

DIPLOMARBEIT

Eine Methodik zum verteilten Informationsmanagement für Energiesimulation in Gebäuden unter Nutzung von openBIM

(A Methodology for information sharing in building energy simulations using openBIM)

eingereicht von:	Yifan Lu
Studiengang:	Bauingenieurwesen
Matrikel-Nr.:	4561244
Abgabe:	08.10.2018
Wiss. Betreuer:	Prof. Dr.-Ing. habil. karsten Menzel M.Sc. Fangzheng Lin

Inhaltsverzeichnis

1. Einführung	1
1.1 Hintergrund	1
1.1.1 BIM und OpenBIM	2
1.1.2 Energiesimulation mit OpenBIM	6
1.2 Aufbau der Arbeit	8
1.2.1 Motivation	8
1.2.2 Kapitelanordnung	9
2. Energiesimulation in Gebäuden	11
2.1 Hintergrund der Gebäudesimulation	11
2.1.1 Berechnung des Energieverbrauches	11
2.1.2 Thermische Behaglichkeit	13
2.1.3 Anforderungen der Energiesimulation	15
2.2 Simulationssoftware	15
3. IFC Anwendung im Bereich Energiesimulation	18
3.1 Die Geschichte des IFC Standards	18
3.2 Hierarchie des IFC Standards	18
3.3 Formaten für IFC Datei	20
3.4 Beschreibungssprache für objektorientierte Modelle	21
3.4.1 EXPRESS und EXPRESS-G	21
3.4.2 Datentypen	22
3.4.3 Datenstruktur	24
3.5 Energiesimulation mit IFC-SPF Datei	25
3.6 Analyse der IFC-SPF Datei	26
3.7 Wände	27
3.7.1 Definition des Standortes einer Wand	29
3.7.2 Die Form einer Wand	30
3.7.3 Materialschichten der Wand	31
3.7.4 Eingestellte Attribute der Wand	34
3.8 Fenster	35
3.9 Türen	38
3.10 Thermische Eigenschaften von Materialien	39
3.11 Räume	41
3.12 Zone und Technische Gebäude Ausrüstung	45
4. Methodik zur Überprüfung des IFC Models	47
4.1 Daten für Bauelementen	47
4.2 Räumliche Daten	49
4.3 Datei für energetische Analyse	50
4.4 Analyse der Fallstudie	51
5. Visualisierung und Kollisionsanalyse	63
5.1 Visualisierung	63
5.2 Kollisionsanalyse	65
5.3 Grafische Benutzeroberfläche (GUI)	68
6. Fehleranalyse für das vorhandene BIM Modell	70
6.1 Grunddaten des Modells	70
6.2 Fehlerprotokoll	71
6.2.1 Kollision zwischen Fenster 2.30 und Fenster 2.22	73

6.2.2 Definition der Öffnungsrichtung	75
6.2.3 Vererbungsfehler bei Fenster 2.53	77
6.2.4 Einschränkungsfelder bei Raum G20.d	78
6.2.5 Beziehungsfehler	79
6.2.6 Redundanz	80
7. Fazit und Ausblick.....	81
7.1 Fazit	81
7.2 Ausblick	81
Literaturverzeichnis	84
Technischer Anhang	86

Abbildungsverzeichnis

Abbildung 1 : Energiekosten nach Anwendungsbereich 2015 [1]	1
Abbildung 2 : BIM-Level Diagram [3]	2
Abbildung 3 : Datenaustausch mit und ohne IFC (Prof. Raimar Scherer)	4
Abbildung 4 : Unterschied zwischen Closed BIM und OpenBIM	5
Abbildung 5 : Arbeitsweise jeder Organisation.....	6
Abbildung 6 : Schlüsselwörter Frequenz	7
Abbildung 7 : Workflow des Berichtes	9
Abbildung 8 : Simulationsmodell [14].....	12
Abbildung 9 : Einfaches Modellbeispiel für die Heizkostensimulation[14].....	13
Abbildung 10 : PMV und PPD nach DIN EN ISO 7730 [16].....	14
Abbildung 11 : Module in IES <VE> (IES <VE> Handbook)	17
Abbildung 12 : IFC Standard Timeline	18
Abbildung 13 : IFC4_layered_architecture [18].....	19
Abbildung 14 : Beziehungen zwischen IFC Ebenen [19]	20
Abbildung 15 : Datentypen in EXPRESS-G [20].....	22
Abbildung 16 : Datentypen in EXPRESS-G [20].....	23
Abbildung 17 : Enumerated Data Type in EXPRESS-G[20]	23
Abbildung 18 : Defined Data Type in EXPRESS-G [20].....	23
Abbildung 19 : Select Data Type in EXPRESS-G [20]	24
Abbildung 20 : Unterklasse von IfcRoot	24
Abbildung 21 : Anforderung des Datenaustausches im Projektablauf	25
Abbildung 22 : Eingabedaten für Energiesimulation	25
Abbildung 23 : Korrespondenz von Entität Definition und praktischen Beispiel.....	26
Abbildung 24 : Hierarchische Struktur von IfcWall [7]	28
Abbildung 25 : Allgemeine Analyseverfahren für Bauelemente	29
Abbildung 26 : Standortinformationen über eine Wand	29
Abbildung 27 : Analyse der Platzierungsstruktur der Wand	30
Abbildung 28 : Entitäten über Formdarstellung der Wand	31
Abbildung 29 : IfcMaterialSelect	32
Abbildung 30 : Materialien Anordnung der Wand (BuildingSMART).....	33
Abbildung 31 : Definition der Eigenschaften der Wand.....	34
Abbildung 32 : Räumliche Struktur von IfcWindow	36
Abbildung 33 : Formparameter für IfcWindow.....	37
Abbildung 34 : Eigenschaften von IfcDoor	39
Abbildung 35 : IfcMaterial und IfcMaterialProperties.....	40
Abbildung 36 : Vererbungsbaum von IfcSpace	42
Abbildung 37 : Nutzungsverfahren der IfcRelReferencedInSpatialStructure (BuildingSMART)	42
Abbildung 38 : Verbindung der IfcSpace mit IfcRelSpaceBoundary	43
Abbildung 39 : Zwei räumliche Grenzen [7]	44
Abbildung 40 : Beziehung zwischen IfcSpace und IfcZone	45
Abbildung 41 : Zusammenbindung von IfcElement und IfcSystem.....	46
Abbildung 42 : Ablauf der Überprüfung	47
Abbildung 43 : Bestandteile für die Überprüfung	47
Abbildung 44 : Arbeitsablauf der Überprüfung für die Wände	48
Abbildung 45 : Ablauf der Überprüfung für räumliche Daten.....	49

Abbildung 46 : Ablauf der Überprüfung für energetische Daten eines Raum	50
Abbildung 47 : Strategie der Validierung	51
Abbildung 48 : Geometrische Daten des Gebäudes	52
Abbildung 49 : Visualisierung in verschiedenen Softwaresystem	55
Abbildung 50 : Darstellung für Materialdaten der Außenwand	56
Abbildung 51 : Darstellung für Materialdaten der Innenwand	56
Abbildung 52 : Darstellung für Materialdaten der Boden	57
Abbildung 53 : Darstellung für Materialdaten der Decke.....	58
Abbildung 54 : Thermische Parameter der Außenwand	59
Abbildung 55 : Eigenschaften der Bauelementen in der IFC Datei.....	60
Abbildung 56 : Geometrische Daten in der IFC Datei	60
Abbildung 57 : Geometrische Daten in IES <VE>	61
Abbildung 58 : Energiekosten eines Raums in der IFC Datei	62
Abbildung 59 : Arbeitsablauf der Visualisierung.....	63
Abbildung 60 : 3D Modell der Fallstudie	64
Abbildung 61 : Der Prozess der Modellierung.....	65
Abbildung 62 : Venn-Diagramme [24]	66
Abbildung 63 : Ablauf der Überprüfung der Kollision	67
Abbildung 64 : Ergebnisse der Überprüfung für Kollision	68
Abbildung 65 : GUI und Quelltext [24].....	69
Abbildung 66 : 3D-Perspektive des Modells	70
Abbildung 67 : Ansichten von Westen und Osten	70
Abbildung 68 : Ansichten von Nordseite und Südseite	71
Abbildung 69 : Fehlerklassifizierung	71
Abbildung 70 : Beispiel für die Kollision	73
Abbildung 71 : Standort der Fenster	73
Abbildung 72 : Ergebnis der Modifizierung	74
Abbildung 73 : Beispiel für Attribut Fehler	75
Abbildung 74 : Quelltext zur Definition der Öffnungsrichtung der Türen.....	75
Abbildung 75 : Ergebnis nach der Modifizierung.....	76
Abbildung 76 : Beispiel für Vererbungsfehler.....	77
Abbildung 77 : Beispiel für Einschränkungsfehler	78
Abbildung 78 : Neues Modell nach der Korrektur	78
Abbildung 79 : Beispiel für Beziehungsfehler.....	79
Abbildung 80 : Fehler bei Redundanz	80
Abbildung 81 : gbxml Datei als Eingabe für IES <VE>	82
Abbildung 82 : Neue Funktionen des Programms	83

Tabellenverzeichnis

Tabelle 1 : Sieben Punkte der Klimabeurteilungsskala	14
Tabelle 2 : Drei Kategorien des Umgebungsklimas	15
Tabelle 3 : Simulationssoftware.....	16
Tabelle 4 : Eigenschaften in Pset_WallCommon.....	34
Tabelle 5 : Attributwerte für IfcSchapeRepresentation	36
Tabelle 6 : IfcMaterialConstituentSet.MaterialConstituents.....	37
Tabelle 7 : Attributsätze für IfcWindow	38
Tabelle 8 : Thermischen Parameter für Materialien	41
Tabelle 9 : Geometrische Daten des Bauelementes	52
Tabelle 10 : Materialiensichten von Außenwand	53
Tabelle 11 : Materialiensichten von Innenwand	53
Tabelle 12 : Materialiensichten von Boden.....	54
Tabelle 13 : Materialiensichten von Decke	54
Tabelle 14 : Thermische Faktoren der Bauelementen	54
Tabelle 15 : Geometrische Daten in Fallstudie 1.....	66
Tabelle 16 : Geometrische Daten in Fallstudie 2.....	66

Abkürzungsverzeichnis

AEC	Architecture Engineering Construction
GHD	Gewerbe, Handel, Dienstleistungen
CDE	Common Data Environment
BIM	Building Information Modeling
IFC	Industry Foundation Classes
bSDD	BuildingSMART Data Dictionary
MVD	Model View Definition
STEP	Standard for The Exchange of Product model data
SPF	STEP Physical File
GUI	Graphic User Interface
GUID	Globally Unique Identifier
HDD	Heating Degree Days
PMV	Predicted Mean Vote
PPD	Predicted Percentage Dissatisfied
2D	Zweidimensional
3D	Dreidimensional



Aufgabenstellung für die Diplomarbeit

Name: Herr Yifan Lu (Matrikel Nr.: 456 1 244)

Vertiefung: Gebäude und Energiemanagement

Thema: Eine Methodik zum verteilten Informationsmanagement fuer Energiesimulation in Gebäuden unter Nutzung von openBIM.

A Methodology for information sharing in building energy simulations using openBIM.

Zielsetzung:

Building Information Modelling (BIM) erlangt in allen Bereichen des Bauwesens eine allgemeine Verbreitung. Trotz der zunehmenden Verbreitung ist die Qualität der Anwendung von BIM noch sehr unetrschiedlich. Ein wesentlicher Grund für die ungenügende Anwendungsqualität ist das Fehlen einer methodischen Vorgehensweise bei der Implementierung von openBIM. Dies gilt insbesondere für interdisziplinäre BIM-Anwendungen.

Im Rahmen der Diplomarbeit soll der Informationsaustausch zwischen Anwendungen des Computer Aided Design (z.B. Autodesk Revit) und Energiesimulationsanwendungen (z.B. IES VE) untersucht werden. Ein Schwerpunkt der Untersuchungen soll die Analyse der Qualität der Modellierung im CAD sein. Die Qualitätsbewertung soll die Wiederverwendbarkeit von BIM-Modellen als Eingabewerte in Energiesimulationswerkzeugen beurteilen.

Fehlerquellen, die zu einer verminderten Qualität des Informationsaustausches führen sollen analysiert und kategorisiert werden.

Auf der Basis der Fehleranalyse soll eine Methodik zur verbesserten Erstellung von openBIM-Modellen in CAD erarbeitet werden.

Arbeitsumfang:

Während der Ausarbeitung sollen folgende Punkte bearbeitet werden.

1. Überblick und Bewertung des Standes der Forschung und Technik im Bereich openBIM-Datenaustauschformate fuer Energiesimulationen.
2. Analyse der Industry Foundation Classes (IFC) im Hinblick auf deren Eignung zur Modellierung von Eingabedaten zur Energiesimulation.
3. Analyse der Qualität von Modellprojekten.
4. Zusammenstellung und Klassifizierung der identifizierten Fehlerquellen.
5. Erarbeitung einer Methodik zur Erstellung qualitativ hochwertiger Entwurfsmodelle in BIM, die sich ohne Nachbearbeitung als Eingabewerte für Energiesimulationen eignen.
6. Verifikation der Methodik durch die Verbesserung eines ausgewählten Modellprojektes (siehe Pkt. 3).

Wiss. Betreuer:

Professor Karsten Menzel

ausgehändigt am:

einzureichen am:



Prof. Dr.-Ing. habil. Karsten Menzel
Verantwortlicher Hochschullehrer

1. Einführung

1.1 Hintergrund

Energie ist die Stütze und treibende Kraft für die wirtschaftliche Entwicklung eines Landes und steht im Zusammenhang mit der nationalen Stabilität und der nachhaltige Entwicklung der ökologischen Umwelt. Mit dem Fortschritt von Wissenschaft und Technologie entwickeln sich sehr schnell die Gesellschaften und die Wirtschaft aller Länder. Der Energiebedarf wächst ebenfalls. Der Energieverbrauch der Bauindustrie spielt eine große Rolle. In der vorliegenden Abbildung 1 wird deutlich angezeigt, dass die Energiekosten für Haushalte größer sind als die anderen Anteile.

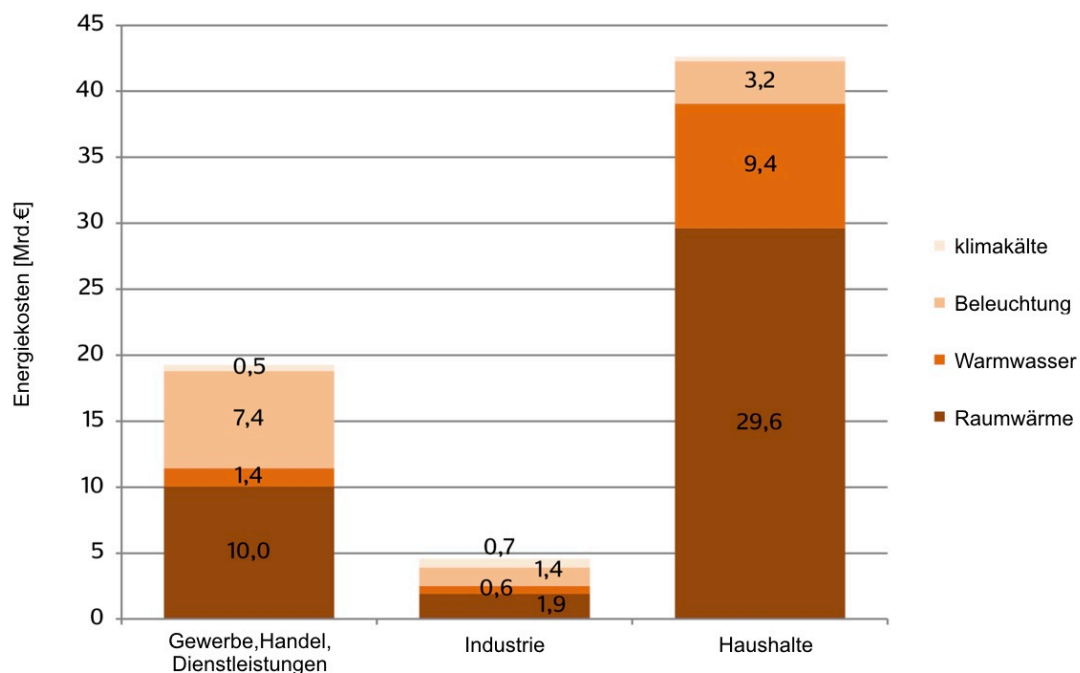


Abbildung 1 : Energiekosten nach Anwendungsbereich 2015 [1]

Aus dem Schaubild geht hervor, dass im Jahr 2015 die Nutzer etwa 66 Mrd. Euro für Gebäudeenergie ausgegeben haben. Davon ca. 42 Mrd. Euro für Raumwärme, ca. 11 Mrd. Euro für Warmwasserbereitung, ca. 12 Mrd. Euro für Beleuchtung und ca. 2 Mrd. Euro für Klimakälte. Im Bereich Gewerbe, Handel, Dienstleistungen (GHD) wird der größte Anteil der Energiekosten (gesamt ca. 19 Mrd. Euro) für Raumwärme aufgewendet (52 %), gefolgt von der Beleuchtung (38 %). Auch in der Industrie stellen die Anteile für Raumwärme (ca. 41 %) und Beleuchtung (31 %) die größten Anteile an den Energiekosten dar (gesamt ca. 5 Mrd. Euro). In Privathaushalten zeigt neben der Raumwärme (70 %) die Warmwasserbereitung (22 %) den wichtigsten Anteil an den Energiekosten (gesamt ca. 42 Mrd. Euro). [1]

Deswegen erhält die Energieeffizienz von Gebäuden mehr und mehr Aufmerksamkeit. Um den Zustand zu verbessern wird die Idee des „Green Building“ aufgestellt. Als „Green Building“ wird ein Gebäude bezeichnet, dass unter dem Leitgedanken der Nachhaltigkeit entwickelt wurde [2]. Die Idee betont die Bewertung des Energieverbrauchs und Einflusses von Gebäuden aus der Perspektive des gesamten Lebenszy-

klus. Für die Untersuchung des Energieverbrauches eines Gebäudes ist die Energiesimulation sehr hilfreich.

Darüberhinaus hat sich die Digitalisierung in den letzten zehn Jahre progressive entwickelt. Die Geschwindigkeit und Genauigkeit einer energetischen Simulation erhöht sich.

1.1.1 BIM und OpenBIM

Building Information Modeling (BIM) bedeutet die Integration und Vernetzung aller relevanten Daten eines Bauwerks in einem virtuellen Datenmodell während des gesamten Lebenszyklus, also von der Konzeption, Planung und Realisierung bis zur Nutzung und zum Rückbau. Die Definition von BIM ist mittlerweile weitgehend in der Fachwelt bekannt. Unklar ist jedoch, wie genau man sich die Form der Datengewinnung und -verarbeitung bei der Bauwerksplanung, -erstellung und -nutzung konkret vorstellen kann. Zum erfolgreichen Einsatz der Methode BIM über den gesamten Lebenszyklus und damit der Informationsaustausch möglichst verlustfrei erfolgen kann, sind Standardisierungen nötig – auf nationaler und auf internationaler Ebene [3].

Gemäß dem „NBS National BIM Report 2017“ wird der Prozess der BIM-Entwicklung in 4 Ebenen (0-3) definiert [4], wie in der .Abbildung 2 angezeigt.

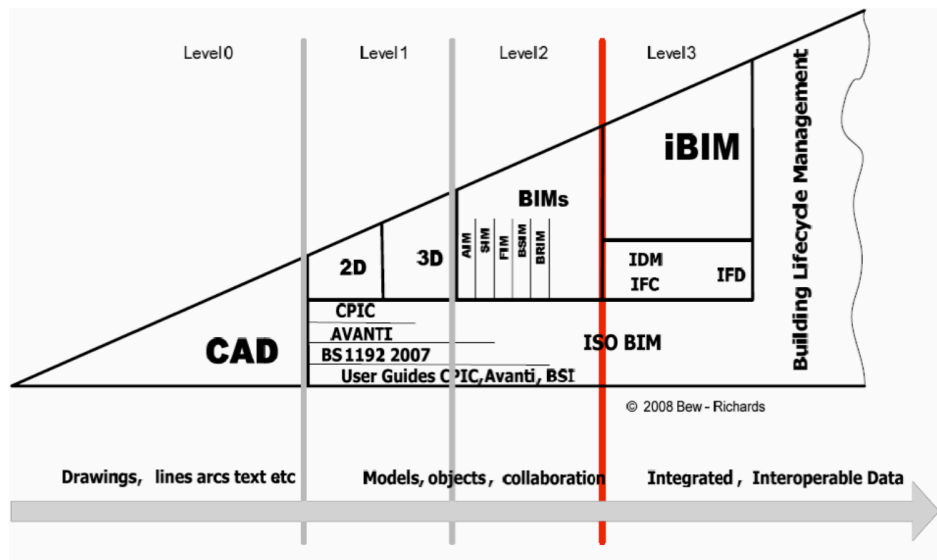


Abbildung 2 : BIM-Level Diagram [3]

• Ebene 0

Ebene 0 ist die einfachsten Form. Diese Ebene bedeutet, es findet keine Zusammenarbeit zwischen Disziplinen statt. 2D CAD Planung wird hauptsächlich als eine Methode genutzt, um Informationen zu produzieren. Papier- und elektronische Zeichnung sind das überwiegend zur Kommunikation genutzte Medium. Die meisten Unternehmen in der Bauindustrie sind bei der Gestaltung der Zusammenarbeit über diese Ebene hinausgegangen.

- Ebene 1

Sehr viele Organisationen in der Bauindustrie befinden sich in dieser Ebenen. Die Besonderheit ist eine Kombination von 3D-Planung und 2D Datenaustausch. Ein Gebäude wird von Architekten in 3D entworfen. Für die Erstellung der gesetzlichen Genehmigungsdokumentation und Gebäudeinformationen werden 2D Dateien akzeptiert. Der Datenaustausch erfolgt aufgrund einer „*Common Data Environment (CDE)*“. Modelle werden nicht zwischen Teammitgliedern verbreitet.

- Ebene 2

In dieser Ebenen verwenden alle Disziplinen ihre eigenen 3D-CAD-Modelle. Es ist nicht notwendig, mit einem einzigen, gemeinsamen Modell zu arbeiten. Das wichtigste Problem ist der Datenaustausch zwischen verschiedenen Softwaretools. Bauinformationen werden über ein gemeinsames Dateiformat ausgetauscht. Damit kann jede Organisation diese Daten mit ihren eigenen Softwaretools überprüfen und weiter bearbeiten.

- Ebene 3

In dieser Ebene wird ein einziges gemeinsames Projektmodell von allen Disziplinen verwendet, dass in einem zentralen Repository gehalten wird. Das Wichtige ist nicht der Datenaustausch sondern das „*Daten Sharing*“. Ein BIM-Server wird als Datenträger aufgebaut. Ein großer Vorteil ist die Informationsintegration. Die Anzahl der Aktivitäten des Datenaustausches werden reduziert [5]. Deswegen sind die Kosten und Fehlerraten gesunken. Diese Vorgehensweise ist auch als „OpenBIM“ bekannt.

Aber es gibt auch Nachteile. Urheberrechtsproblem und Haftungsprobleme müssen berücksichtigt werden.

Die Schwierigkeiten der Anwendung von BIM werden in drei Punkten zusammengefasst:

- Eine große Anzahl von Beteiligten. Ein Projekt ist keine einfache Angelegenheit. Es kann nur durch eine solide Verbindung von mehreren Organisationen durchgeführt werden.
- Lange Periode. Der Lebenszyklus eines Gebäudes kann in folgende einzelne Lebenszyklusphasen eingeteilt werden [6]:
 - Planungsphase
 - Errichtungsphase
 - Nutzungs- und Betriebsphase
 - Instandhaltung- und Modernisierungsphase
 - Umnutzungs-/Weiternutzungsphase
 - Rückbau und Wiederverwendung
 - Recycling
- Vielfältige Softwaretools. Offensichtlich werden verschiedene Softwaretools in verschiedenen Phase eines Projektes genutzt.

OpenBIM ist eine Lösung für die Probleme in der Bauindustrie. Der Begriff ist von „BuildingSMART“ aufgestellt. Er ist eine neue Arbeitsweise, mit dem die Kosten reduziert und die Effizienz gesteigert werden können. wir brauchen standardisierte und

sinnvolle Symbole und eine Grammatik zum „*Data Sharing*“. Bei einheitlichen Open Source Formaten werden die Daten genutzt, gespeichert und weitergegeben. Wie in der unteren Gleichung angezeigt, hat dieser Begriff drei wichtige Bestandteile.

$$\text{OpenBIM} = \text{IFC} + \text{bSDD} + \text{Prozess} [7]$$

Um ein Gebäude entwerfen, bauen und verwalten zu können, wird die Bauindustrie in viele Fachdisziplinen aufgeteilt. Die Fachleute in verschiedenen Bereichen müssen effizient mit denselben Daten des Gebäudes zusammenarbeiten. Im traditionellen Bauprozess werden die Daten über 2D-Bauzeichnungen ausgetauscht. Aber diese Datenübertragung kann viele Missverständnisse und Informationsverlust verursachen. Wegen dieses Koordinierungsproblems gibt es in fast allen traditionellen Bauprojekten kostspielige Fehler, Z.B. Verzögerung und Finanzprobleme. Die Grundursache dafür ist die mangelnde Modernisierung und Informatisierung der Bauindustrie. Wenn man alle Daten eines Gebäudes in einem offenen Format austauschen kann, würden die Arbeiten im einen kompletten Lebenszyklus eines Gebäudes effizient und problemlos erledigt. Ein mögliches Format ist *Industry Foundation Classes (IFC)*. Mit IFC kann jeder Mitarbeiter in einem Projekt Daten austauschen und die Kooperation wird effizienter. Die Abbildung 3 zeigt die Arbeitsweise mit und ohne IFC.

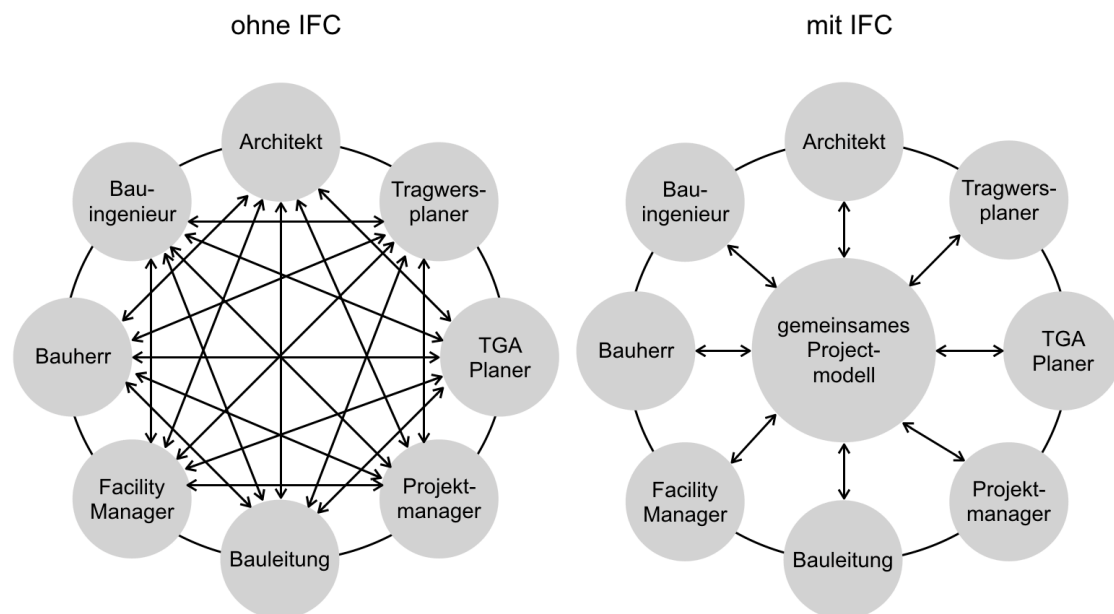


Abbildung 3 : Datenaustausch mit und ohne IFC (Prof. Raimar Scherer)

Es ist einfach zu sehen, dass wenn man mit IFC arbeitet, sich die Anzahl des Datenaustausches reduziert. Der weniger Datenaustausch bedeutet weniger Fehler und bessere Ergebnisse.

Die Kommunikation zwischen unterschiedlichen Bereichen und Software bedarf bestimmter Mittel der Übertragung. IFC ist die beste Variante bis jetzt. Das *BuildingSMART Data Dictionary (bSDD)* wird für die Normalisierung aller Entitäten und Attribute eines Datenmodells genutzt [8]. Damit können wir sicherstellen, dass alle BIM-Softwaretools die Modelldatei richtig verstehen und analysieren können.

OpenBIM unterscheidet sich von der Arbeitsweise des closed BIM. In der vorliegenden Abbildung 4 wird der Unterschied dargestellt.

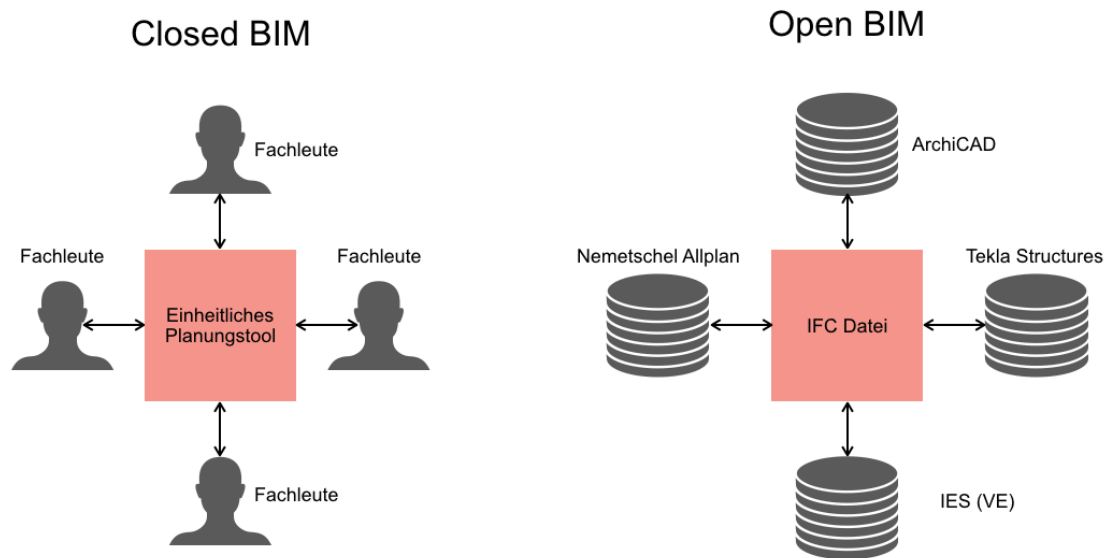


Abbildung 4 : Unterschied zwischen Closed BIM und OpenBIM

Der Prozess von Closed BIM ist auf eine einheitliche Software abgestimmt und setzt voraus, dass alle Projektbeteiligten, wie Architekten und Tragwerksplaner, mit einer bestimmten für das Unternehmen vorteilhaften Planungssoftware arbeitet. Der Vorteil von closed BIM ist die unkomplizierte Koordination von Fachmodellen sowie die Verwendung eines identischen Dateiformats zur Planung. Informationsverluste durch fehlinterpretierte Daten können weitestgehend ausgeschlossen werden. Der Nachteil des geschlossenen BIM-Prozesses ist die eingeschränkte Flexibilität. Closed BIM ist weniger flexibel bei der Übergabe von Modellen und Informationen an externen Fachplaner. Es liegen zwar Möglichkeiten zum Import und Export in Drittherstellerformate vor, nur werden diese mangels Richtlinien und abgestimmten Übergaberoutinen in einem geschlossenen BIM-Prozess nicht weiter verfolgt.

Im Gegensatz zu closed BIM ist open BIM auf eine softwareübergreifende Arbeitsweise samt Einbeziehung verschiedenster Planungstools ausgerichtet [9]. Alle beteiligten Personen können mit ihrer gewohnten Software weiter arbeiten und müssen sich nicht für den BIM-Prozess umstellen. Bspw. kann der Architekt weiter mit ArchiCAD und der Tragwerksplaner (TWPL) mit Tekla Structures arbeiten.

Darüberhinaus wird die traditionelle Arbeitsweise vereinfacht. Der Ablauf eines Projektes kann in viele kleine Bestandteile zerlegt werden. Jeder Bestandteil ist eigentlich eine Grundeinheit. Daraus kann der Ablauf vollständig beschrieben werden. Diese Prozess-Schritte können „Aufgabe“ genannt werden. Der Arbeitsablauf für alle Aufgabe ist fast gleich. Wie in Abbildung 5 dargestellt, wird der Prozess für jede Abgabe in drei Phasen unterteilt. Der erste Fall ist Importieren von Informationen. Dann werden die Informationen bearbeitet. Zum Schluss werden Ergebnisse exportiert. Das Ergebnis kann als Eingabedaten für folgende Schritte benutzt werden. Der Arbeitsablauf funktioniert Schritt für Schritt bis zum Ende.

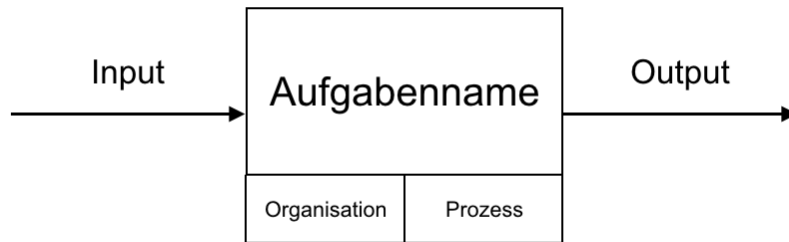


Abbildung 5 : Arbeitsweise jeder Organisation

Mit der Vereinfachung ist es deutlich, dass die Input und Output Dateien den Schwerpunkt für einen Ablauf darstellen. Wenn man den ganzen Ablauf mit einem Meer vergleicht, sind die einzelnen Aufgabe „einsame Insel“ und die Input und Output Dateien sind „Brücken“ zwischen den Inseln. Je besser die Brücken funktionieren, umso effizienter funktioniert der Ablauf. Als Medium für Datenaustausch, muss es zwei Bedingungen erfüllen. Die erste ist die Effizienz, und die zweite ist, dass das Medium von allen Softwaretools erkannt und unterstützt werden muss. In Hinsicht auf die Vorteile von IFC, wird das Format als die beste Wahl betrachtet. In Kapitel 3 wird dies deutlich dargestellt.

1.1.2 Energiesimulation mit OpenBIM

Aufgrund der Vorteile von *OpenBIM*, wird es im Laufe des Projektablaufs verwendet. Heutzutage ist Grünes Gebäude auch ein Schwerpunkt in der Bauindustrie. Die Gebäude zeichnen sich unter anderem durch eine hohe Ressourceneffizienz in den Bereichen Energie, Wasser und Material aus, während gleichzeitig schädliche Auswirkungen auf die Gesundheit und die Umwelt reduziert werden [2]. Offenbar hat die Berechnung des Energieverlustes eines Gebäudes große Bedeutung dafür. Darum muss die Anwendung von OpenBIM in der Fachrichtung Gebäude und Energiesimulation mehr beachtet werden. Um den Zustand der Forschung festzustellen, sind kürzlich veröffentlichte Abhandlungen sehr hilfreich. Mit der Entwicklung von Web-Technik und „*Big Data*“ gibt es heutzutage mehr Möglichkeiten, um eine umfassende Menge Abhandlungen zu analysieren. Auf der Website „ScienceDirect (www.sciencedirect.com)“ kann man Abhandlungen aller Fachdisziplinen suchen und kostenlos herunterladen. Um die Situation und Tendenz der Forschung für BIM zu untersuchen, werden die Abhandlungen in dieser Webseite unter dem Suchbegriff „Building Information Modeling“ recherchiert. Im Zeitraum von 2015 bis 2018 werden mehr als 6000 Artikel über BIM veröffentlicht. Die Schlüsselwörter davon sind mehr als 40000. Abbildung 6 zeigt die 10 Schlüsselwörter mit der höchsten Häufigkeit in allen Abhandlungen.

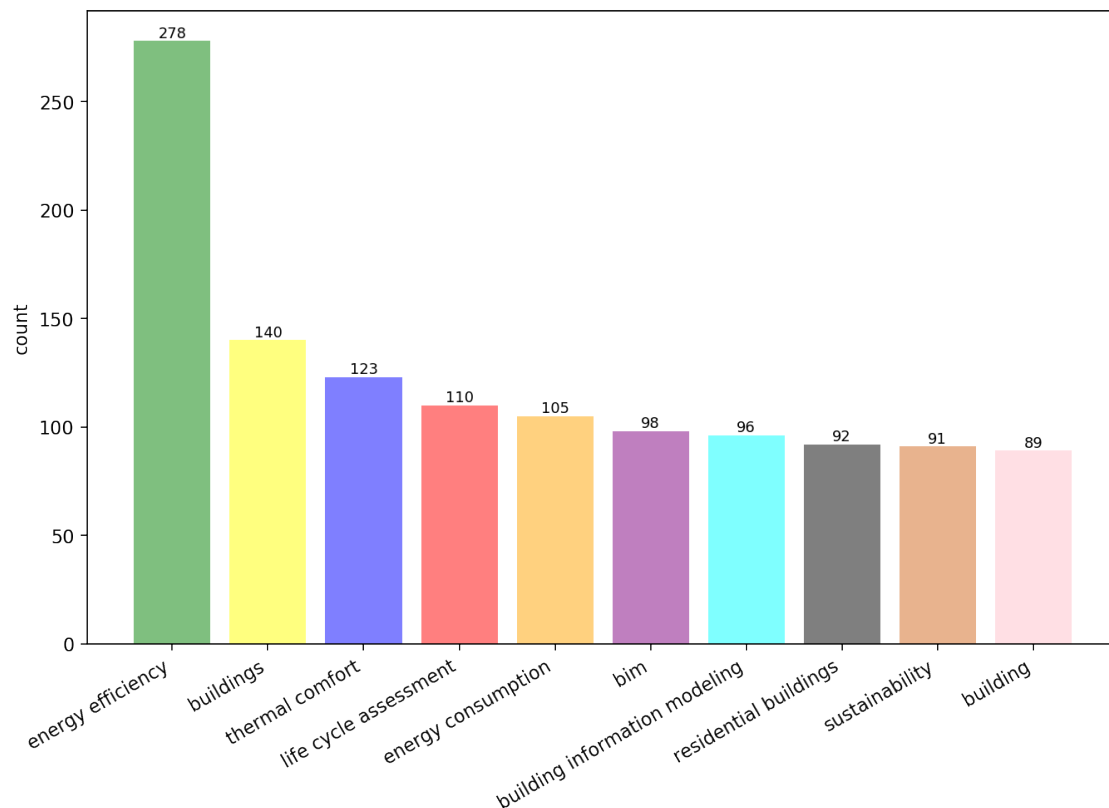


Abbildung 6 : Schlüsselwörter Frequenz

Wie das Diagramm darstellt, hat „energy efficiency“ den ersten Platz. Die Schlüsselwörter „thermal comfort“ und „energy consumption“ spielen auch eine große Rolle. Daraus können zwei Schlussfolgerungen abgeleitet werden. Der erste Fall ist, dass die BIM-Anwendung im Bereich von Energieeffizienz die Hauptforschungsrichtung ist. Und der zweite ist, dass die Forschungen in zwei Komponenten aufgeteilt werden können. Die erste Vertiefung ist Berechnung der Energiekosten eines Gebäudes und die zweite ist thermische Behaglichkeit. Darüberhinaus sind die zwei Richtungen im Ablauf der Energiesimulation zu berücksichtigen. Und die hauptsächliche Teile im Prozess des Datenaustauschs zwischen BIM Software und Energiesimulation Software.

