Semantic Enrichment of As-is BIMs for Building Energy Simulation

88

Huaquan Ying, Hui Zhou, Qiuchen Lu, Sanghoon Lee, and Ying Hong

Abstract

Recent efforts on automatic 3D modeling of existing buildings often result in semantically poor as-is building information models (BIMs). Such a BIM usually consists of a set of objectified surfaces characterized by the building element types they represent and 3D surface geometries. It cannot be directly used as the geometry input of building energy modeling (BEM) as the key concepts of second-level space boundaries (SBs) are missing. This paper proposes a semantic enrichment approach for automatically adding such semantic concepts inferred from the semantically poor as-is BIM. The output of this approach is a corresponding IFC BIM with second-level SBs, which can be further used by various energy simulation tools. Preliminary experimental results using a building surface model demonstrate the performance of the proposed approach.

Keywords

Semantic enrichment • Second-level space boundary • Building information modeling
Building energy modeling • Existing buildings

88.1 Introduction

Energy retrofits of existing buildings play a significant role in reducing global energy consumption [1, 2]. In an energy retrofit project, building energy modeling (BEM) is commonly used for assessing actual building performance, diagnosing malfunctioning building systems, and prioritizing various retrofit strategies quantitatively [2, 3].

Recent efforts have been conducted in automatic 3D modeling of existing buildings for BEM purpose, due to the inaccuracy and labor inefficiency of current manual BEM input preparation practice [4–8]. In these efforts, state-of-the-art surveying technologies such as laser scanning and photogrammetry are widely used to capture the as-is conditions of buildings. The outputs (as-is BIMs) generally consist of recognized building components with reconstructed 3D geometries of their visual surfaces [5–7, 9]. As summarized in [10], there are two main limitations in these efforts from the view of geometric modeling of existing buildings for BEM purpose. First, existing efforts generally focus on reconstructing building components of specific building parts, such as building facades [5, 7], interior spaces [9], or surrounding shades [6]. Thus, their outputs only support some preliminary energy analysis (e.g. building orientation and shading analysis) and cannot be used for the detailed whole-building energy simulation that requires complete building geometry descriptions. Second, their outputs usually lack semantic information required by BEM, especially the concept of second-level space boundary (SB). In BEM, the building geometry is depicted as a collection of objectified SBs, which specify the topological relationships between spaces and their surrounding building elements, space boundary geometries and other related attributes (see

H. Ying (✉) · H. Zhou · Q. Lu · S. Lee
The University of Hong Kong, Kowloon Tong, Hong Kong
e-mail: u3004315@connect.hku.hk

Y. Hong
The University of New South Wales, Sydney, Australia

I. Mutis and T. Hartmann (eds.), *Advances in Informatics and Computing in Civil and Construction Engineering*,
https://doi.org/10.1007/978-3-030-00220-6_88

Sect. 88.2) [12]. However, the results produced by existing efforts only contain the element type that each reconstructed visual surface represents and the corresponding 3D geometry.

To address these limitations, Ying et al. [10] proposed an image-driven framework for automatically constructing complete BEM geometry models for existing buildings. Specifically, that framework aims at directly generating IDF-based models from images for the dedicated building energy performance (BEPS) tool called EnergyPlus.

This paper refines that framework by redesigning the semantic enrichment module to generate as-is IFC BIMs with second-level SBs. These second-level SBs define building geometry data required by BEM in IFC format, which can be further extracted and directly mapped to internal data models of various BEPS tools. In other words, this improvement enables the access of as-is BEM geometry data by various BEPS tools, as IFC is an open and neutral format to support life-cycle BIM data exchange [12]. The proposed semantic enrichment approach takes semantically poor as-is models of existing buildings as an input. Such a model consists of a set of visual surfaces of the entire building. Each surface only carries the information (i.e. building element type it represents and corresponding 3D surface geometry) that can be obtained by most existing as-is modeling efforts (e.g. [5, 6, 9]). The proposed approach consists of a set of rules and computational geometry algorithms, which are developed to infer the second-level SBs and other semantic requirements based on the inputs.

The rest of the paper is structured as follows: Sect. 88.2 explains the semantic requirements of as-is BIMs for BEM; Sect. 88.3 details the proposed semantic enrichment approach; Sect. 88.4 reports a preliminary experimental validation; and Sect. 88.5 concludes this paper with a discussion on the limitations and future work.

