



Change selection and feedback communication of design variant decision using BIM

Scientific work to obtain the degree

Master of Science (M.Sc.)

at the Department of Civil, Geo and Environmental Engineering of the Technical University of Munich.

Supervised by Prof. Dr.-Ing. André Borrmann
Kasimir Forth, M.Sc.
Chair of Computational Modeling and Simulation

Submitted by Baris Omer Balota
e-Mail: baris.balota@tum.de

Submitted on 31. October 2022

Abstract

The building industry significantly impacts the environment and the economy. Incorporating Life Cycle Assessment (LCA) and Life Cycle Costing (LCC) into the decision-making process provides environmental and economic improvement for buildings. From this point of view, Building Information Modeling (BIM) offers a variety of strategies for integrating LCA and LCC studies into the building design. The following topics are essential to this integration process: visualizing the analysis results for different design options and reflecting the design decisions into the BIM model. These mentioned points determine the research topic of this master's thesis. Accordingly, the first part of this thesis deals with identifying different methods for visualizing the analysis results. In this regard, two chart groups were defined: *criterion-specific* and *overview* charts. In contrast to a *criterion-specific* chart, an *overview* chart combines various criteria (e.g., energy performance, cost, CO_2 emissions, etc.) into a single chart. The next step was identifying suitable chart types for each chart group. Furthermore, the second part of this thesis focuses on open BIM and its extension options to communicate design decisions to the BIM model. In this context, the BIM Collaboration Format was extended using BIM Snippets. Based on this, a semi-automated method for updating the BIM model according to design decisions has been introduced. The presented visualization methods enable stakeholders to compare different design options according to their economic and environmental impacts. In addition, the developed semi-automated method reduces the effort required to update the BIM model regarding design decisions.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goal and Scope	2
1.3	Outline	2
2	State of the Art	4
2.1	Building Information Modelling	4
2.1.1	Definition	4
2.1.2	Closed/Open BIM and Little/Big BIM	4
2.1.3	Level of Development	5
2.1.4	Data Exchange	6
2.1.5	Industry Foundation Classes (IFC)	7
2.1.6	BIM Collaboration Format (BCF)	11
2.2	Project Phases in Construction	13
2.3	Environmental and Cost Analyses	14
2.3.1	Life Cycle Assessment (LCA)	14
2.3.2	Life Cycle Costing (LCC)	16
2.3.3	BIM-based Decision-making Through LCA and LCC	17
3	A Method for Comparing and Selecting Design Variants	18
3.1	Research Questions	18
3.2	General Framework	19
3.3	Part I: Comparison of Design Variants	20
3.4	Part II: Design Decisions	24
3.4.1	Communicating Design Decisions	24
3.4.2	Updating BIM Models	25
4	Prototypical Implementation & Case Study	27
4.1	Technical Background: Web Application Development	27
4.1.1	HTML, CSS and JavaScript	27
4.1.2	Full-stack development	28
4.2	Prototypical Implementation	30
4.2.1	Back End	32
4.2.2	Front End	35
4.3	Case Study	43
4.3.1	Exporting IFC Model from Revit	43
4.3.2	Visualizing Model	43
4.3.3	Comparing Design Variants for Non-bearing External Walls	44
4.3.4	Design Decision-Making Process	45
4.3.5	Updating Model	46

5 Conclusion	48
5.1 Review of Research Questions	48
5.2 Limitations	49
5.3 Outlook	50
A Appendix: Project Phases	51
B Appendix: Development Packages	52
C Appendix: Comparison of Design Variants	54
C.1 Load-bearing External Walls	54
C.2 Non-load-bearing External Walls	56
Bibliography	58

List of Figures

1.1	The MacLeamy Curve (CONSTRUCTION USERS ROUNDTABLE, 2004, p. 4)	1
2.1	Different ways of implementing BIM (BORRMANN, KÖNIG, et al., 2018, p. 12)	5
2.2	Data models at different levels (SACKS et al., 2018, p. 90)	6
2.3	Layers of the IFC schema, adapted from Figure 1 from (BUILDINGSMART, 2017b)	9
2.4	Inheritance structure of <i>lfcWall</i> , adapted from the entity inheritance schema in Section 6.1.3.46.2 from (BUILDINGSMART, 2017b)	10
2.5	Adding custom properties to an <i>lfcDoor</i> object (BORRMANN, BEETZ, et al., 2018, p. 31)	11
2.6	General structure of BCF-XML (left) and BCF-API (right), adapted from (SCHULZ & BEETZ, 2021, p. 3) and (BUILDINGSMART, 2022)	12
2.7	LCA phases (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2006, p. 8)	14
2.8	Life cycle stages for a building assessment (EUROPEAN COMMITTEE FOR STANDARDIZATION, 2021, p. 30)	15
2.9	Framework for optimizing building designs based on multicriteria analyses (FORTH et al., 2021, p. 238)	17
3.1	General framework	19
3.2	The schema for determining the visualization category (HOLLBERG et al., 2021, p. 6)	20
3.3	The modified concept for cluster heat maps to compare design variants, adapted from (CERDAS et al., 2017, p. 735)	21
3.4	An example of a criterion-specific chart	22
3.5	An example of an overview chart	23
3.6	The concept of using the BCF with BIM Snippets for communicating design decisions	24
3.7	The <i>IFC_GUID</i> property of the building component matching with the <i>lfcGuid</i> of the BIM Snippet	26
4.1	The DOM of a web page (W3SCHOOLS, n.d.)	27
4.2	Interaction between the front end and the back end via HTTP	29
4.3	General schema of the prototypical implementation	30
4.4	Response from the server for analyzing design variants through LCA and LCC	33
4.5	Database structure, adapted from (SCHULZ, 2020)	34
4.6	UI elements in the web application before logging in and uploading an IFC model	35

4.7	UI elements in the web application after logging in and uploading an IFC model	36
4.8	Modal window showing the details of a BCF issue	37
4.9	UI for showing the analysis results	38
4.10	Modal window to create a BCF issue	39
4.11	The login page of the Revit plugin	40
4.12	The second page of the Revit plugin displayed upon logging in	40
4.13	Visualizing the IFC model in the web application	43
4.14	Comparison of design variants through LCA and LCC	44
4.15	The issue form for creating a new BCF issue	45
4.16	Updated Revit model	46
4.17	Importing the updated model into the web application	47
A.1	Project phases in different countries/regions (ROYAL INSTITUTE OF BRITISH ARCHITECTS, 2020, p. 9)	51
C.1	Analysis results - Price	54
C.2	Analysis results - GWP	54
C.3	Analysis results - Energy	55
C.4	Overview Chart - Radar Chart	55
C.5	Overview Chart - Heat Map	55
C.6	Analysis results - Price	56
C.7	Analysis results - GWP	56
C.8	Analysis results - Energy	57
C.9	Overview Chart - Radar Chart	57
C.10	Overview Chart - Heat Map	57

List of Tables

2.1	A summary of different LODs	6
2.2	Comparison of the RIBA and HOAI standards	13
2.3	The results of the analysis conducted by Bahramian and Yetilmezsoy (2020) to identify the most commonly considered life cycle stages in LCA studies	16
3.1	LCC and LCA indicators used for comparing design variants	20
3.2	Recommended chart types for comparing design options, adapted from Figure 5 from (HOLLBERG et al., 2021, p. 8)	21
3.3	Available chart types for different chart groups	22
4.1	The sub-parameters of a BIM Snippet, adapted from (BUILDINGSMART, 2017a)	34
4.2	Chart types used in the web application	38
4.3	The arguments of the <i>changeTypeId</i> method, adapted from (TALARICO, 2022)	41
4.4	Parameters of the property set <i>LCA_LCC_indicators</i>	42
B.1	Packages used in the implementation of the analysis server	52
B.2	Packages used in the implementation of the web application	52
B.3	Packages used in the implementation of the Revit plugin	53
C.1	Design variants for load-bearing external walls	54
C.2	Design variants for non-load-bearing external walls	56

List of Algorithms

4.1	The function returning the elementIds	41
4.2	The function returning the type ID	42

Acronyms

3D	Three-dimensional
AEC	Architecture, Engineering, and Construction
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
BCF	BIM Collaboration Format
BDL	Building Development Level
BIM	Building Information Modeling
BOQ	Bill of Quantities
DOM	Document Object Model
EU	European Union
GUID	Globally Unique Identifier
GWP	Global Warming Potential
HOAI	Honorarordnung für Architekten und Ingenieure
HTTP	Hypertext Transfer Protocol
ID	Identifier
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
LCA	Life Cycle Assessment
LCC	Life Cycle Costing
LCI	Life Cycle Inventory
LCIA	Life Cycle Impact Assessment
LOD	Level of Development
LP	Leistungsphase
MVD	Model View Definitions
NIBS	National Institute of Building Sciences
PE	Primary Energy
PENRT	Total Use of Non-renewable Primary Energy Resources
PERT	Total Use of Renewable Primary Energy Resources
REST	Representational State Transfer
RIBA	Royal Institute of British Architects
UI	User Interface
XML	Extensible Markup Language

Chapter 1

Introduction

1.1 Motivation

Approximately 36% of global final energy use and 37% of energy-related CO_2 emissions are attributed to the Architecture, Engineering, and Construction (AEC) sector (UNITED NATIONS ENVIRONMENT PROGRAMME, 2021). In the European Union (EU), the AEC sector accounts for 37.1% of waste generation (EUROSTAT, 2020). In addition, the AEC sector has a significant impact on the economy. For instance, this sector contributes 9% of the EU's Gross Domestic Product (GDP) and employs approximately 18 million people (EUROPEAN COMMISSION, n.d.).

World Green Building Council (2022, p. 9) sets the vision for 2050 as a „decarbonized, circular, resilient, and well-designed built environment that facilitates a high quality of life.“ Considering the facts in the previous paragraph, incorporating assessments, such as Life Cycle Assessment (LCA) and Life Cycle Costing (LCC), into the building design to evaluate a design's environmental and economic impacts is essential to achieve this vision.

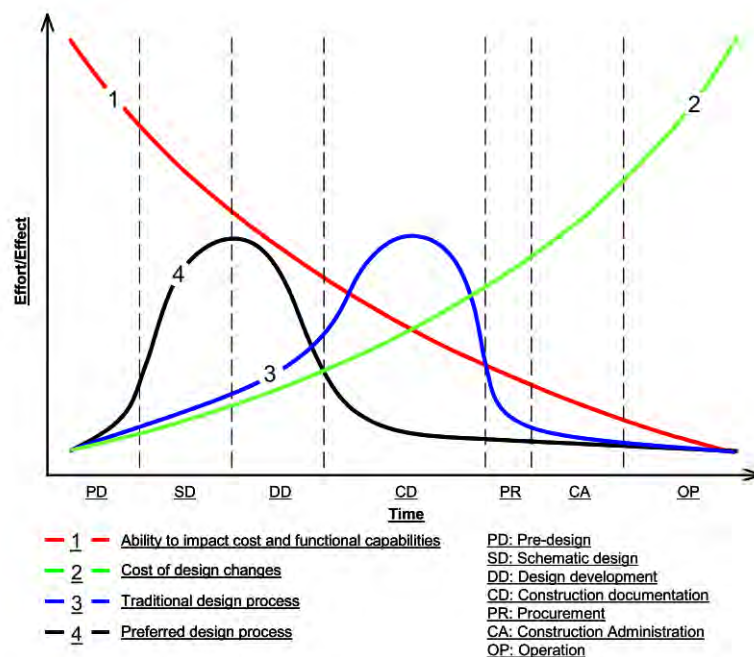


Figure 1.1: The MacLeamy Curve (CONSTRUCTION USERS ROUNDTABLE, 2004, p. 4)

According to Figure 1.1, the MacLeamy Curve (2004), the traditional design process involves the most effort in detailed design stages, resulting in higher costs and less substantial impact on quality. On the other hand, Building Information Modeling (BIM) moves this effort to earlier stages, enabling the integration of computational analyses into building design from early on (BORRMANN, KÖNIG, et al., 2018). Based on this, environmental and economic assessments can significantly impact design decisions throughout the whole design process. Open BIM provides a wide range of software applications for conducting environmental and economic assessments in this context, as it is based on vendor-neutral data exchange formats, such as Industry Foundation Classes (IFC) and BIM Collaboration Format (BCF). A typical open BIM workflow involves sharing BIM models as IFC models and communicating issues in those models through BCF issues. Extension options are available for these data exchange formats.

1.2 Goal and Scope

This master's thesis consists of two objectives. The first objective is to identify visualization methods for comparing design options through LCA and LCC. The first research question relates to this objective and investigates a visualization strategy. The second objective, addressed by the second and third research questions, is to identify extension possibilities of open BIM to automate the updating process of BIM models regarding design decisions. This part of the research examines how BCF issues can incorporate design decisions using BIM Snippets. The research questions addressed in this master's thesis are as follows.

- How can design variants' environmental and economic impact be visualized to support stakeholders in material and component-based decision-making?
- How can design decisions by different project stakeholders be communicated back to the model in the BIM authoring tool?
- What information should be stored in each component of the closed BIM model for the updating process concerning design decisions made in an open BIM workflow?

1.3 Outline

This chapter, *Introduction*, is followed by five other chapters.

Chapter 2, *State of the Art*, consists of three subchapters. The first subchapter presents the general concepts of BIM by focusing on open BIM and its data exchange formats, IFC and BCF. The second subchapter discusses standard practices to define project phases in the AEC sector. Lastly, the third subchapter introduces the technical background regarding LCA and LCC and their integration into BIM processes.

Chapter 3, *a Method for Comparing and Selecting Design Variants*, presents the research questions with their hypotheses. The methodology presented in this chapter consists of a visualization strategy to compare design variants' environmental and economic impacts and a semi-automated method to update BIM models using BCF issues.

Chapter 4, *Prototypical Implementation and Case Study*, focuses on implementing the methodology presented in the previous chapter. The implementation consists of a web application and a Revit plugin. The web application enables comparing different design options according to LCA and LCC results and creating BCF issues for design decisions. Besides, the BIM model in Revit can be updated semi-automatically using the Revit plugin.

Chapter 5, *Conclusion and Outlook*, reviews the research questions and concludes this master's thesis with the limitations of this approach and the possible improvements for the next steps.

Chapter 2

State of the Art

2.1 Building Information Modelling

Despite the growing use of digital technologies, the AEC sector is less advanced than other sectors in utilizing digital information consistently throughout different project phases (BORRMANN, KÖNIG, et al., 2018). While the digital adoption rate, the rate of adapting one or more digital tools, is 66% for manufacturing companies in the EU, it is 40% for construction companies (EUROPEAN INVESTMENT BANK, 2020). The BIM approach uses BIM models, semantically enriched three-dimensional (3D) models, to integrate digital information seamlessly throughout different life cycle stages of a built asset (BORRMANN, KÖNIG, et al., 2018).

2.1.1 Definition

Standards are being established for BIM processes by various organizations. One is the National BIM Standard – United States (NBIMS – US), developed by the Building Information Management Council of the National Institute of Building Sciences (NIBS). The specification provides three definitions of BIM. From the digitalization aspect of built assets, the definition by NIBS (2015, p. 3) is as follows:

„Building Information Model is the digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility, forming a reliable basis for decisions during its life cycle from inception onwards.“

2.1.2 Closed/Open BIM and Little/Big BIM

There are several approaches to implementing BIM. Implementations differ regarding the prevalence of BIM use among project stakeholders, the formats used for data exchange, and the flexibility to choose software from different vendors.

Unlike open BIM, which allows project stakeholders to use software from different vendors and to exchange data in vendor-neutral data formats, closed BIM is limited to software from a single vendor and proprietary data formats. Furthermore, big BIM refers to using BIM models collaboratively between different teams in the same project, in contrast to little BIM, which uses BIM within one project team. (LIEBICH et al., 2011)

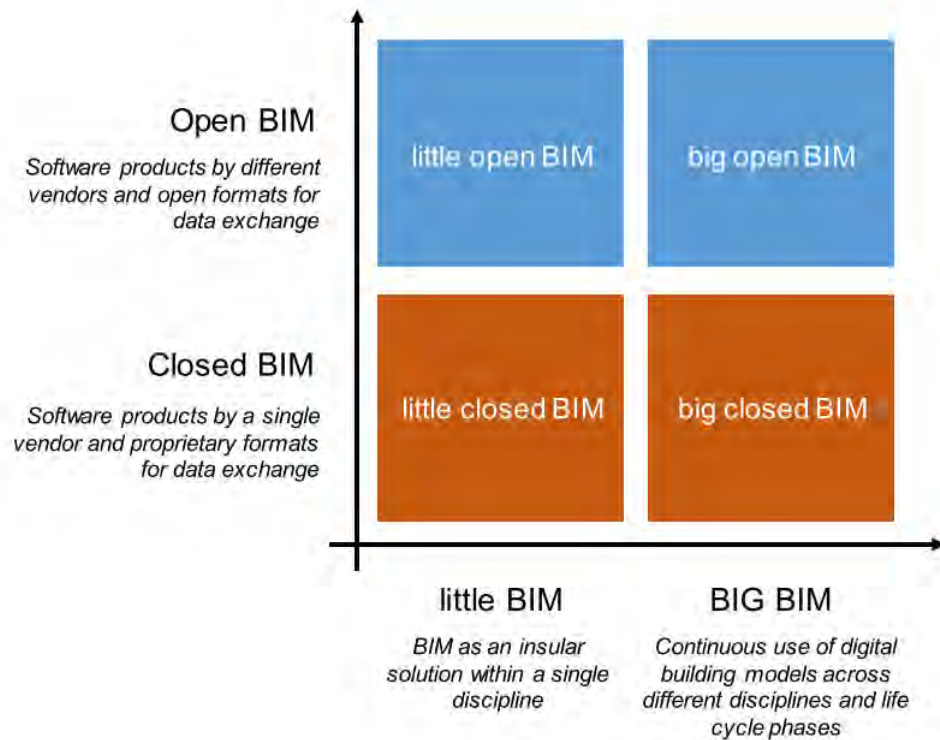


Figure 2.1: Different ways of implementing BIM (BORRMANN, KÖNIG, et al., 2018, p. 12)

As depicted in Figure 2.1, other categories are also available by combining the graph's axes. For instance, big closed BIM refers to using BIM collaboratively but being limited to a single vendor's products (LIEBICH et al., 2011).

2.1.3 Level of Development

Level of Development (LOD) is a specification to standardize the content of BIM models. LOD specifies which geometric details (i.e., Level of Geometry [LOG]) and which non-geometric information (i.e., Level of Information [LOI]) each model component possesses (BORRMANN, KÖNIG, et al., 2018). The developer BIM Forum (2021, p. 5) describes the LOD concept as „a reference tool intended to improve the quality of communication among users of Building Information Models (BIMs) about the characteristics of elements in models.“ The specification defines six different LODs, and the requirements for model components for each LOD are specified in a catalog using illustrations. Table 2.1 provides a summary of different LODs.

Table 2.1: A summary of different LODs

LOD	Explanation (BIMFORUM, 2021, pp. 16, 17)
LOD 100	„The Model Element may be graphically represented in the Model with a symbol or other generic representation, but does not satisfy the requirements for LOD 200. Information related to the Model Element (i.e. cost per square foot, tonnage of HVAC, etc.) can be derived from other Model Elements.“
LOD 200	„The Model Element is graphically represented within the Model as a generic system, object, or assembly with approximate quantities, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.“
LOD 300	„The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of quantity, size, shape, location, and orientation. Non-graphic information may also be attached to the Model Element.“
LOD 350	„The Model Element is graphically represented within the Model as a specific system, object, or assembly in terms of quantity, size, shape, location, orientation, and interfaces with other building systems. Non-graphic information may also be attached to the Model Element.“
LOD 400	„The Model Element is graphically represented within the Model as a specific system, object or assembly in terms of size, shape, location, quantity, and orientation with detailing, fabrication, assembly, and installation information. Non-graphic information may also be attached to the Model Element.“
LOD 500	„The Model Element is a field verified representation in terms of size, shape, location, quantity, and orientation. Non-graphic information may also be attached to the Model Elements.“

2.1.4 Data Exchange

AEC projects consist of project stakeholders from various disciplines, each requiring different software tools. The interoperability of these software tools is crucial to preventing data loss during data exchange (BORRMANN, BEETZ, et al., 2018). To achieve interoperability, it is necessary to develop a data model that „conceptually defines elements required for a target domain and the relationships between the elements“ (SACKS et al., 2018, p. 90).

