

# Integration of BIM and GIS: IFC geometry transformation to shapefile using enhanced open-source approach

Junxiang Zhu<sup>a</sup>, Xiangyu Wang<sup>b,c,\*</sup>, Mengcheng Chen<sup>b</sup>, Peng Wu<sup>a</sup>, Mi Jeong Kim<sup>c</sup>

<sup>a</sup> School of Design and the Built Environment, Curtin University, Bentley 6102, Western Australia, Australia

<sup>b</sup> School of Civil Engineering and Architecture, East China Jiaotong University, Nanchang 330013, China

<sup>c</sup> Department of Housing and Interior Design, Kyung Hee University, Seoul, Republic of Korea

## ARTICLE INFO

### Keywords:

Building Information Modeling (BIM)  
Geographic Information System (GIS)  
Geometry transformation  
Shapefile  
Industry Foundation Classes (IFC)  
Open source  
3D

## ABSTRACT

Efficient data exchange is the foundation of effective integration of Building Information Modeling (BIM) and Geographic Information System (GIS). However, it has not been fully achieved. This study aims to enhance the previously developed Open-Source Approach (OSA), which is for geometry transformation of Industry Foundation Classes (IFC) into the shapefile format. OSA will be strengthened to be enhanced OSA (E-OSA) by developing an enhanced Automatic Multipatch Generation (E-AMG) algorithm that is capable of transforming various types of 3D representation, such as clipping and mapped representation, to boundary representation (Brep). The proposed method was tested and validated using a house model and two building models provided by Karlsruhe Institute of Technology (KIT) as well as a bridge model. The results show that E-AMG has the capacity to successfully transform these models from IFC to shapefile without building element loss. A comparison between Data Interoperability Extension for ArcGIS (DIA), Feature Manipulation Engine (FME) and the proposed E-OSA shows that the proposed method is more reliable than others while dealing with complex models. However, this method currently needs human intervention when handling multiple-clipping geometry. This issue should be addressed in the future to improve the effectiveness of E-OSA.

## 1. Introduction

The integration of Building Information Modeling (BIM) and Geographic Information System (GIS) has attracted enormous attentions from the Architecture, Engineering, Construction (AEC) domain and the geospatial section, due to the rising research interest in smart city [1], urbanization [2], Internet-of-Things [3], et al.

Integration of BIM and GIS can be conducted at several levels. Kang et al. [4] grouped relevant studies into five groups, i.e., schema-based approaches, service-based approaches, ontology-based approaches, process-based approaches, and system-based approaches. Amirebrahimi et al. [5] proposed a three-level framework, which suggests that BIM/GIS integration can be operated at data level, process level, and system level. Among all those types or levels of integration, data-level integration is the most fundamental form of integration that deals with the basic data exchange between the two diverse systems.

The main challenge in data-level BIM/GIS integration is efficient and effective data exchange [6]. If BIM and GIS could exchange data effectively, they can both benefit from each other [7,8]. Otherwise, it

can lead to compatibility difficulties for applications dedicated to specialized building functions [9]. Data exchange problem between BIM and GIS is principally caused by semantics mismatch between the AEC domain and geospatial industry, which can be reflected by data schemas used in these two fields, e.g., Industry Foundation Classes (IFC) for building information exchange in the AEC domain and City Geography Markup Language (CityGML) for representing virtual city models in the geospatial industry. For example, some building elements defined in IFC may not be defined in CityGML, such as stair and beam [7]. As a result, those undefined elements cannot be properly converted to CityGML. In addition to the semantics mismatch, issues regarding geometry transformation also made this problem worse, including wrong georeferencing of the transformed models [10,11] and mapping of Level of Detail (LoD) [12]. Both IFC and CityGML have defined five levels of detail [13], and LoDs have to be properly matched during the transformation.

To achieve efficient data exchange between BIM and GIS, many attempts have been made. Most of them were around IFC and CityGML and used semantic-based techniques to bridge the semantics gap

\* Corresponding author at: School of Civil Engineering and Architecture, East China Jiaotong University, Nanchang 330013, China.

E-mail addresses: [junxiang.zhu@curtin.edu.au](mailto:junxiang.zhu@curtin.edu.au) (J. Zhu), [Xiangyu.wang.perth@gmail.com](mailto:Xiangyu.wang.perth@gmail.com) (X. Wang), [mcchen@ecjtu.edu.cn](mailto:mcchen@ecjtu.edu.cn) (M. Chen), [peng.wu@curtin.edu.au](mailto:peng.wu@curtin.edu.au) (P. Wu), [mijeongkim@khu.ac.kr](mailto:mijeongkim@khu.ac.kr) (M.J. Kim).

<https://doi.org/10.1016/j.autcon.2019.102859>

Received 20 February 2019; Received in revised form 8 May 2019; Accepted 4 June 2019

Available online 19 June 2019

0926-5805/ © 2019 Elsevier B.V. All rights reserved.

between them by modifying current data schemas or creating new data schemas [8,14,15]. For example, Deng et al. [16] developed a reference ontology called Semantic City Model. Borrmann et al. [15] extended IFC for incorporating multi-scale representations of shield tunnels. Kang et al. [4] proposed a BIM/GIS-based information Extraction, Transform, and Load (BG-ETL) architecture. Mignard et al. [9] developed a semantic extension to BIM called Urban Information Modeling to take into account the management of urban elements contained in the building environment, and De laet et al. [17] developed the CityGML extension called GeoBIM to get semantic IFC data into a GIS context. However, these studies were project-specific.

The situation would be much simpler, if a non-semantic data format is used on the GIS side. In the previous case where CityGML is used, two issues need to be properly addressed, i.e., the mapping of classes or semantics matching and mapping of LoD [14,18], and substantial research effort has been put on these two problems. However, if a non-semantic data model is used, the main concern over data-level BIM/GIS integration would be geometry transformation and semantic information transfer.

In a previous study, to solve the geometry transformation problem, a non-semantic data schema, shapefile, is suggested, and Open-Source Approach (OSA) was developed to transform IFC into shapefile [19] and was tested using a bridge model. However, OSA only supports the transformation of swept solid, one of the five ways that IFC uses to represent 3D models, and have not been tested using other types of building model other than bridge. This study aims to enhance the OSA by enabling more types of geometry transformation such as clipping and mapped representation, which has been identified as a problem by Deng et al. [16]. The newly developed algorithm for geometry transformation will be referred to as Enhanced-AMG (E-AMG) in this study.

The rest of the paper is organized as follows: Section 2 introduces the research background and related works. Section 3 explains the methodology, including the general workflow for achieving geometry transformation, detailed workflows for the transformation of various 3D representations. Section 4 uses a house model to validate the proposed method. Section 5 discusses the findings of this paper, and Section 6 ends this paper with a conclusion.

## 2. Research background and related works

### 2.1. Three-dimensional representation methods

There are three basic methods for representing 3D objects, including Boundary Representation (Brep), Constructive Solid Geometry (CSG), and swept solid [7]. For Brep and swept solid, different terms are being used by studies. For example, Brep is also known as B-rep [4] or BRrep [16], while swept solid is also known as sweep volume [10] or sweeping [20].

Brep uses bounding surfaces to represent 3D objects. It is usually used for complex objects, such as stairs (*IfcStair*) and windows (*IfcWindow*). CSG uses the result of a series of Boolean operations (difference, union, and intersection) of simpler objects, or primitives, such as spheres, cones, pyramids, or cylinders to represent 3D objects. Swept solid uses a 2D surface and an extrusion path to define a solid; the path defines the route along which the surface is extruded. Example of Brep, swept solid, and CSG are shown in Fig. 1.

Brep, CSG, and swept solid are basic 3D modeling methods. In IFC, there are two more approaches for 3D representation that are variants of the basic methods, i.e., clipping and mapped representation.

A clipping geometry is the Boolean difference between a solid that is restricted to swept solid and a half space solid [22]. It can be considered as a simplified version of CSG, as only the difference operation is involved. A mapped representation is the use of an existing representation as a representation item in a second representation [23]. The mapping source can be any other types of representation, such as Brep and swept solid.

### 2.2. Building element hierarchy

IFC is a semantic data model, which defines elements contained in the AEC domain as well as the relationships between them. Fig. 2 represents the building element spatial hierarchy of IFC. According to IFC standard, there can only be one project, which may contain one or more sites. A site can contain one or more buildings, while a building can have one or more stories. All the building elements are attached to stories. This hierarchy is also referred to as IFC-tree [19].

Special attention needs to be paid to elements such as window, door, and roof because of their special spatial placement or definition. In terms of spatial hierarchy, doors and windows are elements attached to stories. However, in terms of spatial placement, doors and windows are placed relative to openings, which are further placed relative to walls. In contrast, all other building elements, such as beams, columns, members, and stairs, are placed relative to stories. Roof, as defined by IFC, could be an independent element or an abstract container that may contain slabs and beams. When roof is a container, it has no independent geometry and it is geometrically represented by the geometric representation of its components [24].

### 2.3. IFC geometry transformation and difficulties

Brep, CSG and sweeping are basic methods for solid modeling that are commonly used in computer-aided design (CAD), computer-aided engineering (CAE) [25,26] and BIM [27]. Like other computer-based techniques, CAD and BIM rely on computer graphics technique to process two-dimensional (2D) or three-dimensional (3D) models and display them on 2D screens [28,29]. To fulfil project aims, such as data exchange between different systems or visualization or both, representations may need to be converted to other forms. The conversion can be grouped roughly into two types, namely, within-industry conversion and inter-industry conversion. Within-industry conversion is for applications in the same domain or industry, such as different BIM systems in the AEC domain, while inter-industry conversion is for applications in different domains or industries, such as BIM systems in AEC domain and GIS systems in the geospatial industry.

In the area of solid modeling, transformation among Brep, CSG and sweeping (or within-industry conversion) has been studied for data exchange between different CAD/BIM systems, for example, the bidirectional conversion between CSG and Brep [30,31], despite that it turns out later that a conversion from Brep to CSG is usually impossible, as Brep models carry less information [32]. However, for the BIM/GIS inter-industry conversion, more aspects need to be considered. (1) Not only CSG and swept solid, Brep models need to be transformed as well, e.g., from Brep (IFC) to Brep (shapefile), and (2) building elements' spatial placement has to be adjusted to suit GIS, as BIM uses a relative placement system while GIS uses global coordinate system. Apart from these, (3) different challenges would arise when different receiving formats on the GIS side are used. For example, when shapefile is used, geometry transformation and semantics transfer are the main concerns [19]. However, when CityGML is used, the mapping of semantics and the mapping of level of details (LoD) should also be taken into account [12,16]. These issues do not have to be considered during the within-industry conversion, but they are what would distinguish inter-industry conversion from within-industry conversion.

Even though the within-industry conversion of representations, or geometry transformation, has been well solved in the solid modeling domain, the inter-industry conversion between BIM and GIS still remains a problem, which has been encountered by many researchers trying to integrating BIM and GIS, such as Borrmann et al. [15], Isikdag et al. [33,34] and Deng et al. [16]. The BIM-to-GIS conversion problem can be interpreted in two ways:

- (1) The unstable way using commercial tools. The geometry transformation from BIM to GIS can be completed using commercial tools

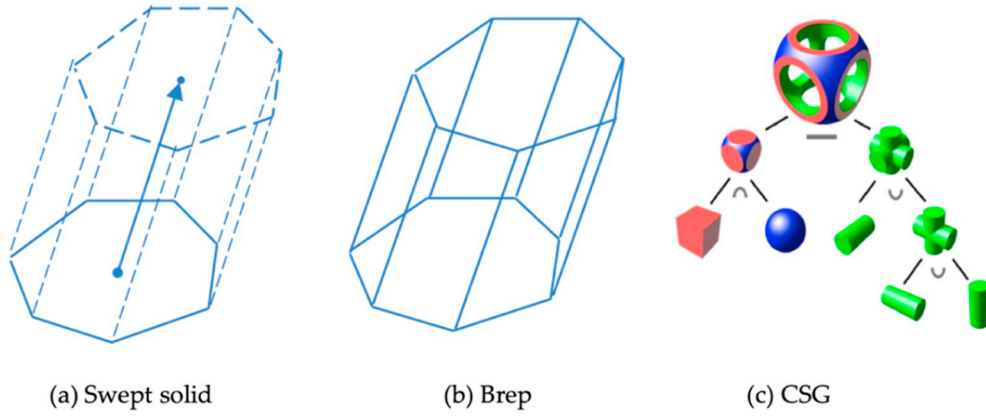


Fig. 1. Example of (a) Swept solid, (b) Brep, and (c) CSG [21].

