DIPLOMARBEIT / MASTER THESIS

BIM TO BEM TRANSFORMATION WORKFLOWS: A CASE STUDY COMPARING DIFFERENT IFC-BASED APPROACHES

unter der Leitung von

Univ.Prof. DI. Dr.techn. Ardeshir Mahdavi

&

Sen.Sci. DI. Dr.techn. Ulrich Pont

E 259-03 Forschungsbereich für Bauphysik und Bauökologie

Institut für Architekturwissenschaften

eingereicht an der

Technischen Universität Wien

Fakultät für Architektur und Raumplanung

von

Oliver Spielhaupter, B.Eng.

11810905

Humboldtgasse 10/6/6004

Wien, im September 2021

ABSTRACT

In the current methodologies amongst AEC (Architecture-Engineering-Construction)-professionals, interdisciplinary collaboration with BIM (Building Information Modelling) processes is considered as state of the art.

Regularly, no common information transformation routine from BIM to BEM (Building Energy Modelling) can be observed. As a result, engineers doing BEM are confronted with a cumbersome, error-prone, time- and cost-intensive manual data transfer process.

In the scope of this thesis a review of some available semi-automated approaches for information transformation from a common data format and structure used in BIM, namely Industry Foundation Classes (IFC), to a state-of-the-art building simulation environment, namely EnergyPlus, is performed.

Thereby, a case study is conducted for two chosen workflows. The case study is based on an artificial building project modelled based on BIM-methodologies in detail in the BIM-authoring tools ArchiCAD and Revit. These building models are afterwards exported to the commonly used data exchange format IFC. Each of the IFC files is afterwards transformed via one unique transformation process to an EnergyPlus Input Data File (IDF). The IFC file based on the ArchiCAD model is transformed via Python scripts using several libraries. The IFC file based on the Revit model is transformed via the serializer OsmSerializer integrated in the IFC file server BIMserver.

The quality and validity of the resulting geometry and semantic data are analysed. Additionally, the simulation results are collated to each other and a base case model, which is manually created via SketchUp and the OpenStudio plug-in. Concluding the validity and comparability of the simulation results are evaluated to indicate the current state of development of the two chosen workflows.

The results of the study show the following aspects (i) issues and restrictions concerning the creation of BIM-models for the use of BEM; (ii) issues concerning the transformation of geometry and semantic data from IFC to IDF with the two chosen workflows; (iii) deviations of the simulation results. However, the results show that there are possibilities to enhance the overall workflow of BIM-based BEM by using parts of the two evaluated workflows, e.g., automatic building construction and material transformation.

Keywords. Building Energy Modelling (BEM); Building Information Modelling (BIM); Dynamic Energy Simulation; Industry Foundation Classes (IFC); Model Transformation

KURZFASSUNG

In der aktuellen Methodik der Gebäudeplanungs-Fachleuten wird die interdisziplinäre Zusammenarbeit mit BIM (Building Information Modelling)-Prozessen als Stand der Technik angesehen.

In der Regel ist keine Informationstransformationsroutine von BIM zu BEM (Building Energy Modelling) implementiert. Dies hat zur Folge, dass Ingenieure, die BEM betreiben, mit einem umständlichen, fehleranfälligen, zeit- und kostenintensiven manuellen Datentransferprozess konfrontiert sind.

Im Rahmen dieser Arbeit wird ein Überblick über einige verfügbare halbautomatische Ansätze zur Informationstransformation von einem in BIM standardmäßig genutzten Datenformat, IFC, zu einer Gebäudesimulationsumgebung, EnergyPlus, durchgeführt.

Es wird eine Fallstudie für zwei ausgewählte Informationstransformationsabläufe durchgeführt. Die Fallstudie basiert auf einem Gebäude, das detailliert mit BIM-Methoden in einer BIM-Umgebung in ArchiCAD und Revit modelliert wird.

Diese Gebäudemodelle werden anschließend in das Datenaustauschformat IFC exportiert. Beide IFC-Dateien werden anschließend über einen Transformationsprozess in ein EnergyPlus Input Data File (IDF) umgewandelt. Die IFC-Datei, die auf dem ArchiCAD-Modell basiert, wird über Python-Skripte unter Verwendung verschiedener Bibliotheken transformiert. Die IFC-Datei, die auf dem Revit-Modell basiert, wird über den Übersetzer OsmSerializer, der in den IFC-Dateiserver BIMserver integriert ist, transformiert.

Die Qualität und Gültigkeit der resultierenden Geometrie und der semantischen Daten werden analysiert. Zusätzlich werden die Simulationsergebnisse miteinander und mit einem Basisfallmodell verglichen, das manuell mit SketchUp und dem OpenStudio-Plug-in erstellt wurde. Abschließend werden die Nutzbarkeit und Vergleichbarkeit der Simulationsergebnisse bewertet, um den aktuellen Entwicklungsstand der beiden gewählten Workflows aufzuzeigen.

Die Ergebnisse der Studie zeigen folgende Aspekte auf: (i) Probleme und Einschränkungen bei der Erstellung von BIM-Modellen für die Nutzung von BEM; (ii) Probleme bei der Transformation von Geometrie- und semantischen Daten von IFC zu IDF mit den beiden gewählten Workflows; (iii) Abweichungen der Simulationsergebnisse. Die Ergebnisse der Studie zeigen jedoch, dass es Möglichkeiten gibt, den Gesamtprozess von BIM-basierten BEM zu verbessern, indem Teile der beiden evaluierten Workflows, z. B. die automatische Gebäudekonstruktions- und die Materialtransformation, verwendet werden.

Stichwörter. Building Energy Modelling (BEM); Building Information Modelling (BIM); Dynamische Gebäudesimulation; Industry Foundation Classes (IFC); Modelltransformation

ACKNOWLEDGEMENTS

First, I would like to thank my supervisors for the helpful input and advice they provided me during the work on this thesis.

To my family and all my friends who have supported me all my life. Thank you for being there.

To my love. Without you, I would not have been able to accomplish this thesis. Thank you for all your support, believing in me, and walking through life with me.

ABBREVIATIONS

AEC	Architecture-Engineering-Construction
ANSI	American National Standards Institute
API	Application Programming Interface
BEM	Building Energy Modelling
BIM	Building Information Modelling
BREEAM	Building Research Establishment Environmental Assessment Method
CAD	Computer Aided Design
CBIP	Common Boundary Intersection Projection
CSV	Comma-separated values
DXF	Drawing Interchange Format for AutoCAD
DWG	CAD drawing format for AutoCAD
gbXML	Green Building Extensible Markup Language, Green Building XML
GUI	Graphical User Interface
HS	Honeybee Schema
HTML	Hypertext Markup Language
HVAC	Heating, Ventilation, Air-Conditioning
IDA ICE	IDA Indoor Climate and Energy
IDD	EnergyPlus Input Data Dictionary
IFC	Industry Foundation Classes
IFC2x3	IFC2x Edition 3, Technical Corrigendum 1
IFC4	IFC4, Addendum 2, Technical Corrigendum 1
IDF	EnergyPlus Input Data File
IDM	Information Delivery Manual
IDP	Integrated Design Process
JSON	JavaScript Object Notation
LEED	Leadership in Energy and Environmental Design
LOD	Level of Development

MVD	Model View Definitions
OSM	OpenStudio Model
PHPP	Passive House Planning Package
SDK	Software Development Kit
SimModel	Simulation Domain Model
STEP	Standard for the Exchange of Product model data
UTF-8	8-Bit Universal Coded Character Set Transformation Format
XML	Extensible Markup Language

CONTENTS

1 INTRODUCTION

1.1 Overview

Building Energy Modelling (BEM) is a growing topic with increasing importance for, e.g., building permission, certification systems or evaluation for possible refurbishment measures. The usage rate of Building Information Modelling (BIM) in the AEC (Architecture-Engineering-Construction), especially in big building projects, is rising. A lot of the information required for BEM is available in BIM. Information transfer from BIM to BEM is an important but still not yet mature topic. Several approaches to transfer information from BIM to data formats used in BEM were developed. In the scope of this thesis, a review of some available BIM-based BEM transformation workflows is performed. The application of two of these workflows on a case study is conducted to identify possibilities, requirements, and limitations.

1.2 Motivation

Increasing requirements due to stricter energy standards and rising energy efficiency of buildings lead to increasingly complex design processes. Therefore, a transition of the conventional design process to an integrated design process (IDP) is required. In the scope of the IDP digital tools are an important factor (Larsson 2009).

BIM is an application of these tools in combination with the IDP. The usage of BIM in the building industry is increasing. Due to a lack of usability and yet unsolved problems in transforming information, there are currently restrictions in the automatic transfer of data from BIM to BEM. In current BIM projects the transfer of data between the different disciplines architects, structural engineers and mechanical engineers is done via shared IFC files (O'Donnell et al. 2019). An automated information transfer from BIM to BEM based on IFC would rapidly decrease the necessary effort for engineers in the field of BEM and would furthermore enhance the collaboration in a recursive building design process (Ramaji et al. 2016).

In recent years, the demand for sustainable building certification according to U.S. Green Building Councils LEED (Leadership in Energy and Environmental Design) or Building Research Establishments BREEAM (Building Research Establishment Environmental Assessment Method) has increased significantly (Schwartz et al. 2013). Easy to use tools such as DesignBuilder (DesignBuilder Software Ltd 2021) or IES VE (Integrated Environmental Solutions Limited 2020) used in the framework of these certification processes enhance energy simulations' usability all over the world.

Without the requirements to do the error-prone manual work of creating in-detail geometry models with well-founded properties, definition of simulation properties, etc., it is possible to do energy simulations with pre-defined assumptions or simplifications. The required efforts and knowledge to produce energy simulation results is therefore decreasing (Picco et al. 2015).

Ramaji et al. (2020a) developed a tool to simplify the transformation process by pre-transforming the Industry Foundation Classes (IFC) output file of BIM authoring tools to analytical IFC files with an optimized schema for further processing. Automatic mapping of properties according to object names is one functionality of this tool called Interpreted Information Exchange.

However, pre-defined assumptions or general mapping of properties to multiple different objects significantly impact the outcome of the energy simulations in these simplified workflows. This could lead to decreasing accuracy and reliability of the simulation results. Therefore, the development of BIM-based BEM workflows with clearly defined requirements for each step of the workflow is needed. The outcome are reliable, accurate and high-quality energy simulations with replicable results.

It is however important to consider the impact of usability of software tools used in the workflows. According to Nielsen (1994), usability is commonly described by the five aspects (i) learnability, (ii) efficiency, (iii) memorability, (iv) errors and (v) satisfaction. These aspects can be summarized by setting some basic requirements to the software interface. (i) It must be easy to get started with the software, (ii) there should be a degree of productivity if the software is used on a professional level, (iii) the required effort for relearning the software should be minimized if it is not used on a regular basis, (iv) there should be a system to minimize the occasion of errors, (v) the overall perception should be to fulfil the demands of the users and to motivate the user to use the tool again (Nielsen 1994). Without satisfying usability perceptions by single users, it is unlikely that even the most accurate and high-quality workflow is used on a regular basis by a broad target audience.

1.3 Objective

The objective of this master thesis is twofold and therefore divided into two research questions: (i) an overview of the current state of semi-automated IFC-based BIM-to-BEM transformation workflows for thermal building performance simulation is provided, (ii) some of the presented routines are comprehensively tested and evaluated in the scope of a case study. The routines aspects depicted in Figure 1 are evaluated in the scope of this case study.

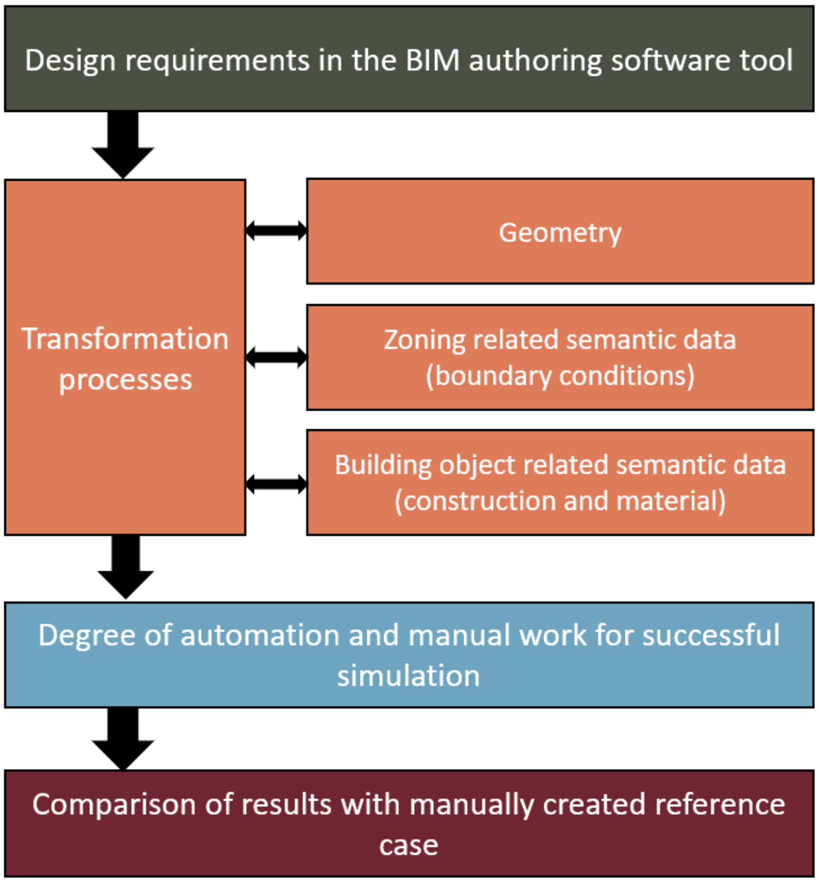

Design requirements in the BIM authoring software tool
Transformation processes
Geometry
Zoning related semantic data (boundary conditions)
Building object related semantic data (construction and material)
Degree of automation and manual work for successful simulation
Comparison of results with manually created reference case

Figure 1, evaluated aspects in the scope of this thesis (own illustration)

The topic of BIM-based BEM is within the AEC context one of the fast-developing topics. Thus, the references and efforts presented in the context of this thesis represent a selection, but not necessarily the full scope of efforts in this field.

2 BACKGROUND

During the design phase of a building project early-stage energy modelling can enhance the control of costs and time of the whole design process. This phenomenon is illustrated in Figure 2 in the so-called MacLeamy curve. With advancing project progress the ability to change functionalities in resulting costs is decreasing exponentially whereas the costs resulting from design changes are rising exponentially. By shifting the energy modelling and evaluation of different variants to an earlier stage of the design process it can effectively impact functional abilities with relative low-cost efforts compared to a later stage.

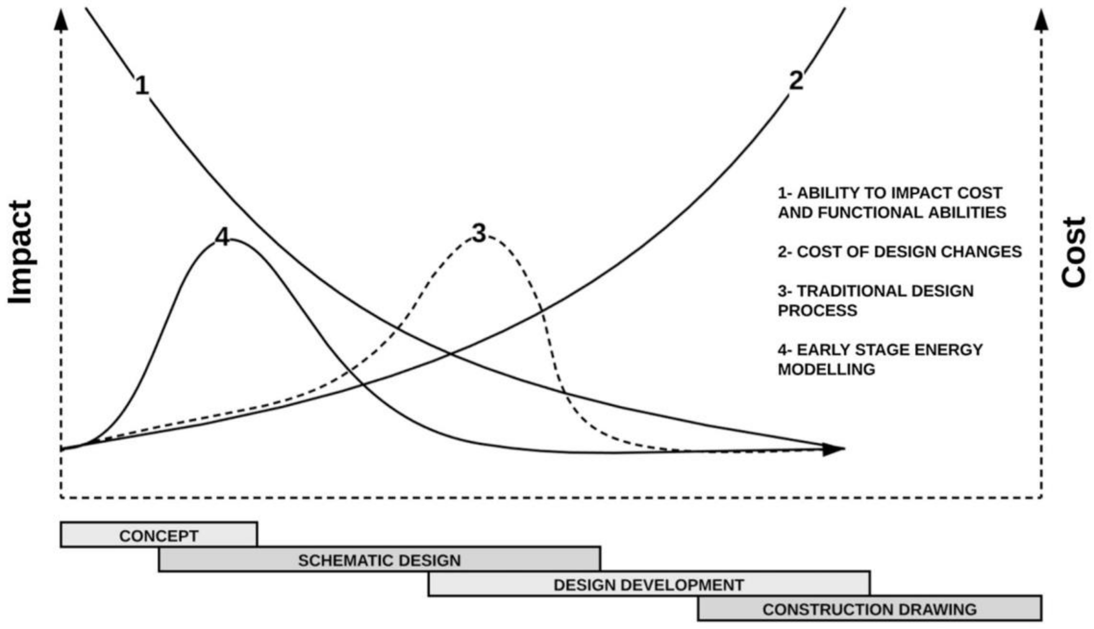

Impact
Cost
1
2
3
4
1- ABILITY TO IMPACT COST AND FUNCTIONAL ABILITIES
2- COST OF DESIGN CHANGES
3- TRADITIONAL DESIGN PROCESS
4- EARLY STAGE ENERGY MODELLING
CONCEPT
SCHEMATIC DESIGN
DESIGN DEVELOPMENT
CONSTRUCTION DRAWING

Figure 2: MacLeamy Curve (Gao et al. 2019) after (Chopson et al. 2015)

In current design processes BIM applications, such as creating a BIM model, are often implemented in an early stage. Using the defined information in the BIM model for energy modelling is another step to improve the design process in this early stage and therefore improve the overall design process significantly.

A literature review of current research regarding the different topics related to BIM-based BEM transformation is presented in the following sections: (i) an introduction to BIM and the concepts associated with the implementation of data exchange definitions, (ii) basics for BEM, especially dynamic energy simulation and commonly used software tools, (iii) the interaction of BIM and BEM along with data schemas for interoperability and information transformation, (iv) workflows for the semi-automated BIM-based transformation to BEM.

2.1 BIM

With the rising complexity of building projects, the importance of an IDP is also rising. Digital tools are essential to manage the information related to the building project. BIM is a concept that was developed as an approach to standardise digital information management (Borrmann et al. 2018).

BIM includes different kinds of information of the building project throughout the building's lifecycle. Geometric representation is one part. Non-geometric data related to the geometric representation is another. Abstract semantic data of the project, such as project participants or the location of the building site is also stored in the model (Borrmann et al. 2021).

Throughout the lifecycle of a project, it is often not possible to use one single BIM model in all phases. The amount of required data and detail level in different parts of the lifecycle can vary widely. For example, other information might be relevant for the tendering process than for the facility management during the usage of the building. For this purpose, BIM models are described with their level of development (LOD). BIM models of the same project with different LODs might vary in content of information and the level of detail. It is therefore necessary to set up clear definitions of the BIM's LOD for further processing (Eastman et al. 2018).

First attempts to implement BIM in the AEC were conducted with the closed BIM approach. In this approach all project participants use the same proprietary software or at least the same data model of a proprietary software tool. Interoperability with users of the same software but other not standardised configurations or users of different software tools is cumbersome and error-prone (Plandata GmbH n.d.). Open BIM, in contrast to closed BIM, is not reliant on one proprietary software or data format but is based on neutral standards and aims for seamless interoperability of all project participants. This approach improves the overall process of BIM by enhancing the general management of the information in the BIM model. Several standards and vendor neutral formats such as Industry Foundation Classes (IFC) were developed to provide the AEC industry tools to work in an open BIM environment (buildingSMART International 2021e). Due to the complexity of the rising amount of information of different kinds, it is required to define standardised ways to interchange this information with other project participants.

2.1.1 Information delivery manual

Information Delivery Manual (IDM) is a format of definitions for data communication of a specific topic, capable of being read by human users. The goal is a data stream that is also successfully interpretable by software tools (buildingSMART International 2021d). The US GSA (2009) developed an IDM for BIM-based energy analysis. Many workflows that were developed afterwards are based on this IDM.

2.1.2 Industry Foundation Classes

IFC is an open standard according to ISO 16739-1:2018 which can describe buildings and related structures in a standardised way. It is developed and supported by buildingSMART. IFC is a digital hierarchy-based data format written in the Express language to describe information relating to projects. There are different formats such as Extensible Markup Language (XML), JavaScript Object Notation (JSON) and Standard for the Exchange of Product model data (STEP) that can encode IFC data (buildingSMART International 2020c).

IFC enables software developers, designers, and further users of BIM to transfer information in a standardised and consistent way. Data exchange in the scope of an integrated design workflow is enhanced (Hitchcock et al. 2011).

Different versions of IFC are available. According to buildingSMART International (2021b) the two official versions are at the moment IFC2x Edition 3, Technical Corrigendum 1 (IFC2x3) and IFC4, Addendum 2, Technical Corrigendum 1 (IFC4). The level of development of the different IFC versions is depicted in Figure 3. Currently there are concepts in use that are not yet mature. However, there are promising possibilities yet to come.

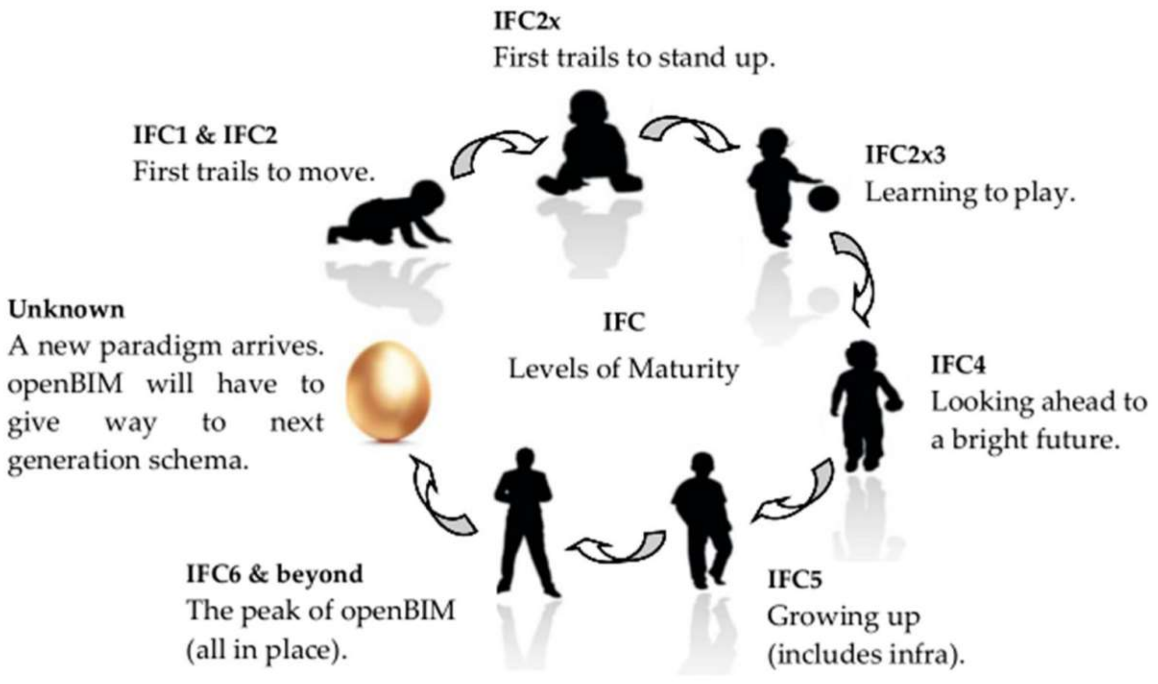

IFC2x
First trails to stand up.
IFC1 & IFC2
First trails to move.
IFC2x3
Learning to play.
Unknown
A new paradigm arrives. openBIM will have to give way to next generation schema.
IFC
Levels of Maturity
IFC4
Looking ahead to a bright future.
IFC6 & beyond
The peak of openBIM (all in place).
IFC5
Growing up (includes infra).

Figure 3: maturity levels of IFC (Jiang et al. 2019)

Knowledge about the IFC structure is necessary to fully understand which information can be defined and how it can be defined in the IFC schema. The IFC data model is structured in four different layers, where "elements in the upper layers can reference elements in the layers below but not vice versa" (Borrmann et al. 2018). The structure of the layers and the inherited classes are depicted in Figure 4. The three most relevant layers for BIM-based BEM workflows are: (i) a core layer with the classes for connections, so-called relationships, between elements and classes that define the spatial elements (ii) an interoperability layer with its classes that describe the most common building elements, such as IfcWall, or IfcWindow, and (iii) a resource layer with classes that contain information concerning the geometry and material properties (Borrmann et al. 2018).

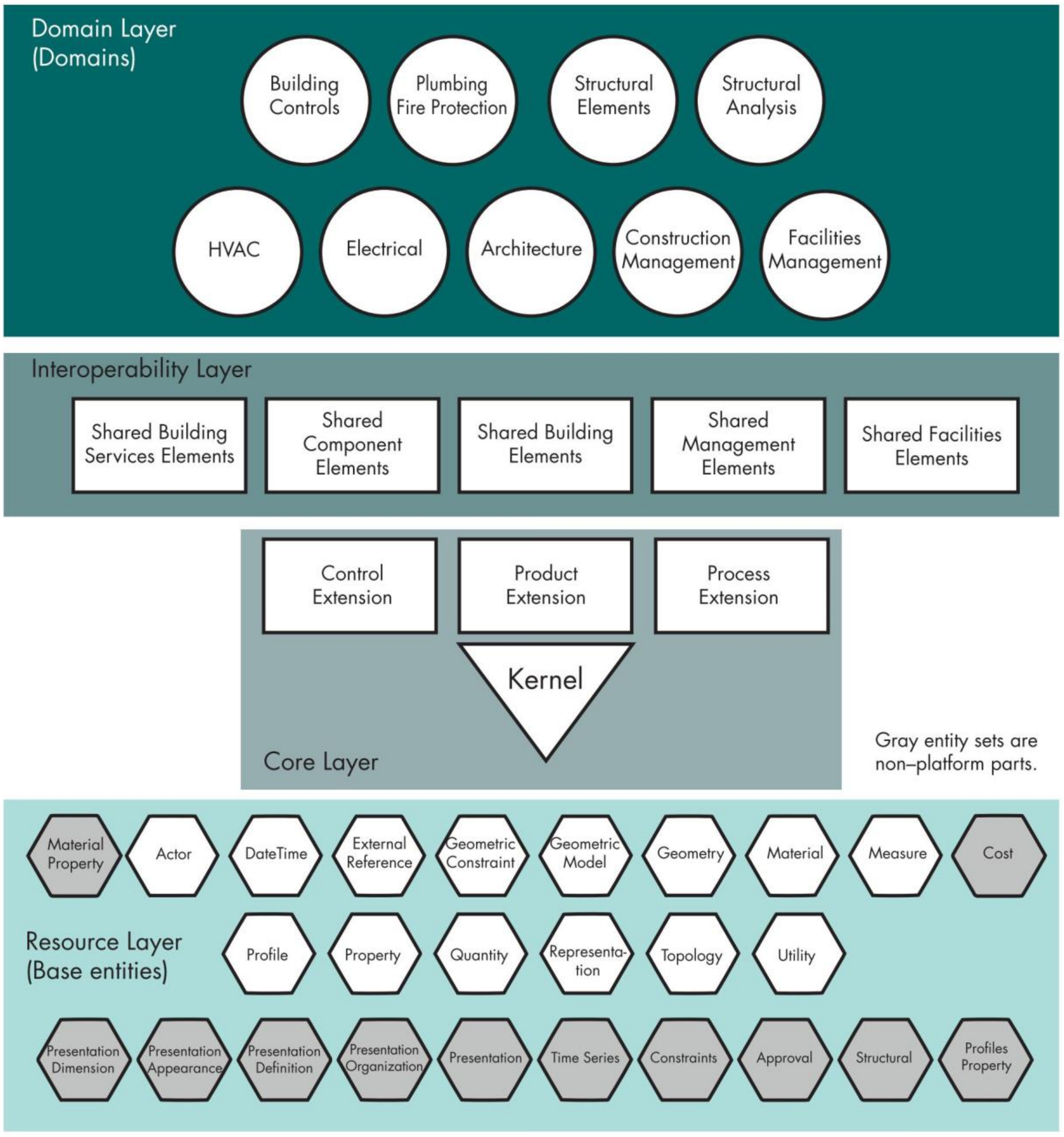


Figure 4: IFC schema and layers (Eastman et al. 2018) after (buildingSMART International 2020b)

IFC is encoded in the American National Standards Institute (ANSI) character set (buildingSMART International 2020a), which includes several characters that are used commonly in the western language area. This might lead to difficulties when information is transformed to other file formats that are not ANSI encoded. Transformation of special characters that are language-wise specific might be required.

IFC data models can have a high amount of information and, therefore, a high level of complexity. However, most of the time only a subset of the available information is required for a seamless data exchange. Unnecessary transfer of information can cause errors and should be avoided. A solution to create standardised subsets of the IFC schema is the usage of Model View Definitions (MVDs).

2.1.3 Model View Definitions

An MVD describes a subset of the complete IFC schema. MVDs define how objects and information are represented in building information models created for a specific usage (Ramaji et al. 2016). The goal of an MVD is to increase the replicability and transparency of data exchange between different software tools (buildingSMART International 2020d). Several standardised MVDs for various purposes were defined over the recent years.

2.2 BEM

Building Energy Modelling relevance has been on the rise in recent years. BEM can provide more reproducible and well-founded results than simplified methods that are still used in the AEC industry, such as manual calculations, rules of thumb and experience based predictions (Gao et al. 2019). Furthermore, due to the increasing demand for green building certification programs such as LEED or BREEAM the need for energy simulations is increasing (Schwartz et al. 2013). Closed system simulation tools and simulation tools with integrated mechanics to manually create simplified buildings dominate the field (Gao et al. 2019). Dynamic energy simulations are used in the field of building performance simulation to achieve realistic and reliable results.

2.2.1 Dynamic Energy Simulation

A dynamic simulation tool has different boundary conditions for distinct time steps. That is contrary to a static simulation tool, such as Excel-Files used for energy certificates in certain countries. Commonly used energy simulation tools require some input that is processed in a simulation engine. The result of the simulation is delivered as an output file which can be further processed. To simplify the usage of a simulation tool, a graphical user interface (GUI) is usually used to define the input and optionally evaluate the output. The general workflow of an energy simulation tool is depicted in Figure 5.

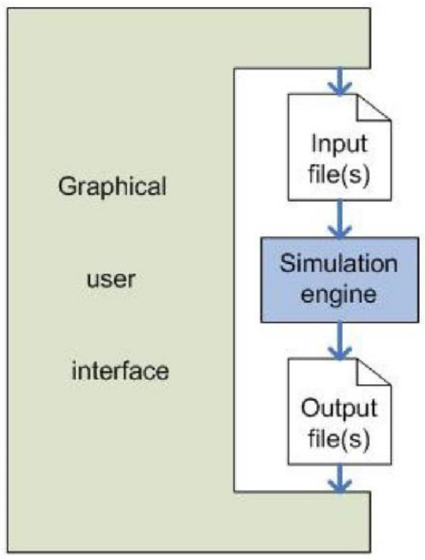


Figure 5: general energy simulation tool workflow (Maile et al. 2007)

Historically different dynamic energy simulation tools were developed and are currently established under building simulation professionals. According to Gao et al. (2019) some of the most-commonly used environments are:

- EnergyPlus (U.S. Department of Energy 2021)
  EnergyPlus is a building energy simulation software engine for energy consumption and process loads. It is managed by the National Renewable Energy Laboratory (NREL). There is no embedded GUI for EnergyPlus. Input is either done via text-based manual input or with the IDF Editor, which supports users with the basic structure and hints for the available classes of EnergyPlus. OpenStudio can be used as an interface for EnergyPlus. However, OpenStudio has its own data format OpenStudio Model (OSM). Another GUI for EnergyPlus is the SketchUp plugin Euclid (Big Ladder Software LLC 2021) which was developed based on a previous version of the OpenStudio plugin. Euclid is based on the EnergyPlus Input Data Format (IDF). The schema format for the IDF is defined in the Input Data Dictionary (IDD) (Alliance for Sustainable Energy, LLC).
- TRNSYS (Thermal Energy System Specialists, LLC 2019)
  TRNSYS is a software environment for the simulation of transient systems. The software is used for several different purposes, such as energy modelling, building performance simulation, and other systems with dynamic behaviour. Combining the TRNSYS interface TRNBuild with the SketchUp plugin TRNSYS3D makes it possible to do dynamic building performance simulations with a GUI.
- IDA Indoor Climate and Energy (EQUA Simulation AB 2020)
  IDA Indoor Climate and Energy (IDA ICE) is a flexible modular software tool for

simulation of buildings, integrated systems and related controls. Advantages over the previously mentioned tools are the extensive possibilities to set up complex technical equipment relatively easy. Complex systems with demand control can be created in an interactive GUI. Among the presented tools, this software tool is the most user-friendly one and therefore often used by engineers.

- Modelica (Wetter 2009)
  The recently emerging, modular, equation-based simulation engine, is highly suitable for simulating new technologies that are cumbersome tom implement in the previous mentioned engines.

In the frame of this thesis, EnergyPlus has been chosen as the desired simulation engine due to the easy-to-use manual check-up possibility of its input data file (IDF). Other advantages are its open-to-public availability (free to use) and the large number of studies indicating the high quality of producible results.

2.3 Interoperability between BIM and BEM

As previously mentioned, the importance of BEM in the AEC environment is rising. Nevertheless, BEM workflows are often implemented with a high effort by AEC professionals with non-standardised settings. A recursive planning process intensifies these efforts. BIM-based BEM is a promising solution to overcome these issues. Standardised semi-automated BIM-based workflows might replace the tedious and error-prone manual work and save time and costs. Additionally, BIM-based BEM workflows can increase consistency and accuracy when performing BEM in an iterative design process (Gao et al. 2019). The major advantages of BIM-based BEM are shown in Figure 6.