88.2 Semantic Requirements

In most BEPS tools, building geometry is defined as a collection of second-level SBs [11]. In IFC, a SB is defined as an objectified relationship by IfcRelSpaceBoundary. To standardize the definition and processing of SBs, a dedicated model view definition (MVD) based on IFC 2 × 3 specification called Space Boundary Add-on View has been published by buildingSMART [13]. In this paper, IFC4 is used as the base specification, as it introduces a new entity called IfcRelSpaceBoundary2ndLevel [14]. Compared to IfcRelSpaceBoundary, this entity stores two additional types of relationships required by BEM: (1) the relationship between pairwise type 2a SBs; and (2) the relationship between a SB of an opening element and the SB of the wall that hosts the opening.

This study designs a dedicated IFC data structure with specific IFC entities (see Fig. 88.1) to store enriched as-is BIM data. In addition, Fig. 88.1 also lists the attributes of relevant entities (except entities in red dashed box) used for storing necessary information. Two attributes (i.e., GlobalID and ObjectPlacement) are used for the entities in red dashed box. The structure design is based on three principles: (1) storing minimal information required for the definition of SBs; (2) keeping consistent with existing energy analysis-related MVDs; and (3) satisfying syntactic constraints from IFC4 specifications.

Except the computation of second-level SB geometries (see the green dashed box in Fig. 88.1), the semantic information that need to be added can be classified into two groups:

Semantic information required by the definition of second-level SBs. This information is required by the attributes of IfcRelSpaceBoundary2ndLevel, including:

(1) GlobalID: assigning a unique identifier for specifying the SB.
(2) Description: specifying the type of the SB in terms of “2a” or “2b”. The SB is type 2a if there is a space on the other side of the building element providing this SB; type 2b, otherwise [13]. The heat transfer between the space that the SB bounds and the space on the other side can occur in the first case, but not in the second case. Therefore, an input surface may need to be spilt into several pieces, in accordance with these concepts.
(3) RelatingSpace: specifying the space object that this SB bounds. The space object can either be an internal space defined by IfcSpace or the outdoor environment by IfcExternalSpatialElement.
(4) RelatedBuildingElement: specifying the building element that provides this SB.
(5) PhysicalOrVirtualBoundary: specifying whether this SB is physical (i.e. provided by a physical element such as walls and slabs) or virtual (i.e. provided by a virtual element such as virtual space separators).
(6) InternalOrExternalBoundary: specifying whether this SB is external (i.e. provided by an external element) or internal (i.e. provided by an internal element).
(7) ParentBoundary: for a SB of an opening (i.e. window or door), specifying the parent SB (usually a SB of a wall) that hosts this SB; for a SB of a shading device, specifying the parent SB that this SB is attached to.