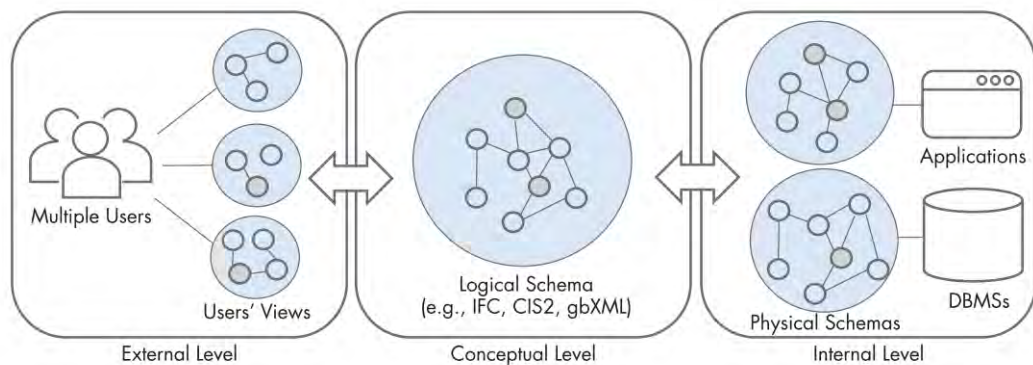


Figure 2.2: Data models at different levels (SACKS et al., 2018, p. 90)

As depicted in Figure 2.2, data is represented by data models at three different levels. According to Sacks et al. (2018), the summary of these levels is as follows:

- Conceptual Level: The Logical Schema describes the content and does not depend on the implementation.
- External Level: Users' Views are the content tailored to users' needs.
- Internal Level: Physical Schemas are the implementations of the Logical Schema concerning specific software applications.

In addition, Sacks et al. (2018) provide three main data exchange methods: direct links, file-based data exchange, and model-server-based data exchange. Below is a summary of these data exchange methods.

Direct link

This method uses Application Programming Interfaces (APIs) to allow data exchange between different software applications. Interfaces provide basic functionalities (e.g., export, import, modification, etc.), enabling the use of building models in other applications for purposes such as conducting analyses. In addition to forming partnerships with other companies, vendors generally have these interfaces for their own products.

File-based data exchange

Building models are shared between project stakeholders using files in vendor-neutral or proprietary data formats. For instance, IFC and BCF are vendor-neutral formats. An example of a proprietary file format is RVT, the native file format from Revit.

Model-server based data exchange

The last data exchange method uses a Database Management System (DBMS) for data exchange. For instance, a Common Data Environment (CDE) centralizes data for all project participants to store and obtain information (PREIDEL et al., 2018). Compared to the file-based data exchange, the model-server-based approach eliminates many issues associated with version control and simultaneous engineering.

2.1.5 Industry Foundation Classes (IFC)

Implementing big BIM requires frequent data exchange between project stakeholders, which can be accomplished through closed or open BIM. The lack of support for all data exchange possibilities in proprietary file formats necessitates using vendor-neutral data exchange formats (BORRMANN, BEETZ, et al., 2018). IFC is a vendor-neutral data

format used for representing buildings and infrastructures digitally. The IFC specification is developed by buildingSMART, and there are two official versions available: version 4.0.2.1, published in 2017, and version 2.3.0.1, published in 2007 (van BERLO, 2022). A non-profit organization, buildingSMART (n.d.-b) defines its vision as „the full realization of the societal, environmental and economic benefits of open sharable infrastructure and building asset information into commercial and institutional processes worldwide.“

Developed with the object-oriented approach, the IFC data model consists of entities representing real-world objects, having specific attributes, and connecting other entities with relationships. An IFC model contains a building's geometry and the associated semantic information. Although quite advanced with the latest version, there are still some problems such as data loss when using IFC models in different software applications. Issues mainly stem from import and export interfaces of software applications. As the IFC schema is quite complex, there are several ways of representing the 3D geometry; therefore, implementing these interfaces for each representation becomes problematic. Model View Definitions (MVD), which specifies the implementation of IFC concerning a particular exchange case, was developed to address this issue by buildingSMART. (BORRMANN, BEETZ, et al., 2018)

Section 2.1.4 explains the data exchange in BIM processes. According to the data models represented at three different levels (see Figure 2.2), IFC corresponds to the Logical Schema at the Conceptual Level, and MVD is the Users' Views at the External Level (SACKS et al., 2018).

Layers of IFC

As depicted in Figure 2.3, the IFC schema consists of four layers: Domain, Interoperability, Core, and Resource. Referencing a layer from another is only possible in the top-to-bottom direction (BORRMANN, BEETZ, et al., 2018). According to the specification (IFC4 ADD2 TC1) by buildingSMART (2017b), the characteristics of these layers are summarized below.

- Domain Layer: The top layer of the schema consists of entities specific to a particular discipline (e.g., architecture, electrical, structural, etc.). For instance, *IfcDoorPanel-Properties* is an entity used in architecture to describe a door panel.
- Interoperability Layer: The elements shared among multiple domains are defined in this layer. An example of the entities is *IfcWall* specified under shared building elements.
- Core Layer: This layer contains the fundamental definitions, which the upper layers can reference and specify. The kernel schema contains the most abstract entities (e.g., *IfcRoot*, *IfcObject*, *IfcProduct*, etc.), and the extension schemas make definitions concerning a product (e.g., *IfcSpace*), process (e.g., *IfcEvent*), and control (e.g., *IfcPerformanceHistory*). In this layer and the abovementioned layers, all enti-

ties inherit the attributes of *IfcRoot*; thus, they contain a Globally Unique Identifier (GUID).

- Resource Layer: The entities defined in this layer do not contain GUIDs, so the independent use of these entities is not possible. Instead, they can be referenced from the upper layers to enrich the content of data in terms of material, geometry, topology, etc.

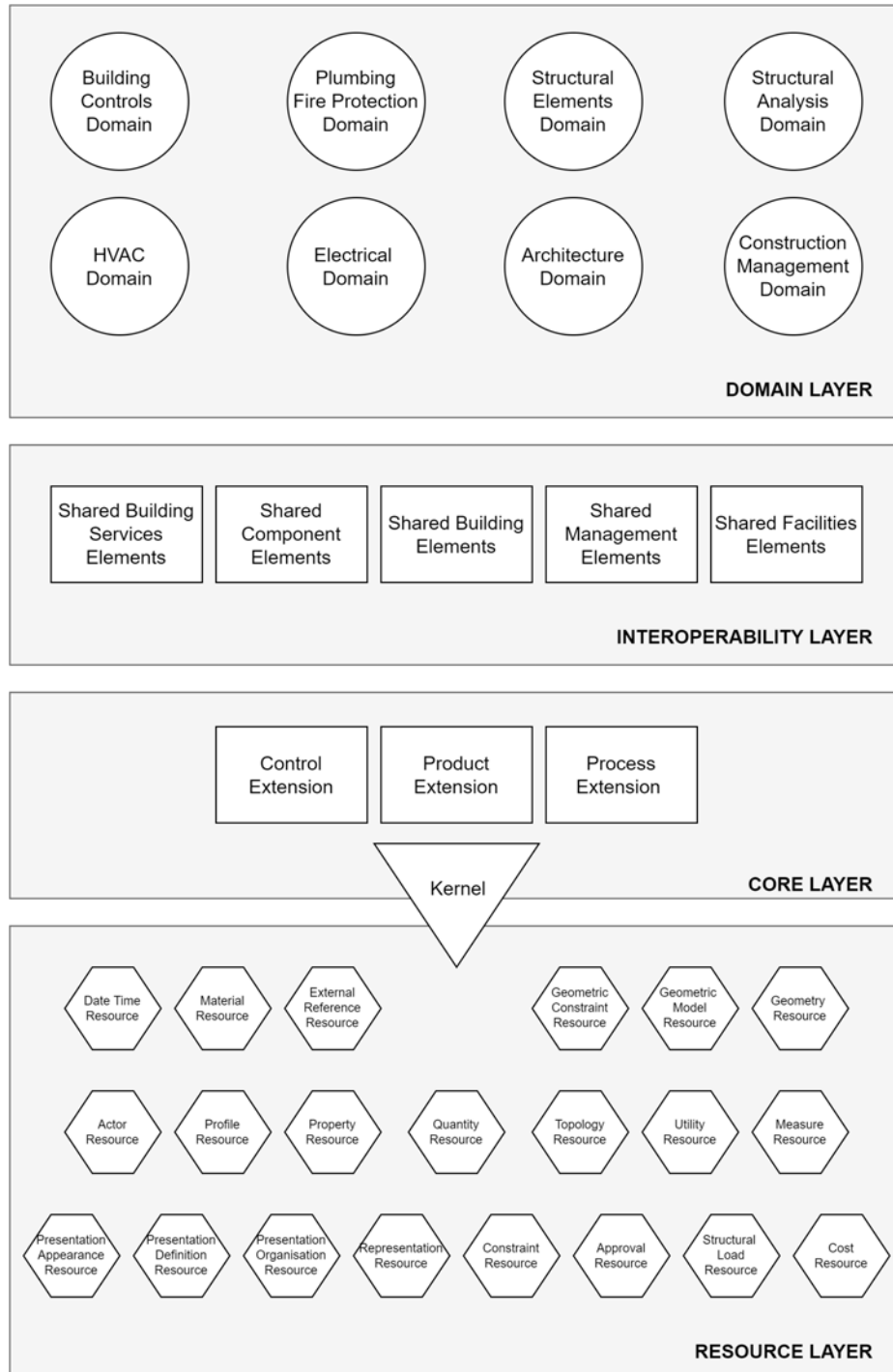


Figure 2.3: Layers of the IFC schema, adapted from Figure 1 from (BUILDINGSMART, 2017b)

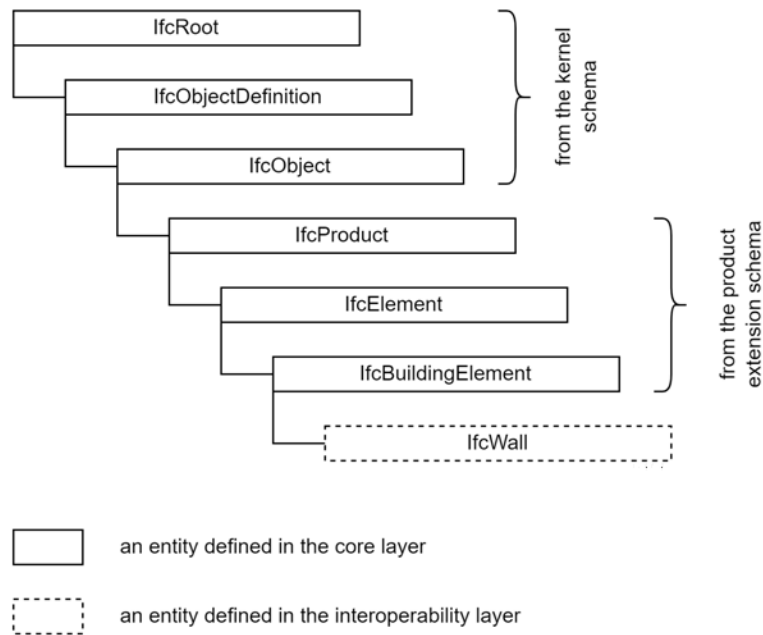


Figure 2.4: Inheritance structure of *IfcWall*, adapted from the entity inheritance schema in Section 6.1.3.46.2 from (BUILDINGSMART, 2017b)

As IFC is based on the object-oriented data model, entities inherit attributes from other entities depending on their position in the inheritance hierarchy (BORRMANN, BEETZ, et al., 2018). Figure 2.4 depicts the inheritance structure of the *IfcWall* entity. According to the figure, *IfcWall* inherits the properties from the entities above.

Extending the Schema

IFC objects consist of static attributes that exist in the schema by default and dynamic attributes that users can add by extending the schema. The *IfcPropertySingleValue* entity enables users to define custom properties, and *IfcPropertySet* contains these definitions and relates them to an object via *IfcRelDefinesByProperties*. For instance, in Figure 2.5, the custom attributes, such as *Thermal Transmittance*, *Is External*, and *Fire Rating*, are defined under the *IfcPropertySet* entity named *Pset Door Common*, which relates to an *IfcDoor* object. Besides, to avoid the creation of the same custom attributes multiple times with different names, buildingSMART provided standardized *IfcPropertySets* for several entities. These property sets are available in the IFC specification. (BORRMANN, BEETZ, et al., 2018)

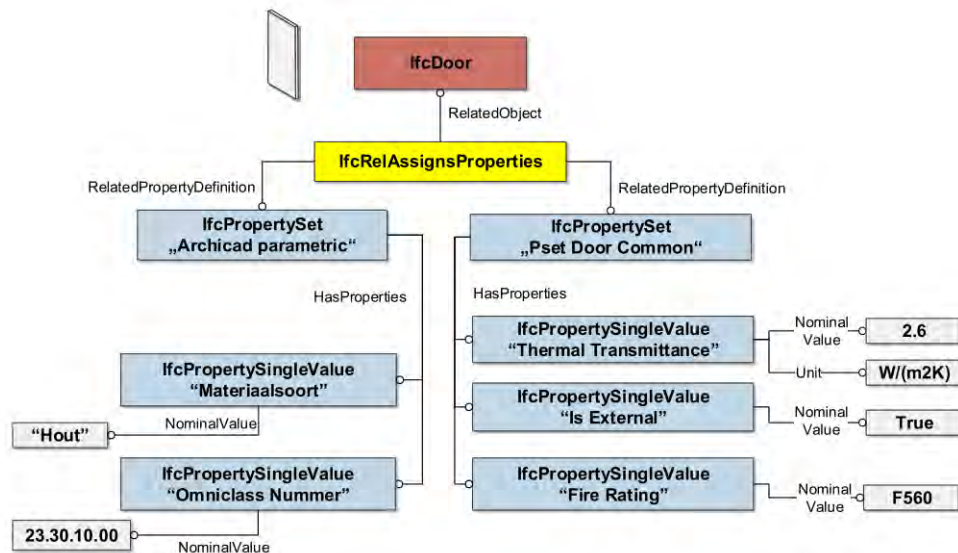


Figure 2.5: Adding custom properties to an *IfcDoor* object (BORRMANN, BEETZ, et al., 2018, p. 31)

2.1.6 BIM Collaboration Format (BCF)

An open BIM standard from buildingSMART, the BCF enables the interchange of issues based on IFC models between BIM applications. The BCF is available in two formats: BCF-XML, based on file exchange, and BCF-API, based on RESTful API. (BUILDINGSMART, n.d.-a)

The BCF is used during different phases of a project. For instance, during the design phase, project teams work with discipline-specific models (e.g., structural and architectural models) and regularly bring these models together to identify clashes between model components. In this case, issues regarding interfering model components can be communicated to responsible stakeholders using the BCF.

Issues in BCF-XML and BCF-API share commonalities. A Topic contains general information (e.g., status, label, etc.) about the issue, and Comments are text entries describing the issue. Besides, a Viewpoint consists of all the necessary information to visualize the issue (e.g., location, camera view, referenced IFC elements, images, etc.). (SCHULZ & BEETZ, 2021)

In addition, starting from version 2.1, BIM Snippets, arbitrary data (e.g., a partial IFC model), can be attached to BCF issues. However, using BIM Snippets is not a common practice in the AEC industry. (ZAHEDI & PETZOLD, 2019)

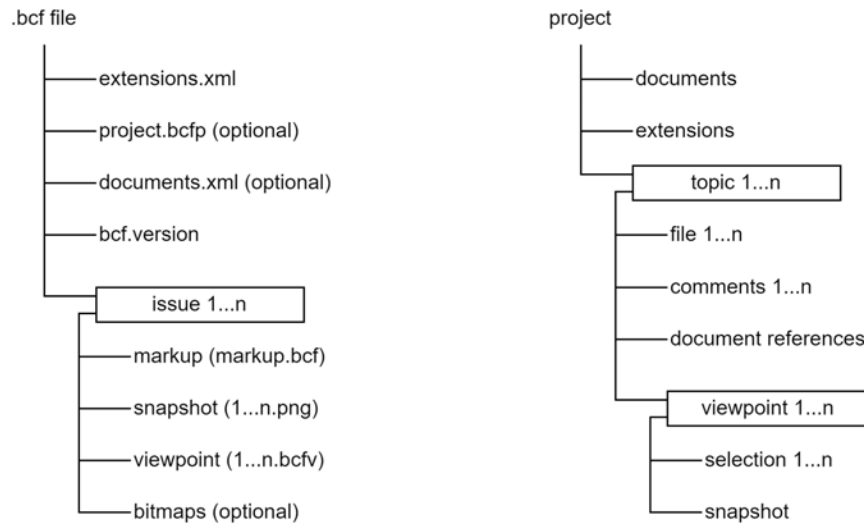


Figure 2.6: General structure of BCF-XML (left) and BCF-API (right), adapted from (SCHULZ & BEETZ, 2021, p. 3) and (BUILDINGSMART, 2022)

BCF-XML

Project stakeholders exchange issues using files with the *.bcf* file extension. Each of these files consists of several XML-based files and can store multiple issues. Figure 2.6 depicts the general structure of BCF-XML. Each issue includes a Markup, Viewpoints, Snapshots, and, optionally, Bitmaps. A Markup file consists of a Topic, Comments, and a link to Viewpoints and Snapshots (SCHULZ & BEETZ, 2021). In Viewpoints, a BCF issue references an IFC model with its spatial coordinates and an IFC element with its GUID (SCHULZ et al., 2021). According to van Berlo and Krijnen (2014), file-based issue exchange is error-prone and causes decentralized project management; therefore, they suggest server-based solutions as an alternative. BCF-XML has four versions: 1.0, 2.0, 2.1 and 3.0 (BUILDINGSMART, 2022).

BCF-API

BCF-API is based on RESTful-API to build server-based applications. As shown in Figure 2.6, a Project is at the top of the hierarchical structure. Since Viewpoints and Comments connect in Topics in the hierarchy, a Topic corresponds to a BCF issue (SCHULZ et al., 2021). BCF-API consists of three versions: 1.0, 2.1, and 3.0 (BUILDINGSMART, 2017a).

A significant drawback of BCF-API is the hierarchical structure in which a request to the above levels is necessary to access the data at a certain level. This structure can be problematic, especially if the number of requests increases or/and the requested data is at a low level in the hierarchy. (SCHULZ & BEETZ, 2021)

2.2 Project Phases in Construction

A project's structure in terms of its phases varies from country to country. In the UK, the Plan of Work prepared by the Royal Institute of British Architects (RIBA) consists of eight stages, each describing the scope of the work and the responsibilities of different disciplines in a project. Figure A.1 in Appendix A compares the Plan of Work with the approaches from other countries or regions. Despite their differences, project phases can be grouped into the following categories: Pre-Design, Design, Construction, Handover, In-Use, and End of Life (ROYAL INSTITUTE OF BRITISH ARCHITECTS, 2020).

In Germany, the common practice to define project phases is using the Honorarordnung für Architekten und Ingenieure (HOAI) standards. It consists of nine project phases, each called Leistungsphase (LP). Table 2.2 compares the RIBA and HOAI standards by matching each LP with a RIBA stage or stages. The comparison of the two standards in Table 2.2 is according to the explanations in (WERNER & PASTOR, 2019) and (ROYAL INSTITUTE OF BRITISH ARCHITECTS, 2020).

The early design stages may cover different project phases, depending on how it is defined. Schneider-Marín and Abualdenien (2019) define the early design stages as those in which the preparation of construction documents has not begun; this corresponds to the phases from LP1 to LP3 in the HOAI standards.

Table 2.2: Comparison of the RIBA and HOAI standards

Stages from RIBA	LPs from HOAI
Stage 0: Strategic Definition Stage 1: Preparation and Briefing	LP1: Grundlagenermittlung
Stage 2: Concept Design Stage 3: Spatial Coordination	LP2: Vorplanung
Stage 4: Technical Design	LP3: Entwurfsplanung
	LP4: Genehmigungsplanung
	LP5: Ausführungsplanung
	LP6: Vorbereitung der Vergabe
LP7: Mitwirkung bei der Vergabe	
Stage 5: Manufacturing and Construction Stage 6: Handover	LP8: Objektüberwachung – Bauüberwachung und Dokumentation
Stage 7: Use	LP9: Objektbetreuung

A BIM model's geometry and semantic content vary across project phases. According to BIM Forum (2021, p. 266), „*project models at any stage of delivery will invariably contain elements and assemblies at various levels of development.*“ Therefore, specifying an LOD for the entire building model in a particular project phase is inaccurate (BIMFORUM, 2021). Abualdenien and Borrmann (2019) introduced a new concept, Building Development Level (BDL), to classify a building model regarding its content. Accordingly, each of the five BDLs consists of model components that require different LODs.

2.3 Environmental and Cost Analyses

2.3.1 Life Cycle Assessment (LCA)

Standardized with ISO 14040 and ISO 14044, International Organization for Standardization (ISO) (2006) defines an LCA as follows:

„LCA addresses the environmental aspects and potential environmental impacts (e.g., use of resources and the environmental consequences of releases) throughout a product's life cycle from raw material acquisition through production, use, end-of-life treatment, recycling, and final disposal (i.e., cradle-to-grave).“

DIN EN 15804:2022-03 specifies the impact categories for an LCA. There are several environmental databases, one of which is ÖKOBAUDAT by the German Federal Ministry for Housing, Urban Development, and Building (2021). ÖKOBAUDAT is compatible with the current and previous versions of the standard.