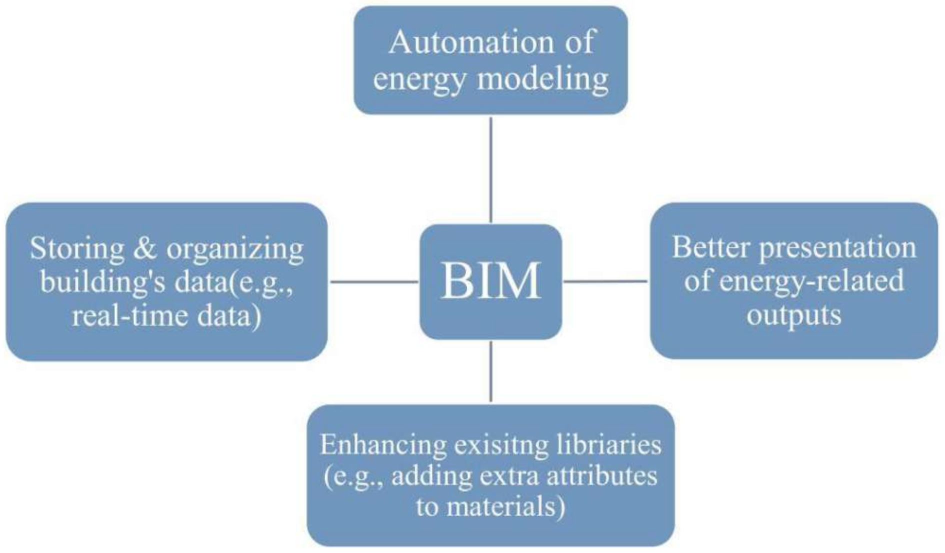

Automation of energy modeling
Storing & organizing building's data(e.g., real-time data)
BIM
Better presentation of energy-related outputs
Enhancing exisitng libriaries (e.g., adding extra attributes to materials)

Figure 6: major advantages of BIM-based BEM *(Kamel et al. 2019)*

Manual information definition for energy simulation produces non-standardised simulation building models. The results with the same geometric input model can vary greatly because they are highly influenced by assumptions made in the process of manual building information definition. Therefore, BIM-based BEM workflows can overcome this inconsistency and lead to standardised results due to fewer assumptions made on professional experience (Hitchcock et al. 2011).

Due to different definitions of architectural spaces and energy simulations thermal zones it is important to specify the thermal zones for an accurate simulation result. This might be done in the process of designing a BIM model with the export to IFC for further usage in energy simulations in mind. However, even with one standardised input model for different transformation workflows the results might vary significantly (O'Donnell et al. 2019).

As depicted in Figure 7, the ideal BIM-based BEM workflow should obtain almost all the required data from the BIM model. The required data for a dynamic energy simulation can be grouped into different categories: 1) localization and the related weather file, 2) geometry, 3) construction and materials, 4) space types and their affiliation to thermal zones, 5) space loads and 6) heating, ventilation, air-conditioning (HVAC) systems and components (Maile et al. 2007).

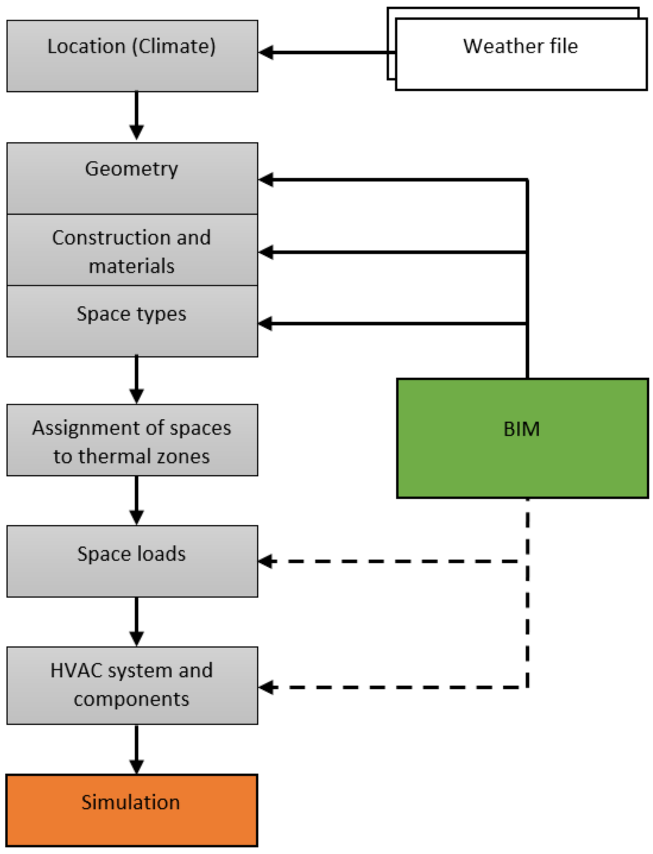

Location (Climate)
Weather file
Geometry
Construction and materials
Space types
BIM
Assignment of spaces to thermal zones
Space loads
HVAC system and components
Simulation

Figure 7: Ideal BIM-based BEM workflow (Maile et al. 2007)

The transition of BIM to the simulation tool can be cumbersome. Tools with pre-set options for simplifying BIM models within the software are prone to errors. The recreation of models in the simulation tool however can be time-consuming and labour-intensive. Therefore, a semi-automated approach with the requirement of BIM usage for BEM should be kept in mind for the model creation in the BIM authoring tool (Fernald et al. 2018).

BIM does not contain information regarding weather conditions (Nasyrov et al. 2014). Information concerning the location of the building project can be derived from the BIM model. However, this information cannot be read and processed by EnergyPlus. An appropriate weather file must be selected based on the location of the building. This is either possible manually or with advanced algorithms coupled to weather file databases (Pont 2014).

Known problems of semi-automated BIM-to-BEM workflows are limitations to simplified geometries (Kamel et al. 2019; O'Donnell et al. 2019), resulting in corrupted geometries (Somboonwit et al. 2017), and missing semantic data (Nasyrov et al. 2014).

The data formats Green Building extensible Markup Language schema (gbXML) and IFC were evaluated regarding the semi-automated usage in building simulation with the software EnergyPlus by Ivanova et al. (2015). Since then, tools for the transfer of data between BIM and BEM have been further developed and enhanced. Recently a new approach using BIM-Server and the EnergyPlus interface OpenStudio has been developed (Ramaji et al. 2020b). Further workflows are software independent and process the raw IFC file via programming scripts that use special libraries, such as IfcOpenShell (Andriamamonjy et al. 2018). This can be useful because some issues in the data transfer from BIM to BEM are caused by the BIM authoring or intermediate tools (Ivanova et al. 2015).

Other approaches to increase the transparency and replicability of BIM-based BEM workflows with different project participants were also developed. Shadrina et al. (2020) proposed a workflow with target values that must be fulfilled by IFC models provided by an architect to ensure usability in thermal simulations. Transfer of geometry, material and construction information from BIM for energy performance certificate calculations was evaluated in a variant study by Reisinger et al. (2019). Another possible usage of BIM models for building design was proposed with PassivBIM, a tool to enable interoperability between BIM and the low energy design software Passive House Planning Package (PHPP) (Cemesova et al. 2015).

The three most frequently used kinds of information transformation workflows are (Ramaji et al. 2020b):

1) export BEM directly from BIM

2) import and transformation of BIM files in a BEM tool

3) transformation of BIM into BEM outside of BIM and BEM tools

This thesis focuses on information transformation workflows listed as the third option: transformation outside BIM and BEM tools. Contrary to the first and second option, this provides possibilities to standardise transformation workflows for different BIM authoring and BEM software solutions. However, some available workflows use common file formats of BEM tools as an intermediate step. Therefore, a strict differentiation between the different kinds of workflows is difficult. Currently developed workflows for BIM-based BEM encompass these categories only partly.

To transform information outside of BIM and BEM tools, it is required to choose a data file schema that acts as an intermediate schema. Due to varying purposes of the schemas, different topologies were developed. The topology, specifications and limitations of the data file schemas have a severe impact on the overall BIM-based BEM transformation process. Therefore, it is required to have a closer look at the available schemas, their properties, possibilities, and limitations.

2.3.1 Data file schema

An interface between the different BIM and BEM tools is required for successful interaction. The different approaches with and without intermediate schema are illustrated in Figure 8, the interfaces with high maintenance effort are shown in orange.

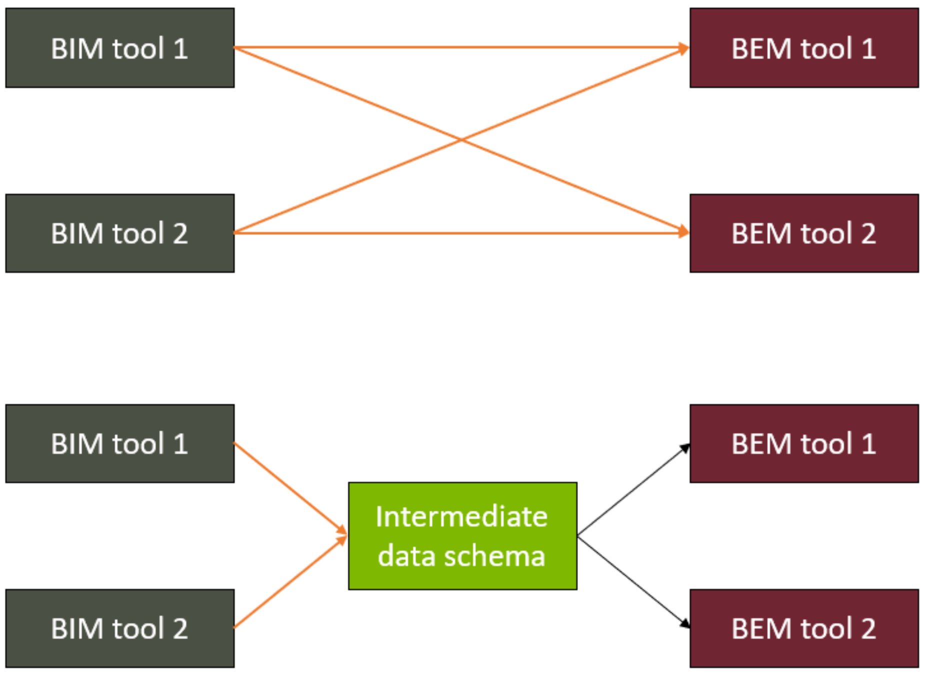


Figure 8, interaction between BIM and BEM tools (own illustration)

Interfaces directly in BIM tools are error-prone due to the high maintenance effort whenever an update to either the BIM or BEM tool is applied. Furthermore, BIM integrated interfaces must be developed for every single type of transformation from one single BIM tool to one single BEM tool.

An open data schema or standard used as an interface is an option to overcome these issues. Maintenance effort is reduced to the interaction of the respective tool to the intermediate data schema. However, too many different open data schemas for the same purpose can again increase the maintenance effort due to the requirement to update the interfaces from a BIM-authoring tool for all of its supported open standard schemas (Ramaji et al. 2020b).

Different data file schemas are currently available for the data conversion from BIM to BEM:

- IFC (buildingSMART International 2020c)
  IFC is a hierarchical data schema used for the exchange of almost any kind of information related to a BIM model across all disciplines throughout all life-cycle phases (Gourlis et al. 2017). With MVDs, it is possible to narrow down the amount of data exported from a BIM model to an IFC to a required degree.
- gbXML (Green Building XML Schema n.d.)
  Green Building XML Schema is written in extensible markup language. It was developed by Autodesk Green Building Studio as an exchange data format for the purpose of energy performance simulations. Issues with the conversion of geometry were reported by Gourlis et al. (2017) and Karlapudi et al. (2020). The main limitation of gbXML for BEM is the usage of the centre-line theory (O'Donnell et al. 2019). This means that it represents input building geometry with the centrelines of the objects instead of the actual space bounding boundaries. This leads to different space volumes and eventually varying and unreliable results (Bazjanac et al. 2016).
- Honeybee Schema (Ladybug Tools LLC)
  Honeybee Schema (HS) is a recently developed schema for interoperability between BIM authoring tools and the BEM interface toolkit Ladybug tools. Software development kits (SDKs) for BIM authoring or CAD tools, e.g., Revit, Rhino or Blender, are used to parse information to the Honeybee Schema. Ladybug toolkits are used as interfaces to BEM tools, e.g., Radiance for lighting simulation or OpenStudio and EnergyPlus for dynamic thermal performance simulation.
- SEMERGY Building Model (Pont 2014)
  SEMERGY Building Model (SBM) uses the structure of another data format, the Shared Object Model (SOM) from SEMPER (Mahdavi 1996), as a baseline (Ghiassi 2013; Pont 2014). Special semantic web technologies for database integration were coupled with existing building performance calculation methods. This enables the schema to overcome issues with incomplete, missing, or incorrect information and the resulting data acquisition. SBM is used to assess existing and newly built buildings

via the web-application SEMERGY (Xylem Technologies).It supports users in the decision making process with optimized solutions. (Fenz et al. 2016)

- Simulation Domain Model (O'Donnell et al. 2011)
  Simulation Domain Model (SimModel) is a combination of different building data standards. Amongst others, it covers functionalities of the IFC, gbXML and the EnergyPlus input data model (Digital Alchemy, Inc. 2021a; Nasyrov et al. 2014). Simergy, a GUI for the EnergyPlus simulation engine, is based on SimModel (Digital Alchemy, Inc. 2021b). Further research efforts that use the data scheme as an intermediate format have been developed in recent years. Giannakis et al. (2019b) used it as an intermediate data format for the information exchange between IFC and EnergyPlus.
- OpenStudio Model (Alliance for Sustainable Energy, LLC)
  OpenStudio Model (OSM) is used as a data model by OpenStudio, which is an open-source interface for a collection of BEM tools. For the dynamic energy simulations, EnergyPlus is used as an engine. OSM is a data schema that adds object-oriented software concepts to the basic EnergyPlus data model IDD. Relationships and inheritance of information can be used to simplify the process of creating complex models.

As mentioned previously, IFC is currently the only standardised and ISO certified open data format for information exchange in the BIM and BEM environment. It is currently offering promising possibilities to act as a baseline for the BIM-to-BEM data transfer (Ramaji et al. 2016). The focus on IFC as the main exchange format in the building industry could decrease the maintenance effort on both sides, BIM, and BEM, and streamline future development. Therefore, in the scope of this thesis, IFC-based BIM-based BEM workflows are to be further evaluated.

2.3.2 IFC and BEM

To fully understand how information from IFC could be retrieved for the purpose of BEM, it is crucial to understand the structure of the IFC data format in detail. An overview of the classes and their connection to each other is given in this section. As mentioned previously, IFC is a hierarchical data format. The data structure of IFC files is defined in the IFC schema. Without this schema, the IFC files would be plain text files. An excerpt of some relevant entities of the IFC data schema and their relationship is depicted in Figure 9. Physical or non-physical objects, such as walls, slabs, the project origin, or even the project itself, are defined in IfcObjectDefinition and its subclasses. Properties of objects are defined in so-called IfcPropertyDefinition objects.

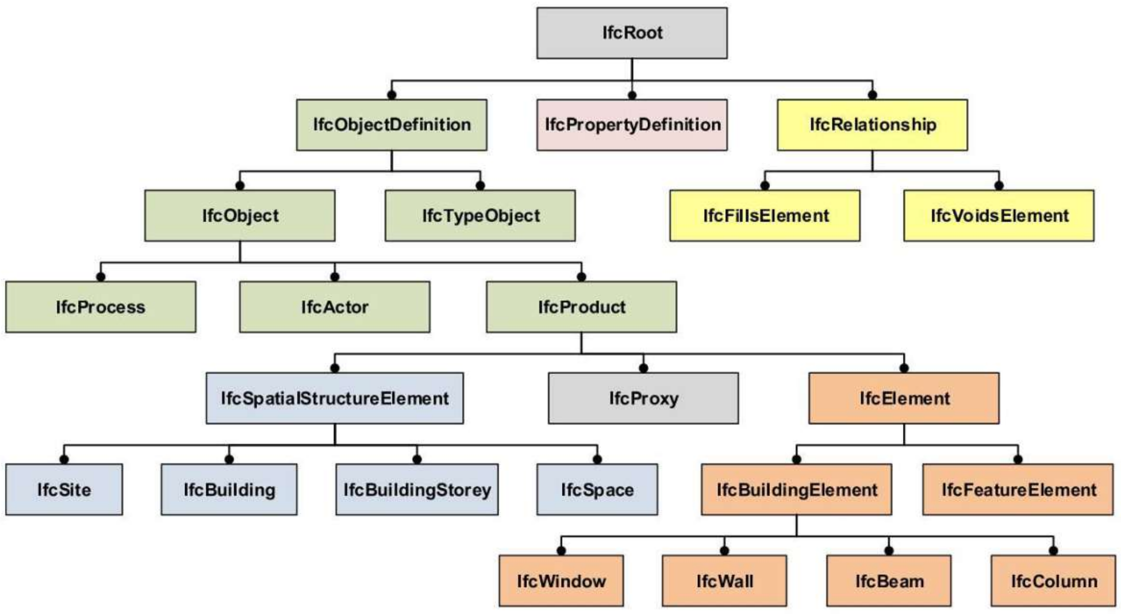


Figure 9: excerpt of entities of the IFC data model (Borrmann et al. 2018)

The subschema of IfcPropertyDefinition is shown in Figure 10. Via the different types of property definition levels, it is possible to assign properties to either single IFC entities or to a type or group of entities. The properties of these conglomerations can then be inherited to the included entities.

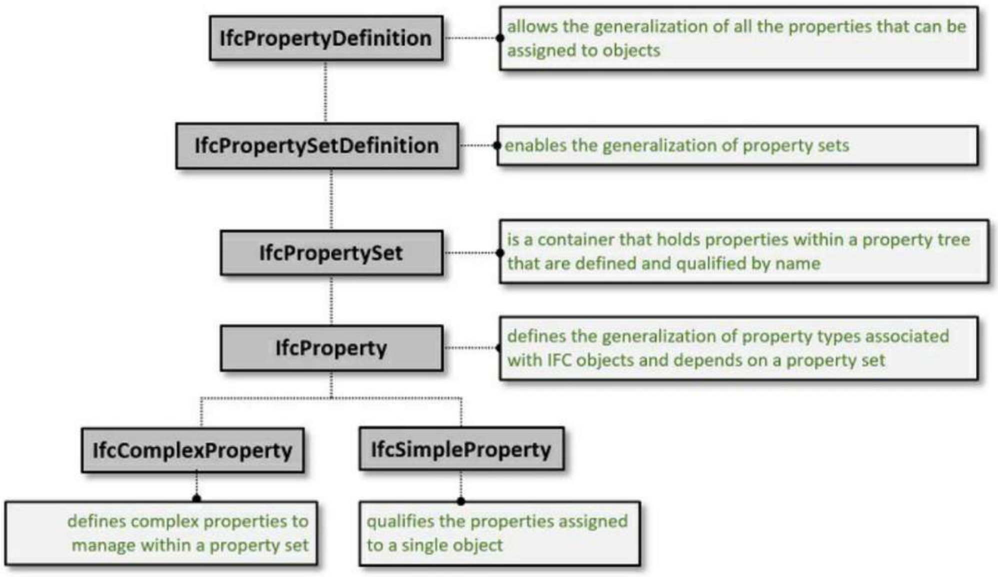

IfcPropertyDefinition
allows the generalization of all the properties that can be assigned to objects
IfcPropertySetDefinition
enables the generalization of property sets
IfcPropertySet
is a container that holds properties within a property tree that are defined and qualified by name
IfcProperty
defines the generalization of property types associated with IFC objects and depends on a property set
IfcComplexProperty
IfcSimpleProperty
defines complex properties to manage within a property set
qualifies the properties assigned to a single object

Figure 10: schema of IfcPropertyDefinition (ACCA software S.p.A. 2020a)

The Express language has limitations regarding direct connections of object entities. Therefore, so-called relations in IfcRelationship, that act as links between these objects, were defined (Ramaji et al. 2020a). The subschema of IfcRelationship is imaged in Figure 11.

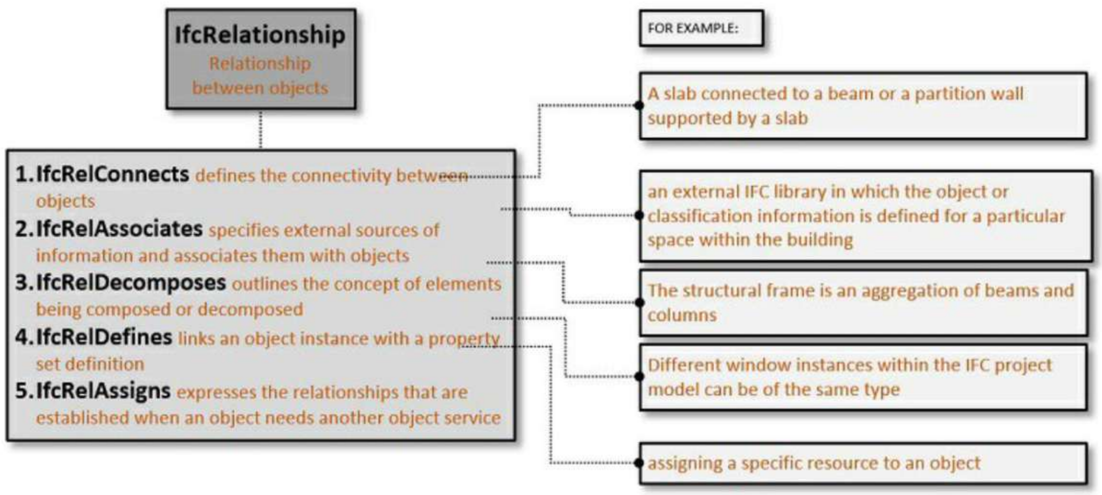

IfcRelationship
Relationship between objects
1. IfcRelConnects defines the connectivity between objects
2. IfcRelAssociates specifies external sources of information and associates them with objects
3. IfcRelDecomposes outlines the concept of elements being composed or decomposed
4. IfcRelDefines links an object instance with a property set definition
5. IfcRelAssigns expresses the relationships that are established when an object needs another object service
FOR EXAMPLE:
A slab connected to a beam or a partition wall supported by a slab
an external IFC library in which the object or classification information is defined for a particular space within the building
The structural frame is an aggregation of beams and columns
Different window instances within the IFC project model can be of the same type
assigning a specific resource to an object

Figure 11: schema of IfcRelationship (ACCA software S.p.A. 2020b)

Several requirements must be fulfilled to use the IFC data model in the BIM-based BEM transformation process. The information in IFC that is relevant for BEM is distributed over several IFC classes and objects. An overview of the required classes and their dependency is illustrated in Figure 12.

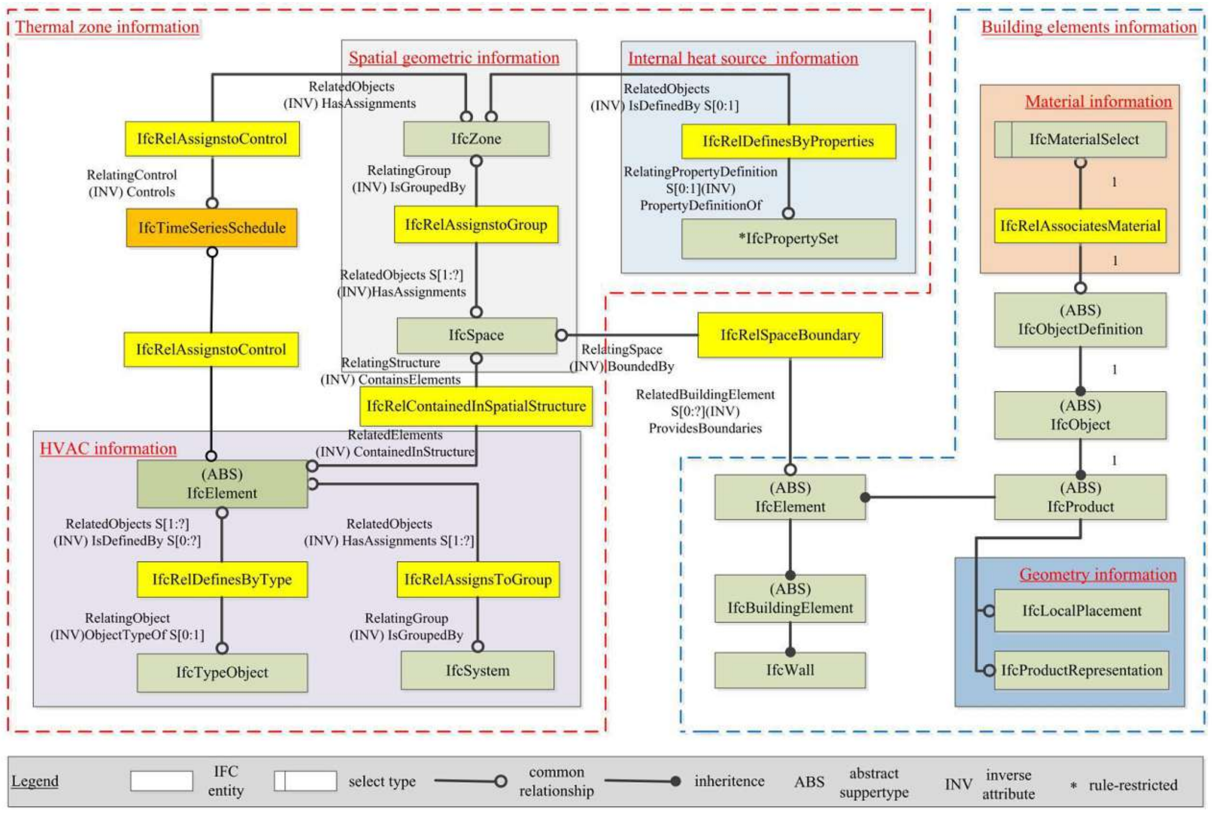

Thermal zone information
Spatial geometric information
Internal heat source information
Building elements information
Material information
RelatedObjects (INV) HasAssignments
IfcRelAssignstoControl
IfcZone
RelatedObjects (INV) IsDefinedBy S[0:1]
IfcRelDefinesByProperties
IfcMaterialSelect
RelatingControl (INV) Controls
RelatingGroup (INV) IsGroupedBy
RelatingPropertyDefinition S[0:1](INV) PropertyDefinitionOf
1
IfcTimeSeriesSchedule
IfcRelAssignstoGroup
*IfcPropertySet
IfcRelAssociatesMaterial
1
RelatedObjects S[1:?] (INV)HasAssignments
(ABS) IfcObjectDefinition
IfcRelAssignstoControl
IfcSpace
RelatingSpace (INV) BoundedBy
IfcRelSpaceBoundary
1
RelatingStructure (INV) ContainsElements
IfcRelContainedInSpatialStructure
RelatedBuildingElement S[0:?](INV) ProvidesBoundaries
(ABS) IfcObject
RelatedElements (INV) ContainedInStructure
HVAC information
(ABS) IfcElement
1
(ABS) IfcElement
(ABS) IfcProduct
RelatedObjects S[1:?] (INV) IsDefinedBy S[0:?]
RelatedObjects (INV) HasAssignments S[1:?]
IfcRelDefinesByType
IfcRelAssignsToGroup
Geometry information
(ABS) IfcBuildingElement
IfcLocalPlacement
RelatingObject (INV)ObjectTypeOf S[0:1]
RelatingGroup (INV) IsGroupedBy
IfcTypeObject
IfcSystem
IfcWall
IfcProductRepresentation
Legend
IFC entity
select type
common relationship
inheritence
ABS abstract suppertype
INV inverse attribute
* rule-restricted

Figure 12: IFC-based information model (Zhi-liang et al. 2012)

As previously described, the IFC data format is capable of containing vast amounts of information. For BEM most of this information is irrelevant. MVDs can create a subset of the complete IFC data format. This subset can be specified to be individually customised for the requirements of the BEM application. The research on MVDs for the purpose of BEM is ongoing. The MVDs that are currently most suitable are Coordination View Version 2.0 with Space Boundary Addon View for IFC2x3 and Annex60 for IFC4 (Pinheiro et al. 2018). However, none of the developed MVDs are yet state-of-the-art.

2.3.3 Space boundary surfaces

A decisive factor in the BIM-based BEM transformation process is the transformation of information related to the zoning in the building. In the scope of BEM, space boundary surfaces are often used to describe the relation of thermal zones to the enclosing geometry objects.

Commonly used BEM software solutions have one basic consideration: two- and three-dimensional transmission and flow are neglected and not included in any calculations. Therefore, all energy flow and transmission are perpendicular to the corresponding surface of the building element. It is necessary to have information concerning the polygonal surfaces through which energy is transferred. This includes all polygonal surfaces between internal spaces or zones and between internal spaces or zones and the environment. These surfaces are called space boundary surfaces. For each object between two internal spaces, an internal space boundary surface and an external space boundary surface exist. Exterior objects are represented by one space boundary that is related to the interior surface of these objects (Bazjanac 2010).

Issues with space boundary geometry were critical for BIM-based energy simulations. Therefore, guidelines were developed to standardise the import and export of such geometry by BIM authoring tools (Hitchcock et al. 2011).

In the IFC data model, different levels of space boundaries can be defined. According to Bazjanac (2010) five levels of space boundaries can be defined. In the scope of IFC this can be reduced to two levels with further differentiation. These are illustrated in Figure 13. 1st-level space boundaries are generated by the enclosing surfaces of a single space. The consideration of each space as a single thermal zone with customizable room conditions requires the implementation of 2nd-level space boundaries. A different material, or the addition of openings in the surface also require 2nd-level space boundaries. A space boundary surface of a single element is split into several space boundary surfaces to encompass these variations (Bazjanac 2010; Giannakis et al. 2019b; Hitchcock et al. 2011).

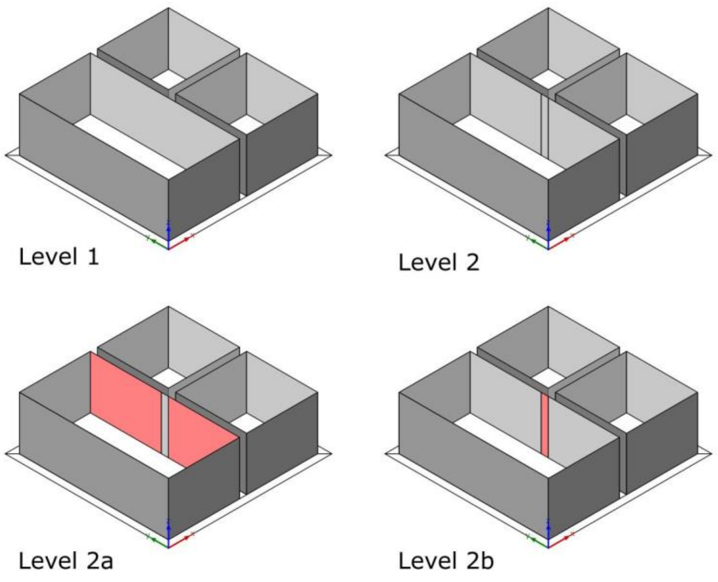


Figure 13: IFC space boundary levels (Borrmann et al. 2018) after (buildingSMART International 2020b)

For BEM the usage of 2nd-level space boundaries is required. Different approaches to create 2nd level space boundaries have been developed in recent years (Bazjanac 2010; Lilis et al. 2017). The algorithms for the generation of space boundaries are partly implemented in BIM authoring tool IFC export functionalities.

2.4 Workflows for IFC to IDF transformation

There is currently no possibility to import any BIM data exchange format directly into EnergyPlus. However, several third-party tools were developed to bridge this gap. These tools either convert the IFC directly to an IDF or first convert the IFC to a third-party file format, and subsequently, transform the third-party file format internally into an IDF and use EnergyPlus as the energy simulation engine. IFC has multiple ways to describe different objects. Therefore, the export of different BIM authoring tools is inconsistent, which poses a problem since the objects must be parsed to create an analytical model for the energy simulation.

The focus of the literature review of BIM-based BEM transformation workflows is laid on research efforts published within the last decade. Different search engines and databases, such as google scholar, science direct, and Research Gate, are used to find relevant research publications. The used keywords are BIM, BEM, IFC, EnergyPlus, Building (Energy) Performance Simulation, Thermal Simulation, and Information Exchange. Several of the found research efforts were also described in the comprehensive literature review by Gao et al. (2019). In this review, BIM-based BEM workflows with different BIM authoring tools, different exchange formats, IFC, gbXML, etc., and different simulation engines were reviewed. Especially IFC and gbXML based workflows and their current development progress are stressed. Based on this review and the additional found research efforts, an excerpt was chosen to be reviewed in the scope of this thesis.

With the application and evaluation of some of these workflows in the case study in mind, the selection of the workflows was made based on the potential, and feasibility. As previously described, EnergyPlus is chosen as the simulation engine. Therefore, only workflows that work with EnergyPlus as the simulation engine were reviewed in detail. It must be stated that the reviewed workflows are only an excerpt of all available workflows. However, because there are similarities between almost all workflows that transform an exchange data format into the EnergyPlus IDF data format, the listed workflows are still representative.

Some workflows that are not listed are either promising but still under development and not yet published, or the development was stopped years ago, and they were already evaluated in detail in existing studies.

However, several workflows for the transformation of information from IFC for the usage in other energy simulation engines are worth mentioning since this thesis also focuses on IFC based approaches. Ladenhauf et al. (2014) developed a workflow to transform data from IFC to XML for the calculation according to national and international standards in ArchiPHYSIK. In further research projects, BIMserver was also used as an intermediate software tool (Ladenhauf et al. 2016). Andriamamonjy et al. (2018) developed a workflow using Python scripts to parse information from IFC via IfcOpenShell to Modelica.