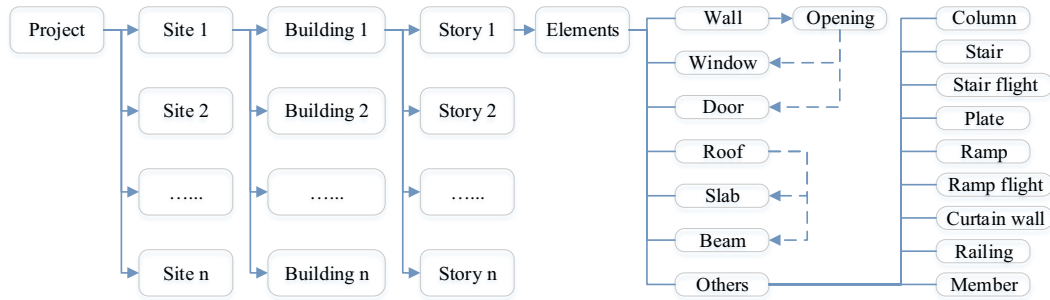


Fig. 2. IFC building element spatial hierarchy, IFC-tree.

such as Feature Manipulation Engine (FME) [35] and Data Interoperability Extension for ArcGIS (DIA) [36]. These tools transform information between different data schemas by mapping classes defined in them. When semantics mismatches exist, this approach is prone to fail or result in information loss. In addition, these commercial tools may crash when processing IFC models [11]. For example, as reported by Boyes et al. [6], geometry errors created during conversion can make FME fail to read IFC and crash.

- (2) Instead of using commercial tools, researchers also develop customized algorithms for geometry transformation. However, in this context, among the five types of 3D representation used by IFC, namely Brep, CSG, swept solid, clipping and mapped representation, only the transformation of swept solid has been thoroughly covered [16,19], while the transformation of CSG, clipping [16] and mapped representation still remains a problem.

### 3. Methodology

IFC have primarily five approaches to represent 3D geometry, including Brep, swept solid, clipping, CSG, and mapped representation. However, in practice, the methods most often used for buildings are Brep, swept solid, mapped representation and clipping. Therefore, this study will focus on the transformation of Brep, swept solid and clipping into Brep of shapefile.

#### 3.1. IFC geometric information extraction and transformation

The steps for IFC geometric information extraction and transformation are shown in Fig. 3. In the previous OSA, coordinate system transformation (CST) and Automatic Multipatch Generation (AMG) algorithm are two crucial operations for generating multipatch using parameters from IFC. In this study, AMG is to be strengthened to be Enhanced Automatic Multipatch Generation (E-AMG) to support transformation of more types of representation. The general workflow

for geometry transformation includes three stages, i.e., obtaining representation and placement of elements from IFC, extracting geometric parameters, and geometry regeneration.

The first step is to obtain the representation and placement of each element through the IFC element hierarchy or IFC-tree. Then depending on the type of representation, different approaches should be followed to acquire geometric parameters for the geometry regeneration. (1) For Brep representation, each ring (face) of a Brep object should be extracted and transformed through CST and E-AMG. (2) For swept solid representation, the sweep parameters, i.e., swept area, extrusion direction and extrusion depth, should be retrieved. After CST, E-AMG should be applied to turn these parameters into Brep. (3) For one-clipping representation, the procedure is similar to swept solid representation. The difference is that the extracted parameters are two rings instead of those sweep parameters. The E-AMG should then be applied to transform the two rings into Brep. (4) For multiple-clipping, each part involved in the clipping should first be turned into intermediate Brep individually, and then through the difference operation, one of the Boolean operations, the final Brep object can be obtained.

In this study, several Python packages will be used to perform CST and E-AMG, including IfcOpenShell for parsing IFC file, Pyshp for shapefile reading and writing, VPython for the generation of shape profiles (swept area) for swept solid, and Numpy for numbers and matrix processing.

#### 3.2. Coordinate system transformation

Since GIS and BIM use different coordinate systems and use coordinate system in different ways, CST is a must for geometry transformation from IFC to shapefile. The coordinates can be transformed using the equation as follows [19]:

$$[x' \ y' \ z'] = [x \ y \ z] \times (\begin{bmatrix} \vec{x} & \vec{y} & \vec{z} \end{bmatrix}^T)^{-1} + [x_0 \ y_0 \ z_0], \quad (1)$$

where  $x'$ ,  $y'$ , and  $z'$  are the transformed coordinates;  $x$ ,  $y$ , and  $z$  are the

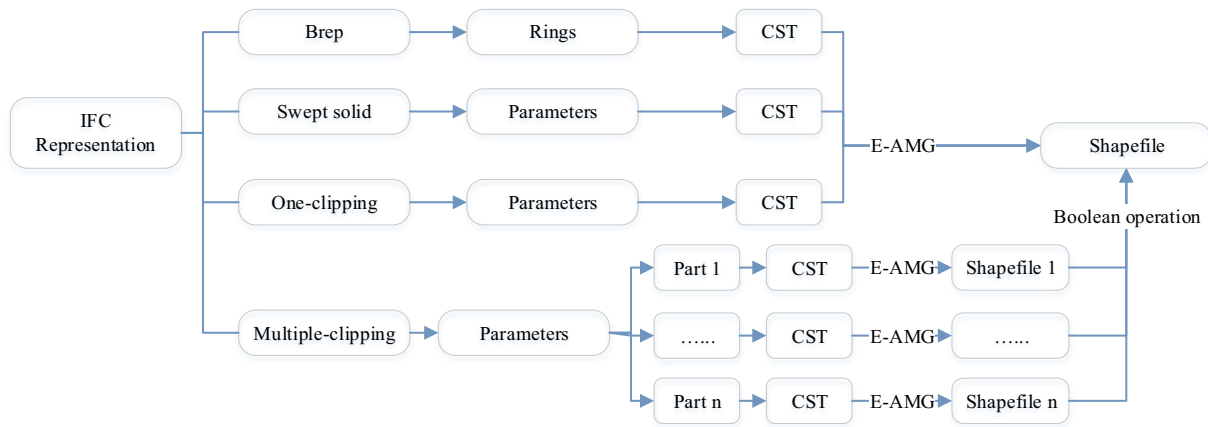


Fig. 3. General workflow for geometric information extraction and transformation.

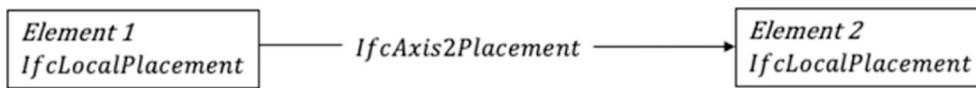


Fig. 4. Relationship between *IfcLocalPlacement* and *IfcAxis2Placement*.

initial coordinates;  $x_0$ ,  $y_0$ , and  $z_0$  are the origin shift,  $\vec{x}$ ,  $\vec{y}$  and  $\vec{z}$  are  $3 \times 1$  vectors indicating the direction of the initial x-axis, y-axis and z-axis respectively.

IFC uses a local placement system that elements are placed relative to other elements. For example, slabs are placed relative to stories and stories are placed relative to buildings. A placement chain can be formed from the end placement of element to the source placement of site. This placement chain is key to CST and can be obtained from the placement of element (*IfcLocalPlacement*). The placements of elements in different levels are connected by *IfcAxis2Placement*, as shown in Fig. 4.

The geometries of building elements must be transformed step by step from the element-level local coordinate system to the final site-level local coordinate system. An algorithm, shown in Table 1, has been developed to conduct the consecutive coordinate system transformation.

The input *Coordinates* refers to the original coordinates, *localPlacement* refers to the initial local placement of element, and the *Coordinates* returned at the end of the algorithm is the transformed coordinates. The logic behind the algorithm is that, keep searching the local placement objects contained in the initial local placement object and put them in a list named *localPlacements*, then perform the CST one by one according to the list to finally obtain the transformed coordinates. A door element is used as an example of consecutive coordinate system transformation in Fig. 5.

Through the unique identifier (id, 17302) of the placement of the

door element, all the rest placements in the placement chain can be identified. Their unique identifiers are indicated in boxes. The numbers above the arrow between boxes indicates the id of *IfcAxis2Placement*. In this case, the coordinates of a point of the door were turned into (7.41, 4.56, 0) from (0, 0, 0) after six consecutive CST.

### 3.3. Brep (IFC) to Brep (shapefile) transformation

Both IFC and shapefile can use Brep to store 3D geometry. The main task of Brep (IFC) to Brep (shapefile) transformation is CST and ensuring the transformed Brep meet the standard of shapefile. The detailed workflow for Brep transformation is represented in Fig. 6.

This workflow has three stages indicated by different colors, i.e., acquiring representation, extracting geometric parameters, and geometry regeneration. During the regeneration stage, after CST, rings extracted from IFC are, however, not qualified shapefile rings yet, because of the different standards used to define a closed ring or loop. In shapefile, the first point and the last point in a sequence of points must be the same to indicate a closed ring. However, in IFC it is not. Fig. 7 shows the difference of IFC and shapefile in defining a closed ring. To define a closed ring formed by  $P_1, P_2, P_3$  and  $P_4$ , IFC uses  $[P_1, P_2, P_3, P_4]$ , while shapefile uses  $[P_1, P_2, P_3, P_3, P_1]$ . Therefore, after retrieving a ring from IFC, the first point must be added to the end of the sequence to meet the requirement of shapefile.

Table 1

Pseudo Python codes for consecutive coordinate system transformation.

```
def keepTransform(Coordinates, localPlacement):
    localPlacements = []
    while localPlacement:
        localPlacements.append(localPlacement.RelativePlacement.id())
        localPlacement = localPlacement.PlacementRelTo
    for item in localPlacements:
        Coordinates = singleCoordinateTransformation(Coordinates)
    return Coordinates
```

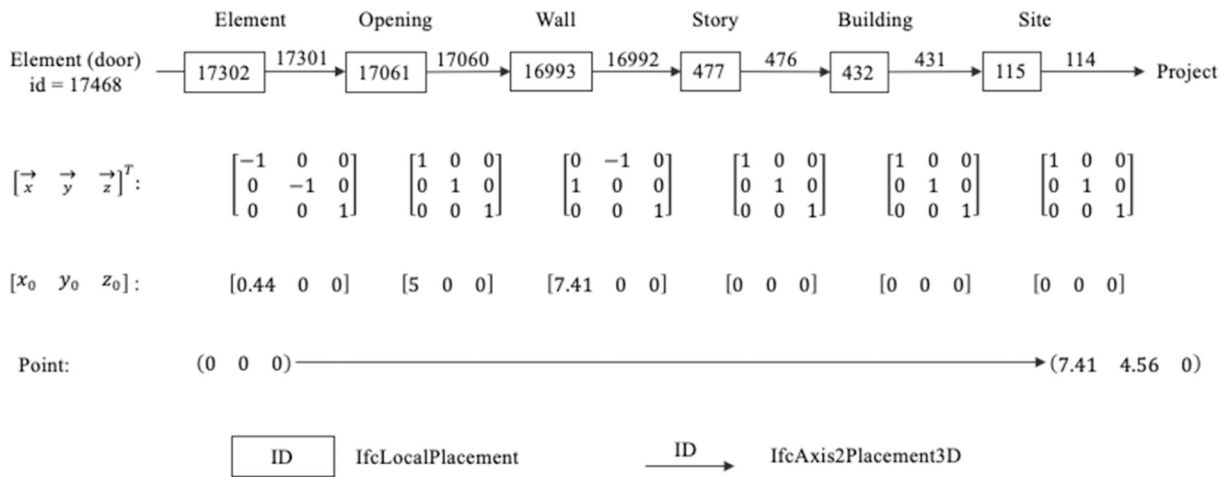


Fig. 5. Consecutive coordinates transformation of a point in door element.

