# AN IFC DATA PREPARATION WORKFLOW FOR BUILDING ENERGY PERFORMANCE SIMULATION

Kyriakos Katsigarakis[1], Georgios Nektarios Lilis[1], Georgios Giannakis[1] and Dimitrios Rovas[2]
[1]Technical University of Crete, Greece
[2]University College London, London, UK

## Abstract

Accurate building energy performance simulation model generation from IFC data files require the input IFC data to be correct and complete. To ensure such data quality conditions, methods integrating IFC data correctness and completeness checks, under a common architecture, are introduced. IFC data correctness is supported by the use of a dedicated geometric error detection tool, which identifies and reports back to the building designer, errors affecting the building's 2nd-level space boundary topology. IFC data completeness is performed by the use of a dedicated for building energy performance simulation Model View Definition schema. The quality checking results are reported in a BIM collaboration file format.

## Introduction

The accuracy of Building Energy Performance Simulation (BEPS) results is determined by the quality of its input data, mainly comprising the building geometry, its construction's materials, internal loads, HVAC systems and components, weather data, operating strategies and schedules, and simulation specific parameters (Maile et al. 2007).

In current practice, to develop a BEPS model, modelers gather and combine 2D drawings such as Architectural and Mechanical Electrical Plumping (MEP) plan views, material data and other information, and manually transform them into the specific input data, required by the respective BEPS engine. This process has two strong weaknesses: a) it is very time-consuming, often requiring more time than is available due to project's deadlines; and b) it is a non-standardized process that produces BEPS models whose results can significantly vary from one modeler to another according to their experience (Berkeley et al. 2014), even given the same initial building design information. A methodology for automated creation of BEPS models could make the BEPS modeling process much more expedient and therefore, less vulnerable to modeling errors. Building Information Models (BIM) are information-rich repositories that could be used to streamline and expedite the collection of such information. Hence, recent research studies focus on BIM-based BEPS models generation (Andriamamonjy et al. 2018, Thorade et al. 2015, Wimmer et al. 2015, Giannakis et al. 2015).

BIM is an object-oriented digital representation of a building, which also can capture part of the information required for generating a BEPS input data file. Relevant open-BIM data schemas include the Industry Foundation Classes (IFC) (ISO 16739-1 2018) and the green-building XML schema (gbXML). Concerning building geometry, IFC supports three dimensional representations, while gbXML supports only planar geometry. For this reason, a plethora of recent studies focuses on developing a methodology to automatically generate geometry inputs for BEPS from IFC files. Additionally, IFC appears to be a more suitable choice as its rich content enables interoperability among different software environments and can easily be updated following a building's life cycle (Hitchcock & Wong 2011).

However, applying an automated data transformation from IFC data to input data of BEPS tools is not a straightforward but a tedious task often due to imperfections of the provided IFC files. Although commercial BIM authoring tools (Revit™, Archicad™, Vasari™) support exportation of IFC files, their quality is not acceptable in the context of BEPS input data generation. Even Revit™, and its IFC4 Design Transfer View (DTV) exporter, which currently seems to be the most mature tool, is far from being perfect: while IFC schema can capture data about material thermal properties, internal gains (schedules and densities) and 2nd-level space boundaries (Bazjanac 2010), requested from the BEPS perspective, relevant information is missing or incorrect in the exported IFC4 DTV files.

## Proposed IFC4 workflow

To examine and eliminate these IFC data imperfections, in this work an IFC4 data workflow is introduced. This workflow aims at delivering error-free IFC4 files ready for BEPS input data generation. The workflow starts with the BIM preparation and the IFC data exportation processes (first block of Figure 1). Since the lack of training of the BIM designer contributes to data inaccuracies, a set of de-

 DOI:10.35490/EC3.2019.188

sign guidelines have been developed to help the designer to define properly all the requested information during the BIM preparation process. After the BIM data are prepared, the IFC data exportation process follows. However, due to issues mentioned above, the exported IFC file is of poor quality (as for the 2$^{nd}$-level space boundary information) or lacks valuable information (as for the material thermal properties and internal gains); for the latter, Revit$^{TM}$'s IFC4 DTV exporter has been modified to correctly populate such information.