Ein Problem für OpenBIM-Anwendungen im Bereich Energiesimulation ist die Kompatibilität von IFC [10]. Wegen der verschiedenen Datenbanken von vielfältigen Softwaretools hat jede Software einen eigenen Algorithmus, um ein Gebäudemodell in der IFC Datei umzuwandeln. Den Umwandlungsprozess können die Fachleute nicht kontrollieren. Darum ist die Qualität der exportierten IFC Dateien nicht befriedigend. Um die Qualität der IFC Datei zu verbessern und Fehler im Umwandlungsprozess zu vermeiden, haben viele Forscher an der Verbesserung von IFC gearbeitet. Eine Methodik bei Sergio Pinheiro ist die Anwendung von *Information Delivery Manual (IDM)* und *Model View Definition (MVD)* [11]. In der Methodik wird die Eingabeanforderung der Simulationsprogramme untersucht. Das IDM / MVD-Framework erfasst und übersetzt die Austauschforderungen in ein IFC-Schema. Durch das spezielle MVD können Fachleute den Prozess des Datenaustausches optimieren.

Im Februar dieses Jahres (2018) hat Zhiliang, Ma eine Abhandlung veröffentlicht. Die Idee bei Ihm ist der Aufbau einer Plattform mithilfe von Ontologie und Semantik Web [12]. Damit werden die Gebäudemodelle überprüft und korrigiert. Die Modelle werden

visualisiert. Der Schwerpunkt der Untersuchung ist die Umwandlung von BIM Daten in ontologische Daten.

Der Zweck von IFC ist Datenspeicherung und -austausch. Darüberhinaus kann man hinterfragen, ob die relationalen Datenbanken zu dieser Funktionen passen? Solihin hat im Jahr 2014 eine Abhandlung dafür veröffentlicht. Die Untersuchung basiert auf IFC-Standard und relationalen Datenbanken. Es wird ein Algorithmus untersucht, um die IFC Datei in eine relationale Datenbank umzuwandeln [13]. Die Methodik hat IFC Datei benutzt, um Daten zwischen Softwaresystemen auszutauschen. Die Motivation der Untersuchung ist, dass IFC Standard nicht durch entwickelt wird. Die Beschreibung eines Gebäudes ist unvollkommen. Deswegen werden Fehler verursacht, wenn ein Modell als IFC Datei exportiert wird. Daher ist eine relationale Datenbank bessere Methode, um ein Gebäudemodell zu speichern und seine Attribute zu beschreiben. Aber es gibt viele Nachteile. Zuerst ist gibt es ein Kompatibilitätsproblem. Viele Softwares haben keine spezielle Schnittstelle für diese Datei. Deswegen muss man für jede Software ein Plug-in entwickeln. Der zweite ist die Anzahl der Datenumwandlung. Die Fachleute müssen ein Gebäudemodell in BIM Software zuerst als IFC Datei exportieren und dann in eine relationale Datenbank umwandeln und speichern. Je mehr Daten ausgetauscht werden, desto mehr Fehler werden verursacht. Zusammenfassend lässt sich feststellen, dass BIM-Anwendungen im Bereich Energiesimulation zwei Schwerpunkte haben. Der erste ist die Überprüfung eines Modells und der zweite ist die Vermeidung und das Korrigieren der Fehler eines Modells. Sie sind auch die Schwerpunkte der Arbeit.

1.2 Aufbau der Arbeit

1.2.1 Motivation

BIM erlangt in allen Bereichen des Bauwesens eine allgemeine Verbreitung. Trotz der zunehmenden Verbreitung ist die Qualität der Anwendung von BIM noch sehr unterschiedlich. Bei der Verwendung des BIM-Verfahrens steigt die Genauigkeit und die Wirkkraft einer energetischen Gebäudesimulation. Jedoch sollte ein konsistenter Datenaustausch zwischen architektonischen Planungsprogrammen und Simulationstools erfolgen, damit die Gebäudeanalyse erfolgreich abgeschlossen werden kann. Die erste Zielstellung der Arbeit ist Untersuchung der Anwendbarkeit von BIM-Methode im Rahmen „Energetische Gebäudesimulation“. Die zweite ist Analyse der Interoperabilität zwischen verschiedene Softwaretools und Analyse die Datenaustausch zwischen BIM architektonischer und energetischer Simulationssoftware basiert auf IFC Datei. Es ist wertvoll, dass die Fehler im Verlauf von Informationsverteilung analysiert werden. Durch die Auswertung des Prozesses der Informationsübertragung wird die Methodik zur Erstellung qualitativ hochwertiger Entwurfsmodelle in BIM untersucht.

Diese Forschung hat theoretische und praktische Bedeutung. Einerseits wird die theoretische Einführung von BIM erzählt. BIM ist eine neue Arbeitsweise, damit wird sich die Bauindustrie in Richtung Informatisierung bewegen. In den letzten Jahren hat sich die BIM-Technologie rasant entwickelt. Die Anwendung von BIM wird nicht nur von der Regierung sondern auch Unternehmen gefördert. Aber die aktuelle Beschränkung ist die Arbeitsweise von Datenspeicherung und -austausch. In dieser Arbeit wird auch die Entwicklung und Anwendung von IFC erzählt. Um Möglichkeiten zu erkunden, wie In-

formationen zwischen verschiedenen Software ausgetauscht werden, spielt die Kombination zwischen IFC und energetische Gebäudesimulation eine wichtige Rolle. Gleichzeitig stellt die Forschung auch für die Erweiterung der Anwendung von BIM bereit. Andererseits ist die Forschung sinnvoll für die Bauindustrie. Mit der Entwicklung der Technologie, ist die Anwendung von BIM der wichtigste Trend. Aber derzeit gibt es verschiedene BIM Softwaretools, unterschiedliche Datenbank und Datenstrukturen. Die Interoperabilität ist nicht ausreichend für Kooperation in den verschiedenen Bereichen. Durch Testen wird die Hauptsache für das Problem herausgefunden. Gegen das Problem wird die wirksame Behandlung durchgeführt. Damit wird interdisziplinäre BIM-Anwendungen beschleunigt.

1.2.2 Kapitelanordnung

Das untere Schaubild 7 zeigt die Anordnung des Berichtes. Zuerst stellt sich eine Frage. Dann wird die Frage analysiert und Grundlage vorbereitet. Darüberhinaus muss man das Problem Schritt für Schritt lösen und eine passende Methodik finden. Zum Schluss wird die Methodik mit einem praktischen Beispiel nachgeprüft.

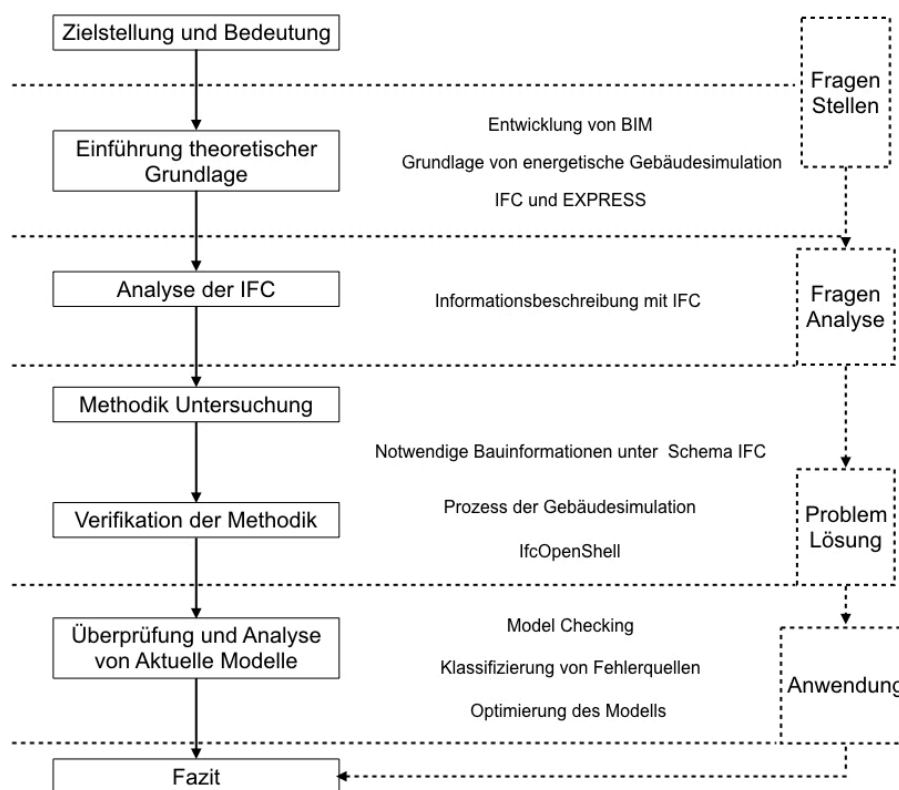


Abbildung 7 : Workflow des Berichtes

Die Arbeit besteht aus 7 Kapiteln. Kapitel 1 ist die Beschreibung des Hintergrund der Arbeit. Es ist sinnvoll, zuerst die relevante Begriffe und Arbeitsweise mit BIM und OpenBIM zu erklären.

Ferner werden in Kapitel 2 die Grundlagen der Energiesimulation dargestellt. Das zweite Kapitel ist hauptsächlich über die wichtigsten Parameter für die Berechnung von

Energiekosten und dann wird das Simulationsprogramm *IES <VE>* gezeigt und der Ablauf des Programms wird auch im zweiten Kapitel beschrieben.

Datenspeicherungsprozesse, Schemata und diverse Dateiformate, die innerhalb des BIM-Verfahrens gebräuchlich sind, werden betrachtet.

Anschließend wird in Kapitel 3 zuerst die Industry Foundation Classes (IFC) ausführlich dargestellt. Dann ist die Beschreibung von vielfältigen Datenformaten wie IFC-SPF, IFCXML und IFC-ZIP. Das IFC-SPF Format ist der Schwerpunkt der Arbeit. Es ist eine Textdatei und ist lesbar in einem Text-Editor. Der Inhalt ist durch eine objektorientierte Sprache *EXPRESS* definiert. Deswegen werden im Kapitel 3 die Datenstruktur und Grammatik von *EXPRESS* erzählt. Aufbauend auf den obengenannten Erkenntnissen, wird auch im Kapitel 3 die Möglichkeit der Kopplung zwischen BIM und der energetischen Gebäudesimulation analysiert.

In Kapitel 4 wird die Methodik zur Überprüfung eines IFC Modells erstellt. Basierend auf der Methodik wird eine Fallstudie untersucht.

In Kapitel 5 ist die Visualisierung mithilfe von IfcOpenShell und PythonOCC erfolgreich. Dann wird eine grafische Benutzeroberfläche aufgebaut. Man kann dadurch eine IFC Datei analysieren und überprüfen.

Auf der Basis des obengenannten Prozesses wird in Kapitel 6 das aktuelle Modellprojekt analysiert. Die identifizierten Fehlerquellen werden klassifiziert und zusammengestellt.

Zum Schluss wird in Kapitel 7 ein Fazit erstellt und werden für die künftige Forschung wünschenswerte Punkte aufgezeigt.

2. Energiesimulation in Gebäuden

Die Gebäude- und Anlagesimulation ist ein wichtiges Werkzeug für die Planung von Gebäuden. Mit der Simulation können Funktionalität und Komfort von Gebäuden erhöht und Energie- und Investitionskosten gesenkt werden. Der Hauptzweck der Simulation des Gebäudeentwurfs ist, die Entscheidungen über die Gestaltungsmöglichkeiten mithilfe klarer Kriterien zu bewerten. Die Gebäudesimulation ist für die Anwendung in frühen und späteren Phasen des Entwurfes sowie für Neubau und den Nachrüstungs- oder Sanierungsentwurf geeignet. Doch aus verschiedenen Gründen wird die Simulation in der Planungspraxis hier zu Lande noch nicht in wünschenswerter Weise genutzt. In diesem Kapitel werden die Grundlagen der energetischen Simulation und verbreitete Simulationssoftware beschrieben. Die Anforderungen an die Eingabedaten für Simulationen werden betrachtet.

2.1 Hintergrund der Gebäudesimulation

Der Energiefluss in einem Gebäude ist ein komplexer dynamischer Prozess mit vielen Faktoren. Im Prinzip können Wärmefluss und Energiebedarf in bestehenden Gebäuden durch Messungen ermittelt werden. Aber diese Untersuchung ist langwierig und kostspielig. Darum wird eine bessere Variante verwendet. Die Entwicklung der Rechnertechnik in den letzten 10-20 Jahren verstärkt die zunehmende Bereitschaft des Ingenieurs, Planers oder Gutachters, das Hilfsmittel Computer in sein tägliches Arbeitsfeld einzubeziehen. Dies betrifft nicht nur die Auslegung von Systemparametern, sondern vor allem die instationäre Simulation auch komplexer physikalischer Zusammenhänge, wie sie bei einer Gebäude-Anlagen-Nutzer-Simulation zu finden sind [14]. Die Computersimulation macht es möglich, ein Gebäude als ganzes zu betrachten und alle Wechselwirkungen formelmässig zu berechnen. Die graphische Darstellung von Temperaturverläufen und Wärmefluss ermöglicht eine einfache Beurteilung des dynamischen Verhaltens und einen raschen Vergleich unterschiedlicher baulicher oder anlagentechnischer Varianten.

Eine Gebäudesimulation hat zwei Richtungen. Zuerst wird der Energieverbrauch berechnet. Das zweite Ziel ist, dass die Fachleute durch die Simulation eines Gebäudes die thermischen Behaglichkeit bestimmen können.

2.1.1 Berechnung des Energieverbrauches

Nach dem Grundprinzip der Wärmeübertragung in Gebäuden können unterschiedliche Berechnungsverfahren des Gebäudeenergieverbrauchs benutzt werden. Der erste Fall ist die Nutzung vereinfachter Gleichungen, die auf der Theorie der stationären Wärmeübertragung basieren. Der zweite Fall basiert auf der Theorie der dynamischen instationären Wärmeübertragung als Simulationsmethode.

Die vereinfachten Berechnungsverfahren wurden vor der Erfindung der Simulationssoftware von allen Fachleuten verwendet. Sie basieren auf Normen. Ein Vorteil davon ist, dass der Prozess der Berechnung vereinfacht werden kann. Aber der Nachteil ist, dass sie eine schlechte Genauigkeit haben. Z. B. Heizgradtage (engl. *HDD: Heating Degree Days*) Methode und BIN Methode.

Darüber hinaus wird eine Simulationsmethode aufgestellt. Sie ist in der Lage, den Energieverbrauch eines Gebäudes präziser zu berechnen. Die Voraussetzung dafür ist die Zusammenarbeit mit Software. Die Fachleute müssen zuerst ein Modell eines Gebäudes in einer Simulationssoftware aufbauen und dann die Randbedingungen hinzufügen. Mit Engine der Simulationssoftware wird das Ergebnis berechnet und grafisch dargestellt.

Wenn das Modell mathematisch formuliert und durch die Ergebnisse des Experiments bestätigt wurde, können wichtige Probleme gelöst werden: Erstens ermöglicht es, einen Teil der Wirklichkeit auf der Grundlage der gegebenen Anfangsbedingungen zu beschreiben und, zweitens, vorherzusagen, wie sich das beschriebene System in verschiedenen Situationen verhalten wird. Besonders nützlich für die Modellierung sind die Erhaltungsgesetze und Bilanzgleichungen. (Zum Beispiel das thermodynamische Gleichgewicht zur Beschreibung der Wärmeleitungsprozesse). Alle Rechenmethoden, auf denen Simulation basiert, lassen sich unterscheiden in statische und dynamische Verfahren.

Hier werden Bilanzgleichungen im stationären Zustand eines Systems überprüft, z. B. Gewinn- und Verlustwärmeströme. Es kommen relativ einfache Berechnungsverfahren zur Anwendung. Nicht berücksichtigt werden hier dynamische Vorgänge im Gebäude oder in der Anlagentechnik (Erwärmung, Auskühlung etc.). Simulationsprogramme nach diesem Verfahren sind meist einfach anzuwenden und insbesondere für weniger komplexe Gebäude geeignet. Planer können mit solchen Programmen schon in frühen Planungsphasen wichtige energetische Gebäudekennwerte abschätzen, nicht jedoch spezielle Untersuchungen vornehmen, wie z. B. sommerliche Überhitzung oder nutzungs- und raumbezogene Betrachtungen.

Die heutige Rechnertechnik erlaubt es, mit numerischen Rechenmethoden das dynamische Verhalten von Gebäuden zu beschreiben. Es können zeitliche Verläufe von Temperaturen, Heiz- oder Kühllasten u. a. berechnet werden. Dabei werden die thermischen Speicherfähigkeit des Gebäudes und dynamische Effekte der Anlagentechnik berücksichtigt. Die vorliegende Abbildung 8 zeigt den Aufbau des Simulationsmodells.

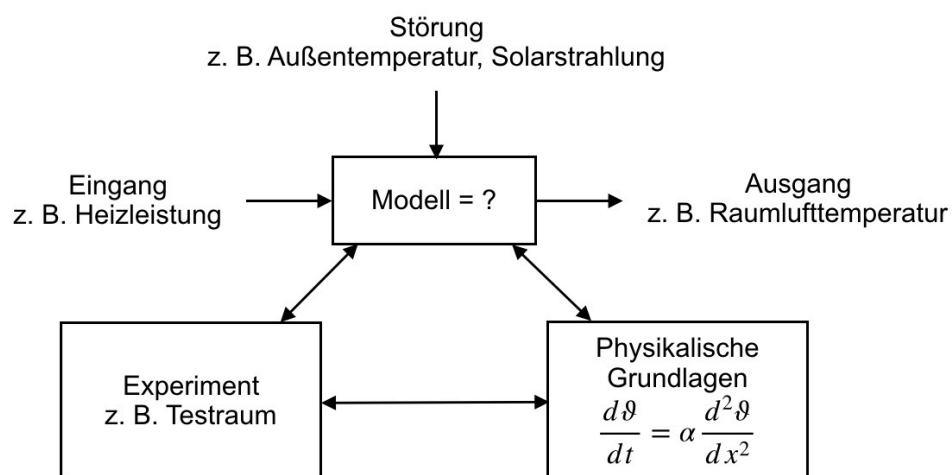


Abbildung 8 : Simulationsmodell [14]

Jede Simulation basiert auf einem Modell als Abbild des realen oder geplanten Gebäudes. Das Simulationsmodell besteht aus Teilmodellen für z. B. das Gebäude, die verschiedenen Gebäudetechniken, die Gebäudedurchströmung etc. Es beschreibt die wesentlichen Eigenschaften der zu untersuchenden Prozesse und deren Wechselwirkungen. Sämtliche Ergebnisse einer Simulation beziehen sich ausschließlich auf dieses Modell. Inwieweit also Simulationsergebnisse auf die originalen Verhältnisse und Prozesse übertragbar sind, hängt entscheidend davon ab, wie genau die Wirklichkeit durch Modellstruktur und Modellparameter nachgebildet wird.

Die Beschreibung des Gesamtmodells der Anlage erfolgt mit Hilfe von Teilmodellen, die miteinander verknüpft werden. Die Abbildung 9 zeigt ein einfaches Modellbeispiel für die Heizkostensimulation.

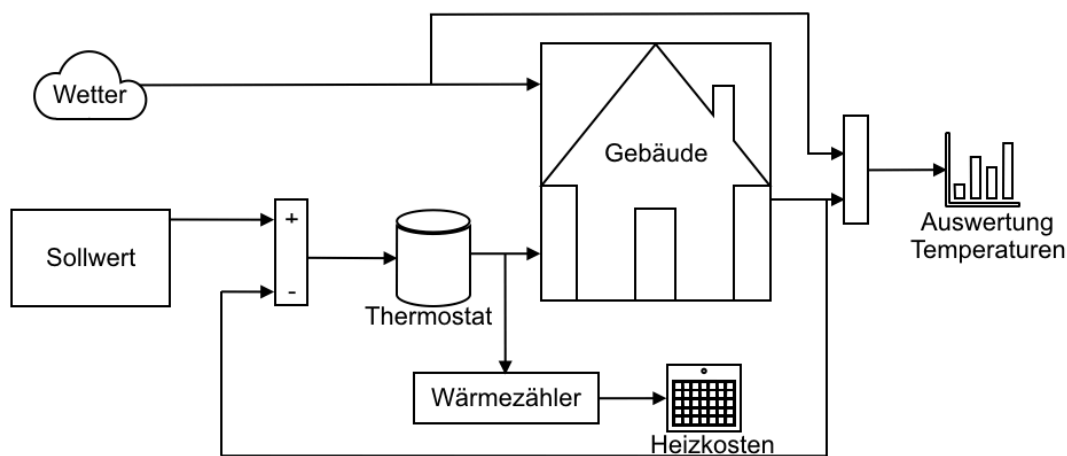


Abbildung 9 : Einfaches Modellbeispiel für die Heizkostensimulation[14]

Aus der Abbildung ist zu ersehen, dass das Ergebnis der Simulation als eine Kombination von Außentemperatur und dem Sollwert der technischen Ausrüstung besteht. Normalerweise wird der Sollwert als „operative Temperatur“ benannt.

2.1.2 Thermische Behaglichkeit

Die thermischen Behaglichkeit wird in DIN EN ISO 7730 als Gefühl, das Zufriedenheit mit dem Umgebungsklima ausdrückt, definiert. Daneben ist die Thermische Behaglichkeit in DIN 1946-2 so definiert: wenn der Mensch Lufttemperatur, Luftfeuchte, Luftbewegung und Wärmestrahlung in seiner Umgebung als optimal empfindet und weder wärmere noch kältere, weder trockenere noch feuchtere Raumluft wünscht [15].

Die Bewertungskriterien und Einflussfaktoren auf die thermischen Behaglichkeit werden durch die Normung verschieden definiert.

Nach DIN EN ISO 7730 hängt das thermische Gleichgewicht von körperlicher Tätigkeit, der Bekleidung sowie der Lufttemperatur, der mittleren Strahlungstemperatur, der Luftfeuchte und Luftgeschwindigkeit ab. Für die Bewertung der thermischen Behaglichkeit werden zwei Indexe eingesetzt:

- PMV: das vorausgesagte mittlere Votum

- PPD: der vorausgesagte Prozentsatz an Unzufriedenen

PMV ist ein Index, der den Durchschnittswert für die Klimabeurteilung durch eine große Personengruppe anhand einer 7-stufigen Klimabeurteilungsskala vorhersagt (Tabelle 1)

+3	heiß
+2	warm
+1	etwas warm
0	neutral
-1	etwas kühl
-2	kühl
-3	kalt

Tabelle 1 : Sieben Punkte der Klimabeurteilungsskala

Der PPD lässt sich aus dem PMV berechnen (Abbildung 10). Der PPD ist eine Voraussetzung darüber, wie hoch der Prozentsatz unzufriedener ist, wenn eine große Personengruppe ein vorliegendes Raumklima bewertet.

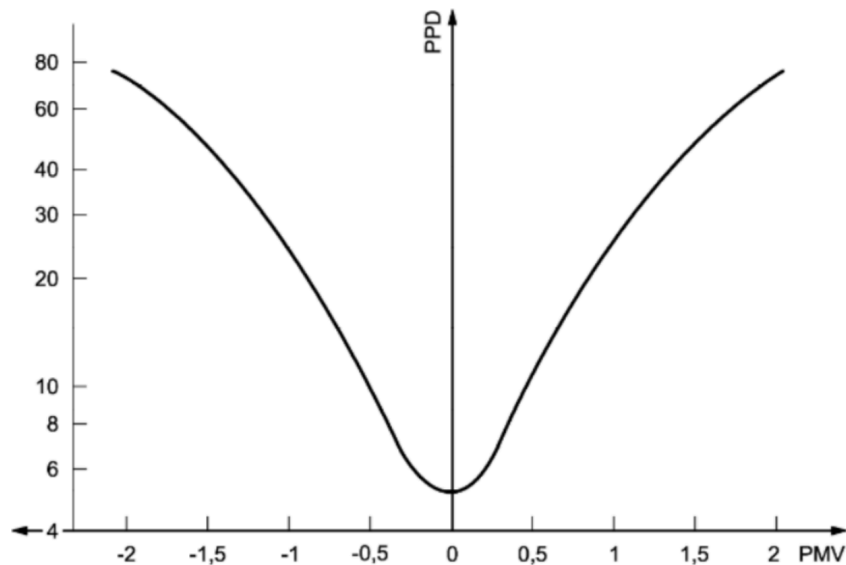


Abbildung 10 : PMV und PPD nach DIN EN ISO 7730 [16]

Die Bewertung der thermischen Behaglichkeit nach PPD und PMV kann drei Kategorien zugeordnet werden, wie in der vorliegenden Tabelle 2 aufgezeigt werden: Kategorien A (94% Zufriedene), B (90% Zufriedene) und C (85% Zufriedene) [16].

Kategorie	PPD in %	PMV
A	< 6	-0.2 <PMV <0.2
B	< 10	-0.5 <PMV <0.5
C	< 15	-0.7 <PMV <0.7

Tabelle 2 : Drei Kategorien des Umgebungsklimas

2.1.3 Anforderungen der Energiesimulation

Für die Energiesimulation sind die Eingabedaten und Randbedingungen notwendig. Die Durchführung der Simulation erfordert die Definition von Eingabedaten. Jede erfolgreiche Simulation erfordert, erstens, gültige Daten und, zweitens, die Fähigkeit der Simulationssoftware, die Daten für einen vorgegebenen Gebäudeentwurf richtig zu bewerten. In der Planungsphase eines Gebäudes wird der architektonische Entwurf zuerst gemacht, und damit wird die technische Gebäudeausrüstung geplant und Energiesimulation mit bestimmten Softwaretools durchgeführt. Deswegen bestehen die Eingabedaten eines Simulationsprogramms aus zwei Teilen. Der erste Teil sind Daten über Bauelemente wie Wände, Türen und Fenster. Sie werden bei der Planung definiert. Bei der Bestimmung der Eigenschaften des Materials wie Wärmekapazität und Wärmedurchlässigkeit muss man beachten. Der zweite Teil sind die Informationen über thermische Zonen und technische Gebäudeausrüstungen. Sie werden vom HVAC Ingenieur aufgrund des Gebäudeentwurfs hinzugefügt. Die Verteilung und Eigenschaften des Anlagesystems und der Zeitplan werden ergänzt.

Zusammenfassend werden die typischen Eingabedaten aufgelistet:

- Geometrie der Gebäude (2D oder 3D),
- Wetterbedingungen,
- Eigenschaften der Stoffe, die für die Bauteile verwendet werden,
- Nutzungsarten,
- Orientierung an den Himmelsrichtungen,
- Informationen über Aufbau und Funktionsprinzip von TGA.

Sie definieren die Schwerpunkte im Prozess der Energiesimulation. Deswegen muss man die Genauigkeit und Konsistenz dieser Daten berücksichtigen, wenn sie zwischen CAD Software und Simulationsprogramm ausgetauscht werden. BIM ist ein neues Arbeitsverfahren aufgrund des IFC Standards. IFC wird als Informationsträger bei mehr und mehr Softwaretools unterstützt. Das Datenformat und die Repräsentationsmethoden von Energiesimulation beziehenden Daten werden in Kapitel 3 weiter erzählt.

2.2 Simulationssoftware

Heutzutage gibt es vielfältige Simulationssoftware für Gebäude energetische Simulation. In der vorliegenden Tabelle 3 werden einige Simulationsprogramme aufgezeigt.

Programm	Hersteller/Anbieter	Andwendung
ADELIN	Lawrence Berkeley National Laborator	Tageslichtbeleuchtung, elektrische Beleuchtung, vollständige Gebäudesimulation, Zweckbauten, 3-D CAD, Modellierung
COMIS		Mehrzonon Gebäudedurchströmung
DOE-2		Energiebedarf, Planung, Modernisierung, Forschung, Wohn- und Zweckbauten
ENERGY PLUS		Kopplung von BLAST und DOE-2, Integration COMIS
SPARK		Objektorientiert, komplexe Systeme, Energiebedarf
TRNSYS	Solar Energy Laboratory, University of Wisconsin-Madison	Planung, Modernisierung, Energiebedarf, komplexe Systeme, Zweckbauten, Kopplung mit COMIS
IES <VE>	Integrated Environmental Solutions	Energiesimulation, Umweltschutz, kommerzielle Gebäude, Wohngebäude, Visualisierung, komplexe Gebäude und Systeme
DEROBLTH	Lund Institute of Technology	Energiebedarf, Heizung, Kühlung, thermischer Komfort

Tabelle 3 : Simulationssoftware

In der vorliegenden Arbeit wird *IES <VE>* als Beispiel untersucht. Das Rechenprogramm *IES <Virtual Environment>* (*IES <VE>*) ist eine umfassende Simulationssoftware. Sie ist in der Lage, dynamischen Energieverbrauch, Beleuchtung, Heizungs- und Lüftungsanlagen zu simulieren. Das Ergebnis wird deutlich in einer Grafik beschrieben. *IES <VE>* hat auch Nachteile. Die Modellierungsfähigkeit ist nicht ausreichend. Deswegen wird ein Gebäudemodell in BIM-Software wie *Revit* aufgebaut und dann als Eingabedaten in *IES <VE>* importiert.

Mit *IES <VE>* können die Fachleute ein einheitliches physikalische Gebäudemodell aufbauen und zur Simulation und Berechnung nutzen. Die Abbildung 11 zeigt alle Module, die bei *IES <VE>* angeboten werden. Die Module *ModelIT*, *SunCast*, *ApacheSim* und *MachoFlo* sind in der Lage, den thermischen Zustand eines Gebäudes dynamisch zu simulieren. Das Modul *ApacheHVAC* hat alle Funktionen über technische Gebäudeausrüstungen.

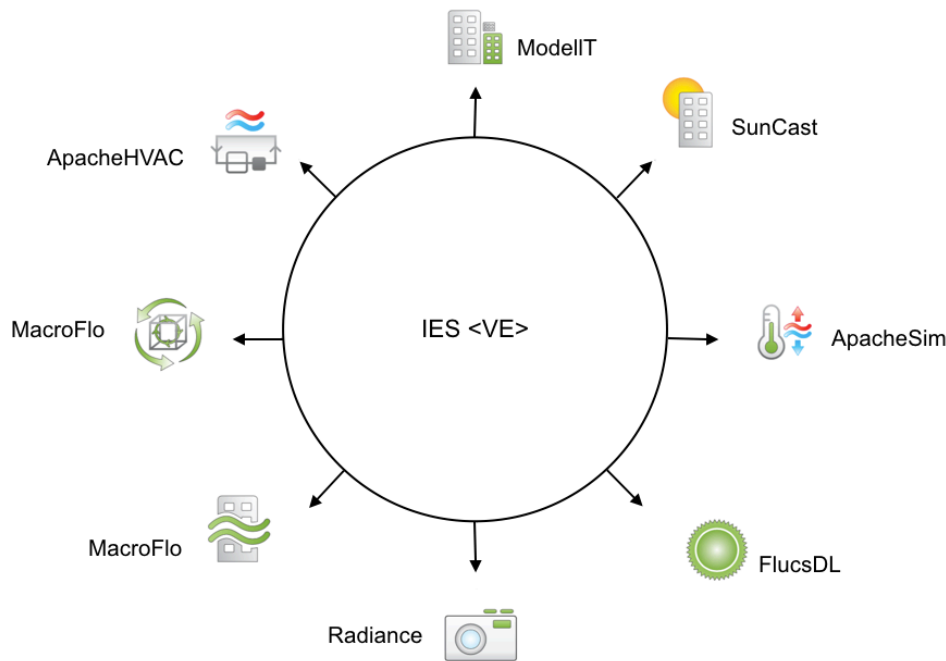


Abbildung 11 : Module in *IES <VE>* (*IES <VE> Handbook*)

Um die geometrischen Informationen eines Gebäudes zu definieren, wird das Modul *ModellT* verwendet. Damit können Fachleute ein ausreichend genaues architektonisches Gebäudemodell erstellen. *SunCast* analysiert, wie Solargewinne das Gebäude beeinflussen. Es hat die Funktion, die Sonnenstrahlung auf der Fassade und Innenflächen sichtbar zu machen. Darüberhinaus wird der Sonnenschutz eines Gebäudes beurteilt. Diese Auswirkungen werden auch in Bezug auf den Wärmegewinn und den Energieverbrauch verwendet. Zur thermischen Simulation spielt das Modul *ApacheSim* eine große Rolle. Es hat die Funktion, die grundlegenden Wärmeleitungs-, Strahlungs- und Konvektionsprozesse an jedem Bauelement zu analysieren. Aufgrund der Analysis werden die Bilanzgleichungen für fühlbare und latente Wärme erstellt und dann werden die Ergebnisse mithilfe der Iterationsverfahren berechnet. Mit *FlucsDL* können die Fachleute Beleuchtungsstärken und Tageslichtfaktoren auf jeder Oberfläche im Modell oder auf bestimmten Arbeitsebenen (z. B. die Höhe eines Schreibtisches) berechnen. *Radiance* ist ein Modell zur Bildsynthese. Es hat die Fähigkeit, eine Abbildung mit hervorragender Qualität zu erstellen. Das Modul *MacroFlo* ermöglicht es, dass Luftbewegungen aufgrund von Wind- und Auftriebseffekten in einer thermischen Simulation berücksichtigt werden und aufgrund des Ergebnisses kann eine numerische strömungsmechanische Simulation (*CFD Simulation*) durchgeführt werden. Schließlich bietet *ApacheHVAC* eine große Flexibilität bei der Definition von HLK-Anlagen und Steuerungssystemen. Ein Benutzer kann entweder mit einer vordefinierten Systemkonfiguration beginnen oder angepasste Systeme erstellen, indem er eine Reihe von Komponenten zusammenstellt und die Steuerungsstrategien festlegt. Mit diesem Ansatz können Benutzer eine breite Palette von allgemeinen Systemtypen oder hochgradig angepasste Systeme modellieren. Das Modul bindet die technische Gebäudeausrüstung mit Solar-, Beleuchtungs- und Luftströmungssimulation ein. Das Ergebnis bezieht sich auf Standort, Klimazone und die Daten des Gebäudes.

3. IFC Anwendung im Bereich Energiesimulation

3.1 Die Geschichte des IFC Standards

Seit 1995 ist ein Data Format, benannt als *Industry Foundation Classes (IFC)* für die Bauindustrie bei *International Alliance for Interoperability (IAI)* veröffentlicht. Der IFC-Standard zitiert einige Standards von Standard for Exchange of Product model data (*STEP*). Er hat das Konzept des Lebenszyklus von *STEP*-Standard übernommen. Die vollständigen Gebäudeinformationen können mithilfe des offenen IFC verlustfrei zwischen zertifizierten Softwaretools ausgetauscht werden.

Der IFC-Standard hat eine lange Geschichte. In 1994 hat das Unternehmen „Autodesk“ eine Branchenallianz veranlasst, um ein System, dass auf „C++“ basiert, für Zusammenarbeit zwischen verschiedene Softwares zu entwickeln. Zwölf amerikanische Unternehmen haben an der Allianz teilgenommen. Ab dem Jahr 1995 hat die *Allianz* für alle interessierte Organisationen geöffnet. In 1997 änderte die *Allianz* ihren Namen in *IAI*. Die reorganisierte neue *Allianz* ist eine Non-Profit-Organisation, die sich für die Verbesserung und Anwendung von IFC-Standard in der Bauindustrie engagiert; bis 2005 wurde der Standard bei der Organisation „*buildingSMART International*“ entwickelt und gepflegt. Die vorliegende Abbildung 12 zeigt die Zeitleiste der Entwicklung des IFC Standards [17].

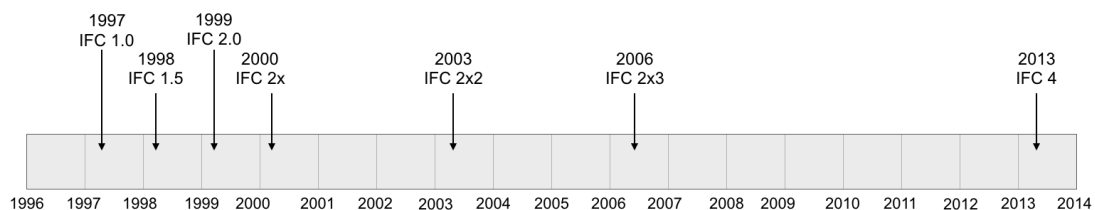


Abbildung 12 : IFC Standard Timeline

3.2 Hierarchie des IFC Standards

Die Struktur von IFC stellt ein modulares und hierarchisches Datenschema dar. Das Modell wird aus vier Ebenen mit unterschiedlichem Detaillierungsgrad zusammengestellt. In der unteren Abbildung 13 ist es einfach zu sehen, dass jede Ebene viele Module umfasst. Verschiedene Module werden verwendet, um unterschiedliche Eigenschaften eines Modells zu beschreiben.