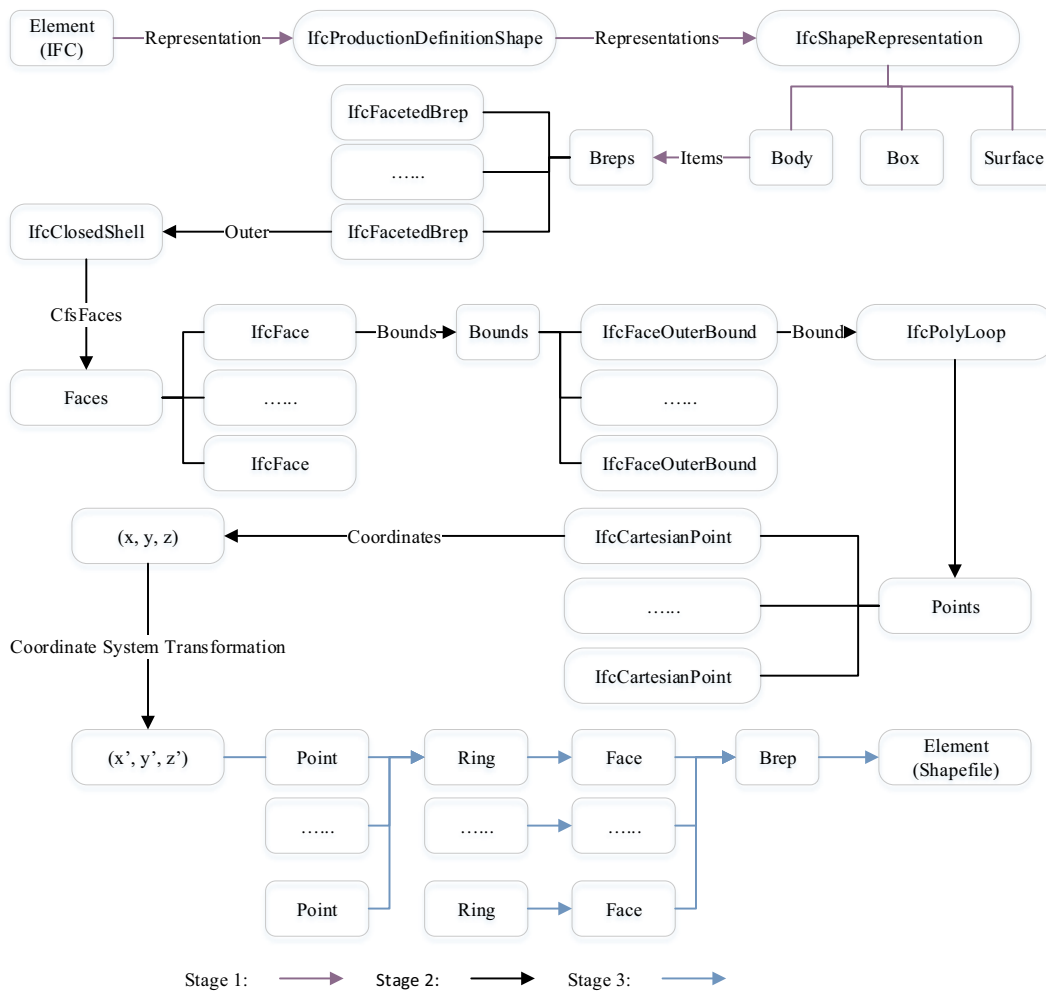


Fig. 6. Workflow for Brep (IFC) to Brep (shapefile) transformation.

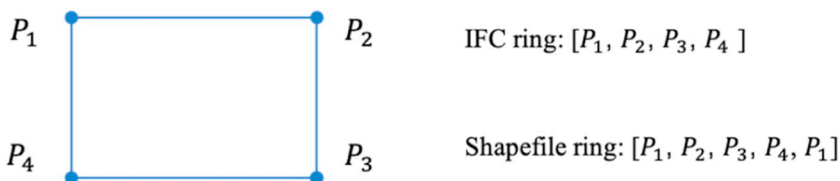


Fig. 7. Different definitions of a closed ring in IFC and shapefile.

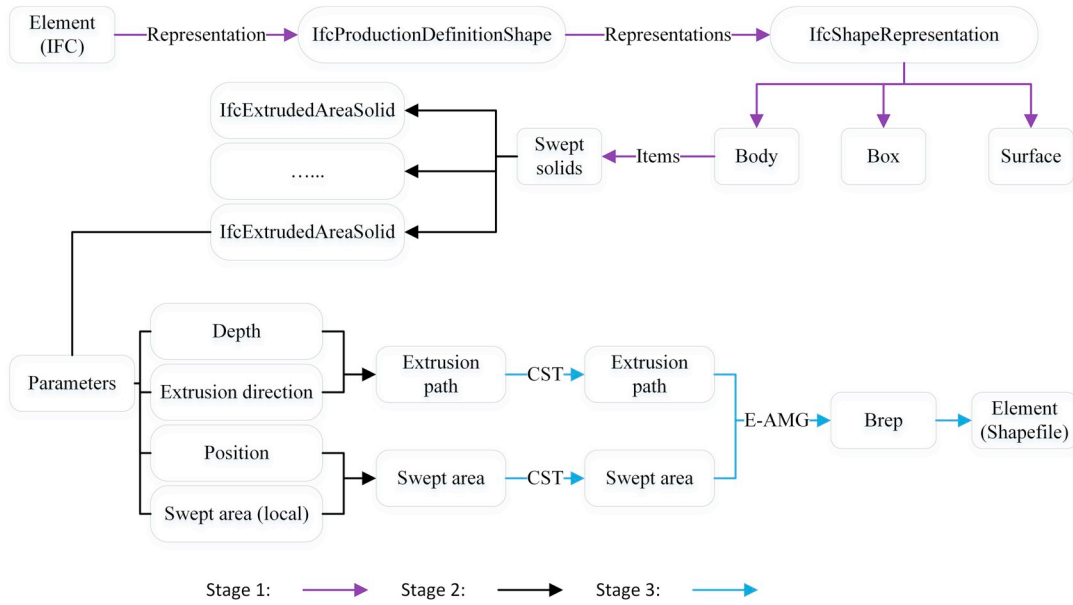


Fig. 8. Workflow for swept solid to Brep transformation.

### 3.4. Swept solid to Brep transformation

During the development of AMG, it is assumed that the initial ring order is clockwise, and the extrusion direction is along the positive-z axis. However, in situations where the initial ring order is unknown, a

more general algorithm is needed. In this study, the Enhanced AMG (E-AMG) algorithm is developed to handle any types of initial rings and extrusion path for the transformation of swept solid into Brep. The workflow for swept solid transformation is represented in Fig. 8. It is divided into three stages by different colors. The extracted and

**Table 2**  
Four basic forms of extrusion.

	Types	Extruded rings and points order
1		$R_1: P_1 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2 \rightarrow P_1$ $R_2: P_1 \rightarrow P_2 \rightarrow P_2' \rightarrow P_1' \rightarrow P_1$ $R_3: P_2 \rightarrow P_3 \rightarrow P_3' \rightarrow P_2' \rightarrow P_2$ $R_4: P_3 \rightarrow P_4 \rightarrow P_4' \rightarrow P_3' \rightarrow P_3$ $R_5: P_4 \rightarrow P_1 \rightarrow P_1' \rightarrow P_4' \rightarrow P_4$ $R_6: P_1' \rightarrow P_2' \rightarrow P_3' \rightarrow P_4' \rightarrow P_1'$
Clockwise ring extruded along positive-z (Type 1)		
2		$R_1: P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$ $R_2: P_1 \rightarrow P_1' \rightarrow P_2' \rightarrow P_2 \rightarrow P_1$ $R_3: P_2 \rightarrow P_2' \rightarrow P_3' \rightarrow P_3 \rightarrow P_2$ $R_4: P_3 \rightarrow P_3' \rightarrow P_4' \rightarrow P_4 \rightarrow P_3$ $R_5: P_4 \rightarrow P_4' \rightarrow P_1' \rightarrow P_1 \rightarrow P_4$ $R_6: P_1' \rightarrow P_4' \rightarrow P_3' \rightarrow P_2' \rightarrow P_1'$
Clockwise ring extruded along negative-z (Type 2)		
3		$R_1: P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$ $R_2: P_1 \rightarrow P_1' \rightarrow P_2' \rightarrow P_2 \rightarrow P_1$ $R_3: P_2 \rightarrow P_2' \rightarrow P_3' \rightarrow P_3 \rightarrow P_2$ $R_4: P_3 \rightarrow P_3' \rightarrow P_4' \rightarrow P_4 \rightarrow P_3$ $R_5: P_4 \rightarrow P_4' \rightarrow P_1' \rightarrow P_1 \rightarrow P_4$ $R_6: P_1' \rightarrow P_4' \rightarrow P_3' \rightarrow P_2' \rightarrow P_1'$
Counter-clockwise ring extruded along positive-z (Type 3)		
4		$R_1: P_1 \rightarrow P_4 \rightarrow P_3 \rightarrow P_2 \rightarrow P_1$ $R_2: P_1 \rightarrow P_2 \rightarrow P_2' \rightarrow P_1' \rightarrow P_1$ $R_3: P_2 \rightarrow P_3 \rightarrow P_3' \rightarrow P_2' \rightarrow P_2$ $R_4: P_3 \rightarrow P_4 \rightarrow P_4' \rightarrow P_3' \rightarrow P_3$ $R_5: P_4 \rightarrow P_1 \rightarrow P_1' \rightarrow P_4' \rightarrow P_4$ $R_6: P_1' \rightarrow P_2' \rightarrow P_3' \rightarrow P_4' \rightarrow P_1'$
Counter-clockwise ring extruded along negative-z (Type 4)		

**Table 3**  
Extrusion algorithms and their ring orders.

Extrusion type/algorithm	Ring order
Extrusion-1	$R_1: P_1 \rightarrow P_n \rightarrow P_{n-1} \rightarrow \dots \rightarrow P_3 \rightarrow P_2 \rightarrow P_1$
	$R_2: P_1 \rightarrow P_2 \rightarrow P_2' \rightarrow P_1' \rightarrow P_1$
	$R_3: P_2 \rightarrow P_3 \rightarrow P_3' \rightarrow P_2' \rightarrow P_2$
	.....
	$R_n: P_{n-1} \rightarrow P_n \rightarrow P_n' \rightarrow P_{n-1}' \rightarrow P_{n-1}$
Extrusion-2	$R_{n+1}: P_n \rightarrow P_{n+1} \rightarrow P_{n+1}' \rightarrow P_n' \rightarrow P_n$
	$R_{n+2}: P_1' \rightarrow P_2' \rightarrow P_3' \rightarrow \dots \rightarrow P_{n-1}' \rightarrow P_n' \rightarrow P_1'$
	$R_1: P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow \dots \rightarrow P_{n-1} \rightarrow P_n \rightarrow P_1$
	$R_2: P_1 \rightarrow P_1' \rightarrow P_2' \rightarrow P_2 \rightarrow P_1$
	$R_3: P_2 \rightarrow P_2' \rightarrow P_3' \rightarrow P_3 \rightarrow P_2$
	.....
	$R_n: P_{n-1} \rightarrow P_n \rightarrow P_n' \rightarrow P_{n-1}' \rightarrow P_{n-1}$
	$R_{n+1}: P_n \rightarrow P_n' \rightarrow P_{n+1}' \rightarrow P_{n+1} \rightarrow P_n$
	$R_{n+2}: P_1' \rightarrow P_n' \rightarrow P_{n-1}' \rightarrow \dots \rightarrow P_3' \rightarrow P_2' \rightarrow P_1'$

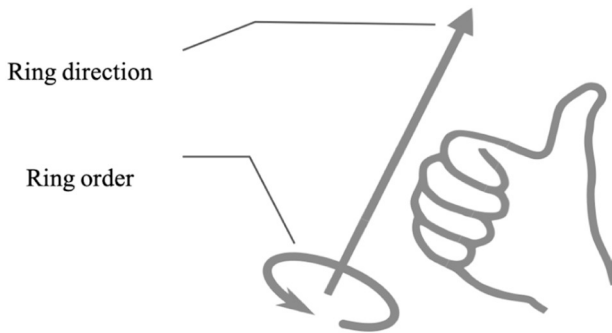


Fig. 9. Ring direction determined by right-hand rule [37].

transformed sweep parameter in the second stage can be used to regenerate Breps using E-AMG.