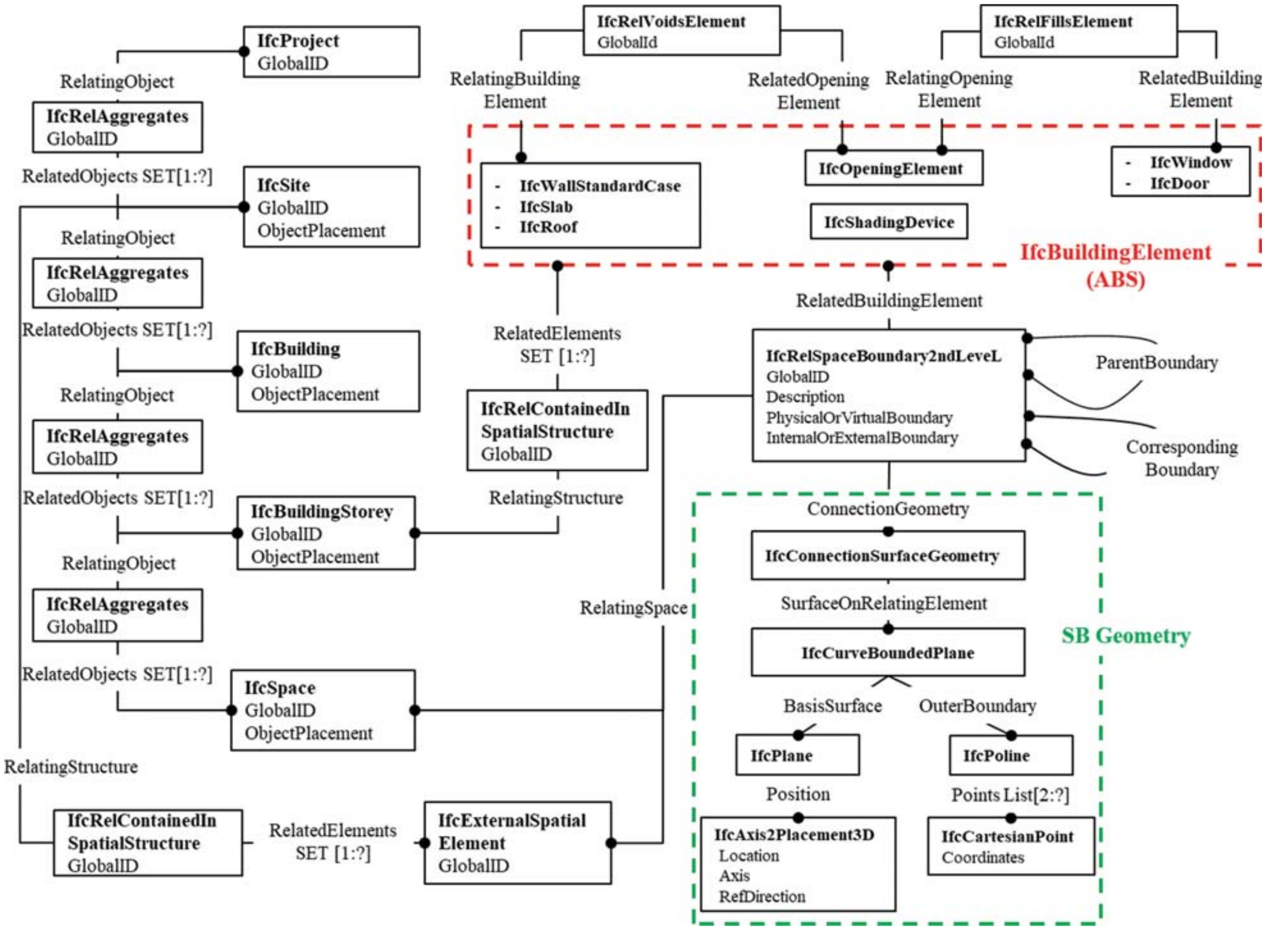


Fig. 88.1 The IFC data structure to store semantically enriched as-is BIM data *Note* Beams, columns and curtain walls are not considered yet in this paper

(8) CorrespondingBoundary: only for a type 2a SB, specifying the symmetrical SB on the other side of the building element providing this SB.

Semantic information required by syntactic constraints of IFC4 specifications.

(1) IFC Root and spatial objects: Entities required to maintain the hierarchical structure, including IfcProject, IfcSite, IfcBuilding, and IfcBuildingStorey.
(2) Aggregation relationships: the one-to-many aggregation relationships including IfcProject-IfcSite, IfcSite-IfcBuilding, IfcBuilding-IfcBuildingStorey and IfcBuildingStorey-IfcSpace should be correctly defined by IfcRelAggregates.
(3) Containment relationships: the one-to-many containment relationships including IfcBuildingStorey-IfcBuildingElement and IfcSite-IfcExternalSpatialElement should be correctly defined by IfcRelContainedInSpatialStructure.
(4) Topological relationships: the topological relationships between physical elements and their hosting opening elements should be correctly defined via IfcRelVoidsElement and IfcRelFillsElement.

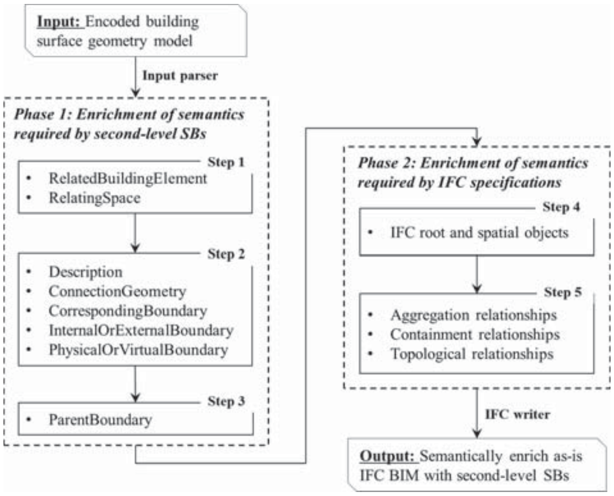


Fig. 88.2 Proposed semantic enrichment approach for as-is BIM

88.3 Semantic Enrichment Approach

The proposed approach for automatically enriching semantically poor as-is models includes two primary phases (see Fig. 88.2): enrichment of semantics required by second-level SBs (see Sect. 3.2) and enrichment of semantics require by IFC specifications (see Sect. 3.3).