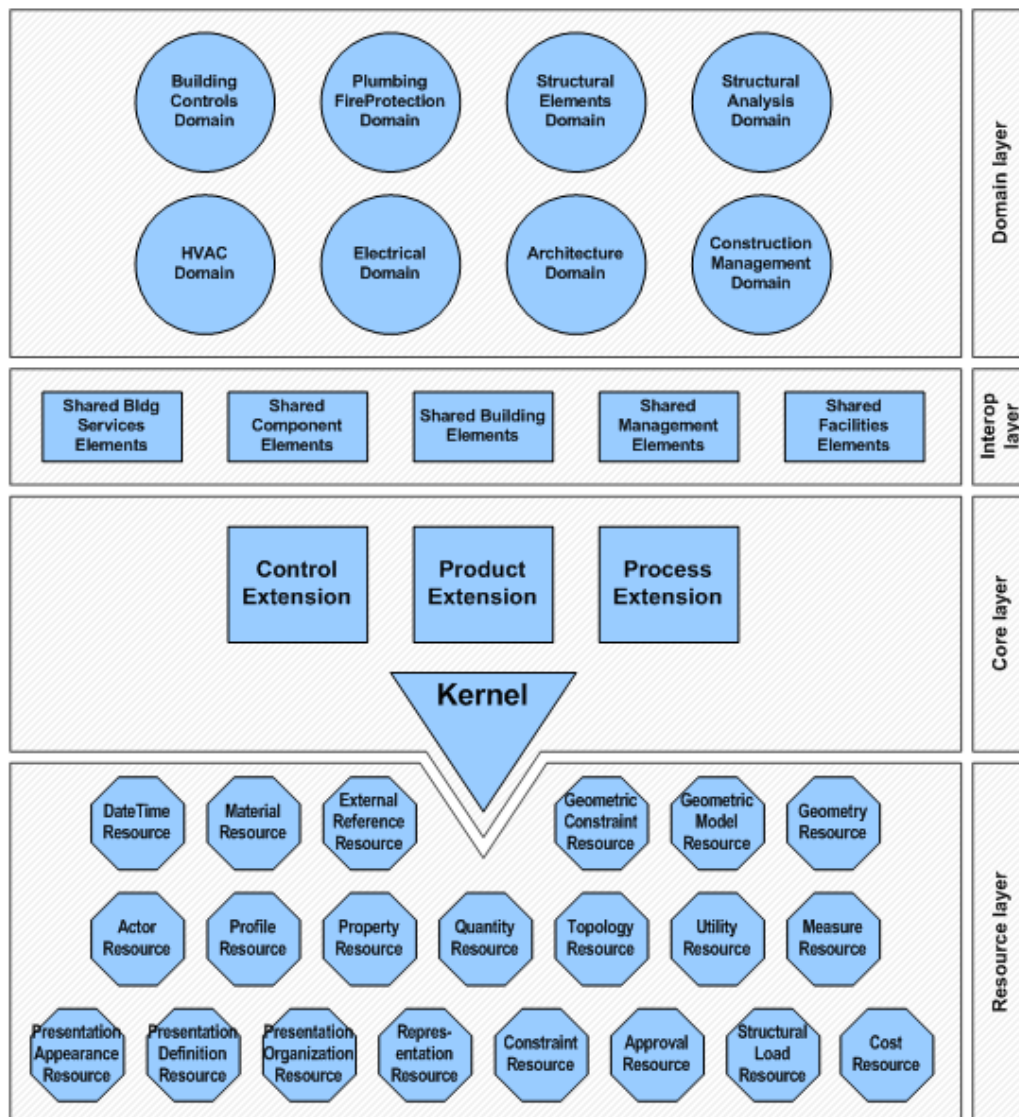


Abbildung 13 : IFC4_layered_architecture [18]

Die Ebenen von unten nach oben sind „Resource Layer“, „Core Layer“, „Interoperability Layer“ und „Domain Layer“ [18].

1. Resource Layer

Der „Resource Layer“ ist die unterste Ebene in der Struktur. Hier werden die Klassen abgebildet, die Grundinformationen beinhalten, beispielsweise zur Beschreibung von Geometrie, Stoff, Kosten, Mengen und anderem. Diese Objekte sind unabhängig vom Anwendungsbereich und werden von anderen Klassen mit höherem Level genutzt. Und sie können nur existieren, wenn sie direkt oder indirekt von einer oder mehreren Entitäten referenziert werden, die von „IfcRoot“ stammen. Im Allgemeinen existiert sie als Attributwert von Entitäten, die in den oberen Ebenen definiert werden.

2. Core Layer

Diese Schicht enthält Objekte, die abstrakte Begriffe abbilden. Sie werden auch verwendet, um Einheiten in den höheren Schichten zu definieren. Zum Beispiel enthält das Kernel-Schema Kernbegriffe wie Akteur, Gruppe, Prozess, Produkt, Beziehung und weitere, die in allen übergeordneten Einheiten des Modells verwendet werden. Das

„Product Extension“-Schema definiert abstrakte Begriffe wie Raumobjekt, Ort, Gebäude, Bauelement, Annotation usw.

3. Interoperability Layer

Diese Ebene umfasst Kategorien, die häufig verwendet werden, und zwar gleichzeitig zwischen mehreren Disziplinen in Gebäudeplanung, Bau und Betrieb. Die meisten der gemeinsamen Gebäude-Einheiten würden in dieser Schicht definiert werden, wie Wände, Decken, Stütze usw.

4. Domain Layer

„Domain Layer“ ist die oberste Ebene in der Struktur. In dieser Schicht werden die Objekte abgebildet, die für einen konkreten Anwendungsbereich spezialisiert sind. z. B. Technische Gebäude Ausrüstung, Konstruktion Analysis, Brandschutz usw.

Die vorliegende Abbildung 14 befasst sich mit strikten Referenzbeziehungen zwischen den vier Ebenen des IFC-Modells. Ein Objekt kann nur die Objekte, die sich in niedrige oder gleiche Ebene befinden nutzen. Die Referenzbeziehung ist einspurig.

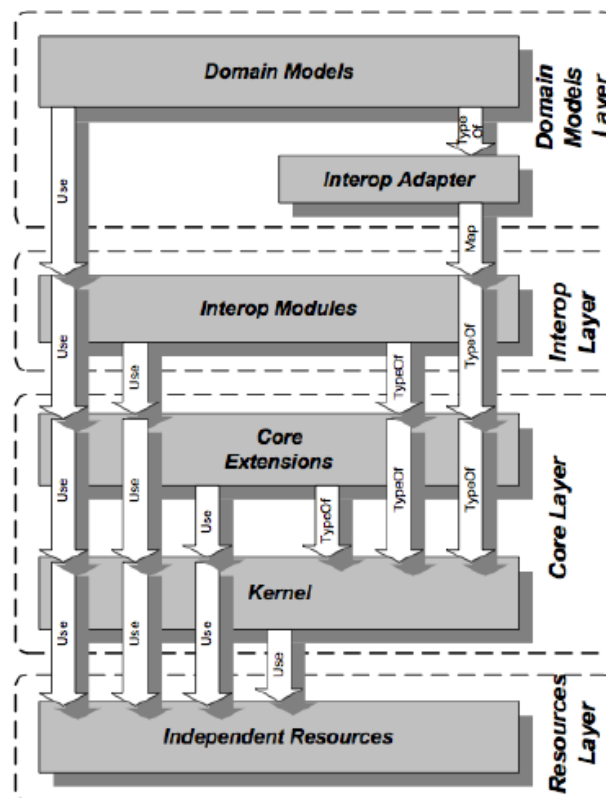


Abbildung 14 : Beziehungen zwischen IFC Ebenen [19]

3.3 Formaten für IFC Datei

IFC definiert mehrere Dateiformate, die angewendet werden können. Und die 3 wichtige Formate finden weitergehende Anwendungen in der Bauindustrie.

- IFC-SPF: IFC STEP-File ist ein Textformat in den Normen „ISO 10303-21“ definiert. Jede Zeile repräsentiert eine Entität eines Modells. Die Dateinamenserweite-

ung ist „ifc“. Die Lesbarkeit und Größe des Formates ist gut. Deswegen wird es von vielen Softwaretools genutzt.

- IFC-XML: Das Format ist ein XML (Extensible Markup Language), die in den Normen ISO 10303-28 („STEP-XML“) definiert wird. Die Dateinamenserweiterung ist „ifc-XML“. Die Vorteile davon sind vielfältig. Zuerst ist die Kapazität groß. Deswegen können viele komplizierte Modelle mit dem Format beschrieben werden. Der zweite Fall ist dass, das Format XML sehr viele Parser hat. Es ist einfach bei Programmierungssprache wie Java, C++ und Python analysiert. Zum Schluss sind die Lesbarkeit und Struktur auch besser als IFC-SPF. Heutzutage wird es mehr und mehr angewendet.
- IFC-ZIP: Das ist ein komprimiertes Format mit IFC-SPF. Die Kapazität davon ist größer und die Erweiterung ist „ifcZIP“.

3.4 Beschreibungssprache für objektorientierte Modelle

Um ein Gebäudemodell umfassend zu beschreiben, wird für die IFC die durch die ISO genormte Sprache *EXPRESS* und *EXPRESS-G* genutzt. Zwei IFC-Formen werden davon verursacht. Sie sind Textform (IFC-SPF) und grafische Form (*EXPRESS-G*).

3.4.1 *EXPRESS* und *EXPRESS-G*

EXPRESS (ISO10303-11) ist eine Modellierungssprache, die bei dem Standard „Standard for the *Exchange of Product Model Data (STEP)*“ definiert ist. Sie hängt nicht von einem bestimmten Softwaresystem ab. Es eignet sich nicht nur für Dateiaustausch, sondern auch für die Implementierung und Verteilung von Produktdatenbanken. Sie ist die Basis von dem STEP-Physical-File (IFC-SPT). Die Vorteile davon sind vielfältig. Zuerst ist, dass sie eine objektorientierte Sprache ist. Die 3 Besonderheiten davon sind Vererbung, Kapselung und Polymorphie. Alle Datei werden in ihre jeweiligen Klassen verpackt. Verschiedene Klassen sind nicht nur selbständig sondern auch einheitlich. Die Änderungen eines Objektes behindern nicht die Andere. Und die Klassen sind auch in der Lage, miteinander aufzurufen. Deswegen können die Wiederverwendbarkeit und Effizienz gestiegen sein. Der zweite Fall ist dass, die Sprache gute Schnittstelle mit ausreichenden Programmierungssprache hat, wie C++, JAVA und Python. *EXPRESS* ist eine Beschreibungssprache. Normalerweise behandelt man die IFC-SPF Datei mit Skripten und Pakete, die durch Programmierungssprache geschrieben sind. Aber sie hat auch Nachteile. Die wichtigste ist, dass die Lesbarkeit ist niedrig. Es ist schwer, die Datei Zeile für Zeile durchzulesen. Deswegen ist *EXPRESS-G* weiterverbreitet [20].

EXPRESS-G ist eine grafische Beschreibungssprache, die in der Lage ist, Beziehungen zwischen zwei Objekte oder ein Objekt und seine Attribute zu dokumentieren. Sie ist eine Variante von *EXPRESS*. Ein Vorteil ist dass, sie die komplizierte Beziehungen zwischen Klassen anschaulich darstellen kann.

3.4.2 Datentypen

Im STEP-Datei sind alle Entitäten die Ergebnisse der Instanziierung von entsprechenden Klassen. Um die Datenstruktur von *EXPRESS* zu analysieren, spielen die Klassen eine wichtige Rolle. Eine Klasse benötigt weitere Informationen, um sie vollständig zu beschreiben. Genauer gesagt werden Daten (Variablen) in einer Klasse verpackt. Wenn man eine Variable definieren will, muss man zu erste den Typ eingeben.

EXPRESS bietet eine Reihe von Datentypen und in *EXPRESS-G* gibt es entsprechende Symbole. In der vorliegenden Abbildung 15 werden 5 Datentypen in *EXPRESS-G* dargestellt.

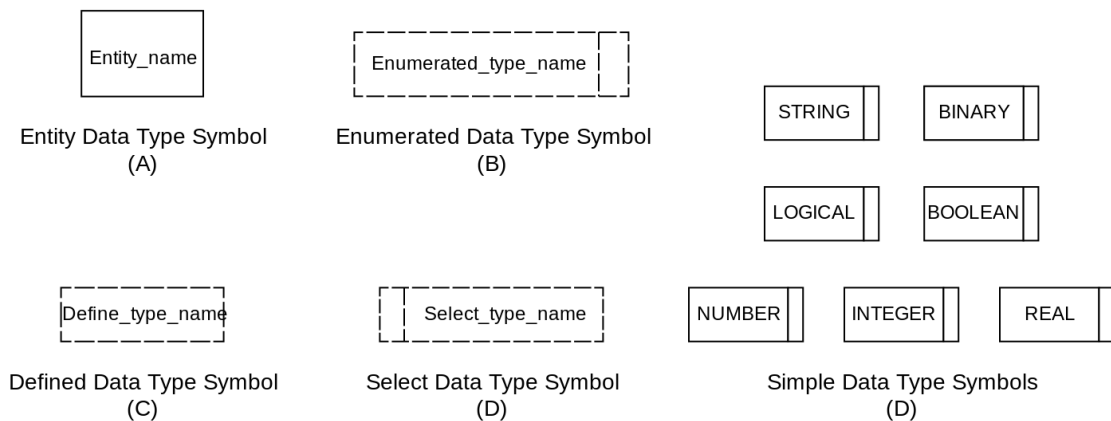


Abbildung 15 : Datentypen in *EXPRESS-G* [20]

Entity Data Type ist der wichtigste Datentyp in *EXPRESS*. Er kann als eine selbständige Entität oder als ein Attribut verwendet werden. Der erste Fall ist, dass eine Entität als Oberklasse oder Unterklasse einer anderen Entität definiert werden kann. Die Beziehung dazwischen ist Vererbung. Die abgeleitete Entität kann alle Attribute der vererbenden Entität bekommen. Der zweite Fall ist, dass die zwei Entitäten keine Vererbungsbeziehung haben. Eine davon wird als Attribut abgerufen. Diese Attribute können obligatorisch oder optional sein. Obligatorisch bedeutet, wenn eine Entität definiert wird, ein Wert des Attributes angegeben werden muss. Optional bedeutet, dass der Wert nicht erforderlich ist. In *EXPRESS-G* gibt es verschiedene Symbole für die zwei Fälle. In der Abbildung 16 geht es um die graphische Beschreibung der Beziehungen. Obligatorische Attribute werden durch eine dünne durchgezogene Linie angezeigt. Eine gestrichelte Linie stellt optionale Attribute dar. Und die dicke durchgezogene Linie kann Vererbungsbeziehungen repräsentieren.

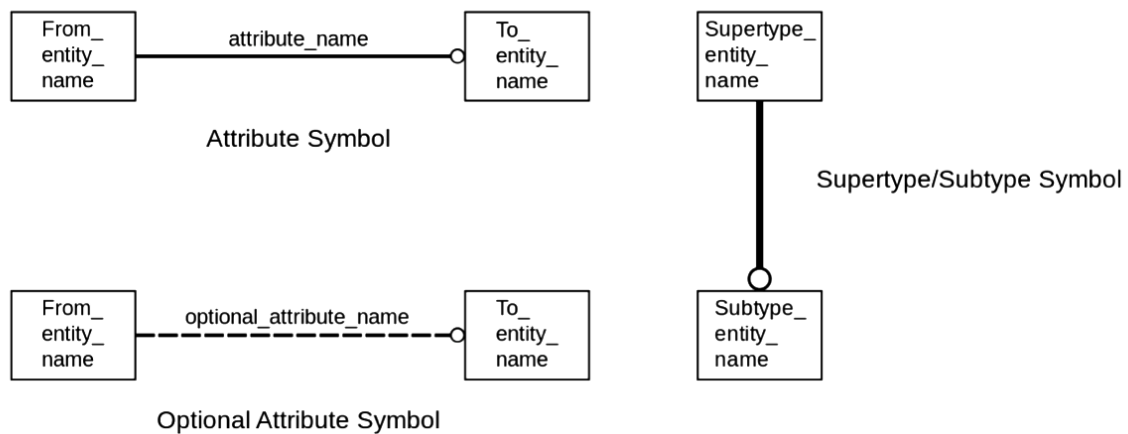


Abbildung 16 : Datentypen in EXPRESS-G [20]

Enumerated Data Type stellt einen Bereich von möglichen Werten eines Attributes in einer Aufzählungsliste bereit. Das Attribut darf nur einen Wert aus dem möglichen Bereich annehmen. Der Typ wird in EXPRESS-G als eine Box mit einer gestrichelten Linie herum dargestellt. Auf der rechten Seite gibt es Doppellinie wie in der Abbildung 17 angezeigt.



Abbildung 17 : Enumerated Data Type in EXPRESS-G[20]

Defined Data Type wird verwendet, um die Bedeutung des Modells klarer zu machen. Der Typ kann in den meisten Fällen „Simple Data Type“ ersetzen. Zum Beispiel für eine Entität IfcOrganisation gibt es ein Attribut „Description“. Der Datentyp dafür soll „STRING“ sein. Aber wie in der vorliegenden Abbildung 18 angezeigt, dass IfcText genutzt werden. In diesem Fall ist IfcText eine Variante von STRING, stellt jedoch eine bequemere Möglichkeit dar, eine Beschreibung zu behandeln.



Abbildung 18 : Defined Data Type in EXPRESS-G [20]

Select Data Type definiert eine Auswahl oder eine Alternative zwischen verschiedenen Optionen. Am häufigsten werden Auswahlen zwischen verschiedenen Entitäten verwendet. Das Symbol ist ähnlich wie Enumerated Data Type. Aber die Doppellinie befindet sich auf der linken Seite wie in Abbildung 19 angezeigt.

Zum Schluss wird Single Data Type erzählt. Er ist den kleinste Teile in EXPRESS und EXPRESS-G. Es gibt insgesamt 7 verschiedene Simple Data Type und die Symbolen werden in der Abbildung 16 (D) dargestellt.

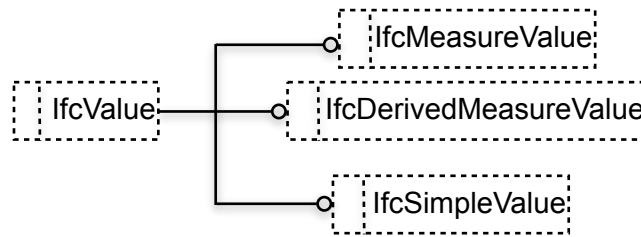


Abbildung 19 : Select Data Type in EXPRESS-G [20]

3.4.3 Datenstruktur

Eine Datenstruktur ist ein Objekt zur Speicherung und Organisation von Daten. Es handelt sich um eine Struktur, weil die Daten in einer bestimmten Art und Weise angeordnet und verknüpft werden. In *EXPRESS* werden Daten in einer Baumstruktur formuliert. Alle austauschbare Daten sind von *IfcRoot* abgeleitet. Sie haben eine einzige Identifikation (GUID). Die Entitäten, die nicht austauschbar sind, haben keine GUID. Sie können nur als Attributen aufgerufen werden. Die untere Abbildung 20 zeigt die Unterklasse von *IfcRoot*.

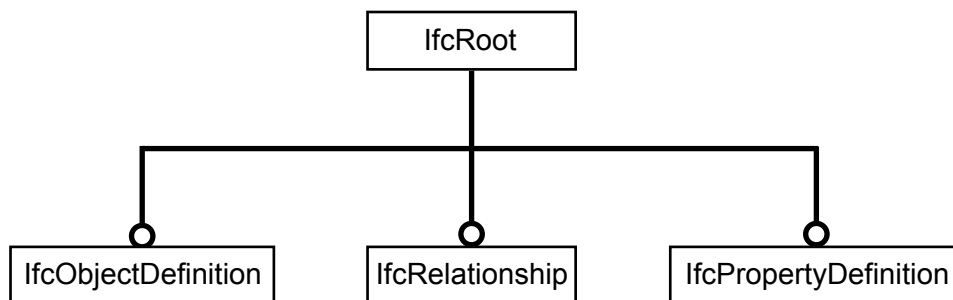


Abbildung 20 : Unterklasse von IfcRoot

IfcObjectDefinition kann verschiedene Objekten ableiten. Sie sind:

- *IfcActor* repräsentiert die Leute oder Organisationen.
- *IfcControl* stellt die Regeln dar, um die Zeit und Kosten zu kontrollieren.
- *IfcGroup* ist eine Menge von Objekten, die gleiche Verwendungszweck haben.
- *IfcProduct* zeigt die Objekten im Raum z. B. Gebäudeelemente (Wände und Fenster) und den räumlicher Standort.
- *IfcProcess* bedeutet ein Arbeitsablauf.
- *IfcResource* enthält die Informationen, die benötigt werden, um die Kosten, den Zeitplan und andere Auswirkungen der Verwendung einer Sache in einem Prozess darzustellen.

IfcRelationship ist die Superklasse für alle Objekten, die die Beziehungen zwischen Entitäten beschreiben. Und *IfcPropertyDefinition* befasst sich mit den Eigenschaften, die mit einem Objekt verknüpfen kann. Diese Baumstruktur ist die Grundlage für austauschbare Entitäten. Die drei Elternklassen sind die wichtigste Klassen für diese Aufgabe. Alle Informationen, die für Energiesimulation notwendig sind, werden davon abgeleitet.

3.5 Energiesimulation mit IFC-SPF Datei

Im Laufe des Projektablaufs wird ein Gebäudemodell bei verschiedenen Fachleute mehrmals verwendet. Aber wie in der folgenden Abbildung 21 dargestellt, ist die Anforderung des Datenaustausches in jedem Schritt unterschiedlich.

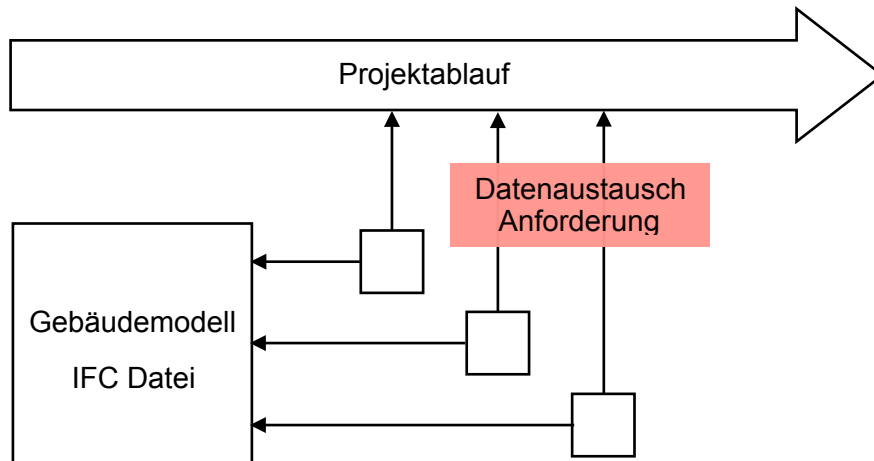


Abbildung 21 : Anforderung des Datenaustausches im Projektablauf

Um Energieverbrauch eines Gebäudes zu simulieren, benötigen Simulationsprogramme Eingabedaten und Randbedingungen. Zuerst muss man den Standort des Gebäudes und Informationen über Klima in Simulationsprogramme definieren. Dann wird das Gebäudemodell als Eingabe importiert. Zum Schluss wird die Simulation durchgeführt und das Ergebnis angezeigt. Jede erfolgreiche Simulation erfordert, erstens, gültige Daten und, zweitens, die Fähigkeit der Simulationssoftware, die Daten für einen vorgegebenen Gebäudeentwurf richtig zu bewerten. Drittens kommt die Analyse der Simulationsergebnisse hinzu, die auf diese Variante des Gebäudeentwurfs gerichtet ist. Wenn auch nur ein Teil der Daten ungültig und nicht konsistent ist, können die Ergebnisse der Simulation in Frage gestellt werden [21]. Die notwendige Eingabedaten für eine energetische Simulation eines Gebäudes sind wie in der vorliegenden Abbildung 22 angezeigt.

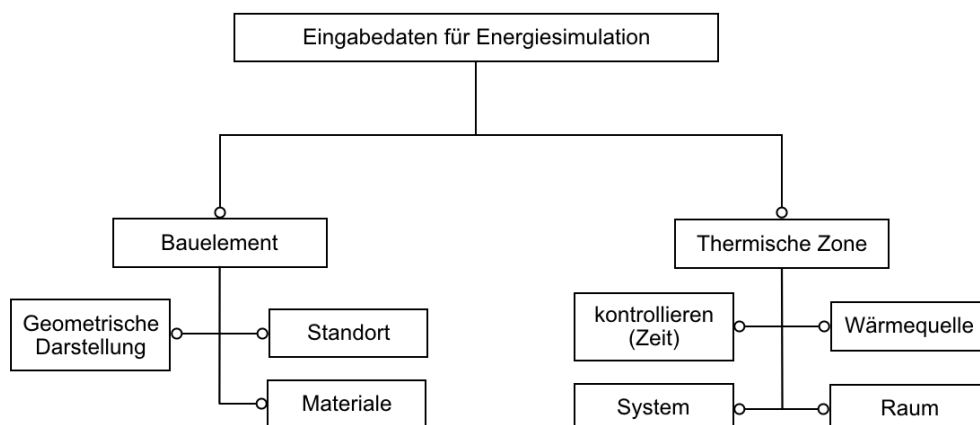


Abbildung 22 : Eingabedaten für Energiesimulation

Wenn ein Gebäudemodell in einen Simulationsprogramm aufgebaut wird, wird zuerst ein Koordinatensystem aufgestellt. Damit können bestimmte Positionen der Bauele-

menten festgestellt werden. Um die Visualisierung zu machen, spielt die geometrische Daten eine wichtige Rolle. Sie ist in der Lage, die Formen und Farbe eines Bauelementes z. B. Türen und Wände zu definieren. Deswegen muss man diese Daten aufmerksam verfolgen, wenn die IFC Datei zwischen BIM-Software und Simulationssoftware ausgetauscht wird. Um die thermischen Fähigkeit eines Bauelementes zu berechnen, müssen die Daten über Material und ihre thermischen Parameter eingegeben werden, folglich sind sie auch die Schwerpunkte. Darüberhinaus werden die Beschreibung von technischen Gebäudeanlagen, Systemen und Anlagen, und die Definition der internen Lasten sowie die Betriebszeitpläne für das Gebäude auch als Eingabedaten importiert.

3.6 Analyse der IFC-SPF Datei

Die IFC-SPF ist eine Abkürzung von IFC-STEP Physical File. Sie ist eine Textdatei und ist in der Lage, mit einem einfachen Texteditor wie beispielsweise mit Notepad unter Microsoft Windows oder mit vi unter Linux betrachtet und bearbeitet werden. Am Anfang der Datei wird „ISO-10303-21“ geschrieben und am Ende ist „END-ISO-10303-21“. Die ganze Datei kann in zwei Teile unterteilt werden. Der erste Teil ist den Kopf. Er beginnt mit „HEADER;“ und endet mit „ENDSEC;“. Inzwischen werden die Beschreibung und Auflagen z. B. IFC2X3 oder IFC4 erzählt. Der zweite Teil ist Datensegment. Er beginnt mit dem Zeichen „DATA;“ und endet mit „ENDSEC;“. Alle Informationen eines Gebäudes werden als Entitäten in diesem Bereich definiert. Die unterliegende Abbildung 23 geht es um den üblichen Rahmen. Eine Wand wird als Beispiel angezeigt.

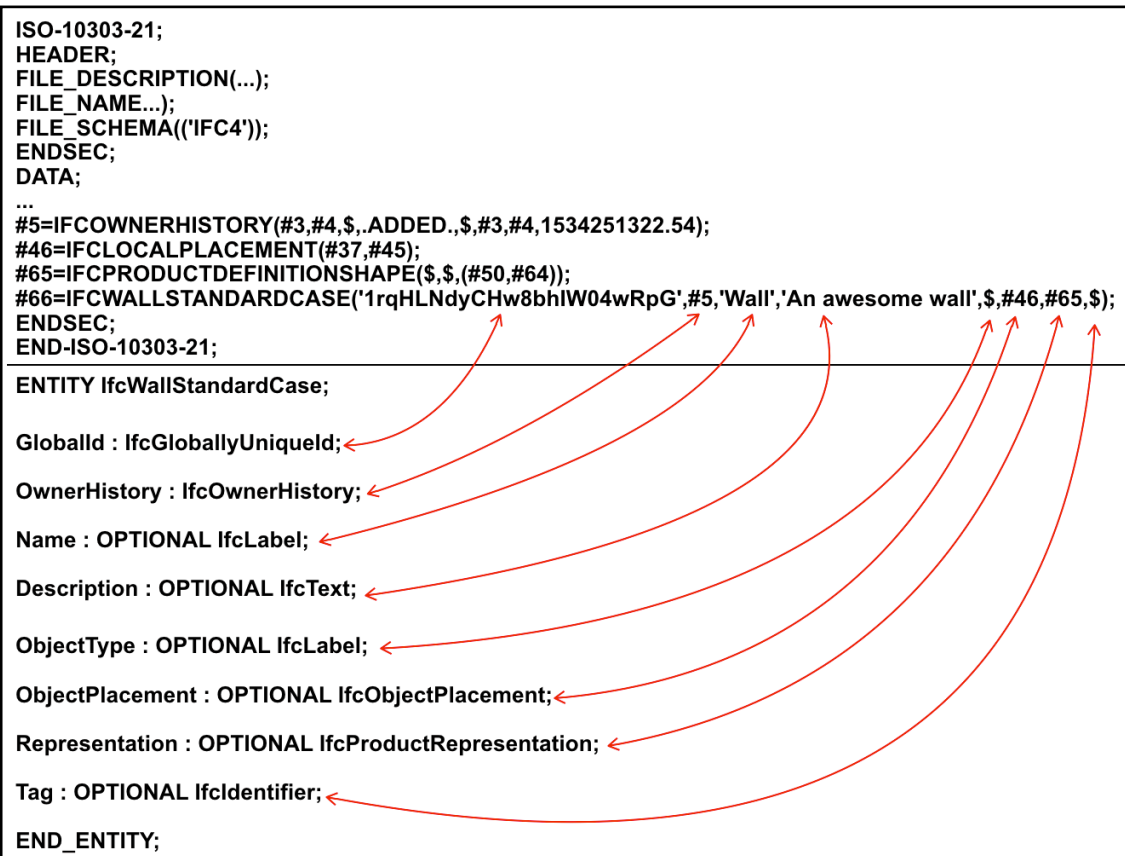


Abbildung 23 : Korrespondenz von Entität Definition und praktischen Beispiel

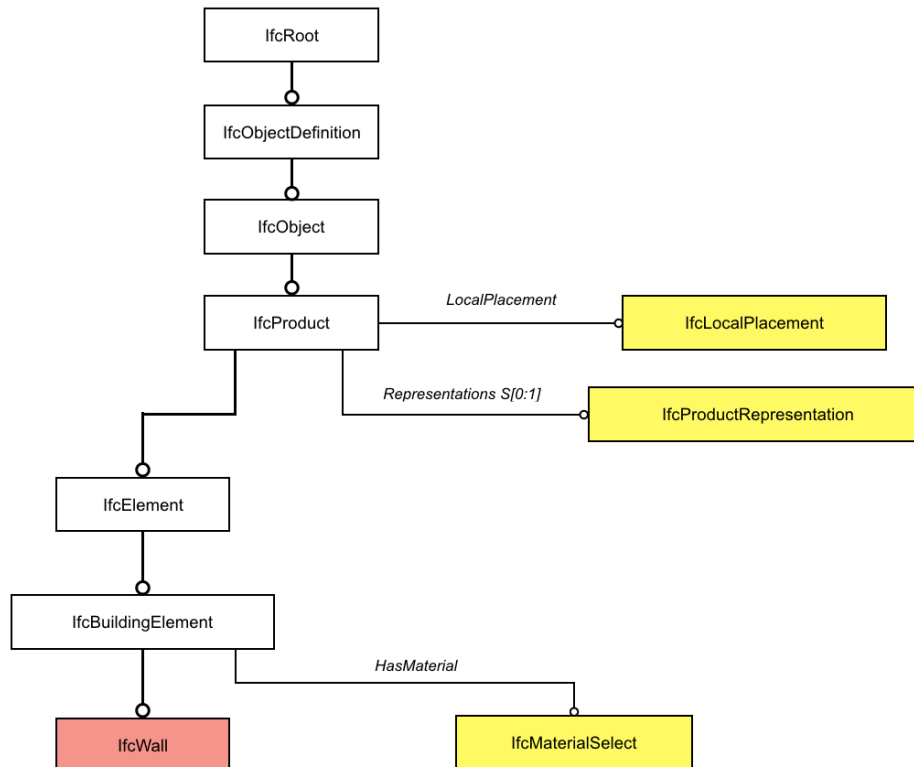
Diese Abbildung kann in zwei Teile aufgeteilt werden. Der obere Teil ist das grundlegende STEP Dateiformat und darunter ist ein Schema, mit welchem eine Wand definiert wird. Zuerst ist es einfach zu sehen, dass die Aussage mit dem name „#66“ die Wand definiert. Wenn andere Entitäten diese Wand als Attribut aufrufen, kann der name „#66“ die Wand repräsentieren. Sie ist eine von *IfcWallStandardCase* instanziierte Entität. Darum hat sie alle Eigenschaften, die im Schema definiert sind. Alle Attribute sind in Klammern geschrieben und durch ein Komma getrennt. Aber nur die Attributwerte werden hier angezeigt. Von dem unteren Teil der Abbildung wird der Attributname informiert und die Reihenfolge sind gleich. Der rote Pfeile kann die name und Werte Eins zu Eins verbinden. Wenn der Wert eines Attributs nicht definiert ist, fügt das System automatisch einen Platzhalter „\$“ hinzu. In diesem Beispiel ist „Tag“ ein optionales Attribut und sein Wert ist nicht angegeben. Deshalb erstellt eine IFC Datei automatisch einen „\$“. Im nächsten Abschnitt werden die notwendige IFC Entitäten für energetische Simulation ausführlich analysiert.

Die Analyse auf IFC Entitäten ist von kapitaler Bedeutung. Auf der einen Seite ist sie die Grundlage von der automatischen und semi-automatischen Überprüfung eines Modells. Auf der anderen Seite kann man die IFC Datei effizient behandeln. *EXPRESS* ist nicht Programmierungssprache sondern die Beschreibungssprache für Gebäudemodelle. Es gibt keine Compiler dafür. Darum kann man nicht direkt damit programmieren. Aus diesem Grund benötigt man zusätzlich eine Programmiersprache wie beispielsweise Java oder C++. Die erfolgreiche Behandlungen wie Durchsuchung, Korrigieren und Visualisierung basieren auf der Analyse der Datenstruktur der IFC Datei.

3.7 Wände

In traditionelle CAD Software wird eine Wand durch ein paar Linien und Punkte definiert. Es gibt keine Verbindung dazwischen und die Wiederverwendbarkeit ist schlecht. Aber in BIM Software wie *Revit* ist die Methode unterschiedlich. Die Wand wird durch Objektorientierte Modellierung definiert. Wenn die Wand als Daten in einer IFC Datei geschrieben wird, muss sie mit *EXPRESS* beschrieben werden. Von den obengenannten üblichen Rahmen der IFC Daten lässt sich feststellen, dass die Wand als Entität in Bereich Datensegment geschrieben wird. Die Wand stellt eine vertikale Konstruktion dar, und trennt zwei Räume voneinander. Wände sind normalerweise vertikal oder fast vertikal. *EXPRESS* haben zwei Klassen für Wände zur Verfügung.

Die erste Klasse ist *IfcWallStandardCase*. Sie wird für alle Wände verwendet, deren Dicke sich nicht entlang der Richtung der Länge verändert. Und die Dicke der Wand kann vollständig durch eine Parameterliste beschrieben werden. In dieser Liste sind die jeweils Schichtdicken der Einzelwände beschrieben. Die Zweite Klasse ist *IfcWall*. Sie definiert die Wände, deren Dicke sich verändert. z. B. polygonale Wände oder Wände mit nicht rechteckigen Querschnitten (z. B. L-förmige Stützwände). *IfcWallStandardCase* ist ein Subtyp von *IfcWall*. Deswegen ist sie in der Lage, alle Attribute von *IfcWall* zu erben. Sie definiert jedoch keine neuen Attribute von selbst. Deswegen ist es genug, um die Attribute von *IfcWall* zu untersuchen. Die untere Abbildung 24 zeigt die hierarchische Struktur von *IfcWall* [22].

Abbildung 24 : Hierarchische Struktur von *IfcWall* [7]

Aus der Abbildung ergeben sich zwei Schlussfolgerungen. Die Klasse *IfcWall* erbt von *IfcRoot*. Deswegen hat sie GUID und alle Untertypen von *IfcRoot* sind austauschbar. Die zweite Schlussfolgerung ist, dass der Standort, die Form und das Material einer Wand durch Attribute kontrolliert werden können. Die drei wichtigsten Attribute werden mit Gelbe markiert. Durch das Attribut „*LocalPlacement*“ kann man den Standort einer Wand festlegen. Das Attribut „*Representations*“ kontrolliert die Informationen über die Form einer Wand. Die zwei Attribute werden von *IfcProduct* geerbt. „*HasMaterial*“ hat die Fähigkeit, die Materialien einer Wand zu beschreiben. Es ist ein Attribut von *IfcBuildingElement*. Die drei Attribute werden im Folgenden detailliert dargestellt. Die Abbildung 25 zeigt die allgemeine Analyseverfahren für Bauelemente. Zuerst wird die Position der Wand und die Beziehungen zu den angrenzenden Elementen festgestellt. Danach ist die Definition der geometrischen Form. Der dritte Fall ist die Informationen über Materialien. Und zum Schluss wird die Darstellung der Eigenschaften der Wand betrachtet, die von Fachleute selbst definiert werden. Die ausführliche Analyse wird im nächsten Abschnitt angezeigt.

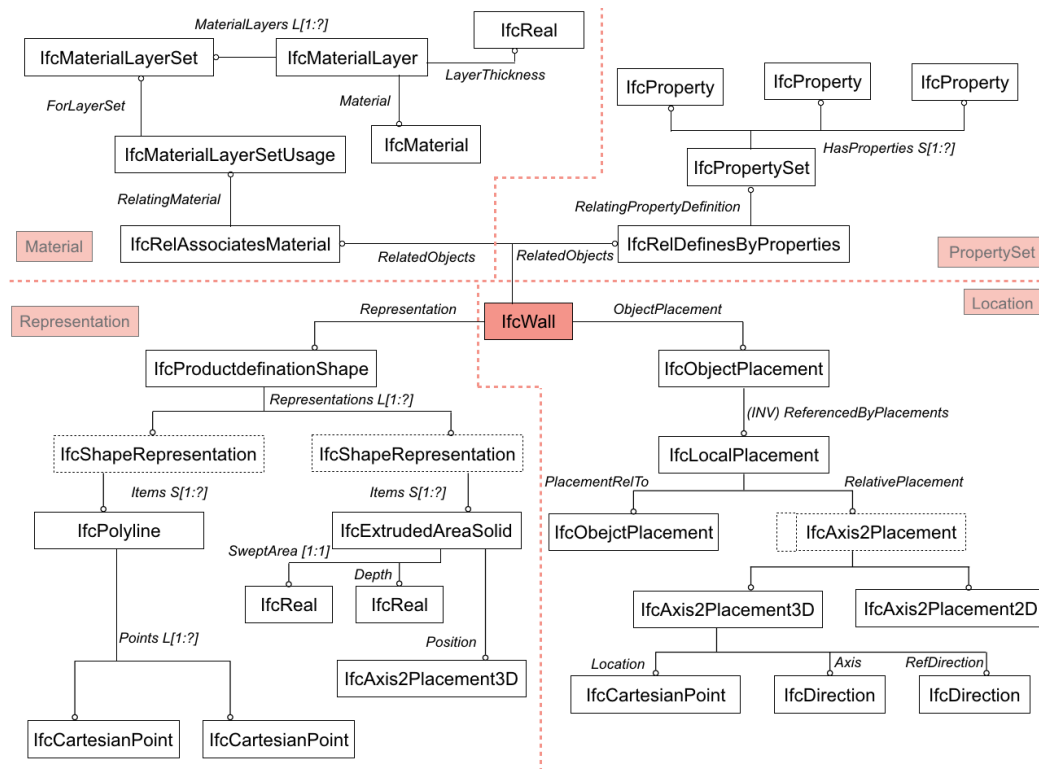


Abbildung 25 : Allgemeine Analyseverfahren für Bauelemente

3.7.1 Definition des Standortes einer Wand

In *EXPRESS* gibt es zwei Methoden, um die Lokalisation eines Elements zu bestimmen. Die eine ist die Definition der Koordinaten des Bauelementes unter dem globalen Koordinatensystem. Die andere ist, dass man die Position mithilfe des lokalen Koordinatensystem die Position festlegt. Die Platzierungsmethode, die häufigste Methode ist, basiert auf einem anderen Koordinatensystem. Sie ist die häufigste Methode in der IFC Dateien. Die vorliegende Abbildung 26 zeigt die Daten über Platzierung einer Wand unter dem lokalen Koordinatensystem.