Some investigated workflows involve an intermediate data format. The advantages of an additional data format are (i) the ability to easily enrich the input file with additional information required for the simulation and (ii) the utility of the file for different purposes, such as lighting simulations.

The reviewed workflows use different data formats for the information exchange. An overview of the workflows and their properties is depicted in Table 1. The listed workflows are described in detail in the following sections.

Table 1, overview of the reviewed transformation workflows with EnergyPlus file as output

Name	Author	Exchange format	Availability
Python and IfcOpenShell	Patel (2020)	IFC2x3	+
OsmSerializer	Ramaji et al. (2020b)	IFC2x3	+
CBIP via SimModel	Giannakis et al. (2019b)	IFC4	-
OpenBIM-based energy analysis software	Choi et al. (2016)	IFC	-
BlenderBIM based on IfcOpenShell and Ladybug	Moult et al. (2020)	HS	-
Space Boundary Tool (SBT-1)	Rose (2012)	IFC2x3	+
SEMERGY	Fenz et al. (2016)	SBM	-

Legend: available (+), not available (-)

2.4.1 Python and IfcOpenShell

In recent years, the interest in an open BIM environment has increased rapidly. Tools, such as IfcOpenShell (Krijnen et al. 2020), were developed as an independent counterpart to the proprietary software solutions. IfcOpenShell is an open-source software library and toolkit for processing of the IFC file format (Krijnen 2019). Information in IFC 2x3 and IFC4 schemas can be accessed. IfcOpenShell uses Open CASCADE to convert the geometry of the IFC files to a representation of the geometry that can be displayed in any CAD software.

There are different possibilities to read and write the IFC format through IfcOpenShell. One option is through the Python module IfcOpenShell-python (Krijnen et al. 2020). Python is “an interpreted, object-oriented, high-level programming language with dynamic semantics” (Python Software Foundation 2021).

In her Master Thesis (2020), Patel developed different Python scripts using the Python library IfcOpenShell-python to transform an IFC file created in ArchiCAD into an IDF. The scripts cover the transformation of geometry, constructions, material properties and schedules. The characteristics of this approach are described in this chapter.

Software environment used by Patel (2020):

- OpenStudio 2.9.1
- EnergyPlus Version 9.2
- ArchiCAD with additional ArchiCAD IFC properties
- IFC2x3, Coordination View 2.0, 2nd level space boundaries
- Python 3

The MVD Coordination View is mainly used for information exchange between the disciplines of architecture, structural engineering, and mechanical engineering. Information concerning space boundaries, building geometry and technical systems is preserved from the source BIM model (O'Donnell et al. 2019).

The workflow with the interacting software environments is illustrated in Figure 14. It has the following steps:

1. Create a building model in ArchiCAD
2. Customize the IFC mapping properties in ArchiCAD
3. Export the IFC file
4. Transform the IFC file with Python scripts into an IDF
5. Run EnergyPlus with the resulting IDF

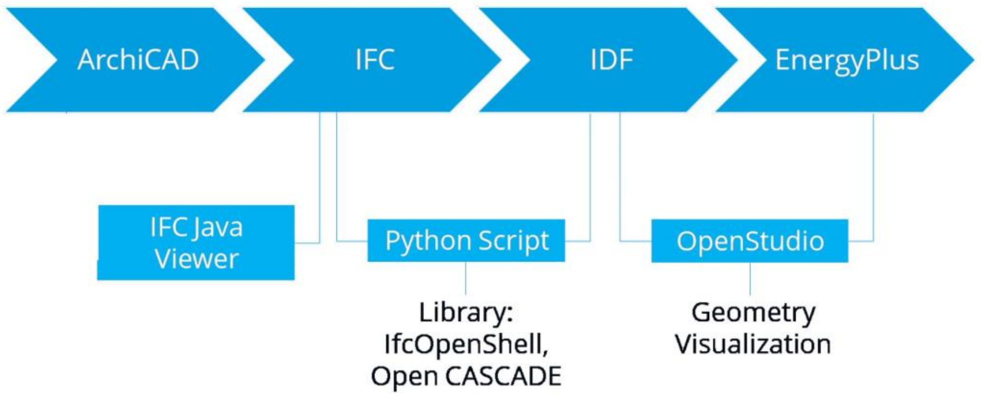


Figure 14, overview of workflow with Python scripts (Patel 2020)

Some special requirements were defined to apply the workflow successfully. It is important to draw the walls in ArchiCAD counter-clockwise. Otherwise, the geometry transformation is not working correctly. The dimensions of the slabs, ceilings and roofs are defined manually in the IFC property settings of ArchiCAD. The reference line for horizontal building elements, slabs, ceilings, and roofs, is the bottom of the building element. The reference line for basement floors is the top of the element. Separate slabs are modelled for each zone. For each ArchiCAD room, a separate zone is automatically created. The area of the zones in the IDF is not defined by the geometry dimensions but set to the value excerpted from the IFC file. The room geometry is defined as the net volume, enclosed by the inner surfaces of the boundary objects. Shading elements are created as ArchiCAD slabs. The ArchiCAD north angle is rotated by -90° from the EnergyPlus north angle. Therefore, the true north is set to 90° (Patel 2020).

2.4.2 OsmSerializer

Ramaji et al. (2020b) developed a workflow that transforms IFC data files via a serializer integrated into BIMserver, which is an open source IFC model server (Beetz et al. 2010).

There are two possibilities to operate the transformation. The first option is directly via the BIMserver by a web page or via code-based input. The second option is via OpenStudio and an interface for the BIMserver, which is described in the scope of this thesis.

The software environment of the workflow is the following:

- BIMserver
- Java v1.7
- OsmSerializer 1.8
- OpenStudio 2.0
- EnergyPlus version 8.6.0
- Revit with a customized IFC exporter
- Custom MVD based on the U.S. General Services Administration (GSA) named GSA-05 with additional property sets in the same data structure.

In a summary, the workflow has the following steps:

1. Create a building model in Revit
2. Export the model from Revit with a custom IFC exporter
3. Import the IFC in the OpenStudio add-on which acts as an interface for the BIMserver
4. Upload the model to the BIMserver
5. Transform the model via the BIMserver extension OsmSerializer
6. Export the resulting OSM model from BIMserver to OpenStudio
7. Enrich the model and run a simulation with EnergyPlus, which creates an IDF

The workflow is depicted in Figure 15.

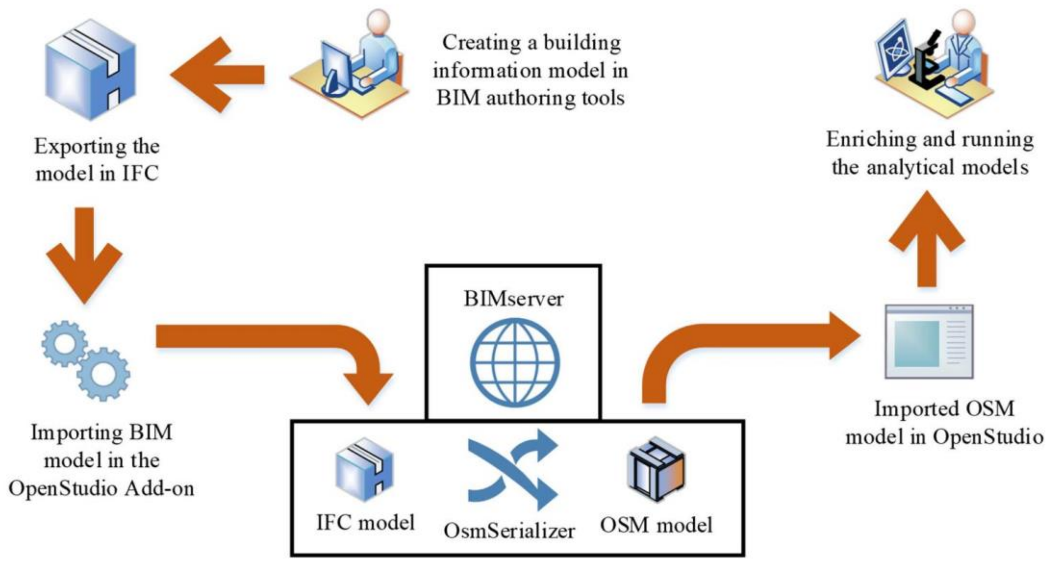

Creating a building information model in BIM authoring tools
Exporting the model in IFC
Enriching and running the analytical models
BIMserver
Importing BIM model in the OpenStudio Add-on
Imported OSM model in OpenStudio
IFC model
OsmSerializer
OSM model

Figure 15: OsmSerializer workflow (Ramaji et al. 2020b)

The general data transformation process through the BIMserver is shown in Figure 16.

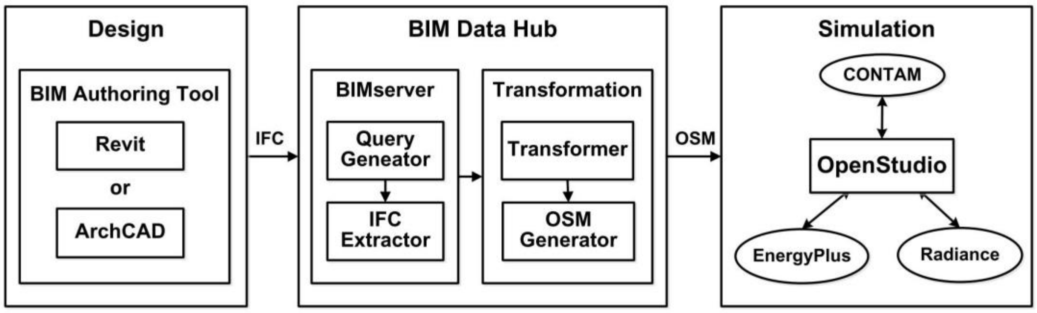


Figure 16: Data transformation through the BIMserver (Yu 2014)

The serializer uses the IfcOpenShell library for geometry transformation. Rooms with the related boundaries are converted to spaces. The assignment to thermal zones is however not automated. Manual assignment of thermal zones is therefore necessary in the post-processing of the IDF.

2.4.3 CBIP via SimModel

Common Boundary Intersection Projection (CBIP) via SimModel has been developed as an alternative to the existing space boundary creation algorithms in the recent years. CBIP calculates 2nd-level Space boundary surfaces. The input for this operation is the geometrical information of the architectural elements. As an output, applicable IFC data classes are added to the IFC file (Giannakis et al. 2019b; Lilis et al. 2017).

IFC4 with the MVD Design Transfer View is used for this workflow. As a BIM authoring tool Revit 2018 is used and the export to IFC is performed via a customised version of the Revit IFC exporter. The unique feature of this workflow is the transformation of curved elements with the CBIP algorithm. 2nd level space boundaries are automatically created from the IFC file, even if this information is not exported from the BIM authoring tool. Standardised enrichment of the resulting model is possible via intermediate transformation to SimModel (Giannakis et al. 2019b; Lilis et al. 2019).

In a summary, the workflow has the following steps:

1. Creating a model in Revit
2. Export the model to IFC with the customised Revit exporter for IFC
3. Creation of the space boundaries in IFC with CBIP
4. Transformation of the IFC to SimModel XML
5. Enrichment of the SimModel XML with additional information
6. Transformation of SimModel XML to EnergyPlus IDF

In the scope of this thesis, it was not possible to get access to the software tools and the related algorithms.

2.4.4 OBES

OpenBIM-based energy analysis software (OBES) is a tool that enables users to standardise the transformation process of IFC to the IDF data format. In OBES different inputs are put together to generate a dataset that be exported as an IDF file. The transformation of the material properties is performed via an extensive library. As a result, an IDF is exported from OBES and used as an input for EnergyPlus (Choi et al. 2016). In the scope of this thesis, it was not possible to get access to the software tools.

2.4.5 BlenderBIM based on IfcOpenShell and Ladybug

In 2015, Ladybug Tools LCC released their tools to make building energy modelling easier by simplifying cumbersome simulation setups. The tools are used as programming scripts that are accessible for the common user via visual scripting interfaces such as Grasshopper for Rhino or Dynamo for Revit. In the last years, these tools were re-written to be independent of BIM authoring tools. As a result, a new data schema, Honeybee Schema, was developed. Interfaces for BIM authoring tools to this schema are under ongoing development (Sobon, 2020). One promising approach for the IFC data schema is the implementation of ladybug-blender. Blender is an open-source BIM authoring tool that also supports the implementation of IfcOpenShell via the BlenderBIM add-on. Currently, the interface to EnergyPlus, Honeybee, has not yet been implemented (Moult et al. 2020). Therefore, this workflow is not further evaluated in the scope of this thesis.

2.4.6 Space Boundary Tool

Space Boundary Tool (SBT-1) was developed by Rose (2012) and it supports only IFC2x3. The development of the tool was suspended as of August 2014 (U.S. Department of Energy 2019). This approach was already evaluated on a case study by Ivanova et al. (2015) and the mapping of materials by Shadrina (2015). Therefore, this workflow is not further evaluated in the scope of the thesis.

2.4.7 SEMERGY

SEMERGY is a web-based optimization tool for the assessment of thermal building performance. Databases for materials are created by retrieving information from different sources, such as databases or manufacturer information. Additional properties concerning interoperability of these materials with each other are implemented. Requirements according to national standards and legal documents are taken into account by the workflow. Investment costs and environmental impact are also considered. The main advantage of the approach is the combination of building energy modelling with a structured way of gathering and describing information, so-called ontologies (Fenz et al. 2016; Pont et al. 2014; Pont et al. 2015).

3 METHODOLOGY

3.1 Overview

A case study with a BIM model is conducted to evaluate BIM-based BEM transformation workflows. Therefore, the BIM model is exported as an IFC file from the proprietary software tools Revit and ArchiCAD. These IFC files are afterwards transformed with available IFC-based BIM-to-BEM tools and workflows. The result of the workflow must be an EnergyPlus Input File (IDF) for further processing in the dynamic simulation software EnergyPlus. As a reference case, a manually created IDF is used to compare the IDF and the simulation results to the different IFC-based IDFs and their results.

3.2 Hypothesis

Automated BIM-based BEM workflows in the field of dynamic thermal building simulation are a yet to be achieved goal. In the frame of this thesis different workflows for the transformation of an IFC data file into an EnergyPlus Data File (IDF) are presented. Afterwards, the presented workflows are tested on a detailed building model. The results of the workflows are evaluated regarding their correctness, degree of automation and the possibility to set them up.

3.2.1 Research Question 1

Which IFC-based BIM-based BEM workflows are currently available and usable for semi-automatic IDF generation with the focus on information related to geometry and building objects for thermal building performance simulations? Which requirements and limitations exist in the scope of the complete process?

Method:
(i) Definition of the required information (geometry, material, constructions) to be integrated into the workflows. (ii) Evaluation of available workflows based on the specified indicators. (iii) Selection of suitable workflows. (iv) Evaluation of the chosen workflows in the steps of a successful transformation process.

3.2.2 Research Question 2

How do the previously chosen workflows perform on a detailed case study?

Method:
(i) Creation of a detailed building in BIM authoring tools. (ii) Export of the BIM model to IFC. (iii) Transformation of the IFC with the chosen workflows. (iv) Review of transformation results concerning consistency and correctness. (v) Evaluation of measurements for manual pre- and

postprocessing. (vi) Comparison of intermediate and simulation results with each other and a manually created baseline model.

3.3 Case Study

The detailed building model for the evaluation in the scope of the case study is chosen based on the desired transformation evaluation process. An artificial building is designed to include unique geometric and semantic properties. The base cubature of the building has three levels and a rectangular shape. There is a basement, a ground floor and a first floor. The building is shown in Figure 17.

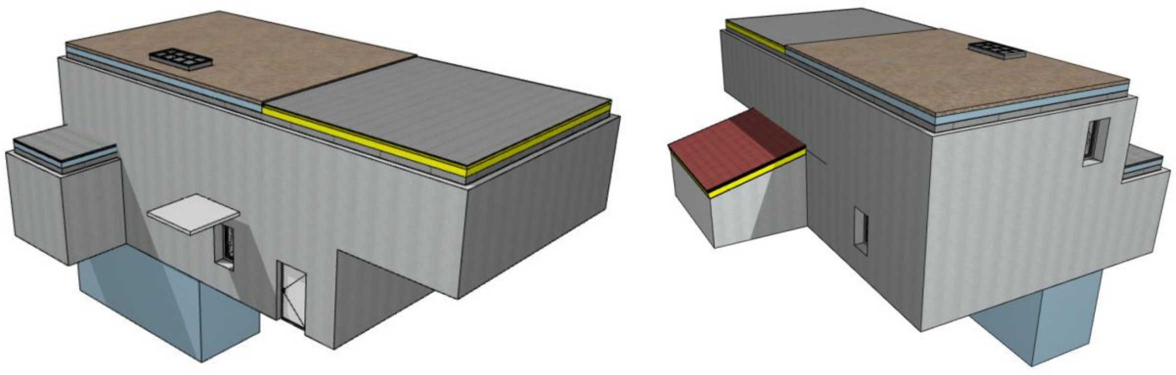


Figure 17, 3D views of the case study building in ArchiCAD

For the purpose of testing different boundary surface properties, some customizations are made to the building. The ground floor plan of the building is depicted in Figure 18. The basement boundary is adapted so that the ground floor is not completely with an underground level. Furthermore, the ground floor boundary is adapted. Two rooms without space above it are added. One with a flat roof without slope (part of Raum1x01), one with a sloped roof (Raum1x06). The interior space distribution is chosen to have rectangular shaped rooms over one level. One atrium (Raum 1x04) over two levels is created by merging two rooms, one from the ground floor, one from the first floor.

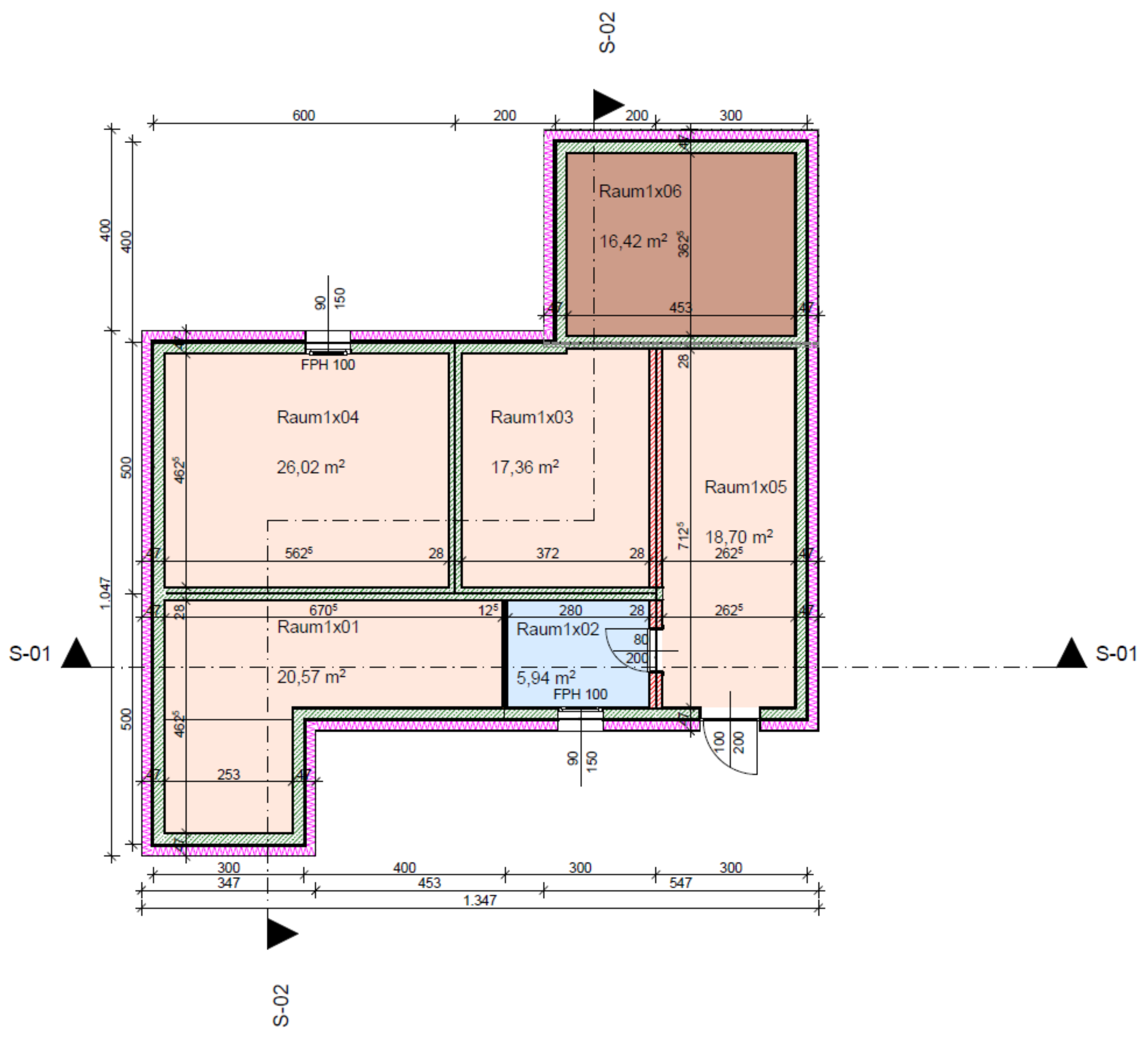


Figure 18, ground floor plan in ArchiCAD

It is assumed that each level is one separate thermal zone. The atrium (Raum1x04) is belonging to the ground floor. The constructions of the building objects are listed in detail in the appendix.

3.4 BIM-to-BEM Transformation

In the scope of this thesis, two workflows for BIM-based BEM that are using the IFC format are evaluated. The chosen workflows are “Python and IfcOpenShell” and “OsmSerializer”, described in section 2.4. Both are free-to-use, publicly available and have promising approaches that could be enhanced further in future research efforts.

The different tools used in the two transformation processes are illustrated in Figure 19. Before running simulations with EnergyPlus is possible, several requirements need to be met and various steps need to be performed before an IFC file is ready to be transformed and postprocessed after the transformation. The required steps are described in detail in the following sections.

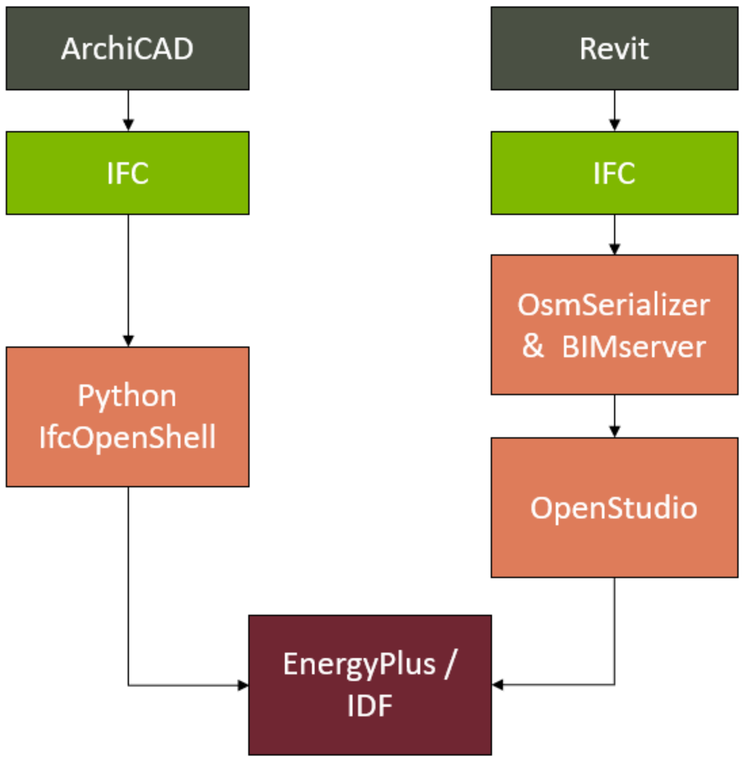


Figure 19, flow chart of the two evaluated workflows (own illustration)

3.4.1 Creation of BIM for later BEM transition

The purpose of the model should be clearly determined before a BIM model is created. If the BIM model is designed for later use for dynamic energy simulations, certain requirements and guidelines for the model generation must be met. Standards for verifying and ensuring the quality of BIM models for the later use for BEM need to be defined to increase the reliability and trustworthiness of BEM results (Nasyrov et al. 2014; O'Donnell et al. 2019).

Examples of guidelines for creating BIM models with the purpose of export and further usage in a transformation process for building energy modelling were developed by Maile et al. (2013), Andriamamonjy et al. (2018) and Giannakis et al. (2019a).

Without an input model that fulfils certain standardised requirements issues and resulting inaccuracies in the further processing of these models will also occur. The creation of objects without overlapping areas is required, and architectural spaces must be assigned to thermal zones. Building information, such as construction and material properties, must be assigned to the relating objects in a clear and standardised way (El Asmi et al. 2015).

It is also important to define standards for representing objects in BIM models in certain design phases. The LOD of the BIM model partly covers this, but further specifications are needed.

Even with specified LOD requirements of a BIM model there might be different interpretations by the project participants. This leads to varying levels of detail in the design and different availabilities of required object properties. Defined standards to describe the modelling of BIM models in different project phases for the later usage in a BEM workflow are necessary (Andriamamonjy et al. 2018). Simplification tools based on these standards can even transform complex models to a processable stage for BIM-based BEM transformation workflows (Ladenhauf et al. 2014).

3.4.2 BIM Export to IFC data format

The building designed in a proprietary software tool in the previous step is exported as an IFC file in the next step, which is one of the most crucial steps in all transformation workflows. The IFC data format can include a lot of information from the designed building, described in section 2.1. For the further workflow, including the dynamic simulation, certain information is required, and a lot of the possible information is unnecessary. Via an MVD, described in section 2.1, it is possible to choose a subset of the available data. The best-suited MVD varies from workflow to workflow and is dependent on the desired IFC format. As described in section 2.3.2, no state-of-the-art MVD is currently available that covers all the information required for the kind of workflows evaluated in the scope of this thesis. Data available in the BIM authoring tool that is lost during the export process can be added manually to the IFC file later on. Due to the different property definitions in the BIM authoring tools, this can be cumbersome. For each workflow, it is essential to choose the right IFC version in combination with the correct MVD.

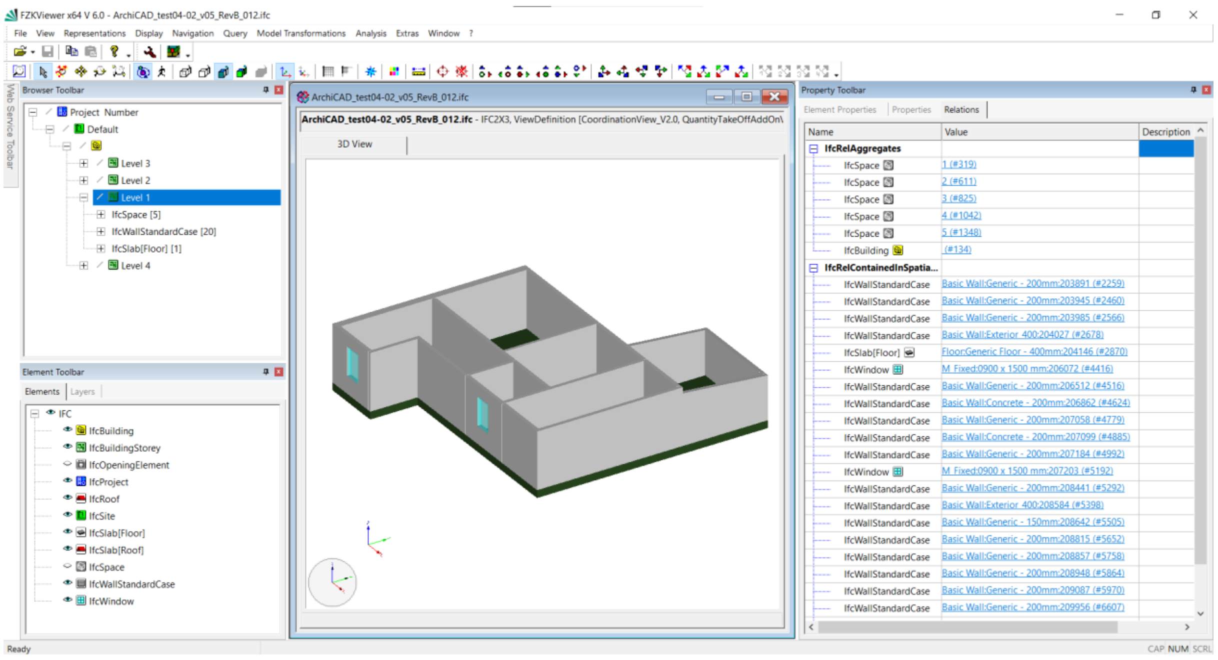
Attempts for the development and standardisation of geometry export were conducted over recent years. Tools such as Revit IFC exporter are under continuous development (Autodesk Inc. 2020). However, there are still issues due to different data management systems in proprietary BIM authoring tools and the lack of easy customizable IFC-exporters.

3.4.3 Compare data with MVD

MVDs ensure that specific data information is included in the exported IFC file. However, there is no automatic check if the exported data is correct. Several data checking and validation tools are currently available. The most widely used tool for this is IfcDoc (buildingSMART International 2021c), which can be used to create custom MVDs or extend existing MVDs with additional requirements. It is possible to validate exported IFC files based on the determined definitions.

3.4.4 Check IFC for consistency

Even if the IFC file is in accordance with the chosen MVD, information can still be missing. Therefore, the IFC should be checked in a viewer to evaluate especially space volumes and closed boundary objects (Patel 2020). Currently, many different tools are available to view IFC files. However, not all tools have the same possibility to evaluate IFC files in as much details as required for the BIM-based BEM workflows. In the scope of this thesis the FZKViewer (Karlsruher Institute of Technology, Institute for Automation and Applied Informatics 2020) is used to check the created IFC files for consistency. The GUI of the FZK Viewer is shown in Figure 20. On the left is the hierarchy tree with all included objects. In the middle is the 3D model. On the right is the property viewer with the possibility to inspect properties of an object and the relations between different objects.



ArchiCAD_test04-02_v05_RevB_012.ifc - IFC2X3, ViewDefinition [CoordinationView_V2.0, QuantityTakeOffAddOn\

Name	Value	Description
IfcRelAggregates		
IfcSpace	1 (#319)	
IfcSpace	2 (#611)	
IfcSpace	3 (#825)	
IfcSpace	4 (#1042)	
IfcSpace	5 (#1348)	
IfcBuilding	(#134)	
IfcRelContainedInSpatia...		
IfcWallStandardCase	Basic Wall:Generic - 200mm:203891 (#2259)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:203945 (#2460)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:203985 (#2566)	
IfcWallStandardCase	Basic Wall:Exterior 400:204027 (#2678)	
IfcSlab[Floor]	Floor:Generic Floor - 400mm:204146 (#2870)	
IfcWindow	M_Fixed:0900 x 1500 mm:206072 (#4416)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:206512 (#4516)	
IfcWallStandardCase	Basic Wall:Concrete - 200mm:206862 (#4624)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:207058 (#4779)	
IfcWallStandardCase	Basic Wall:Concrete - 200mm:207099 (#4885)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:207184 (#4992)	
IfcWindow	M_Fixed:0900 x 1500 mm:207203 (#5192)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:208441 (#5292)	
IfcWallStandardCase	Basic Wall:Exterior 400:208584 (#5398)	
IfcWallStandardCase	Basic Wall:Generic - 150mm:208642 (#5505)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:208815 (#5652)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:208857 (#5758)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:208948 (#5864)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:209087 (#5970)	
IfcWallStandardCase	Basic Wall:Generic - 200mm:209956 (#6607)	

Figure 20, GUI of FZKViewer for IFC model inspection

Some important free-to-use tools for in-depth analysis of IFC files are IfcCheckingTool (Karlsruher Institute of Technology, Institute for Automation and Applied Informatics 2021), IFC File Analyzer Software (Lipman 2017) and IfcValidator (de Laat et al. 2019). In the scope of this thesis these three tools are not used.

3.4.5 Mapping of entities

BIM to IFC

In the scope of this thesis, ArchiCAD and Revit are chosen to be the evaluated BIM authoring tools. According to Gao et al. (2019), these two tools are the widely used tools for IFC-based BIM-to-BEM transformation workflows. The handling of the required material properties in

ArchiCAD and Revit is compared in Table 2. In general, a lot of the information is either available in the pre-defined environment of the BIM authoring tools, or it is possible to create custom properties. Information can be added to these custom properties. Custom properties might be sufficient for a closed BIM environment because all project participants have aligned software tool settings. However, it is cumbersome to work in an OpenBIM environment with custom properties that are not defined according to specific standards. There must be rules on how these properties are defined in the BIM authoring software and exported to a data exchange format such as IFC.

The most crucial thermodynamic properties for thermal building performance simulation with BEM tools, thickness, conductivity, density, and specific heat capacity are standard material properties in ArchiCAD and Revit. More sophisticated properties, mainly concerning the surface, are not defined by default. The export of the available information is defined by the interface of the BIM authoring tool to IFC.