3.4.1. Basic types of extrusion

Depending on the initial order of the extrusion profile and extrusion direction, there are in total four possible combinations, namely (1) a clockwise ring extruded along a positive-z direction, (2) a clockwise ring extruded along a negative-z direction, (3) a counter-clockwise ring extruded along a positive-z direction, and (4) a counter-clockwise ring extruded along a negative-z direction. A simple rectangle with four vertices ( $P_1, P_2, P_3,$  and  $P_4$ ) is used as the extrusion profile to demonstrate the extrusions, which are listed in Table 2.  $P_1', P_2', P_3'$  and  $P_4'$  are extruded points from initial points  $P_1, P_2, P_3$  and  $P_4$  respectively.  $R_1, R_2, R_3, R_4, R_5$  and  $R_6$  are six rings generated using the initial and extruded points.

According to Table 2, these four basic extrusion forms only form two types of extrusion. Type 1 and Type 4 have the same set of ring orders, while Type 2 and Type 3 have the same one. For an initial ring with  $n$  vertices, the extruded rings can then be represented by either extrusion-1 or extrusion-2, as per Table 3. The main difference between extrusion-1 and extrusion-2 is the order of each ring. Extrusion-1 and extrusion-2 together form the swept solid transformation algorithm, as part of the E-AMG.

3.4.2. Determination of extrusion algorithm

To determine which extrusion algorithm is to be used, the relationship between the direction of ring and the extrusion direction has to be determined first. A definition of ring direction is given in this study. The direction of a ring can be determined by the right-hand rule, i.e., if right fingers are curled in the direction of ring order then the right thumb points in the direction of the ring (Fig. 9). Given the definition of ring direction, the extrusion-2 can be simply expressed as

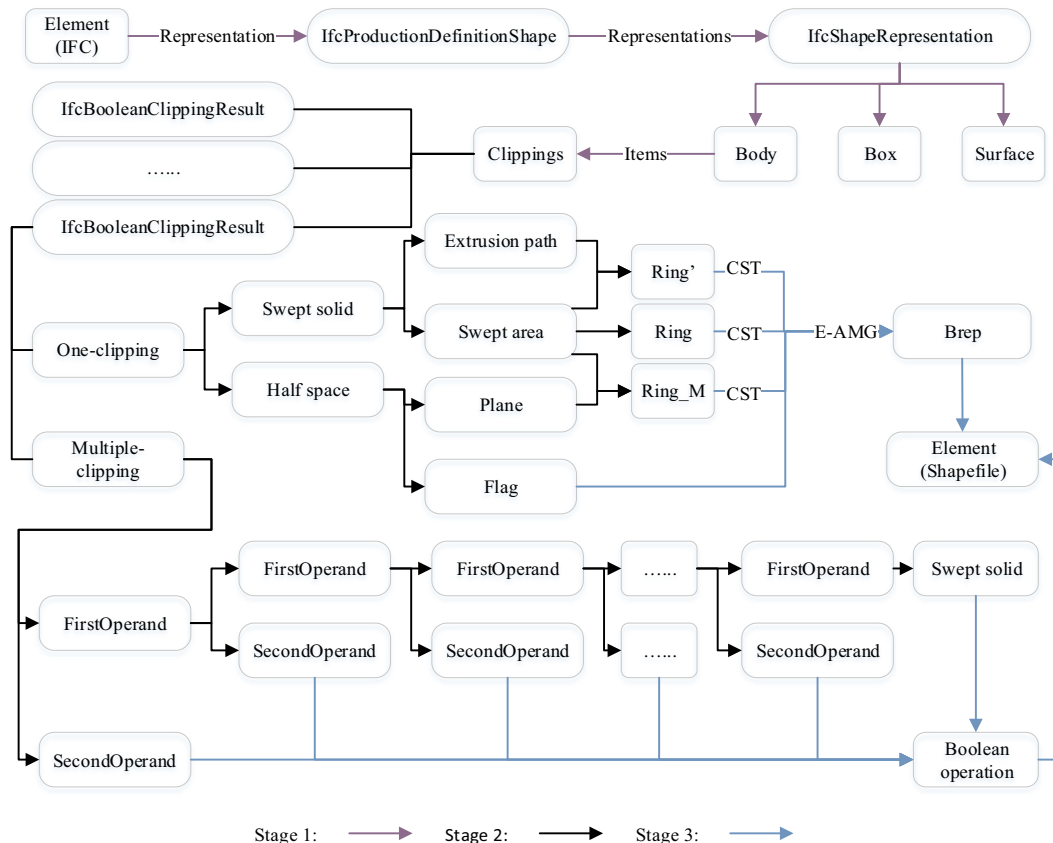


Fig. 10. Workflow for clipping to Brep transformation.

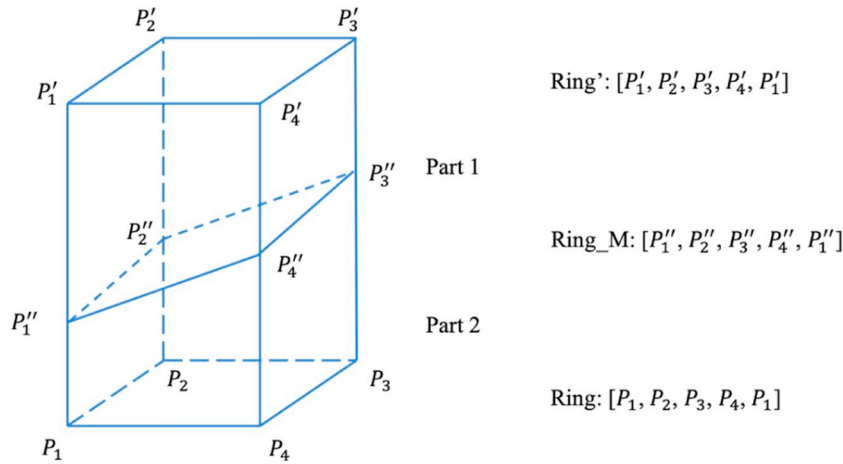


Fig. 11. One-clipping by an unbounded half space.

extrusion along the ring direction.

The ring direction can be indicated by the following equation:

$$D_r = \overrightarrow{P_i P_{i+1}} \times \overrightarrow{P_{i+1} P_{i+2}}, \quad (2)$$

where  $D_r$  is the ring direction,  $\overrightarrow{P_i P_{i+1}}$  is a vector pointing from  $P_i$  to  $P_{i+1}$ ,  $\overrightarrow{P_{i+1} P_{i+2}}$  is a vector pointing from  $P_{i+1}$  to  $P_{i+2}$ , and  $i \leq n - 2$  ( $n$  stands for the number of points in a ring).

Through  $D_r$  and extrusion path ( $D_p$ ), the type of extrusion algorithm to be used can be determined using the following ternary operation:

$$D_r \times D_p > 0? \text{ Extrusion 2: Extrusion 1.} \quad (3)$$

If  $D_r \times D_p > 0$ , which means the extrusion is along the ring direction, algorithm extrusion-2 should be used for Brep regeneration. Otherwise, algorithm extrusion-1 should be used.

### 3.5. Clipping to Brep transformation

The workflow for clipping to Brep transformation is shown in Fig. 10. Logically, as with Brep and swept solid, it contains three stages indicated by different colors.

The clipping method is a swept solid cut by one or more half spaces, either bounded or unbounded. Depending on the times of clipping used to form the final clipping geometry, it can be divided into two subtypes, i.e., one-clipping and multiple-clipping, that different approaches will be applied for each subtype.

#### 3.5.1. One-clipping by unbounded half space

Clipping by unbounded half space means a swept solid cut by an unbounded half space (*IfcHalfSpaceSolid*). *IfcHalfSpaceSolid* is defined by a plane, cutting the space into two, and a *flag* indicating which part is to be retained.

An example of one-clipping by an unbounded half space is represented in Fig. 11. To regenerate Brep, three rings should be obtained, i.e., the initial ring (*Ring*), the extruded ring (*Ring'*), and the ring

formed by intersections of the extrusion lines and the cutting plane (*Ring\_M*). The algorithms used in swept solid transformation can be slightly modified to generate Brep using two of the three rings mentioned above, either *ring* and *ring\_M* or *ring\_M* and *ring'*, depending on the value of *flag*.

An unbounded plane, which is the cutting plane, can be represented by a normal direction to it and a point on it [38], and lines in 3D space can also be represented by symmetric equations [39], then the intersections can be obtained by the following equation:

$$[X \ Y \ Z] = [x \ y \ AX_0 + BY_0 + CZ_0] \times \left( \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A & B & C \end{bmatrix}^T \right)^{-1}, \quad (4)$$

where  $(X \ Y \ Z)$  are coordinates of intersections,  $[A \ B \ C]$  is a normal direction to the cutting plane,  $(X_0 \ Y_0 \ Z_0)$  is a point on the cutting plane, and  $(x \ y)$  are coordinates of points forming the swept area. Take Fig. 11 for example,  $(x \ y)$  are coordinates of  $P_1, P_2, P_3,$  and  $P_4$ ,  $(X \ Y \ Z)$  are coordinates of  $P''_1, P''_2, P''_3$  and  $P''_4$ . The deduction process is represented in Appendix A.

#### 3.5.2. One-clipping by polygonal bounded half space and multiple-clipping

In one-clipping by polygonal bounded half space, a swept solid is cut by a bounded plane defined by *IfcPolygonalBoundedHalfSpace*. An example is represented in Fig. 12.

In this case, the clipping result cannot be directly obtained, instead, the swept solid and polygonal bounded half space have to be first extracted and transformed individually into Brep, then the Boolean operation, difference, should be applied to obtain the final clipping geometry.

Similarly, in the case of multiple-clipping transformation, each part involved in the clipping has to be generated individually, and then the difference operation should be applied one by one to produce the final clipping geometry.