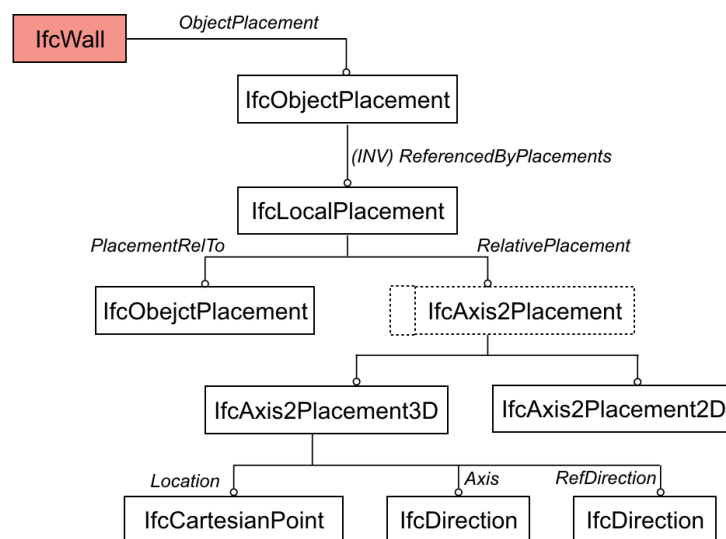


Abbildung 26 : Standortinformationen über eine Wand

Wie das Schaubild darstellt, erfolgt die Platzierung einer Wand durch eine Entität *IfcObjectPlacement*. Und die Entität *IfcLocalPlacement* wird als ein Attributwert bei *IfcObjectPlacement* aufgerufen. Sie ist der Schwerpunkt im Prozess der Platzierung. Die zwei wichtige Attribute davon sind „*PlacementRelTo*“ und „*RelativePlacement*“.

Das erste Attribut ist in der Lage, den Standort der Wand mit einem anderen Koordinatensystem zu verknüpfen. Wie in der unteren Abbildung 27 angezeigt, wird die Position der Wand mit dem Geschoss des Gebäudes verbunden. Und das Koordinatensystem eines Geschosses basiert auf dem Gebäude. Die Platzierungsmethode wird als Iterationsverfahren betrachtet. Wenn man die Position einer Wand definiert, wird zuerst die Position des Projektes festgelegt. Aufgrund des Koordinatensystems des Projektes wird die Baustelle platziert und dann wird fortgefahren.

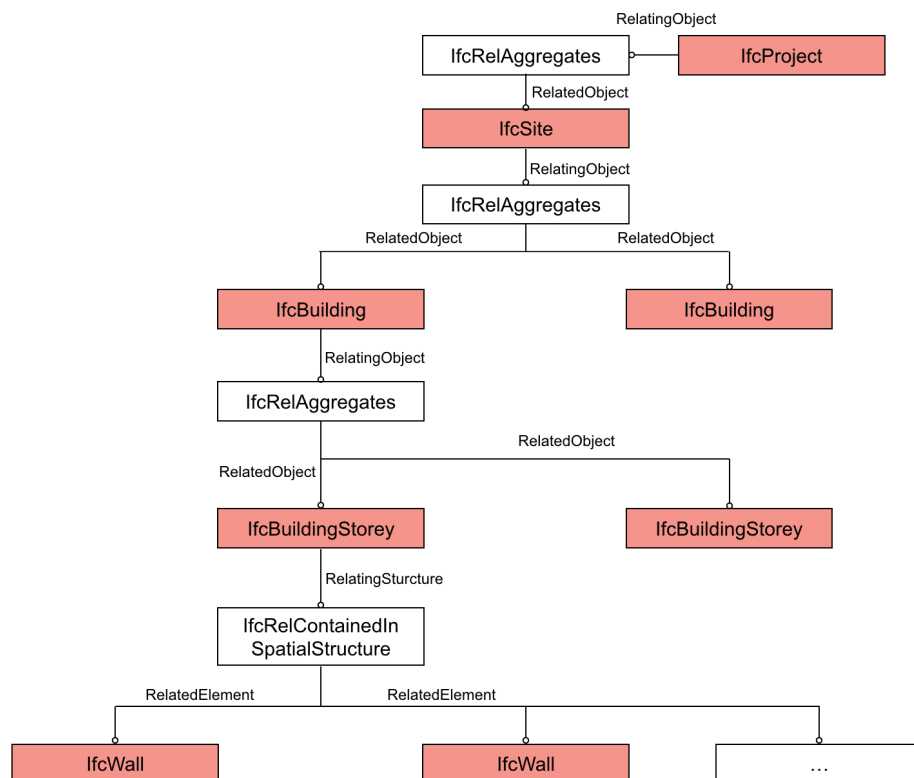


Abbildung 27 : Analyse der Platzierungsstruktur der Wand

Das zweite Attribut „*RelativePlacement*“ hat die Fähigkeit, ein lokales Koordinatensystem zu definieren. Wie in der Abbildung 26 angezeigt, wird die Entität *IfcAxis2Placement* bei *IfcLocalPlacement* aufgerufen. Sie gehört zum Select Data Typ. Dadurch kann man zweidimensionale Koordinatensysteme und dreidimensionale Koordinatensysteme auswählen. In dieser Arbeit wird das 3D Koordinatensystem betrachtet. Es ist durch die Entität *IfcAxis2Placement3D* definiert. Die drei Bestandteile sind *IfcCartesianPoint* und zwei *IfcDirection*. Mithilfe der Entitäten kann man das lokale Koordinatensystem aufbauen.

3.7.2 Die Form einer Wand

Durch das optimale Attribut „*Representation*“ von *IfcWall* wird die Entität *IfcProductDefinitionShape* aufgerufen. Sie hat die Fähigkeit, die Erscheinung eines Bauelementes

zu beschreiben und ermöglicht mehrere Formdarstellung für eine Bauelement. Wie in der Abbildung 28 angezeigt, werden zwei *IfcShapeRepresentation* durch das Attribut „*Representations*“ verwendet. Die Definition der Erscheinung einer Wand kann in zwei Schritten unterteilt werden. Am Anfang werden die Grundlinien der Wand durch *IfcPolyLine* definiert. Aus diesem Schaubild ergibt sich, dass eine *IfcPolyLine* durch eine Liste von *IfcCartesianPoint* definiert wird. Dann wird die Entität *IfcExtrudedAreaSolid* bei der zweiten *IfcShapeRepresentation* aufgerufen. Sie ist in der Lage, eine dreidimensionale Wand durch die Streckung eines Abschnittes zu definieren. Deswegen besteht die Entitäten besteht aus drei Teilen. Durch *IfcAxis2Placement3D* wird ein Koordinatensystem aufgebaut. Der zweite Teil ist das Attribut *SweptArea*. Damit wird ein Abschnitt durch ein paar Punkte definiert. Der dritte Teil ist die Höhe der Wand. Sie wird mithilfe des Attributes *Depth* erklärt.

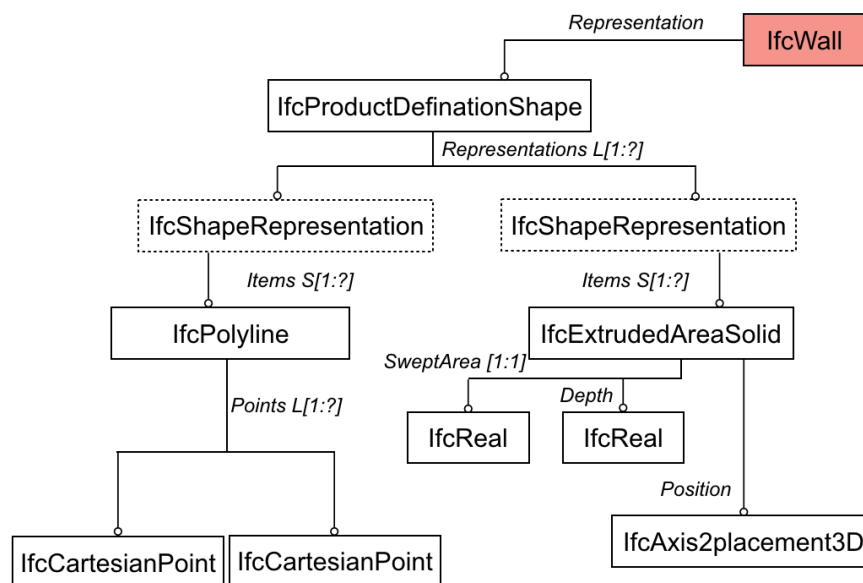


Abbildung 28 : Entitäten über Formdarstellung der Wand

3.7.3 Materialschichten der Wand

Alle im IFC-Standard definierten physischen Objekte besitzen die materielle Eigenschaften. Das Objekt *IfcMaterialSelect* stellt ein paar Entitäten dafür zur Verfügung. Wie in der Abbildung 29 angezeigt, gehört *IfcMaterialSelect* zu *Select Data Typ*. Sie definiert eine Auswahl zwischen verschiedenen Optionen. Wenn sie aufgerufen werden, kann man nur eine davon auswählen.

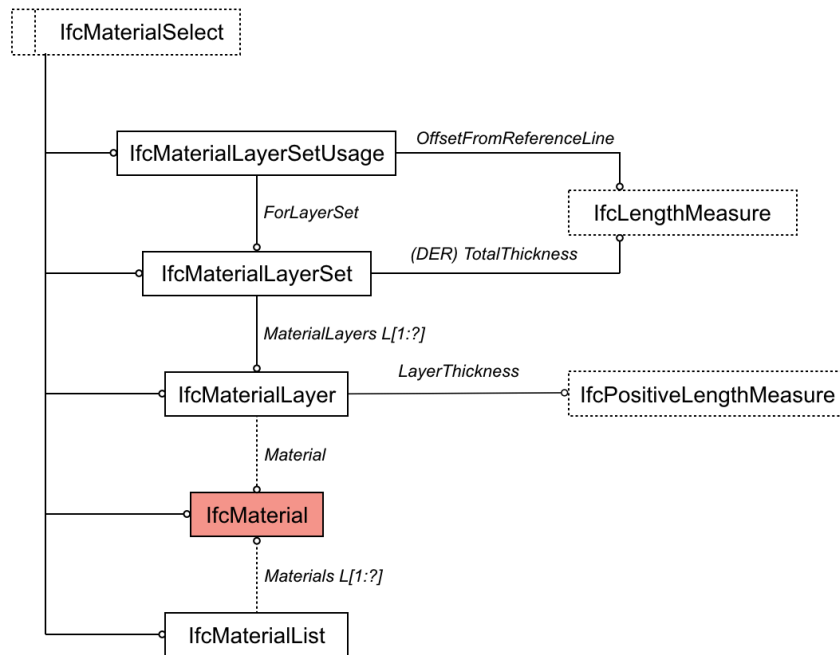


Abbildung 29 : IfcMaterialSelect

Die Entität *IfcMaterial* wird zur Beschreibung eines einzelnen Materials verwendet. Ein Material mit Dicke kann eine *IfcMaterialLayer* verursachen. Und *IfcMaterialLayerSet* besteht aus mehrere *IfcMaterialLayer*. Die Entität *IfcMaterialLayerSetUsage* ist in der Lage, die Anwendung der *IfcMaterialLayerSet* zu optimieren. Die Reihenfolge der Materialschichten wird dadurch gezeigt.

In Bezug auf die Wand ist die Beschreibungsstrategie gleich. Eine Wand besteht aus ein paar Materialschichten und in der Abbildung 30 wird die richtige Anordnung und dafür relevante Klassen in *EXPRESS-G* angezeigt.

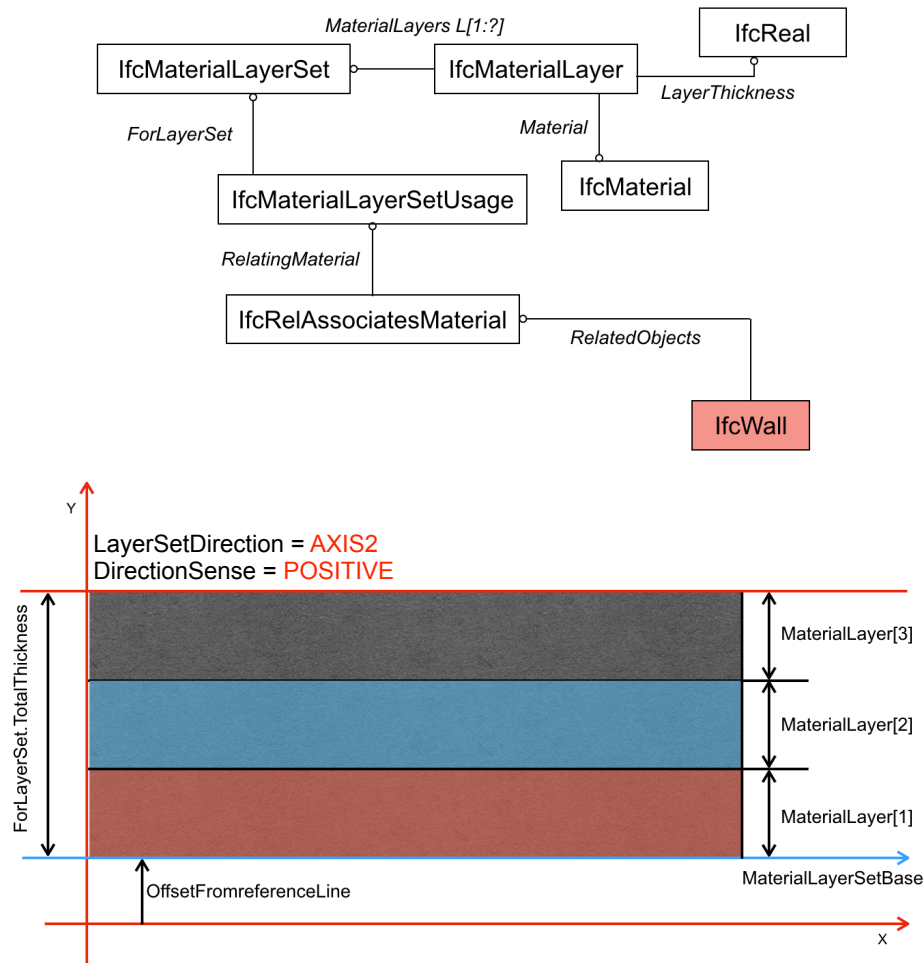


Abbildung 30 : Materialien Anordnung der Wand (BuildingSMART)

Der obere Teil der Abbildung ist die grafische Darstellung der Beziehung zwischen *IfcWall* und *IfcMaterial*. Mithilfe von *IfcRelAssociatesMaterial* wird die Verbindung von *IfcWall* und *IfcMaterialLayerSetUsage* aufgebaut. Und dann können die Fachleute passendes Material und Dicke auswählen und zusammenbinden. Der untere Teil ist ein Beispiel für die Anordnung der Materialien. Das Koordinatensystem ist ein durch das Attribut „*ObjectPlacement*“ der Wand definiertes lokales Koordinatensystem. Die Dicke der Wand ist die Summe von diesen Materialschichten. „*LayerSetDirection*“ und „*DirectionSense*“ sind zwei Attribute von *IfcMaterialLayerSetUsage*. Das Erste hat die Fähigkeit, die Richtung der Materialschichten zu definieren. Für eine Standardwand ist der Wert immer *AXIS2* (*Y*-Achse). Mithilfe von „*DirectionSense*“ wird die Reihenfolge der Materialschichten bestimmt. Der Wert *POSITIVE* bedeutet, dass sie von links nach rechts geordnet werden. Für die Berechnung der thermodynamischen Eigenschaften einer Wand, muss man die thermischen Attribute der Materialien beachten. Sie werden durch *IfcPropertySet* in der IFC Datei definiert. Die Anwendungen und Definitionsmethoden werden im nächsten Abschnitt angezeigt.

3.7.4 Eingestellte Attribute der Wand

Die Definition gemeinsamer Eigenschaften einer Wand ist im IFC Standard durch *IfcPropertySet* erfolgreich. Wie die Abbildung 31 zeigt, wird die *IfcWall* durch die Entität *IfcRelDefinesByProperties* mit *IfcPropertySet* verknüpft. Die Entität *IfcPropertySet* besteht aus einer Reihe von *IfcProperty*, die in der Lage ist, eine Eigenschaft der Wand zu beschreiben.

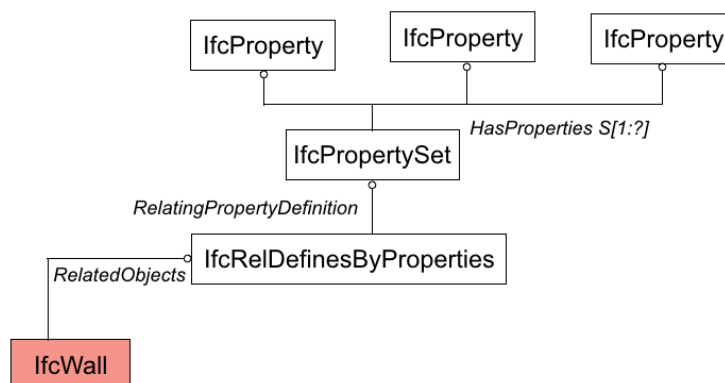


Abbildung 31 : Definition der Eigenschaften der Wand

Die Entität *IfcPropertySet* definiert alle dynamische erweiterbare Eigenschaften eines Bauelementes. Für unterschiedliche Bauelemente sind die Namen von *IfcPropertySet* unterschiedlich. Zur Beschreibung der Eigenschaften der Wand muss die *IfcPropertySet* „Pset_WallCommon“ benannt werden. Die vorliegende Tabelle 4 zeigt alle Eigenschaften, die bei *Pset_WallCommon* beinhaltet sind.

Pset_WallCommon		
Property	Data type	
Reference	P_SINGLEVALUE	IfcIdentifier
Status	P_ENUMERATEDVALUE	IfcLabel
AcousticRating	P_SINGLEVALUE	IfcLabel
FireRating	P_SINGLEVALUE	IfcLabel
Combustible	P_SINGLEVALUE	IfcBoolean
SurfaceSpreadOfFlame	P_SINGLEVALUE	IfcLabel
ThermalTransmittance	P_SINGLEVALUE	IfcThermalTransmittanceMeasure
IsExternal	P_SINGLEVALUE	IfcBoolean
ExtendToStructure	P_SINGLEVALUE	IfcBoolean
LoadBearing	P_SINGLEVALUE	IfcBoolean
Compartmentation	P_SINGLEVALUE	IfcBoolean

Tabelle 4 : Eigenschaften in Pset_WallCommon

Für Energiesimulation eines Gebäudes muss man zwei Eigenschaften einer Wand beachten, die sich in *Pset_WallCommon* befinden.

- *ThermalTransmittance* repräsentiert den Wärmedurchgangskoeffizient (U-Wert). Er ist ein spezifischer Kennwert eines Bauteils. Er wird im Wesentlichen durch die Wärmeleitfähigkeit und Dicke der verwendeten Materialien bestimmt, aber auch durch die Wärmestrahlung und Konvektion an den Oberflächen. Die Einheit ist daher $W/(m^2 \cdot K)$ (Watt pro Quadratmeter und Kelvin).
- *IsExternal* ist die Angabe, ob dieses Bauteil ein Außenbauteil ist (JA) oder ein Innenbauteil (NEIN). Als Aussenbauteil grenzt es an den Aussenraum (oder Erdreich, oder Wasser) [18].

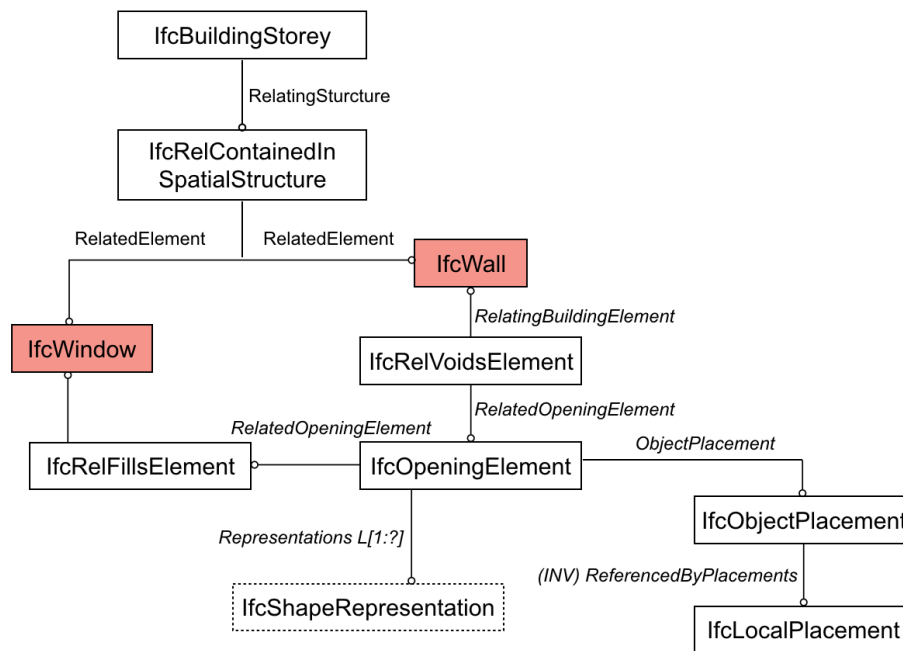
3.8 Fenster

Informationen über Fenster werden in der IFC Dateien durch die Entität *IfcWindow* oder *IfcWindowStandardCase* (IFC 2X4) definiert. Sie sind Unterklassen von *IfcBuildingElement*, darum sind sie in der Lage, die Eigenschaften von *IfcBuildingElement* zu erben. Die Unterschiede dazwischen sind:

- *IfcWindowStandardCase* ist die Unterklasse von *IfcWindow*. Und Sie haben die gleiche Attribute.
- *IfcWindowStandardCase* ist für die Situation geeignet, die ein Fenster geometrische Darstellung für den Umriss und Auskleidungs- und Rahmeneigenschaften hat. Außerdem muss ein *IfcWindowType* bereitgestellt werden, der auf *IfcWindowLiningProperties* und auf viele *IfcWindowPanelProperties* verweist.
- *IfcWindow* wird für alle anderen Vorkommen von Fenstern verwendet, insbesondere für Fenster, die nur „Brep“ oder „SurfaceModel“ Geometrie haben, ohne die Formparameter anzuwenden.

Mit der gleichen Strategie wird die Daten über Fenster unter vier Aspekten betrachtet. Zur Vereinfachung der Analyse wird *IfcWindow* als Beispiel betrachtet.

Die Abbildung 32 zeigt die Entitäten über Platzierung der *IfcWindow*. Es ist einfach zu erfahren, dass *IfcWindow* und *IfcWall* als „*RelatedElement*“ mithilfe von *IfcRelContainedInSpatialStructure* mit *IfcBuildingStorey* verbunden werden. Aber die beiden Entitäten haben keine direkte Verbindung. Für die Verknüpfung dazwischen spielt die Entität *IfcOpeningElement* eine bedeutende Rolle. Sie definiert eine Öffnung an der Wand. Sie sollte nicht direkt mit der räumlichen Struktur des Projekts wie *IfcBuildingStorey* verknüpft werden, sondern durch *IfcRelVoidsElement* mit *IfcWall* verbunden und durch *IfcRelFillsElement* mit *IfcWindow* verbunden wird. Ihre Lokalisation ist durch lokales Koordinatensystem mithilfe der *IfcLocalPlacement* erfolgreich. Zur Bestimmung des Standortes der *IfcWindow* wird ein lokales Koordinatensystem auf *IfcOpeningElement* basiert. Ohne *IfcOpeningElement* existiert die Verbindung zwischen Fenster und Wand nicht und Fehler wird verursacht. Wenn die Modellüberprüfung durchgeführt wird, muss man sich darum als einen Schwerpunkt kümmern.

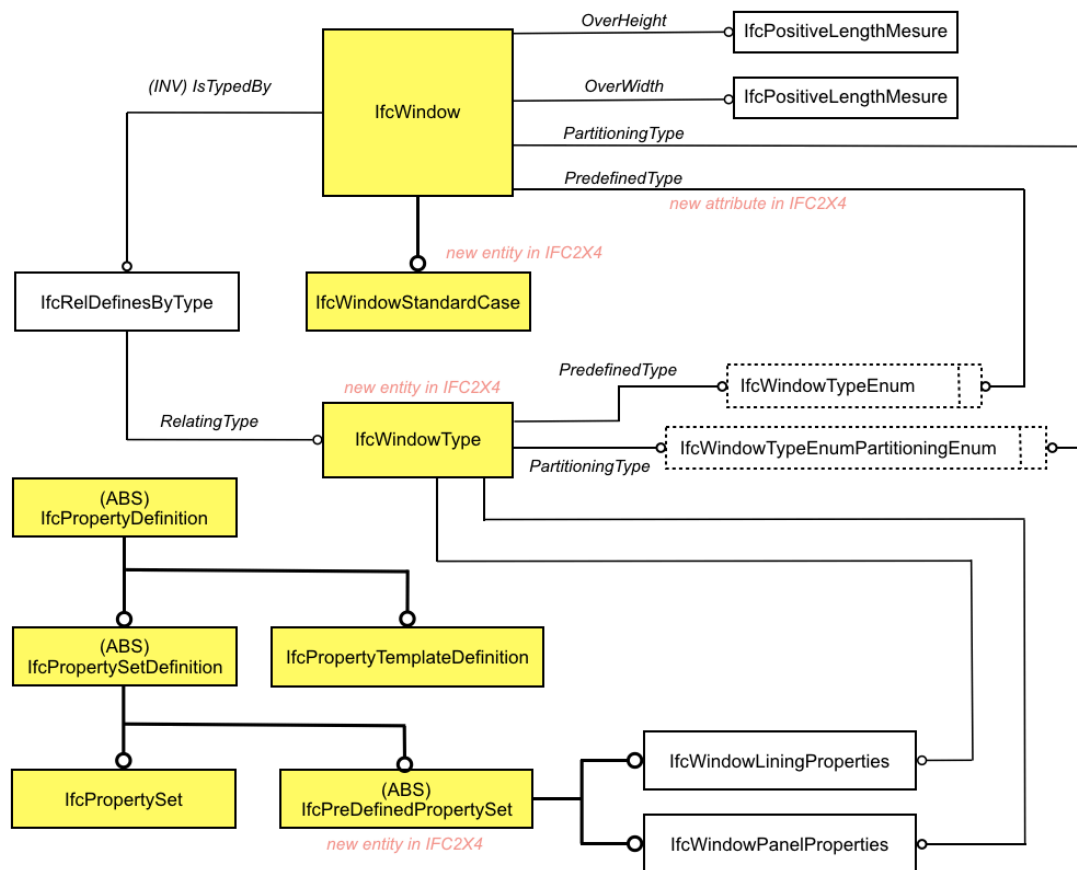
Abbildung 32 : Räumliche Struktur von *IfcWindow*

Die Parameter zur Darstellung der Form eines Fensters werden bei der Entität *IfcWindow* oder *IfcWindowType* mithilfe von *IfcProductDefinitionShape* und ihre Untertyp *IfcShapeRepresentation* angegeben. Das Fensterprofil wird durch eine dreidimensionale geschlossene Kurve innerhalb einer bestimmten Formdarstellung dargestellt. Und der Körper einer Wand wird durch Verwendung der Extrusionsgeometrie bestimmt, wie *IfcWall*. Die zwei Bestandteile stehen mit den Attributen von *IfcShapeRepresentation* im Zusammenhang aber die Darstellungsmethoden sind unterschiedlich. In der Tabelle 5 wird der Parameter dafür angezeigt.

Attribut	Profile	Körper
<i>RepresentationIdentifier</i>	<i>Profile</i>	<i>Body</i>
<i>RepresentationType</i>	<i>Curve3D</i>	<i>SweptSolid</i>

Tabelle 5 : Attributwerte für *IfcShapeRepresentation*

Darüberhinaus zeigt die Abbildung 33 Formparameter, die durch *IfcWindow* und *IfcWindowType* definiert sind. Es ist einfach zu sehen, dass die Höhe und Breite in der *IfcWindow* Entität direkt definiert werden. Außerdem werden die Parameter, die die Form von *IfcWindow* definieren, in *IfcWindowType* und den Attributsätze angegeben, die in *IfcWindowType* enthalten sind, z. B der Partitionierungstyp (Einzelscheibe, Doppelscheibe und mehr) und die Betriebsart der Fenster. Zur Beschreibung der Verkleidung und Rahmen der Fenster werden *IfcWindowLiningProperties* und *IfcWindowPanelProperties* zur Verfügung gestellt.

Abbildung 33 : Formparameter für *IfcWindow*

Die Definition Materialien der Fenster ist gleich wie die Wand. Sie ist durch *IfcMaterialConstituentSet* oder *IfcRelAssociatesMaterial* und *IfcMaterial* erfolgreich. In der Tabelle 6 werden die Schlüsselwörter für *IfcMaterialConstituentSet.MaterialConstituents* angezeigt.

Name	Beschreibung
<i>Lining</i>	Material für Fensterverkleidung
<i>Framing</i>	Material für Fensterrahmen
<i>Glazing</i>	Material für Scheibe

Tabelle 6 : *IfcMaterialConstituentSet.MaterialConstituents*

Die Attributsätze für *IfcWindow* werden vom *IfcPropertySet* definiert und von der *IfcRelDefinesByProperties* verknüpft wie die Wand. Es gibt drei typische Attributsätze *Pset_WindowCommon*, *Pset_DoorWindowGlazingType*, und *Pset_DoorWindowShadingType*. Die Tabelle 7 zeigt die dadurch definierten Eigenschaften, die für Energiesimulation wesentlich sind. Man muss darauf achten, wenn der Datenaustausch durchgeführt wird.

Attributsätze	Attribute Name	Datentyp	Anmerkung
<i>Pset_WindowCommon</i>	IsExternal	IfcBoolean	Außenbauteil
	Infiltration	IfcVolumetricFlowRateMeasure	Luftdurchlässigkeit
	ThermalTransmittance	IfcThermalTransmittanceMeasure	U-Wert
	GlazingAreaFraction	IfcPositiveRatioMeasure	Glasflächenanteil
<i>Pset_DoorWindowGlazingType</i>	GlassLayers	IfcCountMeasure	Schichten
	GlassThickness	IfcPositiveLengthMeasure	Dicke der Schicht
	FillGas	IfcLabel	Gas zwischen zwei Glasschichten
	SolarAbsorption	IfcNormalisedRatioMeasure	Strahlungsabsorptionsgrad
	SolarReflectance		Strahlungsreflectionsgrad
	SolarTransmittance		Strahlungstransmissionsgrad
	SolarHeatGainTransmittance		Gesamtenergiedurchlassgrad
	ShadingCoefficient		mittlere Durchlassfaktor b
	ThermalTransmittanceSummer	IfcThermalTransmittanceMeasure	U-Wert im Sommer
	ThermalTransmittanceWinter		U-Wert im Winter
<i>Pset_DoorWindowShadingType</i>	ReductionCoefficient	IfcNormalisedRatioMeasure	g-Wert

Tabelle 7 : Attributsätze für *IfcWindow*

3.9 Türen

Als Bauelement werden die Daten über Türen mit gleiche Strategie wie die Fenster analysiert. Wie in der Abbildung 34 angezeigt, wird die Platzierung der *IfcDoor* durch *IfcWall* bestimmt. Darüberhinaus werden die Parameter für Form und Materialien durch *IfcShapeRepresentation*, *IfcDoorType* und *IfcMaterial* kontrolliert. Durch *IfcDoorType* werden die Öffnungsverfahren der Tür mithilfe der Entität *IfcDoorTypeOperationEnum* definiert. Und in der IFC 2x4 wird eine neue Entität *IfcDoorTypeEnum* hinzugefügt, um den Typ der Tür zu definieren. Aber für Energiesimulation eines Gebäudes spielen die zwei Teile keine Rolle, darum wird der Bericht nicht ins Detail gehen. Die Attributsätze für *IfcDoor* sind *Pset_DoorCommon*, *Pset_DoorWindowGlazingType*, *Pset_DoorWindowShadingType*. Der Inhalt davon ist wie in der Tabelle 7 angezeigt. Sie sind der Schwerpunkt im Prozess des Datenaustausches.

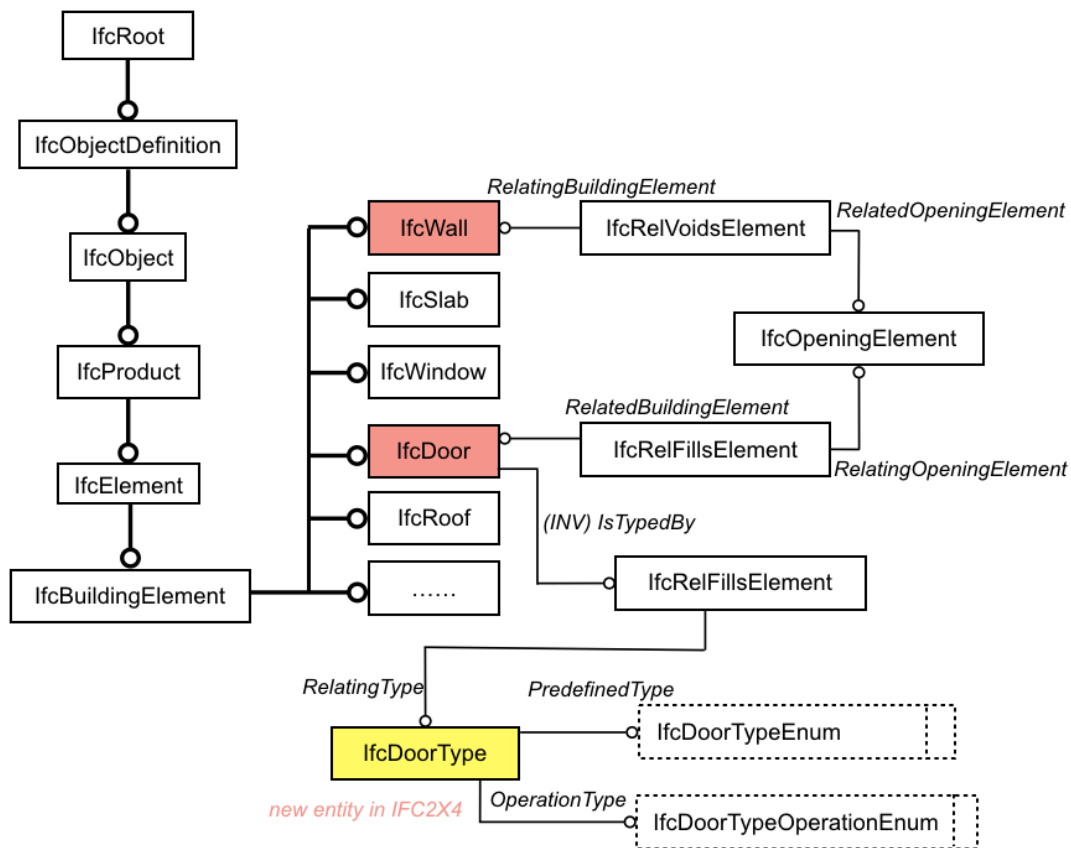


Abbildung 34 : Eigenschaften von IfcDoor

3.10 Thermische Eigenschaften von Materialien

Die thermodynamischen Eigenschaften eines Bauelementes basieren auf ihren Materialien und für vielschichtige Elementen wie Wand und Boden muss man die Parameter jeder Schicht definieren. Wenn eine IFC Datei als Eingabedaten zur Energiesimulation verwendet wird, werden die Daten über Materialien beachtet werden. Die Abbildung 35 zeigt das *EXPRESS-G* Diagram über materiale Definition. Es ist einfach zu sehen, dass die abstrakte Klasse *IfcMaterialDefinition* als Elternklasse der materialen Information beschreibenden Klassen wie *IfcMaterial*, *IfcMaterialLayer* gilt. Sie sind in der Lage, nicht nur das einzige Material sondern auch mehrschichtige Strukturen darzustellen. Ein mehrschichtiges Bauelement wie Wand oder Decke besteht aus vielen *IfcMaterialLayer* und jede *IfcMaterialLayer* hat nur eine *IfcMaterial*. Und die Verbindung von *IfcMaterialLayer* und das Bauelement wird durch eine Beziehungsentität *IfcRelAssociatesMaterial* kontrolliert. Die grundlegenden Eigenschaften eines Materials wie Name und Beschreibung kann man durch *IfcMaterial* direkt definieren. Aber die thermischen Parameter wie U-wert und Dichte werden bei der abstrakten Klasse *IfcMaterialProperties* angeboten. Die Verknüpfung zwischen einem Material und seinen Parametern ist durch das Attribut „*HasProperties*“ möglich. Die Klasse *IfcMaterialProperties* hat ein paar Unterklassen, die in der Lage sind, Attribute in verschiedenen Bereichen zu beschreiben. Wie in der unteren Abbildung angezeigt, ist die Definition der thermischen Parameter eines Materiales durch *IfcThermalMaterialProperties*, *IfcGeneralMaterialProperties* und *IfcExtendedMaterialProperties* erfolgreich.

<i>IfcExtendedMaterialProperties</i> (For Energy Calculation)	<i>ViscosityTemperatureDerivative</i>	<i>IfcReal</i>	<i>kg/(m·K)</i>
	<i>MoistureCapacityThermalGradient</i>		<i>kg/(kg·K)</i>
	<i>ThermalConductivityTemperatureDerivative</i>		<i>W/(m·K²)</i>
	<i>SpecificHeatTemperatureDerivative</i>		<i>J/(kg·K²)</i>
	<i>VisibleRefractionIndex</i>		—
	<i>SolarRefractionIndex</i>		—
	<i>GasPressure</i>	<i>IfcReal</i>	<i>Pa</i>

Tabelle 8 : Thermischen Parameter für Materialien

3.11 Räume

Üblicherweise wurden die digitalen dreidimensionalen Modelle ursprünglich für andere Zwecke, zum Beispiel für eine architektonische oder visuelle Darstellung des Gebäudes erzeugt. Solche Modelle erfüllen aber nicht die erforderliche Qualität der Geometrie für eine Anwendung im Rahmen der energetischen Gebäudesimulation. Es stellt eine große Herausforderung für den Durchführenden dar, die ursprünglich modellierte Geometrie des Gebäudes für die Verwendung innerhalb der Simulationssoftware zu übernehmen oder anzupassen.