The semantic enrichment process is progressive as some semantic information triggers other semantics. Specifically, the geometries of surfaces in the input model are defined by the following rules to ensure a correct surface normal, which is essential to the process: polygon vertices of an exterior surface follow a clockwise order (viewed from outdoor); while, polygon vertices of interior surfaces follow a counter-clockwise order (viewed from interior spaces).

88.3.1 Enrichment of Semantics Required by Second-Level SBs

Step 1: Add semantics of “RelatingSpace” and “RelatedBuildingElement”. The core of this step is to recognize the space that an input surface bounds and the building element that provides it. The recognized space and element are then added to the objectified surface and can be directly inherited by corresponding SBs split from this surface in the next step.

Space recognition. The input surface sets are firstly classified into three groups: surfaces of building exterior, surfaces of interior non-opening elements, and surfaces of interior openings. For the first group of surfaces, an outdoor space object is logically created and then attached to all surfaces via a new attribute “RelatingSpace”.

For surfaces in the second group, a Monte Carlo-based ray tracing algorithm is developed to infer spaces that they bound. The basic idea of the algorithm is described as follows. First, for a surface, a set of rays emitted from the back side (i.e. the side viewed from reverse surface normal direction) of the surface are generated by using Monte Carlo technique. Second, all the surfaces that these rays first hit are detected and reserved after removing the duplicates. Third, for each detected surface, a recursive function consisting of previous two steps is conducted. The recursive operations end when no new surfaces are found. So far, all surfaces bounding the same space with the selected surface are found. Then a corresponding space object is logically created and assigned to these surfaces. The next step is to update the original group by removing the grouped surfaces and start new iterations to cluster all remaining surfaces in spaces.

After processing the second group, all space objects in building interior are recognized. The space object bounded by an opening surface in the third group is identical to the space bounded by the non-opening element surface hosting the opening surface. The exact non-opening element surface can be detected by the following computational method: first, filtering out the surfaces in the second group that are not coplanar with the opening surface; second, examining whether each surface remained in the second group contains the opening surface geometrically.

Building element recognition. Although the element type of each surface is known (given in the input), its corresponding building element remains unknown. Rather than re-constructing the solid geometry, building element recognition, in this paper, aims to group the surfaces belonging to a comment element logically. In order to achieve this objective, this study developed a set of rule-based algorithms. Specifically, rule sets are established respectively for the recognition of different types of building elements (e.g., walls, roofs, slabs, windows, doors, and shading devices). As an example, for an opening surface S_O, another corresponding surface are recognized by checking whether the following rules are satisfied: SameSurfaceTypeWith (S_O), ParallelWith(S_O), DistanceWith(S_O) < *dist* (a constant slightly larger than the thickness of openings), and AfterProjectionOverlapWith (S_O). After all surfaces are grouped, corresponding building elements can be logically created and assigned to relevant surfaces via "RelatedBuildingElement".

Step 2: Add semantics of "Description", "ConnectionGeometry", "CorrespondingBoundary", "InternalOrExternal", and "PhysicalOrVirtual". This step is to compute second-level SBs including geometries and all the remaining semantics except ParentBoundary. The SBs are created at the building element level, more specifically, by processing surfaces belonging to a common building element.

For surfaces of opening elements, the SBs are directly inferred as each surface corresponds to a SB. The rules for defining a SB of an opening are detailed as follows:

- "Description": "2a";
- "ConnectionGeometry": the geometry of the corresponding surface;
- "CorrespondingBoundary": another SB provided by the same element;
- "PhysicalOrVirtual": "Physical";
- "InternalOrExternal": if the surface comes from the building interior, then the value is "Internal"; otherwise, "External".

The concept of second-level SBs is extended for shading elements. Input surfaces of shading elements are directly defined as corresponding SBs by the following rules:

- "Description": "ShadingDevice";
- "ConnectionGeometry": the geometry of the corresponding surface;
- "CorrespondingBoundary": Null;
- "PhysicalOrVirtual": "Physical";
- "InternalOrExternal": "External".