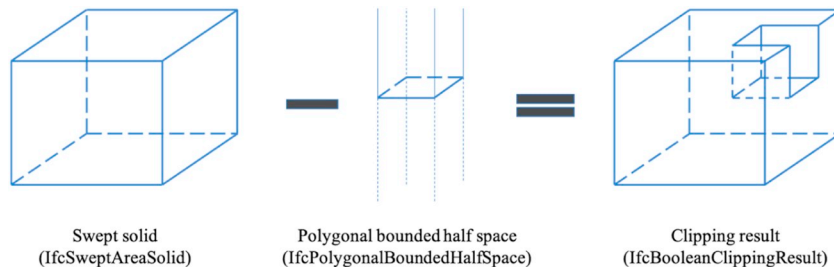
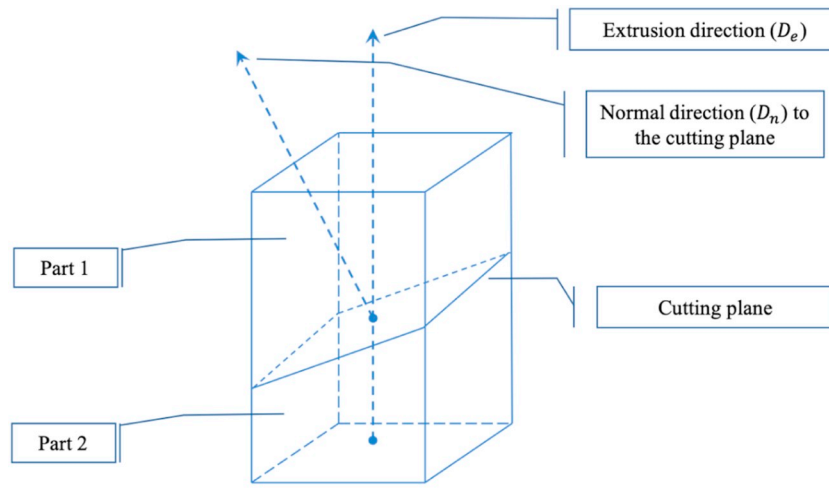


Fig. 12. One-clipping by polygonal bounded half space.





$$\text{If } (D_e \times D_n > 0) = \text{flag: retain Part1, Else: retain Part 2}$$

Fig. 13. Determining the part to be retained using extrusion direction  $D_e$ , normal direction to the cutting plane  $D_n$ , and *flag*.

### 3.5.3. Determining the to-be-retained part

During the regeneration of Brep from one-clipping, there is a need to determine which part is to be retained, since a cutting plane cuts the swept solid into two parts. As shown in Fig. 13, a swept solid is separated into two parts by the cutting plane, i.e., part 1 and part 2.

The part to be retained is depending on three factors, including swept solid extrusion direction ( $D_e$ ), normal direction ( $D_n$ ) to the cutting plane, and the *flag* of the cutting plane which is a Boolean (*true* or *false*). If *flag* is *true*, the part that the normal direction points away from is to be discarded [40]. It can be expressed using the following ternary operation:

$$(D_e \times D_n > 0) = \text{flag? Part 1: Part 2} \quad (5)$$

which means, if  $(D_e \times D_n > 0) = \text{flag}$ , then part 1 is to be retained, otherwise, part 2 is to be retained.

### 3.6. Transformation of mapped representation

A mapped item is the use of an existing representation (the mapping source – mapped representation) as a representation item in a second representation [41]. It contains two parts, i.e. mapping source and mapping target. Mapping source is one of the other 3D representation types, such as swept solid and Brep, and mapping target is constraint to be a Cartesian transformation operator for coordinate system transformation.

The transformation of mapped representation is essentially a transformation of geometry defined in the mapping source. The structure of *IfcMappedItem* is shown in Fig. 14. Once the mapping source has been identified, the strategy for handling Brep, swept solid or clipping can be applied accordingly.

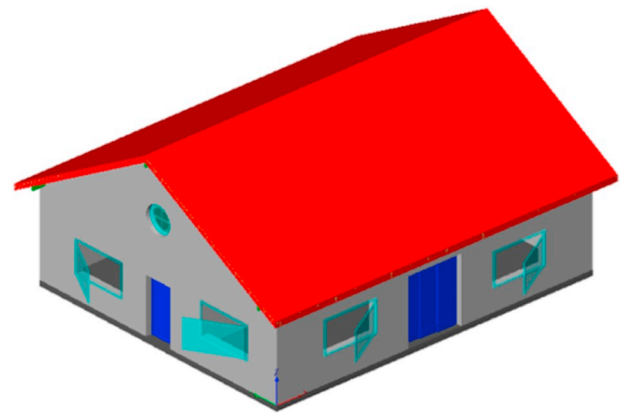


Fig. 15. House model (IFC) visualized using FZK Viewer.

## 4. Validation and results

### 4.1. Transformation of house model

A house model (Fig. 15), provided by Institute for Applied Computer Science at the Karlsruhe Institute of Technology (KIT), was used to test and refine the proposed method. This house model is open to access and can be used unrestrictedly [42]. This two-story house has all the basic elements, including windows, doors, beams, members, walls, floors, roof, stair and railings.

This model has been analyzed using a Python script written to analyze IFC structure. The representation type and quantity of each type are listed in Table 4.

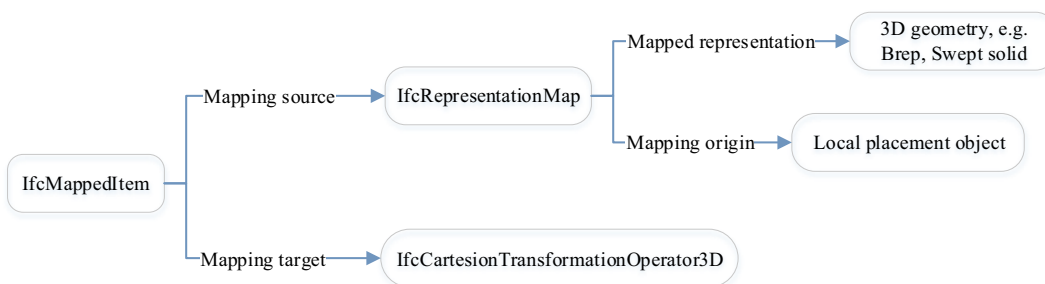


Fig. 14. Structure of *IfcMappedItem*.

**Table 4**  
Representation type and quantity of each element in the house model.

Type	Representation type	Quantity
IfcBeam	Mapped representation	3
	Swept solid	1
IfcDoor	Brep	3
	Mapped representation	2
IfcMember	Mapped representation	42
IfcRailing	Brep	2
IfcSlab	Swept solid	4
IfcStair	Brep	1
IfcWallStandardCase	Clipping	4
	Swept solid	9
IfcWindow	Brep	2
	Mapped representation	9

All the transformed elements are represented in Fig. 16. Most of these elements can be generated automatically using scripts, including members, railings, beams, windows, doors and stair. However, walls and slabs need additional intervention due to the presence of openings in them.

4.1.1. Extraction of openings

Opening represents a void within any element that has physical manifestation. Openings are essential to obtain proper wall and slab representations. In this case, as shown in Fig. 17, there are 17 openings in total, including 5 openings for doors (yellow), 11 openings for windows (light blue), and 1 opening for floor (orange). All the openings in IFC are modeled using swept solids.

4.1.2. Regeneration of walls

In the house model, walls are represented using two types of representation, i.e., swept solid and clipping. Different strategies should be applied to extract walls depending on their representation types. Walls created by swept solid are referred to as basic wall ( $Wall_1$ ), walls created by one-clipping are referred to as one-clipping wall ( $Wall_2$ ), and those created by multiple-clipping are referred to as multiple-clipping wall ( $Wall_3$ ), and the final walls are referred to as  $Wall_{final}$ .

(1) Basic walls

The basic walls are those modeled using swept solid. They can be transformed directly into shapefile using the E-AMG algorithm, which are represented in Fig. 18.

(2) One-clipping walls



Fig. 17. Openings for windows (light blue), doors (yellow) and floor (orange). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Fig. 18. Basic walls ( $Wall_1$ ) modeled using swept solid.

One-clipping walls are those modeled using clipping where a swept solid is cut only by one half space solid. They can be transformed directly using the E-AMG algorithm. The process of one-clipping is represented in Fig. 19, where a swept solid is cut by a plane and the lower part is retained. There are totally two pieces of walls modeled using this method in the house model.

(3) Multiple-clipping walls

Multiple-clipping walls, referred to as  $Wall_3$  in this paper, are those modeled using clipping, where a swept solid is cut by more than one half space solids. Fig. 20 shows the process of producing  $Wall_3$ , in which

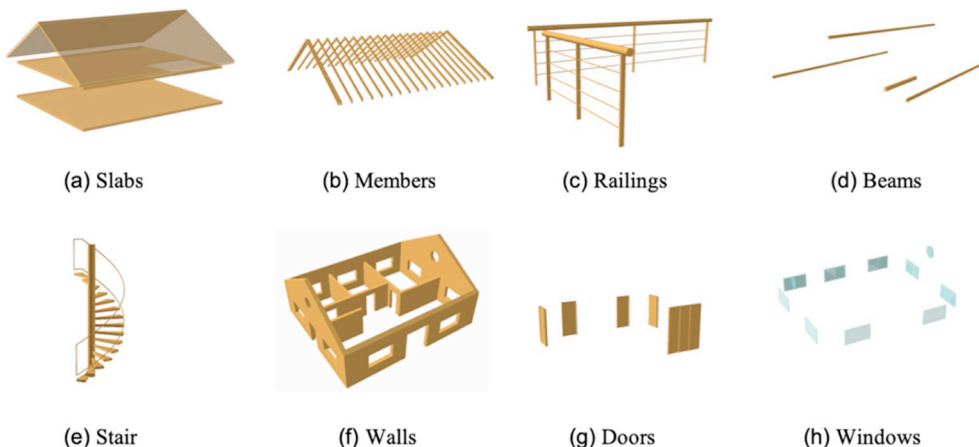


Fig. 16. Transformed elements, including (a) slabs, (b) members, (c) railings, (d) beams, (e) stair, (f) walls, (g) doors and (h) windows.

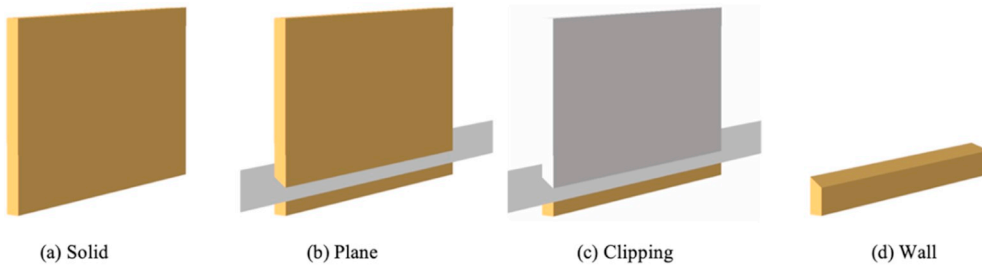


Fig. 19. One-clipping walls generated by a swept solid cut by a half space solid.

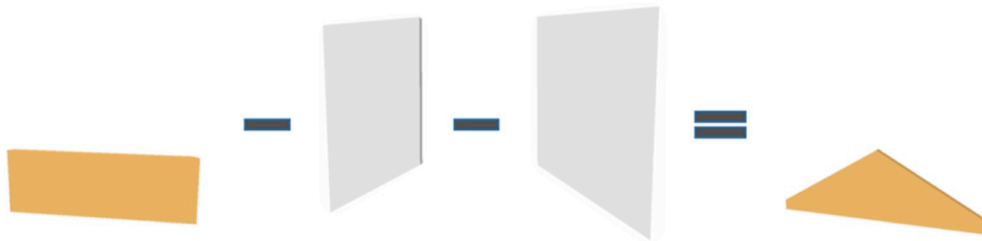


Fig. 20. Multiple-clipping walls generated by a swept solid cut by a polygonal bounded half space.



Fig. 21. Final walls generated by difference operation.



Fig. 22. Final slabs generated by difference operation.