Für die Zwecke der energetischen Simulation sollen im Gebäudemodell die Raumobjekte automatisch oder manuell erstellt werden. Die Modelle, die für die energetische Gebäudesimulation akzeptabel wären, unterscheiden sich im Wesentlichen von der architektonischen Variante. Die ursprünglichen architektonischen Modelle beinhalten zu viele für die Simulation unnötige Details. Wenn sie als IFC Datei in eine Simulationssoftware importiert werden, muss man die unnötigen Daten weglassen und die für Energiesimulation benötigten Daten berücksichtigen. Das vereinfachte Modell erlaubt die direkte Erkennung und Erstellung von Raumobjekten, ohne zum Beispiel die Wandstärke oder andere Details zu berechnen. Die Bauelemente wie Wand und Decke werden nicht nur als Objekte betrachtet, sondern auch als räumliche Grenze berücksichtigt. Zur Optimierung der Qualität eines Modells wird zuerst die Datenstruktur des Raumes im IFC Standard untersucht.

Eine thermische Zone besteht aus mehreren Räumen. Die Integration von verschiedenen Räumen, die für den Aufenthalt von Menschen gedacht sind, in Zonen wird nach gemeinsamen Luftqualitätsbedingungen durchgeführt. Sie werden mit einer gleichen technischen Ausrüstung versorgt. Deswegen müssen im weiteren Verlauf der Konvertierung der Eingabedaten die Raumobjekte in den thermischen Zonen umgeformt und kombiniert werden.

Im IFC Standard werden die räumlichen Informationen in einer Entität *IfcSpace* abgelegt. Die Abbildung 36 zeigt die Vererbungsbeziehung von *IfcSpace*. Aus der Abbildung ist einfach zu sehen, dass die abstrakte Klasse *IfcSpatialStructureElement* vier Unterklassen hat. Sie stellt die Verallgemeinerung aller räumlichen Elemente dar, die zur De-

definition einer räumlichen Struktur verwendet werden können und die häufig zur Bereitstellung einer Projektstruktur zur Organisation eines Bauprojekts verwendet werden. *IfcSpace* ist das kleinste Objekt davon. Es ist in der Lage, alle Attribute von *IfcSpatialStructureElement* zu erben.

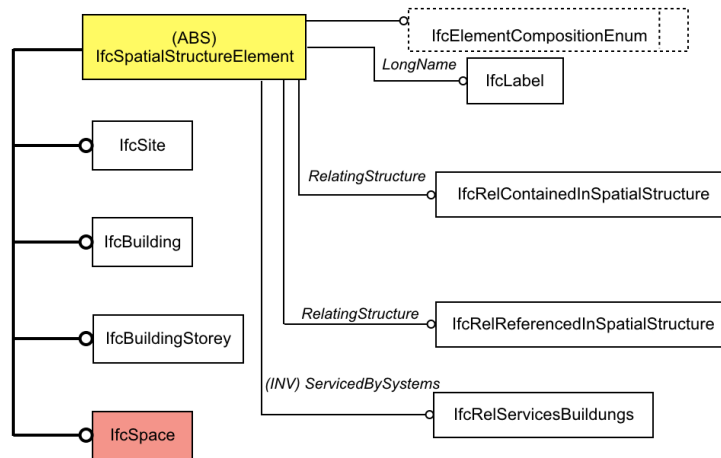


Abbildung 36 : Vererbungsbaum von IfcSpace

Wie in der obengenannten Abbildung 36 angezeigt, werden fünf Attribute bei der Klasse *IfcSpatialStructureElement* definiert. Durch *IfcElementCompositionEnum* wird der Typ des Objektes bestimmt. *IfcRelContainedInSpatialStructure* beschreibt die Beziehungen zwischen Bauelementen wie Wänden und Türen und räumlichen Strukturen wie Geschoss. Die Entität *IfcRelServicesBuildings* hat die Fähigkeit, ein technisches System und einen Raum oder eine Zone zusammenzubinden. Die Abbildung 37 zeigt das Verfahren der Entität *IfcRelReferencedInSpatialStructure*.

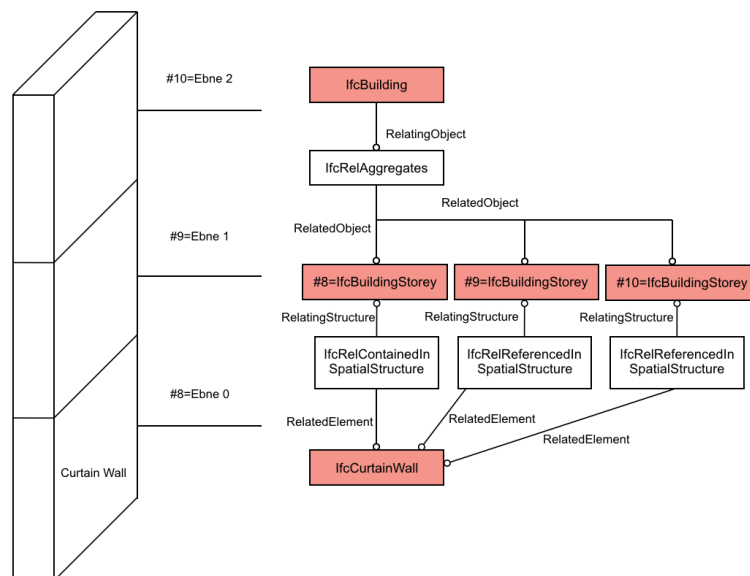


Abbildung 37 : Nutzungsverfahren der *IfcRelReferencedInSpatialStructure* (BuildingSMART)

Diese Entität definiert die Verknüpfung zwischen einem Bauelement und mehreren räumlichen Objekten. Zum Beispiel eine Vorhangsfassade wird definiert in der Ebene 0, aber sie hat auf Ebene 2 und 3 übergreifen. Die Beziehungen der Vorhangsfassade mit der Ebene 2 und 3 werden durch *IfcRelReferencedInSpatialStructure* definiert. Das

ist ein Schwerpunkt im Verlauf der Überprüfung eines Modells. Manchmal werden die übergreifenden Bauelemente wie Vorhangsfassade mehrmals definiert und man muss dies beachten.

Räumliche Grenzen sind für die energetische Simulation von großer Bedeutung. Die Definition davon wird durch die Entität *IfcRelSpaceBoundary* erzählt. Sie definiert die physische oder virtuellen Grenzen eines Raumes und die Beziehungen zwischen einem Raum und einer Umgebung. Die Abbildung 38 zeigt die Verbindung der *IfcSpace* mit der Entität *IfcRelSpaceBoundary*.

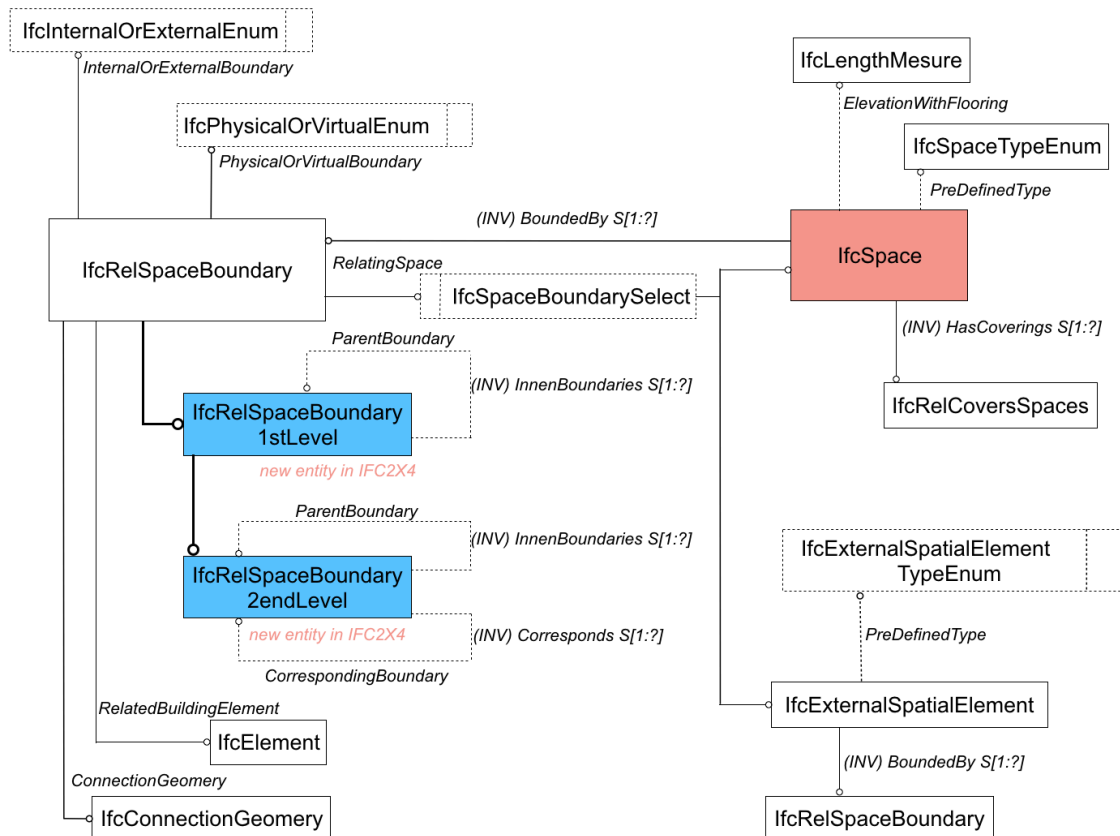


Abbildung 38 : Verbindung der *IfcSpace* mit *IfcRelSpaceBoundary*

Die Entität *IfcPhysicalOrVirtualEnum* hat drei Optionen (*PHYSICAL*, *VIRTUAL*, *NOT-DEFINED*) für den Typ einer räumlichen Grenze angeboten. Im Falle der physischen Raumgrenzen werden die Platzierungen und Formen der Grenze angegeben und das Gebäudeelement, das die Grenze bereitstellt, wird referenziert. Im Fall der virtuellen Raumgrenze können die Platzierungen und Formen der Grenze angegeben werden, jedoch wird kein Gebäudeelement referenziert. Durch *IfcInternalOrExternalEnum* wird bestimmt, ob die räumliche Grenze innere oder äußere ist. Beispielsweise wird die Grenze der Terrasse als äußere Grenze berücksichtigt. Wenn *IfcRelSpaceBoundary* verwendet wird, um eine physische Grenze zwischen zwei Räume auszudrücken, muss das Attribut *PhysicalOrVirtualBoundary* auf *PHYSICAL* gesetzt werden. Das Attribut *RelatedBuildingElement* muss angegeben werden und zeigt auf das Element, das die räumliche Grenze bereitstellt. Das Attribut *ConnectionGeometry* kann eingefügt werden. In diesem Fall beschreibt es die geometrische physische Raumgrenze. Oder es kann weggelassen werden.

Im IFC 2x4 werden die Klasse *IfcRelSpaceBoundary* um zwei neue Klassen *IfcRelSpaceBoundary1stLevel* und *IfcRelSpaceBoundary2ndLevel* erweitert. Somit kann die Bauelementtopologie detaillierter ausgedrückt werden. Wie in der obengenannten Abbildung angezeigt, werden zwei neue Attribute wie *ParentBoundary* und *CorrespondingBoundary* hinzugefügt. Das neue Attribut *ParentBoundary* wurde verwendet, um die Grenzen von zwei oder mehr überlappenden Grenzen zu beschreiben, wie Türen und Fenster. Ein anderes Attribut *CorrespondingBoundary* beschreibt die Beziehung zu gegenüberliegenden Grenzen. Die vorliegende Abbildung 39 zeigt die zwei räumliche Grenzen, die im IFC 2x4 hinzugefügt werden.

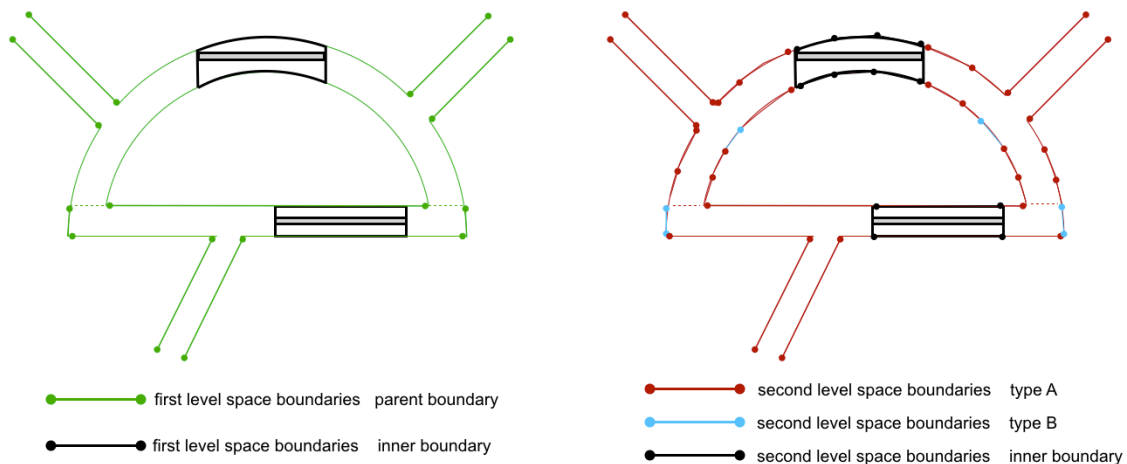


Abbildung 39 : Zwei räumliche Grenzen [7]

Auf der linken Seite der Abbildung ist die räumliche Grenze, die durch *IfcRelSpaceBoundary1stLevel* definiert werden. Sie hängen von der Oberfläche der Bauelemente ab, und die Daten für Materialien eines Bauelementes werden ignoriert. Sie können nicht direkt für die thermische Analyse verwendet werden. Räumliche Grenzen des ersten Niveaus können jedoch als Eingabe für thermische Analyse-Software importiert werden und dann werden die für Energieanalyse benötigte Informationen manuell hinzugefügt. Das Attribut *ParentBoundary* mit inversen *InnerBoundaries* wird bereitgestellt, um die Raumbegrenzungen von Türen, Fenstern und Öffnungen mit der übergeordneten Begrenzung, z. B. einer Wand oder einer Platte, zu verknüpfen und die übergeordnete Begrenzung wird nicht bei der inneren Begrenzung geschnitten sondern überlappen sie sich miteinander.

Auf der rechten Seite ist ein Beispiel von *IfcRelSpaceBoundary2ndLevel*. Sie ist in der Lage, die räumliche Grenzen deutlicher zu beschreiben. Zuerst werden die Materialdaten einer physischen Grenze wie eine Wand berücksichtigt. Wenn beispielsweise eine Wand zwei Räume trennt, wird sie als zwei Begrenzungen betrachtet. Das ist wichtig, wenn Wärmedurchgang zwischen verschiedenen Räumen berechnet wird. Darüber hinaus werden die Raumgrenzen als *PHYSICAL* deklariert, wenn sie mit einem greifbaren Gebäudeelement verknüpft sind, oder als *VIRTUAL*, wenn es kein Gebäudeelement gibt, sondern nur eine funktionale Trennung von zwei Räumen (z. B. Büro und Küche). Wie die obere Abbildung darstellt, gibt es für die Raumgrenzen im zweiten Niveau, drei Typen. Die rote Linie, bedeutet die Grenzen zwischen zwei Räumen. Sie wird als Typ A benannt. Und Typ B (Blau Linie) wird verwendet, wenn ein Bauelement hinter der räumlichen Begrenzung ist. Für Energiesimulation ist *IfcRelSpaceBounda-*

ry2endLevel erforderlich. Deswegen wenn man eine IFC Datei als Eingabe für eine Simulationssoftware importiert, muss er die Auflage davon beachten. Diese Entitäten werden nur bei *IFC 2x4* angeboten.

Das Attributset, das sich auf die *IfcSpace* bezieht, wird von *IfcPropertySet* definiert. Und die beiden Entitäten werden durch *IfcRelDefinesByProperties* verknüpft. Die für Energiesimulation notwendigen Daten werden „*Pset_SpaceThermalRequirements*“ und „*Pset_SpaceThermalDesign*“ definiert.

3.12 Zone und Technische Gebäude Ausrüstung

Im Prozess der Energiesimulation werden die Räume zusammengebunden, die gleiche thermische Bedürfnis haben. Diese Räume werden als eine thermische Zone hingestellt, um die Effizienz der Simulationssoftware zu erhöhen. Im IFC Standard wird sie durch die Entität *IfcZone* dargestellt. Die Abbildung 40 zeigt das EXPRESS-G Diagramm über *IfcZone*.

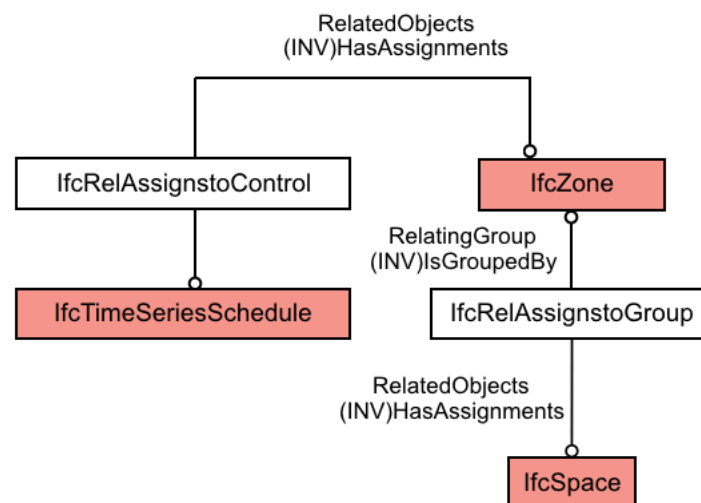


Abbildung 40 : Beziehung zwischen *IfcSpace* und *IfcZone*

Wie in der Abbildung angezeigt, ist die Entität in der Lage, *IfcSpace* und *IfcZone* zusammenzubinden. Der Zeitplan eine thermische Zone wird durch *IfcTimeSeriesSchedule* beschrieben. Sie hat die Fähigkeit, die Heizung, Lüftungsanlage und Beleuchtung einer Zone zu kontrollieren. Die Verbindung von *IfcTimeSeriesSchedule* und *IfcZone* ist durch *IfcRelAssignstoControl* erforderlich.

Alle technischen Ausrüstungen eines Gebäudes wie Pumpe und Heizkessel werden Im IFC Standard durch *IfcDistributionElement* dargestellt. Die zwei Unterklassen davon sind *IfcDistributionFlowElement* und *IfcDistributionControlElement*. Die Erste wird für die Beschreibung wie Kanäle, oder Rohre verwendet, und die zweite definiert die Elementen von Überwachungs- oder Steuerungssystemen wie Sensoren oder Kontrollier. Für weitere Eigenschaften der technischen Ausrüstung sind die Fachleute in der Lage, durch *IfcPropertySetDefinition* aufzubauen. Die vorliegende Abbildung 41 zeigt den Aufbau eines technischen Systems in einem Gebäude.

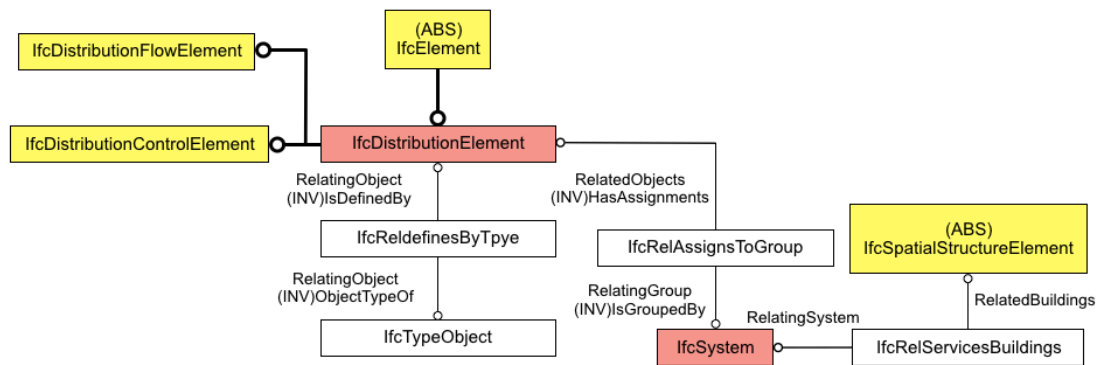


Abbildung 41 : Zusammenbindung von *IfcElement* und *IfcSystem*

Aus der Abbildung ist es einfach zu sehen, dass die Entität *IfcRelServicesBuildings* in der Lage sind, die Verknüpfung zwischen *IfcSystem* und *IfcSpatialStructureElement* wie *IfcSpace* oder *IfcZone* darzustellen. Und die Verbindung von *IfcDistributionElement* und *IfcSystem* ist durch *IfcRelAssignsToGroup* erforderlich.

4. Methodik zur Überprüfung des IFC Models

In diesem Kapitel wird eine Methodik erstellt, um das von einer BIM-Software exportierte Energiesimulationsmodell zu überprüfen. Im Weiteren werden zwei Fallstudien als Beispiele untersucht. Die Programmierungssprache Python und ein Paket „IfcOpenShell“ werden verwendet. Die Visualisierung wird die Software „Solibri Model Viewer“ benutzt. In der vorliegenden Abbildung 42 wird der grundlegende Ablauf dargestellt.

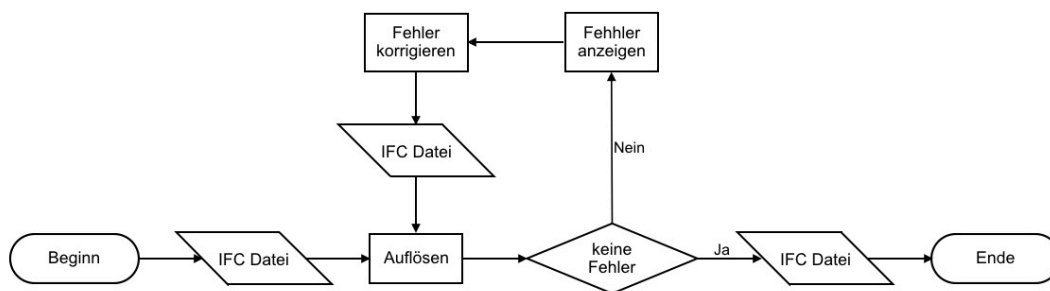


Abbildung 42 : Ablauf der Überprüfung

Wie in der Abbildung dargestellt, wird zuerst eine IFC Datei bei „IfcOpenShell“ übernommen und aufgelöst. Für jede Entität hat der Rechner die Fähigkeit, eine Instanz im Speicher aufzubauen. Mithilfe der bedingten Anweisung wird jede Entität überprüft. Wenn sie die Bedingungen erfüllt, kann das Programm weiter laufen und am Ende werden alle Daten wieder zusammengebunden und als eine IFC Datei wieder exportiert. Aber wenn eine Entität die Bedingungen für Energiesimulationen nicht erfüllt, wird sie markiert und zum Schluss als ein Bericht exportiert. Damit können die Fachleute Fehler korrigieren.

Aufgrund des Typs von Eingabedaten für *IES <VE>* muss man die Überprüfung in zwei Vertiefungen aufteilen. Zuerst werden die Daten für Bauelementen wie Materialien, thermische Eigenschaften und geometrische Parameter überprüft. Und dann muss man die räumlichen Daten und die Informationen für technische Gebäudeausrüstungen wie Heizung und Lüftung überprüfen. Die Analyse werden im weiteren Abschnitten dargestellt.

4.1 Daten für Bauelementen

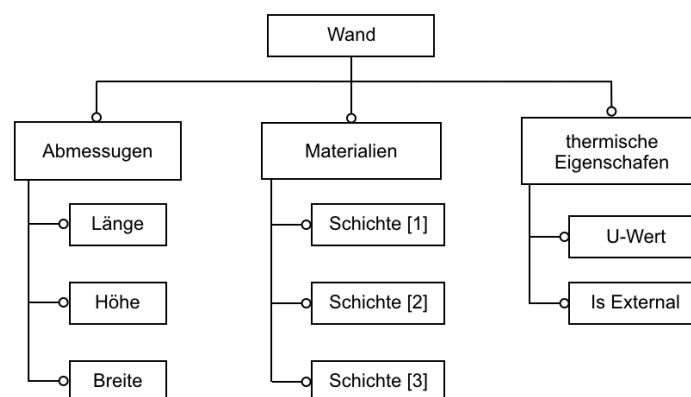


Abbildung 43 : Bestandteile für die Überprüfung

Aufgrund der ähnlichen Definitionsmethode für alle Bauelemente im IFC Standard wird im vorliegenden Bericht, die Wand als Beispiel benutzt. Die Abbildung 43 zeigt die wichtigen Informationen in einer IFC Datei, die man kontrollieren muss. Für die Definition einer Wand gibt es drei Bestandteile. Der erste Fall sind die Abmessungen wie Länge, Breite und Höhe. Der zweite Fall sind das Material der Wand. Im Prozess des Datenaustausches besteht die Möglichkeit, dass diese Daten verloren gehen. Der dritte Fall sind die thermischen Parameter einer Wand. Sie sind die direkten Einflussfaktoren für die Energiesimulation. Abbildung 44 zeigt den Arbeitsablauf der Überprüfung.

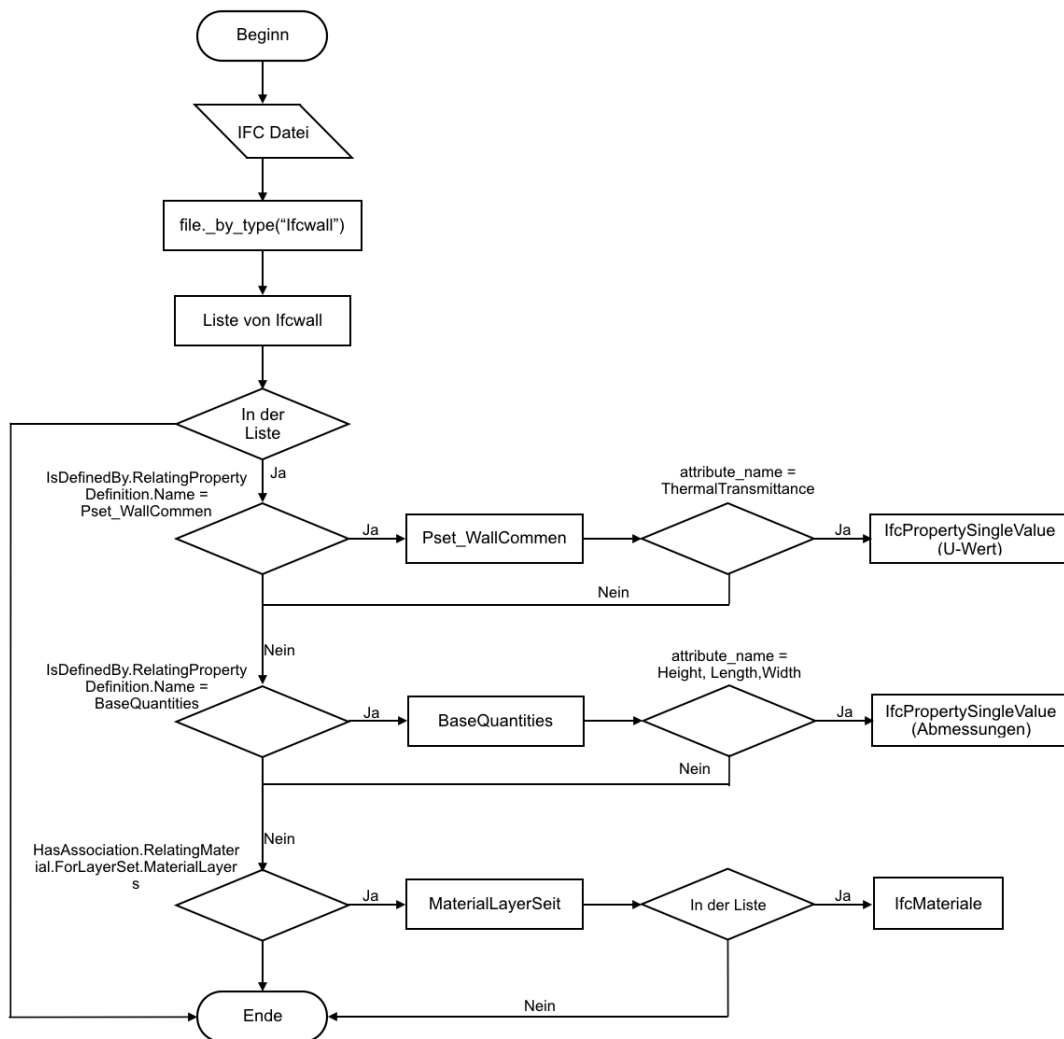


Abbildung 44 : Arbeitsablauf der Überprüfung für die Wände

Am Anfang wird eine IFC Datei geöffnet und für jede Entität gibt es eine entsprechende Instanz im Arbeitsspeicher. Durch eine Funktion werden alle *IfcWallStandardCase* und *IfcWall* ausgewählt und in einer Liste gespeichert. Aufgrund der in Kapitel 3 beschriebenen Datenstruktur von *IfcWall* hat man die Möglichkeit, die relevanten Attribute zu untersuchen. Mithilfe des Namens von jeder Entität werden die Eigenschaften von *IfcWall* festgelegt. Sie sind auch als eine Liste im Rechner gespeichert. Mithilfe einer bedingten Anweisung und Schleifen werden die benötigten Daten einer Wand gefiltert. Dann wird ein Vergleich zwischen BIM Model in *Revit* und IFC Daten durchgeführt. Die Fachleute sind auch in der Lage, die Daten selbst zu korrigieren oder einzufügen. Mit

gleicher Methode wird eine Traversierung in der Liste von *IfcWall* durchgeführt. Nachdem das letzte Element in der Liste überprüft wurde, wird die Bearbeitung beebdet.

4.2 Räumliche Daten

Zur Nutzung von IFC Dateien als Eingabedaten für Simulationssoftware muss man die Entitäten *IfcSpace* und *IfcRelSpaceBoundary* überprüfen. Ohne die zwei Entitäten kann die Simulationssoftware nicht die IFC Datei erkennen. Deswegen ist es notwendig, die Existenz zu überprüfen, bevor die Datei in die Software importiert wird. Darüberhinaus muss man der räumlichen Begrenzungen aller Räume überprüfen. Aufgrund der Datenstruktur der *IfcRelSpaceBoundary* muss die IFC Datei eine Voraussetzung erfüllen. Die Entität *IfcRelSpaceBoundary* muss in der zweiten Stufe sein. Die Abbildung 45 zeigt das Diagramm des Ablaufs.

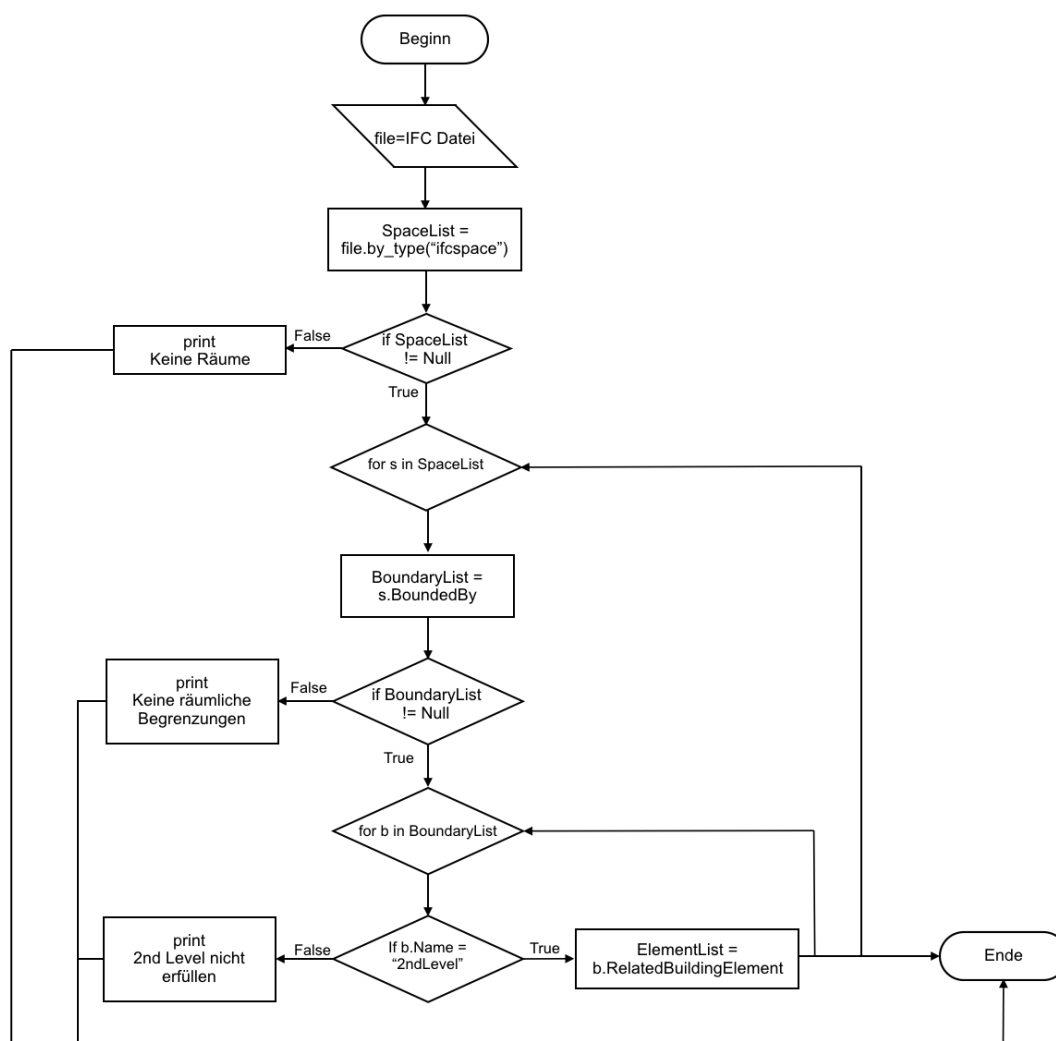


Abbildung 45 : Ablauf der Überprüfung für räumliche Daten

Aus der Abbildung ist es einfach abzuleiten, dass die Entitäten *IfcSpace* zuerst bei einer Funktion ausgewählt werden. Sie werden in einer Liste gespeichert. Wenn die Liste leer ist, wird eine Aussage automatisch ausgedrückt. Dann für jedes Element in der Liste wird eine Beurteilung durchgeführt. Wenn sie keine räumliche Begrenzungen hat, kann der Algorithmus eine Bemerkung erstellen. Wenn das Ergebnis der Beurteilung

positiv ist, wird den Prozess weiter abgearbeitet. Im nächsten Schritt wird die Stufe der räumlichen Begrenzung beurteilt. Wenn eine *IfcRelSpaceBoundary* alle Voraussetzungen erfüllt, werden die Begrenzungen in einer Liste gespeichert. Mithilfe ihres Attributes „RelatedBuildingElement“ wird das verknüpfende Bauelement vermerkt. Wenn das Traversieren fertig ist, werden alle Bemerkungen exportiert und damit haben die Fachleute manuell oder semi-automatisch die IFC Datei zu korrigieren.

4.3 Datei für energetische Analyse

Wie im Kapitel 3 dargestellt, ist die Entität *IfcSpace* ein Element, mit dem Bauelementen und Gebäudeausrüstungen verknüpft sind. Ein Raum wird von Wänden gebildet und durch technische Anlagensysteme versorgt. Deswegen ist es notwendig, diese Informationen im IFC zu ermitteln und überprüfen. Die energetischen Daten eines Raumes werden als Attributen in IFC Dateien gespeichert. Um sie zu ermitteln, muss man zuerst alle *IfcSpace* Entitäten aus der IFC Datei wie im Abschnitt 4.2 angezeigt lesen. Ermittlung der Teile des Anlagensystemes gibt es eine Funktion. Die Abbildung 46 zeigt den Ablauf davon.

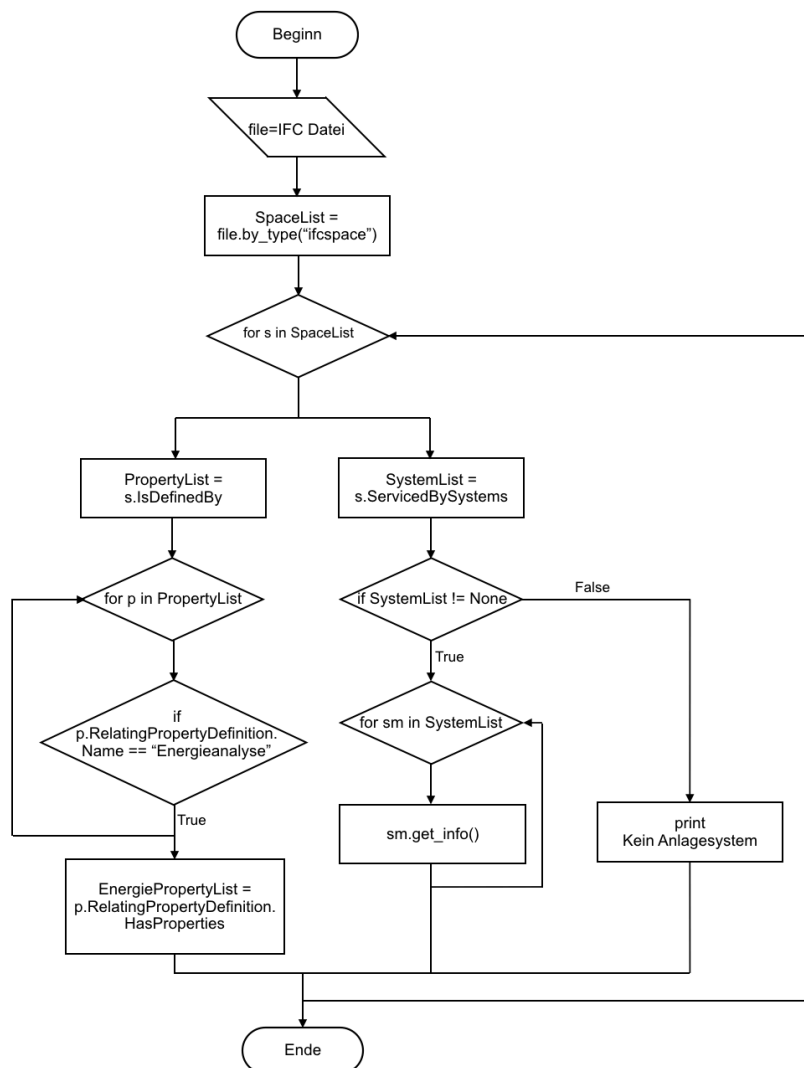


Abbildung 46 : Ablauf der Überprüfung für energetische Daten eines Raum

Wie in der oberen Abbildung gezeigt, werden die Daten der energetischen Analyse (z. B. Beleuchtungslast, Versorgungsleistung oder thermische Belastungen) überprüft. Der Prozess ist in der Lage, die Vollständigkeit der IFC Datei zu testen. Aber die Verfügbarkeit der Daten ist nicht eine notwendige Voraussetzung für Energiesimulationen. In der Simulationssoftware wie *IES <VE>* können die Fachleute selbst die thermischen Bedingungen einstellen.