LCA Phases

Figure 2.7 depicts the general framework of an LCA with its phases and applications. According to ISO 14040:2006, an LCA study consists of four phases: Goal and Scope Definition, Inventory Analysis, Impact Assessment, and Interpretation (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2006). As indicated with the arrows in the framework, an LCA typically requires an iterative process (KLÖPFFER & GRAHL, 2014).

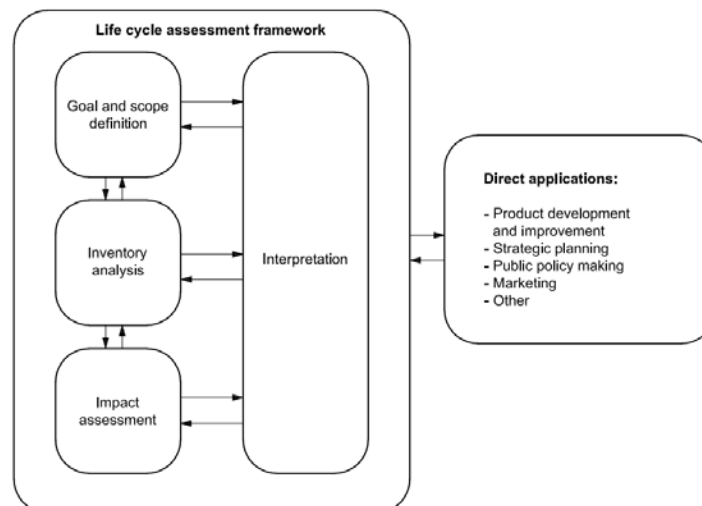


Figure 2.7: LCA phases (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2006, p. 8)

The following paragraph summarizes the LCA phases concerning ISO 14040:2006. After defining the content and the purpose of an LCA study in the first phase, the second stage involves gathering the necessary data and the Life Cycle Inventory (LCI) analysis that

identifies the subject system's inputs and outputs. Based on the data and the findings in the Inventory Analysis, the Life Cycle Impact Assessment (LCIA) is conducted in the third phase to evaluate the system's environmental impact. Lastly, the Interpretation Phase concludes the assessment by reviewing the LCI and LCIA studies and gives recommendations regarding the goals defined in the first phase. (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2006)

Life Cycle Stages of Buildings

According to DIN EN 15978-1:2021-09, a building's life cycle consists of five stages: A0 (Pre-construction), A1-3 (Product Stage), A4-5 (Construction Process), B1-8 (Use Stage), and C1-4 (End of Life Stage). LCA studies typically do not consider A0 as it pertains to activities before construction, which have neglectable environmental impacts. Furthermore, the module D deals with the „recovery and repurposing of resources“ after a building reaches the end of its life. (EUROPEAN COMMITTEE FOR STANDARDIZATION, 2021)

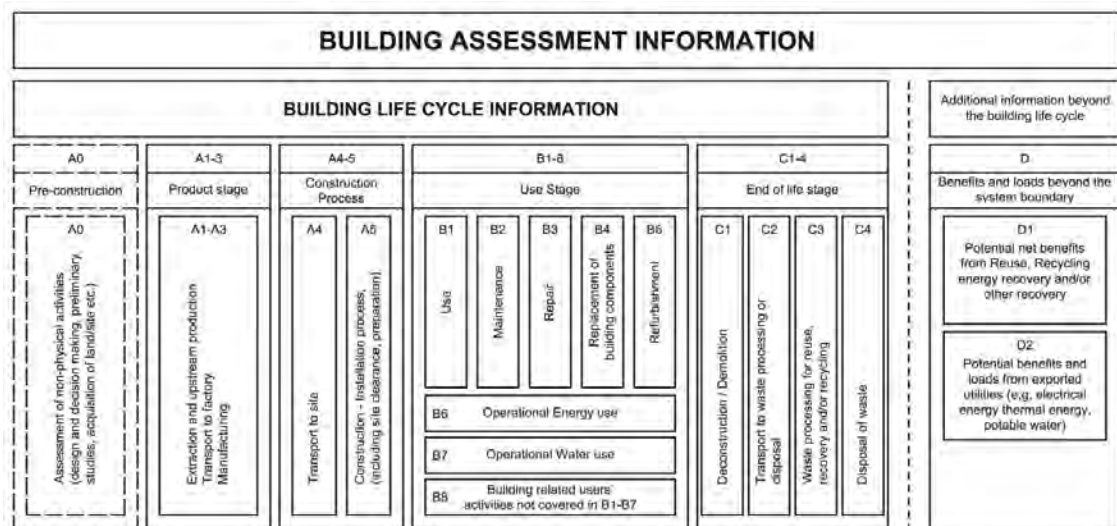


Figure 2.8: Life cycle stages for a building assessment (EUROPEAN COMMITTEE FOR STANDARDIZATION, 2021, p. 30)

Figure 2.8 depicts a building's life cycle as described in DIN EN 15978-1:2021-09, which is still a draft standard. The current norm DIN EN 15978:2012-10 does not provide a separation inside the D module, such as D1 and D2, and does not include A0 and B8.

It is common in practice to not consider the life cycle stages that have a neglectable impact on the final result (BAHRAMIAN & YETILMEZSOY, 2020). Hence, Bahramian and Yetilmezsoy (2020) analyzed eighty-seven case studies to identify which life cycle stages are frequently included or excluded in LCA studies. Except for B3 and B5, the analysis considers all the life cycle stages available in the current standard. Accordingly, while A3, A4, and A5 are the most included stages in LCA studies, the D module is the least. Table 2.3 depicts the results.

Table 2.3: The results of the analysis conducted by Bahramian and Yetilmezsoy (2020) to identify the most commonly considered life cycle stages in LCA studies

Life cycle stages	Percentage ^a (%)
A1: extraction and upstream production	67
A2: transport to the factory	69
A3: manufacturing A4: transport to site A5: construction - installation process	>= 70
B1: use B2: maintenance B4: replacement of building components	57
B6: operational energy use B7: operational water use	63
C1: deconstruction / demolition C2: transport to waste processing or disposal C3: waste processing for reuse, recovery and/or recycling C4: disposal of waste	54
D: benefits and loads beyond the system boundary	11

^aThe percentage (%) of the studies that include the life cycle stage(s)

2.3.2 Life Cycle Costing (LCC)

Standardized with ISO 15686-5:2017, ISO (2017, p. 2) defines the Life Cycle Cost as the „cost of an asset, or its part throughout its cycle life, while fulfilling the performance requirements.“

ISO 15686-5:2017 provides a methodology named Life Cycle Costing (LCC) for evaluating the Life Cycle Costs of a built asset. The standard also defines the term Whole Life Cycle Cost, which differs from the Life Cycle Cost by including additional aspects such as Externalities, Non-construction Costs (e.g., taxes), and Income. From a broader perspective, an action's positive or negative effects on people, the environment, and business are considered in Externalities. (INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2017)

The cost indicators in an LCC study can be associated with project, utility, and maintenance costs. Project costs refer to pre-construction (e.g., design works, obtaining a construction permit, etc.) and construction works. The use stage of a facility consists of utility costs (e.g., expenses for water and energy, etc.) and maintenance costs that occur routinely (i.e., planned maintenance) or in the case of unexpected problems (i.e., reactive maintenance). (GUNDES, 2016)

Besides the abovementioned costs, additional ones occur at the end of a building's life, such as those associated with the C and D modules in Figure 2.8. A common practice is to group Life Cycle Costs using a classification system. For example, Germany uses the cost groups defined in DIN 276 for this purpose.

2.3.3 BIM-based Decision-making Through LCA and LCC

Wastiels and Decuyper (2019) identified five main approaches to integrating LCA calculations with BIM. In the first approach, the Bill of Materials (BOQ) is exported from the BIM authoring tool and used in the analysis software to perform LCA calculations. Instead of using the BOQ, the second approach uses the exported building model (e.g., the IFC model). The third approach includes additional software, a BIM viewer, between the BIM authoring tool and the analysis software to assign LCA information to the IFC model components. In contrast to the previous approaches, which require exporting BIM models, LCA calculations are performed via plugins within the BIM authoring tool in the fourth approach. The fifth approach involves enriching model components in the BIM authoring tool with LCA information and using plugins or external software to conduct LCA studies. Moreover, the literature reviews by Lu et al. (2021) and Santos et al. (2019) considered both LCA and LCC for their integration with BIM.

According to Wastiels and Decuyper (2019), the first approach, using the BOQ, is the most common way of integrating BIM with LCA calculations. In their literature review, Potrč Obrecht et al. (2020) categorized the case studies considering the five approaches mentioned in the previous paragraph. Based on the results, the first approach was the most preferred, followed by the fourth. While the fifth approach was the third most common, the second and the third were the least preferred.

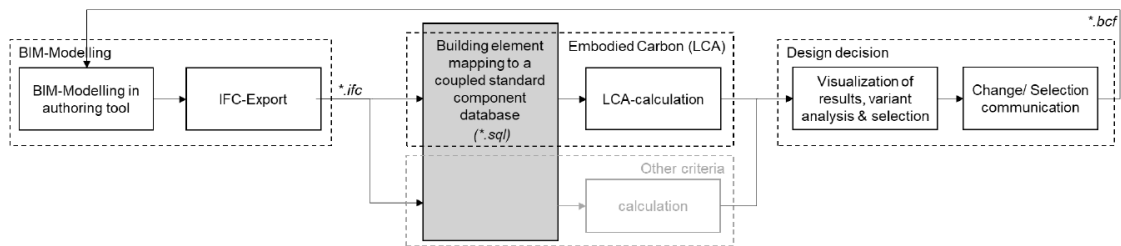


Figure 2.9: Framework for optimizing building designs based on multicriteria analyses (FORTH et al., 2021, p. 238)

Forth et al. (2021) developed a framework for improving building designs by focusing on multicriteria analyses. As depicted in Figure 2.9, although the mentioned research focuses on LCA, other criteria, such as LCC, can also be considered. Based on open BIM, the framework consists of vendor-neutral data formats such as IFC to perform an LCA and BCF to communicate design decisions to the BIM authoring tool.

The current workflow with BCF issues to update the model in the BIM authoring tool is a manual process. The responsible project stakeholder often updates the model concerning the decision expressed as a comment in the BCF issue, which might result in errors. Therefore, another approach that allows automation in the updating process and improves communication between project stakeholders about design decisions is necessary.

Chapter 3

A Method for Comparing and Selecting Design Variants

3.1 Research Questions

This master's thesis addresses three research questions to improve the decision-making process supported by environmental and economic assessments. Below are the research questions and hypotheses, along with additional explanations.

Research Question I: How can design variants' environmental and economic impact be visualized to support stakeholders in material and component-based decision-making?

Hypothesis: Design variants' environmental and economic impact can be visualized using combined metrics from multicriteria analyses.

The first research question focuses on determining a visualization strategy to compare design variants. There are four indicators taken into consideration for design variants' environmental impact: Global Warming Potential (**GWP**), Total Use of Renewable Primary Energy Resources (**PERT**), Total Use of Non-Renewable Primary Energy Resources (**PENRT**), and Total Use of Primary Energy (**PE**). A design variant's economic impact is assessed via the indicator Price, the total cost of the design variant.

Research Question II: How can design decisions by different project stakeholders be communicated back to the model in the BIM authoring tool?

Hypothesis: Design decisions can be communicated back to the model in the BIM authoring tool by extending open-BIM-based automation workflows.

The second research question investigates the open BIM format BCF and its extension possibilities, BIM Snippets. The approach includes communicating design decisions and analyses used in the decision-making process back to the model in the BIM authoring tool.

Research Question III: What information should be stored in each component of the closed BIM model for the updating process concerning design decisions made in an open BIM workflow?

Hypothesis: The closed BIM model should store the information uniquely present in each model component of the open BIM model.

The final question concerns the connection between the closed and open BIM models. Since analyses and design decisions are made using the open BIM model, the components of the closed BIM model should store additional information for the updating process.

3.2 General Framework

The general framework consists of the closed BIM model (i.e., the model in the BIM authoring tool) and the open BIM model (i.e., the IFC model). The model in the BIM authoring tool serves as a common repository of design. On the other hand, the IFC model is exported from the closed BIM model for various applications, in the scope of this thesis, for making design decisions.

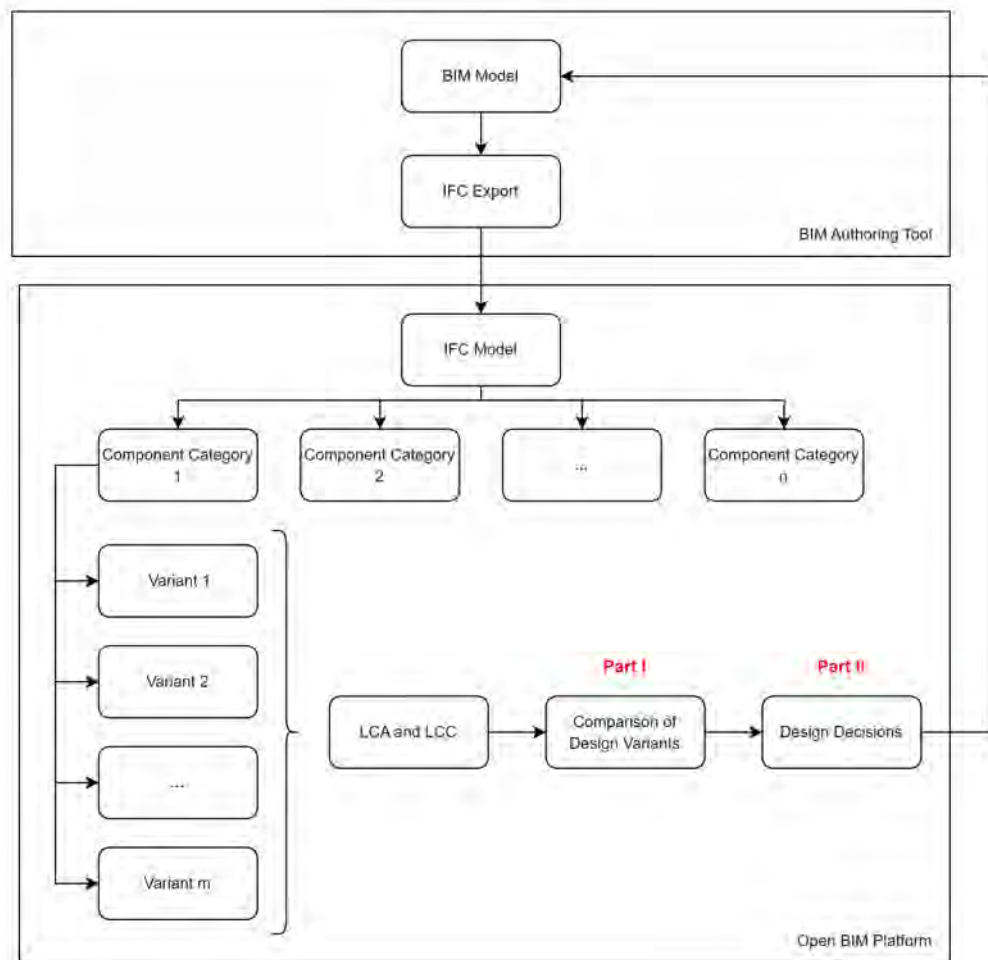


Figure 3.1: General framework

Figure 3.1 depicts the general framework. The IFC model consists of several component categories, each with several design variants. Lammers (2021) combined environmental and cost data from multiple databases into a single database and performed multicriteria analyses based on BIM models in his master's thesis. The first part of this thesis (*Part I – Comparison of Design Variants*) utilizes his work and determines a visualization strategy for comparing design variants provided by Lammers (2021) as examples.

In addition, the second part of this thesis (*Part II – Design Decisions*) focuses on making design decisions based on the IFC model and updating the model in the BIM authoring tool concerning these design decisions.

3.3 Part I: Comparison of Design Variants

Design variants are compared based on the following indicators: Price, GWP, PERT, PENRT, and PE. Table 3.1 summarizes these indicators with their units.

Table 3.1: LCC and LCA indicators used for comparing design variants

Indicator	Unit
Price	EUR
Global Warming Potential (GWP)	kg CO ₂ eq.
Total use of renewable primary energy resources (PERT)	MJ
Total use of non-renewable primary energy resources (PENRT)	MJ
Total Use of Primary Energy (PE)	MJ

Two main chart groups are defined for comparing design variants: an overview chart, which includes several criteria and provides a general overview of the performance of different design variants, and a criterion-specific chart, which displays the results for a single criterion.

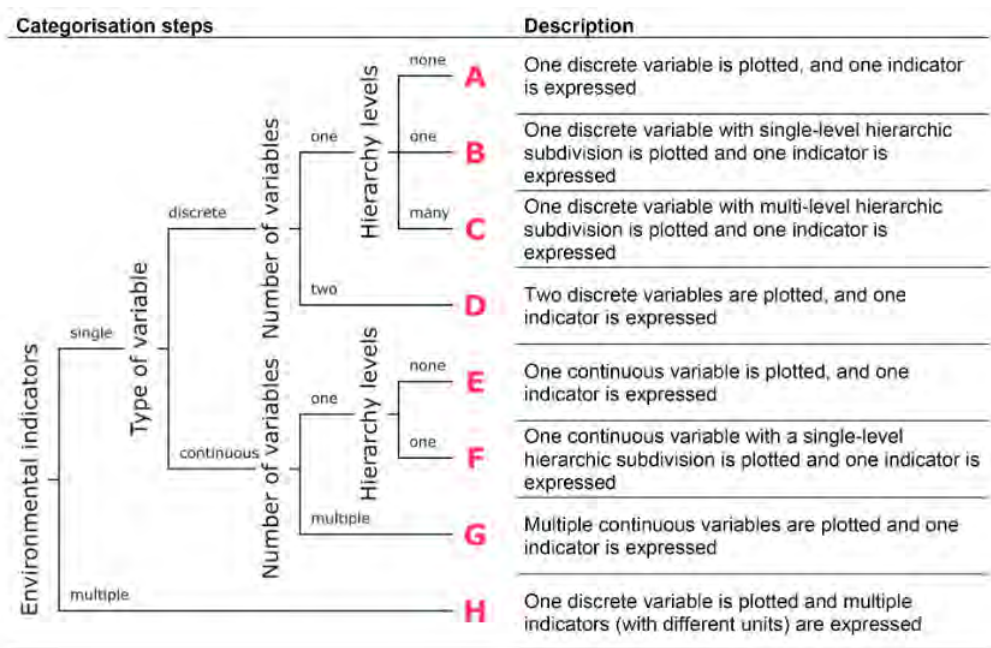


Figure 3.2: The schema for determining the visualization category (HOLLBERG et al., 2021, p. 6)

Hollberg et al. (2021) provided a schema, shown in Figure 3.2, to determine the visualization category, which then determines the chart type. The schema consists of four levels for specifying the visualization category. In the first level, if multiple indicators have different units, the visualization category is H; otherwise, they are considered a single indicator. The next level includes determining the type of variables. In the scope of this thesis, a variable corresponds to a design variant, which is a discrete variable. The third level is

the number of variables. For instance, even though there are several design variants, they all belong to the same variable category; therefore, the number of variables is one. Lastly, the hierarchy levels are used in the schema to determine whether it is possible to decompose data into sub-layers (e.g., visualizing the performance of a building regarding LCA indicators separately for each life cycle stage). Besides, Hollberg et al. (2021) provided chart types matching the visualization categories. Table 3.2 show the recommended charts for comparing different design options; therefore, not all the visualization categories are relevant but only A, B, D, and H. While criterion-specific charts belong to category A, overview charts belong to category H.

Table 3.2: Recommended chart types for comparing design options, adapted from Figure 5 from (HOLLBERG et al., 2021, p. 8)

Category	Recommended Charts
A	Vertical bar chart Horizontal bar chart Grouped bar chart Pictorial fraction chart
B	Stacked bar chart Normalized bar chart Box plot
D	Multiple series 3D bar charts
H	Radial chart/spider chart/polar chart Scatter plot Parallel coordinates

In addition, as an overview chart, a cluster heat map can summarize the performance of different design variants. Cerdas et al. (2017) applied cluster heat maps to visualize LCA results, and this thesis uses their method to build such a visualization tool. Although the concept they represented shows the impact categories of each material of a product's component, it was adjusted to compare design variants for this study.

Product (Building Component)	Design Variants	Life Cycle Phases																	
		P ₁				P ₂				...				P _n					
		Impact Category				Impact Category				Impact Category				Impact Category					
		c ₁	c ₂	c ₃	...	c _n	c ₁	c ₂	c ₃	...	c _n	c ₁	c ₂	c ₃	...	c _n	c ₁	c ₂	c ₃
X ₁	Y ₁	$c_y = (c_j)_{x_1y_1}$																	
	Y ₂	$c_y = (c_j)_{x_1y_2}$																	
																	
	Y _n	$c_y = (c_j)_{x_1y_n}$																	

$\underbrace{\hspace{10em}}_{j = 1, 2, \dots, n}$

Figure 3.3: The modified concept for cluster heat maps to compare design variants, adapted from (CERDAS et al., 2017, p. 735)

Figure 3.3 depicts the modified version of this concept. Every life cycle stage (p_i) consists of impact categories (c_j), and c_y represents a vector calculated for each design variant. For each impact category, the design variant with the maximum/minimum value is assigned to the highest/lowest value in the color schema, and the rest of the cells are colored accordingly.

