Automation in Construction xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Automation in Construction

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

Automatic generation of second-level space boundary topology from IFC geometry inputs

Georgios N. Lilis[a,*], Georgios I. Giannakis[a], Dimitrios V. Rovas[b]

[a]Department of Production Engineering and Management, Technical University of Crete, Greece
[b]Institute for Environmental Design and Engineering, University College London, United Kingdom

ARTICLE INFO

Article history:
Received 4 September 2015
Received in revised form 14 August 2016
Accepted 25 August 2016
Available online xxxx

Keywords:
Industry Foundation Class
Building Information Model
Building energy performance simulation
Second-level space boundaries

ABSTRACT

The Industry Foundation Classes (IFC) is a semantically rich data model providing necessary information to support extraction of information necessary for the setup of building energy simulations. Often, 2nd-level space boundary data contained in IFC, are missing or incorrect. To facilitate the connection between BIMs and energy simulation programs, the Common Boundary Intersection Projection (CBIP) algorithm is introduced. CBIP uses the geometric representations of building entities obtained from IFC files to generate the building's 2nd-level space boundary topology. A prototypical implementation of the CBIP algorithm is used in a complex geometry building, as a verification of the capability of the algorithm to properly identify space boundaries.

1. Introduction

The recent requirementfor efficient allocation of energy resources in the building sector, has resulted in the increased use of building thermal simulations, during both the building design [1,2] and operation phases [3,4]. The accuracy of a thermal simulation model strongly depends on the accurate definition of building geometric characteristics, which include: the building envelope, the building orientation, the configuration of spaces, surfaces and volumes.

In current state of practice, it is quite common that a Computer-Aided Design (CAD) tool is used to represent the geometry. However, such an architectural perspective must be altered, in order for energy simulations to be performed [5]. Hence, building geometrical data extracted from CAD programs have to be manually transformed and combined with material properties to be entered as inputs to energy simulation routines, a process which is both time consuming and error-prone. CAD data have simple semantics, Building Information Models (BIM) [6] provide an improved way for information storage with richer semantics that include: building geometry, material data and information on building services. Open BIM data schemas include the Industry Foundation Classes (IFC) [7] and the green-building XML schema (gbXML) [8]. The popularity of these two open BIM schemas led many leading AEC software companies to implement support for gbXML- and IFC-based exchanges within their BIM authoring suites. Examples of such tools are Revit (AutoDesk) [9] and ArchiCad (GraphiSoft) [10].

A wide variety of promising attempts have been proposed to establish an automated data exchange between BIM and thermal simulation tools. The IDF Generator [11], developed at the Lawrence Berkeley National Laboratory (LBNL), works in conjunction with the Geometry Simplification Tool (GST) and transforms IFC-format building geometry into EnergyPlus input-data file (IDF) [12]; GST simplifies the original building geometry defined in IFC-format and converts it into gbXML-format, while the IDF Generator converts the gbXML-format file into EnergyPlus input-data file. The resulting IDF file contains all information related to building geometry and constructions needed to run an EnergyPlus simulation. IDF Generator is proposed as a semi-automated process, since for complex building geometries a manual manipulation of IDF geometry is required, including some corrections to windows in curtain walls, missing floors and ceilings [13]. The RIUSKA [14], developed by Granlund, uses the DOE-2.1 [15] thermal simulation engine and imports the building geometry from an IFC file, utilizing the BSPro server middleware [16]. Limitations of its IFC import exist, since RIUSKA ignores slabs in the IFC file and simply generates them internally, according to the size of the space defined by the bounding walls. Moreover, high quality of import results are achieved only when RIUSKA is used in conjunction with SMOG, while compatibility problems occur when other CAD tools are used to author and export the IFC file.

In the AEC software industry, Green Building Studio (GBS) [17] web service uploads the geometry in gbXML-format and converts it into a DOE-2.2 or an EnergyPlus-format file. There are studies

* Corresponding author at:Technical University of Crete, Dept. Production Engineering and Management, Kounoupidiana 73100, Greece.

http://dx.doi.org/10.1016/j.autcon.2016.08.044
0926-5805/

proving that several problems occur during the conversion process [18], including incorrect shading surface definitions and omission of some walls. Trimble's SketchUp together with its Openstudio and IFC2SKP plugins, is able to upload any gbXML or IFC well-formatted geometry and convert it into the EnergyPlus or TRNSYS17-format file [19]. However, lack of maturity of the import tool, neglects some information related to floors and ceilings. Virtual Environment (VE), developed by Integrated Environmental Solutions (IES) [20], is an integrated system that uses a proprietary engine, called Apache. IES VE supports import of gbXML and IFC file formats. Nevertheless, import results rely on the correctness of 2nd-level space boundary geometry contained in IFC, which currently is not exported properly by any BIM authoring tool.

Among the two most popular BIM schemes, gbXML and IFC, IFC appears to be a suitable choice as its more rich in content, enables interoperability among different software environments and can be updated according to the building's modifications [21].

Concerning the building geometry, IFC can provide static building information that include geometric configuration and material properties, but in a form that might not be directly usable for the generation of thermal simulation models due to the absence of 2nd-level space boundary information [13]. Hence, a consistent approach is required to extract building geometry information, contained in an IFC file, and subsequently to correctly identify the 2nd-level space boundary information.

