

IFC-based calculation of the Flemish Energy Performance Standard.

R. Verstraeten, P. Pauwels, R. De Meyer

Ghent University, Department of Architecture and Urban Planning

W. Meeus, J. Van Campenhout

Ghent University, Department of Electronics and Information Systems

G. Lateur

Bureau Bouwtechniek NV, Antwerp

ABSTRACT: This paper illustrates our findings concerning space based design methodologies and interoperability issues for today's Building Information Modeling (BIM) environments. A method is elaborated which enables building designers to perform an automated energy use analysis, based on an Industry Foundation Classes (IFC) model derived from a commercial BIM environment, in this case Autodesk Revit 9.1. A prototype application was built, which evaluates the building model as well as vendor-neutral exchange mechanisms, in accordance with the Flemish Energy Performance Regulation (EPR) standard. Several issues regarding the need for space-based building models are identified and algorithms are developed to overcome possible shortcomings.

1 INTRODUCTION

Today's building design practice finds itself subject to increasing expectations in several knowledge domains, striving towards building designs with better performance in a variety of fields. These challenges ask for a detailed evaluation of the building model in an early stage of the design process. At present, however, a building design is evaluated by experts, using specialized simulation tools, *at the end* of the design process, i.e. when important design decisions have already been made, without any quantified feedback (Suter and Mahdavi, 2004). Therefore, the design process would be greatly enhanced by the availability of uncomplicated evaluation methods, incorporated in, or at least closely related to, the designer's modeling environment. This requires a rich modeling environment, capable of communicating design data with external evaluation tools. Building Information Modeling (BIM) applications, in conjunction with efficient vendor-neutral interoperability methods, such as the Industry Foundation Classes (IFC) initiative (<http://www.iai-tech.org/> May2008), are promising technologies to fulfill these requirements.

This paper focuses on the feasibility of this approach by devising a method to automatically evaluate a building design, modeled in Autodesk Revit

9.1, in accordance with the Flemish Energy Performance Regulation (EPR) standard (<http://www2.vlaanderen.be/economie/energiesparen/epb/doc/bijlage1epb.pdf> May2008). The most important challenge arising from the EPR standard is the need for a space-based model, which makes it an interesting test case, since most of today's modeling practice is not oriented that way.

Fazio et. al. (2007) investigated an IFC based framework for delivering building envelope data to a set of simulation engines, showing that it is feasible to automatically derive the required geometrical and material layer information from a BIM application. However, the procedure focuses on the building envelope and less on three-dimensional, space-related issues. In contrast, the research of Lam et. al. (2006) does take the spaces defined by their enclosing constructions into account, and uses them as a start point for the thermal simulation. However, they make rather far going simplifying assumptions concerning the bounding construction geometry. It is precisely this aspect which is addressed in the current contribution. We target the faithful and complete generation of both the internal and external geometry of space-based models as well as their material properties, so to enable accurate calculation of the building energy performance. To realize such a system, various issues have to be faced, notably (i) the ability of contemporary BIM environments to capture and, most importantly, to export the required

information, and (ii) the ability to deploy vendor-neutral formats to convey the information to the processing software. These issues are addressed, a prototype of such a system was actually built and is briefly presented.

The paper is structured as follows. The next section briefly introduces the Flemish standard against which the software has to operate. Then, in section 3, the well-known IFC format is evaluated with respect to the ease with which the information can be represented. In section 4, some of the solutions are presented that have made the construction of a prototype system possible. Conclusions are formulated in section 5.

2 ARCHITECTURAL DATA REQUIREMENTS FOR THE FLEMISH ENERGY PERFORMANCE REGULATION

As imposed by the *European Directive 2002/91/EC of 16 December 2002 on the energy performance of buildings* (Directive 2002/91/EC, 2002), the Flemish Authority has elaborated a standard with which newly erected and renovated buildings have to comply: the Energy Performance Regulation (EPR). The standard describes a steady state calculation of a building's 'characteristic annual primary energy consumption', implying that some simplifying as-

sumptions have been made, for example the use of fixed exterior and interior temperatures, default occupancy and internal heat gains. By doing so the EPR method does not aspire to assess the *real* energy consumption but delivers a workable basis for a comparison between buildings. In the following, aspects of the standard which exclusively focus on residential buildings are elaborated, the so-called EPW method.