This thesis uses the abovementioned visualization strategies by Hollberg et al. (2021) and Cerdas et al. (2017). Accordingly, Table 3.3 summarizes the available chart types for criterion-specific and overview charts.

Table 3.3: Available chart types for different chart groups

Chart Group	Chart Types
Criterion-specific	Vertical bar chart Horizontal bar chart Grouped bar chart Pictorial fraction chart
Overview	Radial chart/spider chart/polar chart Scatter plot Parallel coordinates Heat map

As mentioned, the analysis results for a particular criterion can be observed using criterion-specific charts. If all indicators are expressed in the same units, these charts can also display the results of multiple indicators. For instance, while the cost of different design variants includes only the Price indicator, the energy performance consists of PERT, PENRT, and PE. A criterion-specific chart does not combine design variants' environmental and economic impact.

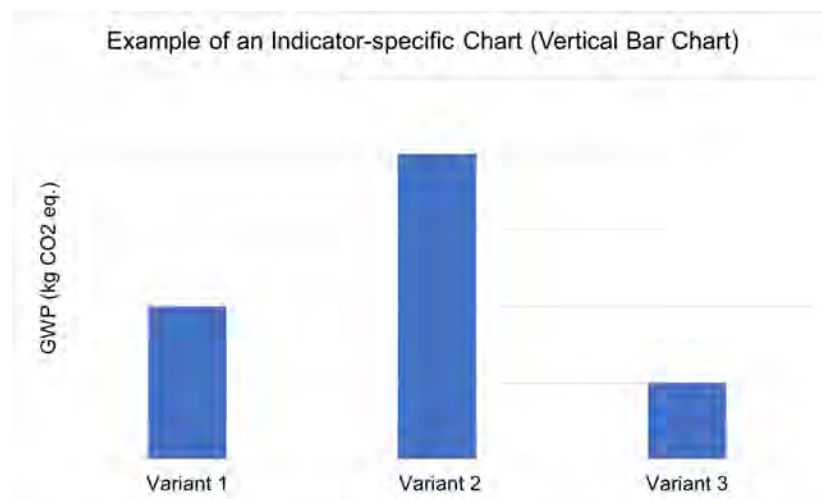


Figure 3.4: An example of a criterion-specific chart

Figure 3.4 is an example of a criterion-specific chart, a vertical bar chart. A comparison of three design variants is illustrated in the figure based on their GWP performance. In the case of combining several indicators, grouped bar charts are suitable.

In contrast to criterion-specific charts, overview charts show the combined environmental and economic impact. These charts are especially beneficial for stakeholders without technical background because they summarize the performance of design variants on different criteria. When using overview charts, normalization is necessary because the indicators are in different units. Normalization is achieved by assigning the highest value to 1 and the lowest value to 0, then calculating the mid-values accordingly.

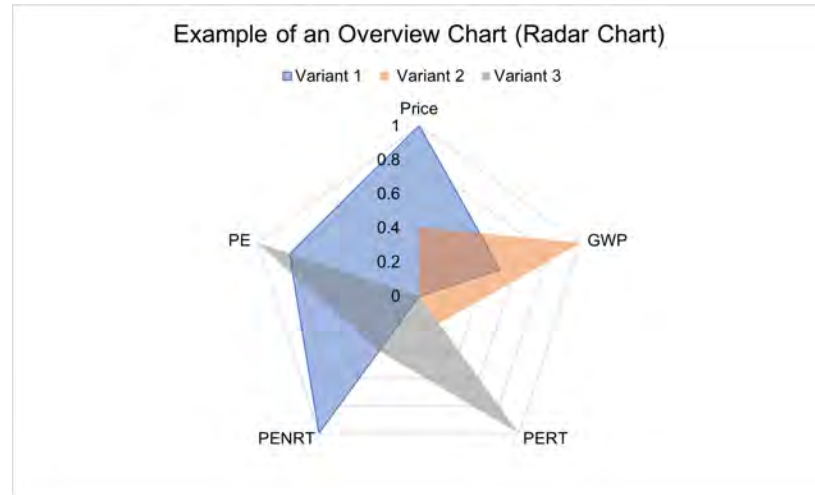


Figure 3.5: An example of an overview chart

Figure 3.5 is an example of an overview chart, a radar chart. For instance, among the three variants, the first variant has the highest Price and PENRT values, the second variant has the highest PE and PERT values, and the third variant has the highest GWP value.

Lammers (2021) made simplifications in conducting LCA and LCC. Therefore, there are limitations in visualizing the results. Since the analysis outputs a single total value, the results cannot be displayed separately for different life cycle stages. For example, in this case, a heat map contains only a single column instead of multiple ones showing different life cycle stages (p_i).

3.4 Part II: Design Decisions

3.4.1 Communicating Design Decisions

As mentioned in Section 2.1.6, the BCF is available in two formats: based on file exchange, BCF-XML, or on server implementation, BCF-API (BUILDINGSMART, n.d.-a). On the one hand, since issues are saved in a database instead of shared as files between stakeholders, BCF-API provides more structured issue management. On the other hand, to use BCF-API in other software applications, such as in BIM authoring tools, a customized plugin is required, as opposed to BCF-XML, which can be imported into already available plugins. It is possible to extend BCF issues using BIM Snippets in both formats. According to buildingSMART (2017a), BIM Snippets can be external or internal files, and a BCF issue always includes this information.

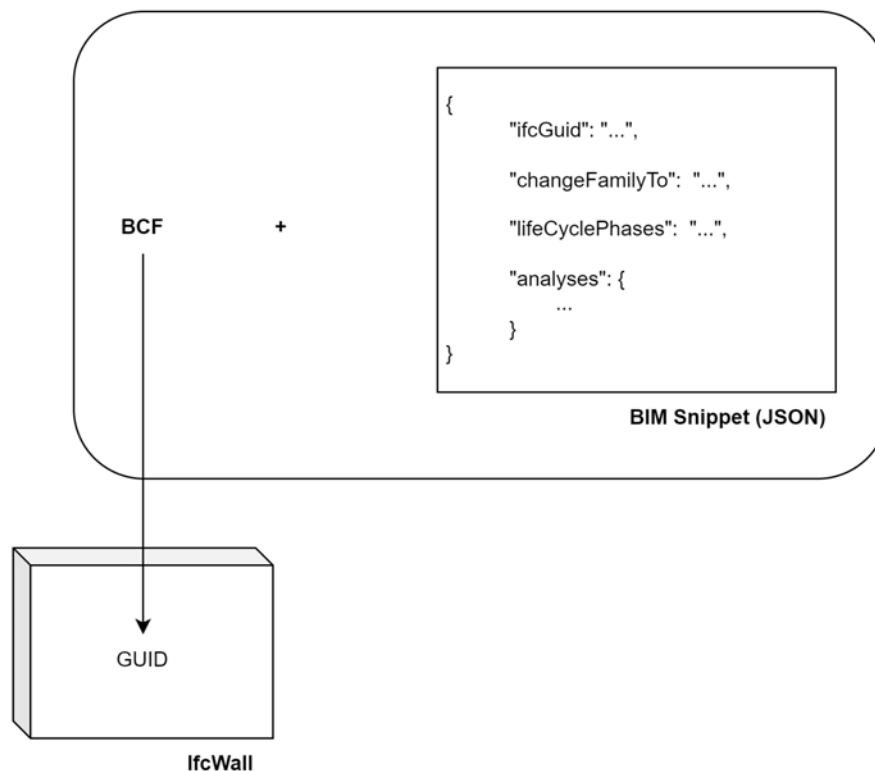


Figure 3.6: The concept of using the BCF with BIM Snippets for communicating design decisions

There are no restrictions regarding the content and the file type of BIM Snippets (BUILDINGSMART, 2022). Figure 3.6 depicts the concept of communicating design decisions using BIM Snippets. As part of the methodology in this thesis, a BIM Snippet is a JSON file attached to the BCF issue. There are four main elements in the JSON file.

- *ifcGuid* is the GUID of the referenced IFC component. BCF issues include this information in the viewpoints. However, it is added to the JSON file for easy access when updating BIM models.
- *changeFamilyTo* is the design decision, the name of the selected design variant.
- *lifeCyclePhases* includes information about the life cycle phases taken into consideration in LCA and LCC.
- *analyses* used in comparing different design variants are also included in BIM Snippets to provide an overview of the rationale behind the decision.

Using BCF-API with the abovementioned concept is advantageous because it informs project stakeholders about updating BIM models and provides documentation of design decisions and analyses. This way, analyses' reruns are minimized since BCF issues referencing IFC components are always retrievable from the database, with BIM snippets attached. Furthermore, since BIM Snippets are in JSON format, they are not only human-readable but can also be read by any other software. Considering that AEC projects involve various stakeholders and software products, this information can also be used in other software to meet different needs.

3.4.2 Updating BIM Models

AEC projects include stakeholders from different disciplines and require a variety of software products. The open BIM model, the IFC model, is the basis for design decisions. However, the model in the BIM authoring tool is updated regularly. Therefore, each building component in the BIM authoring tool store the GUID of the corresponding IFC component. This way, the connection between the IFC components and the components of the closed BIM model is established.

The updating process can be manual, semi-automated, and automated. In the manual updating process, modelers receive design decisions in BIM Snippets, create the variant types, and change building components to the selected ones. For instance, using the IFC GUID, building components can be found and highlighted. Besides visualization, the semi-automated update uses the IFC GUID to choose and change the model components' type automatically. Furthermore, the fully automated process is beyond this thesis' scope. However, in addition to the semi-automated process, it includes the automatic creation of design variant types in the BIM authoring tool.

As shown in Figure 3.7, both the building component in the BIM authoring tool and the BIM Snippet contains the GUID of the corresponding IFC component. Whether manually or semi-/automatically, the building component that needs to be updated is identified by matching its *IFC_GUID* property with the BIM Snippet' *ifcGuid*.

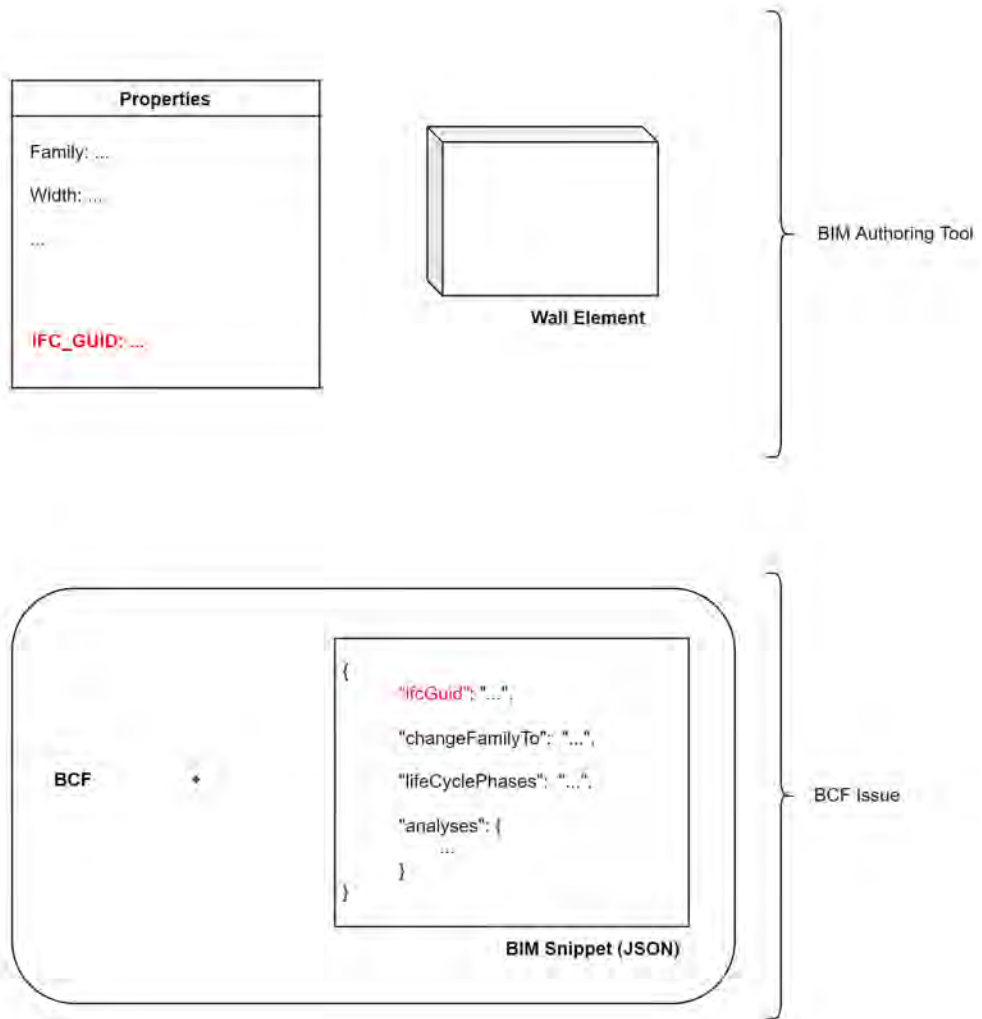


Figure 3.7: The *IFC_GUID* property of the building component matching with the *ifcGuid* of the BIM Snippet

Two conditions must be met for the semi-automated process to work. Firstly, the design variant of the BIM Snippet should exist in the BIM authoring tool with the same name. Secondly, the building component in the BIM authoring tool should correctly store the GUID of its corresponding IFC component.

Chapter 4

Prototypical Implementation & Case Study

This chapter consists of three subchapters. The first chapter introduces the basics of web application development. The second subchapter presents the prototypical implementation, which includes a web application for comparing design variants and making design decisions and a Revit plugin for reviewing BCF issues and updating BIM models. Finally, the last chapter covers the results of a case study with external walls using the web application and the Revit plugin.

4.1 Technical Background: Web Application Development

Web applications and websites run on web browsers. However, in contrast to static content on a website, a web application provides dynamic content based on client-side requests and server-side responses (SHKLAR & ROSEN, 2003).

4.1.1 HTML, CSS and JavaScript

HTML

A markup language, HTML, enables organizing web page content. HTML elements (e.g., images, paragraphs, headings, etc.) are defined using specific tags. For instance, the opening tag `<p>` and the closing tag `</p>` are the tags for creating a paragraph element. HTML tags can include attributes such as *class* and *id*. An HTML file references a CSS file using the *link* tag and a JavaScript file using the *script* tag. Additionally, HTML includes *hypertexts* that allow access to other web pages via direct links. (MDN, 2022c)

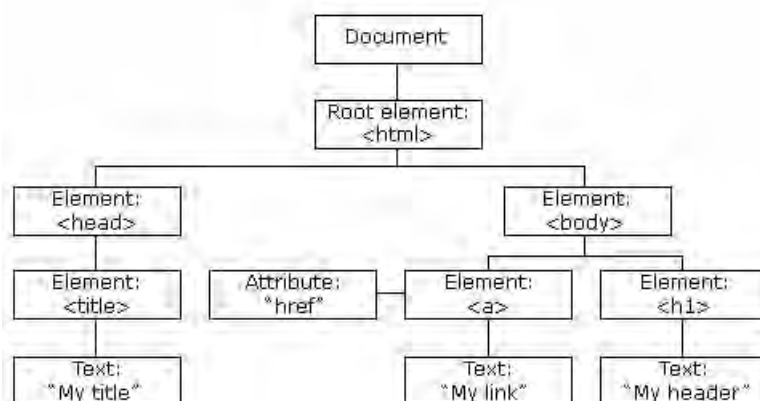


Figure 4.1: The DOM of a web page (W3SCHOOLS, n.d.)

An essential concept, the Document Object Model ([DOM](#)), provides the structure of a web page. Web page content can be accessed and manipulated using the DOM. Figure 4.1 depicts the DOM with its components: the *Document* object (i.e., the web page) and HTML elements. (W3SCHOOLS, n.d.)

CSS

A web page's layout can be modified using CSS. There are two components of CSS: selectors, which specify the elements to be adjusted, and properties, which set the styles. Rather than using the tag name of HTML elements, selecting components from a web page using their attributes, such as *class* and *id*, is possible. (MDN, 2022b)

JavaScript

According to a survey by Stack Overflow (2022), with 65.36%, JavaScript is the most preferred programming language. JavaScript is an interpreted language, so code is executed line by line. The primitive data types in JavaScript are *null*, *undefined*, *string*, *number*, *bigint*, *boolean*, and *symbol*, and the non-primitive data type is *objects*.

JavaScript enables adding interactive features to web pages. Using JavaScript, it is possible to manipulate the DOM of a web page, such as changing its style, adding functionalities, etc.

Web browsers execute JavaScript code using their JavaScript engines. For instance, the JavaScript engines of some web browsers are V8 from Google Chrome, Chakra from Internet Explorer, and SpiderMonkey from Mozilla Firefox. There were interoperability problems after JavaScript was released in 1995 because there was no standard way of implementing JavaScript on web browsers, such as those from Microsoft, Netscape and Sun; consequently, it resulted in the standardization named ECMAScript (WIRFS-BROCK & EICH, 2020).

4.1.2 Full-stack development

Full-stack development comprises front-end and back-end development. Users interact with a web application in the front end. HTML, CSS, and JavaScript, as well as JavaScript front-end frameworks such as React and Angular, are commonly used for front-end development. The back end of a web application represents the server side, which is responsible for responding to the requests from the front end. Programming languages such as PHP, C++, Python, and JavaScript are common in back-end development. Besides, Node is a JavaScript runtime environment commonly used with the Express framework for back-end development.

HTTP

Web browsers and servers communicate via Hypertext Transfer Protocol ([HTTP](#)). There are four main methods to send a request from web browsers: *GET*, *POST*, *PUT*, and *DELETE*. *GET* retrieves data from the server, whereas *POST* adds new data. Furthermore, *PUT* updates data on the server, and *DELETE* removes data.

Asynchronous JavaScript and XML ([AJAX](#)) is a method for retrieving data from servers asynchronously, preventing the web page from making a complete reload when the server responds. The web application instead updates only parts related to the client's request. Using AJAX, it is possible to create single-page web applications which exchange less data with servers and provide a better user experience. AJAX is implemented through the Fetch API. (MDN, 2022a)

JavaScript Object Notation ([JSON](#)) and Extensible Markup Language ([XML](#)) are two standard formats for data exchange between a web browser and a server. JSON utilizes quotation marks and curly brackets instead of tags in XML. According to the research by Nurseitov et al. (2009), JSON showed a significantly better performance in data transmission concerning the required time. Although the average memory utilization was similar for both formats, JSON showed a better performance in system CPU utilization and XML in user CPU utilization.

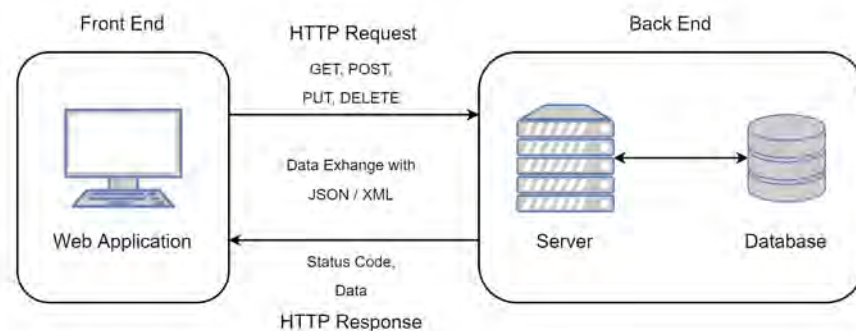


Figure 4.2: Interaction between the front end and the back end via HTTP

Figure 2.14 depicts the communication between a web browser and a server with the standard data exchange formats. Additionally, in his doctoral dissertation, Fielding (2000) introduced a software architecture style called Representational State Transfer ([REST](#)) and defined several constraints with which RESTful web services should comply.

4.2 Prototypical Implementation

The prototypical implementation consists of front-end and back-end applications. The central element in Figure 4.3, the web application is an open BIM platform that serves purposes such as visualizing IFC models, comparing design variants, and making design decisions. There are two back-end applications: one for retrieving LCA and LCC results for different design variants and another for creating and retrieving BCF issues. Furthermore, the Revit plugin enables modelers to access the BCF issues from Revit, the BIM authoring tool by Autodesk. The web application and the Revit plugin communicate with the same server to retrieve the BCF issues containing design decisions in BIM Snippets. Therefore, the BCF database serves as a repository for project stakeholders' decisions, which are then accessible to modelers via communication with the server.