A computational geometry approach is developed to compute relevant SBs for surfaces of other building elements (i.e. walls, slabs, roofs). The basic idea of this approach is described as follows. First, a surface (*S*) from the set of surfaces (*Ss*) of a common building element is selected. Second, all the remaining surfaces are projected on the plane of *S*. Third, the intersection parts and difference parts between *S* and all other projected surfaces are computed. Fourth, SBs ("Description" and "ConnectionGeometry") from *S* are defined by using the following rule: each intersection part refers to a "2a" SB and each difference part a "2b" SB. Fifth, the previous four steps for all other surfaces in *Ss* are repeated to split all these surfaces into corresponding SBs. Finally, the following semantics are added to each SB:

- "CorrespondingBoundary": for a "2b" SB, the value is null; for a "2a" SB, the corresponding SB is another "2a" SB of the related element, which fully overlaps this SB on the plane perpendicular to the surface normal;
- "PhysicalOrVirtual": if the related element is a virtual element, then the value is "Virtual"; otherwise, "Physical".
- "InternalOrExternal": if the original surface comes from building exterior, the value is "External"; otherwise "Internal".

To ensure the correct geometric definition of these SBs, two additional operations are performed. First, the geometry of a SB should be a simple polygon without holes. In this paper, the SBs with holes generated from the above algorithms are further triangulated to remove those holes. Second, a SB should be defined with an outward surface normal. If not, the order of the vertices of corresponding polygon will be reversed.

Step 3: Add semantic of "ParentBoundary". This step is to add the semantic information required by ParentBoundary to each SB output from Step 2. This attribute is only needed for SBs of opening elements and shading devices. The parent boundary (*PSB*) of an opening SB (*OSB*) refers to the SB of a wall that hosts the particular SB. Geometrically, *OSB* is fully contained in *PSB*. A two-step method is developed to assist the detection of *PSB* for *OSB*: first, find out SBs (*CandinateSBs*) that are coplanar with *OSB*; second, examine each SB in *CandinateSBs* until the SB that geometrically contains *OSB* is found. The *PSB* of a SB of an attached shading element refers to the SB of a wall that the element is attached to. It can be detected by checking whether a SB of a wall geometrically contains an edge of this SB.

88.3.2 Enrichment of Semantics Required by IFC Specifications

Step 4: Create IFC root and spatial objects. This step aims to create objects required by IFC specifications for maintaining the IFC hierarchical structure. These objects include IFC root object (Project) and spatial objects (i.e., Site, Building, Building Storey). Objects including Project, Site, and Building can be directly created by using IfcProject, IfcSite, and IfcBuilding entities respectively. Note that a Project may contain more than one Site, which also may contain more than one Building. For the sake of simplicity, this paper assumes that there is one Building and one Site. The building stories are inferred by the gaps between the boundary geometries (i.e. geometries of the surfaces bounding the spaces) of spaces in the vertical direction.

Step 5: Create aggregation, containment and topological relationships. In this step, three types of relationships are created to link generated IFC objects. The aggregation relationships of IfcProject—IfcSite, IfcSite—IfcBuilding, and IfcBuilding—IfcBuildingStorey are defined using the entity IfcRelAggregates. The aggregation relationship between a building storey object and all the space objects in this storey are also defined by IfcRelAggregates. The one-to-many containment relationship between a building storey and all the building elements in this storey is defined by IfcRelContainedInSpatialStructure. This relationship is computed based on relevant SBs and the aggregation relationships between building stories and space objects. SBs provide linkages between a building element and spaces that the building element bounds. The building storey that the building element stands in can thus be determined through the aggregation relationships of building storey—spaces. The topological relationships between opening elements and non-opening elements that host them need to be explicitly reserved. Procedures for computing this type of relationships are as follows. First, all SBs (i.e., *BSBs*) provided by the building element are found via the attribute RelatedBuildingElement of SBs. Second, all SBs (i.e., *OSBs*) of opening elements are found. Third, for each SB in *OSBs*, a check whether its parent boundary matches any SB in *BSBs* is implemented. If so, it means that the opening element is hosted by the building element.