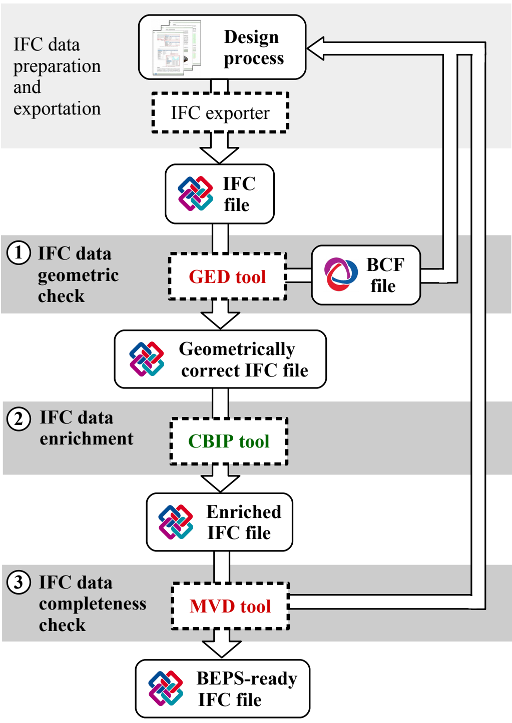


*Figure 1: Overview of the proposed IFC data workflow*

After the BIM preparation and IFC exportation processes, the IFC4 data workflow continues in three main stages. During the first stage (IFC data geometric check) the architectural geometric content of the IFC file is examined using the GED tool (Lilis et al. 2015), for possible errors which affect the BEPS model generation process and if found they are reported in BIM Collaboration Format (BCF). In the second stage (IFC data enrichment) the IFC file is enriched using the Common Boundary Intersection Projection (CBIP) tool with the necessary for BEPS 2$^{nd}$-level space boundary information. Similar enrichment tools, such as the SBT tool (Rose & Bazjanac 2015), have been developed using graph-based methodologies. In the final stage (IFC data completeness check) several checks are performed on the IFC data to ensure acceptable quality for BEPS input data file generation. The checks are performed using the Model View Defintion (MVD) Checking tool. For its development, other supporting tools, relevant libraries and an MVD XML for conventional BEPS have been created. Other attempts in this domain have been primarily focused on developing a MVD XML for conventional and advanced BEPS (Pinheiro et al. 2018). The initial processes and the three main stages of the workflow are described analytically in the following sections.

## IFC data preparation and exportation

To help the BIM designer on creating an error-free (or error-reduced) BIM, from which the maximum level of BEPS information can be exported, a document that presents a set of design guidelines as they apply for Revit$^{TM}$ has been developed – similar guidelines can be developed for other BIM authoring tools.

The guidelines are provided as a set of modelling rules belonging to two broad categories: Static and Dynamic Data. Static data rules cover the building geometry, construction materials, glazing information, building services, and so forth, while dynamic data rules cover the specification of time-varying parameters such as occupancy schedules, internal gains schedules, use of equipment, or occupant actions. A subset of these guidelines are briefly presented below.

Note here that concerning HVAC systems data representation, while IFC is the most widely used openBIM schema, it suffers from limitation in the description of HVAC system in the context of BEPS. Commonly, HVAC modelling in BEPS engines require further information than what is available in an IFC file. Hence, the presented BIM design guidelines, are mostly focused on building energy demands estimation aspects.

### Spaces and HVAC zones

Accurate heating- and cooling-loads simulation can only be accomplished when spaces are created in all internal building closed areas to account for the entire volume of the building model. Spaces should be defined for each closed area of the building, including unoccupied closed air volumes such as plenums. Following the definition of spaces, a checking that they are properly bounded must be performed using floor plans and section views, to guarantee that a) there are no overlaps between the space volume and the attached building elements, and b) there are no void volumes (common case when the space volume is not attached to building elements).

HVAC zones comprise one or more spaces that have common environmental or design requirements. Spaces that are on different levels can be added to the same zone. Zones are rather important for BEPS as they define the granularity of the approximation. Grouping spaces into zones in Revit$^{TM}$ is required to ensure manageable complexity of the BEPS model. When a space has not been assigned to a zone, a new zone for this space is defined automatically.