4.4 Analyse der Fallstudie

Um die Methodik zu validieren und die Analyse zu vereinfachen, wird eine Fallstudie untersucht. Die vorliegende Abbildung 47 zeigt den grundlegenden Ablauf.

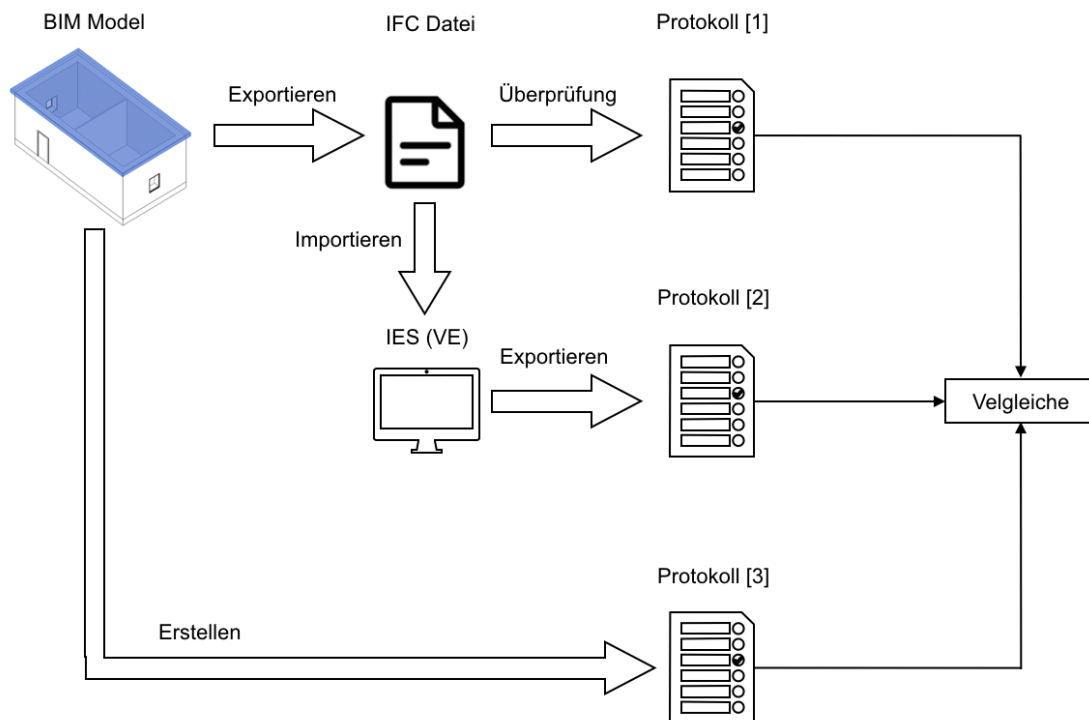
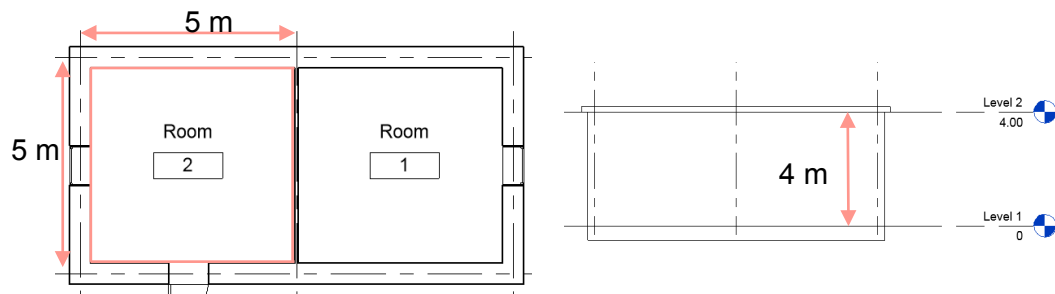


Abbildung 47 : Strategie der Validierung

In der Abbildung wird der Prozess deutlich dargestellt. Zuerst wird ein architektonisches Modell in der Software „*Revit*“ manuell aufgebaut (BIM Modell). Die relevanten Informationen werden überprüft, und das Ergebnis werden in einem Protokoll (Protokoll [3]) hinzugefügt. Dann wird das Modell als IFC Datei exportiert. Die Voraussetzung ist, dass alle notwendigen Einstellungen in *Revit* richtig sind. Danach wird die IFC Datei überprüft. Das Ergebnis wird protokolliert (Protokoll [1]). Parallel wird die IFC Datei in die Simulationssoftware importiert. Darüberhinaus wird noch ein Protokolle (Protokoll [2]) automatisch erstellt. Zum Schluss wird der Vergleich zwischen den drei verschiedenen Protokollen durchgeführt. Die Visualisierung der IFC Datei ist durch die Software „*Solibri Model Viewer*“ erforderlich.

Zuerst ist es notwendig, die Information der Fallstudie zu beschreiben. Als Fallstudie wurde ein Bürogebäude mit zwei Zimmern gewählt. Die allgemeinen geometrischen Daten sind in Abbildung 48 gezeigt.



Geometrische Daten in Revit (Zimmer 2)	
Eigenschaft	Wert
Länge	4.71 m
Breite	4.51 m
Höhe	4 m
Fläche	21.26 m ²
Volumen	85.06 m ³

Abbildung 48 : Geometrische Daten des Gebäudes

Aus der Abbildung ist einfach abzuleiten, dass die geometrische Berechnung eines Raumes basierend auf den Daten der Wand ist. Die Distanz zwischen Außenwand und Innenwand ist 5 Meter (Systemlinien). Die Berechnungen der Flächen und des Volumens basieren auf den Innenflächen der Wände.

Die vorliegende Tabelle 9 zeigt die geometrische Informationen des Bauelements.

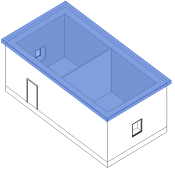
	Länge (cm)	Dicke (cm)	Höhe (cm)
Außenwand Süd	1000	49.2	400
Außenwand Nord	1000	49.2	400
Außenwand West	500	49.2	400
Außenwand Osten	500	49.2	400
Innenwand	500	8	400
Decke	1089	20.9	589
Boden	1000	51	500
Tür x1	213.4	nicht definiert	91.5
Fenster x2	91.5	nicht definiert	122

Tabelle 9 : Geometrische Daten des Bauelementes

Dann werden die Materialien angezeigt, die für den Aufbau des Gebäudes verwendet werden.

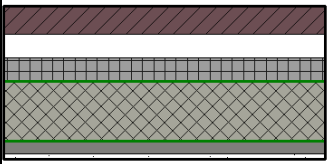
			
Außenwand (Basiswand - Brick on CMU)			
Schichten (Von außen nach innen)	Funktion	Materialien	Dicke (cm)
1	Nichttragende Schicht	Ziegelstein, allgemein	9
2	Luftschicht	Luft	7.6
3	Dämmung	starre Isolierung	7.6
4	Sperrschicht	Feuchteschutz	0.0
5	Tragende Schicht	Betonmauerwerk	19
6	Träger	Rahmen Leichtmetall	4.2
7	Nichttragende Schicht	Gipskartonplatte	1.6

Tabelle 10 : Materialiensichten von Außenwand


			
Innenwand (Partition)			
Schichten (Von außen nach innen)	Funktion	Materialien	Dicke (cm)
1	Nichttragende Schicht	Gipskartonplatte	1.3
2	Nichttragende Schicht	Gipskartonplatte	0.6
3	Tragende Schicht	Rahmen Leichtmetall	4.2
4	Nichttragende Schicht	Gipskartonplatte	0.6
5	Nichttragende Schicht	Gipskartonplatte	1.3

Tabelle 11 : Materialiensichten von Innenwand

			
Boden (Beton)			
Schichten (Von außen nach innen)	Funktion	Materialien	Dicke (cm)
1	Nichttragende Schicht	Vinylzusammensetzung	0.5
3	Nichttragende Schicht	Beton	10
4	Tragende Schicht	Metall-Deck	0.5
5	Tragende Schicht	Stahbeton	40

Tabelle 12 : Materialiensichten von Boden

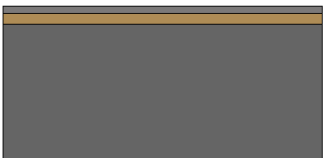
			
Decke (Asphaltschindeln)			
Schichten (Von außen nach innen)	Funktion	Materialien	Dicke (cm)
1	Nichttragende Schicht	Asphalt	1
3	Tragende Schicht	Sperrholzummantelung	1.5
4	Tragende Schicht	Holzbalken Sparrenschicht Batt Isolierung	18.4

Tabelle 13 : Materialiensichten von Decke

Darüberhinaus muss man die energetischen Parameter der Bauelementen beachten. In der vorliegende Tabelle 14 zeigt die thermischen Faktoren, die in *Revit* definiert werden, dargestellt. Sie werden als eine Kontrollgruppe im Vergleich verwendet.

Bauelemente	U-Wert $W/(m^2 \cdot K)$	Thermischer Widerstand (R) $(m^2 \cdot K)/W$	Thermischer wirksame Masse KJ/K	Absorptionsgrad
Außenwan	0.1802	9.4731	30.62	0.1
Innenwand	0.5752	1.7385	3.27	0.1
Decke	0.1016	9.8457	3.17	0.1
Boden	0.0607	16.4856	8.06	0.1
Tür	0.5504	1.8170	nicht definiert	nicht definiert
Fenste	0.5032	1.9873	nicht definiert	nicht definiert

Tabelle 14 : Thermische Faktoren der Bauelementen

Basierend auf den obengenannten Daten wird ein Vergleich durchgeführt. Zuerst werden die drei allgemeinen Erscheinungen in verschiedenen Softwares verglichen. Die untere Abbildung zeigt die Visualisierung in den verschiedenen Softwaresystem.

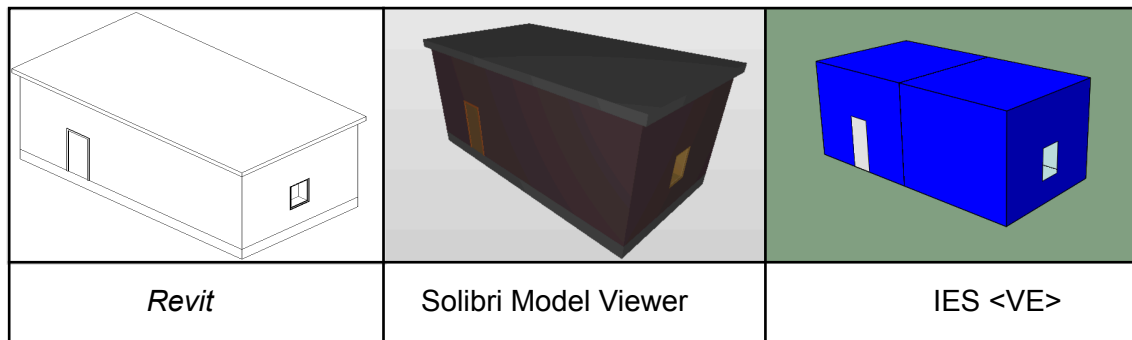


Abbildung 49 : Visualisierung in verschiedenen Softwaresystem

Aus der obengenannten Abbildung ist es einfach zu sehen, dass mit verschiedenen Software—Engines die Darstellungen der Visualisierung verschieden sind. In der erste Spalte ist das originale 3D Modell, das bei *Revit* angeboten wird. Die Formen der Bauelemente werden gut dargestellt aber es gibt keine Einzelheiten. Das IFC Modell befindet sich in der Mitte. Die Software „*Solibri Model Viewer*“ ist in der Lage, IFC Dateien zu lesen und ein 3D Modell aufzubauen. Offensichtlich wird es bei der Software gerendert. Die Farbe und Zeichnungen der Bauelemente werden deutlich angezeigt. Darüberhinaus werden die Tür und das Fenster gut dargestellt. In der letzten Spalte ist das bei *IES <VE>* angebotene Modell. Es ist das schlimmste Modell im Vergleich zu den Vorgängermodellen. Die Erscheinung aller Bauelemente wird bei der Software ignoriert. Sie werden durch ein paar Linien ersetzt. Die räumliche Struktur wird gut dargestellt. Die blauen Oberflächen sind räumliche Begrenzungen. Die Einzelheiten von Tür und Fenster werden nicht dargestellt.

Basierend auf der Analyse können folgende Schlussfolgerungen gezogen werden. Zuerst hat eine IFC Datei die Fähigkeit, die Erscheinung eines Gebäudes gut zu beschreiben. Die Software „*Solibri Model Viewer*“ hat eine gute Schnittstelle zum Lesen einer IFC Datei. Die Kompatibilität zwischen IFC Standard und *IES <VE>* ist nicht ausreichend. Die zweite Schlussfolgerung ist, dass die räumlichen Begrenzungen eine große Rollen bei der Modellierung spielen. Ohne räumliche Begrenzungen ist die Modellierung in *IES <VE>* nicht möglich.

Darüberhinaus lässt sich feststellen, dass man die Überprüfung für räumliche Begrenzungen zuerst machen muss. Zur Anwendung der IFC Datei in *IES <VE>* wird eine Bewertung für die Existenz von *IfcSpace* und *IfcRelSpaceBoundary* zuerst durchgeführt.

Zur Ermittlung der Austauschbarkeit der Materialdaten werden die Protokolle der Materialien verglichen. Basierend auf dem obengenannten Ablauf hat man die Fähigkeit, durch Programmierung die Materialdaten von der IFC Datei auszuwählen. Eine Wand im IFC Standard wird als *IfcWall* oder *IfcWallStandardCase* benannt. In den vorliegenden Abbildungen werden die Ergebnisse des Vergleiches zwischen IFC Datei und *IES <VE>* angezeigt.

IFC Datei

```
walls = ifc_file.by_type("ifcwall")
for wall in walls:
    if wall.ObjectType == "Basic Wall:Exterior - Brick on CMU:264":
        RAM1 = wall.HasAssociations
        for m1 in RAM1[0].RelatingMaterial.ForLayerSet.MaterialLayers:
            print m1.Material.Name, " ",m1.LayerThickness
        break
```

Brick, Common 0.09
Air 0.076
Rigid insulation 0.076
Damp-proofing 0.0
Concrete Masonry Units 0.19
Metal Furring 0.042
Gypsum Wall Board 0.016

IES (VE)

Description: 2013 External Wall ID: STD_WAL1 External Internal

Performance: EN-ISO

U-value: 0.2599 W/m²·K Thickness: 208.900 mm Thermal mass Cm: 21.9500 kJ/(m²·K)
Total R-value: 3.6778 m²·K/W Mass: 46.9780 kg/m² Very lightweight

Surfaces Functional Settings Regulations

Outside Emissivity: 0.900 Resistance (m²·K/W): 0.0400 ☒ Default Solar Absorptance: 0.700

Inside Emissivity: 0.900 Resistance (m²·K/W): 0.1300 ☒ Default Solar Absorptance: 0.550

Construction Layers (Outside To Inside) System Materials... Project Materials...

Material	Thickness mm	Conductivity W/(m·K)	Density kg/m³	Specific Heat Capacity J/(kg·K)	Resistance m²·K/W	Vapour Resistivity GN·s/(kg·m)	Category
[STD_SM1] Rainscreen	3.0	50.0000	7800.0	450.0	0.0001	-	Metals
Cavity	50.0	-	-	-	0.1300	-	-
[STD_EPS] Insulation	81.4	0.0250	20.0	1030.0	3.2560	-	Insulating Materials
[STD_USP] Cement bonded particle board	12.0	0.2300	1100.0	1000.0	0.0522	0.000	Boards, Sheets & Decking
Cavity	50.0	-	-	-	0.1800	-	-
[STD_US1] Plasterboard	12.5	0.2100	700.0	1000.0	0.0595	0.000	Plaster

Abbildung 50 : Darstellung für Materialdaten der Außenwand

IFC Datei

```
walls = ifc_file.by_type("ifcwall")
for wall in walls:
    if wall.ObjectType == "Basic Wall:Interior - 79mm Partition (1-hr):258":
        RAM2 = wall.HasAssociations
        for m2 in RAM2[0].RelatingMaterial.ForLayerSet.MaterialLayers:
            print m2.Material.Name, " ",m2.LayerThickness
        break
```

Gypsum Wall Board 0.013
Gypsum Wall Board 0.006
Metal Stud Layer 0.042
Gypsum Wall Board 0.006
Gypsum Wall Board 0.013

IES <VE>

Description: 2013 Internal Partition ID: STD_PART External Internal

Performance: EN-ISO

U-value: 1.7888 W/m²·K Thickness: 75.000 mm Thermal mass Cm: 8.7500 kJ/(m²·K)
Total R-value: 0.2990 m²·K/W Mass: 17.5000 kg/m² Very lightweight

Surfaces Regulations

Outside Emissivity: 0.900 Resistance (m²·K/W): 0.1300 ☒ Default Solar Absorptance: 0.550

Inside Emissivity: 0.900 Resistance (m²·K/W): 0.1300 ☒ Default Solar Absorptance: 0.550

Construction Layers (Outside To Inside) System Materials... Project Materials...

Material	Thickness mm	Conductivity W/(m·K)	Density kg/m³	Specific Heat Capacity J/(kg·K)	Resistance m²·K/W	Vapour Resistivity GN·s/(kg·m)	Category
[STD_US2] Plasterboard	12.5	0.2100	700.0	1000.0	0.0595	0.000	Plaster
Cavity	50.0	-	-	-	0.1800	-	-
[STD_US3] Plasterboard	12.5	0.2100	700.0	1000.0	0.0595	0.000	Plaster

Abbildung 51 : Darstellung für Materialdaten der Innenwand

In der Abbildung 50 und 51 ist deutlich zu sehen, dass die IFC Datei in der Lage ist, die Materialdaten der Wände gut darzustellen. Die Arte und Dicke für jede Schicht wird gut beschrieben bei IFC. Aber im Vergleich zwischen Revit und IES <VE> ist die Funktion der Schichten verloren (z. B tragende Schicht oder nichttragende Schicht).

Die Situation in IES <VE> ist ganz unterschiedlich. Alle Materialdaten werden nicht übernommen bei der Software. Die Wand wird nach der Voreinstellung der Software aufgebaut. Aufgrund der Datenbank von IES <VE> wird die Schichten einer Wand standardisiert.

Darüber hinaus lässt sich feststellen, dass die schlechte Kompatibilität mit *buildingSMART Data Dictionary* (bSDD) das Problem verursachen kann. Die Beschreibungsverfahren eines Materials in verschiedenen Datenbanken sind unterschiedlich. Die Organisation „BuildingSMART“ hat eine allgemeine Datenbank „bSDD“ zur Verbesserung des Datenaustausches. Die BIM Software *Revit* hat eine gute Schnittstellen für bSDD. Deshalb werden die Daten gut dargestellt, wenn man eine IFC Datei exportiert. Aber die Simulationssoftware IES <VE> ist nicht in der Lage, diese Datei aufzulösen. Deswegen werden die Wände nach der eigenen Datenbank aufgebaut und die IFC Daten ignoriert.

Die Abbildung 52 zeigt das Ergebnis des Vergleichs für den Boden. Er wird in der IFC als *IfcSlab* gespeichert.

IFC Datei	<pre> bodens = ifc_file.by_type("ifcslab") for m in bodens[0].HasAssociations[0].RelatingMaterial.ForLayerSet.MaterialLayers: print m.Material.Name, " ", m.LayerThickness </pre> <p>Vinyl Composition Tile 0.005 Concrete, Lightweight 0.1 Metal Deck 0.005 Structure, Steel Bar Joist Layer 0.4</p>																																								
IES <VE>	<p>Description: 2013 Exposed Floor ID: STD_FLO1 External Internal</p> <p>Performance: EN-ISO</p> <p>U-value: 0.2200 W/m²·K Thickness: 268.200 mm Thermal mass Cm: 85.0000 kJ/(m²·K) Total R-value: 4.3353 m²K/W Mass: 308.7400 kg/m² Very lightweight</p> <p>Surfaces Functional Settings Regulations</p> <p>Outside Emissivity: 0.900 Resistance (m²K/W): 0.0400 <input checked="" type="checkbox"/> Default Solar Absorptance: 0.550 Inside Emissivity: 0.900 Resistance (m²K/W): 0.1700 <input checked="" type="checkbox"/> Default Solar Absorptance: 0.550</p> <p>Construction Layers (Outside To Inside) System Materials... Project Materials...</p> <table border="1"> <thead> <tr> <th>Material</th> <th>Thickness mm</th> <th>Conductivity W/(m·K)</th> <th>Density kg/m³</th> <th>Specific Heat Capacity J/(kg·K)</th> <th>Resistance m²K/W</th> <th>Vapour Resistivity Gv·s/(kg·m)</th> <th>Category</th> </tr> </thead> <tbody> <tr> <td>[STD_PH1] Insulation</td> <td>98.2</td> <td>0.0250</td> <td>700.0</td> <td>1000.0</td> <td>3.9280</td> <td>-</td> <td>Insulating Materials</td> </tr> <tr> <td>[STD_CC2] Reinforced Concrete</td> <td>100.0</td> <td>2.3000</td> <td>2300.0</td> <td>1000.0</td> <td>0.0435</td> <td>-</td> <td>Concretes</td> </tr> <tr> <td>Cavity</td> <td>50.0</td> <td>-</td> <td>-</td> <td>-</td> <td>0.2100</td> <td>-</td> <td>-</td> </tr> <tr> <td>[STD_FBA] Chipboard Flooring</td> <td>20.0</td> <td>0.1300</td> <td>500.0</td> <td>1600.0</td> <td>0.1538</td> <td>-</td> <td>Boards, Sheets & Decking</td> </tr> </tbody> </table>	Material	Thickness mm	Conductivity W/(m·K)	Density kg/m³	Specific Heat Capacity J/(kg·K)	Resistance m²K/W	Vapour Resistivity Gv·s/(kg·m)	Category	[STD_PH1] Insulation	98.2	0.0250	700.0	1000.0	3.9280	-	Insulating Materials	[STD_CC2] Reinforced Concrete	100.0	2.3000	2300.0	1000.0	0.0435	-	Concretes	Cavity	50.0	-	-	-	0.2100	-	-	[STD_FBA] Chipboard Flooring	20.0	0.1300	500.0	1600.0	0.1538	-	Boards, Sheets & Decking
Material	Thickness mm	Conductivity W/(m·K)	Density kg/m³	Specific Heat Capacity J/(kg·K)	Resistance m²K/W	Vapour Resistivity Gv·s/(kg·m)	Category																																		
[STD_PH1] Insulation	98.2	0.0250	700.0	1000.0	3.9280	-	Insulating Materials																																		
[STD_CC2] Reinforced Concrete	100.0	2.3000	2300.0	1000.0	0.0435	-	Concretes																																		
Cavity	50.0	-	-	-	0.2100	-	-																																		
[STD_FBA] Chipboard Flooring	20.0	0.1300	500.0	1600.0	0.1538	-	Boards, Sheets & Decking																																		

Abbildung 52 : Darstellung für Materialdaten der Boden

Aus der oberen Abbildung lässt sich feststellen, dass die Materialdaten für den Boden bei IES <VE> nicht dargestellt werden. Aber außerhalb der Funktion ist, die in der IFC Datei angebotene Darstellung ganz gut.

Dann werden die Materialdaten der Decke (*IfcRoof*) analysiert. Das Ergebnis ist in der Abbildung 53 dargestellt.

IFC
Datei

```

decke = decke = ifc_file.by_type("ifcroof")
try :
    for d in decke[0].HasAssociations[0].RelatingMaterial.ForLayerSet.MaterialLayers:
        print d.Material.Name, " ", d.LayerThickness
except :
    print "keine materiale Daten Verbunden"
materials = ifc_file.by_type("ifcmaterial")
OriginaleMaterial = ["Asphalt Shingle", "Plywood, Sheating", "Structure, Wood Joist/Rafter Layer,Batt Insulation"]
for m in materials:
    if m in OriginaleMaterial:
        print m
print "keine passende Materialien"
        
```

keine materiale Daten Verbunden
keine passende Materialien

IES
<VE>

Description: 2013 Roof ID: STD_ROOF External Internal

Performance: EN-ISO
 U-value: 0.1800 W/m²·K Thickness: 317.000 mm Thermal mass Cm: 98.7500 kJ/(m²·K)
 Total R-value: 5.4163 m²K/W Mass: 255.0360 kg/m² Lightweight

Surfaces Regulations
 Outside: Emissivity: 0.900 Resistance (m²K/W): 0.0400 ☒ Default Solar Absorptance: 0.700
 Inside: Emissivity: 0.900 Resistance (m²K/W): 0.1000 ☒ Default Solar Absorptance: 0.550

Construction Layers (Outside To Inside)
 System Materials... Project Materials...

Material	Thickness mm	Conductivity W/(m·K)	Density kg/m³	Specific Heat Capacity J/(kg·K)	Resistance m²K/W	Vapour Resistivity GN·s/(kg·m)	Category
[STD_PHF] Insulation	154.4	0.0300	40.0	1450.0	5.1467	-	Insulating Materials
[STD_MEM] Membrane	0.1	1.0000	1100.0	1000.0	0.0001	-	Asphalts & Other Roofing
[STD_CC1] Concrete Deck	100.0	2.0000	2400.0	1000.0	0.0500	-	Concretes
Cavity	50.0	-	-	-	0.1600	-	-
[STD_USS] Plasterboard	12.5	0.2100	700.0	1000.0	0.0595	0.000	Plaster

Abbildung 53 : Darstellung für Materialdaten der Decke

Aus der Abbildung 53 ist klar zu sehen, dass die Beschreibungsverfahren für die Decke nicht ausreichend sind. Nicht nur die IFC Datei sondern auch IES <VE> sind nicht in der Lage, die Materialdaten der Decke zu beschreiben. Aufgrund des in Kapitel 3 erzählten Definitionsverfahrens lässt sich durch weitere Untersuchung feststellen, dass in der IFC Dateien bezügliche Entitäten wie *IfcRelAssociatesMaterial*, *IfcMaterialLayerSet* und *IfcMaterial* verloren werden gehen.

Zur Validierung der thermischen Parameter eines Bauelements muss man den Attributsätze beachten. In der Fallstudie haben alle Außenwände den gleichen Typ. Deswegen muss man nicht alle Wände untersuchen. Aufgrund der im Kapitel 3 dargestellten Datenstruktur ist es einfach zu ermitteln, dass man die „Pset_Wallcommon“, „Pset_Slab-Common“ und „Pset_RoofCommon“ in der IFC Datei beachten muss. Wegen des Kompatibilitätsproblems werden die Daten in IES <VE> nicht angezeigt. Das Ergebnis der IFC Datei wird in der Abbildung 53 ermittelt.

Außenwand		
<pre>walls = ifc_file.by_type("ifcwall") print "Attribut", " ", "Wert", " ", "Einheit" print "-----" for wall in walls: if wall.ObjectType == "Basic Wall:Exterior - Brick on CMU:264": for d in wall.IsDefinedBy: if d.is_a() == "IfcRelDefinesByProperties": if d.RelatingPropertyDefinition.Name == "Pset_WallCommon": for p in d.RelatingPropertyDefinition.HasProperties: print p.Name, " ", p.NominalValue.wrappedValue, " ", p.Unit break</pre>		
Attribut	Wert	Einheit

Reference	Exterior - Brick on CMU	None
LoadBearing	False	None
ExtendToStructure	False	None
IsExternal	True	None
ThermalTransmittance	0.18018674475	None

Abbildung 54 : Thermische Parameter der Außenwand

Aus der Abbildung lässt sich feststellen, dass der Wärmedurchgangskoeffizient (U-Wert) der Außenwand richtig angezeigt wird. In der Tabelle 14 werden vier in *Revit* definierte thermische Faktoren angezeigt. Aber in der IFC Datei wird nur noch die Spalte „U-Wert“ angezeigt. Wenn diese Parameter benötigt werden, sind die Fachleute in der Lage, nach der Datenstruktur der *IfcPropertySet* diese selbst zu definieren. Außerdem wird die Einheit davon nicht angezeigt. Deswegen wird sie in der IFC Datei als „None“ ausgedrückt. Durch Programmierung kann man auch direkt in dieser Textdatei die Einheit hinzufügen.

Abbildung 55 zeigt das Ergebnis des Vergleiches für andere Bauelemente. Daraus lässt sich feststellen, dass Innenwand und Boden das gleiche Problem haben. Der U-Wert besteht noch, aber die Einheit ist nicht ausgetauscht. Darüber hinaus wird der Typ des Bauelementes als Referenz aufgerufen. Für die Decke ist die Situation schlechter. Alle thermischen Parameter und die Einheiten davon werden nicht übertragen.

Innenwand	Attribut	Wert	Einheit

	IsExternal	False	None
	LoadBearing	False	None
	ExtendToStructure	False	None
Boden	Reference	Interior - 79mm Partition (1-hr)	None
	ThermalTransmittance	0.575221238938	None

	IsExternal	False	None
	LoadBearing	False	None
Decke	Reference	Steel Bar Joist - VCT on LW Concrete	None
	ThermalTransmittance	0.0606589538925	None
	PitchAngle	0.0	None

	IsExternal	True	None
	Reference	Generic Roof - 300mm	None
	TotalArea	64.1421	None
	ProjectedArea	64.1421	None

	IsExternal	True	None

Abbildung 55 : Eigenschaften der Bauelementen in der IFC Datei

Von der obengenannten Abbildung 53 und 54 wird die Schlussfolgerung abgeleitet. Wenn man aus *Revit* eine IFC Datei exportiert, werden die thermischen Informationen der Entitäten wie *IfcWall* und *IfcSlab* detailliert dargestellt aber der Ausdruck von *IfcRoof* ist nicht ausreichend. Zur Lösung des Problems gibt es zwei Möglichkeiten. Die Fachleute sind in der Lage, durch Programmierung diese Daten selbst aufzubauen. Die zweite Variante ist, die Änderungen der Voreinstellung in *Revit*. Man kann die Einstellung für IFC-Exportieren ändern, um die Decke nicht als *IfcRoof* sondern *IfcSlab* zu exportieren.

Dann wird die Austauschbarkeit der geometrischen Daten überprüft.

<pre>spaces = ifc_file.by_type("ifcspace") for space in spaces: if space.Name == "2": #print space h = space.Representation.Representations[0].Items[0].Depth l = space.Representation.Representations[0].Items[0].SweptArea.XDim b = space.Representation.Representations[0].Items[0].SweptArea.YDim s = l*b v = l*b*h print "Eigenschaft", " ", "Wert", " (Zimmer 2)" print "-----" print "Länge", " ", l, " ", "m" print "Breite", " ", b, " ", "m" print "Höhe", " ", h, " ", "m" print "Fläche", " ", s, " ", "m^2" print "Volumen", " ", v, " ", "m^3"</pre>	
<pre>Eigenschaft Wert (Zimmer 2) ----- Länge 4.715 m Breite 4.51 m Höhe 4.0 m Fläche 21.26465 m^2 Volumen 85.0586 m^3</pre>	

Abbildung 56 : Geometrische Daten in der IFC Datei

Die Abbildung 56 wird in zwei Teile getrennt. Der obere Teil ist das Programmierungsverfahren. Dadurch haben die Fachleute die Möglichkeit, Daten darzustellen oder zu verändern. Der untere Teil ist das Ergebnis der Bewertung. Es ist identisch wie die Daten in *Revit*. Darüber hinaus werden in der Abbildung 56 die Daten in *IES <VE>* angezeigt.

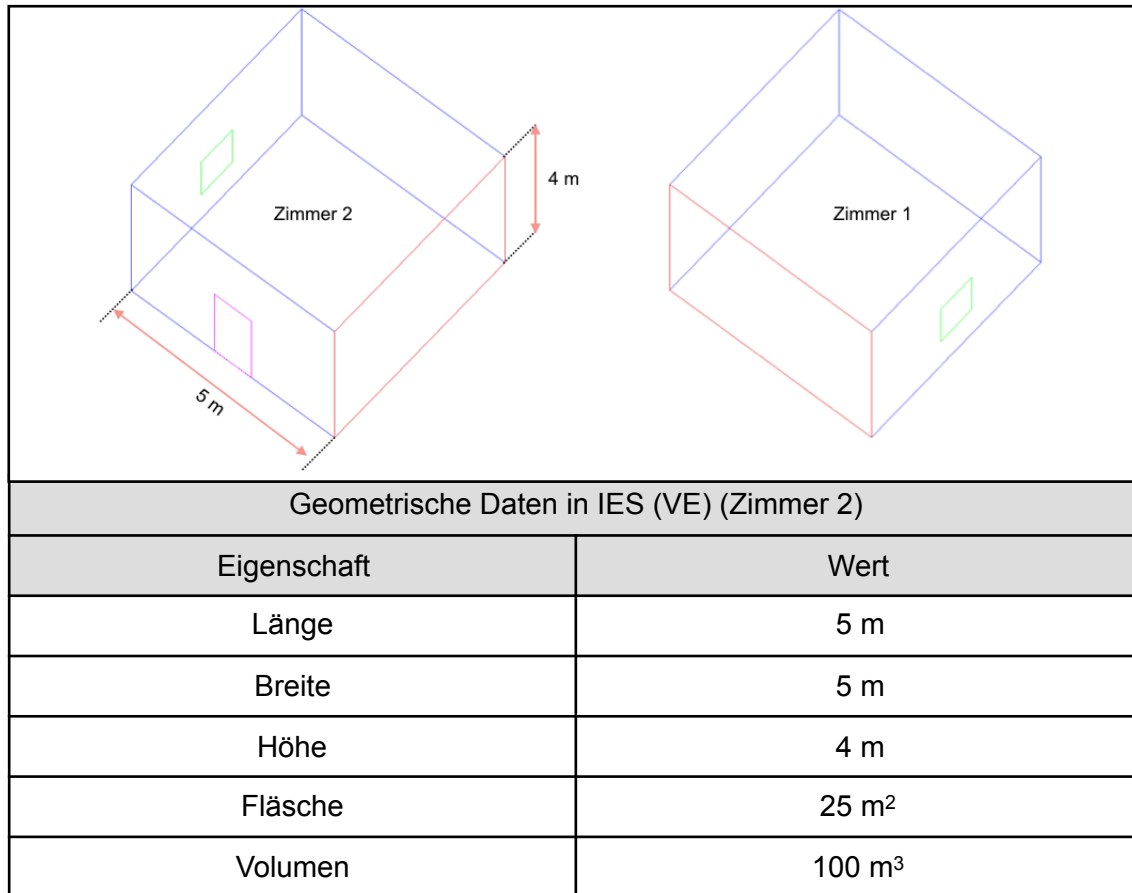


Abbildung 57 : Geometrische Daten in *IES <VE>*

Wegen der schlechten Kompatibilität der Software werden nur die räumlichen Objekte des Modells aufgenommen. Und die Dicke aller Bauelemente wird bei *IES <VE>* ignoriert. Zur Berechnung der Flächen und Volumen werden die Mittellinien der Wände als Grundlage verwendet. Darum bekommt man für die Länge und Breite jeweils 5 Meter.

Bei der Energiesimulation wird durch die Differenz ein großes Problem verursacht. Die Genauigkeit einer Simulation wird reduziert.

Darüberhinaus hat eine IFC Datei die Fähigkeit, die in *Revit* definierten Energieanalyse-Daten zu beschreiben. In der vorliegenden Abbildung 57 wird die Darstellung der Energiekosten eines Raums in der IFC Datei angezeigt.

Es ist deutlich zu sehen, dass alle Parameter der Energiekosten als Eigenschaften eines Raums existieren. Sie bilden eine Menge, die als „Energieanalyse“ bezeichnet wird. Deswegen ist das Definitionsverfahren gleich wie bei *IfcPropertySet*. Aber wenn die Datei in *IES <VE>* importiert wird, werden alle bezüglichen Informationen ignoriert. Dies ist in diesem Fall sinnvoll, da es sich um Berechnungsergebnisse handelt.

```

print "Eigenschaft", "      ", "Wert", " ", "(Zimmer 2)"
print "-----"
for i in space1.IsDefinedBy:
    if i.RelatingPropertyDefinition.Name == "Energieanalyse":
        for p in i.RelatingPropertyDefinition.HasProperties:
            name = p.Name.decode("unicode_escape")
            print name , " " , p.NominalValue.wrappedValue

```



```

Eigenschaft      Wert  (Zimmer 2)
-----
Angegebene Beleuchtungslast      0.0
Angegebene Beleuchtungslast pro Fläche      10.7639104167
Angegebene Versorgungsleistung      0.0
Angegebene Versorgungsleistung pro Fläche      13.9930835417
Beleuchtungslast basieren auf      <Standard>
Beleuchtungslasteinheiten      Leistungsdichte
Fläche pro Person      28.5714285714
Heizlastwerte      <Standard>
Latenter Wärmegewinn pro Person      630.918149013
Luftraum - Beitrag zu Beleuchtung      0.2
Personenzahl      0.0
Tatsächliche Beleuchtungslast      0.0
Tatsächliche Beleuchtungslast pro Fläche      0.0
Tatsächliche Versorgungsleistung      0.0
Tatsächliche Versorgungsleistung pro Fläche      0.0
Versorgungslasteinheiten      Leistungsdichte
Versorgungsleistung basieren auf      <Standard>
Wahrnehmbarer Wärmegewinn pro Person      788.647686266
Wärmegewinn insgesamt pro Person      1419.56583528

```

Abbildung 58 : Energiekosten eines Raums in der IFC Datei

Zusammenfassend lässt sich feststellen, dass die Kompatibilität zwischen IFC Standard und *Revit* besser ist als *IES <VE>*. Die Daten über Materialdicke und Dicke eines Bauelements werden bei IFC gut beschrieben. Während des Datenaustausches besteht den U-Wert noch in der IFC Datei, aber Thermischer Widerstand, Thermischer wirksame Masse und Absorptionsgrad sind verloren gehen. Zur Darstellung der Energiekosten spielt die IFC Datei auch eine Rolle. In Bezug auf *IES <VE>* gibt es mehr Probleme. Die Auflösungsfunktion ist nicht ausreichend. Sie kann nur die geometrischen Daten eines Raums darstellen und die Daten für Bauelemente und die thermischen Eigenschaften eines Raumes werden ignoriert. Zur Anwendung der IFC Datei als Eingabeformat müssen die Fachleute in der Software noch fehlende Daten hinzufügen.