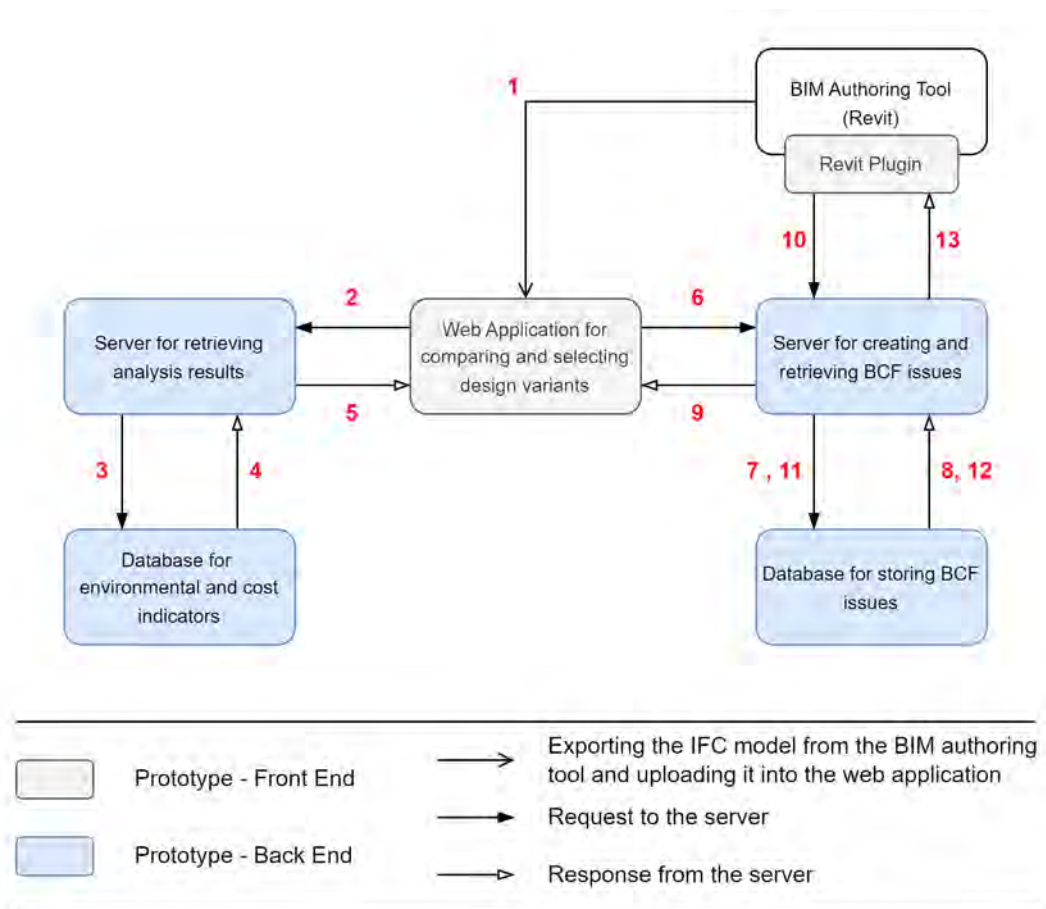


Figure 4.3: General schema of the prototypical implementation

When deciding on a design variant, the numbers highlighted in red in Figure 4.3 occur in the correct order. The following is a detailed explanation of each step.

- 1: The IFC model is exported from the BIM authoring tool and uploaded into the web application.

The web application visualizes the IFC model in a 3D view.

- 2: A request for comparing different design variants through LCA and LCC is made to the server using a user interface (UI) element (e.g., a button). The request contains the imported IFC model.
- 3, 4: The server retrieves data from environmental and cost databases.

The retrieved data is then utilized in the analysis server with the IFC model to conduct LCA and LCC for different design variants.

- 5: The response contains the analysis results in JSON format.

The web application visualizes the results, with which project stakeholders can observe each design variant's environmental and economic impact.

- 6: The server receives a request for creating a new BCF issue, which contains the design decision in the BIM Snippet.
- 7, 8: The server adds the new BCF issue to the BCF database.
- 9: The server responds to the request from the web application with the successful status code.
- 10: The Revit plugin requests all the BCF issues stored in the BCF database.
- 11, 12: The server retrieves the BCF issues from the BCF database.
- 13: The Revit plugin receives the response containing all the BCF issues.

The Revit plugin displays the details of each BCF issue with its BIM Snippet. Upon initiation by the user using a UI element, the Revit plugin updates the model.

The elements of a Revit model are organized hierarchically. Categories are the top level of the hierarchy, consisting of Model Categories (e.g., walls, doors, etc.) and Annotation Categories (e.g., texts, dimensions, etc.). The next level in the hierarchy is Families, which groups items with similar properties. There are three types of families: System Families, Component Families, and In-Place Families. In Revit, predefined families are called System Families. Component Families and In-Place Families are the custom ones that users can create. While Component Families can be imported into multiple Revit projects, In-Place Families are unique to each project. It is possible to extend these families by using Types, which can have different dimensions, material layers, etc. For instance, *Wooden Wall* and *Brick Wall* are two different wall types belonging to the same System Family, *Basic Wall*. (AUBIN, n.d.)

In this prototypical implementation, design variants are different Types in Revit. The Revit plugin updates a component's type concerning the design decision in the BIM Snippet.

The workflow above describes the steps in selecting design variants and updating BIM models. There are other requests/responses between the front-end and back-end applications, for example, logging in to the web application/the Revit plugin and retrieving all the BCF issues to the web application. In the following sections, these features are explained in more detail.

4.2.1 Back End

Analysis Server and Database

The prototypical implementation for BIM-based LCA and LCC calculation by Lammers (2021) is in Python. The code retrieves environmental and cost data from different databases and calculates the results of the indicators Price, GWP, PERT, PENRT, and PE for each design variant using the retrieved data and the IFC model. The implementation by Lammers (2021) provides three design variants, each containing different options for each of the nine model categories. These model categories are non-/load-bearing external and internal walls, external and internal doors, windows, stairs, and floors. This thesis integrates his code into the analysis server and utilizes the design variants from his implementation.

The server is coded in JavaScript using Node (version 16.13.2). Table B.1 in Appendix B contains the details about the packages used for implementing the server. The server provides three URLs, described with their HTTP methods below.

- POST /ifc/upload: IFC models uploaded in the front end are sent to the server and stored in the back end. In this prototypical implementation, the request with the URL results in adding IFC files to a local folder in the back end. Nevertheless, in real-life web applications, a database can store IFC models by assigning a unique identifier (ID) to each model.
- GET /ifc/analyses/{fileName}/{variant}: There are two inputs required for this URL: the file name (i.e., *fileName*) and the variant number (i.e., *variant*). Firstly, the server searches an IFC model with the file name. Due to the high likelihood of different IFC files having the same name in real-life web applications, it is more convenient to use a unique ID. This thesis, however, relies on file names to simplify the implementation. Secondly, depending on the number from one to three, the server runs the code implemented by Lammers (2021) for the corresponding design variant. The front end loops over this request and retrieves the results for all three design variants. Figure 4.4 shows the content of the response from the server. A design variant consists of several model categories, each with LCA and LCC results.
- GET /download/revit/externalWallTypes: The server responds to a request with a Revit project file containing all the design variants for external walls.

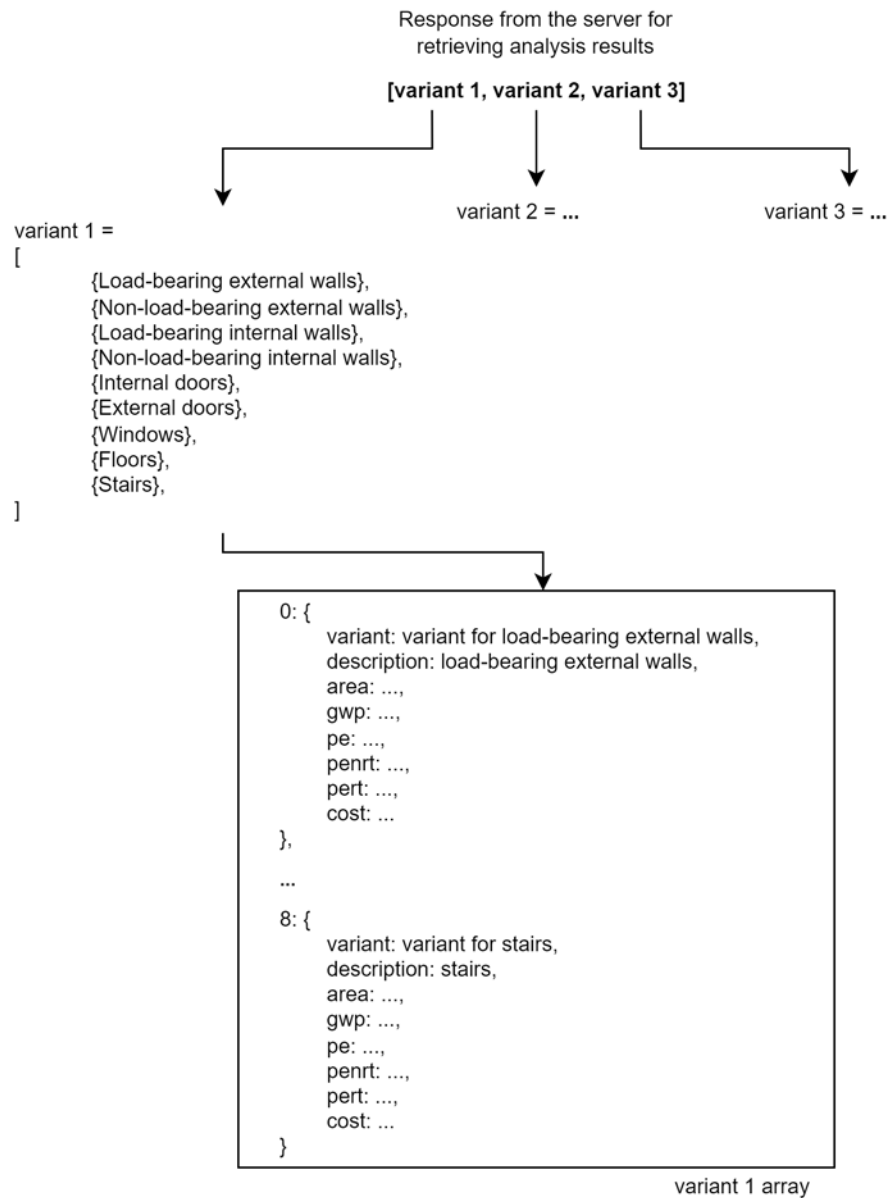


Figure 4.4: Response from the server for analyzing design variants through LCA and LCC

BCF Server and Database

The documentation of BCF-API by buildingSMART (2017a) provides URLs with their HTTP methods to create RESTful APIs. Using version 2.1, Schulz (2020) implemented a server in JavaScript with Node. The BCF server in this thesis is based on his implementation, and BCF issues are created and retrieved via this local server.

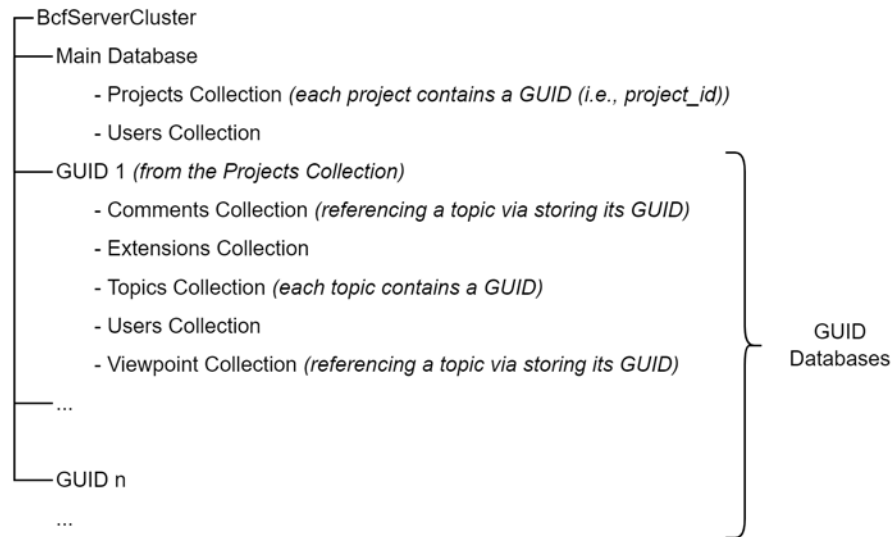


Figure 4.5: Database structure, adapted from (SCHULZ, 2020)

In addition, this thesis also uses the MongoDB database structure that Schulz (2020) provided as part of his implementation. As depicted in Figure 4.5, the cluster consists of two databases: the main database, which stores projects and users, and the database named with the GUID of a project, which includes all project-specific information regarding BCF issues. The implementation by Schulz (2020) creates this structure automatically. It is necessary to create an account in MongoDB, which is a cloud-based database.

The case study (Section 4.3) includes only a single project in the database. Therefore, there is only one GUID database. The Users Collection has two user types: modeler and project stakeholder. A topic's parameters can have constraints defined in the Extensions Collection. Only JSON and XML formats are allowed as a type of BIM Snippets.

According to buildingSMART (2017a), BCF issues include a parameter named *bim_snippet*, which has the following sub-parameters: *snippet_type*, *is_external*, *reference*, and *reference_schema*. Table 4.1 shows the content of these sub-parameters. Besides, A BIM Snippet can be an external file and referenced via a link. In this prototypical implementation, the snippet file is not external because it is attached to the BCF issue. Therefore, another sub-parameter *file* is added to *bim_snippet* to store the snippet within the BCF issue.

Table 4.1: The sub-parameters of a BIM Snippet, adapted from (BUILDINGSMART, 2017a)

Sub-paramater	Type	Explanation by buildingSMART (2017a)
snippet_type	string	„Type of a BIM-Snippet of a topic“
is_external	boolean	„Is the BIM-Snippet external (default = false)“
reference	string	„Reference of a BIM-Snippet of a topic“
reference_schema	string	„Schema of a BIM-Snippet of a topic“

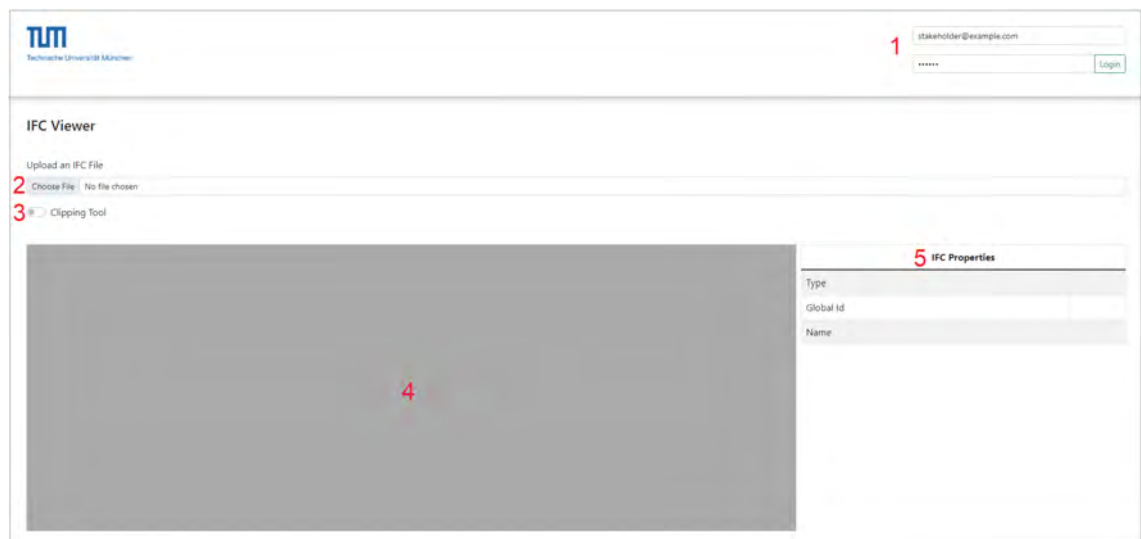
Schulz (2020) did not implement BIM Snippets, so his implementation was modified by adding the parameter *bim_snippet* to topics, and providing two APIs for BIM Snippets, as described in the documentation by buildingSMART (2017a). Below are the URLs and the HTTP methods of the BIM Snippet APIs.

- GET `/bcf/version/projects/project_id/topics/guid/snippet`: „Retrieves a topic's BIM-Snippet as binary file“ (BUILDINGSMART, 2017a).
- PUT `/bcf/version/projects/project_id/topics/guid/snippet`: „Puts a new BIM Snippet binary file to a topic. If this is used, the parent topics BIM Snippet property is_external must be set to false, and the reference must be the file name with extension“ (BUILDINGSMART, 2017a).

4.2.2 Front End

Web Application

The web application includes the following features: visualizing IFC models in a viewer, comparing design variants through LCA and LCC, and creating and displaying BCF issues. The code is implemented in JavaScript using the React framework (version 18.1.0) and bundled into a single file using Rollup (version 2.74). Table B.2 in Appendix B provides a detailed overview of the packages used to implement the front-end application.

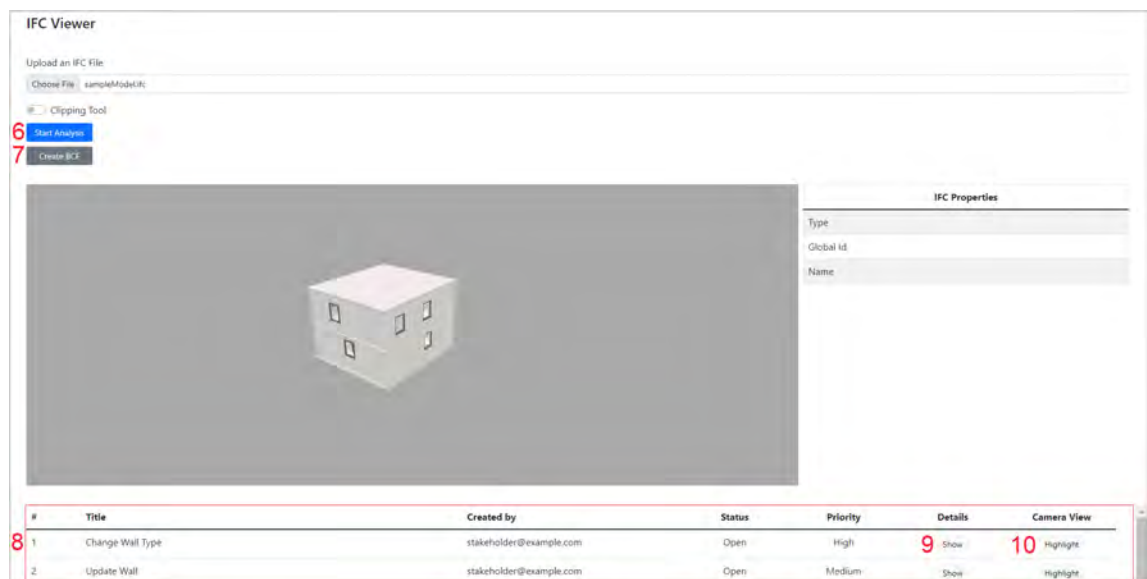


- 1: text fields for entering login credentials and a button for logging in
- 2: a button for selecting and uploading an IFC file
- 3: a toggle switch for de-/activating the clipping feature
- 4: a model container
- 5: a table showing the properties of the selected IFC element

Figure 4.6: UI elements in the web application before logging in and uploading an IFC model

The feature for visualizing IFC models consists of four UI elements: a button for uploading IFC models, a container displaying the uploaded IFC model, a toggle switch for de-/activating the clipping feature, and a table showing the properties of the selected IFC element. The button for uploading files opens a new window to choose an IFC model from a local folder. After uploading, the container displays the IFC model and enables the user to inspect the model with the rotation feature in a 3D view. Activating the clipping feature with the toggle switch allows the user to create a plane surface by double-clicking on the desired location in the IFC model. The plane surface can then be moved up and down along its vertical axis to clip the model. Besides, clicking on or hovering over an IFC element highlights the element. The table next to the model container provides information about the selected IFC element.

As a part of BCF-API, Schulz (2020) implemented the user authentication feature. In his implementation, the server assigns a token to the user upon logging in. Subsequent requests to the server should include this token so that the server identifies the user. In this prototype, the front-end application utilizes this feature by providing UI elements to enable users to enter their credentials to log in. If the server's response is successful, the front-end application stores the token assigned to the user and includes it in future requests. Figure 4.6 shows the general layout of the web application before uploading an IFC model and logging in.