**Material thermal properties**

In a broad sense, two main material categories exist: Opaque and Transparent. Knowledge of material thermal property values is required in the modelling of the various energy transfers this includes absorbed solar shortwave radiation flux, conduction heat transfer through walls, convective transfer to the space (or outside) air and fraction of sunlight transmitted to building interiors.

For Opaque materials, knowledge of the following thermal properties is required: Thickness, Conductivity, Density, Roughness, Specific Heat, Thermal absorptance, and Solar absorptance. Transparent materials require definition of the following parameters: U-factor and Solar Heat Gain Coefficient (SHGC). In Revit, these material properties can be defined by the designer and can be subsequently accessed through the Revit API. To define and retrieve such information, the BIM designer must activate the thermal properties population for each material through the "Thermal" tab.

Most IFC-to-BEPS methods currently use construction and material thermal properties of building elements from fixed predefined libraries (Thorade et al. 2015, Wimmer et al. 2015). However, as mentioned above, this information can be explicitly prescribed in the Revit™ model, and can also be represented in IFC (e.g. IfcThermalConductivityMeasure for thermal conductivity). Surprisingly, the default Revit IFC-exporter is not able to export such information.

**Internal gains and operation schedules**

Building equipment's operation (artificial lighting and electrical equipment), as well as the occupants' presence and the building infiltration rate, contribute to the overall internal gains and can be represented as time-varying functions, called herein internal gains schedules and assigned to every building's space.

The IFC schema includes classes for representing the internal gains schedules (IfcTimeSeries). Objects of these classes can be attached to individual space instances. However, concerning the existing BIM-authoring tools, these schedules can be selected from predefined variants. In Revit™, such information can be defined in building and/or space scale as follows: a) define the building type, which sets a predefined template for the internal gains (schedules and densities), and its infiltration class; b) define the people, artificial lighting and electric equipment (density and schedules) at each space by selecting the space type. At this point, it is worth-mentioning that the designer is allowed to edit/modify non-schedule parameters (densities), but not modify existing schedules, since these schedules have been defined according to relevant standards (ASHRAE 90.1 2010); if a new schedule is required the modeler must define a user-defined schedule and edit its values properly.

**IFC exporter modification**

To overcome limitations of the default Revit IFC4 DTV exporter on exporting information about thermal properties of each building entity's construction material and internal gains (schedules and densities), which have been properly set following the aforementioned guidelines, and to guarantee correct exportation of such information, the default exporter (which is an open source component) has been modified to create a bespoke OptEEmAL version that addresses these shortcomings. The OptEEmAL IFC Exporter supports the latest version of Revit™ (2018.3).

**Exportation setup**

Assuming that the BIM data insertion according to the guidelines and the OptEEmAL IFC exporter's installation have been accomplished, the BIM designer proceeds with the exportation setup where few of the defaults IFC-exporting options are modified properly to retrieve the requested information.

The first modification is the rooms exportation's deactivation. Utilizing the Revit IFC4 DTV exporter and/or its modified version, both spaces and rooms are exported as IfcSpace instances. While spaces definition throughout the models is prerequisite, rooms definition is not: information that a room object could provide in not usable in the context of BEPS. Hence, to avoid duplication of IfcSpaces' geometry representation, the designer must deactivate the rooms' exportation. To this direction, designer must change the IFC options by editing the rooms Revit™ category and setting "Not Exported" in the respective "IFC Class Name".

Setting the rooms as not exported elements, the file is ready to be exported: the IFC4 Design Transfer View is selected as MVD, while a) the Revit™ property sets and b) the IFC common property sets exportation are activated.

## IFC data geometry check

After the IFC file export geometric errors that affect the IFC data enrichment stage (stage 2 Figure 1), (Lilis et al. 2015) must be detected automatically and corrected manually. These errors belong to a subset of all possible errors in IFC data files (detected by commercial IFC data checking software such as Solibri Model Checker, TEKLA and others) and a dedicated Geometric Error Detection (GED) tool have been developed for their detection.