In view of this, several algorithms have been proposed [22–24], which are based on graph theory and convert a three-dimensional architectural building model into the second-level space boundary topology without the need for definition of conditioned building space volumes.

In this work, following a different approach to address the 2nd-level space boundary generation requirement, the Common Boundary Intersection Projection (CBIP) algorithm is presented. A recent study has shown that thermal models obtained based on CBIP algorithm results, are comparable to models of other popular programs [25,26].

CBIP algorithm can be applied to building geometries which do not contain design errors or building space incorrect definitions. In [27], errors that affect the creation of properly defined 2nd-level space boundaries are presented. Commercial software, such as Solibri Model Checker, [28] are able to identify such errors, which are communicated back to the AEC software and corrected manually. With an IFC free of design errors and building space incorrect definitions at hand, its geometric data can be used as input to CBIP algorithm.

Algorithmically, CBIP is divided into four operational stages: the Identification (ID) stage, the Boundary Surface Extraction (BSE) stage, the Common Boundary Intersection (CBI) stage and the Boundary Intersection Projection (BIP) stage which are analyzed in Section 4. CBIP's stages involve geometric operations based on well-known methods for representing shapes, therefore an initial description of such methods, adopted by the algorithm, are presented in Sections 2 and 3. The output of the algorithm used to update the IFC database and the respective Space Boundary class, is described in Section 5, while design requirements and design recommendations to ensure the correct execution of the algorithm are discussed in Sections 6 and 7, respectively. Finally, CBIP has been tested on a demonstration building of a high geometry complexity and its results are presented in Section 9.

2. CBIP algorithm – geometric definitions

CBIP takes as input the geometric representations of various building entities, which are assumed to be polyhedrons, performs certain operations on them and outputs polygonal surfaces which are the 2nd-level space boundaries. Consequently, CBIP algorithm's mathematical foundation consists of geometric operations, applied on geometric representations of the involved building entities.

Various geometric representation methods including the octree and the Boundary representation have been used in Building Information Models [29]. In CBIP two such methods are used. The first, is the Boundary representation (B-rep) [30], described in Section 2.1. B-rep theory is adopted in order to describe each polyhedron by its corresponding boundary polygons. Additionally, to determine the space boundaries which are essentially common surfaces shared by two polyhedrons, the Binary Space Partitioning tree (BSP-tree) polyhedral representation [31] is adopted and described in Section 2.2.

2.1. Boundary representation

The B-rep of a polyhedron **A** associated with a building entity, is denoted by $\partial\mathbf{A}$. Essentially, $\partial\mathbf{A}$ is a set of boundary polygon surfaces $\partial\mathbf{A} = \{A_1, \ldots, A_i, \ldots, A_N\}$ (see Fig. 1). Each boundary polygon surface in this representation, conforms to the right hand outward normal convention: the direction of the normal vector $\hat{n}_{A_i}$ of every boundary polygon A_i evaluated using the right hand, is towards the exterior of the polyhedron **A**, as displayed in Fig. 1. The right hand normal vector direction evaluation method proceeds as follows: when the fingers of the right hand, excluding the thumb, follow the points of the polygon A_i, the thumb points to the direction of the normal vector. Consequently, the outward normal convention, with the direction of the normal vectors evaluated using right hand, requires the boundary polygon points to be correctly ordered: in a counter clock-wise manner when looking from outside the polyhedron (as displayed in Fig. 1).

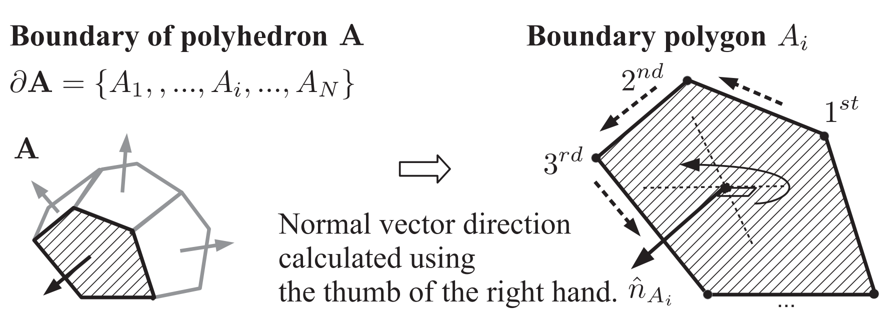

Fig. 1. Boundary representation (B-rep) of a polyhedron as a set of boundary polygon surfaces $\partial\mathbf{A} = \{A_1, \ldots, A_i, \ldots, A_N\}$. When the points of each polygon surface A_i are correctly ordered (counter clockwise when looking from outside the polyhedron) the direction of the normal vector of the surface $\hat{n}_{A_i}$, evaluated using the right hand, is towards the exterior of the polyhedron.

Please cite this article as: G.N. Lilis et al., Automatic generation of second-level space boundary topology from IFC geometry inputs, Automation in Construction (2016), http://dx.doi.org/10.1016/j.autcon.2016.08.044