6: a button for conducting analysis

7: a button for creating a BCF issue

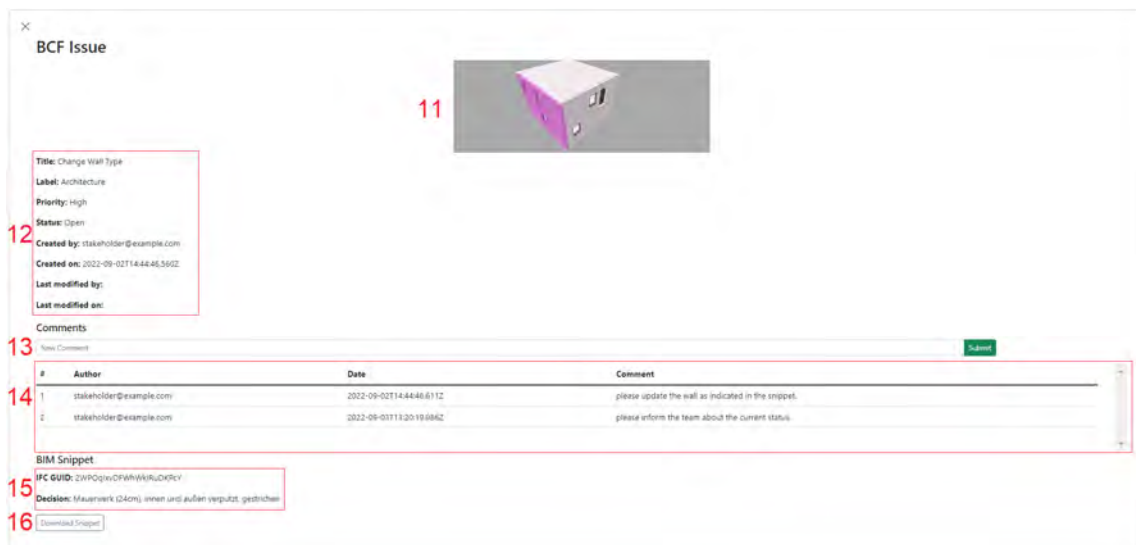
8: a table showing all the BCF issues stored in the database

9: a button for showing the details of the BCF issue

10: a button for highlighting the IFC element referenced by the BCF issue

Figure 4.7: UI elements in the web application after logging in and uploading an IFC model

Upon logging in to the web application and uploading an IFC model, two buttons become available: one for conducting LCA and LCC for design variants and another for creating BCF issues. In addition, the front-end application automatically sends a request to retrieve all the BCF issues stored in the database and displays them in a table under the model container. Selecting an IFC element applies a filter to the table by showing only the BCF issues that reference the selected element. The table includes two buttons: the *Show* button and the *Highlight* button. The first button opens a new modal window showing the BCF issue's details, and the latter selects the IFC element referenced in the BCF issue and highlights it in the model container. Figure 4.7 shows the web application with the additional UI elements available upon uploading an IFC model and logging in. The IFC model in Figure 4.7 is the model used by Lammers (2021) to conduct a case study.



11: a screenshot

12: general information about the BCF issue

13: a text field and a button for adding new comments

14: a table showing all the comments stored in the database

15: information about the BIM Snippet

16: a button for downloading the BIM Snippet

Figure 4.8: Modal window showing the details of a BCF issue

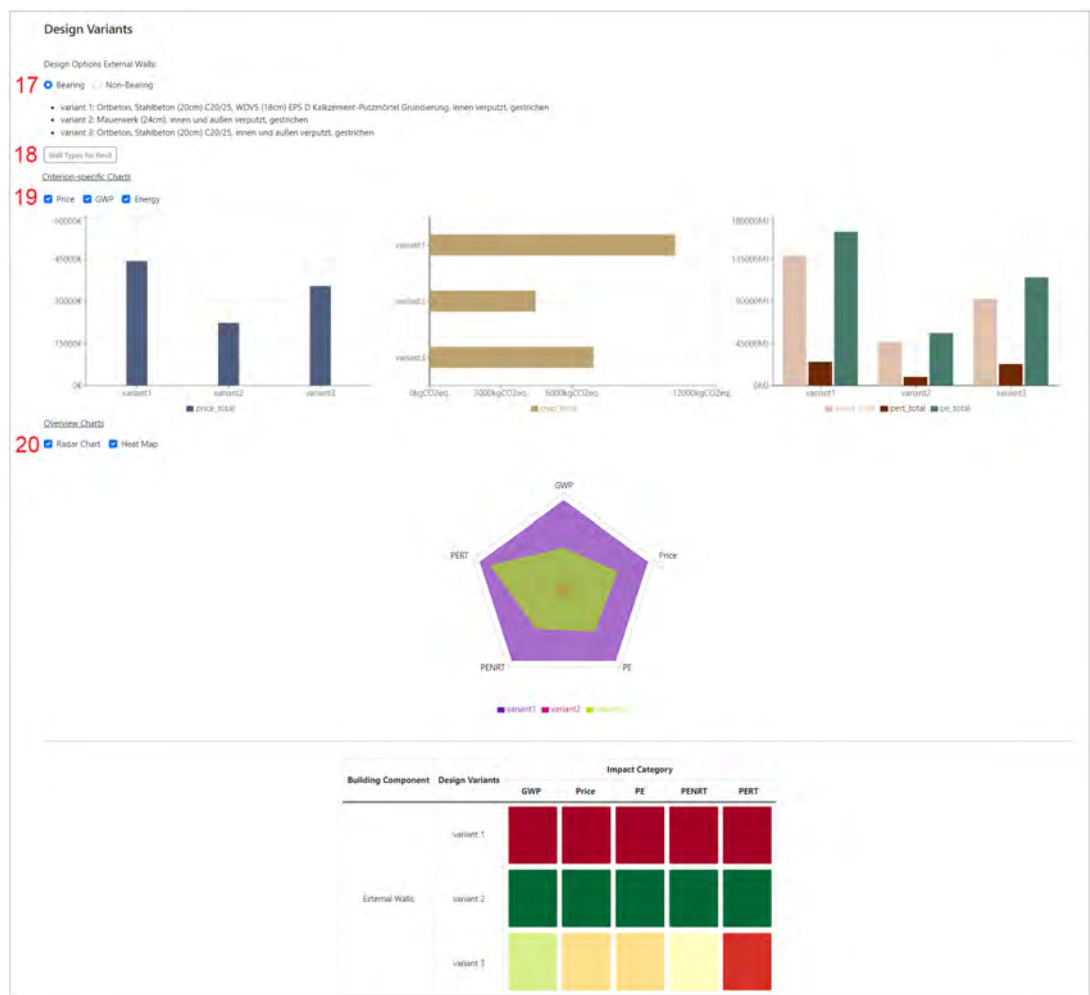
The modal window, available by clicking on the *Show* button, consists of the following details of a BCF issue: general information (e.g., title, label, status, etc.), a screenshot, comments, and information about the BIM Snippet. The text field and the *Submit* button enable users to add new comments to the BCF issue. In addition, using the *Download Snippet* button, users can download the snippet file. Figure 4.8 shows the modal window with its components.

The button shown with the number six in Figure 4.7 starts the analysis. As depicted in Figure 4.4, a response from the analysis server includes the results for different model categories. However, for simplicity, the front-end application visualizes only the results for

non-/load-bearing external walls. Table 4.2 shows the charts used in the web application to visualize the results for the indicators Price, GWP, PERT, PENRT, and PE. Figure 4.9 illustrates these chart types for load-bearing external walls. The color schema of the heat map shown in Figure 4.9 represents the lowest value as green, the mid-value as yellow, and the highest value as red.

Table 4.2: Chart types used in the web application

Chart Group	Indicator(s)	Chart Type
Indicator-specific	Price	Vertical bar chart
	GWP	Horizontal bar chart
	PERT, PENRT, PE	Grouped bar chart
Overview	Price, GWP, PERT, PENRT, and PE	Radar chart
	Price, GWP, PERT, PENRT, and PE	Heat map



- 17: radio buttons for filtering the results concerning non-/load-bearing external walls
- 18: a button for downloading the Revit project file, which contains the design variants
- 19: checkboxes for un-/hiding the criterion-specific charts
- 20: checkboxes for un-/hiding the overview charts

Figure 4.9: UI for showing the analysis results

The web application contains the following UI elements for displaying the analysis results: radio buttons and checkboxes. While the radio buttons allow users to filter the results concerning the options for load-bearing or non-load-bearing external walls, users can un-/hide the charts with the checkboxes. In addition, users can download the Revit project file, which contains the design variants for non-/load-bearing external walls, and import it into their projects to create design variants automatically. Figure 4.9 depicts the UI for showing the analysis results.

21: a text field for entering a title

22: a text field for entering a comment

23: a dropdown list for selecting a label

24: a dropdown list for selecting priority

25: a dropdown list for selecting a design variant

26: a button for submitting the the form

Figure 4.10: Modal window to create a BCF issue

The last feature of the web application is creating a BCF issue with a BIM Snippet. A BCF issue is created by first selecting an IFC element, then clicking on the button shown with the number seven in Figure 4.7. As a result, a modal window appears, which contains form elements. In the modal window, a user should enter a title and a comment to the text fields and select priority and a label from the dropdown lists. Moreover, to attach the BIM Snippet to a BCF issue, choosing the design variant from the dropdown list is required. The web application automatically detects if the external wall is load-bearing or not and displays the design variants accordingly. Figure 4.10 shows the modal window to create a BCF issue.

Plugin

The plugin for the BIM authoring tool displays information about the BCF issues stored in the database with their BIM Snippets. Moreover, the user can update the model concerning

the decision in the BIM Snippet, download the snippet file and add new comments to the BCF issue.

The plugin is implemented in Revit (version 2022) using C# and the Revit API. Revit is the BIM authoring tool in this prototypical implementation since it is widely used in the AEC sector and has extensive online documentation, particularly about its API. Table B.3 in Appendix B shows the packages used to implement the Revit plugin.

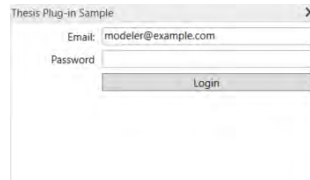


Figure 4.11: The login page of the Revit plugin

The login page is the first page users see. It includes two text fields, where users can enter their credentials, and a button, which sends a request to the server to log in. Figure 4.11 shows the login page with its UI elements.



- 1: information about the user
- 2: a table showing all the BCF issues
- 3: information about the BIM Snippet
- 4: a button for downloading the BIM Snippet
- 5: a button for updating the model
- 6: a button for changing the status of the topic
- 7: a table showing all the comments
- 8: a text field and a button for adding new comments

Figure 4.12: The second page of the Revit plugin displayed upon logging in

As soon as the server sends a successful response, the plugin displays another page that contains a table showing all the BCF issues stored in the database. Upon selecting a BCF issue from the table, Revit zooms into and highlights the corresponding model component. Also, the front-end application retrieves the comments and the BIM Snippet from the server. The comments are displayed in another table, and users can add new comments to the BCF issue with the text field and the button available under the table. The Revit plugin displays only the parameters *IfcGuid* (i.e., the GUID of the corresponding IFC element) and *changeFamilyTo* (i.e., the selected design variant) of the BIM Snippet. However, users can download the snippet file with its full content using the *Download*

Snippet button. In addition, two other buttons are available: the *Apply Changes* button, which updates the model concerning the decision in the BIM Snippet, and the *Close Topic* button, which changes the topic's status to *Closed*. Figure 4.12 shows the layout of the plugin following successful login.

The *Apply Changes* button triggers the Revit API method *changeTypeId*, which changes the type of model components. Table 4.3 shows the arguments for this method. The argument *elementIds* contains the IDs of the elements that need to be updated, and the argument *typeid* is the ID of the type to be applied to the elements (TALARICO, 2022).

Table 4.3: The arguments of the *changeTypeId* method, adapted from (TALARICO, 2022)

Argument	Type	Type Explanation (TALARICO, 2022)
<i>document</i>	Autodesk.Revit.DB Document	„represents an open Autodesk Revit project“
<i>elementIds</i>	System.Collections.Generic ICollection<ElementId>	„a collection of element IDs“
<i>typeid</i>	Autodesk.Revit.DB ElementId	„a unique identification for an element within a single project“

As explained in Section 3.2.2, the model components in the BIM authoring tool store the attribute *IFC_GUID* (i.e., the GUID of their corresponding IFC elements). Thus, by matching the *ifcGuid* of the BIM Snippet with the *IFC_GUID* of the model component, it is possible to find the model component that needs to be updated. In Revit, if the appropriate option is selected in the custom IFC export settings, the model components store this attribute automatically.

Algorithm 4.1: The function returning the elementIds

```

1 List <ElementId> getElementIdOfWall( string snippetIfcGuid , ICollection <Element> walls )
2 {
3     List <ElementId> elementIds = new List <ElementId > ();
4
5     foreach (Element w in walls)
6     {
7         string parameter = w.get_Parameter( BuiltInParameter .IFC_GUID ) ?. AsValueString ();
8
9         if (parameter != null && parameter == snippetIfcGuid)
10        {
11            ElementId wId = w.Id ;
12            elementIds.Add(wId);
13        }
14    }
15    return elementIds;
16 }
17

```

As shown in Algorithm 4.1, the function *getElementIdOfWall* receives two arguments: the *ifcGuid* from the BIM Snippet as a string and all the walls in the Revit project as a collection. The function iterates over all the wall elements. If the parameter *IFC_GUID* exists and equals the *ifcGuid* from the BIM Snippet, the wall's ID is added to the list of

elementIds, which the function returns at the end. In this prototypical implementation, the list of *elementIds* contains only a single ID because each BCF issue references only a single element.

Algorithm 4.2: The function returning the type ID

```

1 ElementId getElementIdOfWallType(string snippetSelection , ICollection<Element> wallTypes)
2 {
3     ElementId typeId = null;
4
5     foreach (ElementType wt in wallTypes)
6     {
7         if (wt.Name == snippetSelection)
8         {
9             typeId = wt.Id;
10        }
11    }
12    return typeId;
13 }
14

```

Algorithm 4.2 shows the code for the function *getElementIdOfWallType*, which receives two arguments: the *changeFamilyTo* parameter from the BIM Snippet as a string and all the wall types in the Revit project as a collection. The function compares each wall's type with the selected type from the BIM Snippet. If they are equal, the variable *typeId* stores the id of the wall type, and the function returns it as a result.

The method *changeTypeId* updates a model component using the results of *getElementIdOfWall* and *getElementIdOfWallType*. Although the front-end applications of this prototypical implementation are focused on the wall elements, the algorithms can be applied to every model component.

In addition, the *Apply Changes* button inserts the analysis results into the corresponding parameters of the wall type. The IFC documentation by buildingSMART (2017b) provides property sets for environmental indicators (*Pset_EnvironmentalImpactIndicators* and *Pset_EnvironmentalImpactValues*). However, importing these property sets into Revit and exporting an IFC model did not work as expected. Although the values of these property sets were available in the Revit model, the exported IFC model did not contain all the values. Therefore, a custom property set was created in Revit to store the analysis results within the wall type. Table 4.4 provides the parameters of the custom property set *LCA_LCC_indicators*.

Table 4.4: Parameters of the property set *LCA_LCC_indicators*

Parameter	Type	Unit
LifeCyclePhase	Text	-
GwpPerUnit	Number	kgCO2 eq./m ²
PePerUnit	Number	MJ/m ²
PertPerUnit	Number	MJ/m ²
PenrtPerUnit	Number	MJ/m ²
CostPerUnit	Number	EUR/m ²

4.3 Case Study

The case study was conducted in Revit (version 2022) using the BIM model by Lammers (2021). The implementation to conduct LCA and LCC for design variants requires the BIM model and the exported IFC model to include the following information. Firstly, the IFC common property set *Pset_WallCommon* provides general information on whether a model component is external, load-bearing, etc. Secondly, the model should contain the calculation of the model components' quantities, such as dimensions and area.

The case study defines the following roles: modeler and project stakeholder. While the modeler works with Revit and updates the model, the project stakeholder uses the web application and is authorized to make design decisions. Often, final design decisions are not made by modelers or people involved in the design but rather by the client. Hence, the project stakeholder role in this case study represents the client's perspective.

4.3.1 Exporting IFC Model from Revit

Before exporting the model, the custom property set, as explained in Table 4.4, should be created in Revit. The modeler exports the model as IFC using a custom export setup instead of Revit's default export settings. Based on version IFC 4.0, the custom setup includes the following options: export IFC common property sets (e.g., *Pset_WallCommon*), export base quantities (e.g., dimensions, area), export user-defined property sets (e.g., *LCA_LCC_indicators*), and store the IFC GUID in an element parameter after export.

4.3.2 Visualizing Model

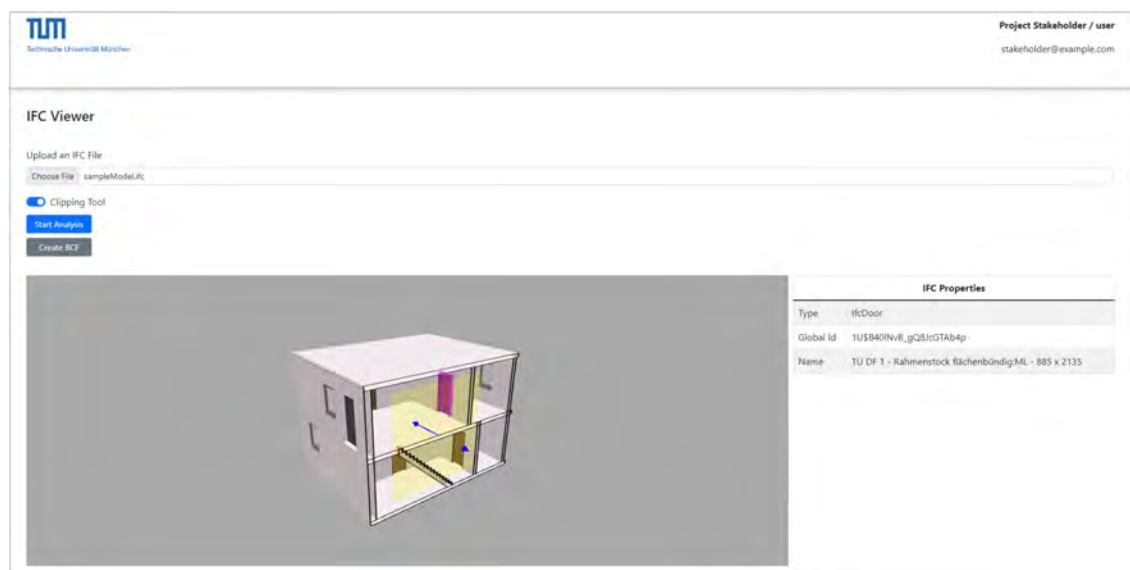


Figure 4.13: Visualizing the IFC model in the web application

The project stakeholder logs in to the web application and uploads the IFC model, which the model container visualizes. The clipping feature enables viewing sections from the model. As soon as the project stakeholder clicks on an IFC element, the *IFC Properties* table shows the following information: *type*, *global ID*, and *name*. If the IFC element consists of the property set *LCA_LL_C_indicators*, the table also indicates this property set's parameters. Figure 4.13 shows the uploaded IFC model and the clipping feature available in the web application. As shown in the figure, the *IFC Properties* table displays information about the selected door element.

4.3.3 Comparing Design Variants for Non-bearing External Walls

The project stakeholder starts the analysis by clicking on the *Start Analysis* button. Once the calculation finishes on the server side and the server responds, the project stakeholder chooses between load-bearing and non-load-bearing external walls using the radio buttons and hides or unhides the charts using the checkboxes.

