



Erarbeiten eines generischen Leitfadens für den geometrischen Export von Gebäudeelementen in IFC

Diplomarbeit

Bearbeiter: Simone Maria Lang

Betreuer: Dipl.-Arch. Romy Guruz

Dipl.-Math. Robert Schülbe

Verantwortlicher

Hochschullehrer: Prof. Dr.-Ing. Raimar Scherer

Dresden, den 20. März 2017

Inhaltsverzeichnis

Abkürzungsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	IV
1 Einleitung / Ziele	1
1.1 Einleitung	1
1.2 Ziel der Arbeit	1
1.3 Aufbau der Arbeit	1
2 Grundlagen	3
2.1 Einblick in Grundelemente der Geometrie	5
2.2 Einblick in die Computergrafik	7
2.3 Industry Foundation Classes	10
3 Geometrische Repräsentation mit IFC4	15
3.1 Allgemeine Struktur der geometrischen Repräsentation	16
3.2 Betrachtung des geometrischen Export in IFC4	18
3.2.1 Wand	20
3.2.2 Geschossdecke	29
3.2.3 Stütze	30
3.2.4 Körpermodell	32
3.3 Auswertung der genutzten Repräsentationen	34
3.3.1 Axis	35
3.3.2 SweptSolid	37
3.3.3 Clipping	39
3.3.4 BRep	40
4 Transformationen	41
4.1 Analyse der Transformationseigenschaft	41
4.2 Transformationsentwürfe	42
4.2.1 Transformation der SweptArea	44
4.2.2 Transformation der SweptSolid-Geometrie	49
5 Generischer Leitfaden	53
5.1 Überblick	53
5.2 Eignung der Darstellungsformen	53
5.3 Empfehlungen für Absprachen	55
6 Ergebnisse	57
6.1 Zusammenfassung und Diskussion	57
6.2 Ausblick	59

Literatur	60
A Anhang	62
A.1 Geometrische Repräsentationen in IFC4	62
A.2 STEP-Files	64
A.2.1 Wand	64
A.2.2 Geschossdecke	75
A.2.3 Stütze	77
A.2.4 Körpermodell	80
A.3 Parametrische Profile	83
A.4 Flussdiagramme	86

Abkürzungsverzeichnis

BEPs	Building Execution Plans
BIM	Building Information Modelling
BRep	Boundary Representation
CSG	Constructive Solid Geometry
DTV	Design Transfer View
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
MVD	Model View Definition
NURBS	Non-Uniform Rational BSplines
OOP	Objektorientierte Programmierung
RV	Reference View
SPF	STEP-Physical File
STEP	Standard for the Exchange of Product model data

Abbildungsverzeichnis

2.1	Darstellung einer Geometrie, welche unterschiedlich interpretierbar ist	3
2.2	Verbindungsmöglichkeiten von fünf Punkten	5
2.3	Elementare zweidimensionale Grundformen mit Krümmung	6
2.4	CSG-Verfahren zur Darstellung einer Wand mit zwei Öffnungen (Quelle: [Borrmann <i>et al.</i> , 2015], S.32)	9
2.5	Varianten der Verschiebegeometrie (Quelle: [Borrmann <i>et al.</i> , 2015], S.33)	9
2.6	Organisation des IFC-Datenmodells (Quelle: [Liebich <i>et al.</i> , 2015-2017])	11
3.1	Austauschszenario	16
3.2	Entitys zur Beschreibung der geometrischen Repräsentation einer <i>IfcProduct</i> -Instanz (Quelle: [Liebich <i>et al.</i> , 2015-2017])	17
3.3	Aufbau der Untersuchung des aktuellen geometrischen Exports mit IFC	18
3.4	Vordefiniertes Element <i>Wand</i>	20
3.5	Nach Verlinkung geordnete Ausschnitt aus einem STEP-File einer <i>Wand</i> mit der Referenzlinie auf der Innenseite (erstellt mit Allplan)	21
3.6	Beispiel der Nutzung von <i>IfcMaterialLayerSetUsage</i> für eine Standardwand (Quelle: [Liebich <i>et al.</i> , 2015-2017])	23
3.7	Vordefiniertes Element <i>Halbkreis Wand</i>	24
3.8	Definition einer gestutzten Kurve. <i>Trim1</i> gibt den ersten und <i>Trim2</i> den zweiten Punkt an. Über das <i>SenseAgreement</i> wird die Gangrichtung bestimmt (Quelle: [Liebich <i>et al.</i> , 2015-2017])	25
3.9	Vordefiniertes Element <i>Abgeschrägte Wand</i>	27
3.10	Vordefiniertes Element <i>Öffnung in einer Wand</i>	28
3.11	Vordefiniertes Element <i>Geschossdecke</i>	29
3.12	Vordefiniertes Element <i>Stütze</i>	31
3.13	Vordefiniertes Element <i>Körpermodell</i>	32
3.14	Informationsgehalt der Achsgeometrie	36
3.15	Informationsgehalt der SweptSolid-Geometrie. Das Attribut <i>SweptArea</i> ist vom Typ <i>IfcProfileDef</i>	38
3.16	Informationsgehalt der Clipping-Geometrie. Der <i>FirstOperant</i> ist immer ein Volumenkörper und der <i>SecondOperant</i> ein Halbraum	39
3.17	Informationsgehalt der BRep-Geometrie.	40
4.1	Beispiel einer abstrakten und gekürzten Transformierung	43
4.2	Auszug aus dem Flussdiagramm des Hauptprogramms <i>SweptArea</i> (vgl. Abb. A.24)	46
4.3	Flussdiagramm des Unterprogramms A zur Umformung einer <i>IfcRectangleProfileDef</i> in eine Polylinie	47
4.4	Flussdiagramm des Unterprogramms E zur Umformung von parametrischen Profilen in Teilkurven	48
4.5	Transformation eines Rechtecks in eine <i>IfcCompositeCurve</i>	49
4.6	Flussdiagramm des Hauptprogramms BRep	50
4.7	Erhaltene Punktmenge P_i aus der <i>SweptArea</i> und zu erzeugende Punktmenge P_i' durch Verschieben der Punktmenge	51

A.1	<i>Wand</i> -Export aus Allplan; Referenzlinie innen	64
A.2	<i>Wand</i> -Export aus Allplan; Referenzlinie außen	65
A.3	<i>Wand</i> -Export aus Allplan, Referenzlinie mitte	66
A.4	<i>Wand</i> -Export aus Revit, Referenzlinie mitte	67
A.5	<i>Halbkreiswand</i> -Export aus Allplan, Referenzlinie innen	68
A.6	<i>Halbkreiswand</i> -Export aus Revit, Referenzlinie innen, Teil 1	69
A.7	<i>Halbkreiswand</i> -Export aus Revit, Referenzlinie innen, Teil 2	70
A.8	<i>Abgeschrägte Wand</i> -Export aus Allplan, Referenzlinie innen	71
A.9	<i>Abgeschrägte Wand</i> -Export aus Revit, Referenzlinie innen, Teil 1	72
A.10	<i>Abgeschrägte Wand</i> -Export aus Revit, Referenzlinie innen, Teil 2	73
A.11	<i>Öffnung in Wand</i> -Export aus Allplan, Referenzlinie innen	73
A.12	<i>Öffnung in Wand</i> -Export aus Revit, Referenzlinie innen	74
A.13	<i>Geschossdecke</i> -Export aus Allplan	75
A.14	<i>Geschossdecke</i> -Export aus Revit	76
A.15	<i>Stütze</i> -Export aus Allplan	77
A.16	<i>Stütze</i> -Export aus Revit, Teil 1	78
A.17	<i>Stütze</i> -Export aus Revit, Teil 2	79
A.18	Körpermodell-Export aus Allplan	80
A.19	Körpermodell-Export aus Revit, Teil 1	81
A.20	Körpermodell-Export aus Revit, Teil 2	82
A.21	Parametrische Profile des Standards IFC4, Teil 1	83
A.22	Parametrische Profile des Standards IFC4, Teil 2	84
A.23	Parametrische Profile des Standards IFC4, Teil 3	85
A.24	Flussdiagramm des Hauptprogramms SweptArea	86
A.25	Flussdiagramm des Unterprogramms B zur Umformung einer <i>IfcCircleProfileDef</i> inn Kreis	87
A.26	Flussdiagramm des Unterprogramms C zur Umformung einer <i>IfcEllipseProfileDef</i> in eine Ellipse	87
A.27	Flussdiagramm des Unterprogramms D zur Umformung einer <i>IfcTrapeziumProfileDef</i> in eine Polylinie	87

Tabellenverzeichnis

3.1	SweptSolid Geometrie des vordefinierten Elementes <i>Wand</i>	22
3.2	Achsen Geometrie des vordefinierten Elementes <i>Wand</i> (Referenzlinie innen)	23
3.3	Achsgeometrie des vordefinierten Elementes <i>Halbkreis Wand</i>	25
3.4	SweptSolid-Geometrie des vordefinierten Elementes <i>Halbkreis Wand</i> . .	26
3.5	Geometrische Repräsentationen eines Halbraumes, benötigt für das vordefinierte Element <i>Abgeschrägte Wand</i>	27
3.6	SweptSolid-Geometrie einer Öffnung, benötigt für die vordefinierte <i>Öffnung in einer Wand</i>	29
3.7	SweptSolid-Geometrie des vordefinierten Elementes <i>Geschossdecke</i> . . .	30
3.8	Achsgeometrie des vordefinierten Elementes <i>Stütze</i>	31
3.9	SweptSolid-Geometrie des vordefinierten Elementes <i>Stütze</i>	31
3.10	Geometrische Repräsentationen einer Begrenzungsfläche, benötigt für das vordefinierte Element <i>Körpermodell</i>	33
3.11	Auflistung der verwendeten Klassen zur geometrischen Repräsentation; A = Allplan; R = Revit	34
3.12	Verwendete geometrische Repräsentationen während der Untersuchung .	35
3.13	Informationsgehalt der Möglichen Verbindungstypen	36
4.1	Benötigte Items zur alternativen Darstellung des jeweiligen Subtyps von <i>IfcParameterizedProfileDef</i> (* - direkte Subklasse)	45
4.2	Eingabewerte und Dateiausgabe der Unterprogramme Polyline und Circle	48
5.1	Übersicht über die gängigen geometrischen Repräsentationen	54
A.1	Übersicht über geometrische Repräsentationen des Standards IFC4, Teil 1 (vgl. [Liebich <i>et al.</i> , 2015-2017])	62
A.2	Übersicht über geometrische Repräsentationen des Standards IFC4, Teil 2 (vgl. [Liebich <i>et al.</i> , 2015-2017])	63

1 Einleitung / Ziele

1.1 Einleitung

Building Information Modelling (BIM) wird als Arbeitsmethode im Bauwesen angewandt. Planung samt Entwurf und Ausführung, als auch die Bewirtschaftung eines Gebäudes werden in einem digitalen BIM-Modell verknüpft. Der Datenaustausch zwischen den beteiligten Fachleuten soll damit grundlegend verbessert werden. Die gemeinsame Datenbasis wird automatisch synchronisiert und neben einer höheren Planungseffizienz werden Fehlerquellen, durch ständiges Einpflegen von Informationen, reduziert. Mit einem herstellerunabhängigen Datenformat, den Industry Foundation Classes (IFC), schaffte buildingSmart eine digitale Beschreibung der Bauwerksgeometrie, welche zwischen verschiedenen Softwareprodukten ausgetauscht werden kann. Für die digitale Modellierung stehen den Softwareherstellern mehrere Methoden zur Verfügung, einige davon werden durch das Datenformat unterstützt. Die Unterschiede im Informationsgehalt und ihrer Struktur können zu technischen Problemen während des Datenaustauschs führen.

1.2 Ziel der Arbeit

Der nicht immer reibungslose Datenaustausch in BIM-Projekten verzögert die flächendeckende Einführung der BIM-Methode. Ein Ansatz zur Regulierung des teilweise fehlerhaften Austauschs von digitalen Modellen ist das Abschließen eines Building Execution Plans (BEPs). Darin einigen sich die Vertragspartner eines gemeinsamen Projekts über den benötigten Projektumfang und dessen Ausführung.

Für den Datenaustausch, welcher mit den standardisierten IFCs erfolgt, sind vor allem Absprachen über die geometrischen Repräsentationen zu treffen. Die Vereinbarungen über die Darstellungsform können nur auf Basis von Informationen über das aktuelle IFC4-Format getroffen werden. Der aktuelle Stand des Datenformates muss auf Informationsgehalt, Variationen und Transformierbarkeit untersucht werden. Daraus soll ein generischer Leitfaden für die Erstellung eines BEPs abgeleitet werden, welcher Empfehlungen für Vereinbarungen bei Einsatz von IFC4 beinhaltet.

1.3 Aufbau der Arbeit

Kapitel 2 gibt einen Überblick zum Stand von Forschung und Technik. Es werden notwendige Fachbegriffe erklärt und definiert. Die Grundlagen werden benötigt um den geometrischen Export in IFC zu analysieren.

Kapitel 3 zeigt einige der geometrischen Repräsentationen in IFC4 anhand konkreter Exports aus zwei gängigen CAD-Programmen und erläutert die Darstellungsformen. Es wird gezeigt, welche geometrischen Informationen erzeugt werden und wie diese variieren können. Aus den identifizierten Informationsgehalten und Variationen werden in Kapitel 5 Empfehlungen für Absprachen abgeleitet.

Kapitel 4 betrachtet die Transformationseigenschaften der vorgestellten geometrischen Repräsentationen. Die Abhängigkeit von den Schnittstellen der CAD-Programme soll mit Transformationen innerhalb des Standards IFC4 überbrückt werden. Anhand von Beispielalgorithmen werden zwei Transformationsvorschläge detaillierter vorgestellt.

Kapitel 5 fasst den analysierten Stand von IFC4 und der darin enthaltenen Darstellungsformen aus den vorangegangenen Kapiteln zusammen. Zusätzlich wird ein Überblick für Beteiligte an BIM-Projekten gegeben. Darüber hinaus wird der Leser dabei unterstützt Entscheidungen über die Wahl und die Ausführung der geometrischen Repräsentation zu treffen.

Kapitel 6 fasst die Arbeit zusammen, diskutiert die Ergebnisse und gibt einen Ausblick auf weitere Arbeiten zum geometrischen Export in IFC4.

2 Grundlagen

Dem Menschen ist es seit jeher ein Bedürfnis mit Zeichnungen Informationen zu übermitteln, indem er auf zweidimensionalen Flächen dreidimensionale Objekte abbildet. Im weiteren Geschichtsverlauf wurden Methoden zum genauen Abbilden erforscht. Beispielsweise um Baupläne zu entwerfen, die eine dreidimensionale Nachbildung möglich machen. Grundsätzlich bestehen alle Darstellungen auf zweidimensionalen Flächen aus Punkten, Linien und Kurven. So auch die historisch überlieferten Baupläne seit dem Mittelalter (vgl. [Niedermaier & Bäck, 2015], S 25).

Dreidimensionales, in einer zweidimensionalen Fläche dargestellt, muss nicht immer eindeutig sein. Dies zeigt ein einfaches Beispiel: In Abbildung 2.1 ist ein Rechteck mit zwei festen Seitenlängen dargestellt. Diese Informationen erhält man beispielsweise, indem man auf der Fläche einen Koordinatenursprung und Koordinatenachsen festlegt. Die Schnittpunkte erhalten Koordinaten durch Längenangaben über die Position relativ zu den Achsen. Aus diesen Informationen ist die Länge der Seitenkanten des Rechtecks ermittelbar. Für einen Betrachter, der die Intension hinter der Abbildung nicht kennt, erschließt sich jedoch nicht zwingend, wie die Darstellung zu interpretieren ist. Ein Grund dafür ist, dass dieser Abbildung die Information fehlt, was dargestellt wird. Weiterhin kann die Abbildung unterschiedlich interpretiert werden, zum Beispiel als eine begrenzte Fläche, als Begrenzung einer Öffnung in einer Fläche oder als Frontansicht eines Hexaeders mit oder ohne Oberflächen.



Abbildung 2.1: Darstellung einer Geometrie, welche unterschiedlich interpretierbar ist

Dieses einfache Beispiel deckt sich mit der Aussage von Schiele, dass ein dreidimensionales Objekt nicht mit vollem Informationsgehalt in einer Ebene darstellbar ist und daher ein Gesamteindruck nicht in zwei Dimensionen vermittelt werden kann. Um alle Informationen eines Objekts abzubilden sei stets ein maßstäbliches Modell nötig (vgl. [Schiele, 2012], S. 4). Diese Ansicht gleicht auch der Begriffsdefinition nach Shum:

“Ein Produktmodell ist ein Informationsmodellschema, das alle relevanten Informationen eines Produkts abbildet. Der abbildbare Informationsgehalt soll dabei das reale Produkt hinreichend genau präsentieren, um Rückschlüsse auf die realen Produkteigenschaften ziehen zu können.” (Shum 1993, zit. In: [Leemhuis, 2005], S.9f)

Das Modell muss demnach dieselbe Form, welche dem eigentlichen Objekt bzw. Produkt innewohnt, beschreiben. Wichtige geometrische Elemente, welche Mittel zur Darstellung von Formen sind, werden in Abschnitt 2.1 näher betrachtet und für ein grundlegendes Verständnis angestrebt.

Die 60er Jahre werden meist als Beginn der Modellierung mithilfe von Computern genannt (vgl. [Mortenson, 1997], S. 1; [Schiele, 2012], S. 5) in denen erste Forschungsprojekte in diesem Bereich entstanden. Die ersten CAD-Programme kamen mit den ersten leistungsfähigen Computern in den 80er Jahren auf den Markt und wurden ab den 90er Jahren flächendeckend zur Planerstellung genutzt (vgl. [Hausknecht & Liebich, 2016], S. 35; [Niedermaier & Bäck, 2015], S. 26f). Nach und nach gab es Überlegungen Projekte objektorientiert dreidimensional zu modellieren. In Abschnitt 2.2 werden einige Methoden zur Darstellung der 3D-Objekte betrachtet, welche in der Computergrafik geläufig sind.

An Bauprojekten sind mittlerweile viele verschiedene Fachdisziplinen beteiligt, welche eigens dafür entwickelte Software nutzen. Die beteiligten Parteien stehen im ständigen Kontakt um sich abzustimmen und das gemeinsame Projekt zu bewältigen. Dies erfordert einen regen Austausch an Informationen zwischen den Planern, meist durch technische Zeichnungen in grafischer Form.

In jedem weiteren Planungsschritt müssen die neusten Informationen in die Arbeit eingepflegt und auf Kompatibilität und Konsistenz geprüft werden. Durch Änderungen anderer Parteien können dabei notwendige Veränderungen in der eigenen Arbeit notwendig werden. Wenn zum Beispiel nachträglich für Versorgungsinstallationen ein zusätzlicher Durchbruch in eine Geschossplatte hinzugefügt werden muss, dann würde dies zu einer Neuberechnung der Statik, mit einem gegebenenfalls neuen Bewehrungsplan, der Geschossdecke führen. Die daraus resultierende erneute Eingabe und Änderung in die jeweilige Software führt zu einem erhöhten Risiko des Informationsverlustes.

Leider sind durch den Einsatz unterschiedlicher Softwareprodukte für die unterschiedlichen Aufgaben am Bau stets Neueingaben nötig (vgl. [Borrmann *et al.*, 2015], S, 78). Um diesen Verlust von Informationen zu verringern und Fehler zu vermeiden gab es ab den 90er Jahren erste Bestrebungen. Für die verlustfreie Weitergabe der Informationen muss Interoperabilität geschaffen werden. Das ist die Fähigkeit der nahtlosen Zusammenarbeit verschiedener Softwarelösungen von verschiedenen Herstellern. Es kam die Idee auf mit vorhanden Standards ein neues offenes Dateiformat zu entwerfen, das von allen am Bau Beteiligten zum Austausch verwendet werden kann.