Table 2, availability of the required material properties in the BIM authoring tools and IFC

	ArchiCAD		**Revit**	
Property name [units]	BIM	IFC	BIM	IFC
Name [string]	+	+	+	+
Roughness [string]	-	-	-	-
Thickness [m]	+	+	+	+
Conductivity [W.m^{-2}.K^{-1}]	+	+	+	-
Density [kg.m^{-3}]	+	+	+	-
Specific heat (specific heat capacity in ArchiCAD) [J.kg^{-1}.K^{-1}]	+	+	+	-
Thermal absorptance [-]	-	-	-	-
Solar absorptance [-]	-	-	-	-
Visible absorptance [-]	-	-	-	-

Legend: available (+), not available (-)

ArchiCAD

ArchiCAD IFC exporter offers extensive possibilities to use and modify pre-defined MVDs. By customizing the settings for data conversion, it is possible to export material properties. The available information of the used material layers is therefore also available in the IFC file. The IFC export settings used in ArchiCAD are depicted in the appendix.

Revit

According to Nasyrov et al. (2014), Revit fails to export these material properties to IFC and gbXML, which is an issue related to the transformation of the built-in Revit schema to the exchange data formats.

Autodesk developed in cooperation with several contributors a customized IFC exporter (Autodesk Inc. 2020) that can replace the built-in exporter. This improved IFC exporter has several possibilities to export the model with different pre-defined or customized MVDs. It is furthermore possible to specify the objects to be exported.

Constructions are exported with their material layers and information of the physical properties of the composed construction. However, the material layers are only exported with their name and thickness. The export of further material layer properties, such as thermal conductivity or specific heat, is not possible with this IFC exporter. Customization of the mapping tables of the available IFC exporter does not solve this issue because materials are handled as special objects that are not just attached as properties to the material layers in Revit (Shadrina 2015).

Nevertheless, this issue can be fixed, e.g., with Dynamo, a visual scripting environment for Revit, which accesses objects through the Revit Application Programming Interface (API). Alternatively, a custom made IFC exporter can be used. However, these solutions require extensive knowledge of Revit API (Ramaji et al. 2020b).

Revit construction objects have the properties “roughness” and “absorptance”. However, there is no clear definition of these properties and, therefore no possibility to map them to the required properties in the IDF data schema (Autodesk Inc. 2021).

IFC to IDF

To fully understand the transformation workflows, it is important to check the mapping of relevant objects in detail. Based on this mapping, it is possible to create an overview of the information retrieved from the IFC file. The mapping of the required IDF class objects from the IFC file for the two chosen workflows is summarised in Table 3. The OsmSerializer workflow uses the OSM as an intermediate data file schema.

Table 3, mapping between IDD/IDF, IFC and OSM (Patel 2020; Ramaji et al. 2020b)

	Python and IfcOpenShell	**OsmSerializer**	
IDF Class Object	**IFC Entity**	**IFC Entity**	**OS entity**
Building	IfcProject	-	-
Site:Location	IfcSite	-	-
Material	IfcRelDefinesByProperties of IfcBuildingElement of IfcWall and IfcSlab	IfcMaterial with related IfcPropertySets	OS:Material
WindowMaterial	-	IfcPropertySets of IfcWindows	OS:WindowMaterial
Construction	IfcRelAssociatesMaterial of IfcBuildingElement of IfcWall and IfcSlab	IfcMaterialLayer Set	OS:Construction
Zone	IfcSpace	IfcSpace	OS:Space with OS:BuildingStorey > OS:ThermalZone
BuildingSurface:Detailed	IfcWall, IfcSlab	IfcWall, IfcSlab, IfcRoof	OS:Surfaces
FenestrationSurface:Detailed	IfcDoor, IfcWindow	IfcOpeningElem ent, IfcDoor, IfcWindow	OS:SubSurface
Shading:Building	IfcSlab	-	-

3.4.6 Addition of missing properties

Required properties missing in the created IDF data model, such as material properties, can be added before further processing. By mapping the existing objects to a construction and material library, the properties can be added in a structured manner (Hitchcock et al. 2011). Another possibility is to add the missing properties manually.

3.4.7 Conversion of IDF to UTF-8

As mentioned in section 2.1, IFC is encoded in American National Standards Institute (ANSI) language format (Shadrina 2015). In the transformation process, the encoding is preserved. However, EnergyPlus is not able to read ANSI encoded files successfully. Therefore, the resulting IDF might include text strings with characters that are not readable by EnergyPlus. One encoding that does not include special characters is the 8-Bit Universal Coded Character Set Transformation Format (UTF-8). If EnergyPlus classes include objects strings with, e.g., language-wise specific special characters, these must be replaced with the synonym characters compliant with UTF-8 without additional encoding symbols. The matching applied in the scope of this thesis is listed in Table 4. Even though this manual work might appear cumbersome. However, it is possible to automate the transformation process with a programming script.

Table 4, matching of special characters that are language-wise specific to UTF-8 compliant characters

language-wise specific special characters	UTF-8 compliant
ä	ae
ö	oe
ü	ue
ß	ss

3.5 Output check

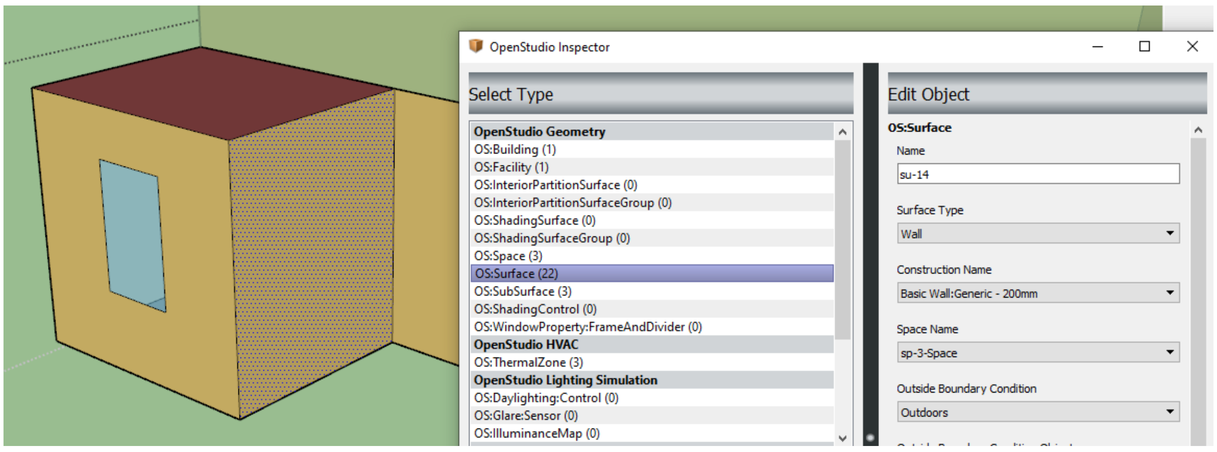
3.5.1 Compare IDFs

A first comparison of the resulting IDFs to each other is made with Notepad++ (Ho 2021) using the compare add-on. Deviating text lines are highlighted.

As a second step, the visual comparison of the different IDFs is made by importing the resulting Drawing Interchange Format for AutoCAD (DXF) files of EnergyPlus into one CAD drawing format for AutoCAD (DWG) file. A visual comparison of the imported models is afterwards possible.

3.5.2 IDF Validation

EnergyPlus lacks a built-in graphical user interface. Therefore, the geometry and corresponding semantic data of the resulting IDFs is validated with SketchUp and the OpenStudio plug-in. The detailed information of each surface is available through the inspector tool of the OpenStudio plug-in. An example of an exterior wall surface is shown in Figure 21.



OS:Surface
Name: su-14
Surface Type: Wall
Construction Name: Basic Wall:Generic - 200mm
Space Name: sp-3-Space
Outside Boundary Condition: Outdoors

Figure 21, validation of surface properties with OpenStudio inspector (own illustration)

The resulting IDFs are afterwards checked with the built-in EnergyPlus IDF editor. Critical issues of the defined classes, such as missing references, are highlighted by the software tool. Obvious mistakes in the IDFs are fixable without much effort.

3.5.3 Manual definition of additional required IDF classes

To run a dynamic simulation with EnergyPlus successfully, some minimum information requirements must be met. The information concerning the building geometry can be retrieved from the IFC file. Information for the HVAC system and additional simulation options need to be defined manually. This information is usually defined based on experience. Solutions to overcome this transformation gap with the possibility of standardisation of the input must be developed. On the other hand, definition of standardised result representation is specified by certain options in EnergyPlus. A simplified HVAC system for evaluation of the heating and cooling load is added to the IDF files. The EnergyPlus system IdealLoadsAirSystem is used for this purpose. The simulation control classes required for a successful simulation are added manually. The detailed information used in the scope of this thesis is listed in the appendix.

Output classes are added to evaluate the results. The geometry is evaluated by the resulting DXF file, which is a simple geometric representation generated by EnergyPlus. The simulation results are available as a Hypertext Markup Language (HTML) file for a comprehensive overview and if required as comma-separated values (CSV) file for detailed analysis.

3.6 Base energy model

As a reference case, the case study building is manually created in SketchUp and afterwards transformed to an IDF file via the OpenStudio SketchUp plugin. The further processing of the IDF is done in the built-in EnergyPlus IDF Editor and in a plain text editor.

First, the geometry of the building is created. The floor plan of the ground floor is drawn and afterwards with the OpenStudio plug-in extruded to the desired two above ground levels. The basement level is created manually. The previously created base building is afterwards modified on each level to match the plans. Second, openings are drawn and with the OpenStudio plug-in automatically matched to the corresponding surfaces. Third, building constructions are assigned to the resulting building surfaces. Fourth, the created spaces are assigned to thermal zones. The resulting model is shown in Figure 22.

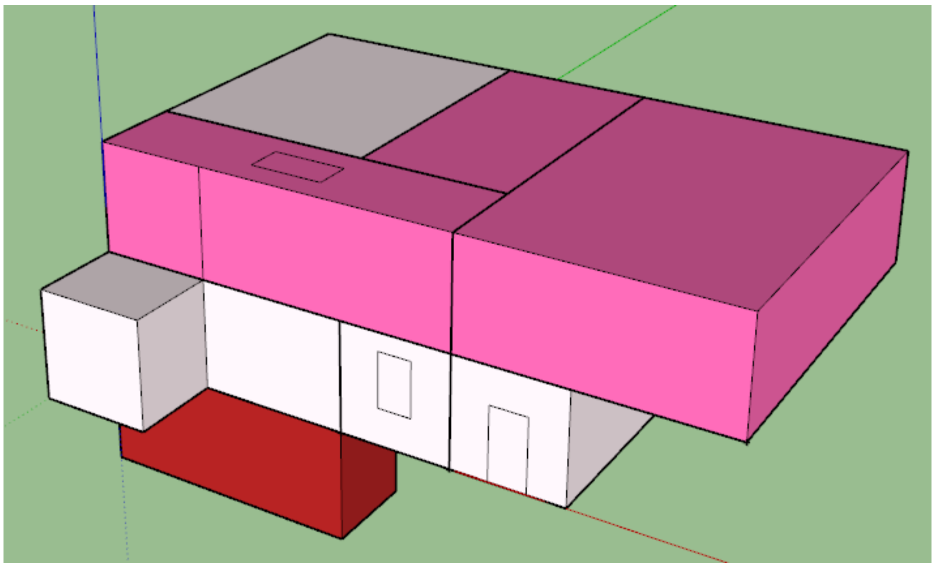


Figure 22, manually created building model in SketchUp; visualization of the three different thermal zones (own illustration)

The model is via the OpenStudio plugin exported as an IDF. The following IDF class objects for the building geometry information are retrieved from the model created in SketchUp:

- Building
- Site:Location
- GlobalGeometryRules
- Zone
- ZoneList
- BuildingSurface:Detailed
- FenestrationSurface:Detailed
- InternalMass

In further processing is done in a plain text editor. The pre-defined constructions and materials are imported into the IDF file. The imported constructions are afterwards assigned to the surfaces and the classes for the HVAC system are added to the IDF and the system is assigned to the thermal zones. Finally, the pre-defined simulation classes are added to the IDF.

4 RESULTS AND DISCUSSION

4.1 Overview

The evaluation results of the two workflows are summarised in Table 5 and are reviewed in detail in the following sections. All listed geometries were evaluated in the scope of this thesis. However, some geometry caused severe issues, which led to critical failures of the workflow. These geometries were partly replaced with equivalent geometries or neglected in the final model to transform and evaluate the simulation results successfully.

Table 5, summary of transformation results

Use case		**Python and IfcOpenShell**	**OsmSerializer**
geometry			
rectangular surfaces	wall for space over one storey	+	+
	wall for space over multiple storeys	-	+
	slab, ceiling, roof, window, door	+	+
	curtain wall	-	-
	skylight	-	-
sloped surfaces	roof	-	+
	wall	-	-
shading elements		+	-
semantic data			
boundary condition	wall for space over one storey exterior	+	+/-
	wall for space over one storey interior	+/-	+/-
	wall for space over multiple storeys	-	+/-
	wall to ground	-	-
	window, door exterior	+	+
	window, door interior	+/-	-
	roof	+	+
	ceiling/roof partly to outside	+	+
	basement floor	+	+
	slab above ground partly to outside	-	-
constructions	wall, slab, ceiling, roof	+	+
	window, door	+/-	+
materials	wall, slab, ceiling, roof	+	-
	air gap	+/-	-
	window, door	+/-	-

Legend: working (+), partly working (+/-), not working (-)

4.2 Python and IfcOpenShell

It is necessary to adapt the python scripts to transform the IFC files successfully. Changes are made to avoid issues with language sensitivity and errors due to model variances. A function to convert the resulting IDF to UTF-8 is implemented in the scripts. The original scripts with marked changes or additions are depicted in the appendix. Additionally, contrary to the design specifications described in section 2.4, adjustments are partly made during the design process of the BIM model in ArchiCAD.

The reference lines of the building elements are created according to the workflow description. Walls and slabs are drawn according to the description by Patel (2020). However, changes to the reference lines for the slabs to the ground are made for a successful transformation process. All reference lines of slabs, ceilings, and roofs are set to be on the bottom of the building element.

The results of the transformation process are the following: The scripts for transforming constructions worked only for multi-layered constructions. No information concerning the roughness is available. Therefore, as a uniform assumption it is set for all materials to “MediumRough”. Air layers are transformed by the script. However, if there are varying thicknesses of air layers used in one or more constructions this is not working because all previously created air layer material types are overwritten during the transformation process. The resulting IDF does contain only one type of air layer. The window and door construction information are missing after the transformation. Therefore, IDF classes for the construction and the related materials are manually added. The reference of the building surfaces to the window construction is added in the python scripts.

The boundary conditions of the interior walls are correctly assigned to the objects. However, the faces are not consistently orientated in the correct direction. This is also valid for interior openings, such as windows or doors, and slabs between zones. Manual correction in SketchUp is required. The room over two storeys is not converted successfully. The interior walls of the plenum in the upper floors is not existing in the IDF file and the corresponding walls on the adjacent zones are created with the wrong boundary definition, “Outdoors” instead of “Zone”. The boundary conditions of walls adjacent to the ground are not defined correctly. In the python scripts, no differentiation between the different types of outside boundaries is implemented.

Slabs of zones that are partly above outside air have the boundary condition “Ground” instead of “Outdoors”. There is also no differentiation of outside boundary conditions for the bottom of zone geometries implemented in the python scripts. The curtain walls are defined in the IfcClass IfcCurtainWall. The object itself has no material information but relationships to other objects within the curtain wall. These connections are stored in the IfcClass IfcRelAggregates for the rails, columns, and panels. The material information is related to these objects, which are represented in the IfcClasses IfcPlate (panel), IfcBuildingElementProxy (rails), and IfcMember (columns). Due to this structure, it is not possible to convert the geometry with the available python scripts. The scripts cannot be run successfully. Therefore, curtain walls are neglected in the evaluation of the simulation results.

Skylight geometry is converted. However, the skylights and the corresponding roofs are not on the same plane and are therefore not converted correctly. Sloped roofs are converted as surfaces without slope with the base z-coordinate. With a degree deviating of 90°, sloped walls are not converted because they are not recognized by the scripts as walls but as roofs. With further assignment of dimensions in ArchiCAD, the objects would be converted as sloped roofs, which is not working properly. Due to the wrong definition of these objects, it is not possible to run the scripts successfully. Sloped walls are therefore neglected in the evaluation of the simulation results. Slabs without the IfcRelSpaceBoundary attribute are converted correctly to IDF Shading:Building objects.

Door objects cause issues in the process of exporting to IFC and viewing the file with the FZKViewer. However, the transformation from IFC to IDF works. The dimensions of the doors are larger than intended because the frame of the doors is included. In the design process, it is necessary to define a unique ID for each object. Especially if objects are copied in ArchiCAD, a manual definition of a unique ID is required. It is possible to double-check the object IDs in the IFC file via the FZKViewer.

The model created in ArchiCAD and the exported IFC are shown in Figure 23. The transformed IDF opened in SketchUp and coloured by construction type and boundary condition is illustrated in Figure 24.

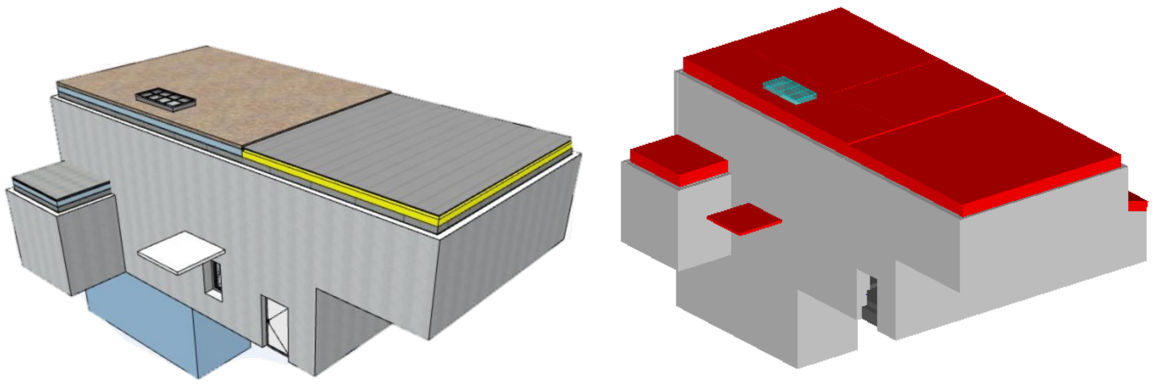


Figure 23, ArchiCAD and IFC model of case study (own illustration)

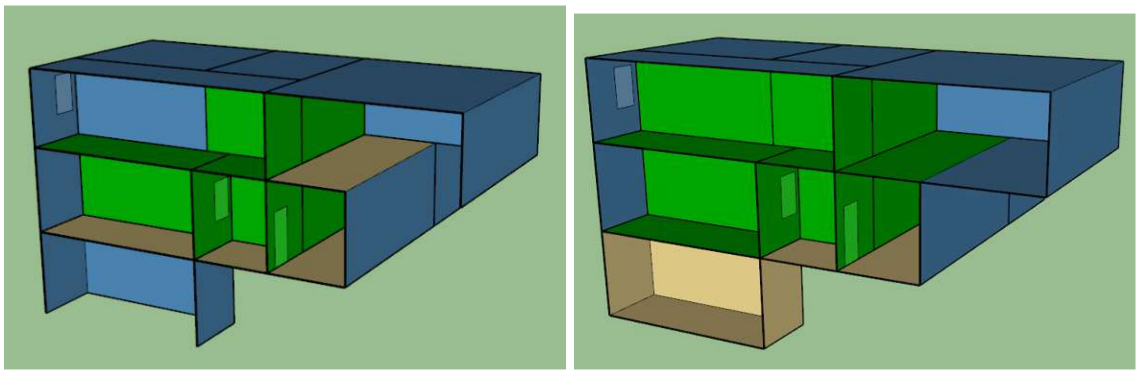


Figure 24, section through the resulting IDF in SketchUp rendered by boundary condition; Transformation result (left) and corrected model (right) (own illustration)

4.3 OsmSerializer

The custom Revit exporter used by Ramaji et al. (2020b) is in the scope of this thesis not available. Therefore, the proposed GSA-05 MVD is not available. The best results to export a simple model from Revit and transform it are achieved with the MVD Coordination View 2.0 with 2nd level space boundaries. For that reason, this MVD is used for the case study.

Ramaji et al. (2020b) used OpenStudio version 2.0.0. In the scope of this thesis the transformation is tested with OpenStudio version 2.0.0 and OpenStudio version 2.9.1. No difference can be detected between the transformation in both versions as they are both performed in the BIMserver and OpenStudio only acts as an interface. For better comparability with the Python workflow, using EnergyPlus version 9.2, which is the engine for OpenStudio version 2.9.1, this version is also used for the OsmSerializer workflow.

The results of the transformation process are the following: A space over two storeys is transformed correctly. Sloped roofs are converted correctly. Slabs of zones that are at least partly above outside air have the boundary condition “Ground” instead of “Outdoors”. The boundary conditions of walls adjacent to the ground are not defined correctly. The boundary conditions of the interior walls are correctly assigned to the objects. However, the faces are not consistently orientated in the correct direction. Slabs have partly the wrong orientation. Manual correction of the boundary conditions in SketchUp is required. Assignment of the created spaces to thermal zones is not implemented in this workflow and is therefore done manually in SketchUp with the OpenStudio add-on. Curtain walls, skylights, shading elements, and interior openings (doors and windows) are not converted at all.

As described in section 3.4.5 it is not possible to retrieve the material information from Revit via the standard IFC exporter. Constructions are exported with the correct thickness but all information regarding the thermal performance of the building objects is missing.

The model created in Revit and the exported IFC are depicted in Figure 25. The transformed IDF opened in SketchUp and coloured by construction type and boundary condition is shown in Figure 26.

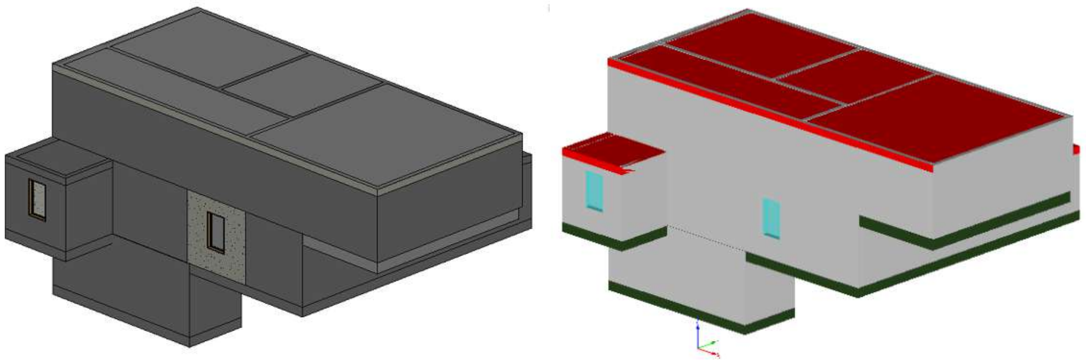


Figure 25, Revit and IFC model of the case study (own illustration)

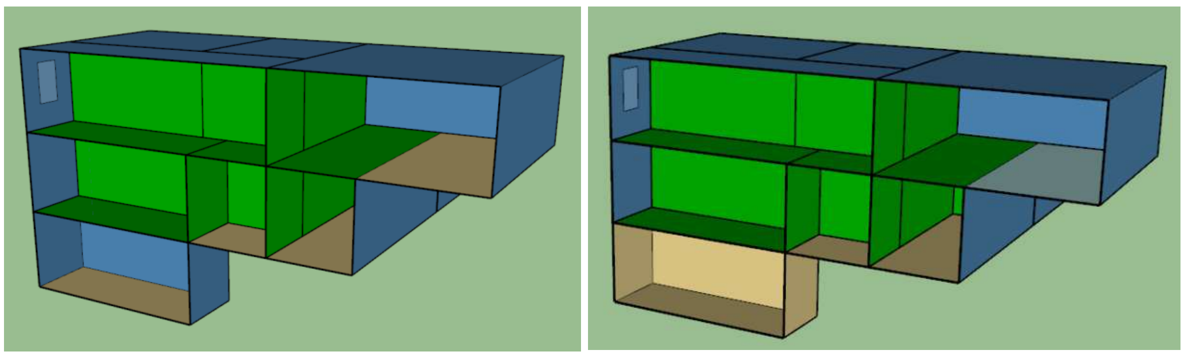


Figure 26, section through the resulting IDF in SketchUp rendered by boundary condition; Transformation result (left) and corrected model (right) (own illustration)

4.4 Simulation results

To be able to compare the simulation results the transformed constructions with the correct materials definitions of the Python and IfcOpenShell workflow were used for the manual baseline model and the OsmSerializer model. The results excerpted from the EnergyPlus HTML output files are listed in Table 6.

Table 6, simulation results

Name	Manual baseline	Python and IfcOpenShell	OsmSerializer
building area [m^2]	300.00	240.00	222.25
building volume [m^3]	1027.00	647.58	812.32
gross wall area [m^2]	397.80	387.90	365.26
heating load [MWh.a^{-1}]	8.75	9.11	7.29
cooling load [MWh.a^{-1}]	0.56	0.26	0.07

The varying building areas can be justified with the calculation methods of the zone areas. In the baseline workflow the zone area is calculated based on the geometry boundaries. In the workflows based on the IFC the areas are calculated in the workflow. This has also an impact on the volumes. Especially in the Python and IfcOpenShell workflow the volume is directly extracted from the ArchiCAD room properties.

The efforts for the chosen workflows are compared in Table 7. In the first row the time for the creation of the 3D building model in SketchUp, ArchiCAD and Revit, the construction assignment, and the definition of the output settings for the IFC exporters is listed. In the second row the time for the transformation to the IDF and the correction of occurring issues is listed. In the third row the time for the definition and addition of the required simulation classes in the IDF is listed

Table 7, comparison of the efforts of the evaluated workflows

Name	**Manual baseline**	**Python and IfcOpenShell**	**OsmSerializer**
Model creation [hours]	2	4.5	3.5
IDF transformation & issue correction [hours]	1.5	0.5	1.5
IDF simulation classes [hours]	0.5	0.5	1
overall duration [hours]	**4**	**5.5**	**6**

Overall, the time required for the manual baseline method is the lowest. With changing geometry in a recursive design process this can change rapidly. The workflows with the semi-automated transformation do have an advantage over the manual creation if different geometry variations should be evaluated. Furthermore, the automatic transformation of the constructions pre-defined in the BIM-authoring tools leads to reduced mistakes in the process of defining the EnergyPlus classes and can therefore lead to an improved replicability and accuracy of the overall process. The tedious efforts for manual definition of constructions and materials can be obsolete with the automatic transformation.

5 CONCLUSION

Aside from the answers to the defined research questions, this chapter describes the limitations of this study, gives a conclusion of the conducted work and an outlook on possible future work in the field of BIM-based BEM.

5.1 Research Questions

5.1.1 Research Question 1

Which IFC-based BIM-based BEM workflows are currently available and usable for semi-automatic IDF generation with the focus on information related to geometry and building objects for thermal building performance simulations? Which requirements and limitations exist in the scope of the complete process?

Answer:

A literature research for BIM-based BEM workflows was conducted. The focus of this research were IFC-based approaches for dynamic thermal building performance simulations with the engine EnergyPlus. A selection of IFC-based BIM-to-BEM workflows was done. The reviewed workflows are only an excerpt of all available workflows. However, due to similarities between almost all workflows that transform an exchange data format into the EnergyPlus IDF data format, the listed workflows are still representative. Two free-to-use, publicly available and promising approaches of this selection were chosen for the in-detail evaluation in a case study. The different steps of a successful BIM-to-BEM transformation workflow were afterwards described in detail, and difficulties were highlighted.

5.1.2 Research Question 2

How do the previously chosen workflows perform on a detailed case study?

Answer:

The two chosen workflows were evaluated in detail on an artificial case study building. The building was created in the BIM authoring tools ArchiCAD 24 and Revit 2020. The first transformation workflow uses an IFC file exported from ArchiCAD and transforms this file via Python scripts and the IfcOpenShell library to an IDF file. Building geometry, construction information and thermal zones were transformed. Window construction and material information was manually added. The second transformation workflow uses an IFC file exported from Revit and transforms this file via the add-on OsmSerializer for BIMserver and OpenStudio to an OSM. Building geometry, constructions without material information and space definitions were transformed. The spaces were afterwards in SketchUp in combination with the OpenStudio plugin manually assigned to thermal zones. Issues, such as the wrong orientation of surfaces, were manually corrected. An IDF file was afterwards exported via the OpenStudio plugin. Afterwards, required simulation parameters were manually added to the resulting IDF files of both workflows before running the simulation. The results of both simulations were finally evaluated and compared to each other and a manually created baseline model.

It can be concluded for both evaluated workflows that the transformation of most used rectangular building objects was done correctly. Sophisticated geometry objects, such as curtain walls or sloped objects were at least partly not converted correctly. Constructions of the building objects were in both workflows translated successfully. For the materials however, issues with the export of the BIM authoring tools occurred. The creation of space boundaries to define the relation between the different spaces and thermal zones was at least partly not done correctly by the workflows. All these issues caused the need for manual post-processing of the resulting IDF files.

5.2 Limitations of this study

In the scope of the this study the topic of BIM-based BEM workflows was partly evaluated. Limitations were made by focusing on workflows for dynamic simulation for thermal building performance simulation and specifying the intermediate data exchange format as IFC and the resulting data format as EnergyPlus IDF.

An excerpt of representative workflows was listed and described. The built-in possibilities in the BIM authoring tools ArchiCAD 24 and Revit 2020 to define properties required for the previously described limitations were evaluated. Furthermore, the export of the available information to IFC was reviewed.

The transformation results for several geometry types with varying boundary conditions and different construction types were evaluated for two chosen transformation workflows.

The required efforts for a successful simulation based on the two transformed IDFs were collated. The time for the preparation of the BIM-model for later transformation, correction of issues caused by the transformation and addition of required simulation properties was compared for the two workflows and one manually created reference base case.

The computational effort, e.g., the runtime of the transformation processes and simulation, is not covered by this study because this aspect is highly influenced by the hardware and software environment used by the individual user. The efforts for the setup of the BIM-authoring tools and the required tools for the two chosen workflows were not evaluated because these efforts are also depending on the used hardware and software environment.

Another topic not covered in the scope of this study is the usability of the evaluated workflows. Even the most accurate and high-quality workflows are not going to be accepted and used by a broad target audience if the usability of the required tools is not satisfying. However, the topic of usability for BIM-based BEM workflows is difficult to cover because several different software tools are used.

5.3 Conclusion

Based on the results of the study for two representative workflows, the following statements are applicable to BIM-based BEM transformation workflows. The incorrect data transformation is mainly a topology-based issue. This could be solved with guidelines and standards that define the structure of the property definition. The main issue for BIM-based BEM is the deviation of models created in the design process in a BIM environment. There are currently no international standards that define how objects need to be created in BIM authoring tools. This leads to severe issues and often to incompatibilities in the (semi-) automated transformation processes for energy performance analysis simulations. Due to several different actors and therefore software and BIM environments in the scope of building projects it is hardly possible to define standards for the unique representation of building objects that are suitable for all project participants. Therefore, automated information transfer between BIM authoring tools and commonly used software tools for BEM is still an elusive goal.

5.4 Future work

Based on the elaborated topics some recommendations for future studies can be given. Most important is the definition of standards for the design of BIM models in the BIM authoring tools for the later usage. At the projects start it must for which purpose the BIM model is created. Standards for the process of creating the BIM model must be considered if the usage for BEM in a later project stage is foreseen.

An important part is the lack of clearly defined export properties for the usage of energy modelling. The definition of clear export standards for BIM authoring tools such as Revit or ArchiCAD could overcome this issue. The exported IFC file should include the semantic information from the BIM authoring tool in a structured manner any workflow can process. This should be independent of the type of algorithm used for geometry processing and included intermediate data formats. Revit IFC exporter should be extended to support material layer property export. This not only affects the BIM-to-BEM workflows for heat balance simulations, but structural or ecological simulations would also benefit from this enhancement.

The topic of usability is encompassing partly the previous mentioned points. The usability of tools and their extensions used in the scope of transformation workflows and the interfaces between the different tools should be evaluated and enhanced to make the transformation workflows on a regular basis viable to a broader target audience.

To avoid issues caused by the intermediate exchange formats, solutions based on the BIM authoring tools API are an alternative solution that could developed in future studies (O'Donnell et al. 2019). However, proficient knowledge and maintenance effort for direct API access would be required for this approach.

Language sensitivity is a common issue in all researched workflows. Building Smart Data dictionary (buildingSMART International 2021a) is a first step towards solving this issue because elements have a standardised naming that is understandable in many different languages. However, further development for the associated properties of the elements is necessary.

BIM-based BEM workflows should integrate a quality check in each process step to ensure the accuracy and correctness of the workflow. A tool for the comparison of BEM models with the input files was developed recently (Hong 2020). The integration of this kind of tool in a standardised BIM-based BEM workflow should be investigated.

Semantic web and connected data approaches, similar to the SEMERGY approach, offer an alternative to the manual definition of correct object properties in BIM for later BEM usage (Karlapudi et al. 2020). With the rising complexity of BIM models and the availability of databases encompassing in-detail information about building objects the significance of this approach could rise in the next years.

6 INDEX

6.1 List of Figures

6.2 List of Tables

7 REFERENCES

ACCA software S.p.A. (2020a) *IFC schema (part 3): the IfcPropertyDefinition* [Online]. Available at https://biblus.accasoftware.com/en/ifc-schema-part-3-the-ifcpropertydefinition/ (Accessed 7 December 2020).

ACCA software S.p.A. (2020b) *IFC schema (part2): the IfcRelationship concept* [Online]. Available at https://biblus.accasoftware.com/en/ifc-schema-the-ifcrelationship-concept/ (Accessed 7 December 2020).