Errors that are detected by the GED tool belong to multiple categories. For example, an architectural construction (e.g. wall, slab, etc.) that intersects with a building space volume, renders the 2nd-level space boundary surface related to this construction and space undetectable (see example in part I of Figure 2). Two intersecting wall opening volumes next to each other, attached to the same building space (see example in part II of Figure 2), produce 2nd-level space boundaries which also intersect with each other (space boundary surface duplication). For all of these reasons, methods for detection and semi-automatic correction of these inaccuracies have been proposed (Lilis et al. 2015).

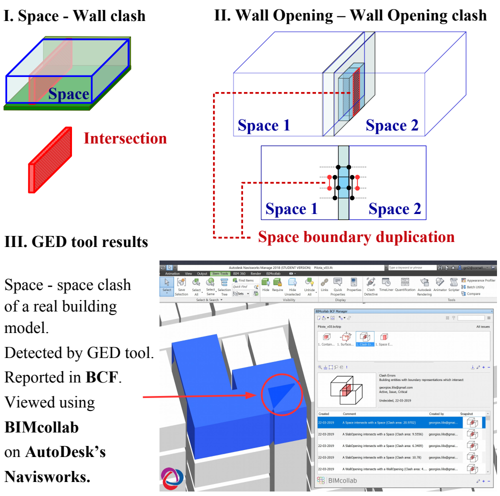


*Figure 2: Geometric errors detected by GED tool*

The GED tool reports the detected errors back to the designer in a BCF file format (part III of Figure 2). The BCF error report file contains the 3D geometric definitions of the architectural elements involved in a geometric error as well as the set of surfaces and line segments indicating at the exact location of the error in the building's three dimensional space.

## IFC data enrichment

After the designer's manual corrections of the aforementioned remaining geometric errors, the exported IFC file is ready to be enriched with the necessary for BEPS geometric content, which is essentially the 2nd-level space boundary surface topology (Bazjanac 2010). This topology contains surfaces through which thermal energy flows among internal building spaces and the building environment (air or ground) and are required in order to assess total thermal energy balance of the building (see example in part II of Figure 3). This surface topology cannot be obtained directly from the geometry of the architectural elements (e.g. walls, slabs, etc.) as the building spaces are not perfectly aligned vertically and horizontally. Additionally, most IFC exporters do not or export this topology or they export it with errors.

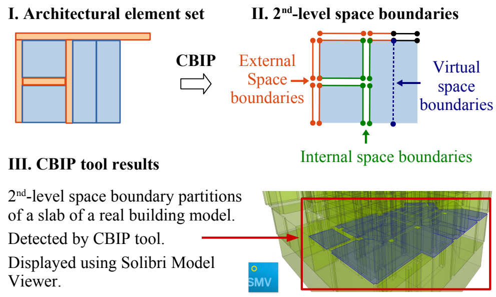


*Figure 3: 2nd-level space boundary surface generation of CBIP tool*

Consequently, to produce this topology in a consistent and precise way, CBIP tool has been developed (Lilis et al. 2017), which performs the necessary geometric transformation operations on the existing IFC architectural geometric content (part I of Figure 3), calculates the 2nd-level space boundary surfaces (part II of Figure 3) and enriches the input IFC file with the required geometric data of these surfaces and their associations to building space volumes and material constructions. As illustrated in the example of part III of Figure 3, CBIP correctly decomposes the floor surface into multiple sub-surfaces depending on the location of the building spaces underneath the slab.

Although the geometric content of the enriched IFC file might be adequate for BEPS model generation (the necessary 2nd-level space boundaries are present with the correct associations to related building elements), additional information related to thermal properties of the materials and time schedules of entities which change state (people, devices, openings, etc.), required for BEPS model generation, might be missing. Consequently, an IFC data completeness checking stage, following the model view definition approach, appears to be necessary, as described in the following section.