In den 80er Jahren entstand, durch Bestrebungen der Autoindustrie, ein standardisierte Rahmenwerk Standard for the Exchange of Product model data (STEP) um geometrische Daten auszutauschen. Im Jahr 1995 wurde die International Alliance for Interoperability (IAI) gegründet, besser bekannt unter dem neuen Namen *buildingSmart*, um die Einführung eines standardisierten Formats, basierend auf dem vorhandenen STEP, effizient voranzubringen. Einer der führenden Beteiligten an dieser Initiative war die Firma Autodesk. Mit dem Formatnamen Industry Foundation Classes (IFC) wurde 1997 die erste Version IFC 1.0 veröffentlicht.

IFC “[...] is an object-oriented data schema based on class definitions representing the objects (such as building elements, spaces, properties, shapes, etc.) that are used by different software applications used in construction or facility management project.” ([bui, 2008-2017])

Mit dem IFC Format konnten nun geometrische und semantische Beschreibungen von bauspezifischen Objekten zwischen verschiedenen Anwendungen ausgetauscht werden, indem dafür Schnittstellen in den Programmen implementiert wurden. Seitdem wird das Format stetig erweitert und umfangreicher. Die momentan am weitesten verbreitete Version ist IFC2x3, welche im Februar 2006 veröffentlicht wurde. Diese wird aktuell durch die neue Releaseversion mit der Bezeichnung IFC4 abgelöst. Die Schnittstellen in den heutigen CAD-Programmen werden überarbeitet und somit IFC4 kompatibel (Stand 2015, [bui, 2008-2017]). IFC4 ist eine Erweiterung und enthält modifizierte und erweiterte Klassen, welche auf alten Klassen des IFC2x3 Formats aufbauen. Es ist daher möglich, dass einige Softwarepakete teilweise noch IFC4 kompatibel sein können. Der objektorientierte Aufbau des Formates IFC und das Rahmenwerk STEP bilden den Abschluss dieses Kapitels (Abschnitt 2.3).

2.1 Einblick in Grundelemente der Geometrie

Die wichtigsten Elemente in der Geometrie sind Punkte und Geraden. Im Rechteck-Beispiel (Abb. 2.1) wurde die Nutzung eines Koordinatensystems für eine eindeutige Positionierung von Punkten in einem geometrischen Raum erwähnt. In der Computergrafik wird ein kartesisches XYZ-Koordinatensystem verwendet, welches sich durch eine immer gleiche Anordnung der Achsen auszeichnet. Die Achsen stehen rechtwinklig zueinander und “[...] wenn bei der Drehung der Achse X nach Y die Drehung um $+Z$ im Sinne einer rechtsdrehenden Schraube erfolgt [...]” ([Schiele, 2012], S.81) liegt dieses System vor.

Punkte sind durch die Koordinaten x, y und z als Tupel $P = (x \ y \ z)$ in einem kartesischen dreidimensionalen System definiert. Zwei Punkte P und Q bilden, geradlinig verbunden, eine Strecke \overline{PQ} . Jede Strecke liegt auf einer unendlich langen Trägergeraden g . Aus n Punkten können Polygone und Polygonzüge erstellt werden, wie in Abb. 2.2 mit fünf Punkten verdeutlicht ist. Das Polygon ist ein Polygonzug, der in sich geschlossen ist und so eine Fläche umschließt. Dreiecke und Vierecke zählen zum Beispiel auch zu den Polygonen.

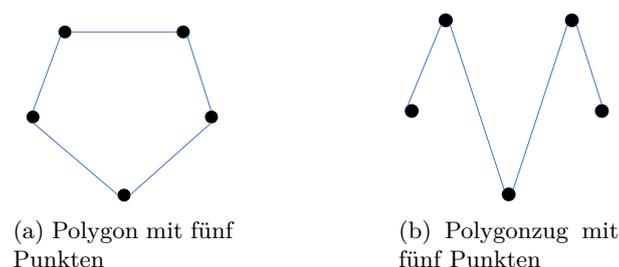
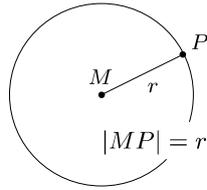
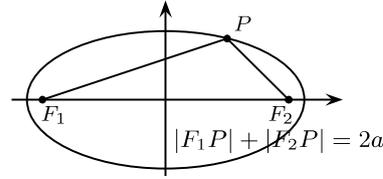


Abbildung 2.2: Verbindungsmöglichkeiten von fünf Punkten

Kreise und Ellipsen sind Grundformen in der Zeichenebene, die nicht aus Geraden bestehen, siehe dazu Abb. 2.3. Ein Kreis ist durch einen Mittelpunkt M und einem Abstand r der Kreispunkte P zu M , auch als Radius bezeichnet, festgelegt. Die Ellipse



(a) Definition eines Kreises (Quelle: [Merziger *et al.*, 2010], S.22)



(b) Definition einer Ellipse (Quelle: [Merziger *et al.*, 2010], S.23)

Abbildung 2.3: Elementare zweidimensionale Grundformen mit Krümmung

ist eine Menge von Punkten die von zwei Brennpunkten F_1 und F_2 eine konstante Abstandssumme ($|F_1P| + |F_2P| = 2a$) haben.

Im dreidimensionalen Raum sind auch Ebenen, zusätzlich zu Punkten und Geraden, zu den Bausteinen der Geometrie zu zählen. Eine Ebene ist eine unendliche, nicht gekrümmte zweidimensionale Fläche im Raum und kann auch als Punktmenge aufgefasst werden. Mit einer Ebene kann außerdem ein Halbraum beschrieben werden, indem der unendlich große Raum an einer Ebene geteilt wird. Alle Punkte auf einer Seite der Ebene gehören zu dem Halbraum. Die Punkte auf der anderen Seite jedoch nicht.

Durch mehrere begrenzte Flächen im Raum, die gemeinsame Außenkanten besitzen, können Körper gebildet werden. Mit geraden Flächen können so zum Beispiel Tetraeder, Quader, Prismen und Pyramiden gebildet werden. Körper können auch durch verschieben begrenzter Flächen erzeugt werden. Bewegt man beispielsweise einen Kreis geradlinig durch den Raum, entsteht entlang der Bewegung ein Zylinder.

Neben dem Kreis und der Ellipse existieren in der Geometrie weitere Kurven mit Krümmungen im zwei- sowie dreidimensionalen Raum. Feiformkurven sind beliebig formbare Kurven und werden auch als Splines bezeichnet. Generell sind Freiformkurven aus Polynomen zusammengesetzt und werden mit einer der folgenden Stetigkeitsbedingungen an den Nahtstellen geglättet (vgl. [Schiele, 2012], S.44f ; [Borrmann *et al.*, 2015]37f):

Punktstetigkeit	der Endpunkt der einen Kurve, ist gleich dem Startpunkt der anderen Kurve
Tangentiale Stetigkeit	am Berührungspunkt besitzen beide Kurven die gleiche Tangente
Krümmungsstetigkeit	am Berührungspunkt besitzen beide Kurven die gleiche Tangente, sowie den gleichen Krümmungsradius

Die Raumkoordinaten der Kurven werden durch Funktionen, die von einem gemeinsamen Parameter abhängen bestimmt. Besonders erwähnenswerte Freiformkurven sind Beziér-Kurven, B-Splines und Non-Uniform Rational BSplines (NURBS). Diese drei Kurvenarten beschreiben Kurven, die sich einer Menge von Kontrollpunkten annähern, jedoch nur den jeweils ersten und letzten Punkt tatsächlich durchlaufen. Die Kontrollpunkte beeinflussen die Krümmung und die Länge der Kurven durch ihre Lage.

Die Kurven schmiegen sich im Anfangs- und im Endpunkt dem Polygonzug aus den Kontrollpunkten an. Der Unterschied dieser drei Kurven ist die charakteristische Basisfunktion, welche mit den Kontrollpunkten multipliziert wird.

Bézierkurven besitzen Polynome als Basiskurven, welche alle Kontrollpunkte beinhalten. Wenn viele Kontrollpunkte gesetzt sind wird das Polynom und damit die Kurve immer hochgradiger. Die Kurven werden global stark von den einzelnen Punkten beeinflusst, obwohl diese nur lokal von Bedeutung sind.

Die B-Splines werden in Teilintervallen berechnet, in denen je ein Polynom, meist mit dem Grad 3, definiert wird. An den Nahtstellen der einzelnen Polynome müssen die Stetigkeitsbedingungen gelten. Einzelne Veränderungen der Kontrollpunkte bleiben so im Teilintervall und beeinflussen nicht die gesamte Kurve.

Die NURBS bzw. deren Basisfunktion basiert auf den B-Splines. Der Verlauf der NURBS ist zusätzlich zu den Kontrollpunkten beeinflussbar, indem den Punkten Gewichtungen zugewiesen werden. Mit diesen Kurven können auch Kreise und Ellipsen beschrieben werden, was mit den ersten Beiden nicht möglich ist.

Mit diesen drei Freiformkurven können auch Freiformflächen beschrieben werden. Die Beschreibung der Kurven wird dafür um eine Dimension und einen zweiten Parameter erweitert. Die Mächtigkeit der Flächendarstellung ist abhängig von der benutzten Freiformkurve. Daher sind NURBS-Flächen am flexibelsten. Mit ihnen können auch Kugel- und Zylinderoberflächen modelliert werden.

2.2 Einblick in die Computergrafik

Für die Darstellung von Objekten stehen in der Computergrafik verschiedene Methoden zur Verfügung. Es wird vor allem zwischen Oberflächenmodellen (expliziten Verfahren) und Volumenmodellen (impliziten Verfahren) unterschieden (vgl. [Schiele, 2012], S.41f). Diese können aber auch kombiniert angewendet werden. Konkret bedeutet das, dass Körper entweder über ihre Oberflächen und Ränder oder durch dreidimensionale geometrische Primitive beschrieben werden können. Unter Primitiven versteht man vorgegebene Grundkörper wie zum Beispiel Quader, Zylinder oder Kugel.

Ein wichtiger Baustein bei der Erstellung von geometrischen Modellen ist die Nutzung von Parametern. Diese erleichtern die Modellierung durch feste Abmessungen wie Breite, Länge, Höhe. Sie sind aber auch formgebend in Kurven. Einige Verfahren werden in diesem Abschnitt vorgestellt.

Boundary Representation Die Boundary Representation (BRep) oder auch Oberflächendarstellung gehört zu den am häufigsten verwendeten Geometrierepräsentationen (vgl. [Sabelka, 2001]; [Borrmann *et al.*, 2015], S.28). Aus hierarchisch geordneten Berandungselementen wird das Objekt, beginnend mit den Knoten, aufgebaut (vgl. [Schiele, 2012], S.53):

Schale/shell	eine Schale ist eine aus Flächen zusammen gesetzte, zusammenhängende Oberfläche
Fläche/facette	eine Fläche ist begrenzt durch n-Konturen, jedoch mindestens durch eine
Kontur/loop	eine Kontur besteht aus n-Kanten aneinandergereiht, die eine Schleife bilden
Kante/edge	eine Kante ist eine orientierte Gerade oder Kurve
Knoten/vertices	zwei Knoten, festgelegt durch Koordinaten, begrenzen zusammen je eine Kante

Mittels topologischer Informationen werden die Elemente miteinander in Beziehung gesetzt. In einer Datenbank wird zum Beispiel in Listen strukturiert, welche Beandlungselemente aus der nächst tieferen Hierachiestufe für das jeweilige Element genutzt werden. Um eine gültige Geometrie zu erhalten sind Randbedingungen zu berücksichtigen (vgl. [Sabelka, 2001]):

- Linien müssen mit genau zwei Knoten verknüpft sein
- Flächen dürfen zu maximal zwei Körpern gehören
- Konturen müssen in sich geschlossen sein
- Flächen und Schalen dürfen nur einen äußeren Rand besitzen

Mit dieser Methode lassen sich Objekte mit planaren Flächen und geraden Kanten beschreiben, wenn nur die niedrigste Hierarchiestufe geometrische Informationen besitzt. Komplexere Körper mit gekrümmten Kanten und Flächen benötigen zusätzliche geometrische Informationen, die den Verlauf der Kanten und die Form der Flächen beschreiben. Gerade für komplexere Körper kann die Datenstruktur sehr aufwendig und unübersichtlich werden und somit die Prüfung der Randbedingungen erschweren.

Eine einfache Form der BRep ist die Tesselation, welche meist zur Darstellung von Gelände, als Grundlage für numerische Simulationen oder zur Visualisierung genutzt wird (vgl. [Schiele, 2012], S.51; [Borrmann *et al.*, 2015] S.30). Mit der Tesselation können Freiformflächen durch ein Netz aus Dreiecken angenähert werden. Knoten, welche in der darzustellenden gekrümmten Fläche liegen, werden genutzt um Kanten und Flächen zu bilden, welche die tatsächliche Freiform skizzieren. Das Dreiecksnetz kann beliebig fein gewählt werden und bringt einen geringeren Speicheraufwand als die Repräsentation mit gekrümmten Flächen mit sich.

Constructive Solid Geometry Die Constructive Solid Geometry (CSG) oder auch Boolesches Geometriemodell greift bei der Darstellung auf Primitivkörper zurück. Diese werden mit den logischen Operatoren Vereinigung, Schnitt und Differenz verknüpft und bilden abschließend einen Körper wie beispielhaft in Abb. 2.4 dargestellt. In dem Konstruktionsbaum sind die einzelnen Schritte festgehalten, welche den Modellierungsvorgang abbilden. Mit dieser Methode lassen sich viele Körperformen erzeugen, unter anderem da die Primitive mit Parametern angepasst werden können.

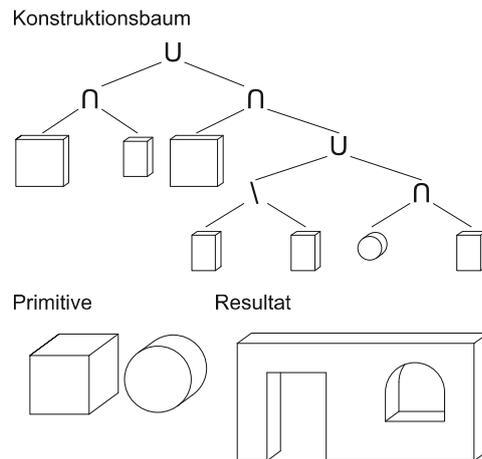


Abbildung 2.4: CSG-Verfahren zur Darstellung einer Wand mit zwei Öffnungen (Quelle: [Borrmann *et al.*, 2015], S.32)

Borrmann kritisiert jedoch den Zwang zur Nutzung der Primitivkörper, was einschränkend sei (vgl. [Borrmann *et al.*, 2015], S.31). Das pure CSG-Verfahren sei deswegen nur noch selten im Einsatz. Vielmehr findet die grundlegende Idee der Nutzung von booleschen Operatoren Anwendung. Das bedeutet, dass bereits erstellte Volumenkörper, welche nicht den Primitiven entsprechen, mit den Operatoren bearbeitet werden können.

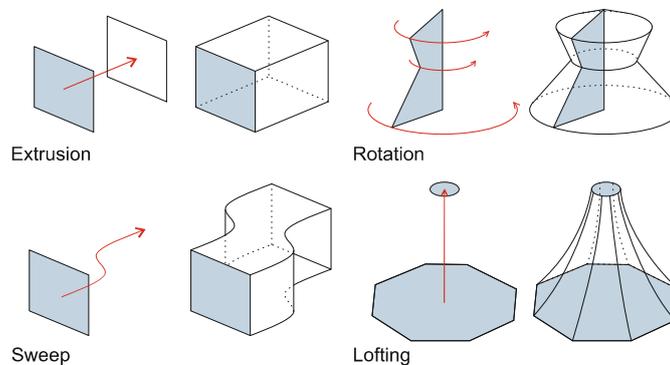


Abbildung 2.5: Varianten der Verschiebegeometrie (Quelle: [Borrmann *et al.*, 2015], S.33)

Verschiebegeometrie Die Verschiebegeometrie oder auch das Extrusions- und Rotationsverfahren setzt noch vor der Nutzung der Primitive für Volumenmodelle an. Der Körper wird bei dieser Methode durch Kombination mindestens einer Fläche und eines Pfades oder einer Achse erstellt.

In der Abb. 2.5 sind vier Varianten der Verschiebegeometrie skizziert. Der Pfad kann entweder geradlinig sein, dann spricht man von einer Extrusion oder durch eine Kurve beschrieben werden, dann spricht man von einem Sweep. Das Lofting wird durch mehrere Flächen erzeugt, die als Querschnitte des Körpers entlang des Pfades fungieren. Volumenkörper können auch durch Rotation um eine Achse erstellt werden.

2.3 Industry Foundation Classes

Die IFCs sind ein von buildingSmart standardisiertes Datenmodell (ISO 16739) um zwischen Mitwirkenden an Projekten der Bauausführung und des Gebäudemanagement Daten auszutauschen. Dieses Datenmodell beruht auf der objektorientierten Programmierung (OOP). Die für IFC wichtigen Elemente werden, im nachfolgenden Unterpunkt, erläutert. In [Gronau *et al.*, 2008-2016] und [Borrmann *et al.*, 2015] (S.45-52) konnten dazu Grundlagen entnommen werden.

In CAD-Programmen erstellte Modelle werden exportiert, indem die Informationen des Modells in die IFC-Datenbank übertragen werden. In einem der Unterpunkte wird der Aufbau des IFC-Datenmodells eingehender behandelt. Es entstehen nach der Vorlage der standardisierten Klassen aus dem Datenmodell die konkreten Objekte. Die Instanzen werden in eine Datei im STEP-Physical File (SPF) Format geschrieben und können dann zwischen den Anwendern verschiedener Anwendungen ausgetauscht werden. Die Projektbeteiligten können diese Datei wiederum in ihre Software importieren um an dem Projekt zu arbeiten. Das Format wird in einem nachfolgenden Unterpunkt beschrieben.

Die IFCs erzeugen ein Gebäudemodell das viele Fachbereiche abdeckt (vgl. Domain Layer in Abb. 2.6). Das gesamte Modell wird meist nicht benötigt, sondern eher Teilm Modelle daraus. Dies kann an der speziellen Fachrichtung oder an der Bauphase liegen. Dazu kommt, dass Informationen durch IFC auf unterschiedliche Weise dargestellt werden können. Aus diesem Grund entwickelte buildingSmart die Model View Definitions (MVDs), welche festlegen, was und wie es übergeben werden soll.

Die Ansichtsdefinitionen bilden eine Anleitung für die Modellerstellung mit IFC für bestimmte Anwendungsbereiche und zeigen auf, welche Informationen erlaubt sind und welche benötigt werden. Diese werden in den Schnittstellen der Softwareprodukte eingebaut und beeinflussen den Export. Der abschließende Unterpunkt dieses Kapitels widmet sich daher den aktuellen MVDs.

Objektorientierte Programmierung Zentrales Konzept der OOP ist das Objekt. Jedes Objekt besitzt bestimmte Eigenschaften bzw. Attribute und bietet Funktionen bzw. Methoden. Die Methoden sind kein Bestandteil des IFC-Datenmodells und werden daher ausgelassen. Ein Objekt kann auch aus anderen Objekten bestehen.

Für jeden Objekttyp gibt es eine Klasse, welche als Schablone zur Erzeugung bzw. Instanziierung genutzt wird. Die Klasse legt fest, wie der jeweilige Objekttyp aufgebaut ist. Das heißt, es können unterschiedlich ausgeprägte Objekte des gleichen Objekttyps erstellt werden.

Eine Klasse kann von einer anderen Klasse erben, dann ist sie eine Subklasse und erbt von ihrer Superklasse. Erben bedeutet in der objektorientierten Modellierung, dass die Subklasse sämtliche Eigenschaften der Superklasse besitzt. Neben den geerbten Eigenschaften können die Subklassen weitere Eigenschaften tragen. Ein spezieller Fall der Superklasse ist die abstrakte Klasse. Diese bietet keine Instanziierung von Objekten,

sondern steht allgemeiner nur zur Vererbung einer Gruppe von Subklassen zur Verfügung.