The EPW method consists of calculating the following aspects of a building's energy demand: spatial heating and cooling, domestic hot water supply and the use of auxiliary equipment. Possible gains by solar panels or cogeneration devices are subtracted from the previously calculated energy demands. All contributions are calculated per month and, if applicable, per *energy zone*, a spatial subdivision which groups the spaces sharing a common heating and ventilation system. The previous terms are multiplied by a fuel factor and ultimately combined to provide the *characteristic annual primary energy consumption*, in short the building energy consumption.

In Table 1 the EPW method is summarized from an architectural point of view, describing those contributions which are explicitly related to architectural aspects of a building design.

Table 1 Architectural data requirements for the EPW method

Energy zone attributes	Transmission loss $Q_{T,zone,m}$ MJ	Ventilation loss $Q_{V,zone,m}$ MJ	Internal gain $Q_{I,zone,m}$ MJ	Solar gain $Q_{S,zone,m}$ MJ
External volume $V_{zone} (m^3)$		x	x	
Bounding construction external area $A_{T,zone} (m^2)$	x	x		
Bounding construction material layer thickness $d (m)$	x			
Bounding construction material layer thermal conductivity $\lambda (W/mK)$	x			
Opening element area $A_{w,i} (m^2)$	x	x		x
Opening element Slope $\theta_w (^\circ)$				x
Opening element orientation $\Phi_w (^\circ)$				x
Opening element glazing fraction $F_{w,i} (-)$	x			x
Opening element glazing solar factor $g_i (-)$	x			x

3 MAPPING THE IFC-SCHEME

The Industry Foundation Classes (IFC) standard is a generic hierarchical object model to abstract building components and processes, it was defined to provide a universal basis for process improvement and information sharing in the construction and facilities management industries (<http://www.iai-international.org/About/Mission.html> May2008).

It is clear that the energy zone object plays a prominent role, since it groups all data required. Therefore, the IFC2x3 model scheme is evaluated in view of the following operations: (1) generating the individual energy zones, (2) collecting bounding constructions, (3) extracting corresponding geometry and material layer properties and, if applicable, (4) processing opening elements, related to their energy zone by the bounding constructions in which they occur. Some of the required data is explicitly embedded in the IFC2x3 model scheme, e.g., material layer thickness d , while other data, such as the opening element glazing fraction $F_{w,j}$ or solar factor g_j , are possibly provided through the *IfcPropertySet* objects. An essential part of the required data however, can only be extracted after (sometimes extensive) processing.

3.1 Internal boundary of energy zones

Combining the functional requirements for an energy zone with the IFC2x3 model scheme delivers the *IfcSpace* entity as best match for the energy zone functionality. According to the definition from IAI an *IfcSpace* "... represents an area or volume bounded actually or theoretically. Spaces are areas or volumes that provide for certain functions within a building.", note that this concerns internal dimensions. Additionally, the different bounding constructions for a given *IfcSpace* are provided by the set of *IfcRelSpaceBoundary* entities, which each deliver a bounding construction by an *IfcBuildingElement* instance. Apart from the *IfcBuildingElement*, the *IfcRelSpaceBoundary* entity provides two very useful objects: *IfcConnectionGeometry* and *IfcInternalOrExternalEnum*. The former establishes the geometrical relationship between an *IfcSpace* and the related bounding construction, the latter states whether that construction neighbors the exterior or interior. It should be mentioned here that, although the *IfcBuildingElement* and *IfcConnectionGeometry* attributes are optional for the *IfcRelSpaceBoundary* entity in the IFC2x3 model scheme, they are considered essential in this project, since by these attributes the connection between a bounding construction and a given *IfcSpace* is geometrically described. The connection geometry would be much harder, if not virtually impossible, to determine

without the *IfcRelSpaceBoundary* entity, taking the separate geometrical representations of *IfcSpaces* and *IfcBuildingElements* entities as a starting point.