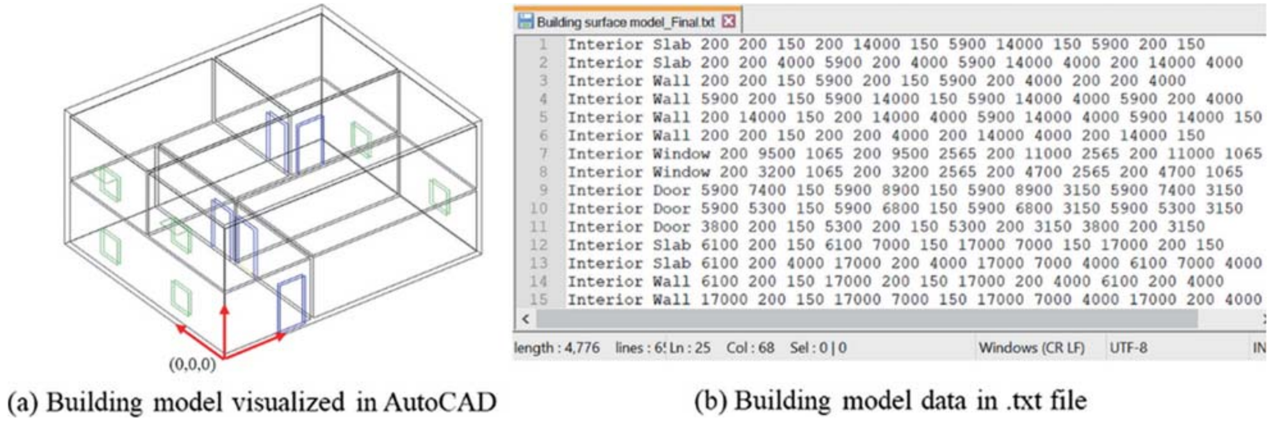
88.3.3 Enriched IFC Model Generation

All the objects, attributes and relationships created so far are ready to be defined with corresponding IFC entities and output as an IFC instance file. The hierarchical coordinate systems of an IFC model are defined through the attribute ObjectPlacemnt of IfcSite, IfcBuilding, IfcBuildingStorey, and IfcSpace. To simplify this process, all these coordinate systems are set to be same with the coordinate system of the input surface geometry model. In this way, there is no need to compute transformation relationships between two linked coordinate systems. It is important to note that a SB needs to be defined in a specific coordinate system, which is relative to the coordinate system of the space that the SB bounds.

88.4 Approach Implementation and Validation

To prove the feasibility of the proposed approach, a prototype application was implemented in C# programing language. The IFC Engine DLL [15] was used as the IFC writer to save the output in IFC. A building surface model was manually prepared for an experimental test from a two-story building (see Fig. 88.3a). This model consists of 65 surfaces covering all the types of building elements targeted in this paper. All these surfaces were coded by the schema “SurfacePosition(Interior/Exterior) SurfaceType SurfaceGeomerty” and saved in a txt file, as shown in Fig. 88.3b.