Die Attribute werden in der Klasse durch einen Namen und einen erforderlichen Datentyp definiert und bei der Instanziierung erhält die Instanz einen Attributwert des vorgegebenen Datentyps. Die Instanzen können auch miteinander in Beziehung stehen, dargestellt durch Assoziationen. Dabei kann festgelegt werden wie viele Objekte einem Objekt unter der Relation zugeordnet werden müssen. Dies kann jedoch auch offengelassen werden. Die Assoziation wird als Attribut eines Objekts hergestellt. Entweder wird direkt auf die Klasse eines anderen Objekts verwiesen oder auf eine Assoziationsklasse, welche Beziehungen als Instanzen erstellt.

IFC-Datenmodell Das Datenmodell besteht aus vier Schichten die aufeinander aufbauen, siehe Abb. 2.6. Detaillierte Informationen wurden der Dokumentation von buildingSmart entnommen (vgl. [Liebich *et al.*, 2015-2017]). Die Elemente in den Schichten können nur auf die darunterliegenden Schichten verweisen. Umgekehrt ist dies nicht möglich.

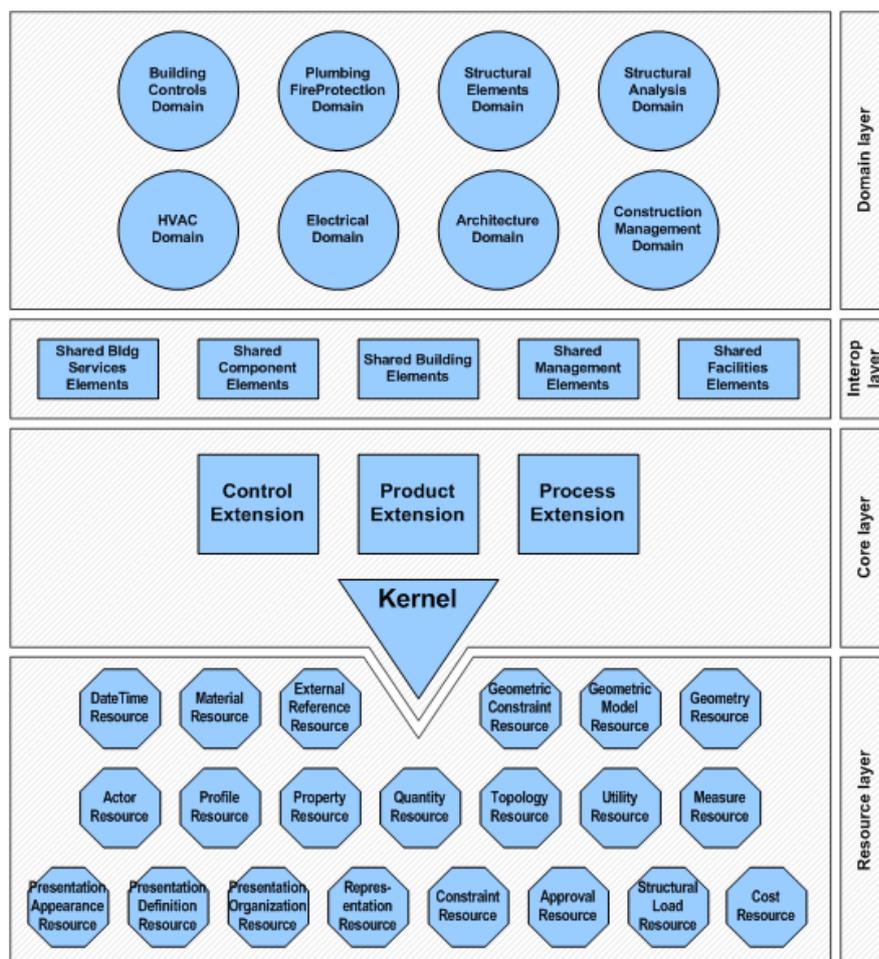


Abbildung 2.6: Organisation des IFC-Datenmodells (Quelle: [Liebich *et al.*, 2015-2017])

Der Core Layer bildet den Kern des Datenmodells und legt die Basisstrukturen, grundlegende Beziehungen und allgemeinen Konzepte fest. Darin enthalten ist im Kernel-

Schema die abstrakte generelle Basisklasse *IfcRoot* von der alle Klassen in der gleichen und darüber liegenden Schichten abstammen. Diese Klasse vererbt folgende Attribute: eine eindeutige Identifikation (in Form einer globalen ID), ein Name, eine Beschreibung und historische Daten. Lediglich die ID ist bei der Instanziierung obligatorisch. Die drei direkten abstrakten Subklassen von *IfcRoot* teilen alle abstammenden Klassen in Objekte, Beziehungen oder Eigenschaften ein.

Die auf dem Kernel-Schema aufbauenden Schemata beschreiben physische und räumliche Objekte eines Gebäudes, sowie deren Beziehung (Product Extension), Prozesse und Abläufe (Process Extension) und Steuerungsobjekte (Control Extension).

Im Interop, oder auch Shared Layer, werden die vom Core Layer abgeleiteten Klassen weiter spezialisiert. Im Domain Layer werden die weiter spezialisierten Klassen in Domänen sortiert, welche jeweils für eine Fachdisziplin stehen.

Die unterste Schicht, der Resource Layer, beinhaltet grundlegende Datenstrukturen, welche von den darüber liegenden Schichten verwendet werden. Alle Klassen des Resource Layers können nur initiiert werden, wenn sie von Objekten aus den drei oberen Schichten referenziert werden. Da sie selbst nicht von *IfcRoot* abstammen, besitzen sie keine ID, also keine Identität. Für die Repräsentation von Objekten sind grundlegend das Geometric Model Resource, das Geometry Resource und das Topology Resource Schema nötig. Darin enthalten sind die von IFC unterstützten geometrischen Repräsentationen von Körpern, die dafür benötigten Elemente, sowie topologiebildende Klassen. Auf die genauen geometrischen Repräsentationen durch das IFC4-Format wird in Abschnitt 3.1 näher eingegangen.

STEP-Physical File Der Aufbau eines SPF ist stets in zwei Teile unterteilt. Die Autoren R. Aderl und D. Trippner stellen den Aufbau des Standards übersichtlich dar (vgl. [Aderl, 2000], S.87-89). In der ersten Zeile der Datei steht die Norm *ISO-10303-21*, welche verwendet wird, und eröffnet das Dokument. Anschließend wird die Header Section mit *HEADER* geöffnet und durch *ENDSEC* geschlossen. Diese Sequenz steht dem Autor der Datei frei zur Angabe von Zusatzinformationen. Diese sind nicht der Norm unterworfen. Nach dem Kopf der Datei beginnt die Beschreibung des Modells in der Data Section durch eine Auflistung der Instanzen. Die Instanziierung wird durch *DATA* geöffnet und am Ende durch *ENDSEC* geschlossen. Die Datei wird durch *END-ISO-10303-21* beendet. Das STEP-File wird komplett in Großbuchstaben verfasst.

Jede Instanz bekommt eine neue Zeile und erhält einen eindeutigen Identifikator, bestehend aus einer “#” gefolgt von einer Zahl. Die Beschreibung der Instanzen besteht aus dem Klassennamen und Parametern. Die Parameter können zum Beispiel Zahlenwerte, Buchstabenfolgen oder andere Objekte sein. Nicht immer muss jeder Parameter auch einen Wert besitzen, stattdessen kann ein “\$” als Platzhalter eingesetzt werden. Die Bedeutung der Attribute ist in der Datei nicht hinterlegt, sondern ist in der Definition des Datenmodells und der jeweiligen Klasse nachzuschlagen. Eine Übersicht zur Interpretation der Klassen bietet zum Beispiel STEP Tools, Inc. (vgl. [Ste, 2015]) oder die Dokumentation von buildingSmart (vgl. [Liebich *et al.*, 2015-2017]).

Im folgendem Auszug einer Data Section wird ein Kreis erzeugt:

```
#361=IFCCIRCLE(#356,5000.);  
...  
#356=IFCAXIS2PLACEMENT2D(#357,#359);
```

Das Objekt #361, als Entität der Klasse *IfcCircle*, hat die Attribute *Position= #356* und *Radius= 5000*. Das Attribut *Position* wird durch eine weitere Instanz vom Typ *IfcAxis2Placement* befüllt. Diese besitzt wiederum zwei Attributen, welche durch weitere Instanzen realisiert werden (vgl. Objekt #356). Das Beispiel repräsentiert einen Kreis mit einem Radius von 5m und einer Position definiert durch die Instanz #356.

Model View Definition Aktuell gibt es im IFC4 Release zwei offizielle MVDs von *buildingSmart*: die Reference View (RV) und die Design Transfer View (DTV).

Die RV ist hauptsächlich für den einseitig gerichteten Austausch gedacht. Es sollen verschiedene Fach- und Teilmodelle zusammengeführt werden, um Kollisionen zwischen Geometrien zu erkennen. Änderungen in der Geometrie sind nicht vorgesehen. Im Fall von Änderungsanfragen muss sich an den Autor gewendet werden, welcher dann eine abgeändertes IFC-Modell erstellt.

Um möglichst kompatibel zu vielen Anwendungen zu sein, soll die geometrische Repräsentation so einfach wie möglich gehalten werden. Das Schema der RV soll zudem so kompakt wie möglich sein, um das Herunterladen auf Mobilgeräten mit geringer Bandbreite zu ermöglichen. Es wird daher von *buildingSmart* empfohlen sich auf Tessellation und einfache Swept Solids zu beschränken. Es sollen genaue, nicht-parametrische geometrische Repräsentation erstellt werden. Daher sind folgende Geometrien in der View erlaubt:

- Body Tessellation Geometry
- Body SweptSolid Geometry
- Body AdvancedSweptSolid Geometry

Diese drei werden in Abschnitt 3.1 näher erläutert. Weitere geometrische Modelle sind nicht im IFC4 RV enthalten, insbesondere Boolesche Operationen, welche für CSG benutzt werden.

Die DTV ist eine Erweiterung der RV. Schnittstellen welche die DTV importieren können, können daher auch die RV korrekt importieren. Andersherum gilt das jedoch nicht. Änderungen an der Geometrie sind in dieser View eingeplant. Nach dem Import können Änderungen gemacht werden.

Die DTV macht Gemeinschaftsarbeiten von verschiedenen Disziplinen möglich, kann aber auch genutzt werden, um Designmodelle auszutauschen. Für den Austausch einer komplexeren Geometrie mit Parametern werden in dieser View folgende geometrische Repräsentationen zusätzlich zu denen der RV verwendet:

- Body BRep Geometry

- Body Advanced BRep Geometry
- Body SurfaceModel Geometry
- Body CSG Geometry
- Body AdvancedSweptSolid Geometry

Diese geometrischen Repräsentationen werden ebenfalls im Abschnitt 3.1 näher erläutert.

3 Geometrische Repräsentation mit IFC4

Die Literaturrecherche zum Export in IFC zeigt, dass mit der Weiterentwicklung des IFC-Formats stets auch der Austausch mit diesem geprüft wurde. Meist bestanden die Untersuchungen aus dem Import von IFC-Modellen in verschiedene Programme mit anschließendem Export des unveränderten Modells. H. Ma, K.M.E. Ha, C.K.J. Chung und R. Amor veröffentlichten 2006 eine Arbeit, in der sie das ursprüngliche IFC-Modell mit dem IFC-Modell nach Im- und Export vergleichen. Die Autoren erklären, dass zwar die IFC-Dateien einem Standard folgen, die CAD-Systeme aber intern ihre eigenen Repräsentationen nutzen (vgl. [Ma *et al.*, 2006], S.1217). Objekte werden beim Importvorgang in die eigene Repräsentation übersetzt und während des Exportvorgangs wieder in standardisierte IFC übertragen. Obwohl bei unveränderten Modellen keine Informationen verloren gehen sollten, ist laut den Autoren eine 1:1 Übertragung, aufgrund von sich unterscheidenden Repräsentationen, in diesem Szenario nicht möglich (vgl. [Ma *et al.*, 2006], S.1217). In ihrer Arbeit wird aufgezeigt, dass nach einem Im- und Export Informationen verändert, entfernt oder hinzugefügt werden können. Der Verlust, herbeigeführt zum Beispiel durch das Entfernen von Trägern während des Exports, wurde dabei als erheblichste Änderung genannt (vgl. [Ma *et al.*, 2006], S.1226).

Žiga Turk und Tomaž Pazlar veröffentlichten 2008 eine ähnliche Untersuchung. Sie verglichen geometrische Exports aus drei gängigen Softwareprodukten. Dabei untersuchten sie jeweils ein Model aus jeder Software, exportierten und importierten diese anschließend in alle genutzten Anwendungen. Auch in dieser Untersuchung wird eindeutig, dass gerade die Schnittstellen zu Veränderungen führen. Durch den Im- und Export sind in den Untersuchungen folgende Probleme aufgetreten:

- Entitäten wurden umgeschrieben, entfernt, ersetzt oder hinzugefügt
- Umwandlungen in andere geometrische Repräsentationen

(vgl. [Pazlar & Turk, 2008], S.6-13). Der Austauschstandard kann folglich nur so gut sein, wie er letztendlich von den Softwareherstellern in den Im- und Exportschnittstellen Einzug findet. Dort gibt es offensichtlich noch Mängel, die zu Fehlern und Verlusten beim Austausch führen, obwohl die Version 4 der IFC als bereits weitgehend ausgereift gilt (vgl. [Borrmann *et al.*, 2015], S.80).

Die tatsächliche Qualität der geometrischen Repräsentation der aktuellen Version IFC4 bildet in dieser Arbeit das zu untersuchende Kernthema. Zur Einführung wird im Abschnitt 3.1 der Aufbau der geometrischen Repräsentation in IFC4 nachvollzogen. Die Informationen darin entstammen der Dokumentation des Standards von buildingSmart (vgl. [Liebich *et al.*, 2015-2017]).

Für ein Verbessern des Datenaustauschs, unabhängig von der Software, sollte neben der Verbesserung der Schnittstellen auch die Weitergabe der Informationen betrachtet werden. Konkret sollte untersucht werden, welchen Informationsgehalt die geometrische Repräsentation besitzt und welche geometrischen Repräsentationen daraus möglichst

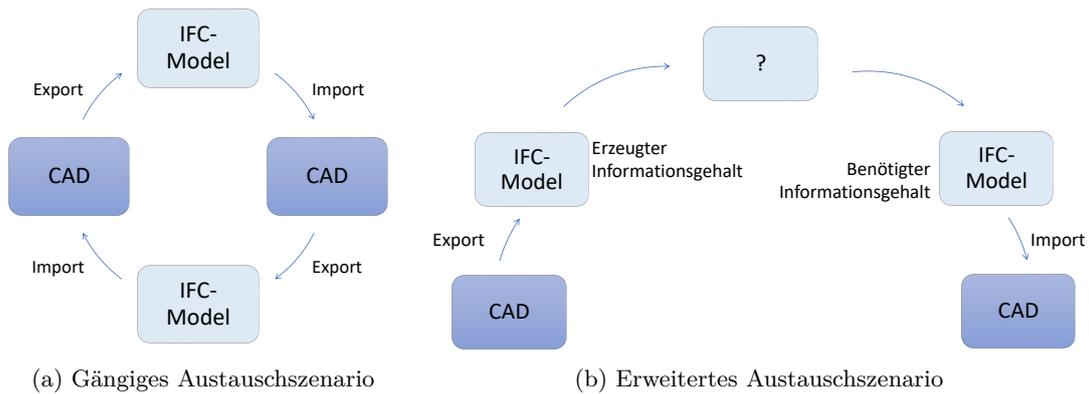


Abbildung 3.1: Austauschscenario

verlustfrei abgeleitet werden können. Die angesprochenen Arbeiten zeigen repräsentativ, dass in Austauschscenarien, wie in Abbildung 3.1a dargestellt, noch Weiterentwicklungen nötig sind. Da die Übertragung aller Informationen in die interne Struktur von Software nicht immer verlustfrei ist, können bereits durch den Import unterschiedliche Modelle in den Programmen entstehen.

Mit einem verkleinerten Szenario, wie in Abbildung 3.1b dargestellt, soll in dieser Arbeit ein besserer Überblick erreicht werden. Deshalb wird der Fokus dieser Arbeit vorrangig auf dem Export liegen. Ein Eliminieren der Importprobleme kann ermöglicht werden, indem Bauelemente modelliert und exportiert werden. In Abschnitt 3.2 wird der aktuelle Export anhand diesem verkleinerten Szenarios an konkreten Elementen untersucht. Der abschließende Abschnitt 3.3 erläutert die geometrischen Informationen der genutzten Repräsentationen innerhalb des IFC-Standards.

In Abbildung 3.1b ist angedeutet, dass für den Import ein benötigter Informationsgehalt eine Rolle spielt. Wird das IFC-Modell vor den Import-Vorgängen bearbeitet, indem die geometrische Repräsentation angepasst wird, könnte der Informationsverlust gegebenenfalls erheblich verringert werden. Das IFC-Modell kann der internen geometrischen Repräsentation des importierenden CAD-Programms angeglichen werden. Die Untersuchung des Exports in diesem Kapitel wird daher Grundlage für eine Auswahl möglicher Transformationen und die Prüfung auf Ausführbarkeit, welche in Kapitel 4 erfolgen.

3.1 Allgemeine Struktur der geometrischen Repräsentation

Alle Elemente in einem Modell, welche eine Geometrie besitzen, sind als *IfcProduct* zusammengefasst. Es gibt eine Vielzahl an Unterklassen, die unterschiedliche Elemente im Baubereich darstellen. Jedem *IfcProduct* wird eine eindeutige GlobalID, eine Platzierung und eine Repräsentation zugeordnet, siehe Abb. 3.2. Die Platzierung kann unterschiedlich erfolgen, aber läuft immer darauf hinaus, dass das Objekt im globalen Koordinatensystem positioniert ist. In der Zuordnung *IfcProductDefinitionShape* ist die geometrische Repräsentation des jeweiligen Objekts gespeichert. Es können einem Produkt mehrere Repräsentationen in Form von *IfcShapeRepresentation*-Instanzen zu-

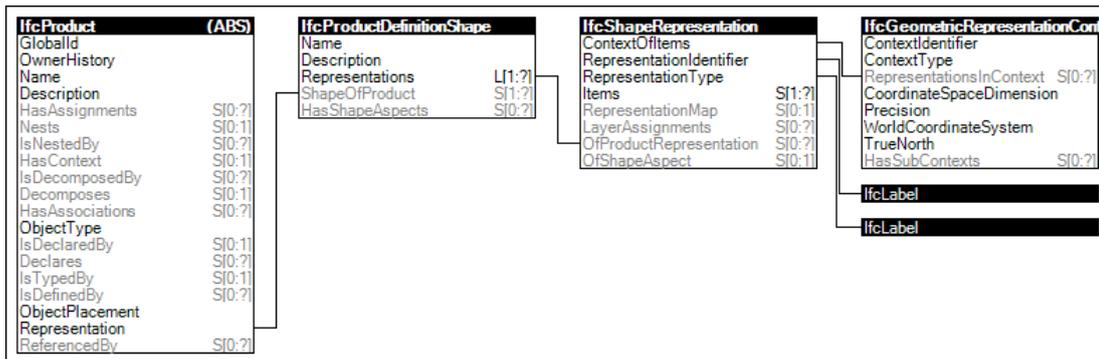


Abbildung 3.2: Entitäts zur Beschreibung der geometrischen Repräsentation einer *IfcProduct*-Instanz (Quelle: [Liebich *et al.*, 2015-2017])

geordnet werden. Dadurch können mehrere Repräsentationsarten des selben Objekts abgedeckt werden.

Die Geometrie selbst wird in dem Attribut vom Typ *IfcShapeRepresentation* durch Attribute, einem Identifier und einem Repräsentationstyp beschrieben. Der Identifier ist ein *IfcLabel*, der wiedergibt, auf welche Weise das Objekt zu repräsentieren ist. Zum Beispiel, ob das Objekt ein 3D-Körper oder eine Oberfläche im 3D-Raum ist. Der Repräsentationstyp ist ebenso ein *IfcLabel* und enthält den Namen der dazu gehörigen geometrischen Repräsentation. Dieser Typ gibt an, welche geometrische Repräsentation instanziiert wird.