Several possibilities are provided by IFC2x3 model scheme to define the connection geometry and different definitions are encountered when observing IFC models generated in practice. The connection geometry for slabs, for example, is often defined as an *IfcCurveBoundedPlane* instance, which provides a closed and planar curve, as a boundary facet. However, in compliance with the model scheme, the wall connection geometry might be defined by means of an *IfcSurfaceOfLinearExtrusion* instance. In that case, the generation of the curve defining a wall connection geometry is far more complicated. Merely a prismatic surface is provided, determined by a two dimensional profile, an extrusion direction and height. Surely, the profile to be extruded matches the line segment where the wall connects to the flooring, but problems arise when trying to define the line segment connecting the wall to the space ceiling. Note that the extrusion height is derived from the *IfcSpace* individual representation, which is in fact the space bounding box and therefore does not necessarily coincide with the exact shape. To overcome this problem, an algorithm has been constructed by which the wall connection geometry is developed more precisely. The development consists of calculating the vertices v_i which constitute a curve describing the wall-space connection. Looping through the vertices of the *IfcCurveBoundedPlane* C1 and C2, as provided by both slabs, and

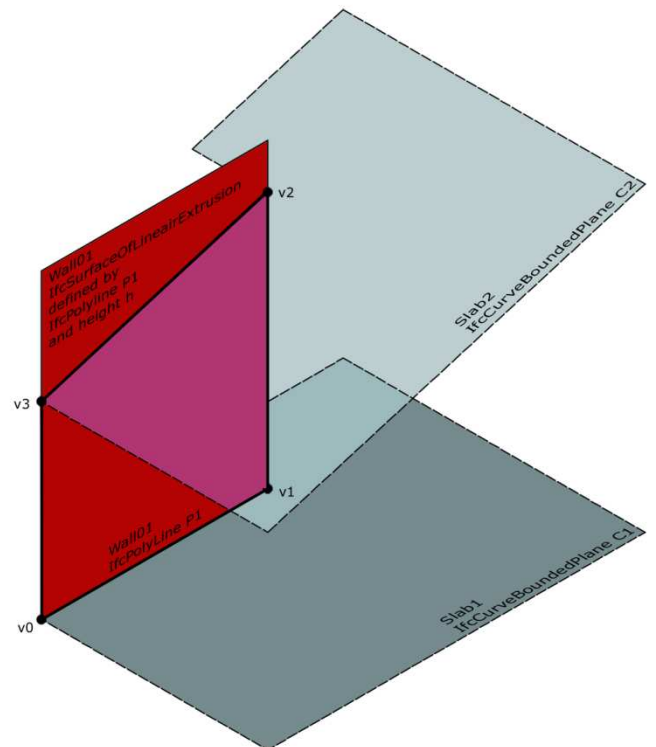


Figure 1. Connection geometry for wall01

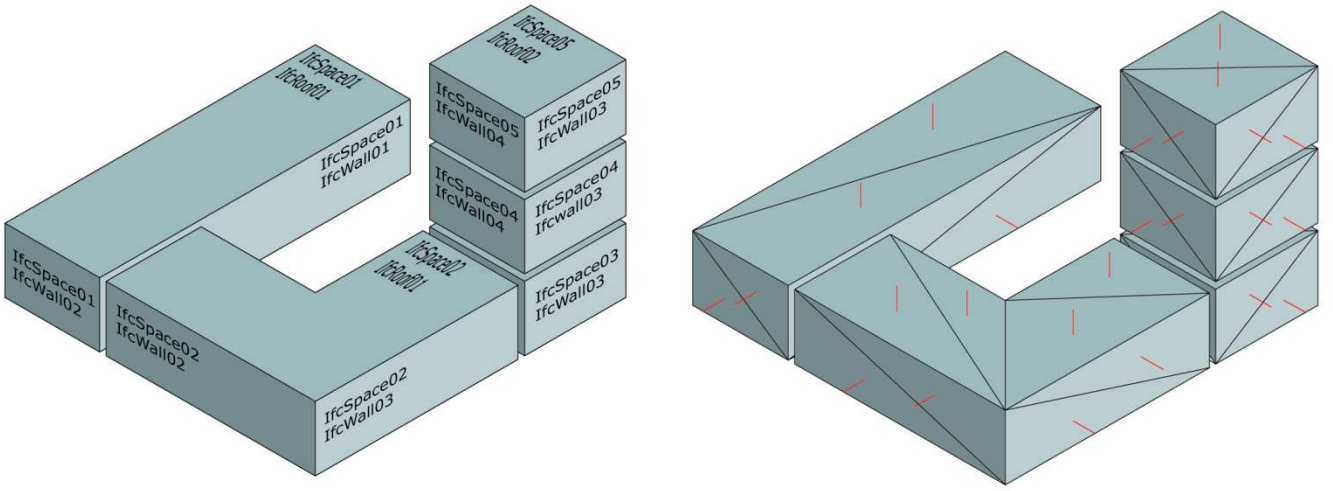


Figure 2. Left pane: Internal geometry for IfcSpaces, right pane: triangulation of curve set, addition of surface normal.

storing those which fit the *IfcSurfaceOfLinearExtrusion* for a specific wall, provides for a set of vertices. Each vertex is a member of both constructions, and thereby describing a bounding curve in the *IfcSurfaceOfLinearExtrusion*, which is determined by the two dimensional profile P_l and height h . Finally, this set is post processed to ensure a correct order, resulting in a closed and planar curve, representing the wall connection geometry for a given space (Fig. 1).

With the *IfcConnectionGeometry* entity for each bounding element provided, the *IfcSpace*'s internal geometry can be processed, resulting in a set of planar curves each related to an *IfcBuildingElement* instance or a specific part of it (Fig. 2). The internal geometry describes the *IfcSpace* as seen from the inside and its representation is assumed to define a closed shell.