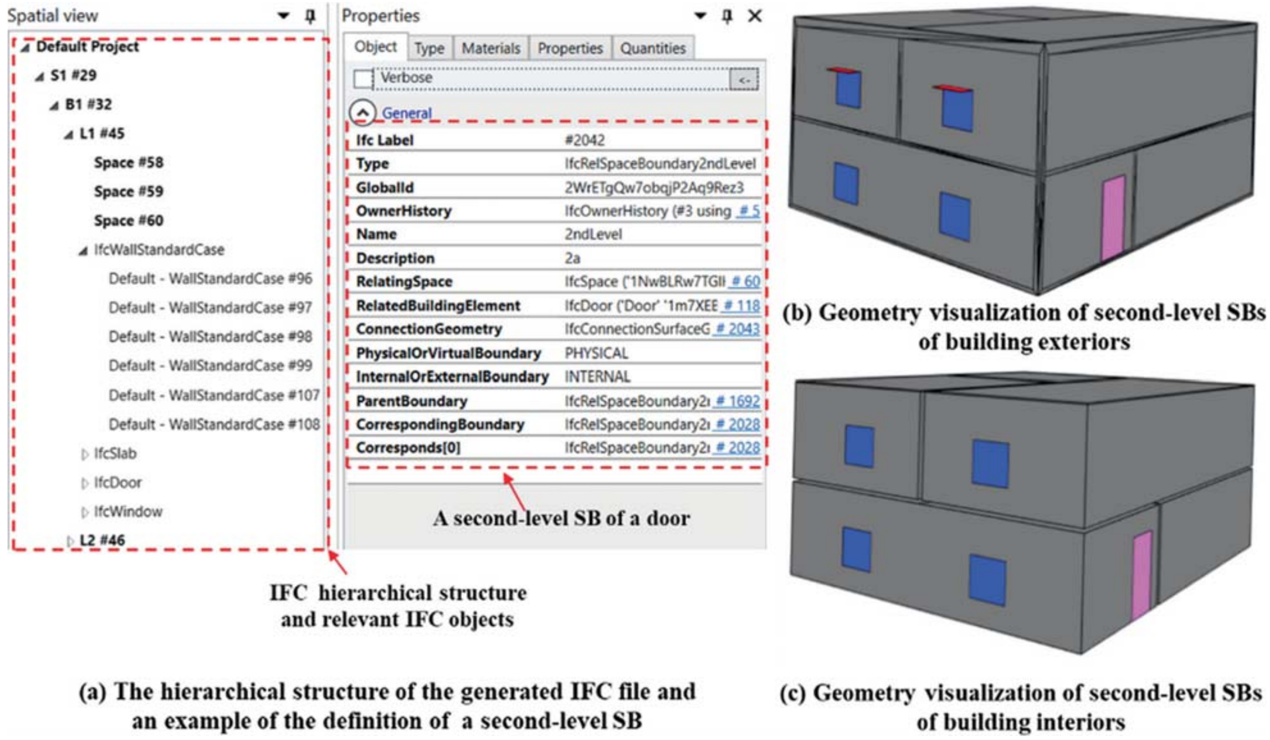
The prototype application parsed and processed the test model and generated a corresponding IFC file with second-level SBs. In the IFC file, a total of 201 second-level SBs including 78 “2a” SBs, 121 “2b” SBs (including 112 triangulated “2b” SBs), and 2 “ShadingDevice” SBs were saved. 6 spaces that these SBs bound, 8 walls, 2 slabs, 5 doors, 6 windows, 2 shading elements and 1 roof that provides these SBs are identified. Furthermore, other objects (i.e. IfcProject, IfcSite, IfcBuillding, and IfcBuildingStorey) and their relationships (i.e. the aggregation, containment and topological relationships specified in Sect. 2) are also correctly defined in the IFC file, as shown in Fig. 88.4a. Figure 88.4b, c display the geometries of the second-level SBs of building exteriors and interiors respectively.



```
Interior Slab 200 200 150 200 14000 150 5900 14000 150 5900 200 150
Interior Slab 200 200 4000 5900 200 4000 5900 14000 4000 200 14000 4000
Interior Wall 200 200 150 5900 200 150 5900 200 4000 200 200 4000
Interior Wall 5900 200 150 5900 14000 150 5900 14000 4000 5900 200 4000
Interior Wall 200 14000 150 200 14000 4000 5900 14000 4000 5900 14000 150
Interior Wall 200 200 150 200 200 4000 200 14000 4000 200 14000 150
Interior Window 200 9500 1065 200 9500 2565 200 11000 2565 200 11000 1065
Interior Window 200 3200 1065 200 3200 2565 200 4700 2565 200 4700 1065
Interior Door 5900 7400 150 5900 8900 150 5900 8900 3150 5900 7400 3150
Interior Door 5900 5300 150 5900 6800 150 5900 6800 3150 5900 5300 3150
Interior Door 3800 200 150 5300 200 150 5300 200 3150 3800 200 3150
Interior Slab 6100 200 150 6100 7000 150 17000 7000 150 17000 200 150
Interior Slab 6100 200 4000 17000 200 4000 17000 7000 4000 6100 7000 4000
Interior Wall 6100 200 150 17000 200 150 17000 200 4000 6100 200 4000
Interior Wall 17000 200 150 17000 7000 150 17000 7000 4000 17000 200 4000
```

(a) Building model visualized in AutoCAD (b) Building model data in .txt file

Fig. 88.3 The test model and its input data



Ifc Label	#2042
Type	IfcRelSpaceBoundary2ndLevel
GlobalId	2WrETgQw7obqjP2Aq9Rez3
OwnerHistory	IfcOwnerHistory (#3 using # 5
Name	2ndLevel
Description	2a
RelatingSpace	IfcSpace ('1NwBLRw7TGI # 60
RelatedBuildingElement	IfcDoor ('Door' '1m7XEE # 118
ConnectionGeometry	IfcConnectionSurfaceG # 2043
PhysicalOrVirtualBoundary	PHYSICAL
InternalOrExternalBoundary	INTERNAL
ParentBoundary	IfcRelSpaceBoundary2ı # 1692
CorrespondingBoundary	IfcRelSpaceBoundary2ı # 2028
Corresponds[0]	IfcRelSpaceBoundary2ı # 2028

(a) The hierarchical structure of the generated IFC file and an example of the definition of a second-level SB

(b) Geometry visualization of second-level SBs of building exteriors

(c) Geometry visualization of second-level SBs of building interiors

Fig. 88.4 The output IFC BIM *Note* in **b** and **c** SBs are colored based on the building element types (window-blue; door-pink; shading element-red; others-gray)

88.5 Conclusions and Future Work

Recent efforts on using state-of-the-art surveying technologies to construct as-is BIMs of existing buildings often result in semantically poor surface geometry models. However, such surface models cannot be directly used as the geometry input for building energy simulations, since the key concept of second-level SBs has not been established. This paper proposed a progressive approach for automatically computing such semantic concepts based on those surface models. The proposed approach provides procedures for constructing semantic as-is IFC BIMs from surface representations of existing buildings.