Jede Repräsentation bekommt durch *IfcGeometricRepresentationContext* Zusatzinformationen. Darin sind unter anderem die Dimensionalität, der globale Nullpunkt, sowie der “wahre Norden”. Das Attribut *TrueNorth* gibt an, wie die lokale Y-Achse des Objekts, in Bezug zur globalen Y-Achse, liegt. Im referenzierten Subkontext *IfcGeometricRepresentationSubContext* ist außerdem noch hinterlegt, um welche Ansicht es sich handelt und ob eine Skalierung vorgenommen wird.

Die geometrische Repräsentation wird durch Repräsentationselemente, Instanzen des abstrakten Supertyps *IfcRepresentationItem*, aufgebaut. In der Dokumentation von IFC4 werden alle standardisierten geometrischen Repräsentationen definiert. Im Anhang sind diese einzusehen. Die Tabelle A.1 gibt einen Überblick über geometrischen Repräsentationen, welche keine Volumenkörper erzeugen. Alle Repräsentationen, welche Volumenkörper erzeugen, sind in der Tabelle A.2 gelistet. In diesen Tabellen ist jeweils der Identifier und der Typ der Repräsentation angegeben, sowie die zu verwendenden *IfcRepresentationItems*. In der Spalte “Kurze Erläuterung” wird beschrieben wie die Darstellung funktioniert und für welche geometrischen Elemente bzw. Körper sie eingesetzt werden kann.

Produkt Platzierung Das Konzept der relativen Platzierung wird durch die Klasse *IfcLocalPlacement* implementiert, indem ein übergeordnetes Koordinatensystem referenziert und darin ein lokales Koordinatensystem initiiert wird.

Durch Koordinaten als Attribute von *IfcCartesianPoint* und je einer horizontalen und

einer vertikalen Richtung, repräsentiert durch *IfcDirection*-Instanzen, wird im Allgemeinen ein lokales Koordinatensystem erzeugt. Das übergeordnete Koordinatensystem wird durch eine Instanz *IfcLocalPlacement* referenziert und kann zu einem Objekt gehören. Jedes Projekt besitzt ein globales Koordinatensystem, in welchem die Elemente des Projekts platziert werden. Sollte kein übergeordnetes Koordinatensystem angegeben sein, so wird automatisch das globale Koordinatensystem referenziert.

Eine Platzierung kann auch über Raster der Klasse *IfcGrid* erfolgen. Dabei werden an den Schnittpunkten der Rasterlinien die Objekte ausgerichtet. Diese Art der Platzierung wird in dieser Arbeit jedoch nicht genutzt, da sie nicht Teil der Geometrie der Körper ist und soll somit nicht weiter betrachtet werden.

3.2 Betrachtung des geometrischen Export in IFC4

Um eine Untersuchung durchzuführen, welche nur auf den Export konzentriert ist, werden die Gebäudeelemente in den gewählten Anwendungen in gleicher Weise modelliert. Dadurch kann der Import als Fehlerquelle ausgeschlossen werden. Als erster Schritt muss, wie in Abbildung 3.3 dargestellt, vor dem Modellieren die Form, die Lage und die semantische Information der Elemente festgelegt werden. Diese müssen mit jedem CAD-Programm modellierbar sein und darin das gleiche Produkt repräsentieren.

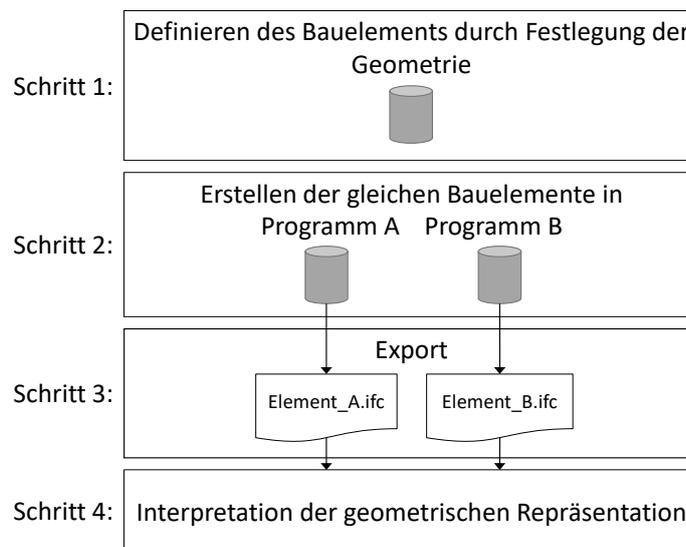


Abbildung 3.3: Aufbau der Untersuchung des aktuellen geometrischen Exports mit IFC

Im Bereich der Modellerstellung sind Nutzer immer von der Funktionalität der Werkzeuge, und damit des jeweiligen Programms, eingeschränkt. Zum Beispiel werden Treppen gegebenenfalls durch Wände, Platten und Träger ersetzt und bilden so eine Treppe ab (vgl. [Monteiro & Martins, 2013], S.249). Die Analyse von IFC-Modellen in Hinblick auf die geometrische Repräsentation soll jedoch Grundlage für einen Leitfaden sein, welcher allgemein gültig ist und sich nicht auf die verwendeten Softwareanwendungen beschränkt. Daher wird sich die Analyse auf einige simple Bauelemente beschränken, welche mit möglichst jedem CAD-Programm, erstellt werden können.

Ein Gebäude besteht stets aus konstruktiven, sowie dekorativen, Bauelementen. Elementar für die Planung sind dabei die umschließenden Bauelemente Boden, Dach und Wand sowie aussteifende Objekte. Anzunehmen ist, dass Programme der Baubranche besonders Wände, Geschossdecken, Stützen und Dächer als grundlegende architektonische Elemente standardisiert hinzufügen können. Die gewählten CAD-Programme verfügen für diese Elemente über Werkzeuge, die in gleicher Weise arbeiten. Aufgrund dessen werden in diesem Abschnitt die Bauelemente Wand, Geschossdecke und Stütze, für Schritt zwei aus Abbildung 3.3, gewählt. Für vielfältige Formen soll am Element Wand zusätzlich mit einer runden Form, einer Öffnung und einer Abschrägung gearbeitet werden.

Für Schritt drei aus Abbildung 3.3, stehen mit IFC4 zwei MVDs zur Verfügung. Um alle möglichen geometrischen Repräsentationen benutzen zu können, wird die DTV verwendet. Im RV ist zudem nicht vorgesehen, dass Änderungen im Modell erzeugt werden.

Nach erfolgtem Export war eine Präferenz in der Wahl der geometrischen Repräsentationen erkennbar. Daher wird der Untersuchung ein Körpermodell hinzugefügt, um eine weitere geometrische Repräsentation zu behandeln. Die Bauelemente werden in den folgenden Abschnitten einzeln betrachtet. Basierend auf der STEP-Datei werden dann, als letzter Schritt aus Abbildung 3.3, die verwendeten geometrischen Repräsentationen des IFC-Standards genauer betrachtet. Auf den Aufbau des STEP-Files wird anhand des ersten Basiselementes *Wand* detaillierter eingegangen, da sich grundlegende Merkmale für alle Elemente wiederholen. Im Anhang A.2 können die, für die geometrische Repräsentation wichtigen, Auszüge aus den STEP-Files eingesehen werden.

Die erstellten IFC4-Modelle aus den unterschiedlichen Programmen können interpretiert und verglichen werden. Generell sind zwei nicht-visuelle Vergleiche der IFC-Modelle möglich: direkt textbasierte und objektbasierte Vergleiche, bzw. eine Kombination aus diesen (vgl. [Pazlar & Turk, 2008], S. 365).

Textbasiertes Vergleichen ist nur selten geeignet, da es verschiedene Arten der Strukturierung einer IFC-Datei gibt. Das heißt, der Aufbau des STEP-Files ist je nach Softwarehersteller unterschiedlich. Daher können die Datensätze nicht “nebeneinander gelegt” und direkt miteinander verglichen werden. Es gibt jedoch Ausnahmen: zum Beispiel wenn nur eine geringfügige Änderung vorgenommen wird und das Modell erneut exportiert wird. Dann kann die gleiche Abfolge von Instanzen mit gleichen Identifikatoren vorhanden sein.

Für einen objektbasierten Vergleich ordnet man zum Beispiel die STEP-Datei nach den referenzierten Identifikatoren. Dieses Verfahren wurde in dieser Untersuchung verwendet, da mit einfachen Mitteln ein Sortieren der Instanzen erfolgen kann. Begonnen wurde mit der Instanz die von *IfcBuildingElement* abstammt und das jeweilige Gebäudeelement initiiert. Die so angeordneten STEP-Files sind anschließend übersichtlicher und können verglichen werden.

In dieser Arbeit wurden folgende Programme genutzt, entweder als Freeware oder Studentenversion:

CAD-Programme Revit® 2017 von Autodesk® und Allplan 2017-0-5 von Allplan GmbH, Tochterfirma der Nemetschek SE. Diese wurden zur Modellerstellung genutzt, da sie zu den gängigen CAD-Programmen in der AEC-Branche (Architecture, Engineering and Construction) zählen.

IfcQuickBrowser Für das textuelle Betrachten auch von sehr großen IFC Dateien. Die IFC Datei wird dabei in Textform geöffnet und kann per Auswahl in einer Baumstruktur dargestellt werden.

WinMerge Ein Open-Source-Programm zum Vergleichen und Zusammenführen von Dateien sowie Ordern. Das Windows-Tool stellt Unterschiede in einem visuellen Textformat dar.

Notepad++ Texteditor, wurde zum Anzeigen und Bearbeiten der STEP-Files genutzt.

SolibriModelViewer v9.7 Visualisierung von IFC-Modellen und nutzbar zur visuellen Kontrolle.

3.2.1 Wand

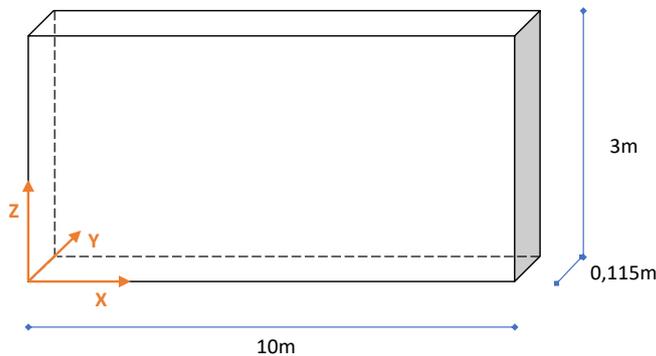


Abbildung 3.4: Vordefiniertes Element *Wand*

Die Abbildung 3.4 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Das Element *Wand* ist festgelegt als eine 11,5 cm dicke Kalksandsteinwand mit einer Länge von 10 m und einer Höhe von 3 m. Die Referenzlinie einer Wand kann bei der Modellierung unterschiedlich positioniert werden: auf der Innenseite, auf der Außenseite oder mittig.

Anhand dieser Linie wird in der Software eine Wand erstellt. Die Abb. 3.5 zeigt einen Ausschnitt aus einer STEP-Datei, sortiert nach der geometrischen Darstellung einer in Allplan modellierten Wand mit der Referenzlinie auf der Innenseite.

Das Objekt *Wand* wird durch eine Instanz vom Typ *IfcWallStandardCase* (Abb. 3.5, Zeile 1, #132) erzeugt. Dieses besitzt eine *GlobalID* und referenziert eine Platzierung und die Form der Wand. Die Platzierung erfolgt mittels *IfcLocalPlacement* (Abb. 3.5, Zeile 2, #64) über zwei Attribute. Mit Hilfe des ersten Attributs *PlacementRelTo* kann das Objekt relativ zu einem anderen Objekt platziert werden. Ist dieses jedoch leer, wird das Objekt direkt im globalen Koordinatensystem platziert. Das zweite Attribut von *IfcLocalPlacement* ist vom Typ *IfcAxis2Placement* (Abb. 3.5, Zeile 3, #61), welches das lokale Koordinatensystem des Objekts platziert. Die Attribute des, in diesem Fall

```

1 #132= IFCWALLSTANDARDCASE('3XjfkwiE9EDRoXpKvYw$_d',$,',',$,,$,#64,#65,$,$);
2 #64= IFCLOCALPLACEMENT($,#61);
3 #61= IFCAXIS2PLACEMENT3D(#62,$,$);
4 #62= IFCARTESIANPOINT((0.,0.,0.));
5 #65= IFCPRODUCTDEFINITIONSHAPE($,$,(#69,#105));
6 #69= IFCSHAPE REPRESENTATION(#19,'Body','SweptSolid',(#84));
7 #19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,*,#15,$,.MODEL_VIEW.,$);
8 #15= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.000000000000000E-5,#10,$);
9 #10= IFCAXIS2PLACEMENT3D(#13,$,$);
10 #13= IFCARTESIANPOINT((0.,0.,0.));
11 #84= IFC EXTRUDEDAREASOLID(#76,#85,#92,3000.);
12 #76= IFCRECTANGLEPROFILEDEF(.AREA.,$,#79,10000.,115.);
13 #79= IFCAXIS2PLACEMENT2D(#80,#82);
14 #80= IFCARTESIANPOINT((5000.,57.5));
15 #82= IFCDIRECTION((1.,0.));
16 #85= IFCAXIS2PLACEMENT3D(#86,#88,#90);
17 #86= IFCARTESIANPOINT((0.,0.,0.));
18 #88= IFCDIRECTION((0.,0.,1.));
19 #90= IFCDIRECTION((1.,0.,0.));
20 #92= IFCDIRECTION((0.,0.,1.));
21 #105= IFCSHAPE REPRESENTATION(#21,'Axis','Curve2D',(#108));
22 #21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis','Model',*,*,*,*,#15,$,.MODEL_VIEW.,$);
23 #15= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,1.000000000000000E-5,#10,$);
24 #10= IFCAXIS2PLACEMENT3D(#13,$,$);
25 #13= IFCARTESIANPOINT((0.,0.,0.));
26 #108= IFCPOLYLINE((#110,#112));
27 #110= IFCARTESIANPOINT((0.,0.));
28 #112= IFCARTESIANPOINT((10000.,0.));

```

Abbildung 3.5: Nach Verlinkung geordnete Ausschnitt aus einem STEP-File einer *Wand* mit der Referenzlinie auf der Innenseite (erstellt mit Allplan)

IfcAxis2Placement3D, sind die Lage des lokalen Koordinatensystems *IfcCartesianPoint* (Abb. 3.5, Zeile 4, #62), sowie zwei Attribute des Typs *IfcDirection*, welche die Achsen und deren Richtung angeben. Sind die letzteren zwei Attribute nicht angegeben werden automatisch zwei Achsen angenommen, welche der X- und der Z-Achse des globalen Koordinatensystems entsprechen. Eine Dritte wird aus diesen abgeleitet und ergibt in diesem Fall die Y-Achse.

Die Form der Wand wird durch *IfcProductDefinitionShape* (Abb. 3.5, Zeile 5, #65) beschrieben. Der Name und eine Beschreibung der Produktdefinition sind die ersten beiden Attribute diesen Typs, welche nicht befüllt wurden.

Das dritte und letzte Attribut beinhaltet die Beschreibung der geometrischen Repräsentation, auf welcher der Fokus liegt. Dabei ist es möglich, mehrere geometrische Repräsentationen durch eine Auflistung dieser zu erzeugen. Die Repräsentationen sind stets vom Typ *IfcShapeRepresentation*. Das Bauelement *Wand* wird darin als dreidimensionaler Volumenkörper (*SweptSolid*) und als zweidimensionale Kurve (*Curve2D*) beschrieben (Abb. 3.5, Zeile 6 und 21, #69 und #105).

Bei der *SweptSolid*-Repräsentation wird ein Profil entlang einer Achse extrudiert. In der Tabelle 3.1, in der ersten Zeile, sind die Instanzen zur Erzeugung dieser Repräsentation durch Allplan grafisch dargestellt. Dem Typ *IfcExtrudedAreaSolid* ist die Form des Profils zugeordnet, in diesem Fall ein *IfcRectangleProfileDef* (Abb. 3.5, Zeile 12, #76), da die Wand ein rechteckiges Profil hat. Die Profil-Parameter sind von *buildingSmart*

definiert, wie in Abbildung A.21c im Anhang, dargestellt. In diesem Beispiel hat das Profil eine Länge $XDim = 10000$ und Breite $YDim = 115$, welche dem Profil direkt als Attribute zugewiesen werden.

Die Profile werden in zwei Arten eingeteilt, in *Area*- und *Curve*-Typen. Die *Area*-Typen füllen die Pofilbegrenzung aus und werden für Volumenkörper verwendet. Die *Curve*-Typen bestehen nur aus der Profilbegrenzung.

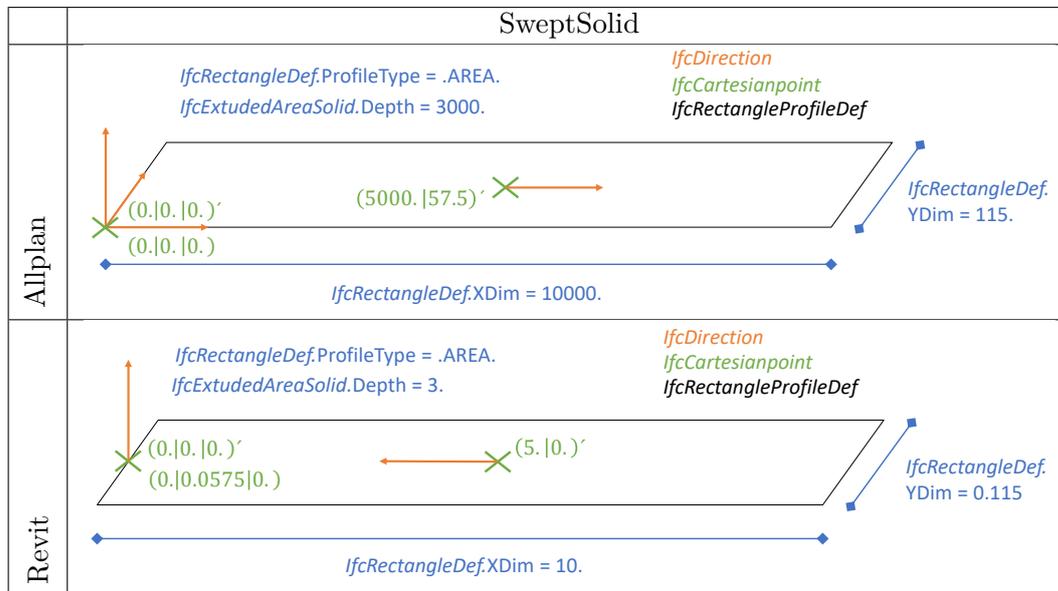


Tabelle 3.1: SweptSolid Geometrie des vordefinierten Elementes *Wand*

Das Attribut des Profils vom Typ *IfcAxis2Placement2D* (Abb. 3.5, Zeile 13, #79) beinhaltet die Koordinaten der lokalen X- und Y-Achse mit $\begin{pmatrix} 5000 & 57.5 \end{pmatrix}$, also jeweils die Hälfte von $XDim$ und $YDim$, sowie die Orientierung der lokalen X-Achse $\begin{pmatrix} 1 & 0 \end{pmatrix}$. Das *SweptSolid*-Objekt bekommt durch das Attribut vom Typ *IfcAxis2Placement3D* (Abb. 3.5, Zeile 16, #85) die Position und die Orientierung innerhalb des dreidimensionalen Raums zugewiesen. Durch das Attribut vom Typ *IfcDirection* (Abb. 3.5, Zeile 20, #92) wird dem *SweptSolid*-Objekt die Extrusionsrichtung zugeordnet. Zusammen mit der Extrusionslänge, eingeführt als Attribut der *IfcExtrudedAreaSolid*-Instanz (Abb. 3.5, Zeile 11), wird die Extrusion definiert.

Die zweite Darstellung der *Wand* als 2D-Kurve wird durch eine Polylinie erzeugt. In Tabelle 3.2 Zeile eins, sind die Instanzen zur Erzeugung dieser Repräsentation durch Allplan grafisch dargestellt. Das Attribut vom Typ *IfcPolyline* (Abb. 3.5, Zeile 26, #108) erzeugt mit den zwei Punkten $\begin{pmatrix} 0 & 0 \end{pmatrix}$ und $\begin{pmatrix} 10000 & 0 \end{pmatrix}$ eine Polylinie. Diese Linie entspricht der Referenzlinie der *Wand* im CAD-Programm.