Alliance for Sustainable Energy, LLC *OpenStudio®* [Computer program]. Available at https://www.openstudio.net/ (Accessed 24 April 2021).

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

Autodesk Inc. (2020) *IFC for Revit (revit-ifc) (*20.3.1.0) [Computer program]. Available at https://github.com/Autodesk/revit-ifc (Accessed 2 December 2020).

Autodesk Inc. (2021) *Wall Type Properties* [Online]. Available at https://knowledge.autodesk.com/support/revit-products/learn-explore/caas/CloudHelp/cloudhelp/2015/ENU/Revit-Model/files/GUID-C6A3DD86-FFAF-4781-8E50-5714965C24BE-htm.html (Accessed 7 February 2021).

Bazjanac, V. (2010) *Space boundary requirements for modeling of building geometry for energy and other performance simulation,* Lawrence Berkeley National Laboratory.

Bazjanac, V., Maile, T., Nytsch-Geusen, C. (2016) 'Generation of building geometry for energy performance simulation using Modelica', in Grunewald, J. (ed) *Proceedings of the CESBP Central European Symposium on Building Physics and BauSIM 2016: September 14-16, 2016, Dresden, Germany,* Stuttgart, Fraunhofer IRB Verlag, pp. 361–368.

Beetz, J., van Berlo, L., de Laat, R., von den Helm, P. (2010) 'Bimserver.org - an Open Source IFC model server', *Proceedings of the CIB W78 27th International Conference on Applications of IT in the AEC Industry CIB-W78, Cairo, 16-18 November 2010.* Cairo, Egypt, International Council for Research and Innovation in Building and Construction (CIB).

Big Ladder Software LLC (2021) *Euclid* [Computer program]. Available at https://bigladdersoftware.com/projects/euclid/ (Accessed 5 April 2021).

Borrmann, A., Beetz, J., Koch, C., Liebich, T., Muhic, S. (2018) 'Industry Foundation Classes: A Standardized Data Model for the Vendor-Neutral Exchange of Digital Building Models', in Borrmann, A., König, M., Koch, C., Beetz, J. (eds) *Building information modeling: Technology foundations and industry practice,* Cham, Springer, pp. 81–126.

Borrmann, A., Forster, C., Liebich, T., König, M., Tulke, J. (2021) 'Germany's Governmental BIM Initiative – The BIM4INFRA2020 Project Implementing the BIM Roadmap', in Santos, E. T., Scheer, S. (eds) *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering,* Springer Nature, pp. 452–465.

buildingSMART International (2020a) *IFC String Encoding: String Encoding & Decoding* [Online]. Available at https://technical.buildingsmart.org/resources/ifcimplementationguidance/string-encoding/ (Accessed 6 December 2020).

buildingSMART International (2020b) *IFC4_ADD2_TC1 - 4.0.2.1 [Official]* [Online]. Available at https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ (Accessed 12 April 2021).

buildingSMART International (2020c) *Industry Foundation Classes (IFC) - An Introduction* [Online]. Available at https://technical.buildingsmart.org/standards/ifc (Accessed 31 October 2020).

buildingSMART International (2020d) *Model View Definition (MVD) - An Introduction* [Online]. Available at https://technical.buildingsmart.org/standards/ifc/mvd/ (Accessed 31 October 2020).

buildingSMART International (2021a) *buildingSMART Data Dictionary* [Online]. Available at https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/ (Accessed 17 February 2021).

buildingSMART International (2021b) *IFC Specifications Database* [Online]. Available at https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (Accessed 12 April 2021).

buildingSMART International (2021c) *IfcDoc* [Online]. Available at https://www.buildingsmart.org/standards/groups/ifcdoc/ (Accessed 11 October 2020).

buildingSMART International (2021d) *Information Delivery Manual (IDM)* [Online]. Available at https://technical.buildingsmart.org/standards/information-delivery-manual/ (Accessed 7 February 2021).

buildingSMART International (2021e) *openBIM Definition* [Online]. Available at https://www.buildingsmart.org/about/openbim/openbim-definition/ (Accessed 12 April 2021).

Cemesova, A., Hopfe, C. J., Mcleod, R. S. (2015) 'PassivBIM: Enhancing interoperability between BIM and low energy design software', *Automation in Construction*, vol. 57, pp. 17–32.

Choi, J., Shin, J., Kim, M., Kim, I. (2016) 'Development of openBIM-based energy analysis software to improve the interoperability of energy performance assessment', *Automation in Construction*, vol. 72, pp. 52–64 [Online]. DOI: 10.1016/j.autcon.2016.07.004.

Chopson, P., Ahuja, S., Haymaker, J., Augenbroe, G. (2015) ‘Practical energy and cost optimization methods for selecting massing, materials, and technologies’, *Future of Architecture Research.* Chicago, pp. 280–287.

de Laat, R., van Berlo, L. (2019) *IfcValidator (*0.0.44) [Computer program]. Available at https://github.com/opensourceBIM/IfcValidator (Accessed 9 May 2021).

DesignBuilder Software Ltd (2021) *DesignBuilder* [Computer program]. Available at https://designbuilder.co.uk// (Accessed 1 January 2021).

Digital Alchemy, Inc. (2021a) *10 Things to know about Simergy* [Online]. Available at https://d-alchemy.com/support/thingstoknow (Accessed 24 April 2021).

Digital Alchemy, Inc. (2021b) *Simergy* [Computer program]. Available at https://d-alchemy.com/products/simergy (Accessed 24 April 2021).

Eastman, C. M., Teicholz, P. M., Sacks, R., Lee, G. (2018) *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors* [Online], Hoboken, New Jersey, Wiley. Available at https://onlinelibrary.wiley.com/doi/book/10.1002/9781119287568.

El Asmi, E., Robert, S., Haas, B., Zreik, K. (2015) ‘A standardized approach to BIM and energy simulation connection’, *International Journal of Design Sciences and Technology*, vol. 21, no. 1, pp. 59–82 [Online]. Available at https://hal-cea.archives-ouvertes.fr/cea-01847290.

EQUA Simulation AB (2020) *IDA Indoor Climate and Energy* [Computer program]. Available at https://www.equa.se/en/ida-ice.

Fenz, S., Heurix, J., Neubauer, T., Tjoa, A. M., Ghiassi, N., Pont, U., Mahdavi, A. (2016) ‘SEMERGY.net: automatically identifying and optimizing energy-efficient building designs’, *Computer Science - Research and Development*, vol. 31, no. 3, pp. 135–140.

Fernald, H., Hong, S., Bucking, S., O'Brien, W. (2018) ‘BIM to BEM translation workflows and their challenges: a case study using a detailed BIM model’, *Proceedings of eSim 2018, the 10th conference of IBPSA-Canada.* Montréal, QC, Canada, May 9-10, pp. 482–491.

Gao, H., Koch, C., Wu, Y. (2019) ‘Building information modelling based building energy modelling: A review’, *Applied Energy*, vol. 238, pp. 320–343.

Ghiassi, N. (2013) *Development of a Building Data Model for a Performance-Based Optimization Environment*, Diploma Thesis, Supervisor: Mahdavi, A., Vienna, University of Technology Vienna.

Giannakis, G., Katsigarakis, K., Lilis, G. N., Álvarez-Diaz, S. (2019a) *GUIDELINES for OptEEmAL BIM Input Files* [Online]. Available at https://www.opteemal-project.eu/files/guidelines_for_opteemal_bim_input_files_v11.pdf (Accessed 14 November 2020).

Giannakis, G., Katsigarakis, K., Lilis, G. N., Rovas, D. (2019b) 'A Workflow for Automated Building Energy Performance Model Generation Using BIM Data', *Proceedings of building simulation 2019: 16th IBPSA International conference and exhibition, 2-4 Sept., Rome.* Rome, International Building Performance Simulation Association, pp. 167–174.

Gourlis, G., Kovacic, I. (2017) 'Building Information Modelling for analysis of energy efficient industrial buildings – A case study', *Renewable and Sustainable Energy Reviews*, vol. 68, pp. 953–963.

Green Building XML Schema (n.d.) *About GreenBuildingXML* [Online]. Available at https://www.gbxml.org/About_GreenBuildingXML_gbXML (Accessed 10 October 2020).

Hitchcock, R. J., Wong, J. (2011) 'Transforming IFC architectural view BIMS for energy simulation: 2011', *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November.* Sydney, International Building Performance Simulation Association, pp. 1089–1095.

Ho, D. (2021) *Notepad++ (*7.9.1) [Computer program]. Available at https://notepad-plus-plus.org/ (Accessed 21 March 2021).

Hong, S. (2020) *Geometric Accuracy of BIM-BEM Transformation Workflows: Bridging the State-of-the-Art and Practice*, Master Thesis, Ottawa, Ontario, Carleton University.

Integrated Environmental Solutions Limited (2020) *IES Virtual Environment* [Computer program]. Available at https://www.iesve.com/software/virtual-environment (Accessed 1 January 2021).

Ivanova, I., Kiesel, K., Mahdavi, A. (2015) 'BIM-generated data models for EnergyPlus: A comparison of gbXML and IFC Formats', *Building Simulation Applications BSA 2015: 2nd IBPSA-Italy conference Bozen-Bolzano, 4th – 6th February 2015.* Bozen-Bolzano, Bozen-Bolzano University Press, pp. 407–414.

Jiang, S., Jiang, L., Han, Y., Wu, Z., Wang, N. (2019) 'OpenBIM: An Enabling Solution for Information Interoperability', *Applied Sciences*, vol. 9, no. 24, p. 5358.

Kamel, E., Memari, A. M. (2019) 'Review of BIM's application in energy simulation: Tools, issues, and solutions', *Automation in Construction*, vol. 97, pp. 164–180.

Karlapudi, J., Menzel, K. (2020) 'Analysis on automatic generation of BEPS models from BIM model', *BauSIM 2020 - 8th Conference of IBPSA Germany and Austria: 23-25 September 2020, Graz University of Technology, Austria; Proceedings.* Graz, Austria, 23-25 September. Graz, Verlag der Technischen Universität Graz, pp. 535–542.

Karlsruher Institute of Technology, Institute for Automation and Applied Informatics (2020) *FZKViewer (*6.0, Build 1816) [Computer program]. Available at https://www.iai.kit.edu/1648.php (Accessed 14 November 2020).

Karlsruher Institute of Technology, Institute for Automation and Applied Informatics (2021) *IfcCheckingTool (*2.2, Build 91) [Computer program]. Available at https://www.iai.kit.edu/english/1649.php (Accessed 9 May 2021).

Krijnen, T. (2019) *Efficient storage and retrieval of detailed building models: multi-disciplinary and long-term use of geometric and semantic construction information*, Dissertation, Eindhoven, Technische Universiteit Eindhoven.

Krijnen, T., Moult, D. (2020) *IfcOpenShell* [Online]. Available at http://ifcopenshell.org/ (Accessed 6 December 2020).

Ladenhauf, D., Battisti, K., Berndt, R., Eggeling, E., Fellner, D. W., Gratzl-Michlmair, M., Ullrich, T. (2016) 'Computational geometry in the context of building information modeling', *Energy and Buildings*, vol. 115, pp. 78–84.

Ladenhauf, D., Berndt, R., Eggeling, E., Ullrich, T., Battisti, K., Gratzl-Michlmair, M. (2014) 'From Building Information Models to Simplified Geometries for Energy Performance Analysis', in Lazarevic, E. V., Krstic-Furundzic, A., Dukic, A., Vukmirovic, M. (eds) *Proceedings of First International Academic Conference on Places and Technologies,* Belgrade, University of Belgrade - Faculty of Architecture, pp. 669–676.

Ladybug Tools LLC *Ladybug Tools* [Online]. Available at https://www.ladybug.tools/ (Accessed 1 January 2021).

Larsson, N. (2009) *The Integrated Design Process; History and Analysis,* International Initiative for a Sustainable Built Environment.

Lilis, G. N., Giannakis, G., Katsigarakis, K., Rovas, D. (2019) 'Space Boundary Topology Simplification for Building Energy Performance Simulation Speedup', *Proceedings of building simulation 2019: 16th IBPSA International conference and exhibition, 2-4 Sept., Rome.* Rome, International Building Performance Simulation Association, pp. 175–181.

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

Lipman, R. (2017) 'IFC File Analyzer Software', *Journal of Research of the National Institute of Standards and Technology*, no. 122 [Online]. DOI: 10.6028/jres.122.015.

Mahdavi, A. (1996) 'Semper: A New Computational Environment for Simulation-based Building Design Assistance', *Proceedings of the 1996 International Symposium of CIB W67: Energy and Mass Flows in the Life Cycle of Buildings.* Vienna, Austria, pp. 467–472.

Maile, T., Fischer, M., Bazjanac, V. (2007) 'Building Energy Performance Simulation Tools - a Life-Cycle and Interoperable Perspective', *Facil. Eng. (CIFE) Working Pap.*, vol. 107.

Maile, T., O'Donnell, J. T., Bazjanac, V., Rose, C. (2013) 'BIM - Geometry modeling guidelines for building energy performance simulation', *Proceedings of BS2013: 13th*

Conference of IBPSA: 13th International Conference of the International Building Performance Simulation Association. Chambery, France, pp. 3242–3249.

Moult, D., Mackey, C. (2020) *ladybug-blender (*v0.1.1) [Computer program]. Available at https://github.com/ladybug-tools/ladybug-blender (Accessed 21 November 2020).

Nasyrov, V., Stratbücker, S., Ritter, F., Borrmann, A., Shan, H., Lindauer, M. (2014) 'Building information models as input for building energy performance simulation – the current state of industrial implementations', in Mahdavi, A., Martens, B., Scherer, R. J. (eds) *eWork and eBusiness in architecture, engineering and construction: Proceedings of the 10th European Conference on Product and Process Modelling (ECPPM 2014), Vienna, Austria, 17-19 September 2014,* Balkema, 1/1/Boca Raton|London|New York|Leiden, CRC Press, pp. 479–486.

Nielsen, J. (1994) *Usability Engineering*, Morgan Kaufmann Publishers Inc.

O'Donnell, J. T., See, R., Rose, C., Maile, T., Bazjanac, V., Haves, P. (2011) 'SimModel: A domain data model for whole building energy simulation', *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association, Sydney, 14-16 November.* Sydney, International Building Performance Simulation Association.

O'Donnell, J. T., van Dessel, M., Maile, T. (2019) 'BIM to Building Energy Performance Simulation: An Evaluation of Current Industry Transfer Processes', *Proceedings of building simulation 2019: 16th IBPSA International conference and exhibition, 2-4 Sept., Rome.* Rome, International Building Performance Simulation Association, pp. 92–99.

Patel, K. P. (2020) *BIM Model Enrichment for Energy Performance Simulation*, Master Thesis, Supervisor: Menzel, K., Dresden, Technische Universität Dresden.

Picco, M., Marengo, M. (2015) 'On the Impact of Simplifications on Building Energy Simulation for Early Stage Building Design', *Journal of Engineering and Architecture*, vol. 3, no. 1, pp. 66–78.

Pinheiro, S., Wimmer, R., O'Donnell, J. T., 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*, vol. 90, pp. 91–103.

Plandata GmbH (n.d.) *Open BIM and Closed BIM: BIMpedia* [Online]. Available at https://www.bimpedia.eu/-/1002-open-bim-und-closed-bim (Accessed 12 April 2021).

Pont, U. (2014) *A comprehensive approach to web-enabled, optimization-based decision support in building design and retrofit*, Dissertation, Supervisor: Mahdavi, A., Vienna, University of Technology Vienna.

Pont, U., Ghiassi, N., Fenz, S., Heurix, J., Mahdavi, A. (2015) 'SEMERGY: Application of Semantic Web Technologies in Performance-Guided Building Design Optimization', vol. 20, Special Issue, pp. 107–120.

Pont, U., Ghiassi, N., Shayeganfar, F., Mahdavi, A., Fenz, S., Heurix, J., Anjomshoaa, A. (2014) 'SEMERGY: Utilizing semantic web technologies for performance-guided building design optimization', in Mahdavi, A., Martens, B., Scherer, R. J. (eds) *eWork and eBusiness in architecture, engineering and construction: Proceedings of the 10th European Conference on Product and Process Modelling (ECPPM 2014), Vienna, Austria, 17-19 September 2014,* Balkema, 1/1/Boca Raton|London|New York|Leiden, CRC Press.

Python Software Foundation (2021) *What is Python? Executive Summary* [Online]. Available at https://www.python.org/doc/essays/blurb/ (Accessed 16 February 2021).

Ramaji, I. J., Memari, A. M. (2020a) 'Interpreted information exchange: implementation point of view', *Journal of Information Technology in Construction*, vol. 25, pp. 123–139.

Ramaji, I. J., Messner, J. I., Leicht, R. M. (2016) 'Leveraging Building Information Models in IFC to Perform Energy Analysis in OpenStudio', *ASHRAE and IBPSA-USA SimBuild 2016: Building Performance Modeling Conference.* Salt Lake City, Utah, pp. 251–258.

Ramaji, I. J., Messner, J. I., Mostavi, E. (2020b) 'IFC-Based BIM-to-BEM Model Transformation', *Journal of Computing in Civil Engineering*, vol. 34, no. 3, 04020005, 1–13.

Reisinger, J., Donkor, L., Loncsek, S., Kovacic, I. (2019) 'BIM-based workflows for building energy modelling – A variant study', *International Conference on Sustainable Materials, Systems and Structures (SMSS2019): Energy Efficient Building Design and Legislation.* Rovinj, Croatia, 20-22 March. Paris, RILEM Publications SARL, pp. 229–236.

Rose, C. (2012) *Space Boundary Tool (SBT)* (00) [Computer program]. Available at https://www.osti.gov/biblio/1323182-space-boundary-tool-sbt.

Schwartz, Y., Raslan, R. (2013) 'Variations in results of building energy simulation tools, and their impact on BREEAM and LEED ratings: A case study', *Energy and Buildings*, vol. 62, pp. 350–359.

Shadrina, A. (2015) *Framework for the transfer of building materials data between the BIM and thermal simulation software*, Diploma Thesis, Supervisor: Mahdavi, A., Vienna, University of Technology Vienna.

Shadrina, A., Gutierrez, A., Sporr, A., Blank-Landeshammer, B., Fallmann, S., Zucker, G., Ruhsam, C., Ferreiro Sistiaga, A., Kogler, K. (2020) *A decision support system for BIM managers on the example of thermal simulation* [Online]. Available at https://www.isis-papyrus.com/Download/whitepapers/academy/A-Decision-Support-System-for-BIM-Managers-on-the-Example-of-Thermal-Simulation.pdf.

Somboonwit, N., Boontore, A., Rugwongwan, Y. (2017) ‘Obstacles to the Automation of Building Performance Simulation: Adaptive Building Integrated Photovoltaic (BIPV) design’, *Environment-Behaviour Proceedings Journal*, vol. 2, no. 5, pp. 343–354.

Thermal Energy System Specialists, LLC (2019) *TRNSYS: Transient System Simulation Tool* [Computer program]. Available at http://www.trnsys.com/ (Accessed 5 April 2021).

U.S. Department of Energy (2019) *Space Boundary Tool* [Computer program]. Available at https://simulationresearch.lbl.gov/projects/space-boundary-tool (Accessed 6 December 2020).

U.S. Department of Energy (2021) *EnergyPlus (*9.2) [Computer program]. Available at https://energyplus.net/ (Accessed 27 March 2021).

US GSA (ed) (2009) *Information Delivery Manual (IDM) for BIM Based Energy Analysis as Part of the Concept Design BIM 2010* [Online]. Available at http://www.blis-project.org/IAI-MVD/IDM/BSA-002/PM_BSA-002.pdf (Accessed 10 January 2020).

Wetter, M. (2009) ‘Modelica-based modelling and simulation to support research and development in building energy and control systems’, *Journal of Building Performance Simulation*, vol. 2, no. 2, pp. 143–161.

Xylem Technologies *SEMERGY* [Online]. Available at https://semergy.xylem-technologies.com/content/ (Accessed 13 April 2021).

Yu, N. (2014) *Information Interoperability between Building Information Modeling Authoring Tools and Simulation Tools to Support Energy Efficient Building Design*, Master Thesis, Supervisor: Wu, D., The Pennsylvania State University, The Graduate School.

Zhi-liang, M., Tonghua, Z., Zhenhua, W., Yan, F. (2012) ‘Transformation from IFC data of design results to IDF data for analysis of building's energy consumption’, *7th International Conference on Innovation in Architecture, Engineering & Construction.* São Paulo, Brazil, 15-17 August.

8 APPENDIX

A. Case study

Table A-1, case study building constructions defined in BIM authoring tools

Building object	Construction	U-value [W.m^{-2}.K^{-1}]
Wall exterior to outside air	concrete wall with external insulation	0.20
Wall exterior to ground	concrete wall with external perimeter insulation	0.25
Wall interior	concrete wall	2.40
	brick wall	0.45
	lightweight wall	0.41
Floor to ground	concrete floor with external insulation	0.14
Floor to outside air	concrete floor with external insulation	0.20
Roof	flat roof with gravel	0.14
	flat roof with metal sheeting	0.16
	roof terrace	0.15
	wooden pitched roof	0.15
Interior ceiling	wooden ceiling	0.21
Window	double layer glazing	1.50