3.2 External boundaries of energy zones.

The internal geometry, which is in fact a closed set of curves, does not yet provide the data needed, i.e., the *IfcSpace* external volume V_{zone} or its bounding constructions external area $A_{T,zone}$. As illustrated in Fig.1, gaps appear between neighboring spaces and the building's outside geometry is completely absent, due to the lack of geometry for all bounding constructions like walls, floors and roofs. We now describe an algorithm which inflates the *IfcSpace* internal geometry, to match its outer boundaries, by generating the boundary elements' volume. In a pre-processing stage, each curve is triangulated and its surface normal justified (Fig. 2).

All construction components, e.g., walls, slabs, roofs, stairs, columns, beams ... are subtypes of the *IfcBuildingElement* entity, which provides the geometrical representations and material usage. The *IfcMaterialLayer* instance has an attribute *Layer Thickness* d_i and *IfcMaterial*. Combining the values

for d_i results in the *IfcBuildingElement*'s total thickness d_T . The *IfcMaterial* instance provides the material name and, possibly, a link to an external reference comprising physical material properties or specifications.

A procedure has been developed to generate the *IfcBuildingElement* geometry, derived from the curve sets described in section 3.1 and its total thickness d_T , which would then be space-related in contrast to its individual representation as described by the *IfcProductDefinitionShape* attribute.

Generation of the *IfcBuildingElement* space-related geometry, needed for the calculation of its volume calculation, consists of defining the outer curve, parallel to the inner space bounding curve, and all curves connecting the inner and outer curves. This is a vertex-based operation. For each vertex v_0 of an *IfcBuildingElement*'s inner curve C , at least one new vertex v_0' is generated. The number of inner curves C_i containing v_0 , the surface normals n_i and corresponding offset distances $d_{T,i}$ are the parameters required for the calculation. In most cases vertex v_0' is derived as the intersection point between three planes, defined by the surface normal and offset distance corresponding with its curve set C_i , however, different algorithms are used depending on the constellation of the inner curves of which v_0 is a common vertex (Fig. 3). Performing the procedure for a *IfcBuildingElement*'s inner curve delivers a closed shell, enabling the calculation of its volume.