Erzeugt man das gleiche Element *Wand* in Allplan mit der Referenzlinie in der Mitte oder der Außenseite, sieht der Export verändert aus. Im Fall “Referenzlinie außen” wurde der Startpunkt zum Modellieren in der Software mit den globalen Koordinaten $\begin{pmatrix} 0 & 0.115 \end{pmatrix}$ gewählt. Der Endpunkt wurde um 10m entlang der X-Achse verschoben. Dementsprechend sind die Koordinaten der Polylinie $\begin{pmatrix} 0 & 115 \end{pmatrix}$ und $\begin{pmatrix} 10000 & 115 \end{pmatrix}$. Die gleiche Veränderung tritt auf im Fall “Referenzlinie mittig”, jedoch mit einem

Curve2D	
Allplan	<p><i>IfcPolyline</i> <i>IfcCartesianpoint</i></p> <p>(0 0.) (1000 0.)</p>
Revit	<p><i>IfcPolyline</i> <i>IfcCartesianpoint</i></p> <p>(0 0.0575 0.) (10 0.)'</p>

Tabelle 3.2: Achsen Geometrie des vordefinierten Elementes *Wand* (Referenzlinie innen)

verschobenen lokalen Koordinatensystem.

In Verbindung mit einem Materialprofil kann aus der Achsgeometrie ein Volumenkörper generiert werden. In Abbildung 3.6 ist die Nutzung des Materialprofils beispielhaft von *buildingSmart* dargestellt. Die *LayerSetDirection* gibt an, entlang welcher Achse (*X – Achse* $\hat{=}$ *AXIS1*, *Y – Achse* $\hat{=}$ *AXIS2*, *Z – Achse* $\hat{=}$ *AXIS3*) die Schichten angeordnet werden und ob mit gleicher (*DirectionSense = POSITIVE*) oder entgegengesetzter Richtung (*DirectionSense = NEGATIVE*). Mittels *OffsetFromReferenceLine* wird die Verschiebung der Basislinie des Materialprofils (*MaterialLayerSetBase*) von der Referenzlinie des Elements angegeben, je nach Vorzeichen des Wertes, in positiver oder negativer Richtung.

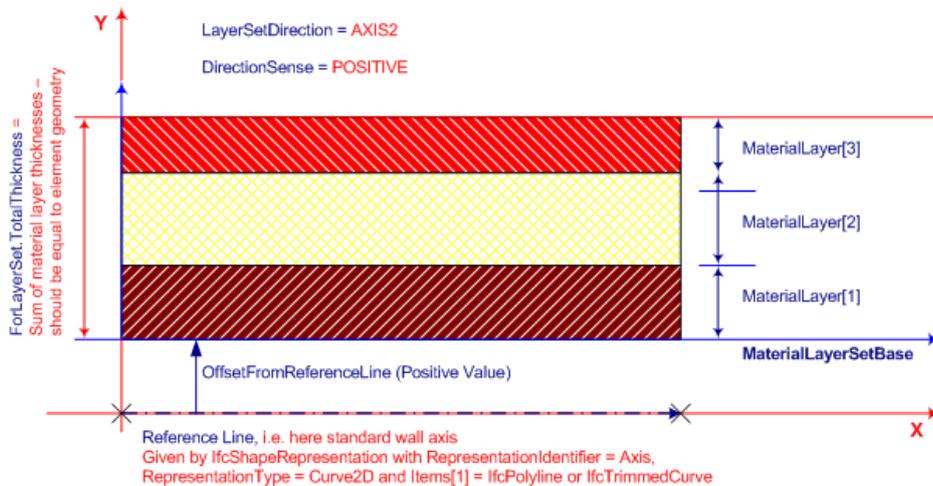


Abbildung 3.6: Beispiel der Nutzung von *IfcMaterialLayerSetUsage* für eine Standardwand (Quelle: [Liebich *et al.*, 2015-2017])

Da im vorliegenden Fall des Wand-Exports drei verschiedene Referenzlinien möglich sind, wird für das gleiche Ergebnis das Materialprofil unterschiedlich definiert. Die Attribute von *IfcMaterialLayerSetUsage* sind, unabhängig von der Positionierung der Referenzlinie, bis auf das Attribut *OffsetFromReferenceLine*, also dem Abstand zur Startlinie, gleich. So wird die Basislinie des Materialprofils im Fall der Referenzlinie in der Mitte um $-0,0575m$ und Außen um $-0,115m$ entlang der *y*-Achse verschoben. Damit ist die Basislinie aller drei Profile gleich der globalen *X*-Achse.

Betrachtet man die Exportdaten aus Revit der auf gleiche Weise modellierten Wand, zeigt sich ein Unterschied. Wie die Tabelle 3.1 zeigt, ist im Revit-Export die Lage des lokalen Koordinatensystems der Wand verschoben und liegt nicht im globalen Koordinatenursprung. Die Achsen sind genauso ausgerichtet wie die globalen X- und Y-Achsen, der Ursprung des lokalen Koordinatensystems liegt jedoch in $\begin{pmatrix} 0 & 0.0575 \end{pmatrix}$, also in der Mitte der Wand. Mit Revit wird die Polylinie für die 2D-Achsendarstellung immer, unabhängig von der Lage der Referenzlinie im Programm, mit den lokalen Koordinaten $\begin{pmatrix} 0 & 0 \end{pmatrix}$ und $\begin{pmatrix} 10000 & 0 \end{pmatrix}$ exportiert, welche damit immer in der Mitte der Wand liegt. Aufgrund dessen wird die Basislinie des Materialprofils um $-0,0575m$ entlang der lokalen y-Achse verschoben, wie im von Allplan erzeugten Fall "Referenzlinie mittig". Das Rechteckprofil für die Extrusion liegt zudem um 180° gedreht. Dies unterstreicht die unterschiedliche Nutzung der gleichen geometrischen Repräsentation bei trotzdem gleichem resultierendem geometrischem Objekt.

Halbkreiswand

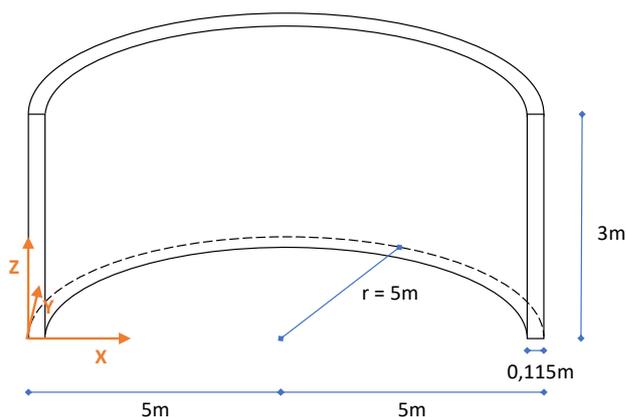


Abbildung 3.7: Vordefiniertes Element *Halbkreis Wand*

Die Abbildung 3.7 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Eine gekrümmte, in diesem Fall halbkreisförmige Wand, wird von beiden Programmen gleichzeitig als 2D-Kurve und als SweptSolid exportiert. In den Tabellen 3.3 und 3.4 sind die Instanzen zur Erzeugung dieser Repräsentationen grafisch dargestellt.

Modelliert wurden die Wände mit der Lage der Referenzlinie innen und einem Radius der Außenkante von 5 m. Bei der Lage der Wandachse unterscheiden sich die Exports aus den Anwendungen wie in dem Beispiel zuvor. In Allplan liegt die Referenzlinie, genauso wie modelliert, auf der Innenkante und in Revit in der Mitte der Wand. Dies wird in der Tabelle 3.3 durch die Nutzung verschiedener Radien deutlich. Erstellt wird die Achsgeometrie durch eine *IfcTrimmedCurve*-Instanz. Diese besitzt als Attribut eine Basiskurve, welche verschiedener Art sein kann, zum Beispiel eine Ellipse, eine Polylinie oder eine BSplineKurve. Die Basiskurve ist im Export aus Allplan ein Kreis mit einem Radius von 5 m und dem lokalen Koordinatensystem positioniert in $\begin{pmatrix} 5 & 0 & 0 \end{pmatrix}$. Die gestutzte Kurve wird, wie in Abbildung 3.8 beschrieben, definiert. So wird die Kurve in diesem Beispiel mit $Trim1 = 0^\circ, Trim2 = 180^\circ$ und dem $SenseAgreement = True$ von 0° nach 180° , entgegen dem Uhrzeigersinn, kreisrund geschlagen.

Das Profil für die Extrusion der SweptSolid-Geometrie wird in Allplan durch einen

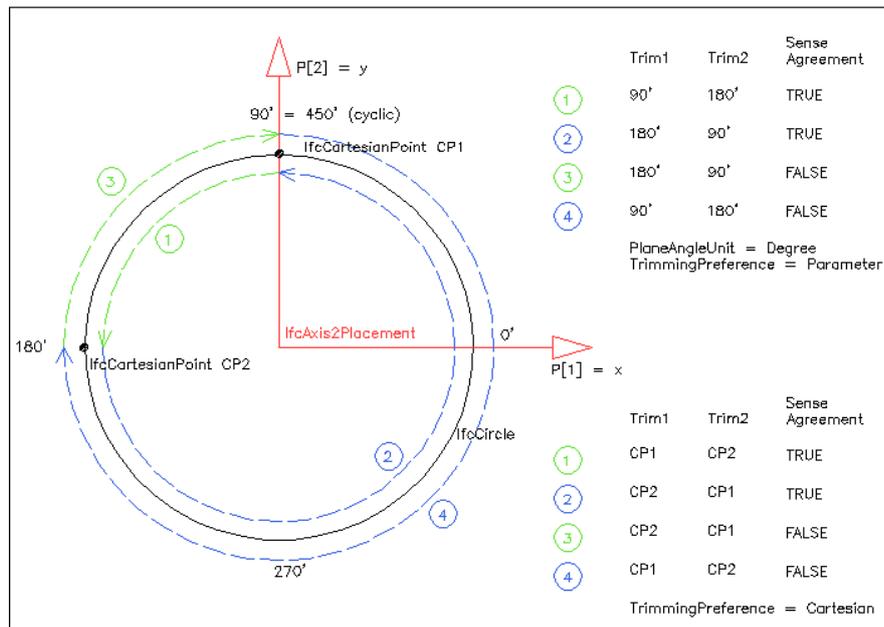


Abbildung 3.8: Definition einer gestutzten Kurve. *Trim1* gibt den ersten und *Trim2* den zweiten Punkt an. Über das *SenseAgreement* wird die Gangrichtung bestimmt (Quelle: [Liebich *et al.*, 2015-2017])

Curve2D	
Allplan	<p><i>IfcCircle</i>.Radius = 5000. <i>IfcTrimmedCurve</i>.SenseAgreement = T.</p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcCircle</i></p> <p><i>IfcTrimmedCurve</i>.Trim2 = 180. (5000. 0. 0.) (0. 0. 0.)' <i>IfcTrimmedCurve</i>.Trim1 = 0.</p>
Revit	<p><i>IfcCircle</i>.Radius = 4.9425 <i>IfcTrimmedCurve</i>.SenseAgreement = T.</p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcCircle</i></p> <p>(0.0575 0. 0.) (0. 0. 0.)' (4.9425 0. 0.)' <i>IfcTrimmedCurve</i>.Trim2 = 180. <i>IfcTrimmedCurve</i>.Trim1 = 360.</p>

Tabelle 3.3: Achsgeometrie des vordefinierten Elementes *Halbkreis Wand*

SweptSolid	
Allplan	<pre> IfcArbitraryClosedProfileDef.ProfileType = .AREA. IfcExtudedAreaSolid.Depth = 3000. IfcDirection IfcCartesianpoint IfcPolyline </pre>
Revit	<pre> IfcArbitraryClosedProfileDef.ProfileType = .AREA. IfcExtudedAreaSolid.Depth = 3. IfcCircle.Radius = 4.9425 / 4.885 IfcTrimmedCurve.SenseAgreement = T. IfcDirection IfcCartesianpoint IfcPolyline IfcCircle </pre>

Tabelle 3.4: SweptSolid-Geometrie des vordefinierten Elementes *Halbkreis Wand*

Polygonzug *IfcPolyline* erstellt. Das lokale Koordinatensystem des Profils wird mit $x = 5m$ auf der globalen X-Achse platziert. Mit 124 lokalen Koordinaten *IfcCartesianpoint* werden die Punkte des Polygonzuges, welcher den Halbkreis skizziert, definiert. Die Punkte sind entlang der lokalen Y-Achse gespiegelt. Das heißt, dass jeder Punkt im positiven Bereich der X-Achse ein Pendant im negativen Bereich hat. Es werden zwar mehr Informationen gespeichert, aber es wird keine exakte Darstellung der runden Wand erstellt.

Im Gegensatz dazu wird das Profil der Wand in Revit durch einzelne Kurven beschrieben. Das heißt, als zusammenhängende Kurve *IfcCompositeCurve* aus mehreren Segmenten. Die Teilkurven sind als *IfcCompositeCurveSegment*-Instanzen in der *IfcCompositeCurve* verlinkt. Die Segmente für die Beschreibung der Wand sind je zwei Polylinien und zwei Kurven.

Die Kurven vom Typ *IfcTrimmedCurve* beschreiben, mit einem Kreis als Basiskurve, die zwei runden Außenränder der Wand und die Polylinien die geraden Außenkanten. Die Repräsentation der Rundung der Wand ist durch Revit exakt, da die Rundung durch eine Kurve direkt beschrieben wird. Die Rundung ist durch Allplan dagegen durch einen Polygonzug mit 124 Punkten, welche auf der Außenlinie liegen, angenähert.

Abgeschrägte Wand

Die Abbildung 3.9 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Die Wand wurde wie das Element *Wand* mit der Referenzlinie innen modelliert. Die obere rechte Ecke wird dann um einen Meter nach unten verschoben. Die Achsgeometrie gestaltet sich wie in den vorherigen Beispielen und wird daher nicht weiter erläutert.

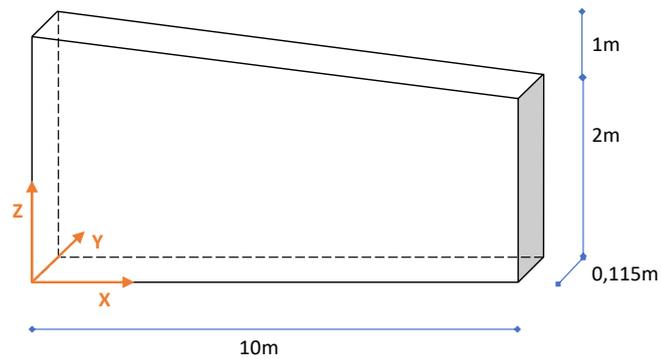


Abbildung 3.9: Vordefiniertes Element *Abgeschrägte Wand*

Clipping	
Allplan	<p><i>IfcHalfSpaceSolid</i>.AgreementFlag = T. <i>IfcBooleanClippingResult</i>.Operator = DIFFERENCE.</p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcPlane</i></p>
Revit	<p><i>IfcHalfSpaceSolid</i>.AgreementFlag = F. <i>IfcBooleanClippingResult</i>.Operator = DIFFERENCE.</p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcPlane</i></p>

Tabelle 3.5: Geometrische Repräsentationen eines Halbraumes, benötigt für das vordefinierte Element *Abgeschrägte Wand*

Die Programme Allplan und Revit nutzen für die Darstellung einer gestutzten Wand das Clipping. Clipping-Geometrien werden mit dem Operator *difference*, einem Volumenkörper und einem Halbraum erzeugt. Dafür wird eine Wand instanziiert, wie im Beispiel *Wand* und durch eine *IfcPlane*, welche einen Halbraum definiert, gestutzt.

In der Tabelle 3.5 sind die Instanzen zur Erzeugung eines Halbraums grafisch darge-

stellt. Die Entität *IfcHalfSpaceSolid* beinhaltet eine *AgreementFlag*, welche anzeigt, auf welcher Seite sich der durch die *IfcPlane* aufgespannten Ebene, der Halbraum befindet. Die Ebene wird durch einen Punkt und Vektoren aufgespannt. Mit *IfcAxis2Placement3D* werden die lokale X- und Z-Achse angegeben. In der X-Y-Ebene wird die *IfcPlane* erzeugt, welche die Berandung des Halbraums darstellt. Konkret im Export aus Allplan verläuft die Ebene durch den globalen Koordinatenpunkt $(0, 0, 3)$, die Z-Achse ist durch den Vektor $(-0,0995, 0, 0,995)$ und die X-Achse durch den Vektor $(0,995, 0, -0,0995)$ definiert.

Die Differenz von der Wand und dem Halbraum erzeugt die gestutzte Wand. Der Export aus Revit unterscheidet sich in der Ausrichtung der Achsen (vgl. Tabelle 3.5) und erzielt durch die *AgreementFlag = False* das gleiche Ergebnis.

Öffnung in Wand

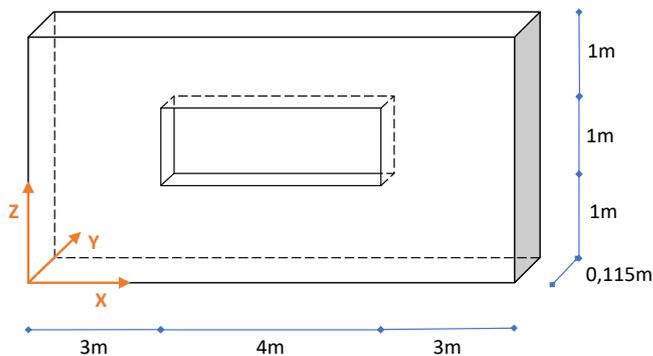


Abbildung 3.10: Vordefiniertes Element *Öffnung in einer Wand*

Die Abbildung 3.10 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Bei der Darstellung einer Wand mit Öffnung werden zwei Objekte erstellt und miteinander in Beziehung gesetzt. Dafür wird die Wand genauso wie im Beispiel vorher durch eine *IfcWallStandardCase*-Instanz repräsentiert und mit einer Öff-

nung verknüpft. Die Beziehung zwischen je einem *BuildingElement* und einem *OpeningElement* wird durch ein Objekt vom Typ *IfcVoidsElement* hergestellt und erzeugt die *Öffnung in der Wand*.

Mit den CAD-Programmen Allplan und Revit wurde wie zuvor jeweils das gleiche Objekt erstellt: eine *Wand*, diesmal mit einer Öffnung von 4 x 1 m in der Mitte der Wand. Die Repräsentation der Wand entspricht dem Beispiel *Wand* und wird daher nicht weiter betrachtet.

In beiden IFC-Files wurde als Repräsentation für die Öffnung *IfcOpeningStandardcase* bzw. *IfcOpeningElement* das SweptSolid-Verfahren genutzt. In der Tabelle 3.6 sind die Instanzen zur Erzeugung der Öffnung grafisch dargestellt. Das Profil für die Öffnung wird in der X-Z-Ebene des globalen Koordinatensystems erstellt und entlang der Y-Achse extrudiert. Da per Definition das Profil in einer lokalen X-Y-Ebene liegt, wird das lokale Koordinatensystem des Objekts und des Profils angepasst.

In Allplan wird das Profil durch Begrenzungslinien erstellt, welche sich nicht schneiden und ein geschlossenes Profil ergeben. Durch eine Polylinie mit vier Koordinaten werden

SweptSolid	
Allplan	<p><i>IfcArbitraryClosedProfileDef.ProfileType = .AREA.</i> <i>IfcExtrudedAreaSolid.Depth = 115.</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcPolyline</i></p>
Revit	<p><i>IfcRectangleProfileDef.ProfileType = .AREA.</i> <i>IfcExtrudedAreaSolid.Depth = 0.115</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcRectangleProfileDef</i></p>

Tabelle 3.6: SweptSolid-Geometrie einer Öffnung, benötigt für die vordefinierte *Öffnung in einer Wand*

die Außenkanten eines Rechtecks erstellt. Das lokale Koordinatensystem des Objekts liegt in $\begin{pmatrix} 3 & 0 & 1 \end{pmatrix}$ mit der lokalen X-Achse in globaler Z-Richtung und der lokalen Z-Achse in globaler Y-Richtung. Das lokale Koordinatensystem des Profils entspricht diesem.