## IFC data completeness check

For an IFC-to-BEPS model data transformation, presence of certain IFC data is prerequisite. More specifically, IFC data have to satisfy the exchange requirements of the BEPS input data generation; hence, certain subsets of the IFC schema are validated by evaluating the IFC entities and their attributes in terms of existence, correctness, uniqueness and conditional dependencies (Chipman et al. 2014). There are two main approaches of developing an IFC validation tool. First using an IFC library, the developer can build static rules by querying the IFC entities and checking their attributes. This approach has some weaknesses: a) it does not support reusability of the defined rules; and b) it requires advance programming skills to validate the relations of the IFC entities under a certain IFC subset. The second approach is the development of an IFC validation tool that supports the Model View Definition (MVD) automated checking (Zhang et al. 2015). An MVD describes the subset of the IFC schema that is needed for a data-exchange in a specific domain, which can be formalized using the mvdXML open standard that has been released by buildingSMART International. The IFC validation tools which support MVD checking use the mvdXML format to import the IFC subsets known as templates and the defined rule sets known as concept to perform an automated validation of the IFC data. To develop such a tool, in this work some supporting tools and libraries have been developed, described in the following sections.

### EXPRESS Parsing Tool

The proposed EXPRESS Parsing Tool is a standalone program written in Java using the code model library and is suitable to generate IFC java classes from the corresponding EXPRESS files which have been released by buildingSMART International. The tool can parse successfully the most frequently used IFC releases, from IFC2x3 TC1 to IFC4x1 final. In the first step, the EXPRESS data are transformed into in-memory objects. Consequently, the defined mapping rule set is applied on top of this in-memory representation to initialize the corresponding Java code model. The mapping rules are summarized in Table 1.

*Table 1: Mapping rules between EXPRESS schema and Java model*

| **EXPRESS Model** | **Java Model** |
|---|---|
| ENTITY | Java Class |
| SELECT | Java Interface |
| ABSTRACT SUPERTYPE | Java Abstract class |
| TYPE | Java Class |
| | Enumeration class |
| | Custom list wrapper |
| | Custom array wrapper |
| | Boolean wrapper class |
| | Logical wrapper class |
| | Real wrapper class |
| | Integer wrapper class |
| | String wrapper class |
| | Binary wrapper class |
| INVERSE | Set Collection |

The Java code model has a built-in method to generate the defined IFC classes and to organize them into packages based on the version of the given EXPRESS schema. Each IFC Java class implements some standard methods which are summarized as follows:

- Public constructor without arguments: instantiates an object of an IFC class;
- Public constructor with arguments: the constructor includes all the defined properties (IFC attributes) as well as the properties which are defined in upper levels (IFC abstract classes);
- Public method to initialize the properties of an IFC class;
- Public method to initialize the collections set of the inverse properties;
- Public getters/setters for each property;
- Public method to generate the step line of an IFC class.

Furthermore, the EXPRESS Parsing Tool generates the IFC factory class which allows the instantiation of the IFC objects by using the IFC class names.

### IFC Library

The proposed IFC Library provides the possibility to parse efficiently the IFC-SPF data and to instantiate in-memory the IFC model of the corresponding IFC file. An embedded API is used for the loaded objects manipulation, providing the developers with useful functionalities such as adding new IFC instances, modifying existing instances or querying the IFC model. The core of this library comes from the output of the EXPRESS Parsing Tool and is organized in multiple packages to handle effectively the supported versions of the EXPRESS schema. The current version of the IFC library can handle a wide range of different IFC schemas (IFC2x3 TC1 to IFC4x1 final). Additionally, the IFC library can initialize the properties of the inverse relations by adding the instance of a class to the set collection of the inverse connected property. The initialization of the inverse relations is archived by calling a specific public method as listed next:

*Listing 1: Part of IfcMaterialProperties class*

```
public IfcMaterialProperties(){
}

public void inverse(){
    if( this.materials != null &&
        this.material.getHasProperties() != null){
            this.material.getHasProperties().add(this);
    }
}
```

*Listing 2: Part of IfcMaterialDefinition class*

```
public IfcMaterialDefinition() {
    this.associatedTo =
    new HashSet<IfcRelAssociatesMaterial>();
    this.hasExternalReferences =
    new HashSet<IfcExternalReferenceRelationship>();
    this.hasProperties =
    new HashSet<IfcMaterialProperties>();
}

public void inverse() {
}
```

Using the inverse set collections, it is possible to get the IfcMaterialProperties instance from the inverse method getHasProperties() of the IfcMaterial class. In this example, the IfcMaterial class is subtype of the IfcMaterialDefinition. The inverse method is called for each IFC instance after the deserialization process.