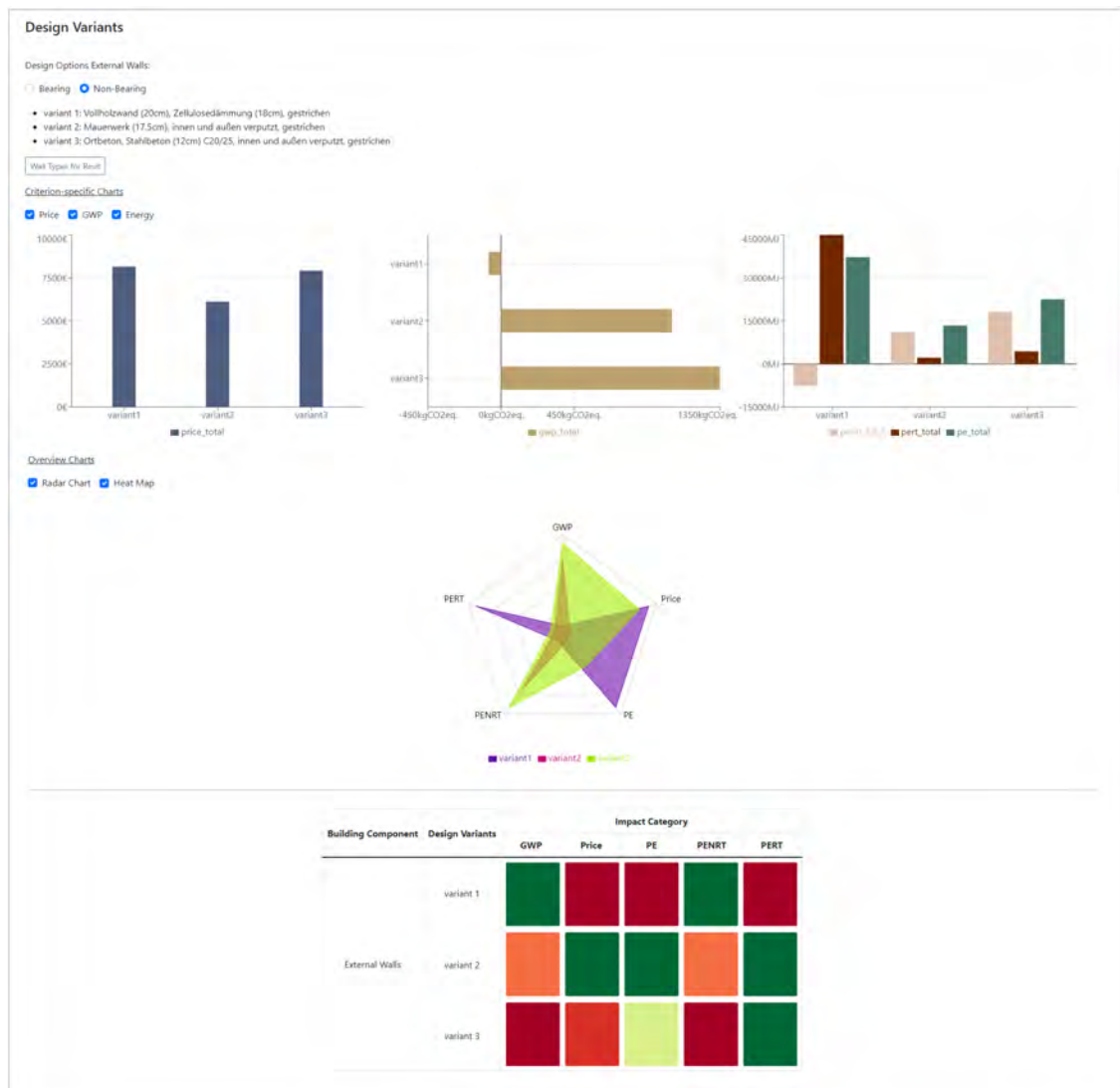


Figure 4.14: Comparison of design variants through LCA and LCC

Figure 4.14 depicts the visualization of the results in the web application for non-load-bearing external walls. The data in the charts display the results by considering all the external walls in the IFC model. The results are summarized as follows.

- The first design variant (*Vollholzwand [20cm], Zellulosedämmung [18cm], gestrichen*) has the highest values of Price, PE, and PERT and the lowest values of GWP and PENRT.
- The second design variant (*Mauerwerk [17.5cm], innen und außen verputzt, gestrichen*) has the lowest values of Price, PE and PENRT.
- The third design variant (*Ortbeton, Stahlbeton [12cm] C20/25, innen und außen verputzt, gestrichen*) has the highest values of GWP and PENRT. Although it does not have the lowest value of PERT, the result is very close to the second design variant and relatively low compared to the first design variant.

4.3.4 Design Decision-Making Process

The project stakeholder selects an external wall from the IFC model and clicks on the *Create BCF* button. The information in the *IFC Properties* table changes depending on the selected component. Below is an example of the information about an external wall in the IFC model.

- *Type*: IfcWall
- *Global Id*: 2WPOqlxvDFWhWkiRuDKPgI
- *Name*: Basic Wall:Außenwand, nicht-tragend

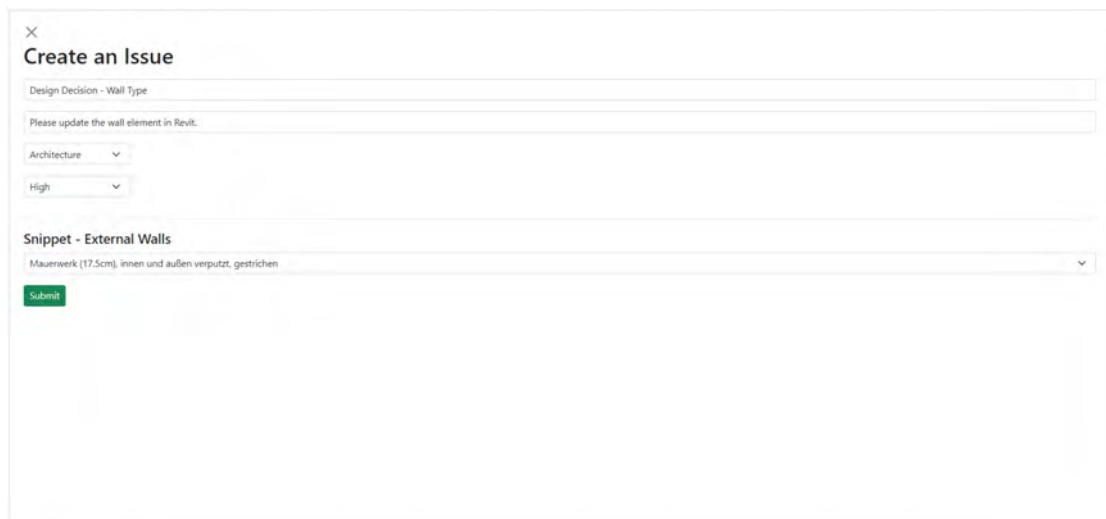


Figure 4.15: The issue form for creating a new BCF issue

After filling out and submitting the issue form shown in Figure 4.15, the BCF issue containing the design decision in the BIM Snippet is sent to the server and saved in the

BCF database. Also, the newly created BCF issue appears in the table shown with the number eight in Figure 4.7. The input from the project stakeholder for creating the BCF issue consists of the following details.

- *Title:* Design Decision - Wall Type
- *Comment:* Please update the wall component in Revit.
- *Label:* Architecture
- *Priority:* High
- *Decision:* Mauerwerk (17.5cm), innen und außen verputzt, gestrichen

4.3.5 Updating Model

The modeler logs in to the Revit plugin and selects the BCF issue from the table, which shows all the BCF issues. Revit highlights the model component, and the plugin shows the comments and the BIM Snippet of the BCF issue. The modeler can download the snippet file or add new comments to the issue.

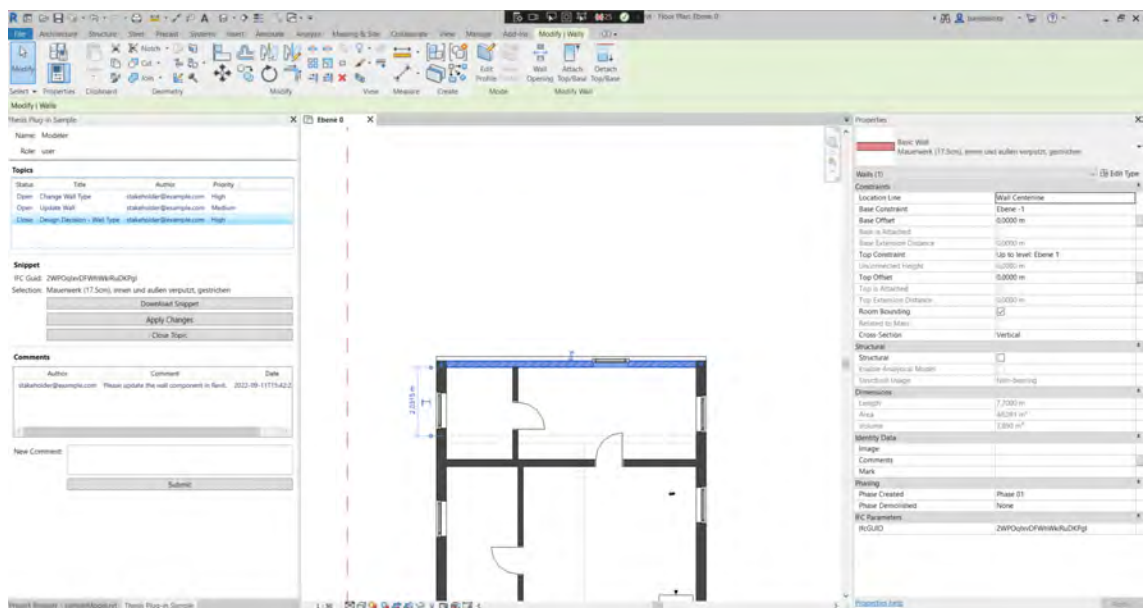


Figure 4.16: Updated Revit model

To update the model by clicking on the *Apply Changes* button, the wall types of the design variants should be present in the Revit project and use the same naming as the web application. The modeler transfers the project standards from the file, which the web application provides by clicking on the button shown with the number eighteen in Figure 4.9. As a result, Revit imports the wall types into the project automatically.

The modeler clicks on the *Apply Changes* button, which changes the model component's type to the one in the BIM Snippet and inserts the analysis results into the wall type's

parameters, which are created with the custom property set *LCA_LCC_indicators*. In addition, the modeler changes the topic's status to *Closed* by clicking on the *Close Topic* button. Figure 4.16 depicts the updated Revit model.

The modeler exports the updated model as IFC. The project stakeholder imports the IFC model into the web application. As shown in Figure 4.17, the name of the model component appears correctly. The analysis results are also displayed in the *IFC Properties* table. Lastly, the status of the BCF issue, the second issue in the table, is *Closed*.

The screenshot shows a web application interface for uploading and analyzing IFC files. At the top, there is a section titled "Upload an IFC File" with a "Choose File" button and a text input field containing "sampleModel_updated.ifc". Below this are buttons for "Clipping Tool", "Start Analysis", and "Create BCF".

In the center, a 3D model of a building is displayed in a grey environment. The building has a white roof and walls, with a prominent purple wall on the right side.

To the right of the 3D model is a table titled "IFC Properties" with the following data:

IFC Properties	
Type	ifcWall
Global Id	ZWPQqwiDFVHWkRuDkPgi
Name	Basic Wall-Mauerwerk (17.5cm), innen und außen verputzt, gestrichen
LifeCyclePhase	A1-A3 & C3-C4
GwpPerUnit	26.230917
PePerUnit	334.950976
PestPerUnit	54.838788
PenrtPerUnit	280.612188
CostPerUnit	152.651954

At the bottom of the interface is a table with the following data:

#	Title	Created by	Status	Priority	Details	Camera View
1	Update Wall	stakeholder@example.com	Open	Medium	Show	Highlight
2	Design Decision - Wall Type	stakeholder@example.com	Closed	High	Show	Highlight

Figure 4.17: Importing the updated model into the web application

Chapter 5

Conclusion

5.1 Review of Research Questions

This master's thesis addressed three research questions. The first research question dealt with developing a visualization strategy for comparing different design options through LCA and LCC (Section 3.3). In this context, two chart groups, criterion-specific and overview charts, are defined. The approaches by Hollberg et al. (2021) and Cerdas et al. (2017) were used to determine visualization methods for each chart group. The second and third questions focused on updating BIM models regarding design decisions based on open BIM. Firstly, to communicate design decisions back to the BIM authoring tool, BCF was extended using BIM Snippets (Section 3.4.1). Secondly, the updating process was automated by storing IFC GUIDs in model components of the BIM authoring tool (Section 3.4.2). Below are the main findings related to each research question.

Research Question I: How can design variants' environmental and economic impact be visualized to support stakeholders in material and component-based decision-making?

Hypothesis: Design variants' environmental and economic impact can be visualized using combined metrics from multicriteria analyses.

Different metrics can be combined using a single chart (e.g., an overview chart) or several charts (e.g., a combination of several criterion-specific charts). This thesis separated the visualization strategy into two chart groups by considering the needs of different stakeholders. Overview charts benefit the client as they simplify the comparison of different design options. Even though the client has greater authority in making design decisions, expert advice might be necessary. For this purpose, criterion-specific charts are beneficial as the exact analysis results can be observed using these charts.

Since criterion-specific charts (see Figures C.1-3 and C.6-8 in Appendix C) are typically different types of bar charts, they do not have an advantage over one another. There are, however, advantages between different overview charts. For instance, radar charts (see Figures C.4, 9 in Appendix C) provide a better overview of the analysis results. On the other hand, by clustering colors together, heat maps (see Figures C.5, 10 in Appendix C) allow for more quick observations—for example, the greener the row of a design variant, the better the performance.

Research Question II: How can design decisions by different project stakeholders be communicated back to the model in the BIM authoring tool?

Hypothesis: Design decisions can be communicated back to the model in the BIM authoring tool by extending open-BIM-based automation workflows.

The approach of extending the BCF through BIM Snippets demonstrates that the BCF can be used to communicate design decisions in addition to its primary use case of commenting on issues in a BIM model. BIM Snippets are in JSON format in the proposed methodology, compatible with various programming languages. Therefore, it is possible to integrate them into various software applications.

Apart from the use case of communicating design decisions, BIM Snippets can be customized for different use cases, as they have no limitations regarding content and format. There are, however, some disadvantages associated with this flexibility. For instance, there are no standards for using BIM Snippets to communicate design decisions, so different custom versions might not be compatible.

Research Question III: What information should be stored in each component of the closed BIM model for the updating process concerning design decisions made in an open BIM workflow?

Hypothesis: The closed BIM model should store the information uniquely present in each model component of the open BIM model.

Each component of the IFC model contains a unique identifier, a GUID. Therefore, storing this information in the closed BIM model allows for finding and updating the corresponding component in the BIM authoring tool. Some BIM authoring tools include this feature by default, making it easy to implement.

The semi-automated method of updating BIM models was developed by combining the approaches presented with the second and third research questions. In addition, as part of the updating process, the analysis results are inserted into the property set of the component types. This feature enriches the semantic information within the BIM model. IFC models contain this information in subsequent exports from the BIM authoring tool, eliminating the necessity of repeating analyses.

5.2 Limitations

The proposed semi-automated method reduces the manual effort for updating BIM models regarding design decisions based on open BIM. However, there are two main limitations, which are described in the following paragraphs.

Firstly, creating design variants (i.e., component types) in the BIM authoring tool involves manual effort, requiring that the variant name in the BIM authoring tool must match the variant name in the BIM Snippet. Automation fails when errors are present, such as naming errors, variants not being available in the BIM authoring tool, etc. The prototypical implementation avoided this problem by allowing users to download a native file containing the design variants from the web application. This native file can be imported into the BIM

authoring tool for creating design variants automatically to avoid errors. Nevertheless, the solution is a part of the prototype, not the method.

Secondly, the semi-automated method relies on another condition: a model component in the BIM authoring tool should store the GUID of its corresponding IFC component. In the updating process, errors occur when the GUID stored in the BIM authoring tool does not match the GUID in the BIM Snippet. For instance, the IFC model is exported from the BIM authoring tool. Design decisions are made for an IFC model component and communicated to the BIM authoring tool. If the corresponding model component in the BIM authoring tool is deleted or remodeled, this example fails to update the BIM model since the BIM Snippet references another GUID. Therefore, the approach is unsuitable for the very beginning of the design phase when the geometry changes frequently.

5.3 Outlook

There is room for improvement in the approach presented in this master's thesis. The following paragraphs explain the possible directions.

To begin with, interacting with IFC models was not considered in the visualization strategy. For instance, in the prototypical implementation, the visualization of IFC models was intended to illustrate the current state of the design, not to compare variants. Therefore, combining the analysis results with the IFC model can be the next step.

Additionally, according to the first limitation in the previous section, design variants are created manually if not imported from a file in the current approach. It may be possible to improve the method by automating the creation of design variants in the BIM authoring tool, for instance, by extending the BIM Snippet concept further. BIM Snippets can contain details about design variants (e.g., the material layers of a component type and their thicknesses), which can then be used to automate the creation of design variants in the BIM authoring tool. Nevertheless, the task remains challenging, as a mapping strategy is necessary to match the materials in the BIM Snippet with those in the BIM authoring tool.

Finally, considering the second limitation described in the previous section, another improvement can be combining this approach with a version control mechanism to keep track of stored IFC GUIDs in the BIM authoring tool. For example, it might be possible to automatically compare IFC GUIDs stored in the BIM authoring tool with those stored in BIM Snippets for each version of the BIM model. Consequently, BCF issues with the BIM Snippets that do not reference a valid GUID can be identified and modified by responsible stakeholders.

Appendix A

Appendix: Project Phases

	Pre-Design		Design				Construction	Handover	In Use	End of Life
	0	1	2		3	4	5	6	7	
RIBA (UK)	Strategic Definition	Preparation and Brief	Concept Design	NOT USED	Developed Design	Technical Design	Construction	Handover & Close Out	In Use	NOT USED
ACE (Europe)	0	1	2.1	2.2	2.3	2.4	3		4	5
	Initiative	Initiation	Concept Design	Preliminary Design	Developed Design	Detailed Design	Construction	NOT USED	Building Use	End of Life
AIA (USA)	NOT USED	NOT USED	Schematic Design	NOT USED	Design Development	Construction Documents	Construction	NOT USED	NOT USED	NOT USED
APM (Global)	0	1	2		3	4	5	6	7	
	Strategy	Outcome Definition	Feasibility	NOT USED	Concept Design	Detailed Design	Delivery	Project Close	Benefits Realisation	NOT USED
Spain	NOT USED	NOT USED	Proyecto Basico	NOT USED	NOT USED	Proyecto de Ejecución	Dirección de Obra	Final de Obra	NOT USED	NOT USED
NATSPEC (Aus)	NOT USED	Establishment	Concept Design	Schematic Design	Design Development	Contract Documentation	Construction	NOT USED	Facility Management	NOT USED
NZCIC (NZ)	NOT USED	Pre-Design	Concept Design	Preliminary Design	Developed Design	Detailed Design	Construct	NOT USED	Operate	NOT USED
Russia	NOT USED	NOT USED	AGR Stage	Stage P	Tender Stage	Construction Documents	Construction	NOT USED	NOT USED	NOT USED
South Africa	NOT USED	1	2	3	-	4	5			
	NOT USED	Inception	Concept and Viability	Design Development	NOT USED	Documentation	Construction	Close Out	NOT USED	NOT USED

Figure A.1: Project phases in different countries/regions (ROYAL INSTITUTE OF BRITISH ARCHITECTS, 2020, p. 9)

Appendix B

Appendix: Development Packages

Table B.1: Packages used in the implementation of the analysis server

Package	Version
cors	2.8.5
downloadjs	1.4.7
express	4.18.1
multer	1.4.4
multer	1.4.4

Table B.2: Packages used in the implementation of the web application

Package	Version
axios	0.27.2
bootstrap	5.1.3
react	18.1.0
react-bootstrap	2.4.0
react-dom	18.1.0
react-modal	3.15.1
reaviz	13.1.6
recharts	2.1.13
tachyons	4.12.0
web-ifc-viewer	1.0.177
@babel/preset-react	7.18.6
@rollup/plugin-babel	5.3.1
@rollup/plugin-commonjs	22.0.0
@rollup/plugin-image	2.1.1
@rollup/plugin-json	4.1.0
@rollup/plugin-node-resolve	13.3.0
@rollup/plugin-replace	4.0.0
rollup	2.74
rollup-plugin-livereload	2.0.5
rollup-plugin-serve	1.1.0
rollup-plugin-styles	4.0.0

Table B.3: Packages used in the implementation of the Revit plugin

Package	Version
Microsoft.AspNet.Mvc	5.2.9
Microsoft.AspNet.Razor	3.2.9
Microsoft.AspNet.WebPages	3.2.9
Microsoft.Web.Infrastructure	1.0.0.0
Newtonsoft.Json	13.0.1

Appendix C

Appendix: Comparison of Design Variants

C.1 Load-bearing External Walls

Table C.1: Design variants for load-bearing external walls

Variant Number	Variant
1	Ortbeton, Stahlbeton (20cm) C20/25, WDVS (18cm) EPS D Kalkzement-Putzmörtel Grundierung, innen verputzt, gestrichen
2	Mauerwerk (24cm), innen und außen verputzt, gestrichen
3	Ortbeton, Stahlbeton (20cm) C20/25, innen und außen verputzt, gestrichen