5. Visualisierung und Kollisionsanalyse

In diesem Kapitel wird zuerst die Methodik zur Visualisierung eines IFC Modells erstellt. Mithilfe der Methodik wird ein Prüfverfahren für die Kollision zwischen zwei Bauelementen dargestellt. Basierend auf diesen Grundlagen wird eine grafische Benutzeroberfläche (GUI) aufgebaut.

5.1 Visualisierung

Mithilfe der im Kapitel 4 erzählten Methodik werden die geometrischen Daten eines Bauelementes wie Länge, Breite und Höhe einfach abgeleitet. Die Informationen für Lokalisierung kann man auch durch Programmierung aus einer IFC Datei auswählen. Die Visualisierung eines Bauelementes kann man durch Programmierung mithilfe von „IfcOpenShell“ und „PythonOCC“ schaffen. In der Abbildung 59 wird der Arbeitsablauf angezeigt.

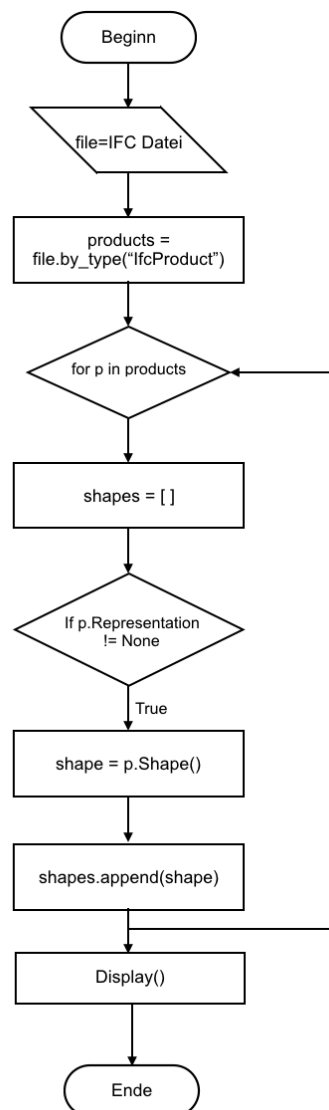


Abbildung 59 : Arbeitsablauf der Visualisierung

Wie in der Abbildung angezeigt, wird eine IFC Datei zuerst importiert und anschließend im Programm aufgelöst. Alle Bauelemente werden automatisch in einer Liste abgespeichert. Dann wird jedes Element gleichzeitig durch eine Funktion beurteilt. Wenn es geometrische Daten hat, werden diese Parameter ausgewählt und in einer anderen Liste gespeichert. Wenn der Durchlauf fertig ist, werden alle geometrischen Parameter automatisch gesammelt und am Bildschirm angezeigt.

Die Abbildung 60 zeigt die Visualisierung des im Kapitel 4 definierten Modells.

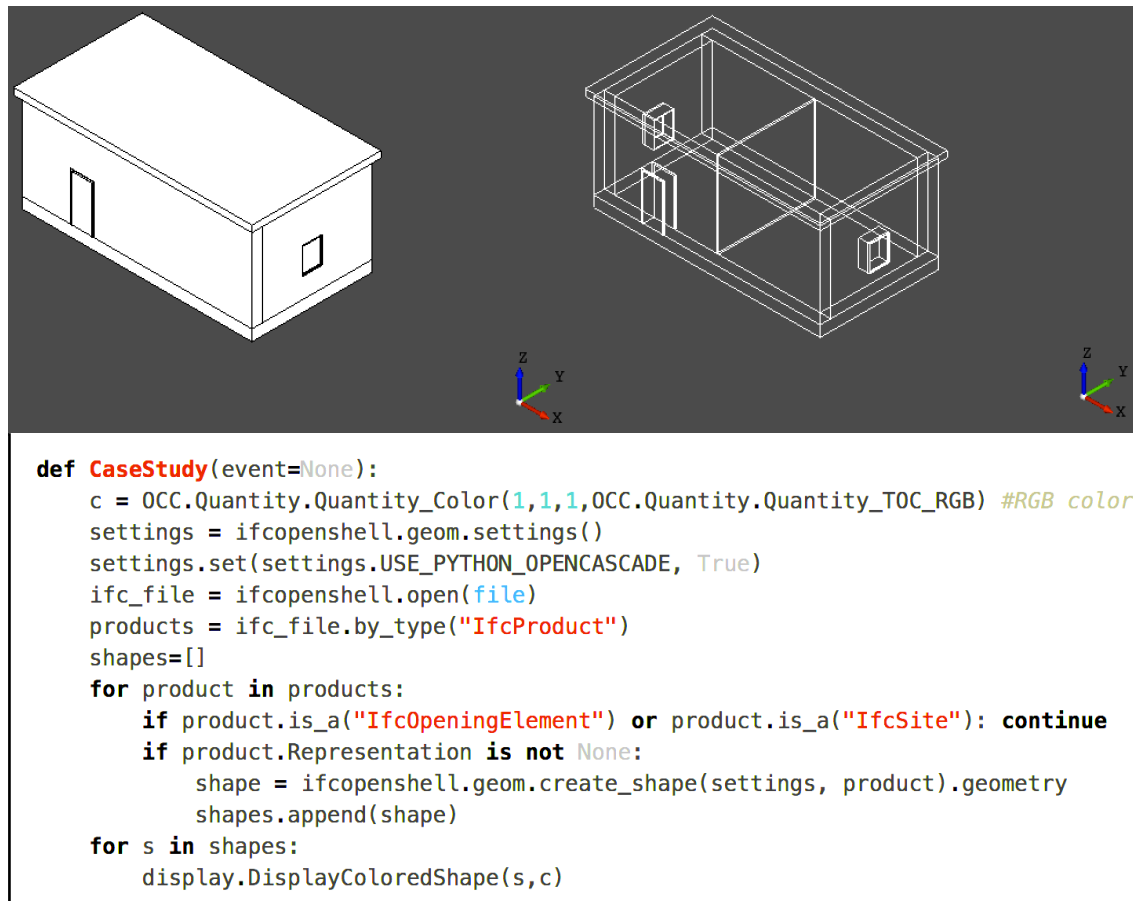


Abbildung 60 : 3D Modell der Fallstudie

Die Abbildung wird in zwei Teile aufgeteilt. Der obere Teil sind zwei Schemen des Verfahrens. Das Programm ist in der Lage, nicht nur die Oberfläche eines Modells sondern auch ein Drahtmodell darzustellen. Im Vergleich mit der Visualisierung, die durch „Solibri Model Viewer“ erstellt wird, hat das Programm Vor- und Nachteile. Obwohl es ein Modell detailliert beschreiben kann, ist das 3D Modell nicht realistisch. Der Algorithmus zur Bildsynthese ist nicht so gut wie der bei „Solibri Model Viewer“. Aber das Programm hat auch Vorteile. Zuerst ist es besser anwendbar. Die Programmiersprache „Python“ ist eine „Open Source“ Sprache. Sie ist nicht auf das Betriebssystem beschränkt. Deswegen ist das Programm in der Lage, unter allen Betriebssystemen wie Windows, Linux und MacOS zu funktionieren. Der zweite Fall ist, dass durch das Programm ein Drahtmodell erstellt werden kann. Damit werden die Einzelheiten eines Bauelementes dargestellt. Man kann durch Programmierung die IFC Datei verändern und ein 3D Modell schnell aufbauen. Der untere Teil der Abbildung 60 ist der Quelltext.

Darüberhinaus kann man durch das Programm jedes Bauelement mit verschiedenen Farben visualisieren. In der Abbildung 61 wird der Prozess der Modellierung angezeigt.

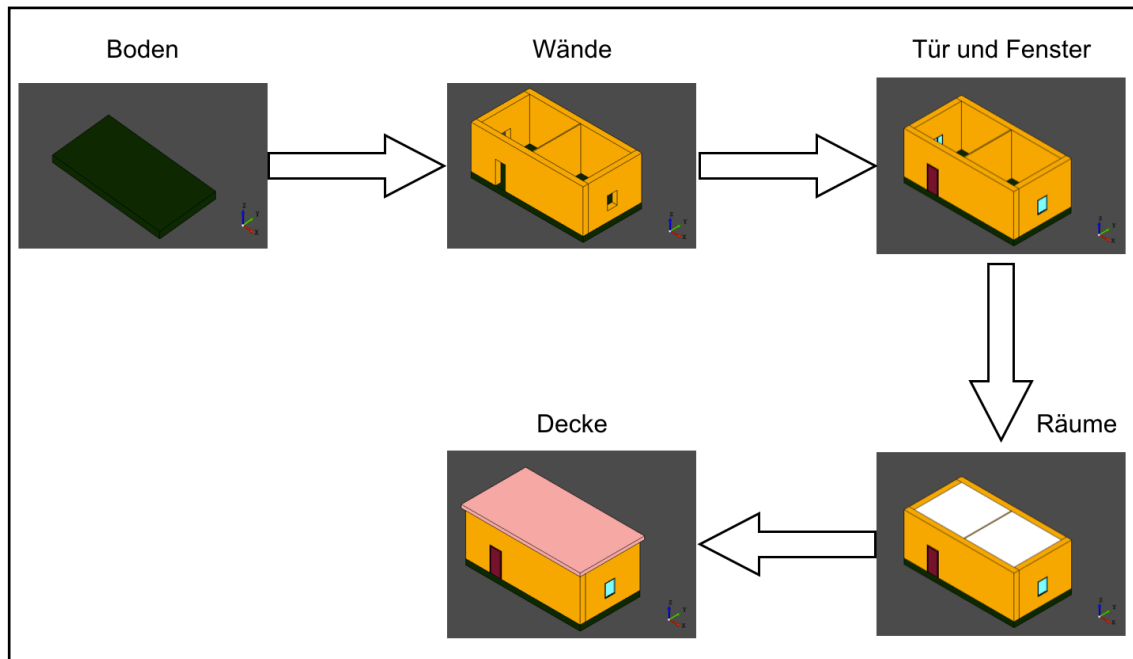


Abbildung 61 : Der Prozess der Modellierung

Die Visualisierung ist die Voraussetzung für die Kollisionsanalyse. Mit der Methodik ist es möglich, ein beliebiges Bauelement zu visualisieren. Im Kapitel 4 wird die Methodik zur Abfrage der Abmessungen und Eigenschaften eines Bauelementes erzählt. Mit der Visualisierung des einzigen Bauelementes wird der Prozess verbessert, und das Verfahren wird anschaulich aufgebaut.

5.2 Kollisionsanalyse

Als Kollisionserkennung oder Kollisionsabfrage (*englisch Collision Detection*) wird in der Algorithmischen Geometrie das Erkennen, das Berühren oder Überlappen zweier oder mehrerer geometrischer (starrer oder deformierbarer) Objekte im zwei- oder dreidimensionalen Raum verstanden [23].

Im Prozess des Datenaustausches gibt es die Möglichkeit, dass zwei Bauelemente miteinander kollidieren. Die Kollision kann Fehler verursachen und sie wird die Qualität und Anwendbarkeit eines Modells beeinträchtigen. Deswegen spielt die Überprüfung der Kollision für die Überprüfung eines IFC Modells eine große Rolle.

Die mathematische Grundlage ist die Berechnung zwischen Mengen. In der vorliegenden Abbildung 62 werden die verschiedenen Ergebnisse der Berechnung angezeigt.

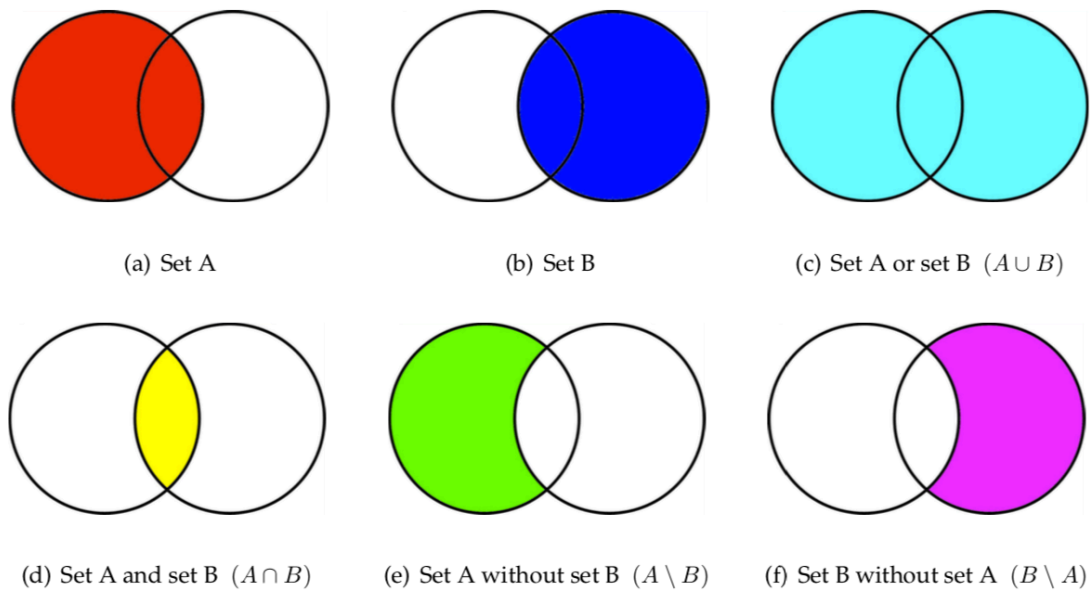


Abbildung 62 : Venn-Diagramme [24]

Venn-Diagramme stellen alle Beziehungen zwischen den betrachteten Mengen dar. Daher kann man an ihnen Zusammenhänge ablesen. Basierend auf den Diagrammen wird die Methodik zur Kollisionsanalyse abgeleitet. Durch boolesche Operatoren zwischen zwei Bauelementen werden die Beziehung und relative Standorte ermittelt. Mithilfe des Programms wird das Ergebnis im Bildschirm angezeigt [24].

In diesem Abschnitt werden zwei Fallstudien zur Validierung der Methodik untersucht. In der ersten Fallstudie werden zwei Objekte aufgebaut. Dazwischen gibt es überlappende Bereiche. In der zweiten Fallstudie werden zwei Objekte miteinander aufgebaut aber es gibt keine Überlappung. In der unteren Tabellen 15 und 16 werden die geometrischen Parameter der Objekten angezeigt.

Fallstudie 1				
Objekt	Koordinate	X Dim (cm)	Y Dim (cm)	Z Dim (cm)
Objekt1	(0,0,0)	50	30	10
Objekt2	(0,0,0)	10	30	50

Tabelle 15 : Geometrische Daten in Fallstudie 1

Fallstudie 2				
Objekt	Koordinate	X Dim (cm)	Y Dim (cm)	Z Dim (cm)
Objekt1	(0,0,0)	50	30	10
Objekt2	(-10,0,0)	10	30	50

Tabelle 16 : Geometrische Daten in Fallstudie 2

In der vorliegenden Abbildung 63 wird die Methodik zur Überprüfung der Kollision angezeigt.

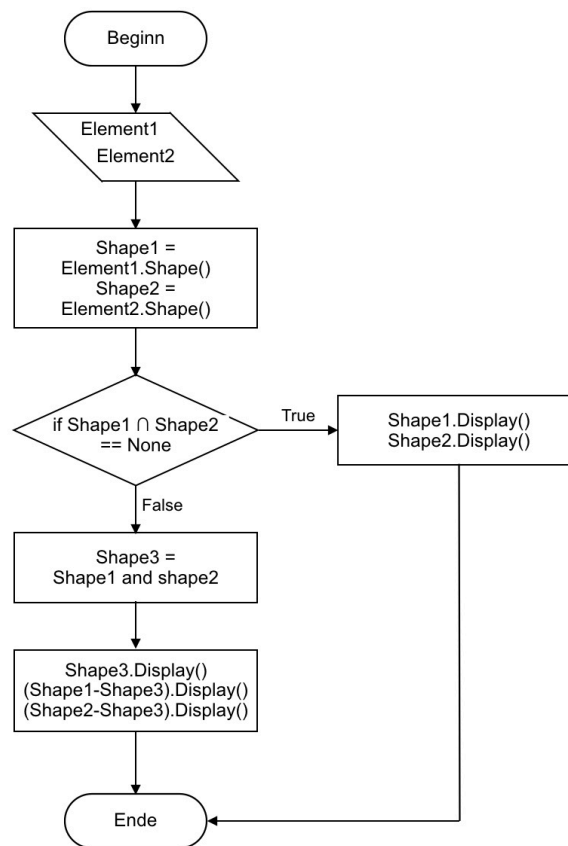


Abbildung 63 : Ablauf der Überprüfung der Kollision

Wie in der obengenannten Abbildung angezeigt, werden zuerst die zwei Objekte durch Programmierung analysiert und die 3D Modelle der Objekte werden aufgebaut. Dann werden boolesche Operationen zwischen diesen durchgeführt. Wenn das Ergebnis richtig ist, wird der überlappende Bereich erstellt. Die Repräsentationen werden am Bildschirm angezeigt. Die Überlappung wird in Rot dargestellt. Ein falsches Ergebnis bedeutet, dass keine Überlappung zwischen diesen zwei Objekten existiert. Die Darstellungen werden in Weiß angezeigt.

Aufgrund der Strategie werden die Ergebnisse in der vorliegenden Abbildung 64 angezeigt. Sie wird in zwei Teile aufgeteilt. Der linke Teil ist die Visualisierung der Fallstudien und der rechte Teil ist die dazugehörige Funktion. Es ist einfach zu sehen, dass es in der ersten Fallstudie keine Kollision gibt. In der zweiten Fallstudie wird der überlappende Bereich in Rot angezeigt.

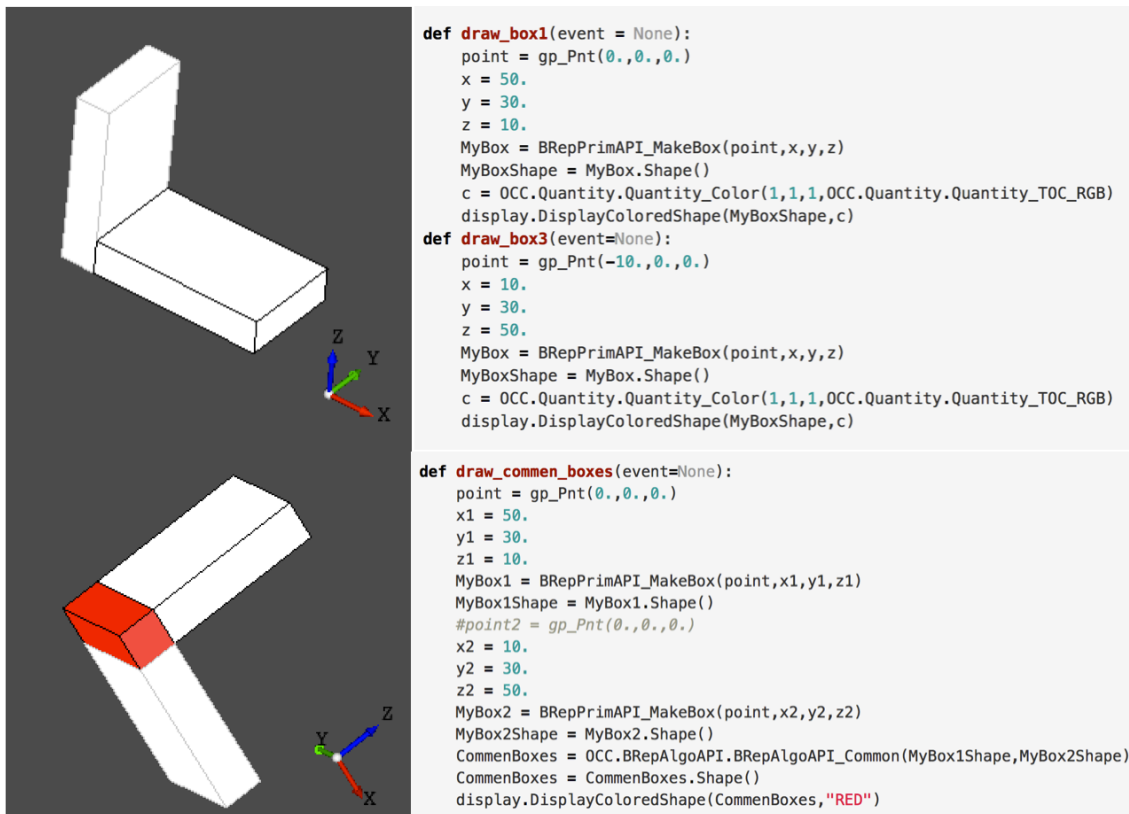
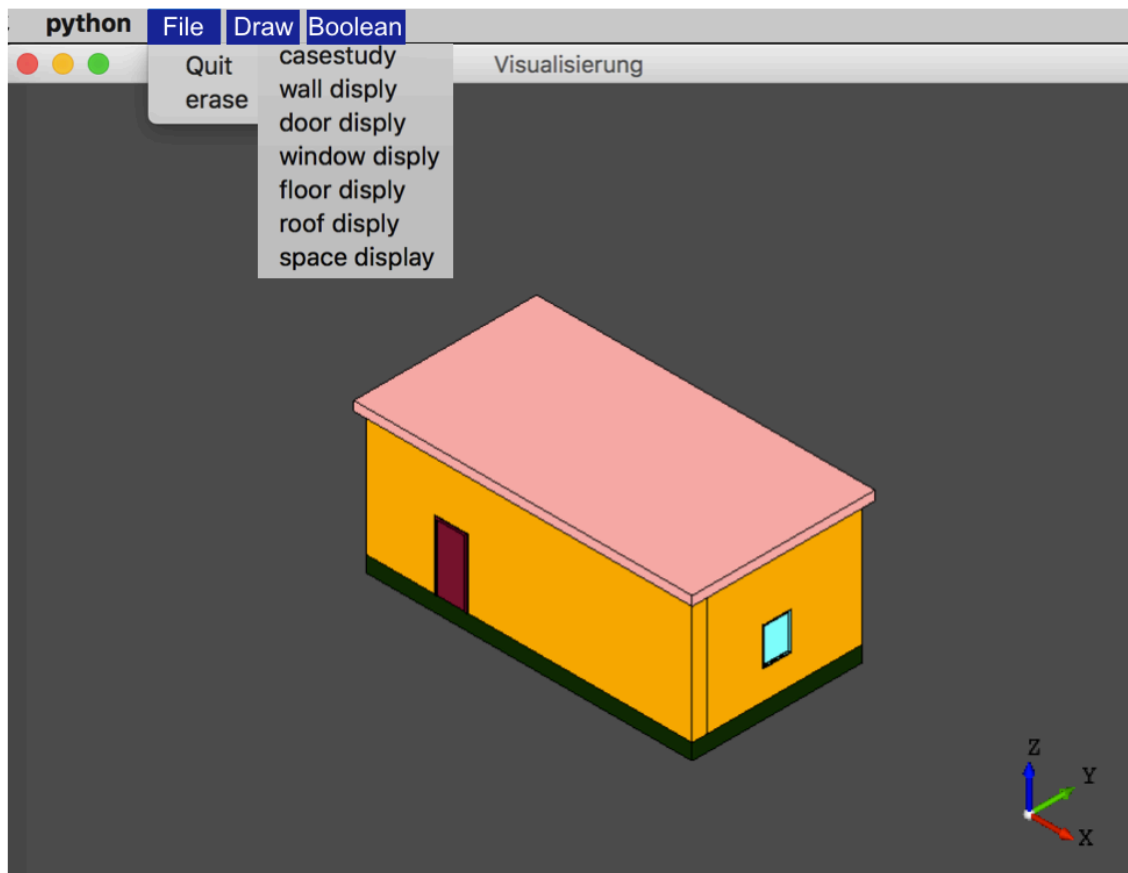


Abbildung 64 : Ergebnisse der Überprüfung für Kollision

5.3 Grafische Benutzeroberfläche (GUI)

GUI bezeichnet eine Form von Benutzerschnittstellen eines Computers. Sie hat die Aufgabe, Anwendungssoftware auf einem Rechner mittels grafischer Symbole, Steuerelemente oder auch Widgets genannt, bedienbar zu machen. Aufgrund der obengenannten Funktionen kann man eine GUI erstellen. Sie ist in der Lage, nicht nur Visualisierung sondern auch Überprüfungsfunktionen zu einem Paket zusammenzufügen. Alle Funktionen können durch Klicken auf die entsprechende Schaltfläche erreicht werden. Man kann die Methodik dahinter ignorieren. Die Skalierbarkeit wird erhöht. Aufgrund dieser Plattform kann man mehr Funktionen entwickeln.

Zum Aufbau der GUI wird ein Paket „WxPython“ verwendet. Der grundlegende Antrieb zur Visualisierung wird bei „PyQt“ angeboten. Bei Booleschen Operatoren spielt das Paket „PythonOCC“ eine große Rolle. In der vorliegenden Abbildung 65 wird die GUI und der Quelltext angezeigt [25].



```
class AppFrame(wx.Frame):
    def __init__(self, parent):
        wx.Frame.__init__(self, parent, -1, "Visualisierung",
                           style = wx.DEFAULT_FRAME_STYLE, size = (640,480))
        self.canva = wxViewer3d(self)
        self.menuBar = wx.MenuBar()
        self._menus = {}
        self._menu_methods = {}
        #self.SetBackgroundColor(wx.RED)
        # Function for creating new menus like File, Edit, View, and so on
        # The stuff appearing at the top
        def add_menu(self, menu_name):
            _menu = wx.Menu()
            self.menuBar.Append(_menu, "&" + menu_name)
            self.SetMenuBar(self.menuBar)
            self._menus[menu_name] = _menu
        # Function for creating new menu items like File-New, File-Exit, Edit-Copy,
        # Edit-Cut, Edit-paste, and so on
        # The stuff appearing if a menu is selected
        def add_function_to_menu(self, menu_name, _callable):
            _id = wx.NewId()
            assert callable(_callable), 'the function supplied isnt callable'
            try:
                self._menus[menu_name].Append(_id, _callable.__name__.replace('_', ' ').lower())
            except KeyError:
                raise ValueError, 'the menu item %s doesnt exist' % (menu_name)
            self.Bind(wx.EVT_MENU, _callable, id=_id)
```

Abbildung 65 : GUI und Quelltext [24]

6. Fehleranalyse für das vorhandene BIM Modell

6.1 Grunddaten des Modells

Das Modell ist ein Bürogebäude. In der Abbildung 66 wird das 3D Modell angezeigt.

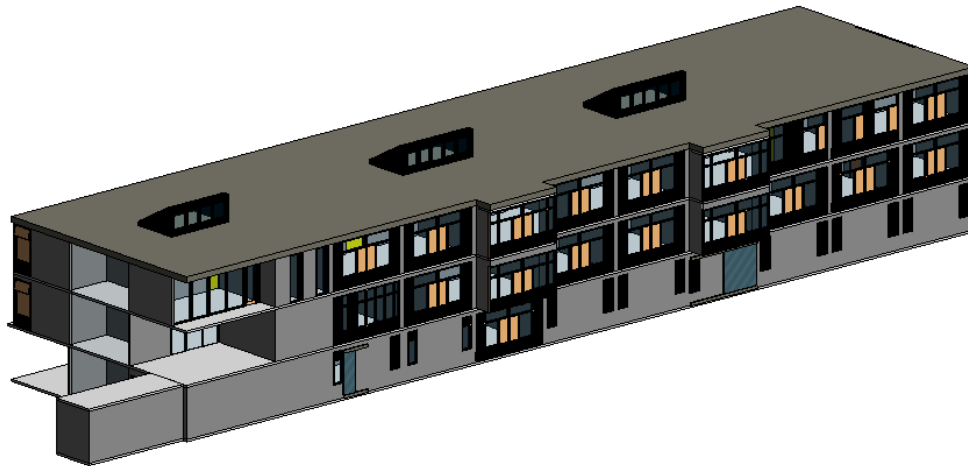


Abbildung 66 : 3D-Perspektive des Modells

Wie in der Abbildung dargestellt, besteht das Gebäude aus drei Ebenen. Die Höhe ist 11.5 Meter und die Schichthöhe ist 3.5 Meter. Der Eingang des Gebäudes ist an der Südfassade. Das Dach ist ein Flachdach und es gibt drei Luken darauf. Die Abbildung 67 zeigt die Ansichten von Westen und Osten.

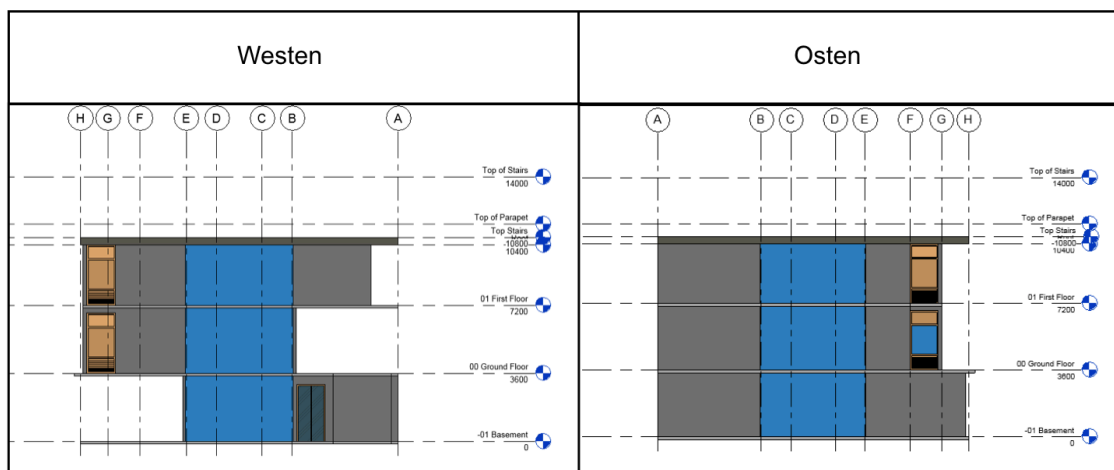


Abbildung 67 : Ansichten von Westen und Osten

In der Abbildung ist zu erkennen, dass zwei Glasfassaden sich in beiden Seiten des Gebäudes befinden. Ein weiterer Ausgang befindet sich auf der Westseite.

Die Abbildung 68 zeigt die Nordseite und Südseite des Gebäudes.

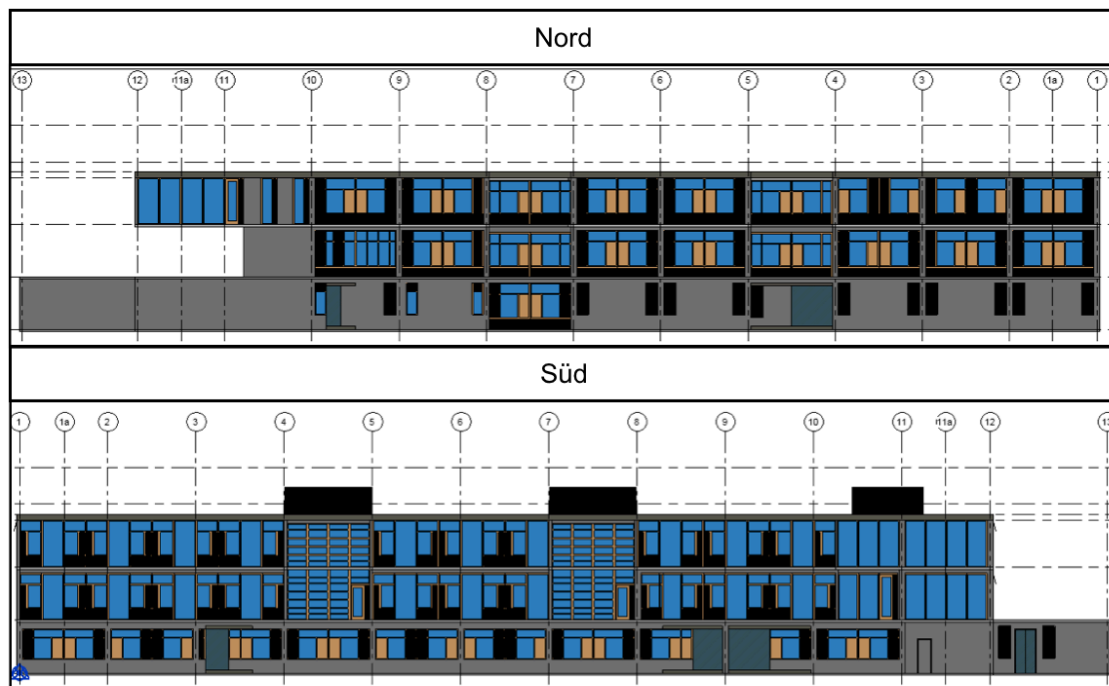


Abbildung 68 : Ansichten von Nordseite und Südseite

Das Modell wird in *Revit* aufgebaut und wird als Eingabedatei in der Software *IES <VE>* zur Energiesimulation importiert. Basierend auf der im Kapitel 2 und 3 erzählten Theorie wird das Modell zuerst als IFC Datei exportiert und dann wird die IFC Datei als Eingabe verwendet. Aufgrund der Komplexität des Modells und der Kompatibilität zwischen IFC Standard und *Revit* werden im Prozess der Umwandlung viele Fehler verursacht. Im nächsten Abschnitt werden die Fehler klassifiziert und angezeigt.

6.2 Fehlerprotokoll

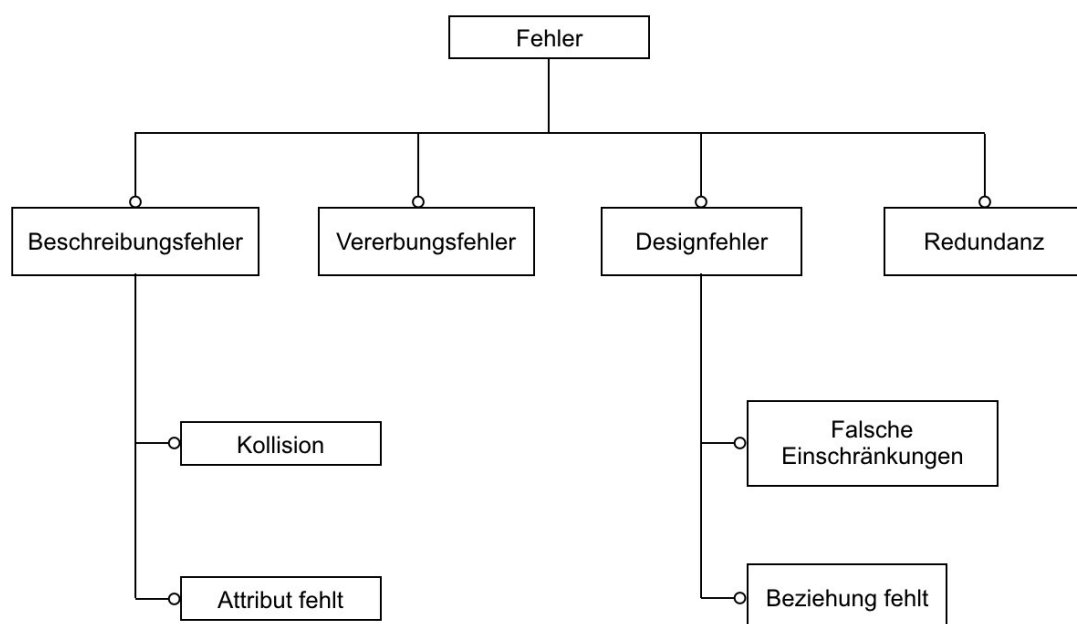


Abbildung 69 : Fehlerklassifizierung

Nach der Untersuchung und Überprüfung des vorhandenen Modells können die Fehler in vier Kategorien unterteilt werden. Wie in der Abbildung 69 dargestellt, werden die Kategorien jeweils als „Beschreibungsfehler“, „Vererbungsfehler“, „Designfehler“ und „Redundanz“ benannt. Darüber hinaus haben die „Beschreibungsfehler“ und „Designfehler“ Unterkategorien.

Aufgrund der Grundlagen des IFC Standards lässt sich feststellen, dass das IFC Modell durch eine objektorientierte Sprache *EXPRESS* beschrieben wird. Jedes Bauelement wird als eine Instanz aufgerufen. Die Eigenschaften eines Bauelementes wie Standort, geometrische Form und Materialien werden als Attribute definiert. Wenn diese Eigenschaften Fehler verursachen, werden sie als Beschreibungsfehler klassifiziert.

Ein Vererbungsfehler wird aufgrund falscher Vererbungsbeziehungen erstellt. Zum Beispiel sollte eine Wand bei der Entität *IfcWall* beschrieben werden, aber wegen der schlechten Austauschbarkeit oder des menschlichen Fehlers wird eine Wand als *IfcSlab* definiert. Die beiden Entitäten haben verschiedene Attribute und Funktionen. Daraus ergeben sich die Probleme.

Designfehler hat zwei Unterkategorien. Sie werden in Einschränkungsfehler und Beziehungsfehler unterteilt. Wenn man in *Revit* eine Wand definiert, werden bei der Software zwei Verfahren zur Beschreibung der Höhe der Wand angeboten. Das erste Verfahren ist, dass durch die Definition der geometrischen Parameter eine Wand aufgebaut wird. Die zweite Variante ist, durch Einschränkungen wie die verbundenen Elemente der Wand erstellt werden. Wenn eine falsche Verknüpfung eines Bauelementes eingegeben wird, wird dadurch ein Fehler verursacht. Beziehungsfehler bedeutet, wenn man ein Fenster oder eine Tür modelliert. Im IFC Standard ist die *Verknüpfung* zwischen *IfcWindow* und *IfcWall* durch *IfcOpening* erfolgreich. Wenn die Entität im Prozess des Datenaustausches verloren geht, wird ein Fehler erstellt. Diese Fehler werden als Beziehungsfehler klassifiziert.

Zum Schluss wird die Ursache für Redundanz beschrieben. Die räumlichen Objekte werden in *Revit* nicht nur automatisch, sondern auch manuell erstellt. Die Typen der räumlichen Objekte sind vielfältig, z. B. Raum und MEP-Raum. Aber im IFC Standard gibt es nur eine entsprechende Entität *IfcSpace*. Wenn ein BIM Modell zu einer IFC Datei umgewandelt wird, werden alle räumlichen Objekte als *IfcSpace* exportiert. Das Verhalten kann Redundanz verursachen.

Aufgrund der Klassifizierung werden einige Beispiele aufgelistet. Basierend auf der dargestellten Methodik werden die obengenannten Fehler korrigiert.