A.1. ArchiCAD

A.1.1. ArchiCAD model

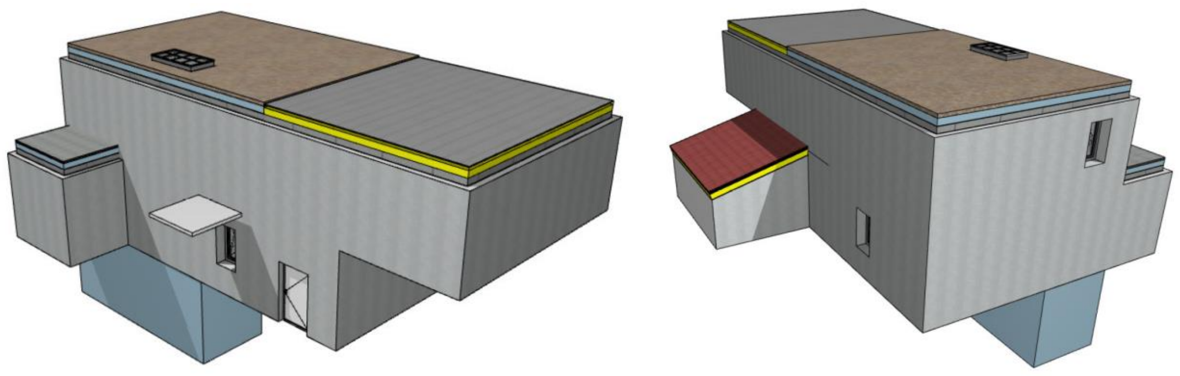


Figure A-1, ArchiCAD 3D-views

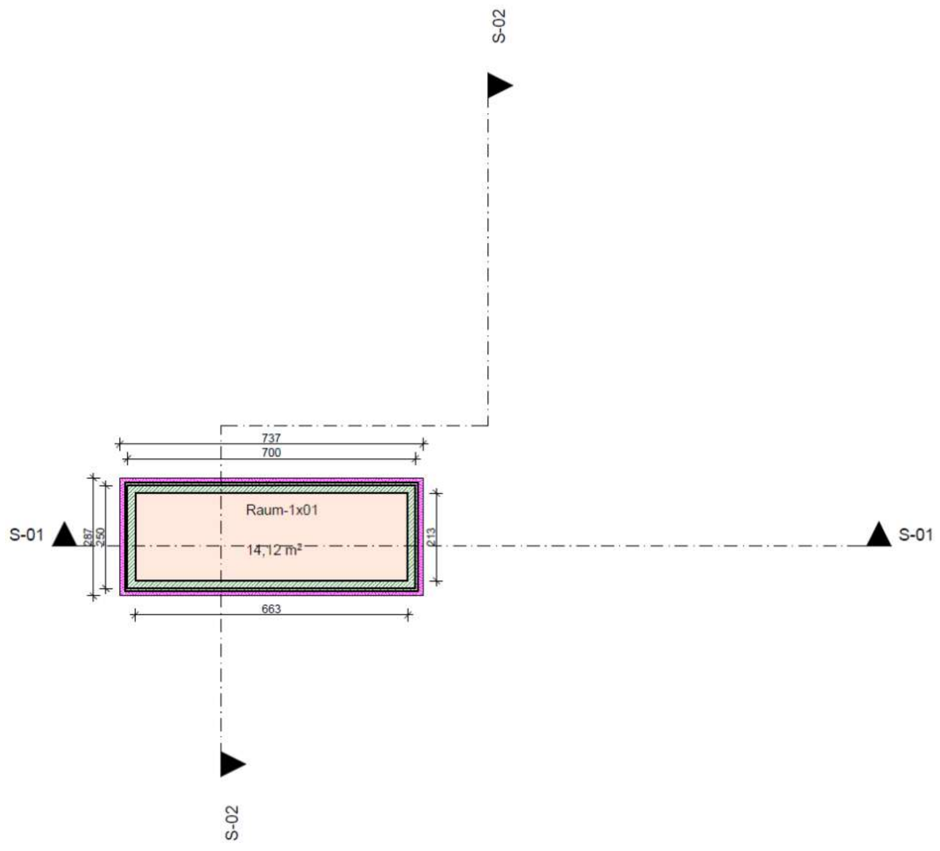


Figure A-2, ArchiCAD basement plan

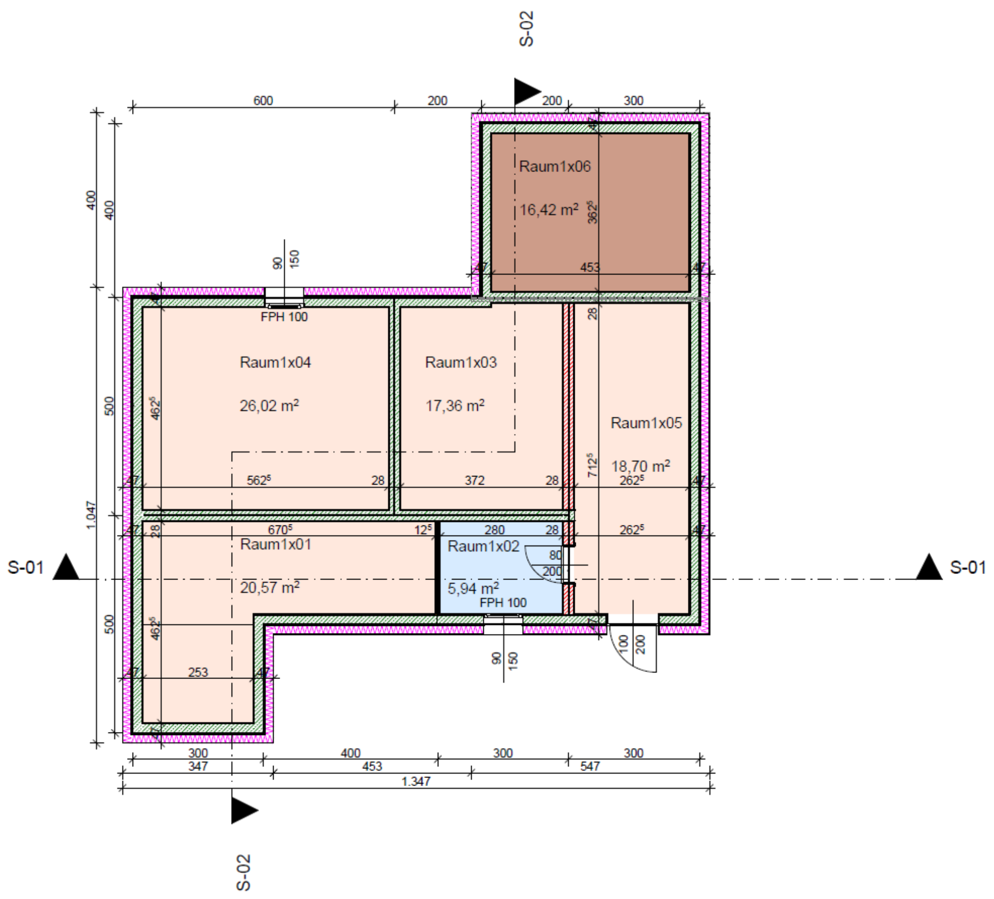


Figure A-3, ArchiCAD ground floor plan

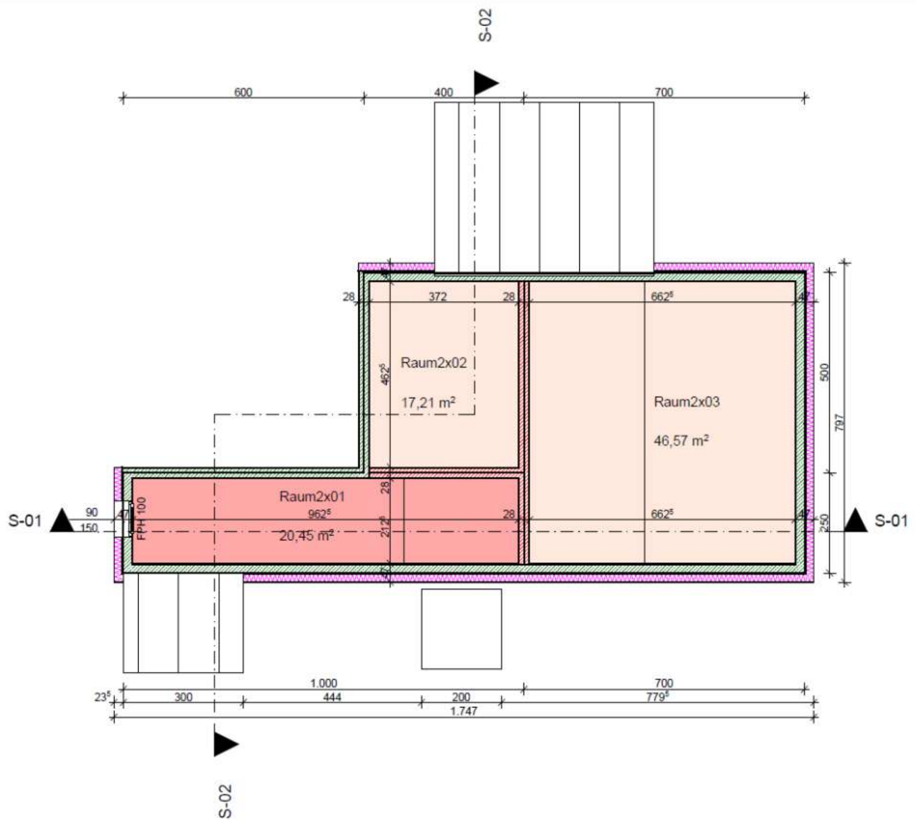


Figure A-4, ArchiCAD first floor plan

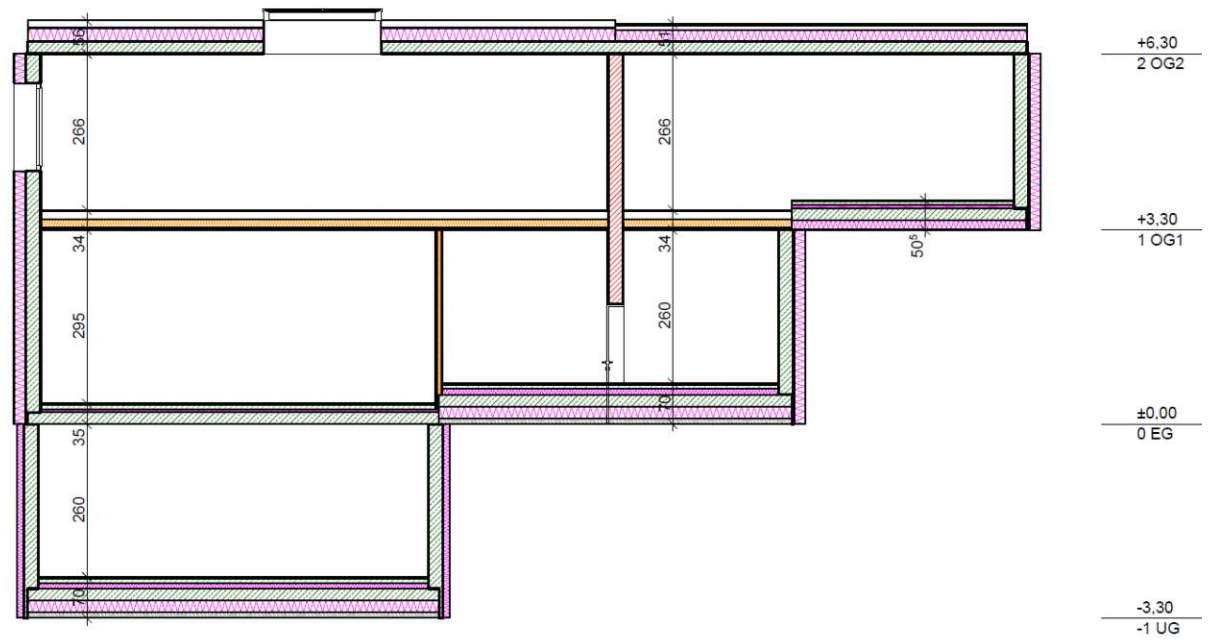


Figure A-5, ArchiCAD horizontal section

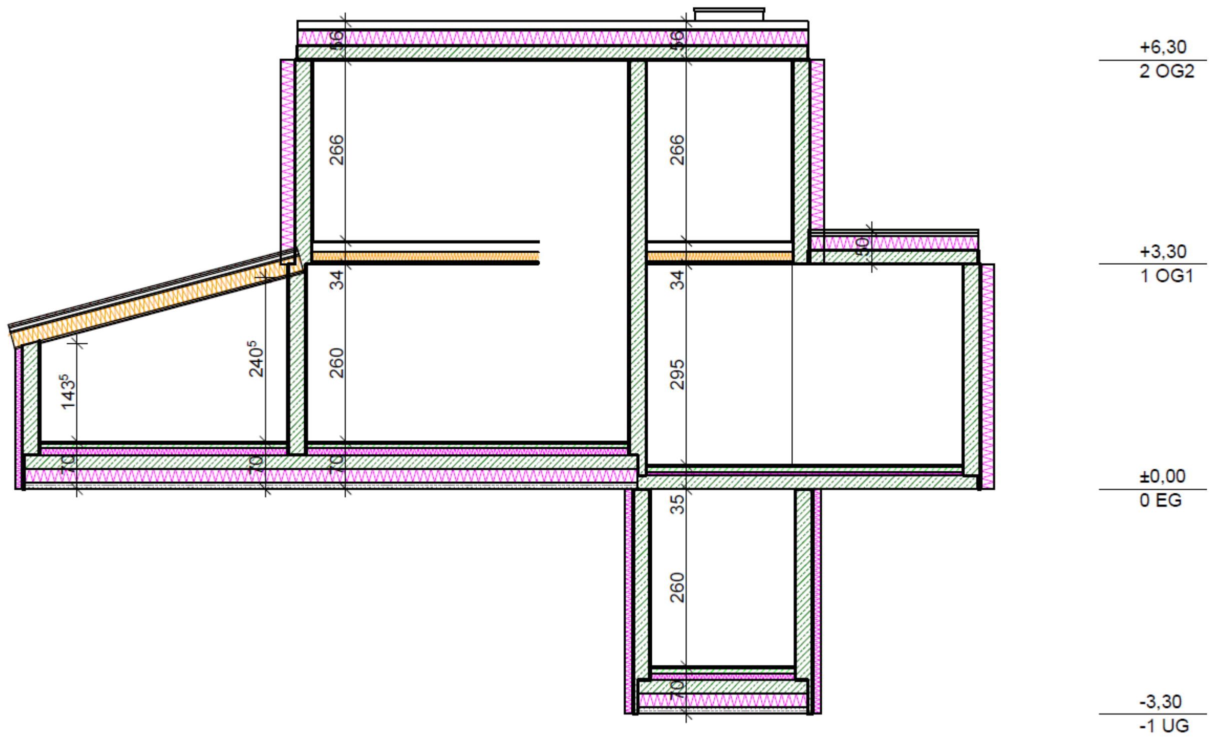
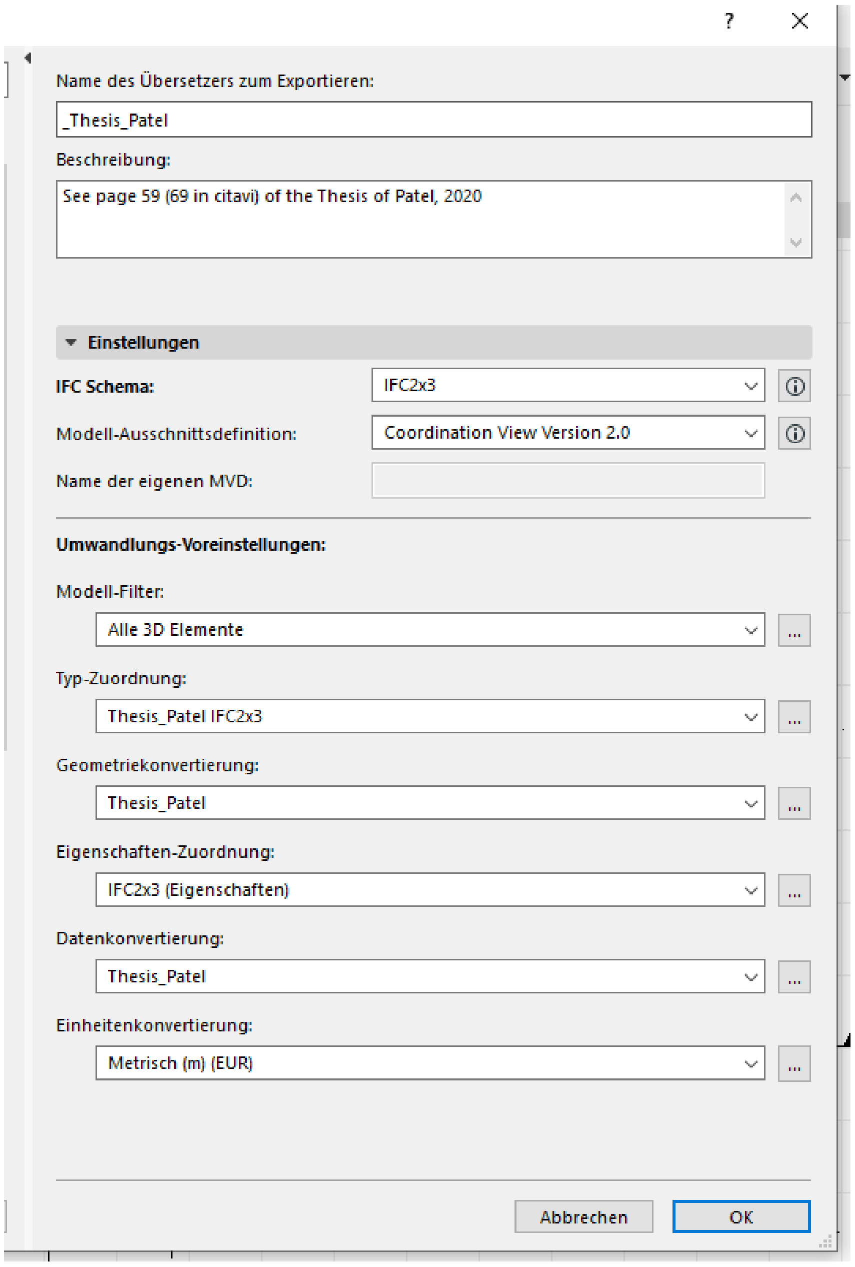


Figure A-6, ArchiCAD vertical section

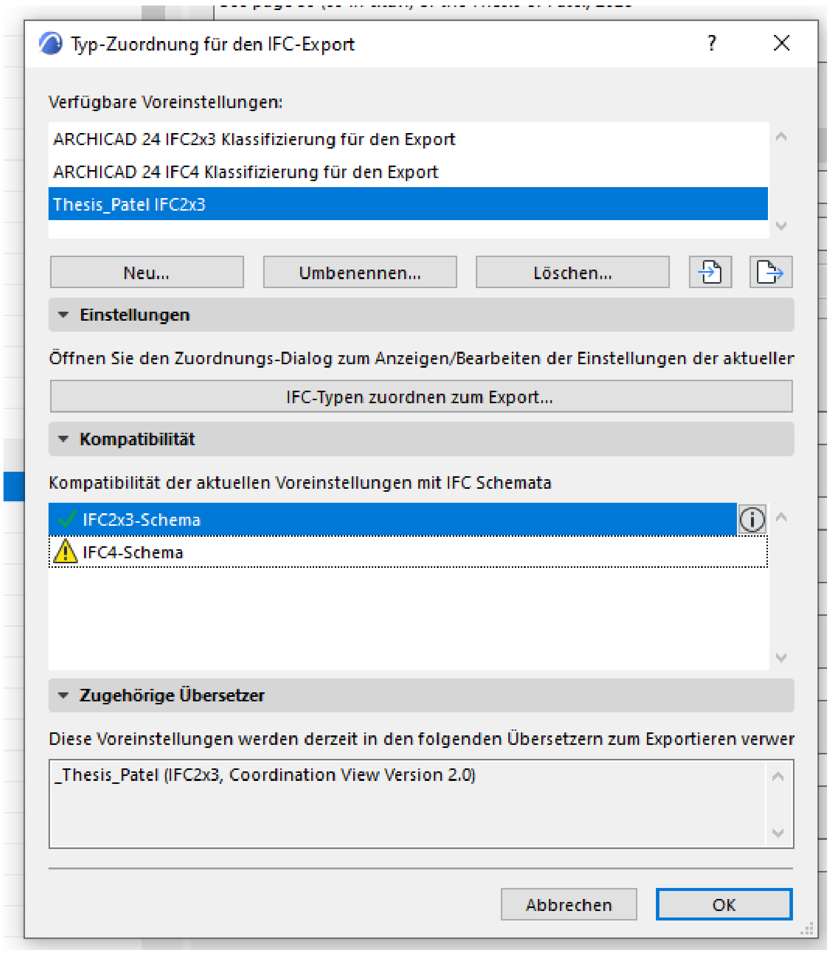
A.1.2. IFC export settings


Name des Übersetzers zum Exportieren:
_Thesis_Patel
Beschreibung:
See page 59 (69 in citavi) of the Thesis of Patel, 2020

IFC Schema: IFC2x3
Modell-Ausschnittsdefinition: Coordination View Version 2.0
Name der eigenen MVD:

Umwandlungs-Voreinstellungen:
Modell-Filter: Alle 3D Elemente
Typ-Zuordnung: Thesis_Patel IFC2x3
Geometriekonvertierung: Thesis_Patel
Eigenschaften-Zuordnung: IFC2x3 (Eigenschaften)
Datenkonvertierung: Thesis_Patel
Einheitenkonvertierung: Metrisch (m) (EUR)

Figure A-7, ArchiCAD IFC export settings part 1



Typ-Zuordnung für den IFC-Export

Verfügbare Voreinstellungen:

ARCHICAD 24 IFC2x3 Klassifizierung für den Export
ARCHICAD 24 IFC4 Klassifizierung für den Export
Thesis_Patel IFC2x3

Öffnen Sie den Zuordnungs-Dialog zum Anzeigen/Bearbeiten der Einstellungen der aktueller

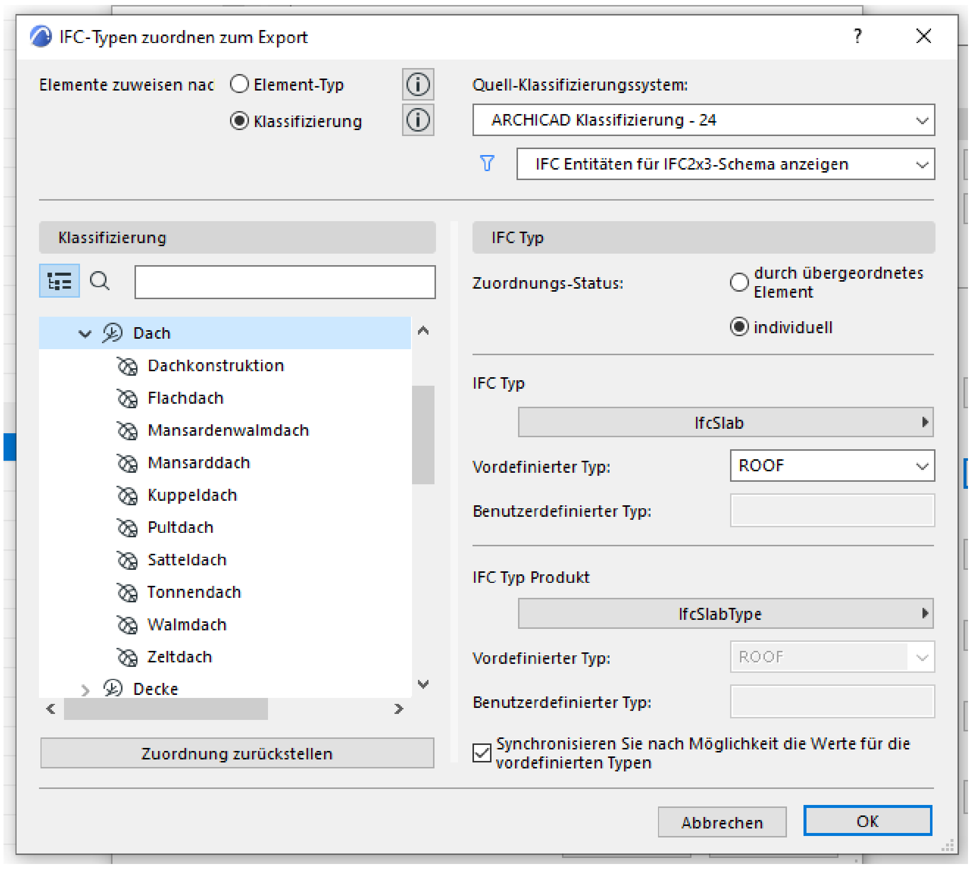
Kompatibilität der aktuellen Voreinstellungen mit IFC Schemata

✓ IFC2x3-Schema
⚠ IFC4-Schema

Diese Voreinstellungen werden derzeit in den folgenden Übersetzern zum Exportieren verwer

_Thesis_Patel (IFC2x3, Coordination View Version 2.0)

Figure A-8, ArchiCAD IFC export settings part 2


IFC-Typen zuordnen zum Export

Elemente zuweisen nac: Element-Typ [✓ Klassifizierung]

Quell-Klassifizierungssystem: ARCHICAD Klassifizierung - 24

IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung

- Dach
  - Dachkonstruktion
  - Flachdach
  - Mansardenwalmdach
  - Mansarddach
  - Kuppeldach
  - Pultdach
  - Satteldach
  - Tonnendach
  - Walmdach
  - Zeltdach
- Decke

Zuordnung zurückstellen

IFC Typ

Zuordnungs-Status: durch übergeordnetes Element [✓ individuell]

IFC Typ: IfcSlab

Vordefinierter Typ: ROOF

Benutzerdefinierter Typ:

IFC Typ Produkt: IfcSlabType

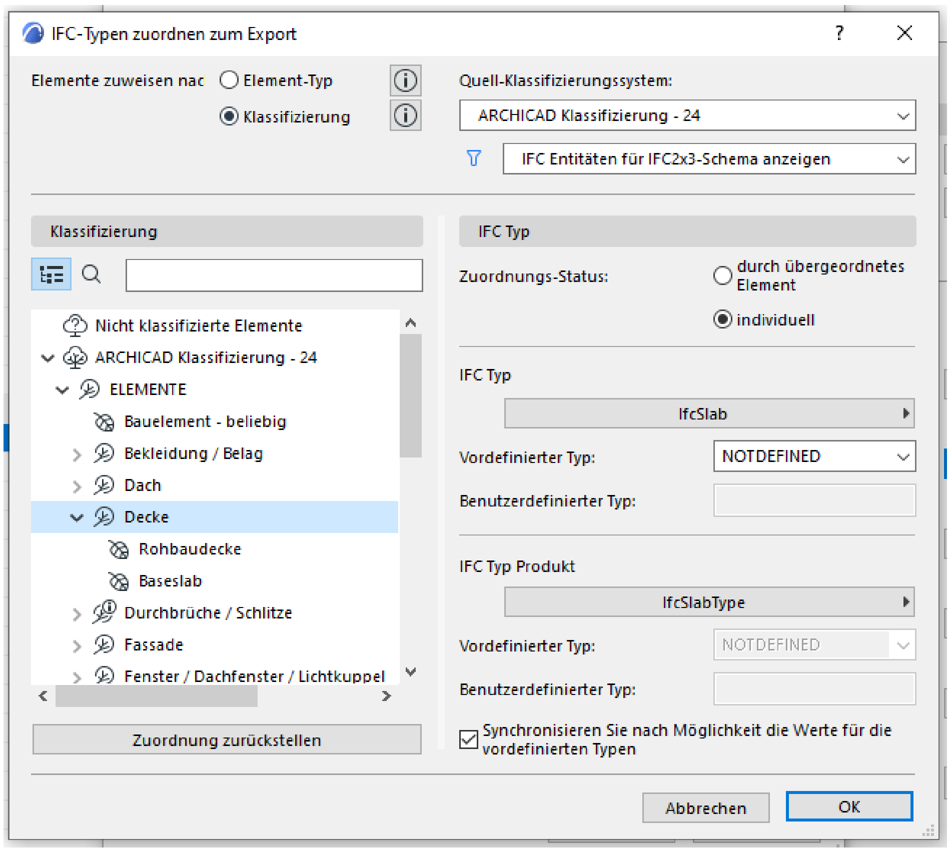
Vordefinierter Typ: ROOF

Benutzerdefinierter Typ:

☑ Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Abbrechen | OK

Figure A-9, ArchiCAD IFC export settings part 3


IFC-Typen zuordnen zum Export

Elemente zuweisen nac: Element-Typ [✓ Klassifizierung]

Quell-Klassifizierungssystem: ARCHICAD Klassifizierung - 24

IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung

- Nicht klassifizierte Elemente
- ARCHICAD Klassifizierung - 24
  - ELEMENTE
    - Bauelement - beliebig
    - Bekleidung / Belag
    - Dach
    - Decke
      - Rohbaudecke
      - Baseslab
    - Durchbrüche / Schlitze
    - Fassade
    - Fenster / Dachfenster / Lichtkuppel

Zuordnung zurückstellen

IFC Typ

Zuordnungs-Status: durch übergeordnetes Element [✓ individuell]

IFC Typ: IfcSlab

Vordefinierter Typ: NOTDEFINED

Benutzerdefinierter Typ:

IFC Typ Produkt: IfcSlabType

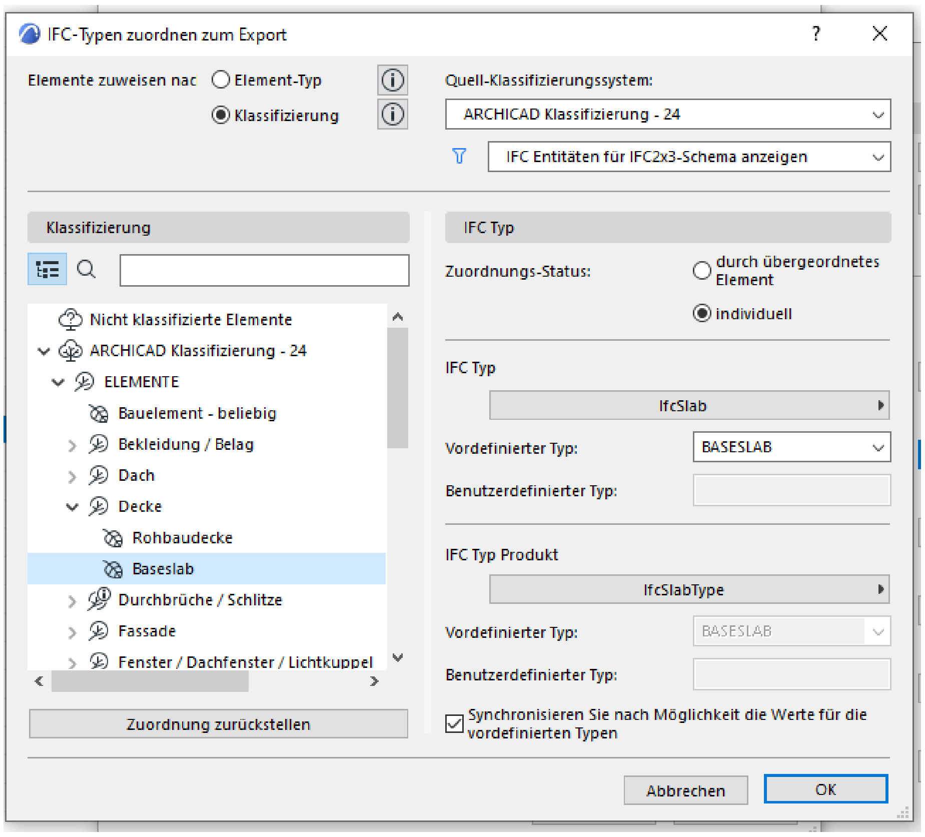
Vordefinierter Typ: NOTDEFINED

Benutzerdefinierter Typ:

☑ Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Abbrechen | OK

Figure A-10, ArchiCAD IFC export settings part 4



IFC-Typen zuordnen zum Export

Elemente zuweisen nac	Element-Typ [✓ Klassifizierung]
Quell-Klassifizierungssystem:	ARCHICAD Klassifizierung - 24
	IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung

- Nicht klassifizierte Elemente
- ARCHICAD Klassifizierung - 24
  - ELEMENTE
    - Bauelement - beliebig
    - Bekleidung / Belag
    - Dach
    - Decke
      - Rohbaudecke
      - **Baseslab**
    - Durchbrüche / Schlitze
    - Fassade
    - Fenster / Dachfenster / Lichtkuppel

Zuordnung zurückstellen

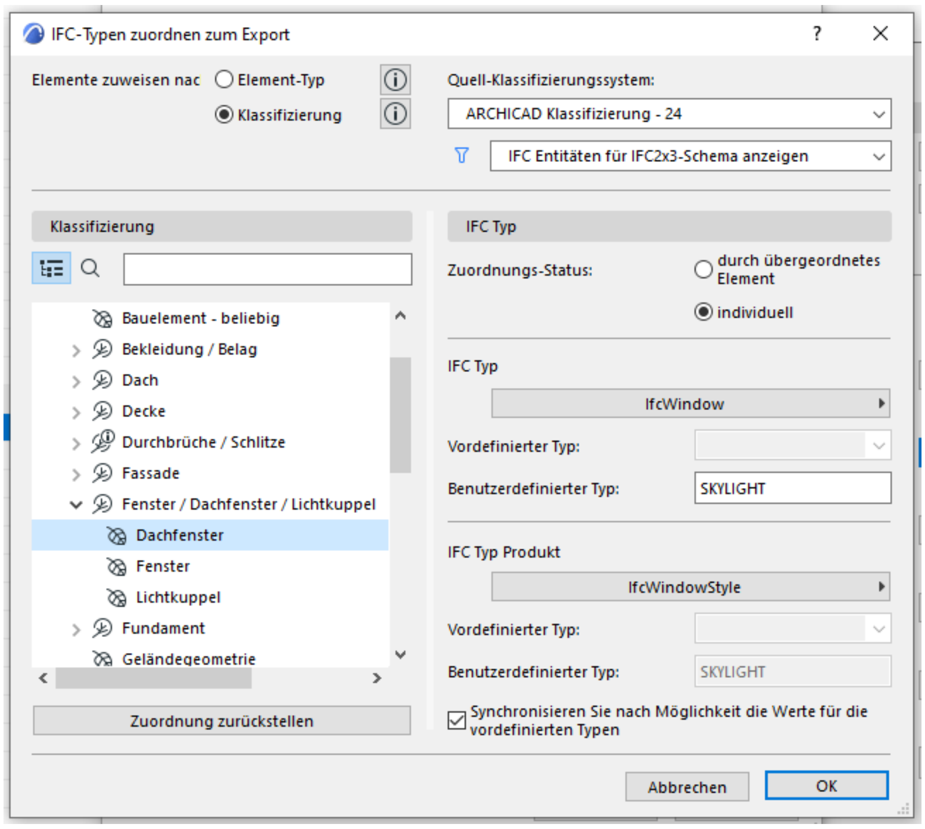
IFC Typ

Zuordnungs-Status:	durch übergeordnetes Element [✓ individuell]
IFC Typ	IfcSlab
Vordefinierter Typ:	BASESLAB
Benutzerdefinierter Typ:	
IFC Typ Produkt	IfcSlabType
Vordefinierter Typ:	BASESLAB
Benutzerdefinierter Typ:	

[✓] Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Abbrechen OK

Figure A-11, ArchiCAD IFC export settings part 5



IFC-Typen zuordnen zum Export

Elemente zuweisen nac	Element-Typ [✓ Klassifizierung]
Quell-Klassifizierungssystem:	ARCHICAD Klassifizierung - 24
	IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung

- Bauelement - beliebig
- Bekleidung / Belag
- Dach
- Decke
- Durchbrüche / Schlitze
- Fassade
- Fenster / Dachfenster / Lichtkuppel
  - **Dachfenster**
  - Fenster
  - Lichtkuppel
- Fundament
- Geländegeometrie

Zuordnung zurückstellen

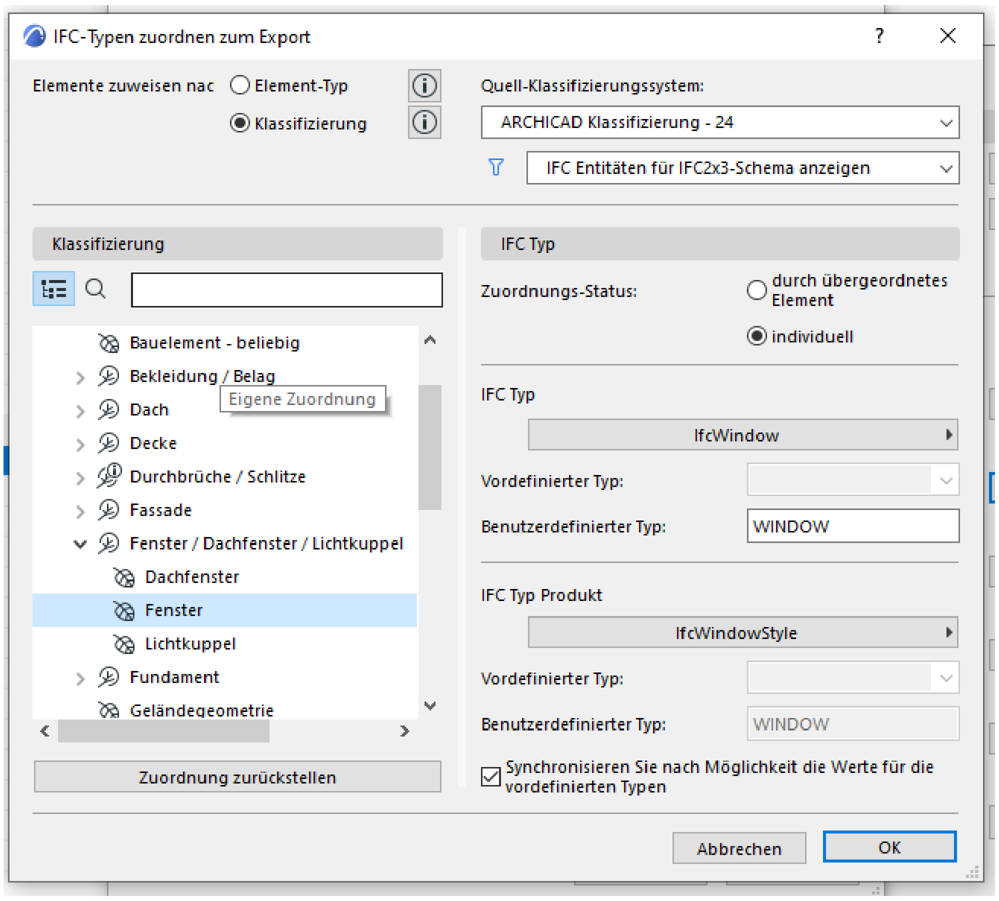
IFC Typ

Zuordnungs-Status:	durch übergeordnetes Element [✓ individuell]
IFC Typ	IfcWindow
Vordefinierter Typ:	
Benutzerdefinierter Typ:	SKYLIGHT
IFC Typ Produkt	IfcWindowStyle
Vordefinierter Typ:	
Benutzerdefinierter Typ:	SKYLIGHT

[✓] Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Abbrechen OK

Figure A-12, ArchiCAD IFC export settings part 6


IFC-Typen zuordnen zum Export

Elemente zuweisen nac ○ Element-Typ [✓ ● Klassifizierung]

Quell-Klassifizierungssystem: ARCHICAD Klassifizierung - 24

IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung
- Bauelement - beliebig
- Bekleidung / Belag
- Dach
- Decke
- Durchbrüche / Schlitze
- Fassade
- Fenster / Dachfenster / Lichtkuppel
  - Dachfenster
  - Fenster
  - Lichtkuppel
- Fundament
- Geländegeometrie

Eigene Zuordnung

IFC Typ

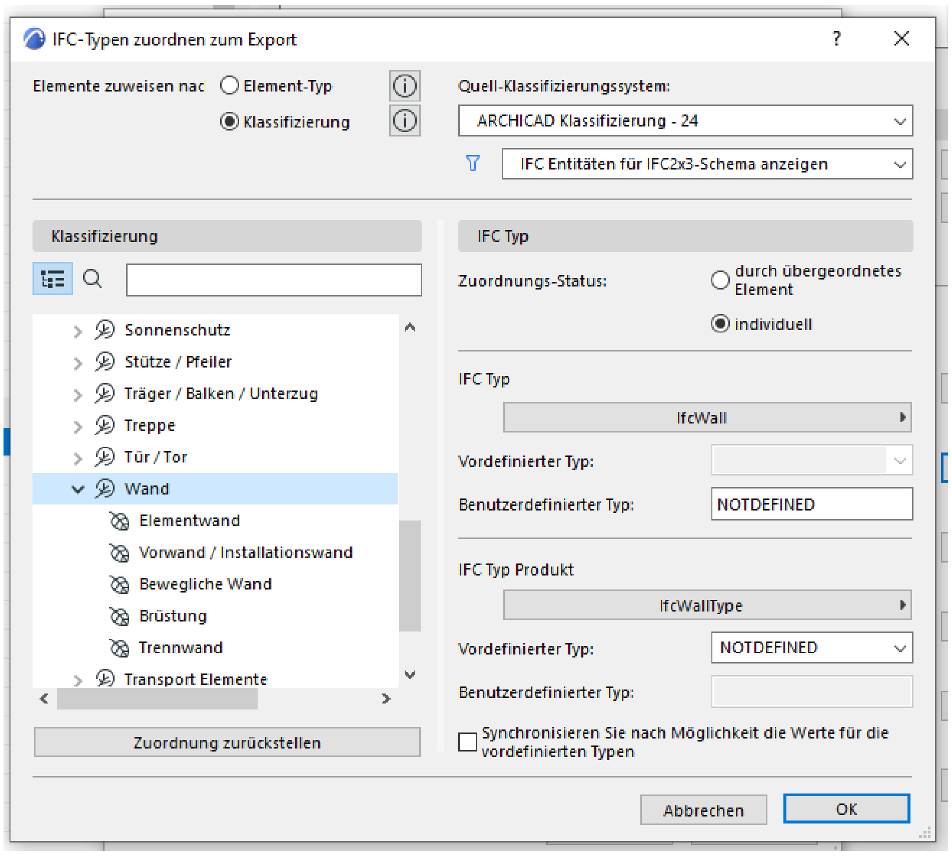
Zuordnungs-Status: ○ durch übergeordnetes Element [✓ ● individuell]

IFC Typ: IfcWindow
Vordefinierter Typ:
Benutzerdefinierter Typ: WINDOW

IFC Typ Produkt: IfcWindowStyle
Vordefinierter Typ:
Benutzerdefinierter Typ: WINDOW

☑ Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Figure A-13, ArchiCAD IFC export settings part 7


IFC-Typen zuordnen zum Export

Elemente zuweisen nac ○ Element-Typ [✓ ● Klassifizierung]

Quell-Klassifizierungssystem: ARCHICAD Klassifizierung - 24

IFC Entitäten für IFC2x3-Schema anzeigen

Klassifizierung
- Sonnenschutz
- Stütze / Pfeiler
- Träger / Balken / Unterzug
- Treppe
- Tür / Tor
- Wand
  - Elementwand
  - Vorwand / Installationswand
  - Bewegliche Wand
  - Brüstung
  - Trennwand
- Transport Elemente

IFC Typ

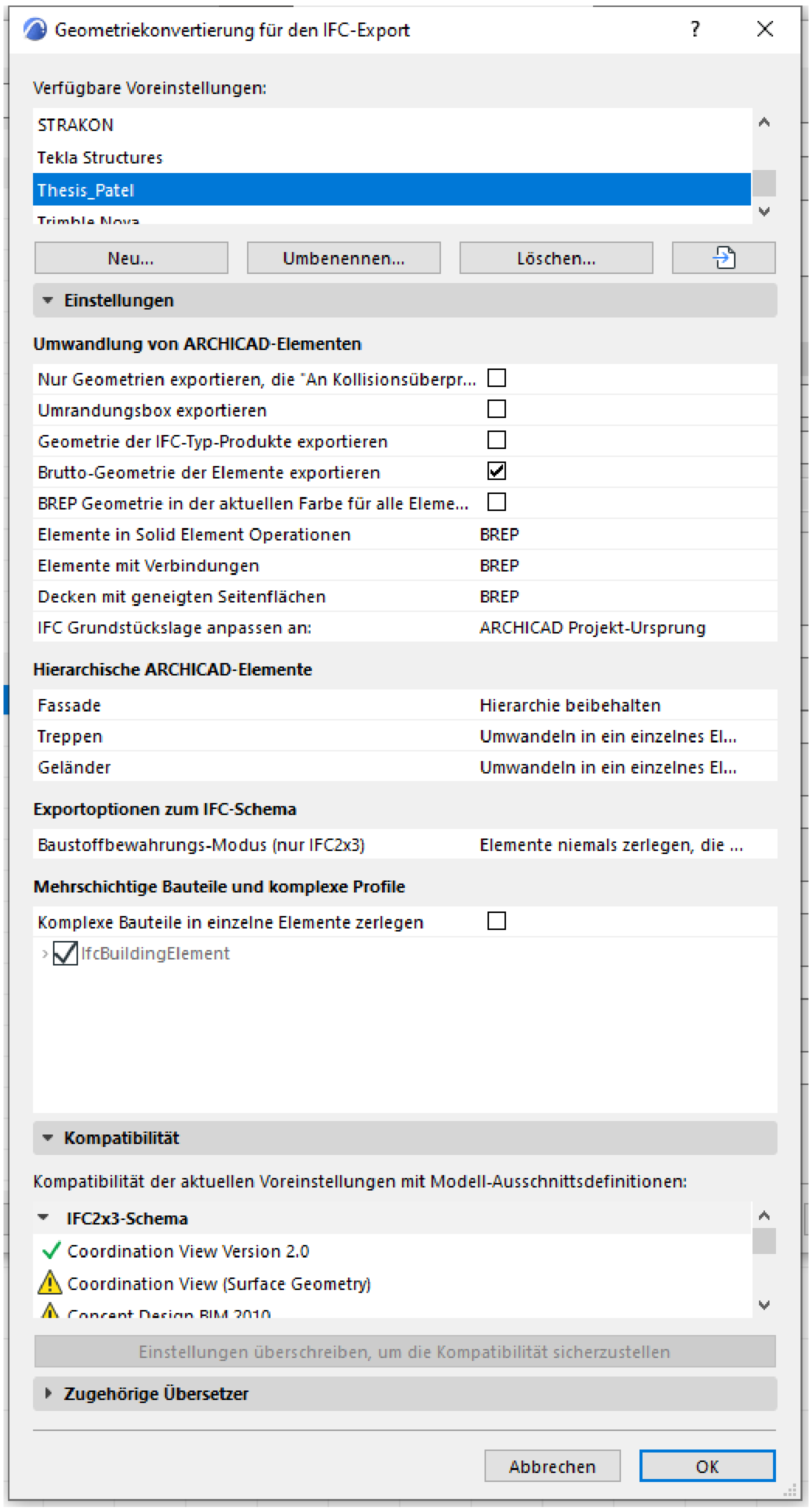
Zuordnungs-Status: ○ durch übergeordnetes Element [✓ ● individuell]

IFC Typ: IfcWall
Vordefinierter Typ:
Benutzerdefinierter Typ: NOTDEFINED

IFC Typ Produkt: IfcWallType
Vordefinierter Typ: NOTDEFINED
Benutzerdefinierter Typ:

☐ Synchronisieren Sie nach Möglichkeit die Werte für die vordefinierten Typen

Figure A-14, ArchiCAD IFC export settings part 8



Geometriekonvertierung für den IFC-Export

Verfügbare Voreinstellungen:

- STRAKON
- Tekla Structures
- Thesis_Patel
- Trimble Nova

Umwandlung von ARCHICAD-Elementen

Nur Geometrien exportieren, die "An Kollisionsüberpr...	☐
Umrandungsbox exportieren	☐
Geometrie der IFC-Typ-Produkte exportieren	☐
Brutto-Geometrie der Elemente exportieren	☑
BREP Geometrie in der aktuellen Farbe für alle Eleme...	☐
Elemente in Solid Element Operationen	BREP
Elemente mit Verbindungen	BREP
Decken mit geneigten Seitenflächen	BREP
IFC Grundstückslage anpassen an:	ARCHICAD Projekt-Ursprung

Hierarchische ARCHICAD-Elemente

Fassade	Hierarchie beibehalten
Treppen	Umwandeln in ein einzelnes El...
Geländer	Umwandeln in ein einzelnes El...