3.3 Introducing the opening elements

In Table 1 several requirements in relation to opening elements are listed: area $A_{w,j}$, slope θ_w , orientation Φ_w , glazing fraction $F_{w,j}$ and solar factor g_j . In contrast with the first three requirements, which relate to the geometrical aspects of a void made in a construction element, the latter two relate to the objects which fill the void.

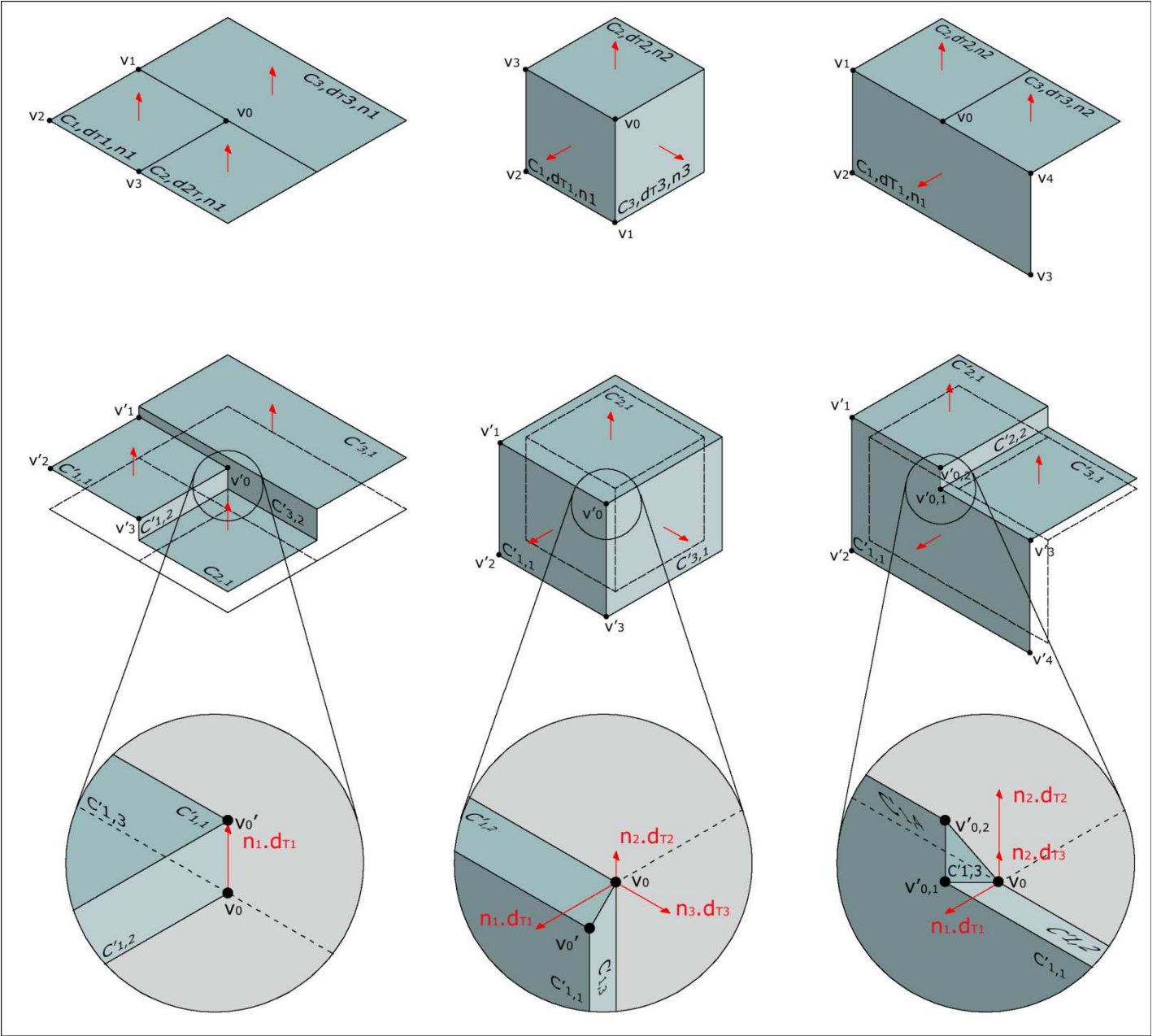


Figure 3. Offset cases: one, three and two surface normals for a given vertex v_0 , as a member of C_1 .