Das Profil wird bei der Anwendung von Revit durch ein Rechteckprofil repräsentiert und entlang der globalen Y-Achse extrudiert. Der Unterschied zwischen den beiden Exports ist die Lage der jeweiligen Koordinatensysteme und die Beschreibungsart der Profilaußenkanten.

3.2.2 Geschossdecke

Die Abbildung 3.11 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Eine Geschossdecke

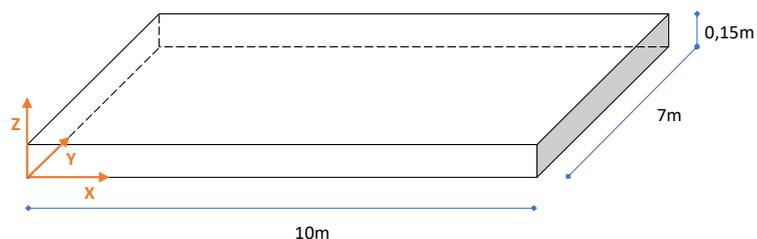


Abbildung 3.11: Vordefiniertes Element *Geschossdecke*

SweptSolid	
Allplan	<p> <code>IfcRectangleDef.ProfileType = .AREA.</code> <code>IfcExtudedAreaSolid.Depth = 150.</code> </p> <p> <code>IfcDirection</code> <code>IfcCartesianpoint</code> <code>IfcRectangleProfileDef</code> </p> <p> <code>IfcRectangleDef.YDim = 7000.</code> </p>
Revit	<p> <code>IfcRectangleDef.ProfileType = .AREA.</code> <code>IfcExtudedAreaSolid.Depth = 0.15</code> </p> <p> <code>IfcDirection</code> <code>IfcCartesianpoint</code> <code>IfcRectangleProfileDef</code> </p> <p> <code>IfcRectangleDef.XDim = 7.</code> </p>

Tabelle 3.7: SweptSolid-Geometrie des vordefinierten Elementes *Geschossdecke*

cke mit den Maßen 10m Breite, 7m Tiefe und einer Dicke von 0,15m ist in Revit und Allplan ähnlich zu modellieren. Mit festgelegter Deckendicke muss die Breite und die Tiefe modelliert werden. Es gibt dabei die Möglichkeit das Profil der Geschossdecke durch fünf Punkte zu beschreiben oder durch zwei Punkte aufzuspannen. Beide Modellierungsarten ergeben den gleichen Export bei gleicher Software.

In der Tabelle 3.7 sind die Instanzen zur Erzeugung der geometrischen Repräsentation grafisch dargestellt. Abgesehen von der Positionierung der lokalen Koordinatensysteme ist der einzige Unterschied zwischen den Exports aus Revit und Allplan die Extrusionsrichtung des Profils. Allplan erstellt die Geometrie der Geschossdecke vergleichbar zur Geometrie der Wand. Revit dagegen legt das Profil der Geschossdecke in die X-Y-Ebene auf Höhe $z = 0.15m$ und extrudiert das Profil entgegen der globalen Z-Achse, also “von oben nach unten” (vgl. Tabelle 3.7).

3.2.3 Stütze

Die Abbildung 3.12 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird. Zum Erstellen einer Stütz muss zuerst das Profil der Stütz festgelegt und anhand einem Transportpunkt im Projekt positioniert werden. Hier sei nun die Stütze rechteckig mit den Maßen $0,25m \times 0,25m$ und einer Höhe von $3m$.

In Allplan kann der Transportpunkt, anhand welchem die Stütze platziert wird, in eine Ecke oder in die Mitte des Profils gelegt werden. In Revit ist dieser immer in der Mitte des Profils. Für die gleiche Positionierung im CAD-Programm muss in Revit ein Hilfsraster angewendet werden.

Die Exports der *Stütze* aus den zwei CAD-Programmen unterscheiden sich. In Allplan wird die *Stütze* als 3D-Achse und als SweptSolid-Körper repräsentiert, in Revit nur als SweptSolid-Körper. In den Tabellen 3.8 und 3.9 sind die Instanzen zur Erzeugung der geometrischen Repräsentationen grafisch dargestellt. Die Achsgeometrie wird, unabhängig vom Transportpunkt in der Software, immer gleichen erstellt. Die *IfcPolyline* der Achsgeometrie liegt immer auf der gleichen Seitenkante der Stütze. Dies unterscheidet sich von der Achsgeometrie der *Wand*. Bei dieser wurde die Achsgeometrie in Abhängigkeit von der Referenzlinie erzeugt.

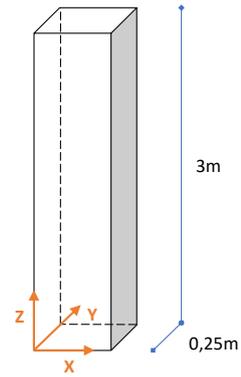


Abbildung 3.12: Vordefiniertes Element *Stütze*

Curve3D	
Allplan	<p><i>IfcPolyline</i> <i>IfcCartesianpoint</i></p>

Tabelle 3.8: Achsgeometrie des vordefinierten Elementes *Stütze*

SweptSolid	
Allplan	<p><i>IfcRectangleDef.ProfileType = .AREA.</i> <i>IfcExtrudedAreaSolid.Depth = 3000.</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcRectangleProfileDef</i></p> <p><i>IfcRectangleDef.YDim = 250.</i></p>
Revit	<p><i>IfcRectangleDef.ProfileType = .AREA.</i> <i>IfcExtrudedAreaSolid.Depth = 3.</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcRectangleProfileDef</i></p> <p><i>IfcRectangleDef.YDim = 0.25.</i></p>

Tabelle 3.9: SweptSolid-Geometrie des vordefinierten Elementes *Stütze*

Revit erstellt im Gegensatz zu Allplan für die Repräsentation der Stütze eine *MappedRepresentation*. Der Hintergrund dieser Nutzung ist die Reduzierung der Anzahl von *ShapeRepresentation* Instanzen für das Sichtfenster, da in einem Projekt meist eine Vielzahl an Stützen der gleichen Art verwendet werden. Damit wird die Größe des IFC-Files verringert. Alle Stützen im Revit-Export verweisen dann auf die gleiche *IfcRectangleProfileDef*-Instanz für die Extrusion. Wie die Tabelle 3.9 zeigt, sind die SweptSolid-Geometrien abgesehen davon gleich aufgebaut.

3.2.4 Körpermodell

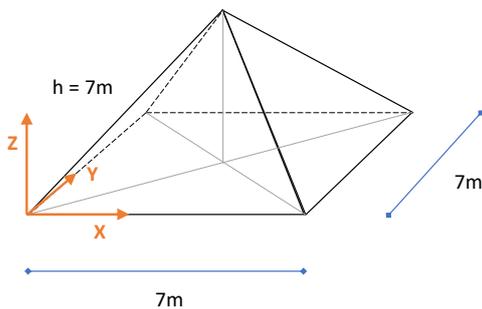


Abbildung 3.13: Vordefiniertes Element *Körpermodell*

Bei der Planung eines Projekts kann in der Entwurfsphase auf reine Körpermodelle zurückgegriffen werden. Diese können einfache geometrische Grundformen besitzen oder speziell für ein Projekt modelliert werden. Die Körpermodelle bestehen lediglich aus einer Oberfläche mit dazugehörigen Kanten und Punkten. Als Beispiel für die Körpermodelle sei hier eine vierseitige Pyramide als Element mit einer Grundfläche von $7m \times 7m$ und einer

Höhe von $7m$ gewählt. Die Abbildung 3.13 zeigt das vordefinierte Element, welches mit den ausgewählten CAD-Programmen modelliert wird.

Allplan nutzt zur Repräsentation die BRep und Revit die Advanced BRep. Der Unterschied zwischen den Repräsentationen ist die hinzugefügte Nutzung von NURBS-Flächen in der erweiterten Methode.

Da das *Körpermodell* nur ebene Flächen hat ist diese im Grunde nicht erforderlich. Die Repräsentationen der Pyramide durch die Software ist aus fünf einzelnen Flächen aufgebaut. In der Tabelle 3.10 wird der Aufbau an einer Fläche gezeigt.

Die gesamte Form wird mit der einfachen BRep durch ein Objekt der Klasse *IfcFacetedBrep* erstellt. Die Oberfläche wird durch ein Objekt vom Typ *IfcClosedShell* definiert, welches aus einer Menge von *IfcFace* besteht und eine geschlossene Oberfläche beschreibt. Je eine Menge aus vorhandenen Punkten ergibt einen Polygonzug der die äußere Berandung der jeweiligen Berandungsflächen *IfcFace* repräsentiert (vgl. Tabelle 3.10, Zeile 1).

Bei der erweiterten Methode, welche im Export aus Revit genutzt wird, ist die Beschreibung der Form detaillierter. Es wird jede Kante mit einer Kantengeometrie vom Typ *IfcCurve* mit Start- und Endpunkt beschrieben, welche wiederum zu einem geschlossenen Zug in *IfcEdgeLoop* zusammengefügt werden. Die Außengrenze *IfcFaceOuterbound* der jeweiligen Fläche und eine unbegrenzte Fläche vom Typ *IfcSurface* definieren die jeweiligen Berandungsflächen *IfcAdvancedFace* (vgl. Tabelle 3.10, Zeile 2). Hier ist die unbegrenzte Fläche vom Subtype *IfcPlane*, da es sich um eine ebene Fläche handelt.

BRep	
Allplan	<p><i>IfcFaceOuterBound.Orientation = .F.</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcPolyloop</i> <i>IfcFace</i></p>
Advanced BRep	
Revit	<p><i>IfcAdvancedFace.SameSense = .T.</i> <i>IfcFaceOuterBound.Orientation = .T.</i> <i>IfcOrientedEdge.Orientation = .T.</i> <i>IfcEdgeCurve.SameSense = .T.</i> <i>IfcVector.Magnitude = 0.3048</i></p> <p><i>IfcDirection</i> <i>IfcCartesianpoint</i> <i>IfcLine</i> <i>IfcPlane</i></p>

Tabelle 3.10: Geometrische Repräsentationen einer Begrenzungsfläche, benötigt für das vordefinierte Element Körpermodell

3.3 Auswertung der genutzten Repräsentationen

Im folgenden Abschnitt werden die geometrischen Repräsentationen, welche bei der Export-Untersuchung zum Einsatz kamen, ausgewertet. In Tabelle 3.11 sind alle während der Untersuchung verwendeten Klassen zur geometrischen Repräsentation der Elemente aufgelistet. Die benötigten Klassen konzentrieren sich in der Zeile 1-6 der Tabelle 3.11 überwiegend auf geometrische Basiselemente wie Koordinaten, Richtungsvektoren, aber auch geometrische Grundformen und Kurven mit dazugehörigen Parametern.

Bei der Repräsentation des *Körpermodells* ist der Unterschied in der Zeile 7 der Tabelle 3.11, in Abhängigkeit vom genutzten Programm, besonders auffällig. Der Grund dafür ist, dass Allplan das Objekt, mit ausschließlich geraden Oberflächen, mit der dafür erforderlich einfachen BRep beschreibt. Revit hingegen verwendet die erweiterte BRep-Methode, welche nicht zwangsläufig von planaren Flächen ausgeht und daher die Flächen und Kanten explizit beschreibt.

Die Elemente *Wand*, *Abgeschrägte Wand* und *Geschossdecke* sind, wie in den Zeilen 1,4 und 6 der Tabelle 3.11 zu sehen, unabhängig vom verwendeten Programm mit Hilfe der selben Klassen repräsentierbar. Auch sonst ist in den Zeilen 2,3 und 5 nur eine sich gering unterscheidende Auswahl genutzter Klassen abzulesen. Daraus lässt sich die Schlussfolgerung ziehen, dass die als grundlegend angenommenen Bauelemente annähernd gleich exportiert werden. Ein solches Ergebnis weist auf eine gute Qualität des IFC4 Standards und ein Einhalten von diesem durch die Softwarehersteller hin.

	ifcExtrudedAreaSolid	ifcRectangleProfile	ifcCartesianPoint	ifcDirection	ifcPolyline	ifcAxis2Placement2D	ifcAxis2Placement3D	ifcArbitraryClosedProfileDef	ifcTrimmedCurve	ifcCircle	ifcCompositeCurve	ifcCompositeCurveSegment	ifcBooleanClippingresult	ifcHalfSpaceSolid	ifcPlane	ifcMappedItem	ifcRepresentationMap	ifcCartesianTransformationOperator3D	ifcFacetedBrep	ifcClosedShell	ifcFace	ifcFaceOuterBound	ifcPolyloop	ifcAdvancedBrep	ifcAdvancedFace	ifcEdgeLoop	ifcOrientedEdge	ifcEdgeCurve	ifcVertexPoint	ifcLine	ifcVector
Wand	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	A	R	
Öffnung																															
Halbkreis Wand																															
Schräge Wand																															
Stütze																															
Geschossdecke																															
Körpermodell																															

Tabelle 3.11: Aufistung der verwendeten Klassen zur geometrischen Repräsentation; A = Allplan; R = Revit

Anhand der Auswahl der verwendeten Klassen, zur Darstellung des *Körpermodells*, in der Zeile 7 der Tabelle 3.11 ist zu erkennen, dass die Auswahl von der geometrischen Repräsentation abhängig ist. In diesem Fall ist die Wahl der geometrischen Repräsentation auch von der Software abhängig. Daher sind in Tabelle 3.12 die verwendeten Repräsentationen noch einmal nach Gebäudeelement und CAD-Programm aufgegliedert. Darin wird deutlich, dass für die Mehrzahl der betrachteten Bauelemente, unabhängig von der Software, die gleichen geometrischen Repräsentationen genutzt werden. Die Stütze

Geometrische Repräsentation	Gebäudeelement	
	Allplan Export	Revit Export
Axis 2D Geometry	<i>Wand, Halbkreis Wand, Abgeschrägte Wand</i>	<i>Wand, Halbkreis Wand, Abgeschrägte Wand</i>
Axis 3D Geometry	<i>Stütze</i>	-
Body SweptSolid Geometry	<i>Wand, Halbkreis Wand, Öffnung, Stütze, Geschossdecke</i>	<i>Wand, Halbkreis Wand, Öffnung, Stütze, Geschossdecke</i>
Body Clipping Geometry	<i>Abgeschrägte Wand</i>	<i>Abgeschrägte Wand</i>
Body Brep Geometry	<i>Körpermodell</i>	-
Body AdvancedBrep Geometry	-	<i>Körpermodell</i>

Tabelle 3.12: Verwendete geometrische Repräsentationen während der Untersuchung

erweist sich in Zeile 2 der Tabelle 3.12 dabei neben dem Körpermodell als Ausnahme. Für fünf von sieben Elementen aus der Betrachtung in Abschnitt 3.2 wurde die SweptSolid Geometrie verwendet.

In den folgenden Abschnitten soll auf die Merkmale der verwendeten Repräsentationen eingegangen werden. Dabei soll herausgearbeitet werden, welche geometrischen Elemente für die Repräsentationen verwendet werden können und welche Konsequenzen dies hat.

3.3.1 Axis

Die Achsgeometrie wird, wie in Unterpunkt 3.2.1 erwähnt, für Elemente genutzt die einem Pfad folgen. Die reine Achsgeometrie wird durch ein Repräsentationselement vom Typ *IfcBoundedCurve* erstellt. Das heißt, dass die Repräsentation durch einen Startpunkt, einen Endpunkt und einer definierten Verbindung entsteht. Die Verbindungstypen des Start- und Endpunktes, aufgezeigt in Abbildung 3.14, sind mathematisch klar definiert. Die Subtypen *IfcTrimmedCurve*, *IfcPolyline*, *IfcBsplineCurve* und *IfcCompositeCurve* haben im IFC4-Standard die in Tabelle 3.13 gelisteten Informationen.

Es ist anzunehmen, dass durch die Anwendung des jeweiligen Werkzeugs im Programm eine eindeutige Wahl des repräsentierenden Pfades beim Export nach der Modellierung stattfindet. Beide verwendeten Programme besitzen die notwendigen Modellierungsoptionen, welche auf den dargestellten Kurven (Subklassen von *IfcBoundedCurve*) basieren.

Vergleicht man die Lage der Punkte bei gleichem Element, mit gleicher Modellierungsweise, zeigt sich keine einheitliche Repräsentation (vgl. Abschnitt 3.2.1). Die Lage der Achse wird offensichtlich in Allplan durch Festlegung der Referenzlinie während des Modellierens bestimmt. Der Export von Revit weist bei variierender Lage der Referenz-

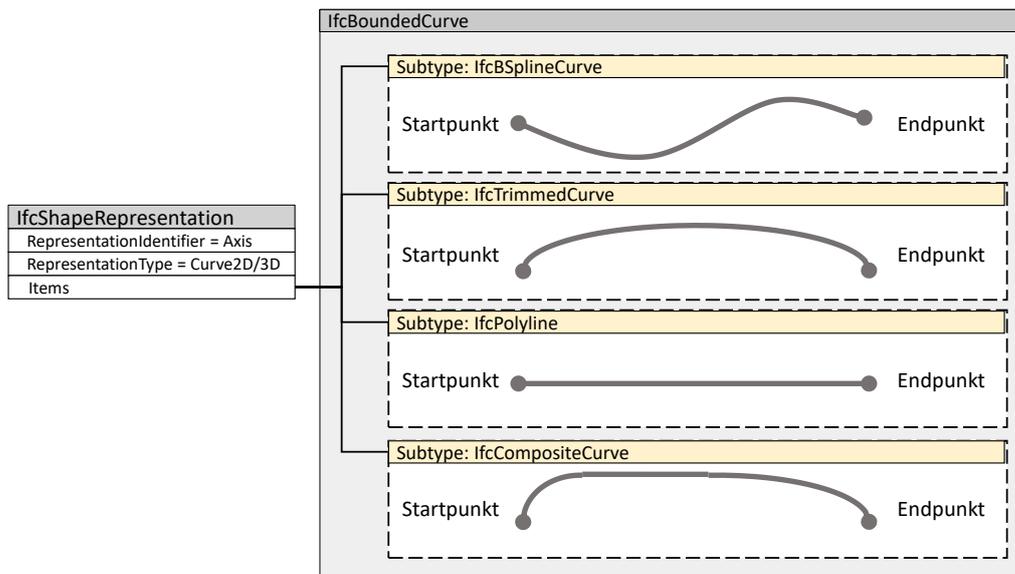


Abbildung 3.14: Informationsgehalt der Achsgeometrie

Repräsentationselement	beinhaltete Informationen
<i>IfcTrimmedCurve</i> - begrenzte Kurve	<ul style="list-style-type: none"> • Basiskurve • Start- und Endpunkt in Form von Koordinaten oder Gradangabe auf der Kurve • Laufrichtung auf der Basiskurve
<i>IfcPolyline</i> - Polylinie	<ul style="list-style-type: none"> • Liste von n-Punkten in Koordinatenform
<i>IfcBSplineCurve</i> - Bézier-Spline-Kurve	Parameter einer Basisfunktion: <ul style="list-style-type: none"> • Grad • Kontrollpunkte • Form der Kurve (Polylinie, Kreisbogen, elliptischer-, parabolischer-, hyperbolischer Bogen, unspezifisch)
<i>IfcCompositeCurve</i> - zusammengesetzte Kurve	<ul style="list-style-type: none"> • Liste der verwendeten n-Kurven

Tabelle 3.13: Informationsgehalt der Möglichen Verbindungstypen

linie dagegen keine unterschiedliche Lage der Achsgeometrie auf. Daraus folgt, dass bei gleicher Modellierungsart unterschiedliche Repräsentationen durch die Achsgeometrie entstehen.

Eine bloße Betrachtung der Achsgeometrie in einer IFC4-Datei kann daher keine eindeutige Aussage über die Lage einer Wand liefern, denn der Repräsentation selbst ist keine Information darüber beigefügt. Es ist also nicht bekannt, auf welche Linie des Wandprofils sich die Achse bezieht. Zur Betrachtung eines Geschossgrundrisses kann es ausreichen, wenn es rein um die grobe Bewertung oder Abstimmung der Proportionen oder der Raumaufteilungen geht.