**MVD Checking**

One of the main goals of the MVD specification is to help the community develop BIM tools that allow the automatic validation of IFC data against rule sets that are based on the mvdXML standard. Mainly, three steps are needed to achieve automatic validation of IFC data: a) the preparation of the rule set (mvdXML) using buildingSMART certified tools such as the ifcDoc (see figure 4); b) the model checking service that applies the given rules to the loaded IFC instances; c) the report generation service that generates reports based on open standards such as the BCF.

For the creation of valid mvdXML files the ifcDoc software is used. This software has been developed by buildingSMART International in order to improve the consistent and computer-interpretable definition of the MVD specification. The user can easily create a custom model view and assign new concepts under it. Each concept has an applicable entity (IFC class), a connection to an existing template (subsection of the IFC schema) and the applied rules based on logical operations (IFC entities or attributes that are checked by type or by value under certain conditions).

The proposed MVD checking tool is a standalone service written in pure Java without having external dependencies to third party libraries. The main goal was to achieve a clean end-to-end design where the different parts are separated based on their functionalities. Thus, the EXPRESS Parsing Tool is used to generate the IFC Library which in its turn is used internally by the model view checking process. There are two main features that the checking process uses from the IFC Library: a) the IFC-SPF serialization/deserialization feature to load the IFC data in-memory and to perform queries; and b) the inverse connections of the IFC entities to validate the existence of an IFC schema's subset starting from an applicable IFC root entity. The MVD checking tool takes as an input the IFC-SPF data as well as the MVD-XML data. The output of the tool is the boolean result of the validation for each given concept. The serialization of the results can be represented in a graphical environment or using open standards such as the BCF that supports workflow communication in BIM processes.

From a BEPS viewpoint, the partition of building construction (interior/exterior walls, floors, roofs) and opening (door, window) surfaces into $2^{nd}$-level space boundary (Bazjanac 2010) surfaces is a prerequisite. This process is performed by the CBIP tool. The result of the CBIP tools execution is the enrichment of the IFC file with relevant $2^{nd}$-level space boundary information (IfcRelSpaceBoundary2ndLevel class population).

The $2^{nd}$-level space boundary content of the enriched IFC file is checked by the model view checking process, using the following four checking rules, which are implemented under the same concept using multiple template rules:

1. The existence of the $2^{nd}$-level space boundaries is checked by examining the presence of the IfcRelSpaceBoundary2ndLevel class instances in the enriched IFC file.

2. For each IfcRelSpaceBoundary2ndLevel populated instance, the presence of its related building element is examined.

3. For each IfcRelSpaceBoundary2ndLevel instance, the related building elements type, the value of the ExternalOrInternal property, the corresponding space boundary (if the space boundary is INTERNAL) and the parent space boundary (if the space boundary refers to an opening) are examined.

4. For each IfcRelSpaceBoundary2ndLevel instance the relating space index, is checked.

5. Finally, the last rule checks if there is an IfcRelSpaceBoundary2ndLevel instance with ExternalOrInternal property value "EXTERNAL_EARTH".

If any missing or invalid information is detected the corresponding template rule fails to validate the defined parameters. These parameters are summarized for different space boundary types (internal, external, corresponding and parent) and related building elements (wall, slab, door, window, plate, opening and virtual element) in Table 2.

*Table 2: Space boundary checking rules*

| Applicable IFC Entity | Boundary Condition | Corresponding Space Boundary | Parent |
|---|---|---|---|
| IfcWall | Internal | Missing | - |
| IfcSlab | Internal | Missing | - |
| IfcDoor | External | - | Missing |
| | Internal | - | Missing |
| | Internal | Missing | - |
| IfcWindow | External | - | Missing |
| | Internal | - | Missing |
| | Internal | Missing | - |
| IfcPlate | External | - | Missing |
| | Internal | - | - |
| IfcOpening | External | - | - |
| IfcVirtualElement | External | - | - |