In general the geometrical representation for the voiding *IfcOpeningElement* consists of an extruded solid object, defined by a two dimensional profile and extruded orthogonally to the wall's face. Sometimes a boundary-representation object is used as the opening's geometrical representation. In that case the boundary-representation bounding box is calculated and analogously treated as the previous solid object.

The key to establishing the relationship between the *IfcOpeningElement* geometry and the *IfcSpace* inner geometry is the two dimensional profile, used by the extruded solid object, by which the opening is described. By projecting this profile, parallel to the extrusion axis onto the host construction's inner curve, the opening's geometry is embedded within the bounding construction's new representation. A new curve is generated, representing the part of the

construction where the opening is located and the opening's area $A_{w,j}$, slope θ_w and orientation Φ_w are calculated.

The solar factor g_j and the glazing fraction $F_{w,j}$ are retrievable via the *IfcWindow* or *IfcDoor* instance, accessed through the *IfcRelFillsElement* instances.

4 PROJECT CASE, FEEDING THE EPW APPLICATION FROM REVIT 9.1

In this project, Autodesk® Revit 9.1 was used to model an office building design and perform its export to the IFC2x3 data format. The building consists of six spaces, arranged around a patio in a split-level layout. Spaces are defined by means of the Revit room object, construction components are pro-