Wenn die Dicke und die Materialschichten des Elements konstant sind, kann aus der 2D-Achsgeometrie bzw. der 3D-Achsgeometrie und dem Materiallayer das Element generiert werden. Erst durch Hinzunahme des Materialprofils wird klar, wie das Profil der Wand gelegen ist und wie die Achse dementsprechend darin positioniert ist.

Wie in Abschnitt 3.2.1 diskutiert, wird auch die abgeschrägte Wand mit dieser Repräsentation dargestellt. Würde man die abgeschrägte Wand basierend auf der Achsgeometrie erstellen wollen, gäbe es keine Information über die fehlende Ecke. Folglich würde mit dem Material eine ungestutzte Wand erstellt werden.

Es sollte erwähnt werden, dass die 3D-Achsgeometrie, laut buildingSmart, für Stützen gedacht ist. Aber wie in Tabelle 3.12 zusammengefasst, wird diese Repräsentation nicht immer umgesetzt. Die Lage der Achsgeometrie in der Stütze ist im Gegensatz zur Lage der Achsgeometrie in der Wand immer an der selben Stelle (vgl. Abschnitt 3.2.3). Die Achse wird in einer Kante der Stütze definiert. Für statische Berechnungen wäre es jedoch sinnvoller diese in der Mitte der Stütze zu positionieren. Die reine Betrachtung der 3D-Achse enthält keine Information über die Lage und müsste, wie im Beispiel *Wand*, in Kombination mit dem Profil der Stütze betrachtet werden.

3.3.2 SweptSolid

Die SweptSolid-Geometrie wird für die Repräsentation von 3D-Formen von Festkörpern benutzt, die einen Querschnitt besitzen. Ein geschlossener Grundriss wird entlang einem Pfad extrudiert um den Körper zu generieren, dies ist mit dem IFC-Format auf mehrere Arten möglich. Das Profil kann entlang einer Linie oder entlang einem Kreisbogen gestrichen werden. Außerdem ist mit Subtypen der Items (vgl. Abb. 3.15) auch eine Verjüngung entlang dem Pfad möglich, indem zusätzlich zum Start- auch ein Endprofil definiert ist.

Essentiell ist bei dieser Repräsentationsart das Profil, denn die Extrusionslänge und -richtung sollten eindeutig und verlustfrei exportierbar sein. Richtungs- und Längenangaben sind durch entsprechende Eingaben, verknüpft mit Koordinaten, eindeutig. In den Exportuntersuchungen wurden die Profile durch *IfcArbitraryClosedProfileDef* und *IfcRectangleProfileDef* repräsentiert. Auf diese Beiden soll daher der Fokus gelegt werden.

Das rechteckige Profil *IfcRectangleProfileDef* ist Subklasse von der abstrakten Super-

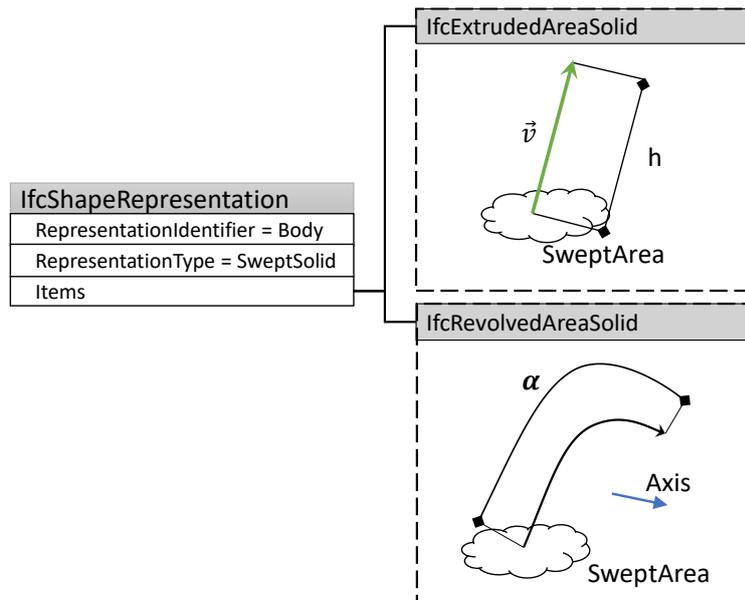


Abbildung 3.15: Informationsgehalt der SweptSolid-Geometrie. Das Attribut *SweptArea* ist vom Typ *IfcProfileDef*

klasse *IfcParameterizedProfileDef* und zeichnet sich durch eine parametrisierte geometrische Form, wie in Abbildung A.21c im Anhang, aus. Das heißt, vorgegebene geometrische 2D-Formen werden mit Parametern flexibel angepasst.

Mit den parametrischen Profildefinitionen lassen sich Profile eindeutig beschreiben. Allerdings nur, wenn sie einer, von buildingSmart vordefinierten, Form entsprechen. Dazu gehören im IFC4 Ellipsen, Kreise, Rechtecke, Trapeze und verschiedene Trägerprofile, wie zum Beispiel das I- oder T-Profil. Die parametrischen Profile sind im Anhang A.3 einzusehen. Wenn davon ausgegangen werden kann, dass die parametrischen Profile von buildingSmart auch in den Schnittstellen unterstützt werden, dann sind die Repräsentationen eindeutig.

IfcArbitraryClosedProfileDef ist ein direkter Subtyp von *IfcProfileDef* und ist laut Borrmann die gebräuchlichste Subklasse (vgl. [Borrmann *et al.*, 2015], S.108). Diese Aussage ist mit der Tabelle 3.11 nicht übereinstimmend, denn darin basierten nur zwei Repräsentationen auf dieser Klasse. Der Grund könnte die begrenzte Untersuchungsbreite sein, aber auch der Umstand, dass in dieser Arbeit eine neue IFC-Version untersucht wird.

IfcArbitraryClosedProfileDef beschreibt ein geschlossenes Profil durch die Außenlinie des abstrakten Typs *IfcCurve*. Die Außenlinie des Profils muss eine geschlossene Kurve sein, um durch Extrusion oder Rotation einen Volumenkörper zu bilden. Die Subtypen *IfcCircle*, *IfcEllipse* als geschlossene Kurven oder eine Liste von Teilkurven, zusammengefasst in *IfcCompositeCurve*, dienen der Bildung einer Außenlinie.

Die Repräsentation der Halbkreiswand wurde jeweils in Allplan und Revit mit einer Instanz von *IfcArbitraryClosedProfileDef* dargestellt, aber trotzdem unterschiedlich genutzt. Allplan nutzt zur Repräsentation des Wandprofils eine Polylinie, welche 124

Punkte auf dem ursprünglichen Profil-Umriss verbindet und Revit nutzt zwei Halbkreise und zwei Linien um einen Umriss zu erstellen.

3.3.3 Clipping

Die Clipping-Geometrie wird für die Repräsentation von 3D-Formen von Festkörpern benutzt, die an einer Ebene abgeschnitten werden. Dafür wird ein SweptSolid-Körper und ein Halbraum benötigt. Durch die Differenz der beiden Objekte entsteht das eigentliche Objekt, das repräsentiert werden soll.

Für den ersten Operand kann statt einem SweptSolid-Körpers auch eine bereits gestutzte Geometrie *IfcBooleanClippingResult* verwendet werden. Der Halbraum wird durch eine Instanz von *IfcHalfSpaceSolid* erstellt und hat eine Fläche vom Typ *IfcSurface*, sowie eine Richtung die anzeigt, welche Seite “gefüllt” ist.

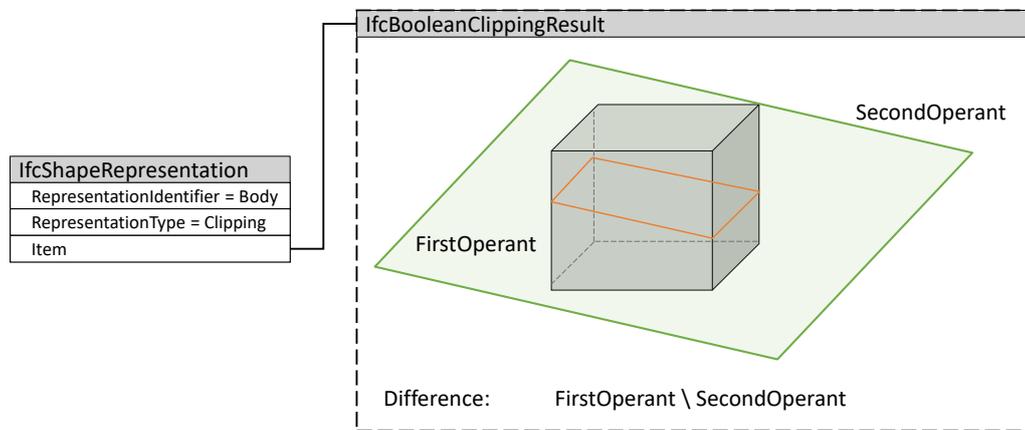


Abbildung 3.16: Informationsgehalt der Clipping-Geometrie. Der *FirstOperand* ist immer ein Volumenkörper und der *SecondOperand* ein Halbraum

Aufgrund der Vielzahl an möglichen Repräsentationen eines Volumenkörpers und dem Hauptgrund der Repräsentation, das Kappen des Körpers, wird der Fokus auf den Halbräumen liegen. Laut der Dokumentation von buildingSmart ist die Art der Fläche abhängig davon, ob der Bearbeitungsbereich begrenzt oder nicht begrenzt ist (vgl. [Liebich *et al.*, 2015-2017]). Im Normalfall ist keine Begrenzung vorhanden, sodass dann eine unbegrenzte Fläche vom abstrakten Typ *IfcElementarySurface*, mit den zwei Subtypen *IfcCylindricalSurface* und *IfcPlane*, zu benutzen ist.

Die planare Ebene als Grenzfläche *IfcPlane* wurde im Abschnitt 3.2.1 betrachtet. Die Modellierung der gekappten Wand wurde in beiden CAD-Programmen durch Manipulation der Eckpunkte und nicht durch einen Schnitt erzeugt. Das heißt, die Schnittebene wurden identifiziert und dem Modell, für den Export, hinzugefügt. Die Qualität der Clipping-Geometrie hängt von den Operanden ab. In dieser Arbeit wurde der zweite Operand, richtig erkannt. Es kann davon ausgegangen werden, dass die *IfcPlane* bei geraden Schnittebenen ohne Fehler erstellt wird. Der erste Operand hat daher den größeren Einfluss auf die Qualität, da er keine feste geometrische Repräsentation hat.

3.3.4 BRep

Die BRep, die Begrenzungsflächen-Geometrie, wird für die Repräsentation von 3D-Formen von Festkörpern benutzt. Die Beschreibung des Körpers wird über die Beschreibung der Außenflächen erzeugt. Zu unterscheiden ist dabei zwischen der Basismethode mit lediglich ebenen Flächen, initiiert durch *IfcFacetedBrep*, und der um gekrümmte Flächen und Kanten erweiterten Methode, initiiert durch *IfcAdvancedBrep*.

Die Oberfläche des Körpers wird durch ein Objekt vom Typ *IfcClosedShell* erstellt. Darin enthalten sind alle Flächen, je nach Methode als *IfcFace* oder *IfcAdvancedFace*. In der einfachen Repräsentation verweisen die Begrenzungsflächen auf *IfcLoop*. Die Eckpunkte der Flächen, *IfcCartesianpoint*, werden referenziert von *IfcLoop* und bilden so die Kanten. In der erweiterten Variante werden die Kanten und die Flächen durch Basis-Splines und deren Parameter, sowie Kontrollpunkte und Funktionsgrad, beschrieben.

Durch die Aufgliederung in Flächen und wiederum in Eckpunkte und Kanten macht das Begrenzungsflächenmodell zum „[...] mächtigsten und flexibelste[n] Verfahren der Geometriemodellierung [...]“ ([Borrmann *et al.*, 2015], S. 105). Ein Nachteil dieser Aufgliederung sind große resultierende STEP-Files die schwer nachvollziehbar sind. Man betrachte dazu die geordneten STEP-Files im Anhang A.2.4, welche lediglich ein Bauelement und kein komplettes Projekt darstellen.

Der Vorteil der klaren Definierung ohne unterschiedliche Ausführungsmöglichkeiten der Repräsentation wiegt viel. Ein Eingreifen in die Geometrie ist zudem einfach möglich, indem einzelne Punkte verschoben werden können. Die Randbedingungen für eine gültige Geometrie aus Abschnitt 2.2 müssen dabei erfüllt bleiben.

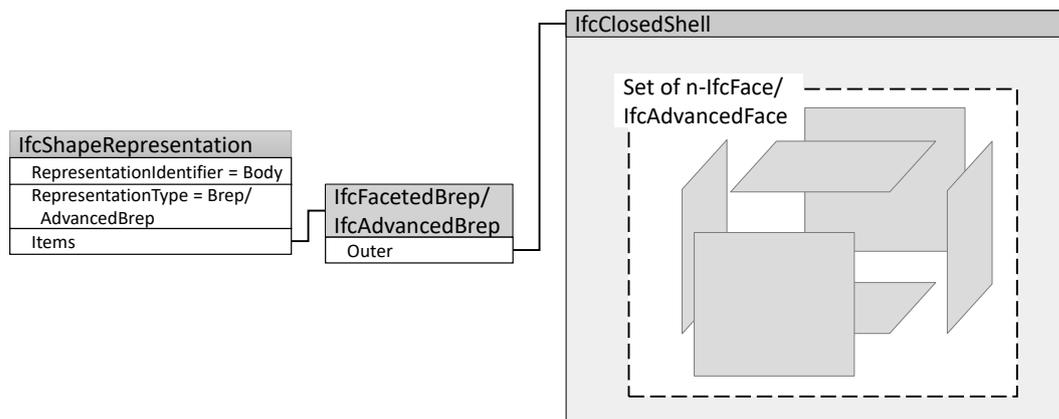


Abbildung 3.17: Informationsgehalt der BRep-Geometrie.

4 Transformationen

4.1 Analyse der Transformationseigenschaft

Wie in Kapitel 3 bereits erwähnt ist neben dem Export auch der Import von IFC-Modellen in verschiedene Softwareprodukte eine mögliche Fehlerquelle. Ein Ansatz zur Verbesserung der Interoperabilität kann, neben der Optimierung der Programm-Schnittstellen, eine automatisierte Transformation der IFC-Modelle sein. Wenn zum Beispiel noch keine Zertifizierung der Software und der Schnittstellen vorliegt oder die Software die aktuelle Repräsentation, aufgrund der großen Vielfalt von buildingSmart standardisierten geometrischen Repräsentationen (siehe Abschnitt 3.1), nicht unterstützt, kann dies nötig sein.

Transformiert wird dann, um die IFC-Modelle zu bereichern oder zu verändern und die Interoperabilität mit IFC zu verbessern. In der Untersuchung, an einigen ausgewählten volumetrischen Elementen, wurden sechs, von buildingSmart standardisierte, Repräsentationen verwendet. Diese wurden anschließend in vier Geometrierepräsentationen zusammengefasst und betrachtet: Achsgeometrie, SweptSolid-Geometrie, Clipping und BRep.

Die Achsgeometrie beinhaltet stets eine Achse und kann, mit einem Materialprofil, ein Volumenkörper erzeugen. Die Kombination dieser Elemente ist der SweptSolid-Geometrie sehr ähnlich. Eine Transformation in die SweptSolid-Geometrie erscheint daher durchaus möglich. Die vorangegangene Untersuchung zeigte jedoch, dass parallel zur Achsgeometrie auch SweptSolid-Geometrien erstellt werden, daher ist dies nicht zwingend nötig. In eine Clipping-Geometrie zu transformieren erscheint nicht sinnvoll, da kein gekappter Körper erzeugt wird.

Bei der Wahl der Bauelemente wurde auf die Modellierbarkeit mit grundlegenden Werkzeugen geachtet. Die geometrischen Repräsentationen SweptSolid und Clipping (mit SweptSolid-Körper als Operand) basieren auf den Modellierungsschritten im jeweiligen Programm, daher erscheint die Wahl der Repräsentationen konsequent. Im IFC-Modell ist der Konstruktionsbaum nach dem Export noch vorhanden: die SweptSolid-Geometrie erzeugt Profile, welche entlang einer Strecke geführt werden (vgl. Abb. 3.15) und die Clipping-Geometrie erzeugt das endgültige Element nach dem Kappen eines kompletten Körpers (vgl. Abb. 3.16).

Der Import in eine Software erfordert, dass das IFC-Modell, im Fall des SweptSolids dessen Konstruktionsschritte, in das interne Datenmodell übertragen werden kann. Ein Mapping kann nur gelingen, wenn die Software die Darstellung mit IFC interpretieren kann. Ähnlich verhält es sich mit parametrisierten Profilen für das SweptSolid-Modell. Das IFC4-Format bietet eine Auswahl an geometrischen Formen für Profile an, welche durch Parameter definiert werden. Für den Import muss die Form bekannt sein, um mit den Parametern ein Mapping in die Software zu bewerkstelligen. Für diesen Fall soll in dieser Arbeit eine Transformation der *SweptArea* innerhalb der SweptSolid-Geometrie untersucht werden. Im Abschnitt 4.2.1 wird dies anhand eines Programmentwurfs dargestellt.

Eine implizite SweptSolid-Geometrie kann durchaus mehr Rechenleistung beanspruchen als eine explizite BRep-Geometrie. In der BRep liegen geometrische Elemente explizit vor, daher müssen diese zur Darstellung des Modells nicht berechnet werden. Aus diesem Grund wird im Abschnitt 4.2.2 die Transformierbarkeit von SweptSolid-Körpern in BRep-Körper, welche nicht auf einem Konstruktionsbaum basieren, anhand eines Programmentwurfs dargestellt.

Aus einer Clipping-Geometrie eine auf Extrusion basierende Repräsentation zu erzeugen ist nicht möglich: ein Volumenkörper kann auf jegliche Art geschnitten werden, daher ist ein Profil nicht ermittelbar, um darauf eine geometrische Repräsentation aufzubauen. Eine BRep-Geometrie kann dagegen erzeugt werden. Zum Beispiel könnte der erste Operand der Differenz von einem SweptSolid in einen BRep-Körper transformiert werden. Anschließend müssten die Punkte im Halbraum manipuliert werden. Ein Weg könnte sein, die Punkte über die Schnittgeraden der Seitenflächen mit der Schnittebene zu manipulieren. Auf diese Transformation wird in dieser Arbeit nicht weiter eingegangen, da diese Aufgabe komplexer ist und gesondert betrachtet werden sollte.

Die BRep-Geometrien bestehen aus expliziten Punkten. Aus diesen automatisiert eine geometrische Form für eine Extrusions-Geometrie zu ermitteln ist nicht standardisiert möglich. In eine Clipping-Geometrie zu transformieren wäre dagegen dann denkbar, wenn ein umschließendes Primitiv als Basiselement erstellt werden kann und die resultierende Form durch Schnittebenen erzeugt wird. Auf diese Transformation wird im Folgenden nicht weiter eingegangen, da sie ebenso eine höhere Komplexität aufweist. In einer eigenen Arbeit könnten die Methoden zur Findung der Schnittebenen untersucht werden.

4.2 Transformationsentwürfe

Die Analyse der Transformationseigenschaften der geometrischen Repräsentationen fließt in Entwürfe für Algorithmen, welche eine automatisierte Transformation bezwecken. Eine händische Transformation sollte möglichst nicht angestrebt werden. In einem Projekt wird eine große Menge an Bauelementen erzeugt, welche eine STEP-Datei schnell sehr umfangreich werden lassen. Die Entwürfe der Transformationen setzen voraus, dass die Eingabe ein STEP-File ist. Die Instanzen müssen anhand ihrer Klassennamen erkannt und ausgewertet werden. Durch die immer gleiche Struktur der Instanziierung sollten die Attribute einlesbar sein. Als Ausgabe sind Instanzen vorgesehen, die direkt in die STEP-Datei geschrieben werden. Dieser Vorgang wird im nächsten Unterpunkt kurz erläutert.