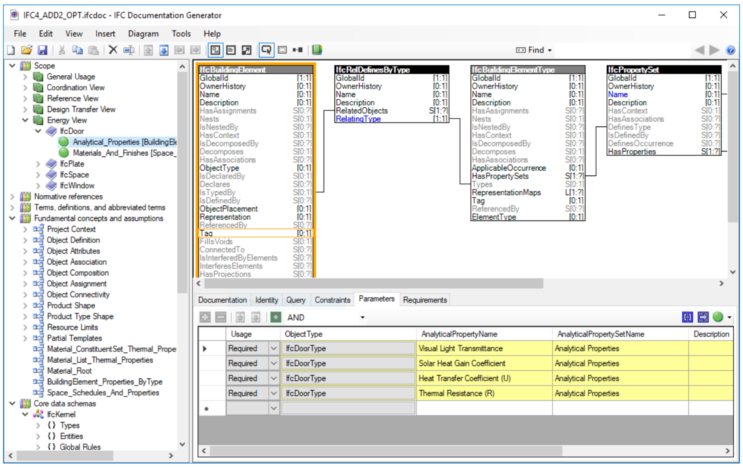

*Figure 4: ifcDOC tool*

As mentioned above, other IFC data that are required in the context of BEPS include the space occupants presence, artificial lighting, electrical equipment operation (which act as thermal sources in terms of BEPS) and related operation schedules. These data are exported by the modified IFC Revit™ exporter by populating a new Pset class, named Pset_Space InternalGainsDesign, assigned to each IfcSpace class. The properties of this new class are checked against the property name and the entity type as listed in Table 3.

*Table 3: Space checking template rules of Pset SpaceInternalGainsDesign*

| Property name | Entity type |
|---|---|
| OccupancySchedule | IfccIreggularTimeSeries |
| LightingSchedule | IfccIreggularTimeSeries |
| EquipmentSchedule | IfccIreggularTimeSeries |
| HeatGainLighting | IfcPropertySingleValue |
| HeatGainEquipment | IfcPropertySingleValue |
| AreaPerOccupant | IfcPropertySingleValue |
| HeatGainPerOccupant | IfcPropertySingleValue |
| InfiltrationRate | IfcPropertySingleValue |

BEPS require a number of properties to be assigned to materials for various building constructions as well. These properties, depending on the material type, are: for opaque materials Conductivity, Density and Specific Heat; for transparent materials U-factor and Solar heat gain coefficient (SHGC). Two new MVD concepts depending on the building element type are applied: a) Opaque materials (e.g. walls, floors, roofs); and b) Transparent materials (e.g. windows, plates, doors).

The properties under the opaque and the transparent materials classes are checked against the property name and the entity type as listed in Table 4 and Table 5, respectively.

*Table 4: Opaque material property rules*

| Property name | Property type |
|---|---|
| SpecificHeatCapacity | IfcPropertySingleValue |
| ThermalConductivity | IfcPropertySingleValue |
| MassDensity | IfcPropertySingleValue |

*Table 5: Transparent material property rules*

| Property name | Property type |
|---|---|
| Visual Light Transmittance | IfcPropertySingleValue |
| Solar Heat Gain Coefficient | IfcPropertySingleValue |
| Heat Transfer Coefficient (U) | IfcPropertySingleValue |
| Thermal Resistance (R) | IfcPropertySingleValue |

## Conclusions

The automated BEPS model generation from IFC data has received lately considerable attention. However, imperfections of the provided IFC files prevent the BIM-to-BEPS data transformation process from being widely and automatically applicable.

The focus of this work has been the proposal of an IFC data quality checking workflow towards populating IFC files ready for BEPS input data generation. The proposed workflow consists of four main stages: a) data preparation and exportation; b) geometric data correctness check; c) data enrichment; and d) data completeness check.

Describing these stages concisely, in the first stage, design guidelines along with a modified IFC exporter are provided to enable exportation of information about materials thermal properties and internal gains. In the second stage, geometric errors that affect the correct 2$^{nd}$-level space boundaries population are automatically detected and reported in a BCF file. These errors are corrected manually by the designer and an geometric error-free IFC that includes information about materials thermal properties and internal gains is exported to be processed by the third stage; here, the IFC data are enriched with the 2nd-level space boundaries data. In the final stage, a data completeness check is performed for the enriched IFC file to ensure that the BEPS information that the IFC schema could capture is present.