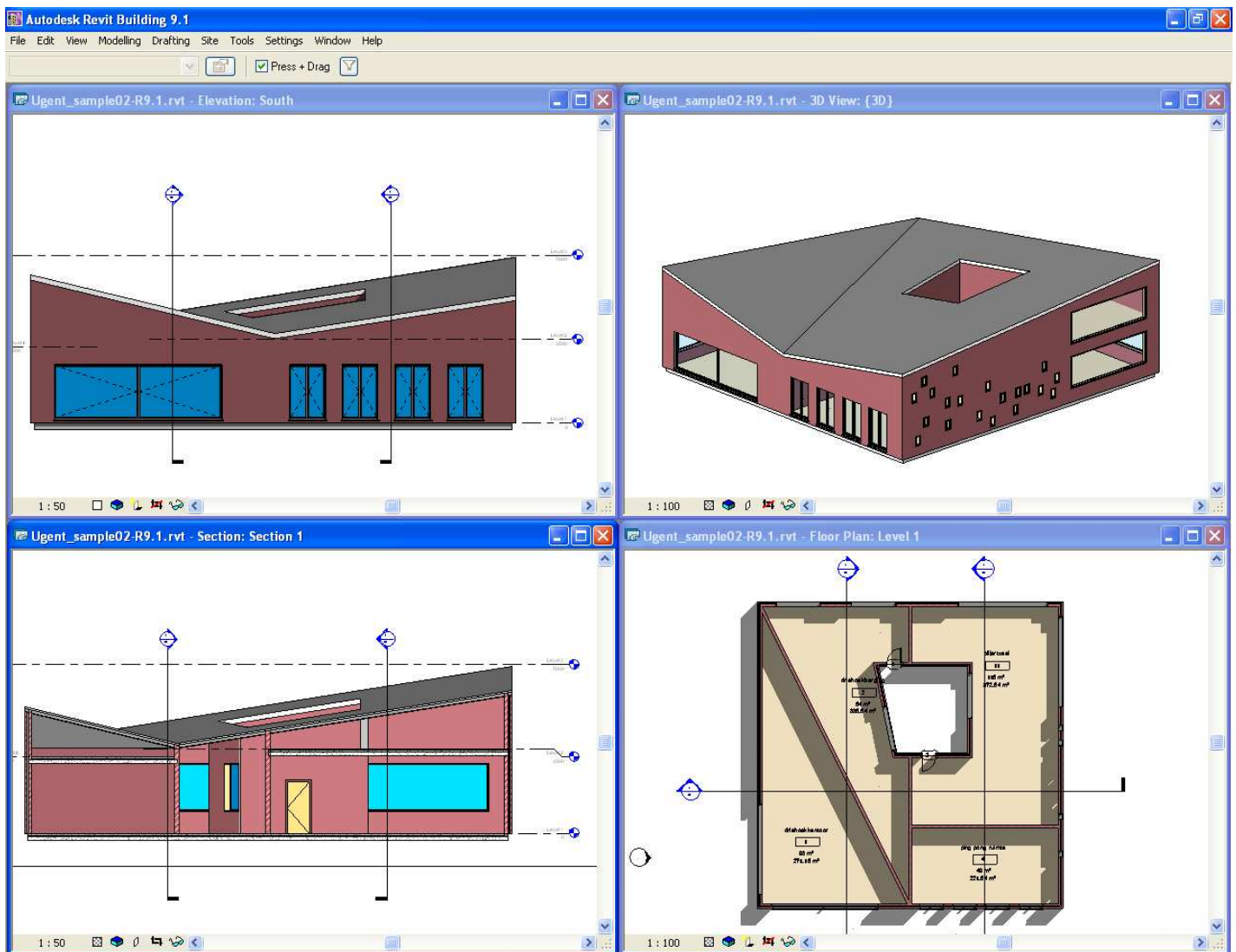


Figure 4. Office building design in Autodesk Revit 9.1

vided with corresponding material layers and opening elements are modeled as simple, prismatic voids or more complex shapes, e.g. recessed windows (Fig. 4).

Reading the IFC file.

A preparatory procedure is developed which translates the IFC2x3 model scheme, described in the Express language, into a hierarchical class library for Java or C#. This procedure includes four steps. Initially, the current version of the Express scheme is read and walked, by means of an ANTLR parser (<http://www.antlr.org/> May2008), and an abstract syntax tree structure is generated. Then, by traversing the tree structure, information concerning data types, enumerations, selects and entities is stored in a set of IFC classes. Subsequently, an code generator was developed, which translates the newly generated IFC classes into a Java or C# class library. Finally, a dispatch table is generated, defining a constructor call for each class. By developing this procedure, a generic tool is created by which each new version of the IFC model scheme can be translated into a Java or C# class library.

Reading a specific IFC file comes down to mapping each IFC entity on the related C# class and, by using the dispatch table, calling the corresponding constructor which recursively generates the related objects.

4.1 Processing the data.

All IFC objects in memory are analyzed and the useful data is exported to an XML file, describing the construction component geometry and used material layers in a space based structure. The EPW calculation procedure starts from this XML input file. This file format was introduced to enable not only the import of model files based on IFC data format, but also using other formats as a data provider, e.g. the GbXML data format, supported by the Demeter plug in for Google SketchUp (http://www.greenspaceresearch.com/demeter_tutorial2.html May2008). Next, the interior and exterior space boundaries are computed as described in the previous sections.

Although in the IFC model the possibility exist to describe a material by its physical properties, or add