Exportoptionen zum IFC-Schema

Baustoffbewahrungs-Modus (nur IFC2x3)	Elemente niemals zerlegen, die ...

Mehrschichtige Bauteile und komplexe Profile

Komplexe Bauteile in einzelne Elemente zerlegen	☐

☑ IfcBuildingElement

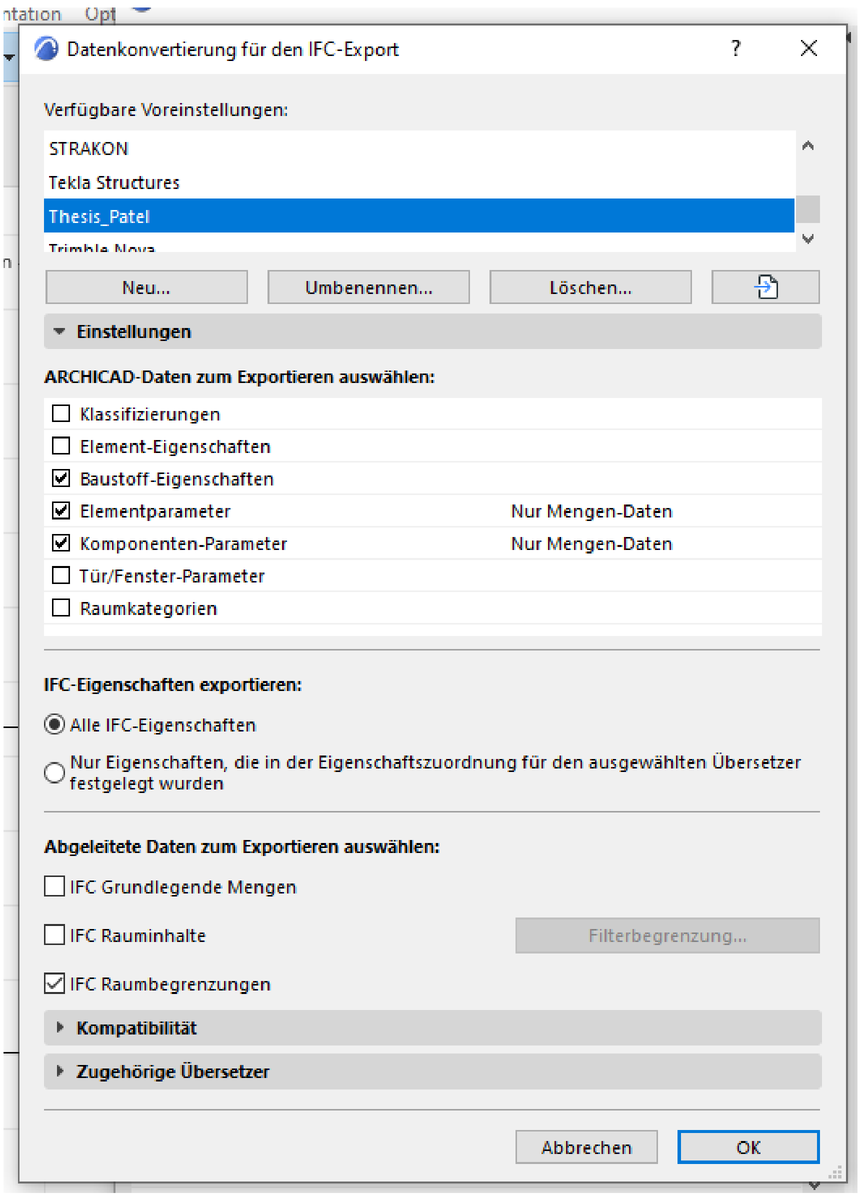
Kompatibilität

Kompatibilität der aktuellen Voreinstellungen mit Modell-Ausschnittsdefinitionen:

IFC2x3-Schema
- ✓ Coordination View Version 2.0
- ⚠ Coordination View (Surface Geometry)
- ⚠ Concept Design BIM 2010

Einstellungen überschreiben, um die Kompatibilität sicherzustellen
Zugehörige Übersetzer
Abbrechen
OK

Figure A-15, ArchiCAD IFC export settings part 9


Datenkonvertierung für den IFC-Export

Verfügbare Voreinstellungen:

STRAKON
Tekla Structures
Thesis_Patel
Trimble Nova

Einstellungen

ARCHICAD-Daten zum Exportieren auswählen:

☐ Klassifizierungen	
☐ Element-Eigenschaften	
☑ Baustoff-Eigenschaften	
☑ Elementparameter	Nur Mengen-Daten
☑ Komponenten-Parameter	Nur Mengen-Daten
☐ Tür/Fenster-Parameter	
☐ Raumkategorien	

IFC-Eigenschaften exportieren:

[✓ Alle IFC-Eigenschaften] Nur Eigenschaften, die in der Eigenschaftszuordnung für den ausgewählten Übersetzer festgelegt wurden

Abgeleitete Daten zum Exportieren auswählen:

☐ IFC Grundlegende Mengen
☐ IFC Rauminhalte
☑ IFC Raumbegrenzungen

Figure A-16, ArchiCAD IFC export settings part 10

A.2. Revit

A.2.1. Revit model

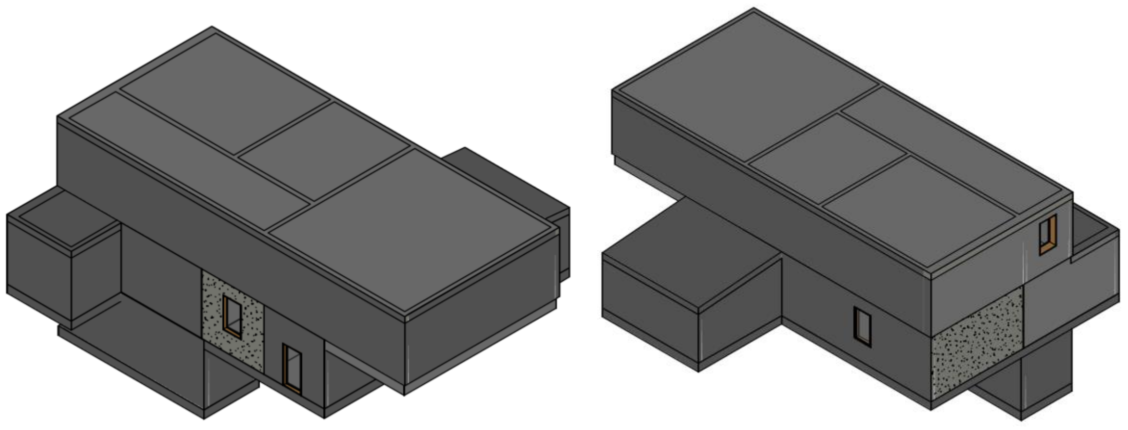


Figure A-17, Revit 3D views

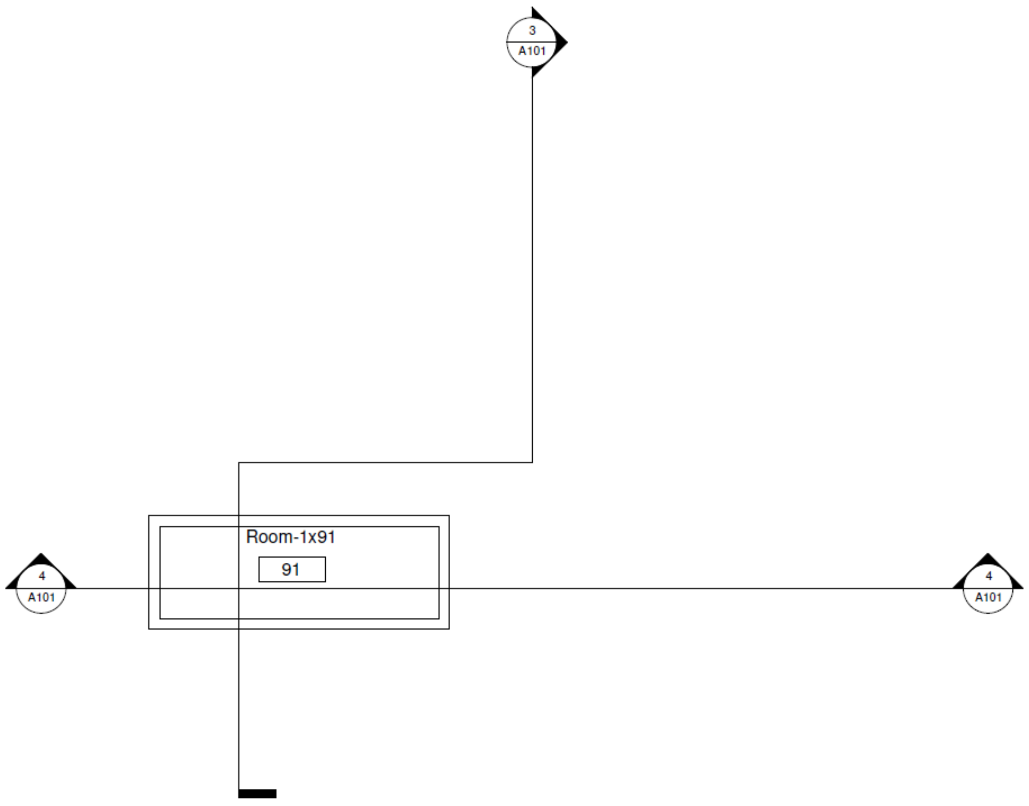


Figure A-18, Revit basement plan

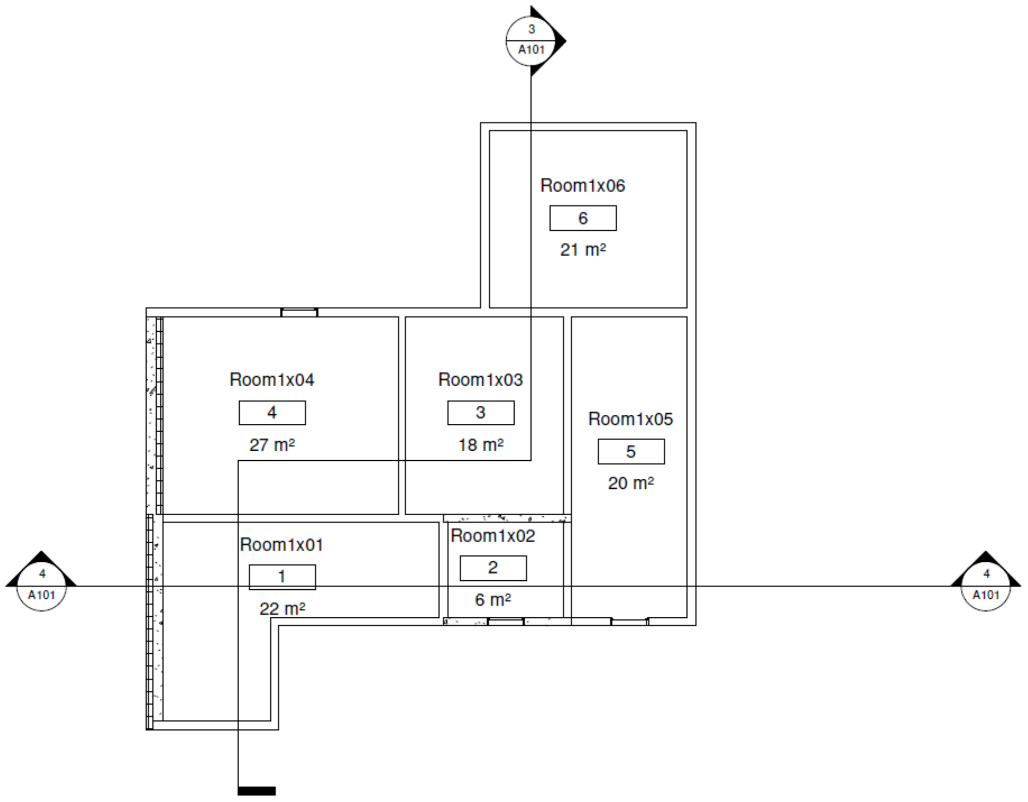


Figure A-19, Revit ground floor plan

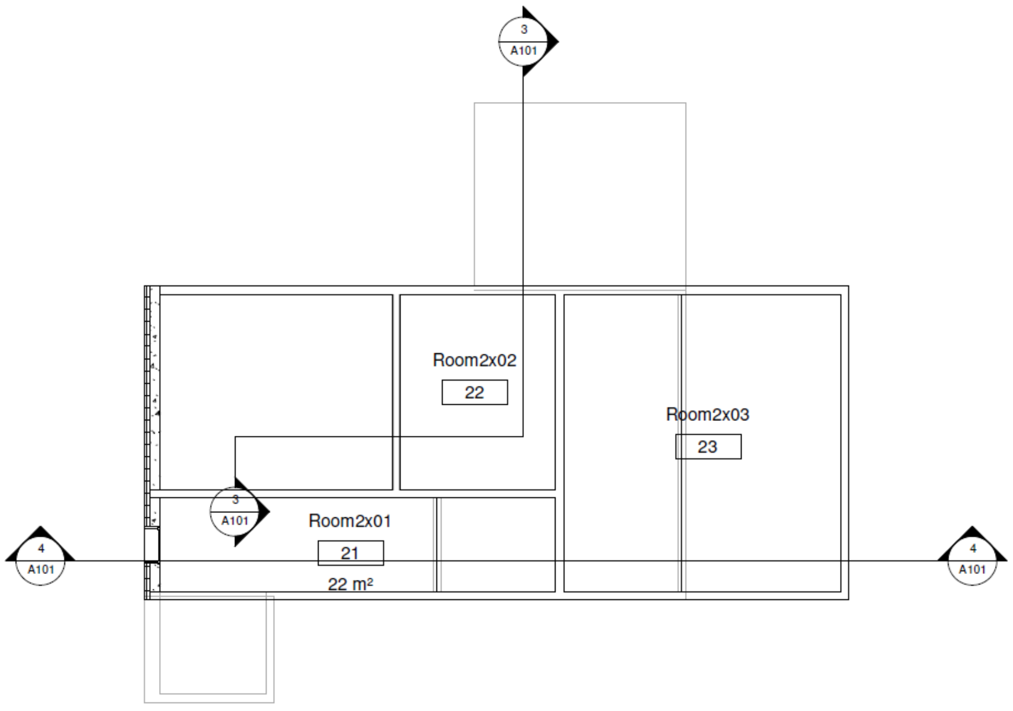


Figure A-20, Revit first floor plan

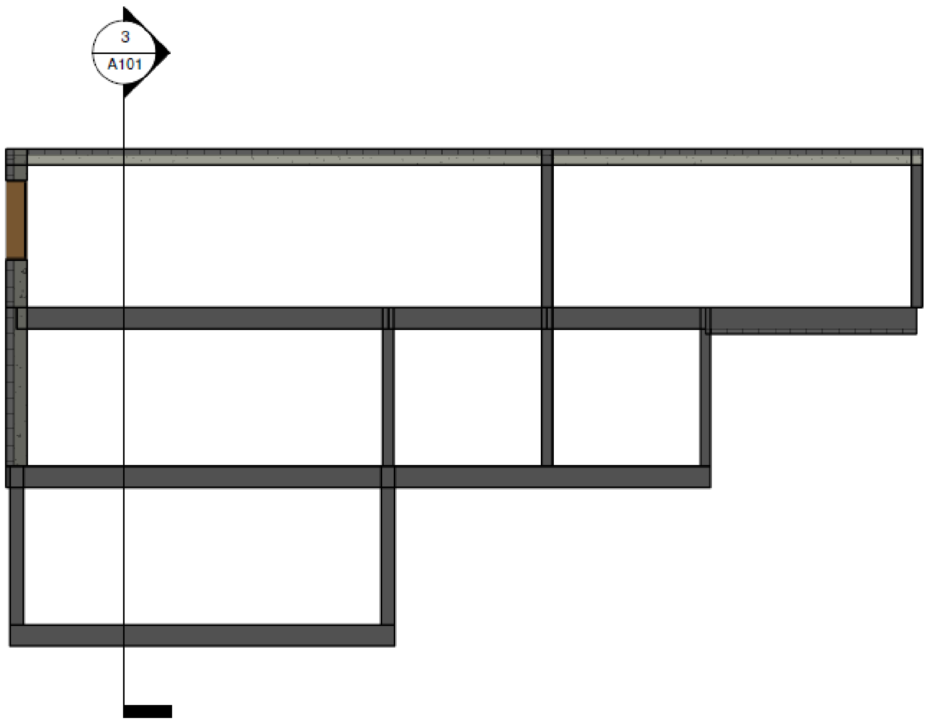


Figure A-21, Revit horizontal section

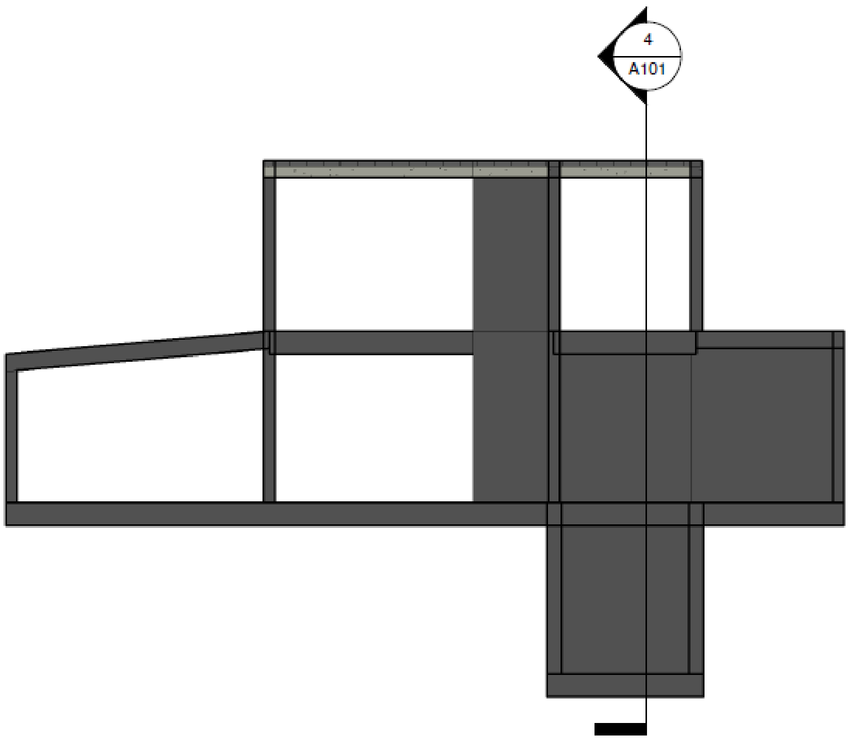
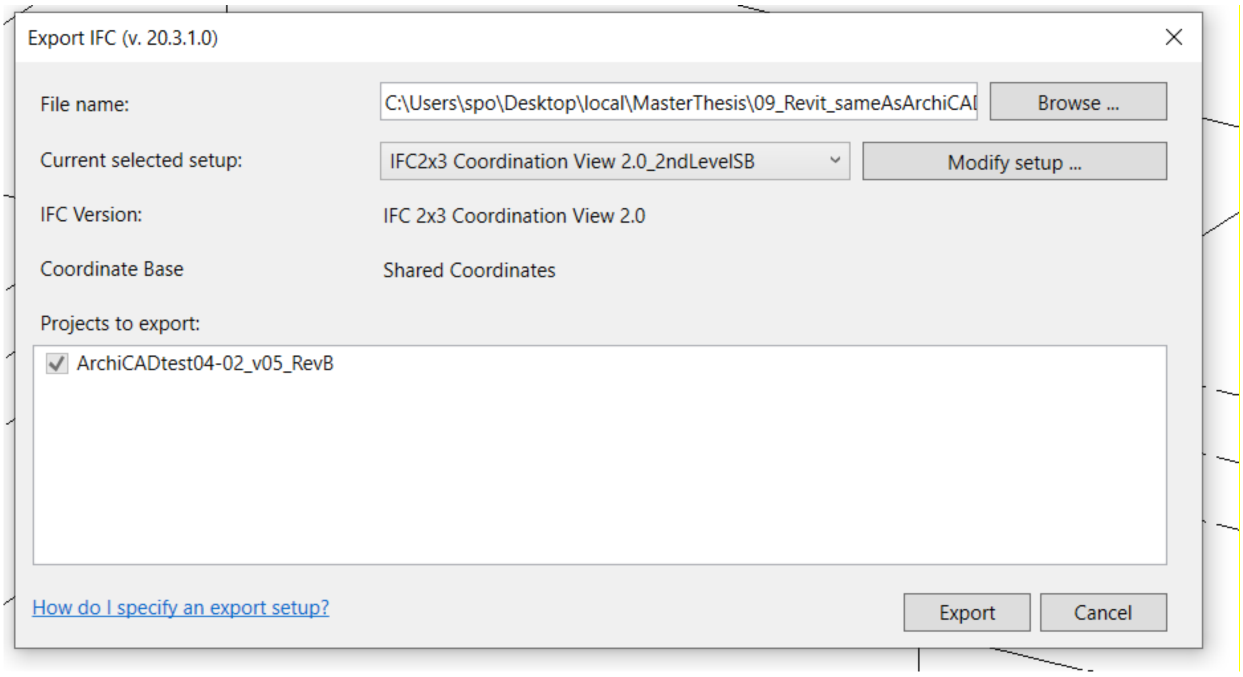


Figure A-22, Revit vertical section

A.2.2. IFC export settings



Export IFC (v. 20.3.1.0)

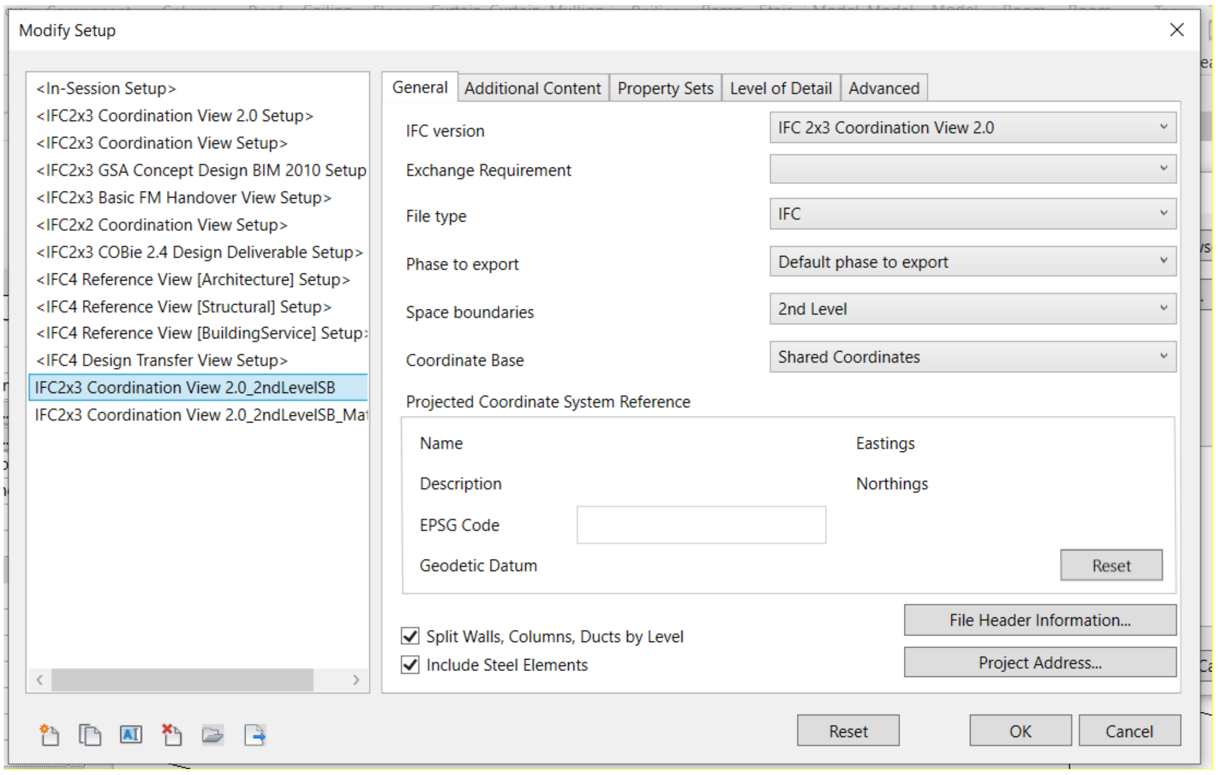
File name:	C:\Users\spo\Desktop\local\MasterThesis\09_Revit_sameAsArchiCAI
Current selected setup:	IFC2x3 Coordination View 2.0_2ndLevelSB
IFC Version:	IFC 2x3 Coordination View 2.0
Coordinate Base	Shared Coordinates

Projects to export:

☑ ArchiCADtest04-02_v05_RevB

How do I specify an export setup?

Figure A-23, Revit IFC export settings part 1



Modify Setup

<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x3 COBie 2.4 Design Deliverable Setup>
<IFC4 Reference View [Architecture] Setup>
<IFC4 Reference View [Structural] Setup>
<IFC4 Reference View [BuildingService] Setup
<IFC4 Design Transfer View Setup>
IFC2x3 Coordination View 2.0_2ndLevelSB
IFC2x3 Coordination View 2.0_2ndLevelSB_Ma

General | Additional Content | Property Sets | Level of Detail | Advanced

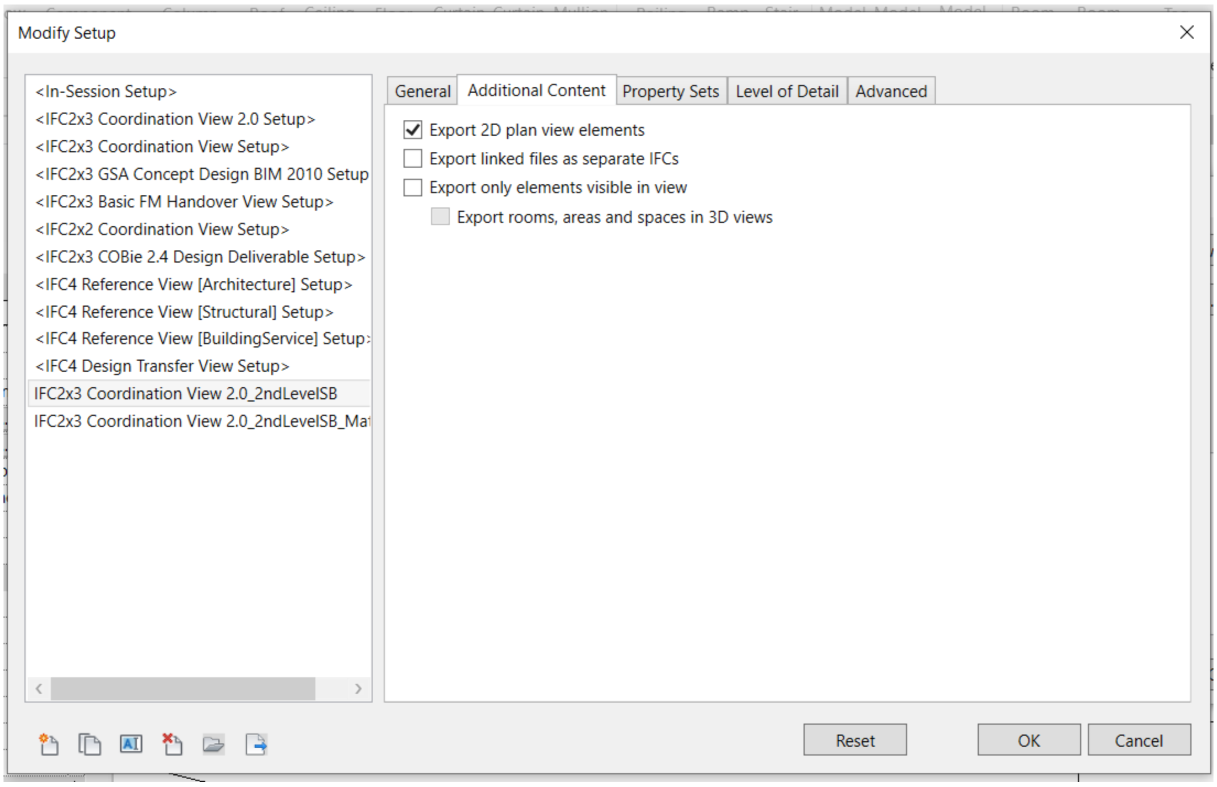
IFC version	IFC 2x3 Coordination View 2.0
Exchange Requirement	
File type	IFC
Phase to export	Default phase to export
Space boundaries	2nd Level
Coordinate Base	Shared Coordinates

Projected Coordinate System Reference

Name	Eastings
Description	Northings
EPSG Code	
Geodetic Datum	

☑ Split Walls, Columns, Ducts by Level
☑ Include Steel Elements

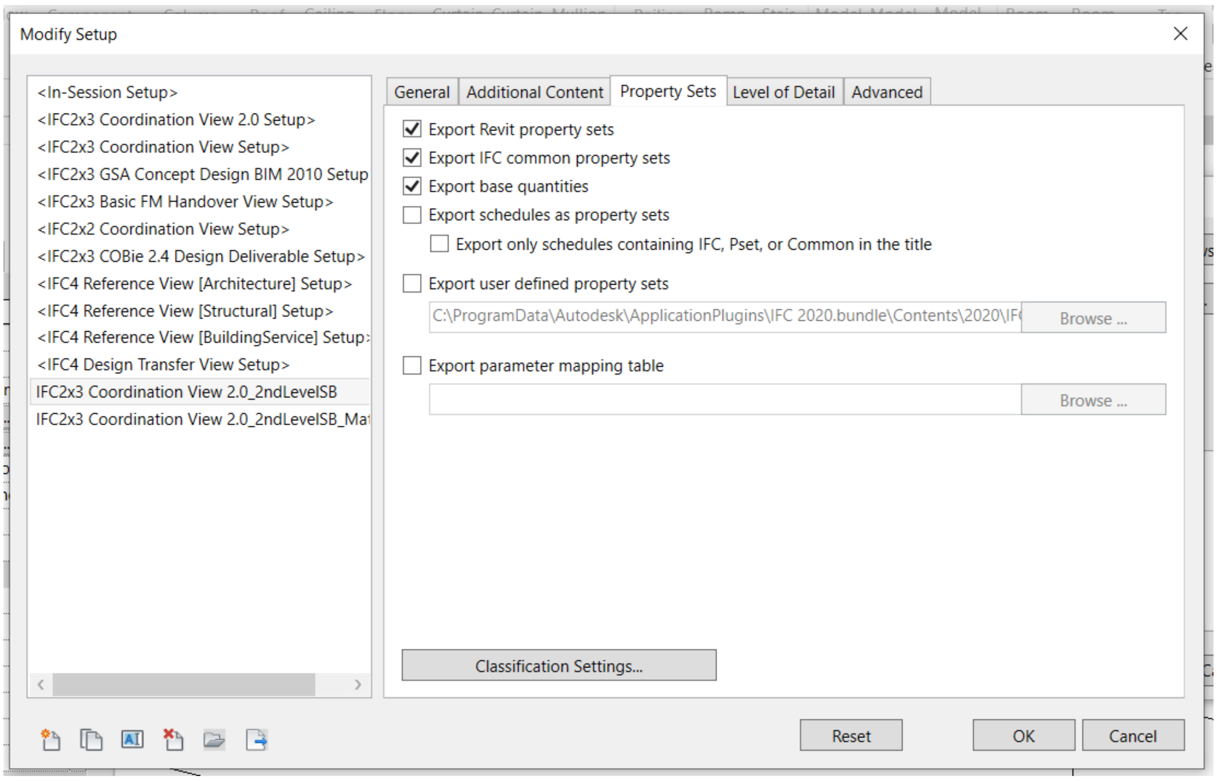
Figure A-24, Revit IFC export settings part 2