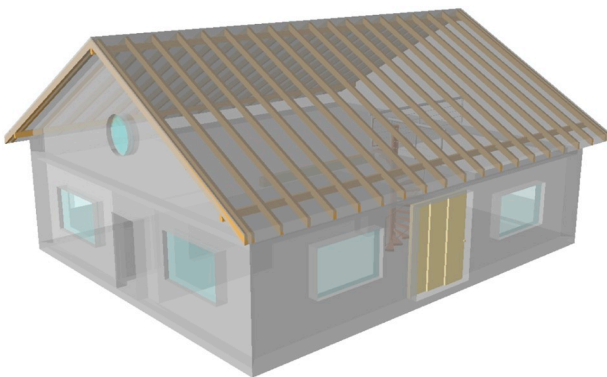


Fig. 23. The final transformed house model visualized in ArcScene.

a swept solid (yellow) is cut twice by two solids (white). There are totally two pieces of walls in this house model created using this method.

(4) Final walls ( $Wall_{final}$ )

The final walls are those with openings and ready to be part of the final transformed house model. They can be obtained through the following Boolean operations:

Table 5

Quantity of entity before and after transformation.

Element	Quantity		
	IFC	Shapefile	IFC = Shapefile
Beam	3 (mapped representation) 1 (swept solid)	4 (Brep)	Yes
Door	3 (Brep)	5 (Brep)	Yes
Member	2 (mapped representation)		
Railing	42 (mapped representation)	42 (Brep)	Yes
Slab	2 (Brep)	2 (Brep)	Yes
Slab	4 (swept solid)	4 (Brep)	Yes
Stair	1 (Brep)	1 (Brep)	Yes
Wall	4 (clipping) 9 (swept solid)	13 (Brep)	Yes
Window	2 (Brep) 9 (mapped representation)	11 (Brep)	Yes

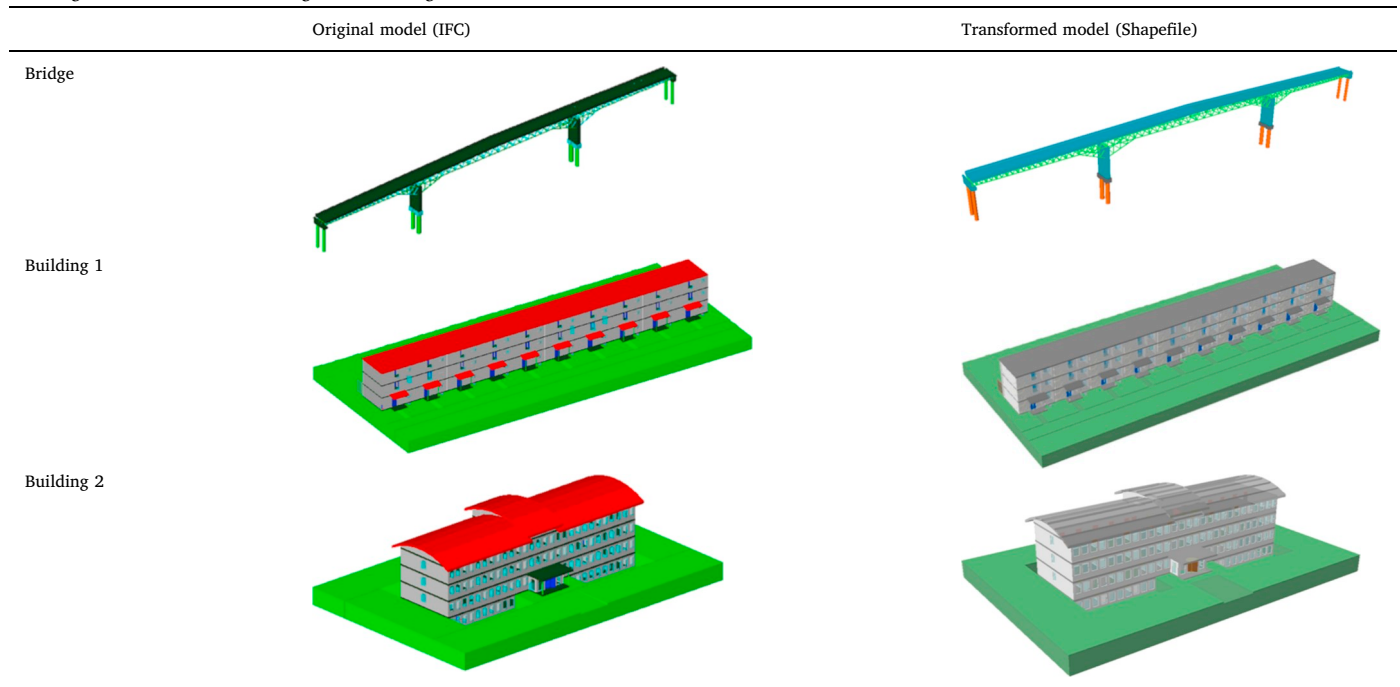
$$Wall_{final} = (Wall_1 \cup Wall_2 \cup Wall_3) - Openings.$$

Fig. 21 shows the process of producing the final walls, which is the union of  $Wall_1$ ,  $Wall_2$  and  $Wall_3$  subtracted by the openings.

4.1.3. Regeneration of slabs

Semantically, slabs contain two types of element, i.e., floor and roof. The final slabs can be obtained using a way similar to walls. The process of producing the final slabs are represented in Fig. 22.

**Table 6**  
The original and transformed bridge and building models.



**Table 7**  
Quantity of building element before and after transformation.

Model	Element	Quantity		
		IFC	Shapefile	IFC = Shapefile
Bridge	Beam	195	195	Yes
	Column	11	11	Yes
	Footing	6	6	Yes
	Member	380	380	Yes
	Slab	16	16	Yes
Building 1	Beam	10	10	Yes
	Column	20	20	Yes
	Door	170	179	Yes
	Railing	120	120	Yes
	Slab	120	120	Yes
	Stair	30	30	Yes
	Wall	281	281	Yes
Building 2	Window	80	80	Yes
	Column	2	2	Yes
	Door	77	77	Yes
	Railing	12	12	Yes
	Slab	26	26	Yes
	Stair	4	4	Yes
	Wall	121	121	Yes
	Window	206	206	Yes

4.1.4. The final transformed house model

By assembling all the building components, i.e., slabs, beams, members, walls, windows, doors, stair and railing, the final transformed house model can be obtained. The final house model is visualized in ArcScene where the appearance of each element can be set individually (Fig. 23). By setting the walls, windows and slabs to be transparent, both exterior and interior structure of the house can be revealed.

The quantity of each building element is used to determine the quality of transformation. All geometries have been transformed successfully from IFC to shapefile, and the quantity of each element remains the same. For example, there are 5 doors in the original IFC model, i.e., 3 doors represented by Brep and 2 doors represented by mapped representation, and after transformation, there are 5 doors in the shapefile model. The details are represented in Table 5.

4.2. Validation using other models

Another two building models from KIT, referred to as building 1 and building 2 in this study, and one bridge model were used to validate the E-AMG, and the result shows E-AMG can successfully transform those models. Table 6 shows the original and transformed bridge and building models. IFC models and shapefile models are visualized in FZK Viewer and ArcScene, respectively.

The quantity of each building element before and after transformation is shown in Table 7. No building element was lost during the transformation.

5. Discussion

5.1. Comparison between IFC entity and shapefile entity

A side-by-side comparison between IFC components and shapefile components is performed and shown in Table 8 using the house model. To have a better comparison, the appearances of shapefile components are set as close to IFC components as possible.

In terms of geometry and visualization, there is no obvious difference between IFC and shapefile, except the doors and windows. The windows and doors in IFC are more vivid than those in shapefile that doors and windows in IFC can be open or close, but in shapefile, all doors and windows are closed.

5.2. Supported and validated representations

In IFC, there are 16 defined building elements, and each building elements supports one or more types of representation [43]. For example, IfcBeam supports the use of swept solid, clipping and mapped representation, and IfcSlab supports swept solid, Brep, clipping and mapped representation. The supported representations by each building element and those validated in this study are listed in Table 9.


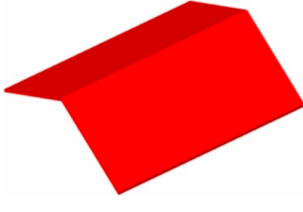


Among the 16 building elements defined by IFC plus the opening element, 12 of them have been tested, such as beam, door, member, opening, railing, slab, wall and window. IfcRoof was not directly tested, but in this case, it was represented by slabs which has been validated.

**Table 8**  
Comparison between IFC entity and shapefile entity.

Element	Shapefile	IFC
Windows		
Doors		
Walls		
Stair		
Beams		
Members		
Floor		

(continued on next page)

**Table 8** (continued)

Element	Shapefile	IFC
Roof		
Railings		

**Table 9**  
Supported and validated representations.

IFC type	Validated?	Supported representation	Validated representation
IfcBeam	Yes	Swept solid Clipping Mapped representation	Swept solid Mapped representation
IfcDoor	Yes	Swept solid Brep Mapped representation	Brep Mapped representation
IfcMember	Yes	Swept solid Clipping Mapped representation	Mapped representation
IfcOpeningElement	Yes	Swept solid Brep Mapped representation	Swept solid
IfcRailing	Yes	Brep Mapped representation	Brep
IfcSlab	Yes	Swept solid Clipping Brep Mapped representation	Swept solid
IfcStair	Yes	Brep or representation of its components	Brep
IfcWallStandardCase	Yes	Swept solid Clipping	Swept solid Clipping
IfcRoof	Yes	Brep or representation of its components	Swept solid
IfcWindow	Yes	Swept solid Brep Mapped representation	Brep Mapped representation
IfcPlate	No	Swept solid Clipping Brep Mapped representation	Not applicable
IfcWall	Yes	Swept solid Clipping Brep Mapped representation	Brep Mapped representation
IfcColumn	Yes	Swept solid Clipping Mapped representation	Swept solid
IfcRamp	No	Brep or representation of its components	Not applicable
IfcRampFlight	No	Swept solid	Not applicable
IfcStairFlight	No	Brep	Not applicable
IfcCurtainWall	No	Brep Mapped representation	Not applicable

Elements such as plate, ramp, ramp flight, stair flight and curtain wall were not included in these models and thus were not validated, however, the representation types they support have all been validated, i.e., swept solid, clipping, Brep and mapped representation. Therefore, it can be inferred that the geometry of these elements can also be transformed using the proposed approach.

### 5.3. Comparison between DIA, FME and E-OSA

To assess the performance of the proposed method, E-OSA is compared with DIA and FME in transforming the same models used in this study. For DIA, a set of parameters, defining what information to extract from IFC and how to extract, need to be determined before transformation, and in this study the settings recommend by Kuehne were used to obtain the most successful outputs [44]. For FME, a data conversion workflow was developed for each building element. Fig. 24 represents the conversion workflow using the transformation of beam for example.

The experiment results are listed in Table 10, which shows the performance of E-OSA outshines both DIA and FME. For the four models used in this study, DIA and FME can only transform the bridge model and failed to convert the house and building models, while E-OSA managed to successfully process all the models. DIA and FME failed to transform the house and building models as they both crashed during operation. The crash issue of FME was also reported by Boyes et al. [6]

### 5.4. Over-detailed entities

After transformation, some building element entities have only one geometry, e.g., wall, slab and member, which means an object is represented by one geometry only. However, some entities have more than one geometry, such as doors, windows and stairs. For example, a door entity (id = 31079) has 32 components in total, including one door frame, two handle locks that each has two constituents, and three panels that each has nine constituents.

These over-detailed elements greatly increase the model size and can be troublesome for both visualization and spatial analysis. Methods are needed to simplify them for better visualization efficiency and, if they are to be used in web-based application, better data transfer efficiency. Table 11 lists the types of over-detailed windows, railings, doors and stair, as well as the quantity of geometries contained in them, and Fig. 25 represents their appearances.