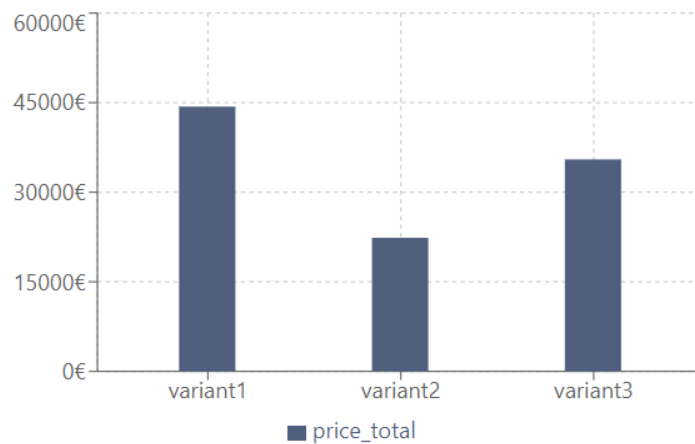


Figure C.1: Analysis results - Price

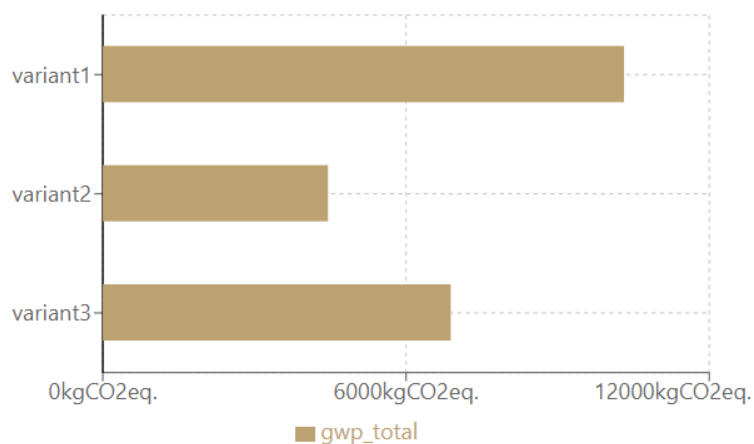


Figure C.2: Analysis results - GWP

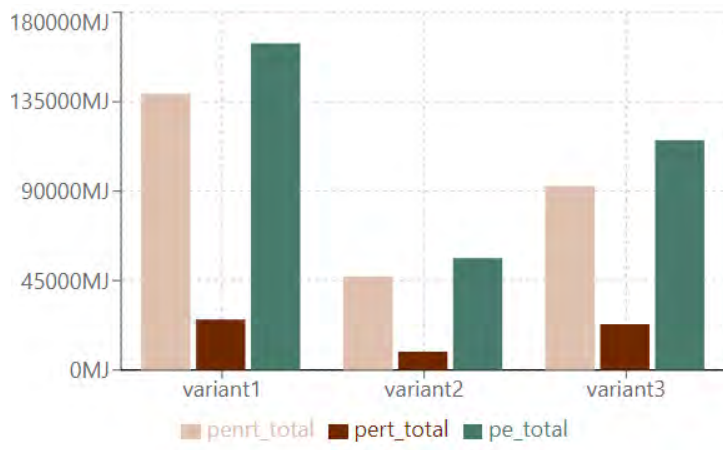


Figure C.3: Analysis results - Energy

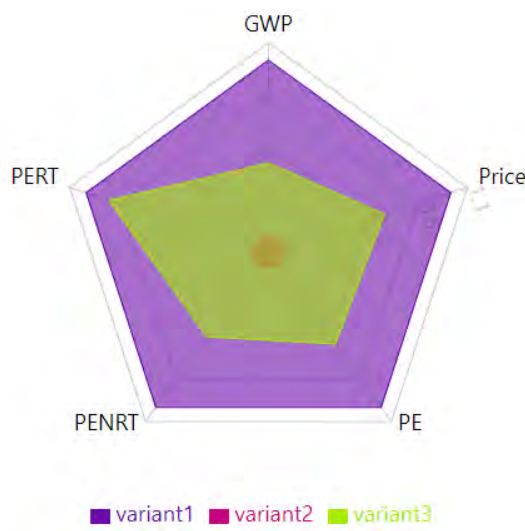


Figure C.4: Overview Chart - Radar Chart

Building Component	Design Variants	Impact Category				
		GWP	Price	PE	PENRT	PERT
External Walls	variant 1	High	High	High	High	High
	variant 2	Medium	Medium	Medium	Medium	Medium
	variant 3	Low	Low	Low	Low	High

Figure C.5: Overview Chart - Heat Map

C.2 Non-load-bearing External Walls

Table C.2: Design variants for non-load-bearing external walls

Variant Number	Variant
1	Vollholzwand (20cm), Zellulosedämmung (18cm), gestrichen
2	Mauerwerk (17.5cm), innen und außen verputzt, gestrichen
3	Ortbeton, Stahlbeton (12cm) C20/25, innen und außen verputzt, gestrichen



Figure C.6: Analysis results - Price

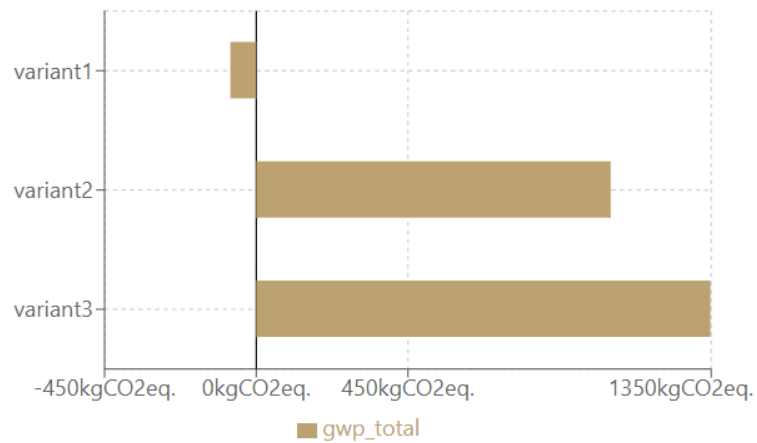


Figure C.7: Analysis results - GWP

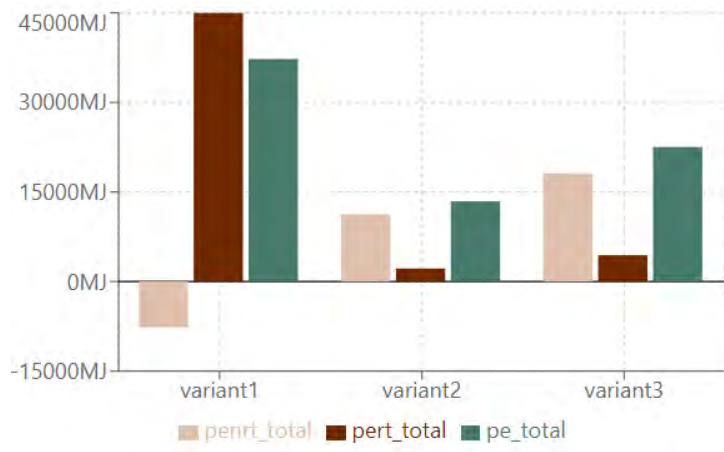


Figure C.8: Analysis results - Energy

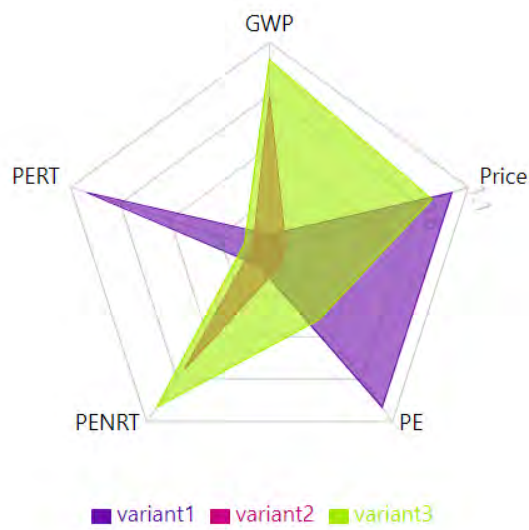


Figure C.9: Overview Chart - Radar Chart

Building Component	Design Variants	Impact Category				
		GWP	Price	PE	PENRT	PERT
External Walls	variant 1	High	High	Low	High	High
	variant 2	Low	Low	High	Low	Low
	variant 3	High	High	Low	High	High

Figure C.10: Overview Chart - Heat Map

Bibliography

- ABUALDENIEN, J., & BORRMANN, A. (2019). A meta-model approach for formal specification and consistent management of multi-LOD building models. *Advanced Engineering Informatics*, 40(April), 135–153. <https://doi.org/10.1016/j.aei.2019.04.003>
- AUBIN, P. (n.d.). *Revit families: A step-by-step introduction*. Autodesk University. Retrieved October 6, 2022, from <https://www.autodesk.com/autodesk-university/article/Revit-Families-Step-Step-Introduction-2018>
- BAHRAMIAN, M., & YETILMEZSOY, K. (2020). Life cycle assessment of the building industry: An overview of two decades of research (1995–2018). *Energy and Buildings*, 219(July). <https://doi.org/10.1016/j.enbuild.2020.109917>
- BIMFORUM. (2021). *Level of Development (LOD) specification part I, guide, & commentary*. BIM Forum. Retrieved June 1, 2022, from <https://bimforum.org/>
- 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 A. BORRMANN, M. KÖNIG, C. KOCH, & J. BEETZ (Eds.), *Building information modeling: Technology foundations and industry practice* (1st ed., pp. 81–126). Springer, Cham. <https://doi.org/10.1007/978-3-319-92862-3>
- BORRMANN, A., KÖNIG, M., KOCH, C., & BEETZ, J. (2018). Building Information Modeling – Why? What? How? In A. BORRMANN, M. KÖNIG, C. KOCH, & J. BEETZ (Eds.), *Building information modeling: Technology foundations and industry practice* (1st ed., pp. 1–24). Springer, Cham. <https://doi.org/10.1007/978-3-319-92862-3>
- BUILDINGSMART. (n.d.-a). *BIM Collaboration Format (BCF) - An introduction*. Retrieved June 12, 2022, from <https://technical.buildingsmart.org/standards/bcf/>
- BUILDINGSMART. (n.d.-b). *Vision and mission*. Retrieved September 19, 2022, from <https://www.buildingsmart.org/about/vision/>
- BUILDINGSMART. (2017a, January 16). *BCF REST API*. https://github.com/buildingSMART/BCF-API/tree/release_2_1
- BUILDINGSMART. (2017b). *Industry Foundation Classes 4.0.2.1 version 4.0 - Addendum 2 - Technical corrigendum 1*. buildingSMART. Retrieved June 6, 2022, from https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/
- BUILDINGSMART. (2022, July 19). *BIM Collaboration Format v3.0 technical documentation*. https://github.com/BuildingSMART/BCF-XML/tree/release_3_0/Documentation
- BUNDESMINISTERIUM FÜR WOHNEN, STADTENTWICKLUNG UND BAUWESEN. (2021, June 25). *Ökobaudat*. <https://www.oekobaudat.de/>
- CERDAS, F., KALUZA, A., ERKISI-ARICI, S., BÖHME, S., & HERRMANN, C. (2017). Improved visualization in LCA through the application of cluster heat maps. In S. TAKATA, Y. UMEDA, & S. KONDOH (Eds.), *Procedia cirp* (pp. 732–737). Elsevier B.V. <https://doi.org/10.1016/j.procir.2016.11.160>

- CONSTRUCTION USERS ROUNDTABLE. (2004). *Collaboration, integrated information and the project lifecycle in building design, construction and operation* (WP-1202 August). CURT. <https://kcuc.org/wp-content/uploads/2013/11/Collaboration-Integrated-Information-and-the-Project-Lifecycle.pdf>
- EUROPEAN COMMISSION. (n.d.). *Construction sector*. Internal Market, Industry, Entrepreneurship and SMEs. Retrieved October 15, 2022, from https://single-market-economy.ec.europa.eu/sectors/construction_en
- EUROPEAN COMMITTEE FOR STANDARDIZATION. (2021). *Sustainability of construction works - Methodology for the assessment of performance of buildings - Part 1: Environmental Performance* (DIN EN 15978-1:2021-09). <https://www.beuth.de/de/norm-entwurf/oenorm-en-15978-1/346084228>
- EUROPEAN INVESTMENT BANK. (2020). *Who is prepared for the new digital age? Evidence from the EIB Investment Survey*. Publications Office. <https://doi.org/10.2867/03951>
- EUROSTAT. (2020). *Generation of waste by economic activity* (Online Data Code: TEN00106). Retrieved October 15, 2022, from https://ec.europa.eu/eurostat/databrowser/view/TEN00106__custom_3595540/default/table?lang=en
- FIELDING, R. T. (2000). *Architectural styles and the design of network-based software architectures* (Doctoral dissertation). University of California, Irvine.
- FORTH, K., ABUALDENIEN, J., BORRMANN, A., FELLERMANN, S., & SCHUNICHT, C. (2021). Design optimization approach comparing multicriterial variants using BIM in early design stages. In C. FENG, T. LINNEN, & I. BRILAKIS (Eds.), *Proceedings of the 38th international symposium on automation and robotics in construction* (pp. 235–242). International Association for Automation; Robotics in Construction (IAARC). <https://doi.org/10.22260/isarc2021/0034>
- GUNDES, S. (2016). The use of life cycle techniques in the assessment of sustainability. *Procedia - Social and Behavioral Sciences*, 216(January 2016), 916–922. <https://doi.org/10.1016/j.sbspro.2015.12.088>
- HOLLBERG, A., KISS, B., RÖCK, M., SOUST-VERDAGUER, B., WIBERG, A. H., LASVAUX, S., GALIMSHINA, A., & HABERT, G. (2021). Review of visualising LCA results in the design process of buildings. *Building and Environment*, 190(March 2021). <https://doi.org/10.1016/j.buildenv.2020.107530>
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. (2006). *Environmental management — Life cycle assessment — Principles and framework* (ISO 14040:2006(E)). <https://iso.org/standard/37456.html>
- INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. (2017). *Buildings and constructed assets — Service life planning — Part 5: Life-cycle costing* (ISO 15686-5:2017(E)). <https://www.beuth.de/de/norm/iso-15686-5/278003880>
- KLÖPFER, W., & GRAHL, B. (2014). *Life Cycle Assessment (LCA): A guide to best practice*. Wiley-VCH.
- LAMMERS, B. (2021). *IFC-based variant analysis considering multicriterial sustainability analysis of buildings* (Master's thesis). Technical University of Munich.
- LIEBICH, T., SCHWEER, C.-S., & WERNIK, S. (2011). *Die Auswirkungen von Building Information Modeling auf die Leistungsbilder und Vergütungsstrukturen für Architekten*

- und Ingenieure sowie auf die Vertragsgestaltung (tech. rep.). Bundesinstituts für Bau-, Stadt- und Raumforschung (BBSR) im Bundesamt für Bauwesen und Raumentwicklung (BBR). Retrieved September 19, 2022, from <https://www.bmvi.de/SharedDocs/DE/Anlage/DG/Digitales/bim-auswirkungen-schlussbericht.html>
- LU, K., JIANG, X., YU, J., TAM, V. W., & SKITMORE, M. (2021). Integration of life cycle assessment and life cycle cost using building information modeling: A critical review. *Journal of Cleaner Production*, 285, 125438. <https://doi.org/10.1016/j.jclepro.2020.125438>
- MDN. (2022a, September 9). *Ajax*. MDN Web Docs. Retrieved October 4, 2022, from <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>
- MDN. (2022b, September 25). *CSS: Cascading Style Sheets*. MDN Web Docs. Retrieved October 4, 2022, from <https://developer.mozilla.org/en-US/docs/Web/CSS>
- MDN. (2022c, September 13). *HTML: HyperText Markup Language*. MDN Web Docs. Retrieved October 4, 2022, from <https://developer.mozilla.org/en-US/docs/Web/HTML>
- NATIONAL INSTITUTE OF BUILDING SCIENCES. (2015). *National BIM Standard - United States version 3*. NIBS. Retrieved May 25, 2022, from <https://www.nationalbimstandard.org/>
- NURSEITOV, N., PAULSON, M., REYNOLDS, R., & IZURIETA, C. (2009). Comparison of JSON and XML data interchange formats: A case study. In D. CHE (Ed.), *Proceedings of the isca 22nd international conference on computer applications in industry and engineering, caine 2009* (pp. 157–162). ISCA. Retrieved July 1, 2022, from https://www.researchgate.net/publication/220922905_Comparison_of_JSON_and_XML_data_interchange_formats_A_case_study/references
- POTRČ OBRECHT, T., RÖCK, M., HOXHA, E., & PASSER, A. (2020). BIM and LCA integration: A systematic literature review. *Sustainability*, 12(14). <https://doi.org/10.3390/su12145534>
- PREIDEL, C., BORRMANN, A., MATTERN, H., KÖNIG, M., & SCHAPKE, S.-E. (2018). Common Data Environment. In A. BORRMANN, M. KÖNIG, C. KOCH, & J. BEETZ (Eds.), *Building information modeling: Technology foundations and industry practice* (1st ed., pp. 279–291). Springer, Cham. <https://doi.org/10.1007/978-3-319-92862-3>
- ROYAL INSTITUTE OF BRITISH ARCHITECTS. (2020). *RIBA Plan of Work 2020 overview*. RIBA. Retrieved June 16, 2022, from <https://www.architecture.com/knowledge-and-resources/resources-landing-page/riba-plan-of-work>
- SACKS, R., EASTMAN, C., LEE, G., & TEICHOLZ, P. (2018). *BIM handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility Managers* (3rd ed.). John Wiley & Sons, Inc.
- SANTOS, R., COSTA, A. A., SILVESTRE, J. D., & PYL, L. (2019). Integration of LCA and LCC analysis within a BIM-based environment. *Automation in Construction*, 103(February), 127–149. <https://doi.org/10.1016/j.autcon.2019.02.011>
- SCHNEIDER-MARIN, P., & ABUALDENIEN, J. (2019). A framework to facilitate an interdisciplinary design process using BIM. In M. STERNAL, L.-C. UNGUREANU, L.

- BÖGER, & C. BINDAL-GUTSCHE (Eds.), *31. forum bauintformatik : 11.–13. september 2019 in berlin. proceedings* (pp. 67–74). Universitätsverlag der TU Berlin. <https://doi.org/http://dx.doi.org/10.14279/depositonce-8763>
- SCHULZ, O. (2020, September 29). *bcfServer*. GitHub. Retrieved August 1, 2022, from <https://github.com/Design-Computation-RWTH/bcfServer>
- SCHULZ, O., & BEETZ, J. (2021). Image-documentation of existing buildings using a server-based bim collaboration format workflow. In J. ABUALDENIEN, A. BORRMANN, L.-C. UNGUREANU, & T. HARTMANN (Eds.), *Eg-ice 2021 workshop on intelligent computing in engineering* (pp. 108–117). Universitätsverlag der TU Berlin. <https://doi.org/10.14279/depositonce-12021>
- SCHULZ, O., ORASKARI, J., & BEETZ, J. (2021). BcfOWL: A BIM collaboration ontology. In M. POVEDA-VILLALÓN & P. PAUWELS (Eds.), *Proceedings of the 9th linked data in architecture and construction workshop - Idac2021* (pp. 142–153). Retrieved June 12, 2022, from <http://ceur-ws.org/Vol-3081/>
- SHKLAR, L., & ROSEN, R. (2003). *Web application architecture: Principles, protocols and practices* (1st ed.). John Wiley & Sons Ltd.
- STACK OVERFLOW. (2022). *2022 developer survey*. Retrieved July 1, 2022, from <https://survey.stackoverflow.co/2022/>
- TALARICO, G. (2022). *Revit API 2022*. Online Documentation for the Revit API. Retrieved September 11, 2022, from <https://www.revitapidocs.com/2022/>
- UNITED NATIONS ENVIRONMENT PROGRAMME. (2021). *2021 global status report for buildings and construction: Towards a zero-emission, efficient and resilient buildings and construction sector* (tech. rep.). Nairobi. www.globalabc.org
- van BERLO, L. (2022, September 19). *IFC specifications database*. buildingSMART. <https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/>
- van BERLO, L., & KRIJNEN, T. (2014). Using the BIM collaboration format in a server based workflow. *Procedia Environmental Sciences*, 22, 325–332. <https://doi.org/10.1016/j.proenv.2014.11.031>
- W3SCHOOLS. (n.d.). *JavaScript HTML DOM*. Retrieved July 1, 2022, from https://www.w3schools.com/js/js_htmlDOM.asp
- WASTIELS, L., & DECUYPERE, R. (2019). Identification and comparison of LCA-BIM integration strategies. *IOP Conference Series: Earth and Environmental Science*, 323(1), 012101. <https://doi.org/10.1088/1755-1315/323/1/012101>
- WERNER, U., & PASTOR, W. (2019). *Vergabe- und Vertragsordnung für Bauleistungen Teil A und B / Bauvertrag §§ 650 a-v / Verordnung über Honorare für Leistungen der Architekten und der Ingenieure* (35th ed.). dtv Verlagsgesellschaft mbH und Co. KG.
- WIRFS-BROCK, A., & EICH, B. (2020). JavaScript: The first 20 years. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1–189. <https://doi.org/10.1145/3386327>
- WORLD GREEN BUILDING COUNCIL. (2022). *EU policy whole life carbon roadmap* (tech. rep.). <https://www.worldgbc.org/EURoadmap>

ZAHEDI, A., & PETZOLD, F. (2019). Interaction with analysis and simulation methods via minimized computer-readable BIM-based communication protocol. In J. P. SOUSA, G. C. HENRIQUES, & J. P. XAVIER (Eds.), *Architecture in the age of the 4th industrial revolution - proceedings of the of the 37th ecaade and the 23rd sigradi conferences* (pp. 241–250). eCAADe, SIGraDi, FAUP. https://doi.org/DOI:10.5151/proceedings-ecaadesigradi2019_140

Declaration

I hereby affirm that I have independently written the thesis submitted by me and have not used any sources or aids other than those indicated.

Location, Date, Signature