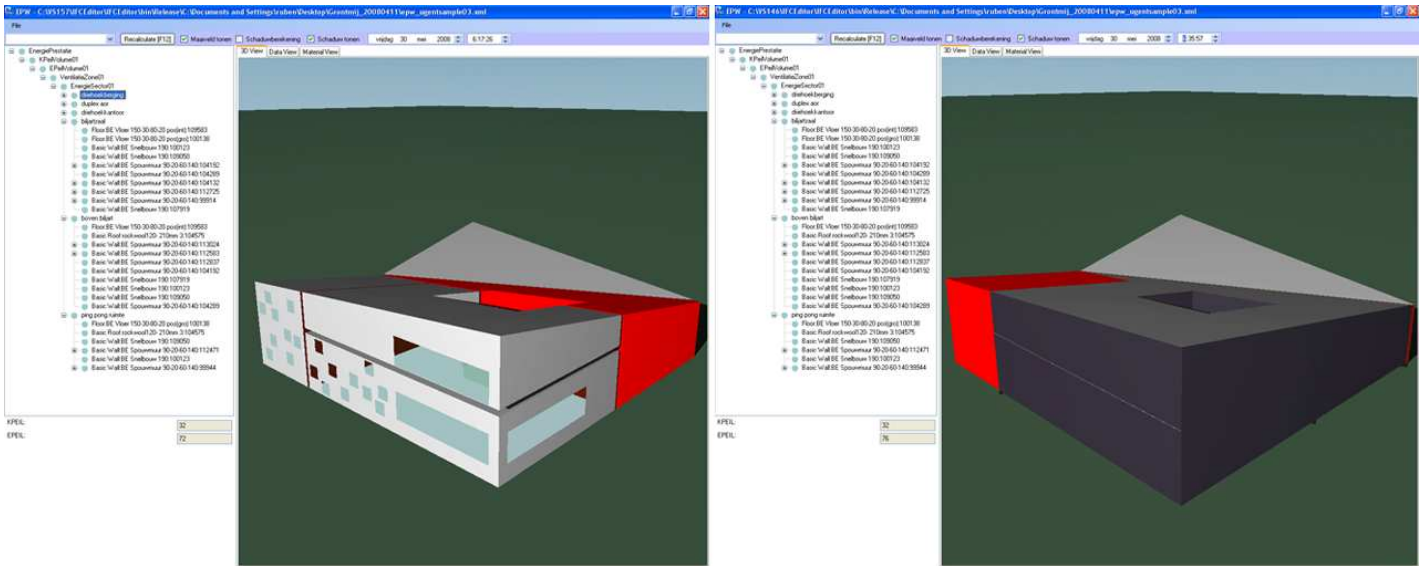


Figure 5. Prototype EPW application, 3D view on internal and external geometry

an external reference, in practice, nothing more than visualization properties can be found in IFC representations. An internal material database is introduced to overcome this problem, supplying, among others, the thermal properties. Once a material or material layer exists in the database it will be automatically attributed to its corresponding construction component, meaning that the assignment is a non-recurrent user intervention.

4.2 User Interface.

The user interface consists of five views. The first is a view which enables the user to brows the project by means of a tree structure, displaying all components in a space-based structure. A construction component missing a specified material layer or material layer set will therefore be indicated with a dedicated icon. Objects higher in the hierarchy, depending on the incomplete component, are indicated accordingly. The material layer assignment is crucial for the calculation to start, therefore a view by which the material library is accessed is introduced. Additionally, three-dimensional views are provided, enabling the user to check for possible misinterpretations on behalf of the model geometry. Since both internal as external geometry are essential, both are rendered in separate views (Fig. 5).

Finally, the results are displayed in the data view, enabling the designer to evaluate the model and look for possible design ameliorations.

5 CONCLUSION.

In this exploratory research project the feasibility to exchange IFC model data between Autodesk Revit9.1 and the Flemish EPW method is investigated, resulting in a software prototype which successfully

illustrates the objectives. Several issues regarding the need for a space-based building model are identified and algorithms are developed to overcome possible shortcomings. This research can be further expanded to address space based design methods and representations. The supply of a space based building model by the source application could be strongly ameliorated by the exact three dimensional representation of spaces and the mandatory use of more explicit IFC entities, such as curve-like descriptions for the connection geometry instances.

6 ACKNOWLEDGEMENTS

The project team acknowledges the funding support from the Ghent University IOF council and is grateful to Ir Wim Meeus for his helpful contribution.

7 REFERENCES.

- DIRECTIVE 2002/91/EC. 2002. Directive 2002/91/EC of the European parliament and of the council of 16 December 2002 on the energy performance of buildings.
- Fazio, P., He, H.S., Hammad, A., Horvat, M. 2007. IFC-based framework for evaluating total performance of building envelopes. In *Journal of Architectural Engineering*, 1, 44-53
- Lam, K. P., Wong, N. H., Shen, L. J., Mahdavi, A., Leong, E., Solihin, W., Au, K. S. & Kang, Z. 2006. Mapping of industry building product model for detailed thermal simulation and analysis. In *Advances in Engineering Software*, 37, 133-145.
- Suter, G. & Mahdavi, A. 2004. Elements of a representation framework for performance-based design. In *Building and Environment*, 39, 969-988.