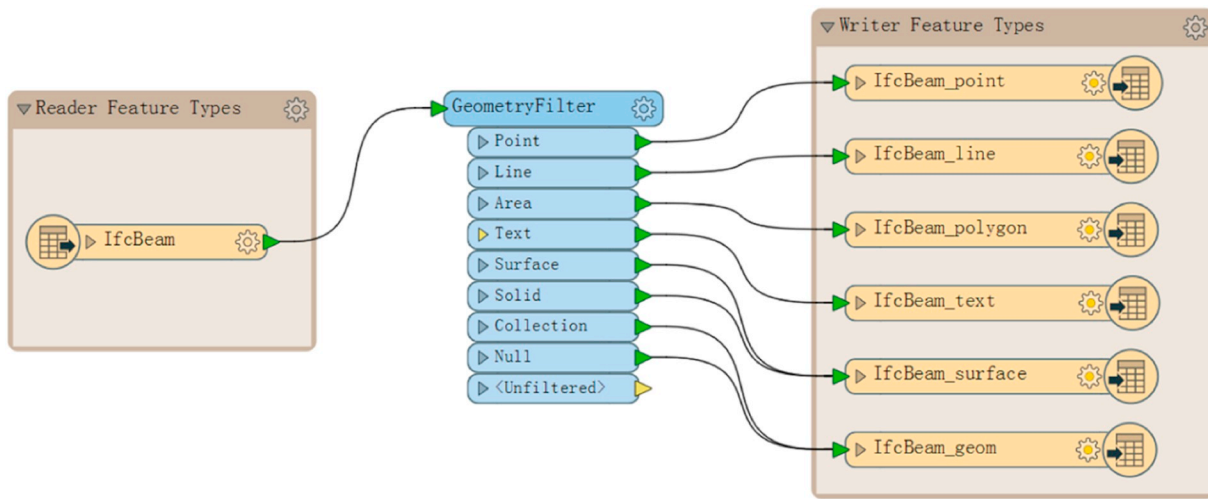


Fig. 24. The data conversion workflow used by FME, taking the conversion of beam for example.

**Table 10**  
Comparison between DIA, FME and E-OSA.

Model	Transformation completed?		
	DIA	FME	E-OSA
Bridge	Yes	Yes	Yes
House	No	No	Yes
Building 1	No	No	Yes
Building 2	No	No	Yes

**Table 11**  
List of over-detailed entities.

Element type	ID	Quantity of geometry
Window 1	23024, 23944, 27833, 28113, 31818, 32098, 32829, 33109, 33389	7
Window 2	66459, 74280	11
Railing	49831, 59075	12
Door 1	27013	14
Door 2	31079	32
Stair	14502	17

5.5. Comparison between shapefile and CityGML

Shapefile is used as recipient of BIM information in this study, while CityGML is another prominent data schema for BIM/GIS integration. A comparison between shapefile and CityGML is presented in Table 12.

Shapefile is a non-semantic data model developed by ESRI in 1998, which supports 2D and 3D geospatial data. In terms of 3D representation, it adopted Brep for solid modeling indicated by the data type “multipatch”. As a native format of GIS, it can exchange data easily with other non-GIS 3D software packages, such as Collaborative Design Activity (COLLADA) and SketchUp [45]. In contrast, CityGML is a semantic data model submitted primarily by Autodesk, Inc. and Bentley Systems, Inc. to the Open Geospatial Consortium (OGC), which is an international not for profit organization committed to making quality open standards for the global geospatial community [46].

Shapefile and CityGML have their own advantages over each other in BIM/GIS integration. The advantages of shapefile include: (1) no need to map element definitions, (2) no geometry loss caused by semantics mismatch, and (3) no need to match LODs. The advantage of CityGML is that it is a prominent data format for bidirectional information exchange between BIM and GIS, while shapefile can only be used for one-way transformation from BIM to GIS.

There are concerns over the capacity of shapefile for handling complex 3D geometry. This study has revealed that shapefile is capable of holding BIM models, even those complex entities such stairs, doors and windows.

5.6. Comparison between E-AMG and AMG

In this study, the previous developed AMG was enhanced to be E-AMG. AMG was developed only to transform swept solid to Brep using sweep parameters, i.e., swept area and extrusion path. E-AMG has the capacity to transform more types of representation. Apart from the swept solid, Brep, clipping and mapped representation can also be transformed. In addition, a new multipatch generation algorithm has been added to E-AMG that uses two initial rings to produce Brep instead of the sweep parameters. The differences between E-AMG and AMG in supported representation types are shown in Table 13. E-AMG is more advanced than AMG and can be used to process building models that are more complicated than the bridge model used during the validation of AMG.

5.7. Potential solution for the transformation of CSG

Transformation of CSG is not studied in this paper, however, a possible solution for its transformation is inspired by the transformation of multiple-clipping to Brep. In the transformation of multiple-clipping, all the relevant parts were generated individually, and the final geometry was obtained through Boolean operation. In the case of CSG, the potential solution constitutes three steps. (1) The first step is on the GIS side to realize parameterized modeling of basic solid objects (primitives) such as cuboids, cylinders, prisms, pyramids, spheres and cones, since these primitives are foundation of CSG; (2) the second step is to obtain the specific parameters of primitives from IFC and the CSG trees which indicate the relationship between each object; (3) the last step is to perform Boolean operations, such as union, difference and intersection, to obtain the final geometry.

6. Conclusions

This study enhanced the previously developed OSA for transforming IFC geometry into shapefiles. In the enhanced approach, an E-AMG algorithm has been developed to generate multipatches (Brep) from more types of representation from IFC, such as Brep, swept solid, clipping and mapped representation in a more reliable way. A house model, a bridge model and two building models have been used to test and validate the proposed method, and the main findings are as follows:

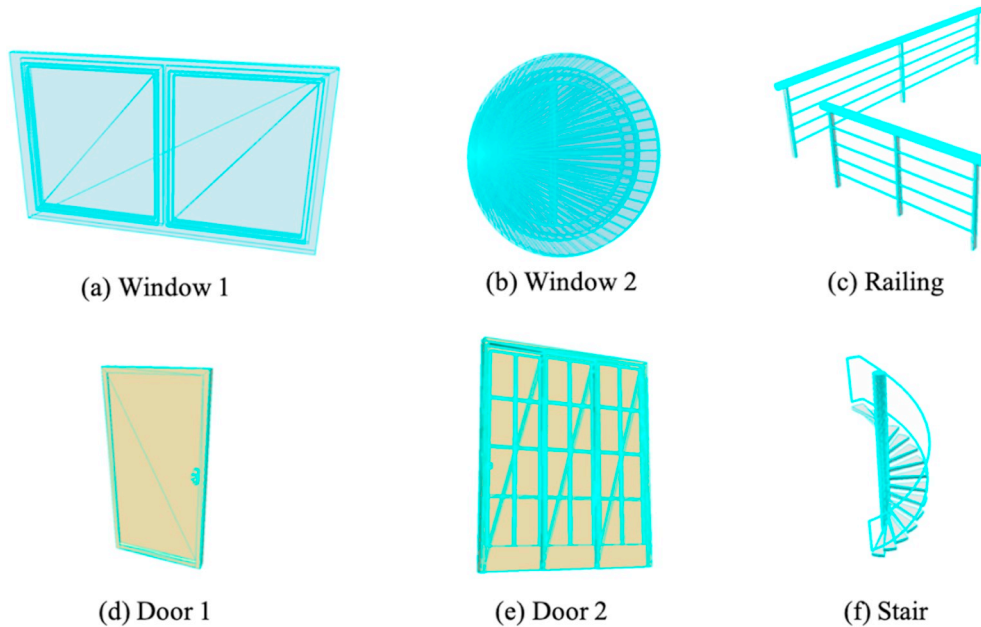


Fig. 25. Examples of over-detailed elements, including (a) window 1, (b) window 2, (c) railing, (d) door 1, (e) door 2 and (f) stair.

Table 12  
Comparison between CityGML and shapefile.

		CityGML	Shapefile
Year of issue		2008	1998
Developers		Primary submitter: Autodesk, Inc., Bentley Systems, Inc. Approver: OGC	Environmental Systems Research Institute, Inc. (ESRI)
Semantic data schema?		Yes	No
Geometry type		3D	2D/3D
Representation type		Surface model	Solid model
Open standard?		Yes	Yes (proprietary but open)
BIM/GIS integration	Semantics mapping	Yes	No
	LoD mapping	Yes	No
	Information exchange	Bidirectional	Unidirectional (from BIM to GIS)

Table 13  
Supported representation types by AMG and E-AMG.

Type	Subtype	Supported?	
		AMG	E-AMG
Swept solid	Arbitrary profile	Yes	Yes
	Rectangle profile	Yes	Yes
	Circle profile	Yes	Yes
Brep	Single Brep	No	Yes
	Multiple Brep	No	Yes
Clipping	One-clipping by half space	No	Yes
	One-clipping by polygonal bounded half space	No	Yes
	Multiple-clipping by half space	No	Yes
	Multiple-clipping by polygonal bounded half space	No	Yes
Mapped representation	/	No	Yes
CSG	/	No	No

- (1) E-OSA is more reliable in transforming complex models, such as houses and buildings, compared with DIA and FME. In this study, E-OSA can process all the models used, while DIA and FME can only convert the bridge model.
- (2) All shape representations of building elements can be retained during transformation. There is barely no geometric information loss during transformation using the proposed method. E-AMG can transform what IFC models carry. As a result, some transformed

elements may even be over-detailed, such as some doors, windows, railings and stairs. These over-detailed elements should be further processed for better visualization and data transfer efficiency.

- (3) E-AMG is more advanced and robust than the previous AMG. It considers all possible extrusion situations and can be used to generate multipatch using either sweep parameters or two initial rings. Also, E-AMG supports more types of representation. Apart from the swept solid, which is the only representation that AMG is able to process, E-AMG can also handle Brep, clipping and mapped representation.
- (4) Compared with using CityGML, the use of shapefile eliminates the needs of arduous classes mapping and LoD mapping. Also, using shapefile can produce solid models rather than surface models. However, using shapefile can only achieve one-way data transformation from BIM to GIS, while CityGML is prominent for bidirectional information exchange between BIM and GIS.

E-OSA can perform better geometry transformation from IFC to shapefile. However, the following limits cannot be ignored: (1) the transformation from CSG to Brep has not been realized, even though a potential solution has been proposed. (2) The proposed method is by far not a 100% automated process when dealing with multiple-clipping geometries. In the future, these two aspects need to be improved.

The ultimate goal of our research is free information exchange between BIM and GIS. This contains two aspects: BIM-to-GIS transformation and GIS-to-BIM transformation. Each type of transformation is



composed of geometry transformation and semantics transfer. At this stage, the authors are working on geometry transformation from BIM to GIS, while semantics transfer from BIM to GIS and the GIS-to-BIM transformation will be studied in the future.

### Author contributions

Junxiang Zhu and Xiangyu Wang conceived of and designed the experiments; Junxiang Zhu performed the experiments and analyzed the data; Junxiang Zhu and Mengcheng Chen wrote the paper. Junxiang Zhu, Peng Wu and Mi Jeong Kim revised the paper.