6.2.1 Kollision zwischen Fenster 2.30 und Fenster 2.22

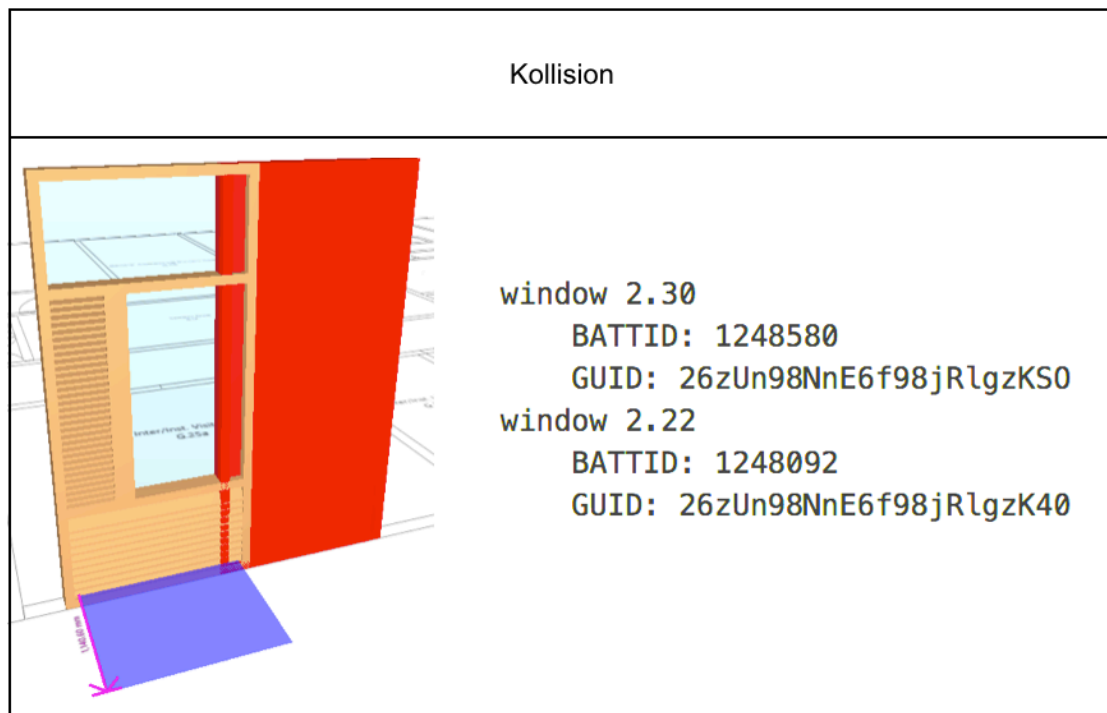


Abbildung 70 : Beispiel für die Kollision

In der Abbildung 70 wird ein Beispiel für die Kollisionsfehler dargestellt. Es ist einfach zu sehen, dass eine Überschneidung zwischen den zwei Fenstern entsteht. Die GUID davon werden angezeigt. Man kann durch GUID die Fehler schnell korrigieren.

Zuerst wird die vorhandene IFC Datei mithilfe der im Kapitel 5 erzählten GUI analysiert. Die Suchfunktion ist in der Lage, ein Bauelement zu lokalisieren und in dem 3D Modell anzuzeigen. Die Abbildung 71 zeigt den Standort der Fenster.

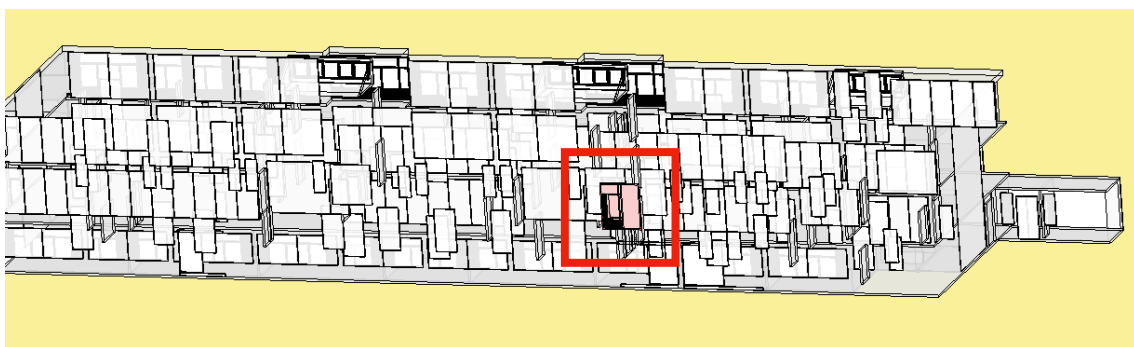


Abbildung 71 : Standort der Fenster

In der Abbildung ist zu erkennen, dass die zwei Fenster sich in der erste Ebene befinden, und wie in der Abbildung 70 dargestellt, sie sich überschneiden.

Um das Problem zu lösen, werden zwei Strategien aufgestellt. Die einfachste ist Modifizierung des originalen Modells in *Revit*. Anschließend wird das modifizierte Modell wieder als IFC Datei exportiert. Außerdem wird eine Variante basierend auf der Methodik angeboten. Durch Programmierung ist es möglich, die IFC Datei direkt zu modifizie-

ren und das Ergebnis wird im Bildschirm angezeigt. Die grundlegende Methodik ist Parallelverschiebung einer Fenster entlang der X-Achse. Wenn die Fläche des überlappenden Bereiches null ist, wird ein neues Modell erzeugt und im Bildschirm angezeigt. Die Abbildung 72 zeigt das Ergebnis und den dazugehörigen Quelltext.



Abbildung 72 : Ergebnis der Modifizierung

Zur Modifizierung der mögliche neue überlappung zwischen das neue Fenster und die Wand wird bei PythonOCC eine Funktion angeboten.

CuttedShapes = OCC.BRepAlgoAPI.BRepAlgoAPI_Cut(Wand,neue Fenster)

Mithilfe der Funktion wird den überlappenden Bereich der Wand geschnitten.

Prinzipiell werden die Kollisionsfehler zwischen zwei Objekten durch das Verfahren modifiziert. Aber eine Beschränkung davon ist die Leistung des Rechners. Der Grundgedanke ist Zirkulation und Berechnung der Flächen. Wenn die Datei zu groß und kompliziert ist, fällt die Geschwindigkeit ab und die Modifizierung dauert sehr lange.

6.2.2 Definition der Öffnungsrichtung

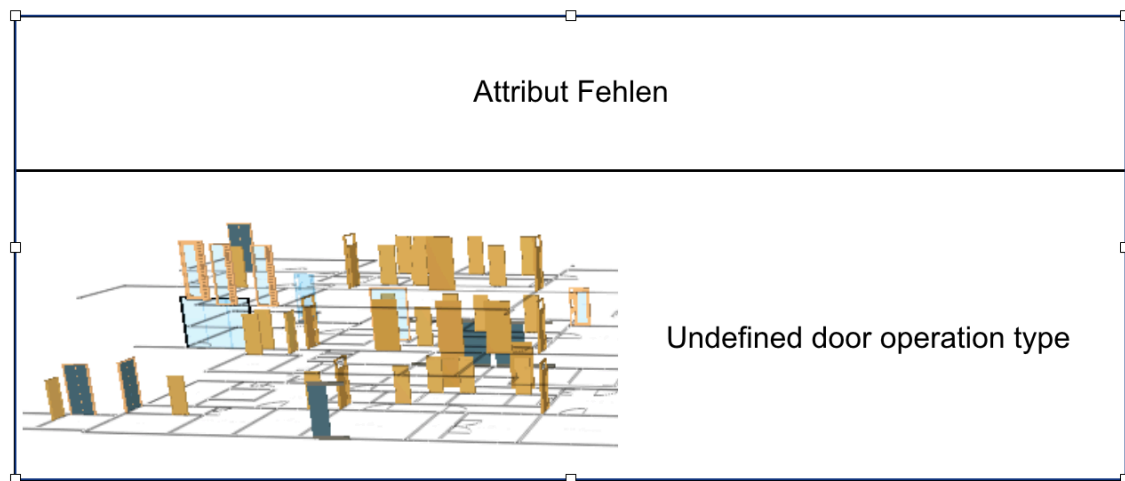


Abbildung 73 : Beispiel für Attribut Fehler

Die Abbildung 73 zeigt die fehlenden Attribute. Wie in der Abbildung angezeigt, werden die Steuerungsverfahren wie die Öffnungsrichtung der Türen nicht definiert. Wenn eine IFC Datei erstellt wird, wird die bezügliche Entität als Platzhalter dargestellt.

Basierend auf der Datenstruktur der *IfcDoor* lässt sich feststellen, dass die Öffnungsrichtung einer Tür in der IFC Datei durch „OperationType“ beschrieben wird. Sie wird durch die Entität *IfcDoorStyle* definiert. Um die nicht definierte Öffnungsrichtungen hinzufügen werden zuerst alle bezüglichen Entitäten *IfcDoorStyle* ausgewählt und dann durch Programmierung modifiziert. Die Abbildung 74 zeigt den Quelltext zur Modifizierung.

```
ifcfile = ifcopenshell.open("/Users/luyifan/Desktop/DA/g20f_anderung.ifc")
ifcdoorstyles = ifcfile.by_type("ifcdoorstyle")
not_defined = []
for s in ifcdoorstyles:
    if s.OperationType == "NOTDEFINED":
        not_defined.append(s)
for s in not_defined:
    s.__setattr__("OperationType", "USERDEFINED")
```

Definition der
Öffnungsrichtung
↙

Abbildung 74 : Quelltext zur Definition der Öffnungsrichtung der Türen

Der Datentyp von *IfcDoorTypeOperationEnum* ist ein Aufzählungstyp. Ein paar Öffnungsrichtungen werden bei IFC Standard angeboten. Benutzer können den richtigen Typ nach ihren Bedürfnissen auswählen. Als Beispiel werden „USERDEFINED“ in diesem Modell verwendet. Die Abbildung 75 zeigt das Ergebnis nach der Modifizierung.

Original	Korrigiert
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
NOTDEFINED	USERDEFINED
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
SINGLE_SWING_RIGHT	SINGLE_SWING_RIGHT
SINGLE_SWING_LEFT	SINGLE_SWING_LEFT
NOTDEFINED	USERDEFINED
NOTDEFINED	USERDEFINED
DOUBLE_DOOR_SINGLE_SWING	DOUBLE_DOOR_SINGLE_SWING
NOTDEFINED	USERDEFINED

Die
Öffnungsrichtung
kann man selbst
definieren

Abbildung 75 : Ergebnis nach der Modifizierung

Aus der Abbildung ist es deutlich zu sehen, dass die nicht definierten Öffnungsrichtungen hinzugefügt werden.

6.2.3 Vererbungsfehler bei Fenster 2.53

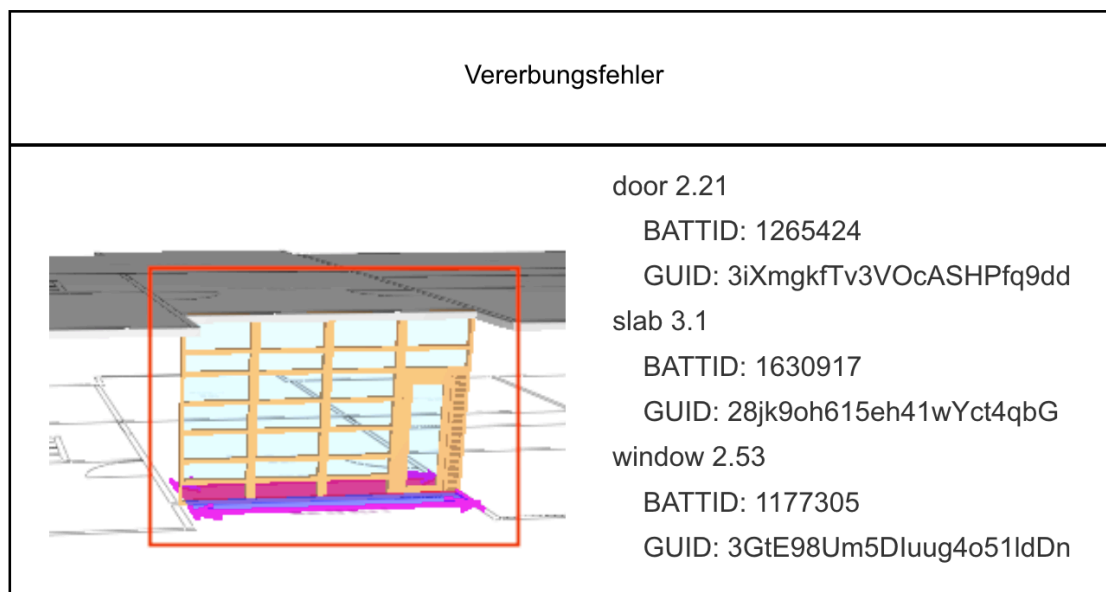


Abbildung 76 : Beispiel für Vererbungsfehler

In der Abbildung 76 wird ein Vererbungsfehler angezeigt. Aus der Abbildung lässt sich feststellen, dass die sogenannten Fenster 2.53, der Fachman falsch klassifiziert hat. Das Objekt soll als *IfcCurtainWall* definiert werden. Aber in der IFC Datei wird es als *IfcWindow* benannt. Aufgrund der Datenstruktur des IFC Standards ist die Verbindung zwischen *IfcWindow* und *IfcWall* durch *IfcOpening* erfolgreich. Aber die Entität *IfcOpening* existiert nicht. Zusammenfassend lässt sich feststellen, dass sie einen Vererbungsfehler verursachen.

Die einfachste Strategie zur Korrektur der Fehler ist Modifizierung des originalen Modells in *Revit*. Durch Änderung des Entwurfs und Auswahl des richtigen Elementes wird das Problem gelöst. Programmierung ist auch eine Variante, aber das Verfahren ist zu kompliziert. Ein Grundgedanke ist, dass zuerst die Entität *IfcWindow* gelöscht wird. Und dann wird eine neue Entität *IfcCurtainWall* wieder aufgebaut. Der Prozess ist gleich wie in *Revit*. Aber *Revit* hat bessere Werkzeuge und Funktionen. Deswegen wird das Programmierungsverfahren nicht in der Arbeit beschrieben.

6.2.4 Einschränkungsfelder bei Raum G20.d

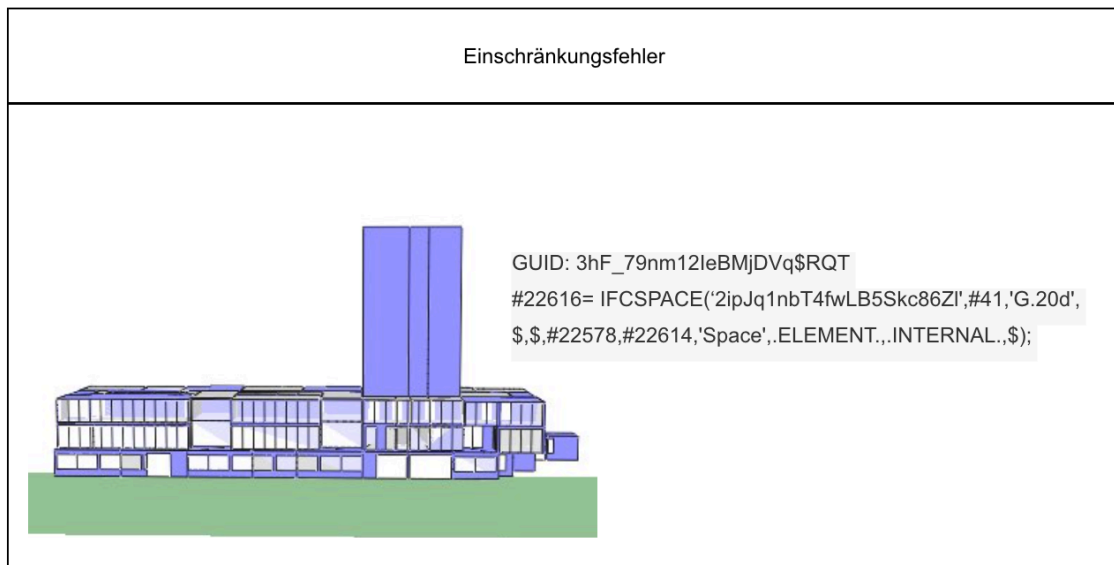


Abbildung 77 : Beispiel für Einschränkungsfelder

Die Abbildung 77 zeigt einen Einschränkungsfelder. Bei einem Raum, der sich in der zweiten Ebenen befindet, wird die Obergrenze falsch definiert. Sie sollte mit der Decke verbunden werden, aber die Verknüpfung ist verloren. Wenn räumlich Objekte einstellt werden, wird ein Fehler dabei verursacht.

Um das Problem zu lösen, muss die Obergrenze des Raumes hinzugefügt werden. Für die Fehler ist die Modifizierung in *Revit* auch die beste Variante. Der Grund dafür ist, dass die Fehler in dem Modell nur einmal existieren. Durch *Revit* wird die Modifizierung vereinfacht und beschleunigt. Die Abbildung 78 zeigt das Ergebnis der Änderung.

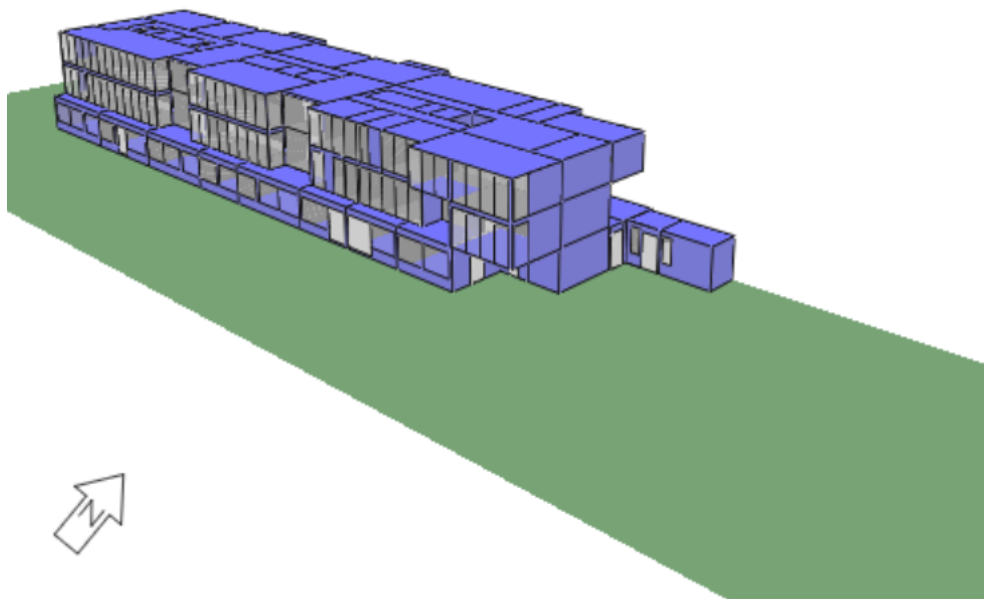


Abbildung 78 : Neues Modell nach der Korrektur

Aus der Abbildung ist es einfach zu sehen, dass das unerwartete Objekt verschwindet. Die Modifizierung ist erfolgreich.

6.2.5 Beziehungsfehler

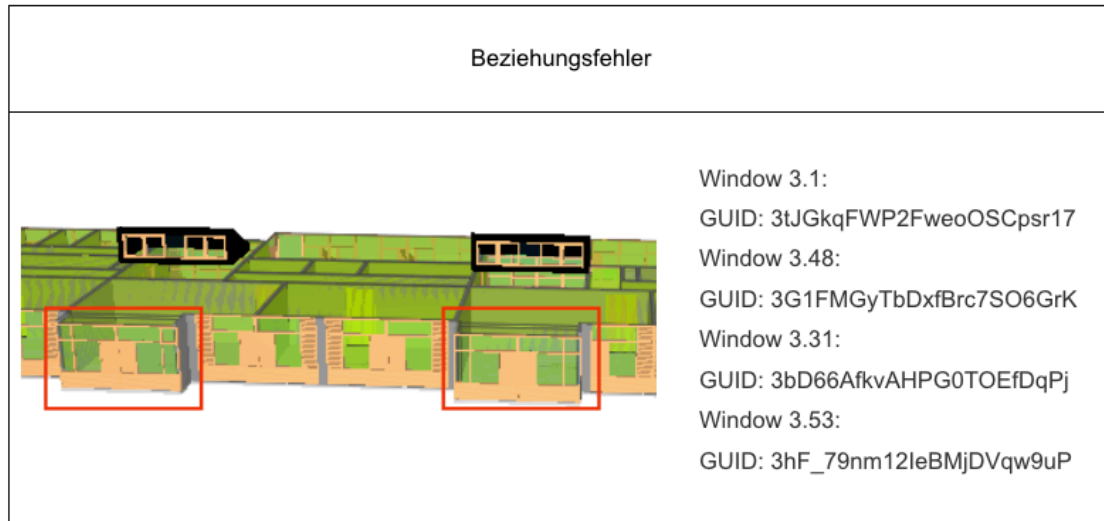


Abbildung 79 : Beispiel für Beziehungsfehler

Die Abbildung 79 zeigt ein Beziehungsfehler. Aus der Abbildung ist es einfach abzuleiten, dass die zwei Fenster nicht in der Lage sind, mit der Wand verknüpft zu werden. Im Prozess der Umwandlung wird die Entitäten *IfcOpening* nicht automatisch eingestellt. Das ist die Ursache der Fehler.

Zur Modifizierung der Fehler kann man zuerst in *Revit* die Fenster löschen und vier passende Fenster auswählen und dann wieder aufbauen.

6.2.6 Redundanz

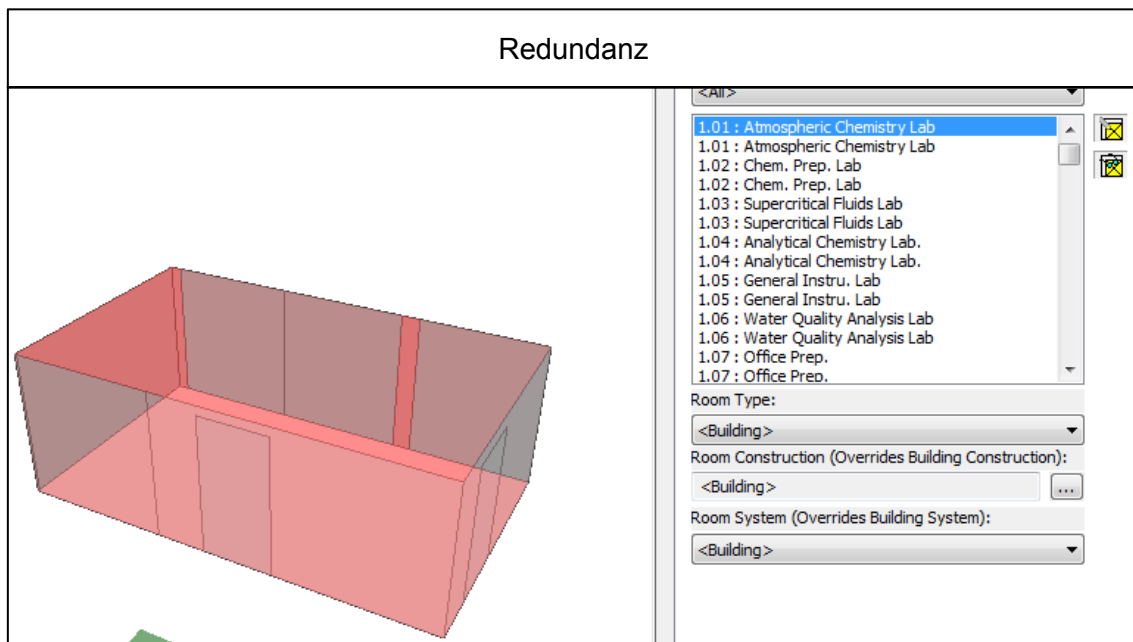


Abbildung 80 : Fehler bei Redundanz

Zum Schluss werden die durch Redundanz verursachten Fehler in der Abbildung 80 angezeigt. Aus der Abbildung kann man deutlich sehen, dass alle räumlichen Objekte doppelt definiert werden. In *Revit* werden die Fehler nicht visualisiert. Aber wenn das Modell zur IFC Datei umgewandelt wird, werden für die räumlichen Objekte zwei entsprechende *IfcSpace* entstehen. Die Hälfte davon ist bedeutungslos. Diese Fehler werden als Redundanz klassifiziert.

Für das Problem kann man die Einstellung in *Revit* verändern, wenn eine IFC Datei exportiert wird, kann das Objekt „MEP-Raum“ nicht als „*IfcSpace*“ exportiert werden. Die falsche Einstellung ist der Grund des Problems. Man muss während des Datenaustausches die Einstellungen der Software *Revit* beachten.

7. Fazit und Ausblick

7.1 Fazit

BIM Technik ist eine Revolution in der Bauindustrie. Sie ist von weitreichender Bedeutung. Zuerst ist sie in der Lage, die Arbeitsverfahren in der Bauindustrie zu verbessern. Die Kommunikation und Kooperation zwischen den Fachleuten wird bequemer. Die Koordination und Kompatibilität zwischen verschiedenen Softwares werden erhöht. Darüber hinaus kann man die Leistungsfähigkeit erhöhen. BIM ist eine wichtige Verkörperung der Informatisierung in der Bauindustrie. Das Verfahren verbindet das Bauingenieurwesen mit fortschrittlicher Computertechnologie. Basierend auf BIM werden viele interdisziplinäre Projekte erstellt und schnell entwickelt, wie energieeffiziente Gebäude, Internet der Dinge (*IdD*) oder 3D Druck mit Beton. Technologische Innovation bedeutet industrielle Modernisierung. Alte rückständige Produktionsmethoden werden eliminiert. Die neue und effiziente Arbeitsweise wird verwendet und hat die zwangsläufig Auswirkungen auf die Produktionsbeziehung. Daher betrifft die neue Technologie nicht nur eine Branche, sondern die gesamte Gesellschaft.

In Bezug auf Kooperation muss man das Medium aufmerksam verfolgen. Ein gutes Medium ist in der Lage, den Prozess des Datenaustausches zu verbessern. IFC Dateien übernehmen heutzutage diese Rolle. Diese werden als Verknüpfung zwischen mehreren Softwares verwendet. Aber der IFC Standard wird zurzeit nicht weiter entwickelt. Während der Verwendung treten viele Fehler auf. Die Genauigkeit kann nicht garantiert werden.

In der Arbeit wird die Nutzung der IFC Datei als Eingabe zur Energiesimulation untersucht. Basierend auf der Grundlage von Energiesimulation und Datenstruktur der IFC Datei wird eine Methodik zur Überprüfung des IFC Modells entwickelt. Zur Validierung der Methodik wird ein Gebäudemodell aufgebaut und analysiert. Darüber hinaus wird eine *GUI* als Plattform zur Visualisierung aufgebaut.

Aufgrund der Untersuchung lässt sich feststellen, wie IFC Dateien in der Lage sind, die Informationen eines Modells gut darzustellen. Aber die Simulationssoftware *IES <VE>* kann nur die räumlichen Daten erkennen. Wenn eine IFC Datei als Eingabe bei *IES <VE>* verwendet wird, wird ein Modell basierend auf den räumlichen Daten automatisch erzeugt. Die Eigenschaften eines Bauelementes und Materialdaten werden bei der Software ignoriert. Man muss in der Simulationssoftware die fehlenden Daten wieder hinzufügen. Allgemein gesagt, die Verknüpfung zwischen *IES <VE>* und IFC Datei ist nicht ausreichend. Aber die IFC Datei ist in der Lage, fast alle Informationen eines Gebäudes zu beschreiben.

Zum Schluss wird das vorhandene IFC Modell überprüft. Anschließend werden die Fehler in vier Typen klassifiziert. Für jeden Typ wird ein Beispiel angegeben. Mit dem obengenannten Protokoll sind die Fachleute in der Lage, schnell die Fehler zu lokalisieren und zu korrigieren.

7.2 Ausblick

Zusammenfassend kann man sagen, dass die Verwendung der IFC Dateien als Eingabe in *IES <VE>* viele Probleme verursachen kann. Die Schnittstelle der Software muss

in der nächsten Version verbessert werden. Als Variante wird von der Software noch ein Datentyp „*gbxml*“ angeboten. Man kann das Modell in *Revit* als eine „*gbxml*“ Datei exportieren und dann in *IES <VE>* importieren.

Zur Ermittlung der Durchführbarkeit des Verfahrens wird das im Kapitel 4 aufgebaute Modell wieder als „*gbxml*“ exportiert und untersucht. Die Abbildung 81 zeigt das Ergebnis der Untersuchung.

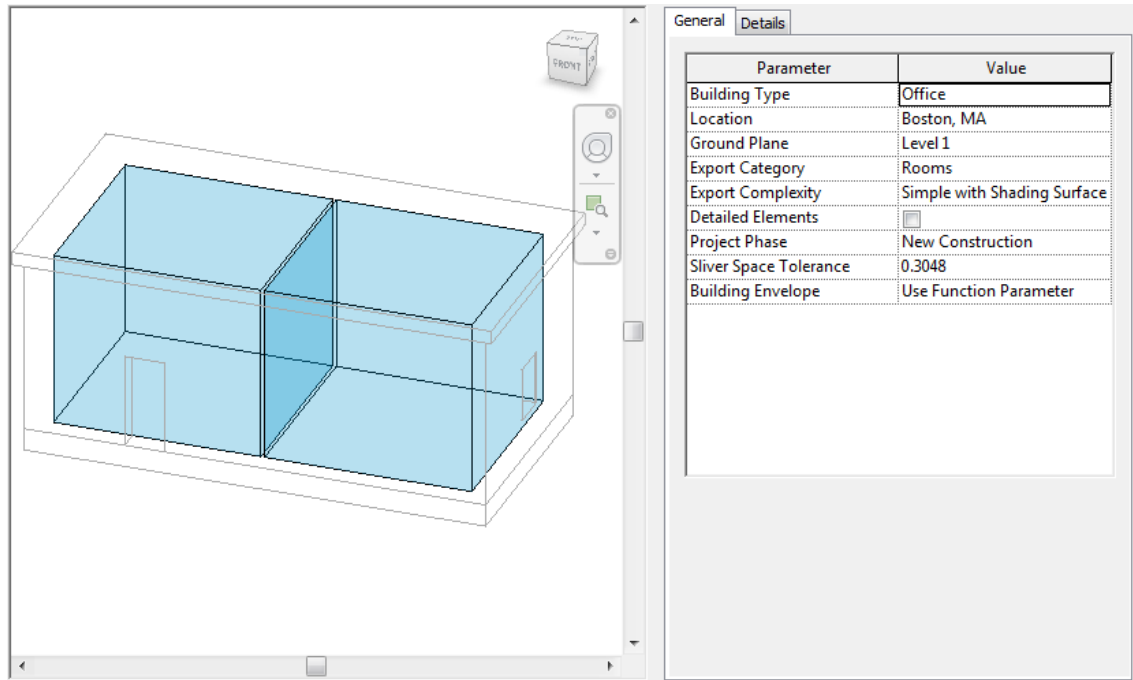
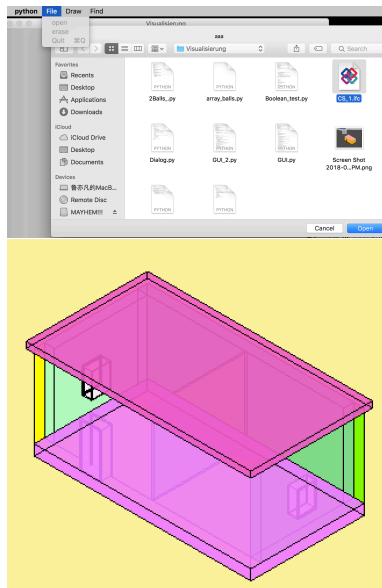


Abbildung 81 : *gbxml* Datei als Eingabe für *IES <VE>*

Aus der Abbildung lässt sich feststellen, dass nicht nur die räumlichen Daten sondern auch alle Bauelemente importiert werden. In Bezug auf die Software ist die Austauschbarkeit der „*gbxml*“ Datei besser als IFC Datei. Aber die Materialdaten und die thermischen Eigenschaften werden auch ignoriert.

Alles in allem ist die Austauschbarkeit von IFC noch nicht vollständig entwickelt wie es bei *gbxml* der Fall ist. Sie müssen noch verbessert werden.

Darüber hinaus wird das im Kapitel 5 erzählte Programm verbessert. Ein paar Funktionen werden hinzugefügt. Die Abbildung 82 zeigt die neuen Funktionen des Programms. Durch den Druckknopf „Open“ kann man IFC Dateien importieren. Die neue Version hat die Schnittstelle angeboten, um die Farbe und Transparenz jedes Bauelementes zu ändern. Die Hintergrundfarbe der *GUI* kann man auch nach Wunsch selbst einstellen.



```
def Open(event):
    wildcard = "IFC Files (*.ifc)|*.ifc"
    dlg = wx.FileDialog(None, "aaa", os.getcwd(), wildcard, style = wx.FD_OPEN)
    if dlg.ShowModal() == wx.ID_OK:
        file = dlg.GetPath()
        file = file.encode('unicode-escape').decode('string_escape')
        global file
    dlg.Destroy()

def rgb_color(r,g,b):
    return OCC.Quantity.Quantity_Color(r, g, b, OCC.Quantity.Quantity_TOC_RGB)

random_color = rgb_color(random(),random(),random())
display.DisplayShape(s,color=random_color, transparency= 0.3)

def set_bg_gradient_color(self, R1, G1, B1, R2, G2, B2):
    """ set a bg vertical gradient color.
    R, G and B are floats.
    """
    aColor1 = rgb_color(float(R1)/255., float(G1)/255., float(B1)/255.)
    aColor2 = rgb_color(float(R2)/255., float(G2)/255., float(B2)/255.)
    self.View.SetBgGradientColors(aColor1, aColor2, 2, True)
```

Abbildung 82 : Neue Funktionen des Programms

Als eine primäre Untersuchung der Anwendung und Überprüfung der IFC Datei erfüllt das Programm die erwarteten Anforderungen und bietet eine Visualisierung des IFC Modells. Aber als eine Probe hat das Programm noch Probleme. Ein Problem ist die langsame Rechner Geschwindigkeit bei Importierung einer großen Datei. Ein Grund dafür ist die Effizienz des Codes. Von diesem Programm werden nur grundlegende Funktionen angeboten. Deswegen kann es nicht mit kommerzieller Software wie „Solibri Model Checker“ verglichen werden. Daher sind Änderungen und Verbesserungen in der Zukunft erforderlich.

Literaturverzeichnis

- [1] D. Energie-Agentur, Der dena-Gebäudereport...: Statistiken und Analysen zur Energieeffizienz im Gebäudebestand. Deutsche Energie-Agentur, 2012.
- [2] U. G. B. Council, Green building design and construction: LEED reference guide for green building design and construction. US Green Building Council, 2009.
- [3] K. Hausknecht and T. J. B. I. M. a. n. P. Liebich, Stuttgart, "BIM-Kompendium," 2016.
- [4] W. R. Thompson, "The UK BIM Revolution," 2017.
- [5] Y. Jiang *et al.*, "BIM server requirements to support the energy efficient building lifecycle," in *Computing in Civil Engineering (2012)*, 2012, pp. 365-372.
- [6] M. Vonier, Nachhaltiges Kommunikationsmanagement zur Erlangung sozialer Akzeptanz von Bau-und Infrastrukturprojekten: dargestellt anhand einer Fallstudie zweier Kraftwerksprojekte. na, 2013.
- [7] B. SMART. Available: <http://www.buildingsmart.com>
- [8] R. Petrie, "buildingSMART Data Dictionary Strategic Overview," ed: Visited, 2016.
- [9] A. Borrmann, M. König, C. Koch, and J. Beetz, "Building Information Modeling: Why? What? How?," in *Building Information Modeling*: Springer, 2018, pp. 1-24.
- [10] C.-S. Cho, D. Chen, and S. J. J. o. K. Woo, "Building information modeling (BIM)-Based design of energy efficient buildings," vol. 2, no. 1, pp. 1-6, 2012.
- [11] S. Pinheiro *et al.*, "MVD based information exchange between BIM and building energy performance simulation," vol. 90, pp. 91-103, 2018.
- [12] Z. Maa and Z. Liua, "Ontology-and Freeware-Based Platform for Rapid Development of BIM Applications with Reasoning Support."
- [13] W. Solihin, C. Eastman, Y.-C. Lee, and D.-H. Yang, "A simplified relational database schema for transformation of BIM data into a query-efficient and spatially enabled database," *Automation in Construction*, vol. 84, pp. 367-383, 2017/12/01/ 2017.
- [14] A. J. G.-G. Perschk, "Gebäude-und Anlagensimulation—Ein" Dresdner Modell", vol. 131, no. 4, p. 178, 2010.
- [15] A. Dentel and U. J. D. P. K. H. Dietrich, HafenCity Universität, Institut für Energie und Gebäude, "Thermische Behaglichkeit-Komfort in Gebäuden," 2013.
- [16] B. W. Olesen, K. J. E. Parsons, and buildings, "Introduction to thermal comfort standards and to the proposed new version of EN ISO 7730," vol. 34, no. 6, pp. 537-548, 2002.
- [17] M. Laakso and A. J. I. Kiviniemi, "The IFC standard: A review of history, development, and standardization, information technology," vol. 17, no. 9, pp. 134-161, 2012.
- [18] T. J. B. I. Liebich, "IFC4—The new buildingSMART standard," 2013.

-
- [19] V. Bazjanac and D. B. Crawley, "The implementation of industry foundation classes in simulation tools for the building industry," 1997.
- [20] I. 10303-11, "Industrial Automation Systems and Integration--Product Data Representation and Exchange--Part 11: Description Methods: The EXPRESS Language Reference Manual," ed: International Organisation for Standardisation Geneva, 1994.
- [21] V. Nasyrov, "Building Information Models als Input für energetische Gebäudesimulation," Dissertation, Technische Universität München, München, 2013.
- [22] E. Tauscher, E. Mikulakova, K. Beucke, and M. König, "Automated generation of construction schedules based on the IFC object model," in *Computing in Civil Engineering (2009)*, 2009, pp. 666-675.
- [23] M. Lin and S. Gottschalk, "Collision detection between geometric models: A survey," in *Proc. of IMA conference on mathematics of surfaces*, 1998, vol. 1, pp. 602-608.
- [24] R. Jansohn, "Geometry Modeling And Visualisation Using SciPy And python-OCC r0.10," 2010.
- [25] H. J. L. j. Talbot, "wxPython, a GUI Toolkit," vol. 2000, no. 74es, p. 5, 2000.

Technischer Anhang

Eine CD ist als technische Anhang abgegeben. Die Dateien sind:

Nr.	Name
1	requirement.txt
2	GUI.py
3	überprüfung.py
4	Boolean_test.py
5	Aufbau_einer_Wand.py
6	CS_1.ifc

Notwendige Python-Bibliotheken:

Name	Version
ifcopenshell	0.5.0-dev
pythonocc	0.18.1
jupyter	1.0.0
wxPython	4.0.3

Erklärung zur Anfertigung der Diplomarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt und ist auch noch nicht veröffentlicht worden.

Dresden, den 08. Oktober 2018

A handwritten signature in black ink, appearing to read 'Yifan Lu'.

Yifan Lu