Modify Setup

<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x3 COBie 2.4 Design Deliverable Setup>
<IFC4 Reference View [Architecture] Setup>
<IFC4 Reference View [Structural] Setup>
<IFC4 Reference View [BuildingService] Setup
<IFC4 Design Transfer View Setup>
IFC2x3 Coordination View 2.0_2ndLevelSB
IFC2x3 Coordination View 2.0_2ndLevelSB_Ma

- ✓ Export 2D plan view elements
- ☐ Export linked files as separate IFCs
- ☐ Export only elements visible in view
- ☐ Export rooms, areas and spaces in 3D views

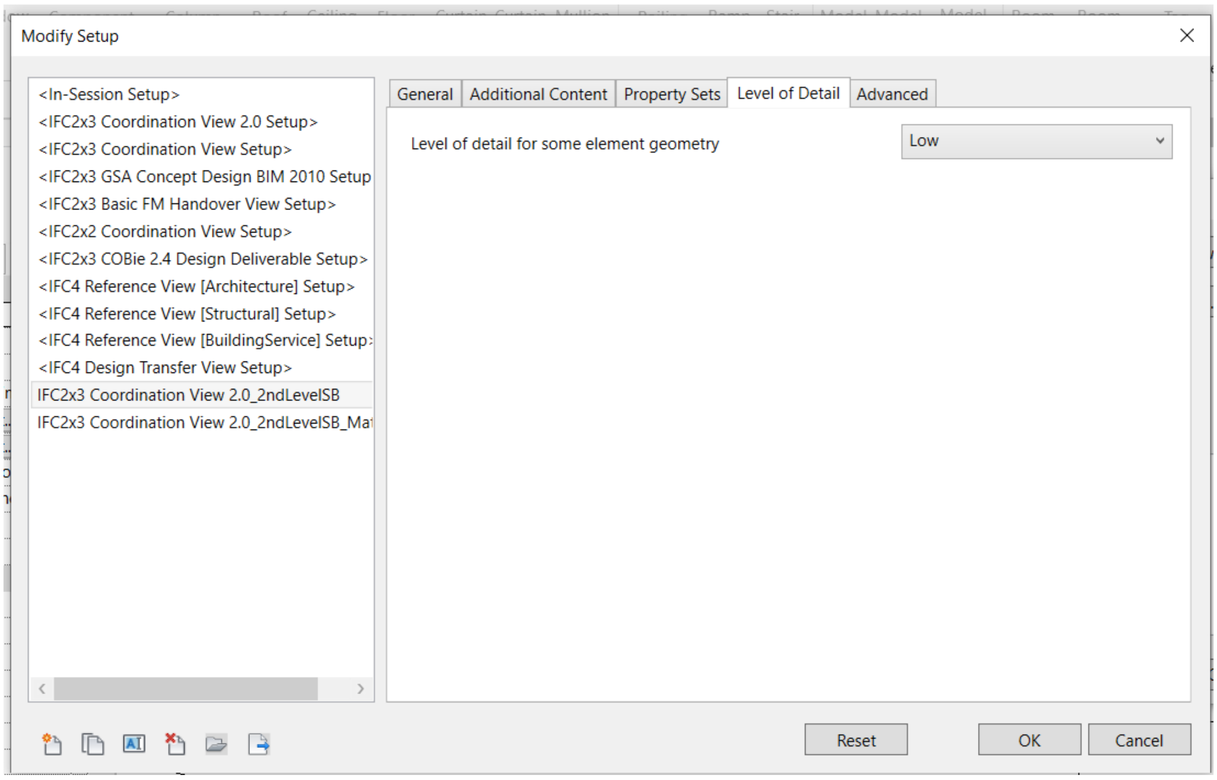
Figure A-25, Revit IFC export settings part 3


Modify Setup

<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x3 COBie 2.4 Design Deliverable Setup>
<IFC4 Reference View [Architecture] Setup>
<IFC4 Reference View [Structural] Setup>
<IFC4 Reference View [BuildingService] Setup
<IFC4 Design Transfer View Setup>
IFC2x3 Coordination View 2.0_2ndLevelSB
IFC2x3 Coordination View 2.0_2ndLevelSB_Ma

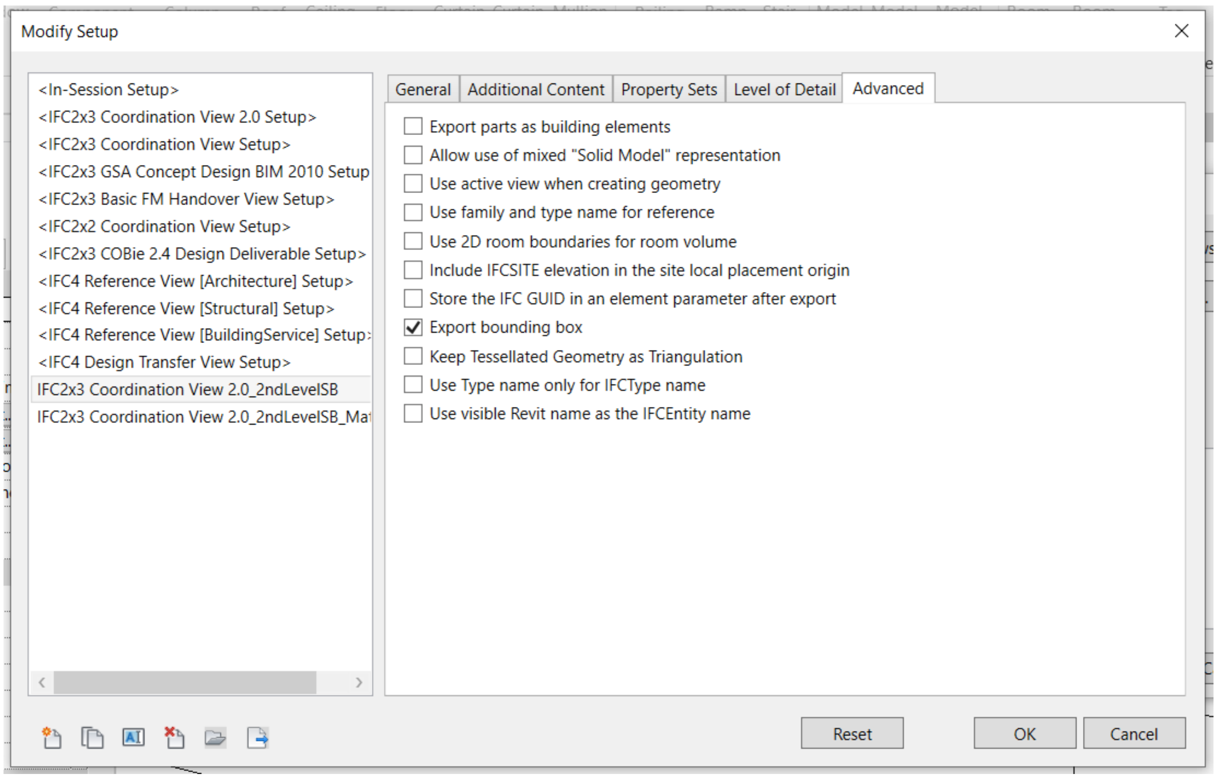
- ✓ Export Revit property sets
- ✓ Export IFC common property sets
- ✓ Export base quantities
- ☐ Export schedules as property sets
  - ☐ Export only schedules containing IFC, Pset, or Common in the title
- ☐ Export user defined property sets
  C:\ProgramData\Autodesk\ApplicationPlugins\IFC 2020.bundle\Contents\2020\IF
- ☐ Export parameter mapping table

Figure A-26, Revit IFC export settings part 4


Modify Setup
<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x3 COBie 2.4 Design Deliverable Setup>
<IFC4 Reference View [Architecture] Setup>
<IFC4 Reference View [Structural] Setup>
<IFC4 Reference View [BuildingService] Setup
<IFC4 Design Transfer View Setup>
IFC2x3 Coordination View 2.0_2ndLevelSB
IFC2x3 Coordination View 2.0_2ndLevelSB_Ma
General
Additional Content
Property Sets
Level of Detail
Advanced
Reset
OK
Cancel

Level of detail for some element geometry: Low

Figure A-27, Revit IFC export settings part 5


Modify Setup
<In-Session Setup>
<IFC2x3 Coordination View 2.0 Setup>
<IFC2x3 Coordination View Setup>
<IFC2x3 GSA Concept Design BIM 2010 Setup
<IFC2x3 Basic FM Handover View Setup>
<IFC2x2 Coordination View Setup>
<IFC2x3 COBie 2.4 Design Deliverable Setup>
<IFC4 Reference View [Architecture] Setup>
<IFC4 Reference View [Structural] Setup>
<IFC4 Reference View [BuildingService] Setup
<IFC4 Design Transfer View Setup>
IFC2x3 Coordination View 2.0_2ndLevelSB
IFC2x3 Coordination View 2.0_2ndLevelSB_Ma
General
Additional Content
Property Sets
Level of Detail
Advanced
Reset
OK
Cancel

- [ ] Export parts as building elements
- [ ] Allow use of mixed "Solid Model" representation
- [ ] Use active view when creating geometry
- [ ] Use family and type name for reference
- [ ] Use 2D room boundaries for room volume
- [ ] Include IFCSITE elevation in the site local placement origin
- [ ] Store the IFC GUID in an element parameter after export
- [✓] Export bounding box
- [ ] Keep Tessellated Geometry as Triangulation
- [ ] Use Type name only for IFCType name
- [ ] Use visible Revit name as the IFCEntity name

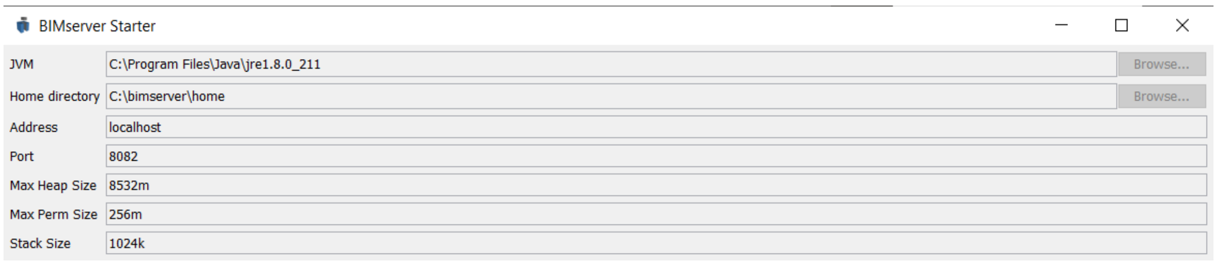
Figure A-28, Revit IFC export settings part 6

B. Software environment

Table B-1, software packages used for the case study

Software tool	add-on / library
ArchiCAD 24 (3008 AUT EDU)	
Revit 2020 (ENU)	IFC exporter v.20.3.1.0
FZKViewer. x64 v6.0	
EnergyPlus 9.2	
OpenStudio v2.9.1	
BIMserver 1.3.4	Java 1.8
	OsmSerializer 1.8
SketchUp Make 17.2.2555	OpenStudio add-on v2.9.1
	Euclid v0.9.4.2
Python 3.7.9	IfcOpenShell v0.6.0
	Open CASCADE 7.4.0
AutoCAD 2019	
Notepad++ v7.9.1	compare

C. BIMserver settings


BIMserver Starter

JVM	C:\Program Files\Java\jre1.8.0_211	Browse...
Home directory	C:\bimserver\home	Browse...
Address	localhost	
Port	8082	
Max Heap Size	8532m	
Max Perm Size	256m	
Stack Size	1024k	

Figure C-1, BIMserver settings

D. Table for added IDF class objects

Table D-1, added IDF class objects

IDF class object	Parameter	Value
Version	Version Identifier	9.2
SimulationControl	Do Zone Sizing Calculation	No
	Do System Sizing Calculation	No
	Do Plant Sizing Calculation	No
	Run Simulation for Sizing Periods	No
	Run Simulation for Weather File Run Periods	Yes
	Do HVAC Sizing Simulation for Sizing Periods	No
	Maximum Number of HVAC Sizing Simulation Passes	1
Timestep	Number of Timesteps per Hour	4
RunPeriod	Begin Month	01
	Begin Day of Month	01
	End Month	12
	End Day of Month	31
	Use Weather File Holidays and Special Days	No
	Use Weather File Daylight Saving Period	No
	Apply Weekend Holiday Rule	No
	Use Weather File Rain Indicators	No
	Use Weather File Snow Indicators	No
	Treat Weather as Actual	No

IDF class object	**Parameter**	**Value**
GlobalGeometryRules (already defined in OsmSerializer workflow)	Starting Vertex Position	LowerLeftCorner
	Vertex Entry Direction	Counterclockwise
	Coordinate System	Relative
	Daylight Reference Point Coordinate System	Relative
	Rectangular Surface Coordinate System	Relative
HVACTemplate:Thermostat	Constant Heating Setpoint	20
	Constant Cooling Setpoint	24
HVACTemplate:Zone: IdealLoadsAirSystem	Heating Limit	NoLimit
	Cooling Limit	NoLimit
	Dehumidification Control Type	None
	Humidification Control Type	None
	Outdoor Air Method	None
	Demand Controlled Ventilation Type	None
	Outdoor Air Economizer Type	None
	Heat Recovery Type	None

E. Python Scripts with marked changes

```python
"""BUILDING"""
import math
import ifcopenshell
def north(x, y):
    theta = math.degrees(math.atan2(y,x)) - 90
    return theta
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','w')
building = ifcfile.by_type('IfcProject')[0]
idf.write('Building,\n')
name = building.Name
idf.write('\t'+str(name)+',\n')
trueNorth = building.RepresentationContexts[0].TrueNorth.DirectionRatios
deg = north(trueNorth[0],trueNorth[1])
idf.write('\t'+str(deg)+';\n\n')
idf.close()


"""SITE:LOCATION"""
import ifcopenshell
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','a')
def dmsdd(d, m, s):
    dd = d + float(m)/60 + float(s)/3600
    return dd
site = ifcfile.by_type('IfcSite')
idf.write('Site:Location,\n')
for element in site:
        #Name = element.SiteAddress.Town ###changed
        Name = "Vienna" ###changed
        idf.write('\t'+str(Name)+',\n')
        a,b,c = element.RefLatitude[0:3]
        lat = dmsdd(a,b,c)
        idf.write('\t'+str(lat)+',\n')
        x,y,z = element.RefLongitude[0:3]
        long = dmsdd(x,y,z)
        idf.write('\t'+str(long)+',\n')
        idf.write('\t,\n')
        idf.write('\t'+str(element.RefElevation)+';\n\n')
idf.close()


"""ZONE"""
import ifcopenshell
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','a')
Spaces = ifcfile.by_type('IfcSpace')
for space in Spaces:
    idf.write('Zone,\n')
    idf.write('\t'+str(space.LongName)+',\n') #Name
    idf.write('\t,\n') #RelativeNorth
    #Origin
    idf.write('\t,\n')
    idf.write('\t,\n')
    idf.write('\t,\n')
    #Type and Multiplier
    idf.write('\t1,\n')
    idf.write('\t1,\n')
    #CeilingHeight,Volume,Area
    idf.write('\t,\n')
    spa = space.IsDefinedBy
    for sp in spa:
        if sp.is_a('IfcRelDefinesByProperties'):
            if sp.RelatingPropertyDefinition.is_a('IfcElementQuantity'):
                prop = sp.RelatingPropertyDefinition.Quantities
                for pro in prop:
#                   if pro.Name == 'Net Volume': ###changed
                    if pro.Name == 'Netto-Volumen':
                        volume = pro.VolumeValue
                        idf.write('\t'+str(volume)+',\n')
                for pro in prop:
#                   if pro.Name == 'Area': ###changed
```

```
                    if pro.Name == 'Fläche':
                        area = pro.AreaValue
                        idf.write('\t'+str(area)+',\n')
        elif sp.is_a('IfcRelDefinesByType'): ###changed
            pass
        else:
            idf.write('\t,\n')
            idf.write('\t,\n')
    #ConvectionAlgorithm
    idf.write('\t,\n')
    idf.write('\t,\n')
    idf.write('\tYes;\n\n') #PartOfFloorArea
idf.close()


"""04_MATERIAL"""
import ifcopenshell
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','a')
BuildingElement = ifcfile.by_type('IfcBuildingElement')
mList = [] ###changed
walls = ifcfile.by_type('IfcWall') ###changed
wall_x = walls[0] ###changed
mList = []
for element in BuildingElement:
    if element.is_a('IfcDoor') or element.is_a('IfcWindow'):
        continue
    material = element.IsDefinedBy
    mat_list=[]
    thick_list=[]
    for mat in material:
        if mat.is_a('IfcRelDefinesByProperties'):
            if mat.RelatingPropertyDefinition.Name == 'Material Properties':
                prop = mat.RelatingPropertyDefinition.HasProperties
                for compProp in prop:
                    mat_n = compProp.HasProperties
                    for x in mat_n:
                        matList = []
                        if x.Name not in mList:
#                           if x.Name.find('Air')!=-1: ###changed
                            if x.Name.find('Luft')!=-1: ###changed
                                matList.append('Material:AirGap')
                                matList.append(x.Name)
                                mList.append(x.Name)
                                xy = x.HasProperties
                                for y in xy:
                                    if y.Name == "ThermalConductivity":
                                        matList.append(y.NominalValue.wrappedValue)
                            else:
                                matList.append('Material')
                                matList.append(x.Name)
                                mList.append(x.Name)
#                               matList.append('\t,\n') #Roughness
                                matList.append('MediumRough') #Roughness ###changed
                                xy = x.HasProperties
                                for y in xy:
                                    if y.Name == "ThermalConductivity":
                                        matList.append(y.NominalValue.wrappedValue)
                                for y in xy:
                                    if y.Name == "MassDensity":
                                        matList.append(y.NominalValue.wrappedValue)
                                for y in xy:
                                    if y.Name == "SpecificHeatCapacity":
                                        matList.append(y.NominalValue.wrappedValue)
                        mat_list.append(matList)
            if mat.RelatingPropertyDefinition.Name == 'Component Quantities':
                thick = mat.RelatingPropertyDefinition.Quantities
                if element.is_a('IfcSlab'):
                    for compQ in thick:
                        thic = compQ.HasQuantities
                        for t in thic:
#                           if t.Name == 'Skin Thickness':
```

```
                            if t.Name == 'Schichtdicke': ###changed
                                thick_list.append(t.LengthValue)
                else:
                    for compQ in reversed(thick):
                        thic = compQ.HasQuantities
                        for t in thic:
#                           if t.Name == 'Skin Thickness':
                            if t.Name == 'Schichtdicke': ###changed
                                thick_list.append(t.LengthValue)
    for i in range(len(mat_list)):
        if mat_list[i] == []:
            continue
        else:
            li=list(mat_list[i])
#           if mat_list[i][1].find('Air') != -1:
            if mat_list[i][1].find('Luft') != -1: ###changed
                li[2] = thick_list[i]/mat_list[i][2]
            else:
                li.insert(3,thick_list[i])
            for j in range(len(li)):
                if j == len(li) - 1:
                    idf.write('\t'+str(li[j])+';\n\n')
                else:
                    idf.write('\t'+str(li[j])+',\n')
idf.close()


"""05_CONSTRUCTION"""
import ifcopenshell
def constr_layer(entity):
    construction = entity.HasAssociations[0].RelatingMaterial.Materials
    for layer in construction:
        layer_list.append(layer.Name)
    if element.is_a('IfcWall'):
        layer_list.reverse()
    for i in range(len(layer_list)):
        if i == len(layer_list) - 1:
            idf.write('\t'+str(layer_list[i])+';\n\n')
        else:
            idf.write('\t'+str(layer_list[i])+',\n')
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','a')
BuildingElement = ifcfile.by_type('IfcBuildingElement')
constr = []
for element in BuildingElement:
    if element.is_a('IfcDoor') or element.is_a('IfcWindow'):
        continue
    Name = element.IsDefinedBy

    for name in Name:
        if name.is_a('IfcRelDefinesByType'):
            if name.RelatingType.Name not in constr:
                idf.write('Construction,\n')
                constr.append(name.RelatingType.Name)
                idf.write('\t'+str(name.RelatingType.Name)+',\n')

                layer_list = []
                if element.is_a('IfcSlab'):
                    if element.PredefinedType == 'ROOF':
#                   if element.PredefinedType == 'ROOF' or element.PredefinedType == 
'BASESLAB': ###changed
                        construction = element.HasAssociations[0].RelatingMaterial.
                        ForLayerSet.MaterialLayers

                        material = []
                        for mat in construction:
                            material.append(mat.Material.Name)

                        for i in range(len(material)):
                            if i == len(material) - 1:
                                idf.write('\t'+str(material[i])+';\n\n')
                            else:
```

```python
                                idf.write('\t'+str(material[i])+',\n')
                    else:
                        constr_layer(element)
                else:
                    constr_layer(element)
idf.close()


"""BUILDINGSURFACE AND SHADING"""
import ifcopenshell
import ifcopenshell.geom
from OCC.Core import TopoDS
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)
def global_coord(entity):
    elem = ifcopenshell.geom.create_shape(settings, entity)
    shape = TopoDS.TopoDS_Iterator(elem.geometry).Value()
    transform = shape.Location().Transformation().TranslationPart().Coord()

    x1,y1,z1 = transform[0],transform[1],transform[2]
    idf.write('\t'+str(x1)+','+str(y1)+','+str(z1)+',\n')
    return x1,y1,z1
def slabDim(entity):
    Length = () ###changed
    Width = () ###changed
    for props in entity.IsDefinedBy:
        if props.is_a('IfcRelDefinesByProperties'):
            if props.RelatingPropertyDefinition.is_a('IfcPropertySet'):
                prop = props.RelatingPropertyDefinition.HasProperties
                for pro in prop:
                    if pro.Name == 'Length':
                        Length = pro.NominalValue.wrappedValue
                    if pro.Name == 'Width':
                        Width = pro.NominalValue.wrappedValue
                if Length != () and Width != (): ###changed
                    return Length,Width ###changed
def internal(entity):
    if entity.is_a('IfcSlab'):
#       if entity.ContainedInStructure[0].RelatingStructure.Name == 'Ground Floor':
        if entity.ContainedInStructure[0].RelatingStructure.Name == 'EG': ###changed
            idf.write('\t,\n')
        else:
            idf.write(str(entity.ProvidesBoundaries[0].RelatingSpace.LongName)+',\n')
    else:
        idf.write('\t'+str(entity.ProvidesBoundaries[-1].RelatingSpace.LongName)+
',\n')
    idf.write('\tNoSun,\n')
    idf.write('\tNoWind,\n')
def external(entity):
    idf.write('\tOutdoors,\n')
    idf.write('\t,\n')
    idf.write('\tSunExposed,\n')
    idf.write('\tWindExposed,\n')
def xpos():
    x2,y2,z2 = x1+Length, y1, z1
    x3,y3,z3 = x1+Length, y1, z1+Height
    x4,y4,z4 = x1, y1, z1+Height
    idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
    idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
    idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
def xneg():
    x2,y2,z2 = x1-Length, y1, z1
    x3,y3,z3 = x1-Length, y1, z1+Height
    x4,y4,z4 = x1, y1, z1+Height
    idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
    idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
    idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
def ypos():
    x2,y2,z2 = x1, y1+Length, z1
    x3,y3,z3 = x1, y1+Length, z1+Height
    x4,y4,z4 = x1, y1, z1+Height
    idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
    idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
    idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
def yneg():
    x2,y2,z2 = x1, y1-Length, z1
    x3,y3,z3 = x1, y1-Length, z1+Height
    x4,y4,z4 = x1, y1, z1+Height
    idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
    idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
    idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
ifcfile = ifcopenshell.open('thesis.ifc')
idf = open('ifcidf.idf','a')
BuildingElement = ifcfile.by_type('IfcBuildingElement')
for element in BuildingElement:
    #ShadingElement
    if element.ProvidesBoundaries == ():
        idf.write('Shading:Building,\n')
        idf.write('\t'+str(element.Name)+',\n')
        idf.write('\t,\n')
        idf.write('\t0,\n')
        global_coord(element)
        print(element)
        Length, Height = slabDim(element)
        idf.write('\t'+str(Length)+',\n')
        idf.write('\t'+str(Height)+';\n')
    else:
        #BuildingSurface or FenestrationSurface
        if element.is_a('IfcDoor') or element.is_a('IfcWindow'):
            continue
        else:
            idf.write('BuildingSurface:Detailed,\n')
        idf.write('\t'+str(element.Name)+',\n')
        #SurfaceType
        if element.is_a('IfcWall'):
            idf.write('\tWall,\n')
        else:
            if element.is_a('IfcSlab'):
                if element.PredefinedType == 'ROOF':
                    idf.write('\tRoof,\n')
                else:
                    idf.write('\tFloor,\n')
        #Construction
        Name = element.IsDefinedBy
        for name in Name:
            if name.is_a('IfcRelDefinesByType'):
                idf.write('\t'+str(name.RelatingType.Name)+',\n')
        #Zone
        if element.is_a('IfcWall'):
            Space = element.ProvidesBoundaries[0]
        else:
            Space = element.ProvidesBoundaries[-1]
        idf.write('\t'+str(Space.RelatingSpace.LongName)+',\n')
        #Outside Boundary Condition/Outside Boundary Condition Object/Sun and Wind Conditions
        if element.is_a('IfcSlab'):
            if element.PredefinedType != 'ROOF':
#               if element.ContainedInStructure[0].RelatingStructure.Name == 'Ground Floor':
                if element.ContainedInStructure[0].RelatingStructure.Name == 'EG': ###changed
                    idf.write('\tGround,\n')
                else:
                    idf.write('\tZone,\n')
                internal(element)
            else:
                external(element)
        else:
            if Space.InternalOrExternalBoundary == 'EXTERNAL':
                external(element)
            else:
                idf.write('\tZone,\n')
                internal(element)
        #View Factor to Ground
        if element.is_a('IfcWall'):
            idf.write('\t0.5,\n') 
        else: 
            if element.PredefinedType == 'ROOF':
                idf.write('\t0,\n') 
            else: 
                idf.write('\t1,\n')
        idf.write('\t,\n') #Number of Vertices
        #Vertices
        x1,y1,z1 = global_coord(element)
        if element.is_a('IfcSlab'):
#           print(element) ###changed
            Length, Width = slabDim(element)
            if element.PredefinedType != 'ROOF': 
                x2,y2,z2 = x1,y1+Width,z1
                x3,y3,z3 = x1+Length,y1+Width,z1 
                x4,y4,z4 = x1+Length,y1,z1
                idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
                idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
                idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
            else:
                x4,y4,z4 = x1,y1+Width,z1
                x3,y3,z3 = x1+Length,y1+Width,z1 
                x2,y2,z2 = x1+Length,y1,z1
                idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n') 
                idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n') 
                idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
        else: 
            for props in Name: 
                if props.is_a('IfcRelDefinesByProperties'): 
                    if props.RelatingPropertyDefinition.is_a('IfcElementQuantity'):
                        prop = props.RelatingPropertyDefinition.Quantities
                        for pro in prop: 
#                           if pro.Name == 'Length of Reference Line':
                            if pro.Name == 'Länge der Referenzlinie': ###changed
                                Length = pro.LengthValue 
#                           if pro.Name == 'Height':
                            if pro.Name == 'Höhe': ###changed
                                Height = pro.LengthValue

            WallDir = element.ObjectPlacement.RelativePlacement.RefDirection.DirectionRatios
            if WallDir == (1.0,0.0,0.0):
                xpos()
            elif WallDir == (-1.0,0.0,0.0):
                xneg()
            elif WallDir == (0.0,1.0,0.0):
                ypos()
            else:
                yneg()
idf.close()


"""FENESTRATIONSURFACE"""
import ifcopenshell
import ifcopenshell.geom 
# from OCC.Core import TopoDS ###changed
##from OCC import TopoDS ###changed
settings = ifcopenshell.geom.settings()
settings.set(settings.USE_PYTHON_OPENCASCADE, True)
ifcfile = ifcopenshell.open('thesis.ifc') 
idf = open('ifcidf.idf','a')
BuildingElement = ifcfile.by_type('IfcBuildingElement')
for element in BuildingElement:
    #BuildingSurface or FenestrationSurface
    if element.is_a('IfcWindow') or element.is_a('IfcDoor'): 
        idf.write('FenestrationSurface:Detailed,\n')
    else:
        continue
    idf.write('\t'+str(element.Name)+',\n')
    #SurfaceType
    if element.is_a('IfcWindow'):
        idf.write('\tWindow,\n')
    else:
        idf.write('\tDoor,\n')
#   idf.write('\t,\n') #Construction
    idf.write('\tSgl Clr 3mm,\n') #Construction  ###changed
    #Building Surface (Wall Relation)
    RelatedWall = element.FillsVoids[0].RelatingOpeningElement.VoidsElements[0].RelatingBuildingElement
    idf.write('\t'+str(RelatedWall.Name)+',\n')
    #Outside Boundary Condition Object
    space = element.ProvidesBoundaries[0]
    if space.InternalOrExternalBoundary == 'INTERNAL':
        RelSpace = element.ProvidesBoundaries[1]
        idf.write('\t'+str(RelSpace.RelatingSpace.LongName)+',\n')
    else:
        idf.write('\t,\n')
    idf.write('\t,\n') #View Factor to Ground
    idf.write('\t,\n') #Frame and Divider Name
    idf.write('\t1,\n') #Multiplier
    idf.write('\t,\n') #Number of Vertices
    #Vertices
    elem = ifcopenshell.geom.create_shape(settings, element)
    shape = TopoDS.TopoDS_Iterator(elem.geometry).Value()
    transform = shape.Location().Transformation().TranslationPart().Coord()
    x1,y1,z1 = transform[0],transform[1],transform[2]
    direction = RelatedWall.ObjectPlacement.RelativePlacement.RefDirection.DirectionRatios
    if direction == (0.0,-1.0,0.0):
        x1,y1,z1 = x1,y1+(element.OverallWidth/2),z1
        x2,y2,z2 = x1, y1-element.OverallWidth, z1
        x3,y3,z3 = x1, y2, z1+element.OverallHeight
        x4,y4,z4 = x1, y1, z1+element.OverallHeight
        idf.write('\t'+str(x1)+','+str(y1)+','+str(z1)+',\n')
        idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
        idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
        idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
    elif direction == (0.0,1.0,0.0):
        x1,y1,z1 = x1,y1-(element.OverallWidth/2),z1
        x2,y2,z2 = x1, y1+element.OverallWidth, z1
        x3,y3,z3 = x1, y2, z1+element.OverallHeight
        x4,y4,z4 = x1, y1, z1+element.OverallHeight
        idf.write('\t'+str(x1)+','+str(y1)+','+str(z1)+',\n')
        idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
        idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
        idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
    elif direction == (1.0,0.0,0.0):
        x1,y1,z1 = x1-(element.OverallWidth/2),y1,z1
        x2,y2,z2 = x1+element.OverallWidth, y1, z1
        x3,y3,z3 = x2, y1, z1+element.OverallHeight
        x4,y4,z4 = x1, y1, z1+element.OverallHeight
        idf.write('\t'+str(x1)+','+str(y1)+','+str(z1)+',\n')
        idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
        idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
        idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
    else:
        x1,y1,z1 = x1+(element.OverallWidth/2),y1,z1
        x2,y2,z2 = x1-element.OverallWidth, y1, z1
        x3,y3,z3 = x2, y1, z1+element.OverallHeight
        x4,y4,z4 = x1, y1, z1+element.OverallHeight
        idf.write('\t'+str(x1)+','+str(y1)+','+str(z1)+',\n')
        idf.write('\t'+str(x2)+','+str(y2)+','+str(z2)+',\n')
        idf.write('\t'+str(x3)+','+str(y3)+','+str(z3)+',\n')
        idf.write('\t'+str(x4)+','+str(y4)+','+str(z4)+';\n\n')
idf.close()


"""UTF-8 conversion of idf file"""
idf = open('ifcidf.idf')
content = idf.read()
idf.close()
#source: https://gist.github.com/johnberroa/cd49976220933a2c881e89b69699f2f7
def remove_umlaut(string):
    """
    Removes umlauts from strings and replaces them with the letter+e convention
    :param string: string to remove umlauts from
    :return: unumlauted string
    """
    u = 'ü'.encode()
    U = 'Ü'.encode()
    a = 'ä'.encode()
    A = 'Ä'.encode()
    o = 'ö'.encode()
    O = 'Ö'.encode()
    ss = 'ß'.encode()

    string = string.encode()
    string = string.replace(u, b'ue')
    string = string.replace(U, b'Ue')
    string = string.replace(a, b'ae')
    string = string.replace(A, b'Ae')
    string = string.replace(o, b'oe')
    string = string.replace(O, b'Oe')
    string = string.replace(ss, b'ss')

    string = string.decode('utf-8')
    return string
result = remove_umlaut(content)
idf = open('ifcidf.idf','w')
idf.write(result)
idf.close()
```