### Appendix A. Equation for acquiring the intersections

$$\begin{array}{l}
 \text{Normal direction to plane: } [A \ B \ C] \\
 \text{A point on plane: } (x_0 \ y_0 \ z_0) \\
 \text{Swept area: } [x \ y] \\
 \text{Intersections: } [X \ Y \ Z]
 \end{array}
 \left. \begin{array}{l} \\ \\ \\ \\ \end{array} \right\} \Rightarrow \text{Plane: } AX + BY + CZ = Ax_0 + By_0 + Cz_0$$

$$\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \Rightarrow \text{Lines: } \begin{cases} X = x \\ Y = y \end{cases}$$

$$\text{Intersections: } \begin{cases} X = x \\ Y = y \\ AX + BY + CZ = Ax_0 + By_0 + Cz_0 \end{cases} \Rightarrow \begin{cases} X + 0Y + 0Z = x \\ 0X + Y + 0Z = y \\ AX + BY + CZ = Ax_0 + By_0 + Cz_0 \end{cases}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A & B & C \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} x \\ y \\ Ax_0 + By_0 + Cz_0 \end{bmatrix} \Rightarrow [X \ Y \ Z] = [x \ y \ Ax_0 + By_0 + Cz_0] \times \left( \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ A & B & C \end{bmatrix} \right)^{-1}$$

### References

[1] S. Yamamura, L. Fan, Y. Suzuki, Assessment of urban energy performance through integration of BIM and GIS for smart city planning, *Procedia engineering* 180 (2017) 1462–1472 <https://doi.org/10.1016/j.proeng.2017.04.309>.

[2] H. Tashakkori, A. Rajabifard, M. Kalantari, A new 3D indoor/outdoor spatial model for indoor emergency response facilitation, *Build. Environ.* 89 (2015) 170–182 <https://doi.org/10.1016/j.buildenv.2015.02.036>.

[3] F.G. Brundu, E. Patti, A. Osello, M. Del Giudice, N. Rapetti, A. Krylovskiy, M. Jahn, V. Verda, E. Guelpa, L. Rietto, IoT software infrastructure for energy management and simulation in smart cities, *IEEE Transactions on Industrial Informatics* 13 (2) (2017) 832–840 <https://doi.org/10.1109/TII.2016.2627479>.

[4] 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 <https://doi.org/10.1016/j.autcon.2015.03.019>.

[5] S. Amirebrahimi, A. Rajabifard, P. Mendis, T. Ngo, A BIM-GIS integration method in support of the assessment and 3D visualisation of flood damage to a building, *J. Spat. Sci.* 61 (2) (2016) 317–350 <https://doi.org/10.1080/14498596.2016.1189365>.

[6] G. Boyes, C. Ellul, D. Irwin, Exploring BIM for operational integrated asset management—a preliminary study utilising real-world infrastructure data, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences IV-4/W5* (2017) 49–56 <https://doi.org/10.5194/isprs-annals-IV-4-W5-49-2017>.

[7] J. Zhu, G. Wright, J. Wang, X. Wang, A critical review of the integration of geographic information system and building information modelling at the data level, *ISPRS International Journal of Geo-Information* 7 (2) (2018) 66 <https://doi.org/10.3390/ijgi7020066>.

[8] E.P. Karan, J. Irizarry, J. Haymaker, BIM and GIS integration and interoperability based on semantic web technology, *J. Comput. Civ. Eng.* 30 (3) (2015) 04015043 [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000519](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000519).

[9] C. Mignard, C. Nicolle, Merging BIM and GIS using ontologies application to urban facility management in ACTIVE3D, *Comput. Ind.* 65 (9) (2014) 1276–1290 <https://doi.org/10.1016/j.compind.2014.07.008>.

[10] K.A. Ohori, F. Bijjecki, A. Diakité, T. Krijnen, H. Ledoux, J. Stoter, Towards an integration of GIS and BIM data: what are the geometric and topological issues, *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. IV-4/W5*, 2017, pp. 1–8, <https://doi.org/10.5194/isprs-annals-IV-4-W5-1-2017>.

[11] J. Zhu, P. Wang, X. Wang, An assessment of paths for transforming IFC to shapefile

### Declaration of Competing Interest

The authors declare no conflicts of interest.

### Acknowledgments

This research was supported by the Australian Government through the Australian Research Council's Linkage Projects (#LP140100873 and #LP160100528) and Discovery Project (#DP180104026 and #DP170104613). The authors would like to thank the anonymous reviewers for their comments and suggestions that helped improve the comprehensiveness and clarity of our paper.

for integration of BIM and GIS, 26th International Conference on Geoinformatics, Kunming, China, 2018, pp. 1–5, <https://doi.org/10.1109/GEOINFORMATICS.2018.8557099>.

[12] S. Donkers, H. Ledoux, J. Zhao, J. Stoter, Automatic conversion of IFC datasets to geometrically and semantically correct CityGML LOD3 buildings, *Trans. GIS* 20 (4) (2016) 547–569 <https://doi.org/10.1111/tgis.12162>.

[13] B. Forum, Level of development specification. Available online: [https://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Guide\\_2017-11-06-1.pdf](https://bimforum.org/wp-content/uploads/2017/11/LOD-Spec-2017-Guide_2017-11-06-1.pdf) (Accessed on 14 August 2018).

[14] Y. Deng, J.C. Cheng, C. Anumba, A framework for 3D traffic noise mapping using data from BIM and GIS integration, *Struct. Infrastruct. Eng.* 12 (10) (2016) 1267–1280 <https://doi.org/10.1080/15732479.2015.1110603>.

[15] A. Borrmann, T.H. Kolbe, A. Donaubaauer, H. Steuer, J.R. Jubierre, M. Flurl, Multi-scale geometric-semantic modeling of shield tunnels for GIS and BIM applications, *Comput. Aided Civ. Inf. Eng.* 30 (4) (2015) 263–281 <https://doi.org/10.1111/mice.12090>.

[16] Y. Deng, J.C. Cheng, C. Anumba, Mapping between BIM and 3D GIS in different levels of detail using schema mediation and instance comparison, *Autom. Constr.* 67 (2016) 1–21 <https://doi.org/10.1016/j.autcon.2016.03.006>.

[17] R. de Laat, L. Van Berlo, Integration of BIM and GIS: the development of the CityGML GeoBIM extension, *Advances in 3D Geo-Information Sciences*, Springer, 2011, pp. 211–225 (ISBN: 978-3-642-12670-3).

[18] T.W. Kang, C.H. Hong, IFC-CityGML LOD mapping automation using multi-processing-based screen-buffer scanning including mapping rule, *KSCE J. Civ. Eng.* 22 (2) (2018) 373–383 <https://doi.org/10.1007/s12205-017-0595-9>.

[19] J. Zhu, X. Wang, P. Wang, Z. Wu, M.J. Kim, Integration of BIM and GIS: geometry from IFC to shapefile using open-source technology, *Autom. Constr.* 102 (2019) 105–119 <https://doi.org/10.1016/j.autcon.2019.02.014>.

[20] Wikipedia, Solid modeling, Available online [https://en.wikipedia.org/wiki/Solid\\_modeling](https://en.wikipedia.org/wiki/Solid_modeling), Accessed date: 29 January 2019.

[21] Wikipedia, Constructive solid geometry, Available online [https://en.wikipedia.org/wiki/Constructive\\_solid\\_geometry](https://en.wikipedia.org/wiki/Constructive_solid_geometry), Accessed date: 26 January 2019.

[22] buildingSMART, IfcBooleanClippingResult, Available online <http://www.buildingsmart-tech.org/IFC2x3/TC1/html/ifcgeometricmodelresource/lexical/ifcbooleanclippingresult.htm>, Accessed date: 26 January 2019.

[23] buildingSMART, IfcMappedItem, Available online <http://www.buildingsmart-tech.org/IFC2x3/TC1/html/ifcgeometryresource/lexical/ifcmappeditem.htm>, Accessed date: 26 January 2019.

[24] buildingSMART, IfcRoof, Available online <http://www.buildingsmart-tech.org/IFC2x3/TC1/html/ifcsharedbldgelements/lexical/ifcroof.htm>, Accessed date: 25

- January 2019.
- [25] D. Um, *Solid Modeling, Rapid Prototyping Applications, CAD and CAE Theory*, Springer, New York, 978-3-319-21822-9, 2016.
- [26] I. Stroud, *Boundary Representation Modelling Techniques*, Springer, London, 2006 ISBN: 0-387-84628-312-4.
- [27] C. Eastman, P. Teicholz, R. Sacks, K. Liston, *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, John Wiley & Sons, Hoboken, 978-0-470-54137-1, 2011.
- [28] J.D. Foley, A. Van Dam, *Fundamentals of interactive computer graphics*, Addison-Wesley Reading, MA, 1982 (ISBN: 0-201-14468-9).
- [29] D.J. Eck, Introduction to Computer Graphics, Available online <http://math.hws.edu/graphicsbook/>, Accessed date: 4 May 2019.
- [30] A.A. Requicha, Representations of rigid solid objects, *Computer Aided Design Modelling, Systems Engineering, CAD-Systems*, Springer, 978-3-540-38324-6, 1980, pp. 1–78.
- [31] M.O. Benouamer, D. Michelucci, Bridging the Gap between CSG and Brep Via a Triple Ray Representation, the Fourth ACM Symposium on Solid Modeling and Applications, vol. 14, (1997), pp. 68–79, <https://doi.org/10.1145/267734.267755>.
- [32] B. Wassermann, S. Kollmannsberger, T. Bog, E. Rank, From geometric design to numerical analysis: a direct approach using the finite cell method on constructive solid geometry, *Computers & Mathematics with Applications* 74 (7) (2017) 1703–1726 <https://doi.org/10.1016/j.camwa.2017.01.027>.
- [33] U. Isikdag, G. Aouad, J. Underwood, N. Trodd, Investigating the applicability of IFC in geospatial environment in order to facilitate the fire response management process, Available online <https://core.ac.uk/download/pdf/104426.pdf>, Accessed date: 2 April 2019.
- [34] U. Isikdag, S. Zlatanova, Towards defining a framework for automatic generation of buildings in CityGML using building information models, *3D Geo-Information Sciences*, Springer, 978-3-642-09951-9, 2009, pp. 79–96.
- [35] S.S. Inc., FME - The Simple Solution for Complex Integration, Available online <https://www.safe.com/> Accessed on 12 February 2019.
- [36] P. Tobiáš, An investigation into the possibilities of BIM and GIS cooperation and utilization of GIS in the BIM process, *Geoinformatics FCE CTU 14 (1) (2015) 65–78* <https://doi.org/10.14311/gi.14.1.5>.
- [37] Wikipedia, File:Right-Hand Grip Rule.svg, Available online [https://en.wikipedia.org/wiki/File:Right-hand\\_grip\\_rule.svg](https://en.wikipedia.org/wiki/File:Right-hand_grip_rule.svg), Accessed date: 25 January 2019.
- [38] P. Dawkins, Section 1–3: Equations of Planes, Available online <http://tutorial.math.lamar.edu/Classes/CalcIII/EqnsOfPlanes.aspx> Accessed on 8 February 2019.
- [39] P. Dawkins, Section 1–2: Equation of Lines, Available online <http://tutorial.math.lamar.edu/Classes/CalcIII/EqnsOfLines.aspx> Accessed on 8 February 2019.
- [40] buildingSMART, IfcHalfSpaceSolid, Available online <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometricmodelresource/lexical/ifcHalfSpaceSolid.htm> Accessed on 8 February 2019.
- [41] buildingSMART, IfcMappedItem, Available online <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometryresource/lexical/ifcMappedItem.htm>, Accessed date: 25 January 2019.
- [42] IfcWiki, KIT IFC Examples, Available online [http://www.ifcwiki.org/index.php?title=KIT\\_IFC\\_Examples](http://www.ifcwiki.org/index.php?title=KIT_IFC_Examples), Accessed date: 25 January 2019.
- [43] buildingSMART, Industry Foundation Classes IFC2x Edition 3 Technical Corrigendum 1, Available online <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/>, Accessed date: 25 January 2019.
- [44] D. Kuehne, BIM-GIS Integration with IFC, Available online <https://www.esri.com/arcgis-blog/products/product/3d-gis/bim-gis-integration-with-ifc/>, Accessed date: 4 May 2019.
- [45] ESRI, The Multipatch Geometry Type, Available online <https://support.esri.com/en/white-paper/1483>, Accessed date: 26 January 2019.
- [46] OGC, OGC making location count, Available online <http://www.opengeospatial.org/> Accessed on 2 February 2019.