The presented workflow has been focused strictly on data preparation for building energy demands estimation aspects, since HVAC data inclusion have not been thoroughly investigated. A concrete implementation of such data in the proposed workflow remains the main subject of future work.

## Acknowledgments

This research has been partially funded by the European Commission H2020-EeB5-2015 project "Optimised Energy Efficient Design Platform for Refurbishment at District Level" under contract #680676 (OptEEmAL).

## References

Andriamamonjy, A., Saelens, D. & Klein, R. (2018), 'An automated IFC-based workflow for building energy performance simulation with Modelica', *Automation in Construction* **91**, 166–181.

ASHRAE 90.1 (2010), 'Energy standard for buildings except low-rise residential buildings', *ASHRAE/IESNA Standard* .

Bazjanac, V. (2010), Space boundary requirements for modeling of building geometry for energy and other performance simulation, *in* 'Proceedings of the 27th CIB W78 Conference', Cairo, Egypt, 16-19 November 2010.

Berkeley, P., Haves, P. & Kolderup, E. (2014), Impact of modeler decisions on simulation results, *in* 'Proceedings of the ASHRAE/IBPSA-USA Building Simulation Conference'.

Chipman, T., Liebich, T. & Weise, M. (2014), 'Specification of a standardized format to define and exchange model view definitions with exchange requirements and validation rules (b1. 0)'.

Giannakis, G., Lilis, G. N., Garcia, M., Kontes, G., Valmaseda, C. & Rovas, D. (2015), A methodology to automatically generate geometry inputs for energy performance simulation from IFC BIM models, *in* 'Proceedings of the 14th IBPSA Building Simulation Conference', Hyderabad, India, 7-9 December 2015, pp. 504–511.

Hitchcock, R. & Wong, J. (2011), Transforming IFC Architectural View BIMs for Energy Simulation, *in* 'Proceedings of the 12th IBPSA Building Simulation Conference', Sydney, Australia, 14-16 November 2011, pp. 1089–1095.

ISO 16739-1 (2018), 'Industry Foundation Classes (IFC) for data sharing in the construction and facility management industries – Part 1: Data schema', *International Organisation for Standardisation* .

Lilis, G. N., Giannakis, G. I. & Rovas, D. V. (2017), 'Automatic generation of second-level space boundary topology from IFC geometry inputs', *Automation in Construction* **76**, 108–124.

Lilis, G. N., Giannakis, G. & Rovas, D. (2015), Detection and semi-automatic correction of geometric inaccuracies in IFC files, *in* 'Proceedings of the 14th IBPSA Building Simulation Conference', Hyderabad, India, 7-9 December 2015, pp. 2182–2189.

Maile, T., Fischer, M. & Bazjanac, V. (2007), 'Building energy performance simulation tools-a life-cycle and interoperable perspective', *Center for Integrated Facility Engineering (CIFE) Working Paper* **107**, 1–49.

Pinheiro, S., Wimmer, R., ODonnell, J., Muhic, S., Bazjanac, V., Maile, T., Frisch, J. & van Treeck, C. (2018), 'MVD based information exchange between BIM and building energy performance simulation', *Automation in Construction* **90**, 91–103.

Rose, C. M. & Bazjanac, V. (2015), 'An algorithm to generate space boundaries for building energy simulation', *Engineering with Computers* **31**(2), 271–280.

Thorade, M., Rädler, J., Remmen, P., Maile, T., Wimmer, R., Cao, J., Lauster, M., Nytsch-Geusen, C., Müller, D. & van Treeck, C. (2015), An open toolchain for generating modelica code from building information models, *in* 'Proceedings of the 11th International Modelica Conference', pp. 383–391. Versailles, France, 21-23 September 2015.

Wimmer, R., Cao, J., Remmen, P., Maile, T., ODonnel, J., Frisch, J., Streblow, R., Müller, D. & van Treeck, C. (2015), Implementation of advanced bim-based mapping rules for automated conversions to modelica, *in* 'Proceedings of the 14th IBPSA Building Simulation Conference', pp. 419–426. Hyderabad, India, 7-9 December 2015.

Zhang, C., Beetz, J. & Weise, M. (2015), 'Interoperable validation for ifc building models using open standards', *Journal of Information Technology in Construction (ITcon)* **20**(2), 24–39.