In addition, the developed processes and methods for inferring spatial objects, building elements, aggregation relationships, containment relationships and topological relationships are applicable to the generation of IFC-based BIMs from scratch. A prototype application was developed and preliminary experimental results using a building surface model demonstrate the feasibility of the approach.

This approach has some limitations to be addressed in future. First, this approach requires that the surface geometries should be appropriately defined to satisfy specific requirements on surface normal. However, this may not be assured by the input models. An automatic pre-checking and correction method is suggested. Second, this approach is limited to processing polygonal surfaces. Curved surfaces need to be manually segmented. Third, the approach considers the building elements essential to BEM only. Some interior elements like ceilings that are usually found in existing buildings have not been considered. Finally, some algorithms in this approach need to be improved to address various cases in real-word buildings. For example, the algorithm for clustering surfaces of slabs assumes that there is only one slab in a building story, which is not always common in the practice.

Acknowledgements The work described in this paper was supported by a grant from Graduate Collaborative Research Awards funded by Universitas 21.

References

1. Sustainable Buildings and Construction Programme. http://web.unep.org/10yfp/programmes/sustainable-buildings-and-construction-programme. Accessed 16 Apr 2018
2. Ma, Z., Cooper, P., Daly, D., et al.: Existing building retrofits: methodology and state-of-the-art. Energy Build. **55**, 889–902 (2012)
3. Sanhudo, L., Ramos, N.M., Martins, J.P., et al.: Building information modeling for energy retrofitting-a review. Renew. Sustain. Energy Rev. **89**, 249–260 (2018)
4. Cho, Y.K., Ham, Y., Golpavar-Fard, M.: 3D as-is building energy modeling and diagnostics: a review of the state-of-the-art. Adv. Eng. Inform. **29**(2), 184–195 (2015)
5. Wang, C., Cho, Y.K., Kim, C.: Automatic BIM component extraction from point clouds of existing buildings for sustainability applications. Autom. Constr. **56**, 1–13 (2015)
6. Díaz-Vilariño, L., Lagüela, S., Armesto, J., Arias, P.: Semantic as-built 3D models including shades for the evaluation of solar influence on buildings. Sol. Energy **92**, 269–279 (2013)
7. Cao, J., Metzmacher, H., O'Donnell, J., et al.: Facade geometry generation from low-resolution aerial photographs for building energy modeling. Build. Environ. **123**, 601–624 (2017)
8. Ham, Y., Golparvar-Fard, M.: Mapping actual thermal properties to building elements in gbXML-based BIM for reliable building energy performance modeling. Autom. Constr. **49**, 214–224 (2015)
9. Xiong, X., Adan, A., Akinci, B., Huber, D.: Automatic creation of semantically rich 3D building models from laser scanner data. Autom. Constr. **31**, 325–337 (2013)
10. Ying, H., Lu, Q., Zhou, H., Lee, S.: A framework for constructing semantic as-is building energy models (BEMs) for existing buildings using digital images. In: ISARC, Berlin, Germany (2018)
11. Bazjanac, V.: Space boundary requirements for modeling of building geometry for energy and other performance simulation. In: 27th International Conference on CIB W78, Cairo, Egypt (2010)
12. International Organization for Standardization, ISO 10303-11: Industrial automation systems and integration—Product data representation and exchange—Part 11: Description methods: The EXPRESS language reference manual (2004). https://www.iso.org/standard/38047.html. Accessed 16 Mar 2018
13. Häfele, K.H., Liebich, T.: IFC2x3 space boundary implementation guide summary (2010)
14. BuildingSMART, IfcRelSpaceBoundary2ndLevel. http://www.buildingsmart-tech.org/ifc/IFC2x4/rc2/html/schema/ifcproductextension/lexical/ifcrelspaceboundary2ndlevel.htm. Accessed 14 Mar 2018
15. RDF, IFC Engine DLL. http://rdf.bg/ifc-engine-dll.php. Accessed 01 Dec 2017