledge interoperability, ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Las Vegas, Nevada, USA 2007, pp. 111–120.
[40] C. Rosse, J.L. Mejino Jr., A reference ontology for biomedical informatics: the foundational model of anatomy, J. Biomed. Inform. 36 (6) (2003) 478–500.
[41] B. Andersson, M. Bergholtz, A. Edirisuriya, T. Ilayperuma, P. Johannesson, J. Gordijn, B. Grégoire, M. Schmitt, E. Dubois, S. Abels, Towards a Reference Ontology for Business Models, Conceptual Modeling-ER 2006, Springer 2006, pp. 482–496.
[42] M. Herrlich, H. Holle, R. Malaka, Integration of CityGML and Collada for High-Quality Geographic Data Visualization on the PC and Xbox 360, in: H. Yang, R. Malaka, J. Hoshino, J. Han (Eds.), Lecture Notes in Computer Science, Springer, Berlin/Heidelberg 2010, pp. 270–277.
[43] R. Lipman, Details of the mapping between the CIS/2 and IFC product data models for structural steel, J. Inf. Technol. Constr. 14 (1) (2009) 1–13.
[44] J. Döllner, H. Buchholz, Continuous Level-of-Detail Modeling of Buildings in 3D City Models, in: C. Shahabi, O. Boucelma (Eds.),13th Annual ACM International Workshop on Geographic Information Systems, ACM New York, Bremen, Germany 2005, pp. 173–181.
[45] I.C. Wu, S.H. Hsieh, Transformation from IFC data model to GML data model: methodology and tool development, J. Chin. Inst. Eng. 30 (6) (2007) 1085–1090.
[46] F. Leite, A. Akcamete, B. Akinci, G. Atasoy, S. Kiziltas, Analysis of modeling effort and impact of different levels of detail in building information models, Autom. Constr. 20 (5) (2011) 601–609.
[47] Innovation CRC Construction, National Guidelines for Digital Modelling, Cooperative Research Centre for Construction Innovation, Brisbane, Australia, 2009.
[48] H. Fan, L. Meng, Automatic Derivation of Different Levels of Detail for 3D Buildings Modeled by CityGML, in: M.-J. Kraak (Ed.), Proceedings of 24th International Cartography Conference (ICC2009), International Cartographic Association, Santiago Chile 2009, pp. 15–21.