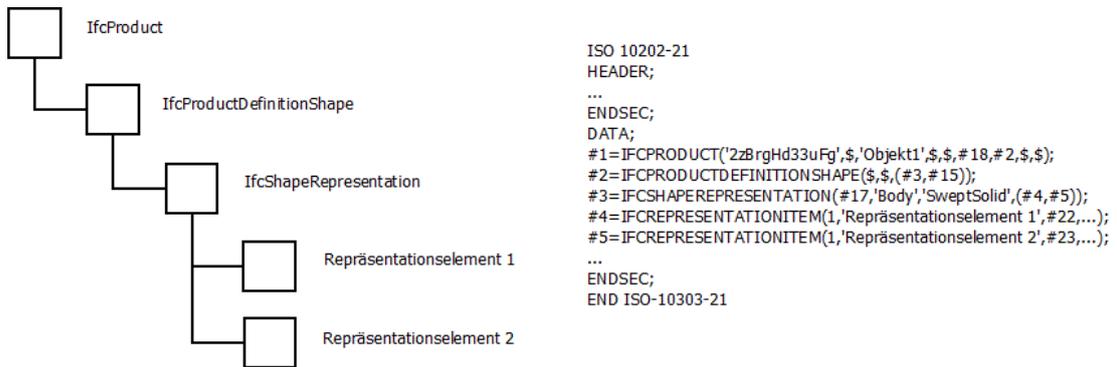
Unabhängig von der geometrischen Repräsentation können gekrümmte Linien durch Polygonzüge approximiert werden. Das dies in der Praxis durchaus vorkommt zeigt das Beispiel der Halbkreiswand in Abschnitt 3.2.1. Wenn die Kurve in eine ausreichend große Anzahl von Geraden unterteilt wird, bleibt der Flächeninhalt und die Form des Profils annähernd identisch. Wie eine Approximation einer Kurve als Poly-

linie funktionieren kann, wird im letzten Unterpunkt *Approximation* dargestellt. Mit diesen Grundlagen werden in den Abschnitten 4.2.1 und 4.2.2 zwei mögliche Transformationen beispielhaft entworfen.

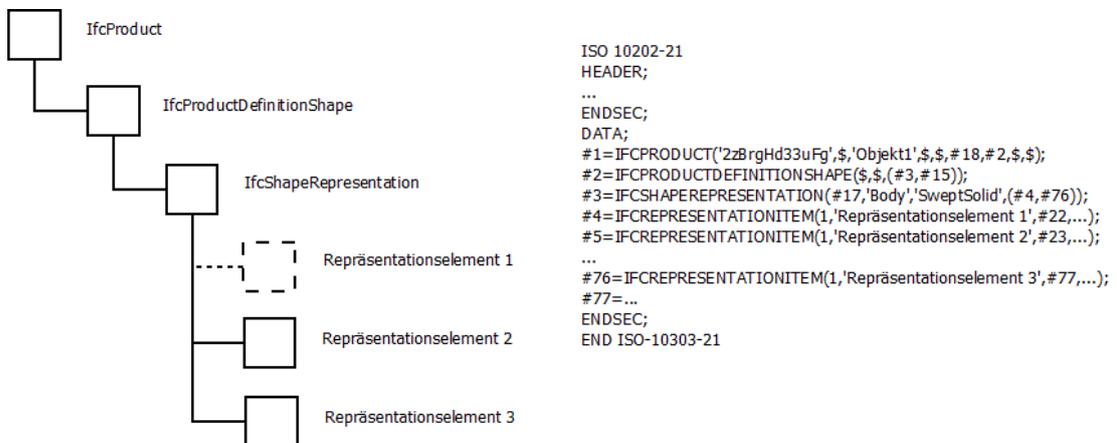
Umformung eines STEP-Files In den STEP-Files sind die Instanzen in Textform festgehalten. Die Instanzen referenzieren einander durch ein “#” gefolgt von der eindeutigen Identifikationsnummer. Die Identifikationsnummern beginnen in jeder Datei mit #1 und werden mit jeder neuen Instanz erhöht. Um die Verlinkung einer Instanz zu ändern muss die alte Verlinkung entfernt und durch eine neue ersetzt werden.

Wenn ein Attribut einer Instanz aus einer Liste von referenzierten Objekten besteht, kann ein weiteres Objekt gegebenenfalls hinzugefügt werden, anstatt nur ein anderes zu ersetzen. Dies ist zuvor in der Dokumentation des Datenmodells zu prüfen, da Listen sowohl begrenzt, als auch unbegrenzt vorliegen können.

Eine neue Verlinkung erfolgt im Normalfall mit einer neu eingeführten Instanz. Das heißt, am Ende der Data Section wird mit der nächst höheren Identifikationsnummer eine neue Instanz beschrieben. In Abbildung 4.1 wird die Bearbeitung eines STEP-Files, so wie sie in den anschließenden Algorithmen vorgesehen ist, dargestellt.



(a) Produktstruktur und Physical File vor der Transformation



(b) Produktstruktur und Physical File nach der Transformation

Abbildung 4.1: Beispiel einer abstrakten und gekürzten Transformierung

Approximation Die Approximierung einer Kurve als Polylinie kann durch Einsetzen in die Gleichung, zum Beispiel in der parameter Darstellung, der jeweiligen Kurve erfolgen. Gegeben sei ein Kreis mit dem Radius r und dem Parameter t mit ($0 \leq t \leq 2\pi$):

$$\vec{x}(t) = \begin{pmatrix} r \cos t \\ r \sin t \end{pmatrix} \quad (4.1)$$

([Merziger *et al.*, 2010], S.21). Oder eine Ellipse mit der großen Halbachse a in X-Richtung und der kleinen Halbachse b in Y-Richtung:

$$\vec{x}(t) = \begin{pmatrix} a \cos t \\ b \sin t \end{pmatrix} \quad (4.2)$$

([Merziger *et al.*, 2010], S.28) . In Form von zwei Gradzahlen wird die Form zu einer Kurve gestutzt. Diese zwei Trimpunkte müssen als Start und Endpunkt der Polylinie verwendet werden. Um diese in die parameter Darstellung einzusetzen ist ein Umrechnen zwischen Grad- und Bogenmaß erforderlich:

$$\frac{\alpha}{180} = \frac{t}{\pi} \quad (4.3)$$

Der Winkel α kann mit diesem Zusammenhang in eine Länge t (und umgekehrt) überführt werden. Der Kreis- bzw. Ellipsenbogen kann in n -Teilstrecken geteilt werden, indem der Kreisbogen vom ersten Trimpunkt, in n -Schritten, bis zum zweiten Punkt abgelaufen wird. Jeder Schritt erzeugt durch Einsetzen der Länge t in Gleichung 4.1 oder 4.2 einen Koordinatenpunkt. Die Koordinatenpunkte bilden dann die sortierte Liste der Polylinie *IfcPolyline*.

4.2.1 Transformation der SweptArea

Das Attribut *SweptArea* von *IfcExtrudedAreaSolid* bzw. *IfcRevolvedAreaSolid* kann unterschiedlich repräsentiert werden. Die Betrachtung der Halbkreiswand (Abschnitt 3.2.1) und der Öffnung in einer Wand (Abschnitt 3.2.1) haben dies verdeutlicht.

Im letzteren Beispiel wurde ein Rechteck durch *IfcParameterizedProfileDef* parametrisch und durch *IfcArbitraryClosedProfileDef* mit Kurven dargestellt. Die Überführung eines parametrischen Profils in ein zusammengesetztes Profil ist, wie in der Einführung bereits erwähnt, dann sinnvoll, wenn der Software des Empfängers keine, oder nur eine begrenzte Auswahl, an parametrischen Profilen zur Verfügung steht.

Insgesamt gibt es 14 Subtypen vom Typ *IfcParameterizedProfileDef*. Diese können umgewandelt werden, indem die Außenlinien der angebotenen Profile durch Subklassen von *IfcCurve* beschrieben werden. Dafür müssen die benötigten Items in einer Reihenfolge angeordnet werden, sodass sie ein geschlossenes Profil bilden. In Tabelle 4.1 sind die Klassen gelistet, deren Instanzen als geometrische Elemente für eine alternative Profilerstellung nötig sind. Die Klassen *IfcBSpline*, *IfcPolyline*, *IfcCircle* und *IfcEllipse* sind die grundlegend nutzbaren geometrischen Elemente für geschlossene Kurven.

	IfcBSplineCurve	IfcPolyline	IfcCircle	IfcEllipse
IfcCircleProfileDef*				
IfcEllipseProfileDef*				
IfcRectangleProfileDef*				
IfcTrapeziumProfileDef*				
IfcRoundedRectangleProfileDef				
IfcCircleHollowProfileDef				
IfcRectangleHollowProfileDef				
IfcAsymmetricShapeProfileDef*				
IfcCShapeProfileDef*				
IfcIShapeProfileDef*				
IfcLShapeProfileDef*				
IfcTShapeProfileDef*				
IfcUShapeProfileDef*				
IfcZShapeProfileDef*				

Tabelle 4.1: Benötigte Items zur alternativen Darstellung des jeweiligen Subtyps von *IfcParameterizedProfileDef* (* - direkte Subklasse)

Wie die Tabelle 4.1 zeigt, wird *IfcBSpline* nicht benötigt um vom IFC4-Format unterstützte Profile zu ersetzen. Die Parameter der Subklassen von *IfcParameterizedProfileDef* können im Anhang A.3 genauer betrachtet werden. Die ersten vier Profile der Tabelle 4.1 können mit nur einer Instanz des jeweiligen geometrischen Elements repräsentiert werden. Die restlichen Profile müssen durch aneinander gereihete geometrische Elemente repräsentiert werden. Dafür müssen diese in einer Liste in *IfcCompositeCurve* referenziert werden. Für Teilstücke eines Kreises muss *IfcCircle* als *BasisCurve* referenziert werden.

Kurven können wie zu Beginn beschrieben durch Stützpunkte und deren geradlinigen Verbindung als *IfcPolyline* approximiert werden. Dieser Fall ist in der Tabelle 4.1 jedoch nicht berücksichtigt, da er als separate Transformation betrachtet wird.

Die Parameter der parametrischen Profile müssen ausgelesen und für die Anpassung der Koordinaten und Trimpunkte der geometrischen Elemente genutzt werden. Für die Instanz einer *IfcPolyline* sind lediglich Koordinaten nötig, welche im lokalen Koordinatensystem von *IfcExtrudedAreaSolid* platziert werden. Die Instanzen *IfcCircle* und *IfcEllipse* erfordern dagegen ein eigenes lokales Koordinatensystem im übergeordneten lokalen Koordinatensystem der *IfcExtrudedAreaSolid*, welches initialisiert werden muss. Das lokale Koordinatensystem des parametrischen Profils wird nach der Transformation nicht mehr benötigt.

Eine Möglichkeit zur Transformierung ist für jedes parametrische Profil eine eigene Transformation zu implementieren. Dies bedeutet, dass je Profil eine Methode geschrieben werden muss. Der Aufwand der einfachen Profile (Tabelle 4.1, Zeile 1-4) ist als überschaubar einzuschätzen, da sie mit je einer Instanz auskommen können. Die Instanziierung der geometrischen Elemente *IfcPolyline* und *IfcCircle* ist redundant, sodass die Reihenfolge der Teilkurven für die Profile in Tabelle 4.1, Zeile 5-14 das Hauptaugenmerk bildet. Die Teilkurven werden von *IfcCompositeCurveSegment*-Instanzen referenziert und in einer *IfcCompositeCurve* zusammengefasst. Aus diesen Überlegungen lässt sich schließen, dass eine automatische Transformation ermöglicht werden kann.

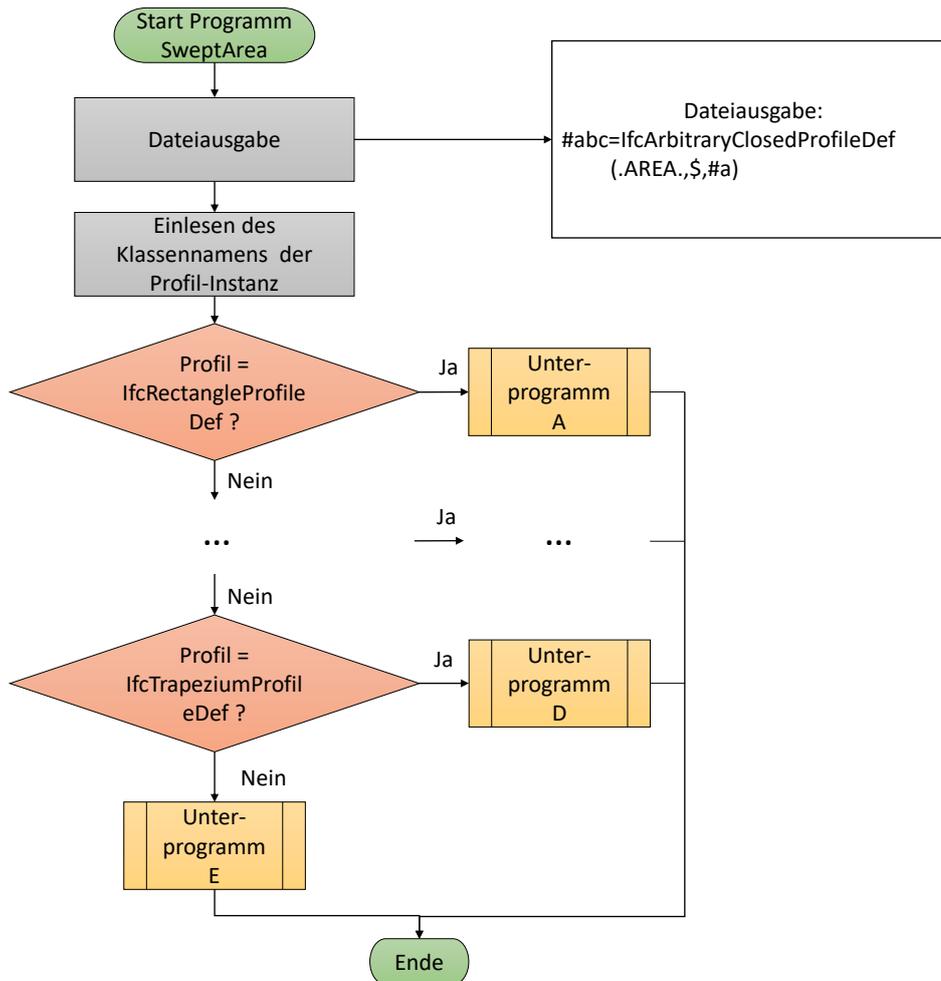


Abbildung 4.2: Auszug aus dem Flussdiagramm des Hauptprogramms *SweptArea* (vgl. Abb. A.24)

In Abb. 4.2 ist ein Auszug aus dem Hauptprogramm einer solchen Transformation abgebildet. Auf den Start des Programms folgt die Instanziierung einer neuen Repräsentation des Profils durch eine Dateiausgabe. Die Identifikationsnummer der neu eingeführten *IfcArbitraryClosedProfileDef*-Instanz muss in den Attributen von *IfcExtrudedAreaSolid* die Nummer des parametrischen Profils ersetzen.

Aus dem STEP-File wird anschließend die Klasse des Profils ausgelesen und dem Programm als Wert übergeben. Dann wird abgeglichen, welches Profil vorliegt und das entsprechende Unterprogramm gestartet. Die Unterprogramme A, B, C und D sind sich sehr ähnlich und können im Anhang A.4 eingesehen werden. Das Prinzip dieser Unterprogramme ist im Flussdiagramm des Unterprogramms A in Abb. 4.3 dargestellt.

Die formgebenden Parameter der Profile sind dem jeweiligen Subtyp von *IfcParameterizedProfileDef* im STEP-File als Attribute zugeordnet und im Anhang A.3 grafisch dargestellt. Die vorgegebene Reihenfolge der Attribute ist der Dokumentation von *buildingSmart* ([Liebich *et al.*, 2015-2017]) zu entnehmen.

Im Unterprogramm A sind dies die Parameter *XDim* und *YDim*, welche aus dem STEP-File vor der Transformation aus der Instanz des *IfcRectangleProfileDef* zu entneh-

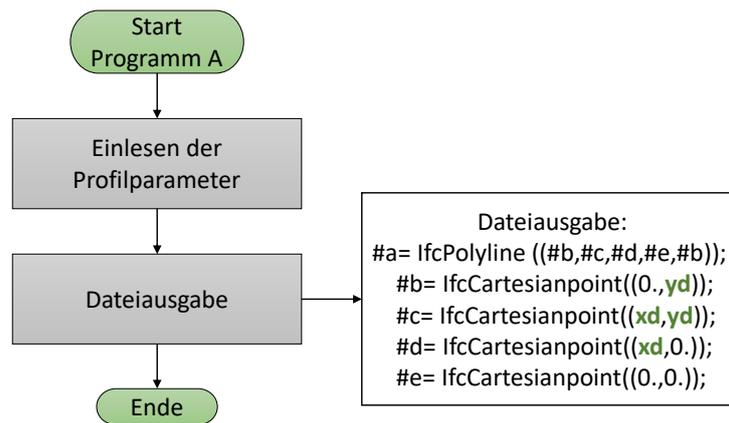


Abbildung 4.3: Flussdiagramm des Unterprogramms A zur Umformung einer *IfcRectangleProfileDef* in eine Polylinie

men sind: $\#... = \text{IFCRECTANGLEPROFILEDEF}(\text{ProfileType}, \text{ProfileName}, \text{Position}, \text{XDim}, \text{YDim})$.

Die Größen werden im Programm in den Variablen xd und yd gespeichert und für die Instanziierung der neuen Repräsentation genutzt (siehe Abb. 4.3, Dateiausgabe). Die *IfcCartesianpoint*-Instanzen sind im lokalen Koordinatensystem des *IfcExtrudedAreaSolid* zu setzen, welches in der unteren linken Ecke des Profils seinen Ursprung hat. Der Startpunkt der Polylinie wird auch als Endpunkt in *IfcPolyline* referenziert. Die Punkte dazwischen müssen in einer Reihenfolge referenziert werden. In der Abb. 4.3 wurden die Punkte im Uhrzeigersinn abgelaufen.

Das Unterprogramm *E* wird aufgerufen, wenn es sich nicht um ein einfaches Kreis-, Rechteck-, Ellipsen- oder Trapezprofil handelt. Der Ablauf des Unterprogramms *E* (siehe Abb. 4.4), ist abhängig vom vorliegenden Profil. In diesem Programm wird eine zusammengesetzte Kurve *IfcCompositeCurve* erstellt. Je nach vorliegendem Profil ist im Programm festgelegt, wie viele Teilkurven *IfcCompositeCurveSegment* erstellt werden. In der Implementierung wird ein Startpunkt festgelegt und ausgehend von diesem werden im Uhrzeigersinn die Teilkurven erstellt werden. Es kann beispielsweise immer in der oberen linken Ecke begonnen werden.

In der Reihenfolge, in der die Teilkurven aneinanderstoßen, werden dann je nach Art der Kurve die Unterprogramme *Circle* und *Polyline* geöffnet. Die zuvor eingelesenen Parameter werden den Unterprogrammen zur Erstellung der Teilkurven übergeben. In Abhängigkeit von den Parametern werden Start- und Endpunkt der Teilkurve übergeben, bzw. für Kreisbögen zusätzlich ein Radius, der Ursprung und die Ausrichtung des lokalen Koordinatensystems. Mit den übergebenen Parametern wird dann mit jedem Aufruf die Instanziierung der Teilkurven realisiert, wie in Tabelle 4.2 gelistet.

Mit diesem Programmentwurf ist festgelegt, dass das Rechteck- und das Trapezprofil in eine Polylinie und nicht in einzelne Linien gewandelt wird. Es könnte jedoch von Nutzen sein, auch diese Profile in Teilstegmente zu überführen. In diesem Fall kann nach der Abfrage des Profils eine Abfrage erfolgen, welche Repräsentation bevorzugt wird. Danach kann entschieden werden, ob Unterprogramm *A* bzw. Unterprogramm *D*

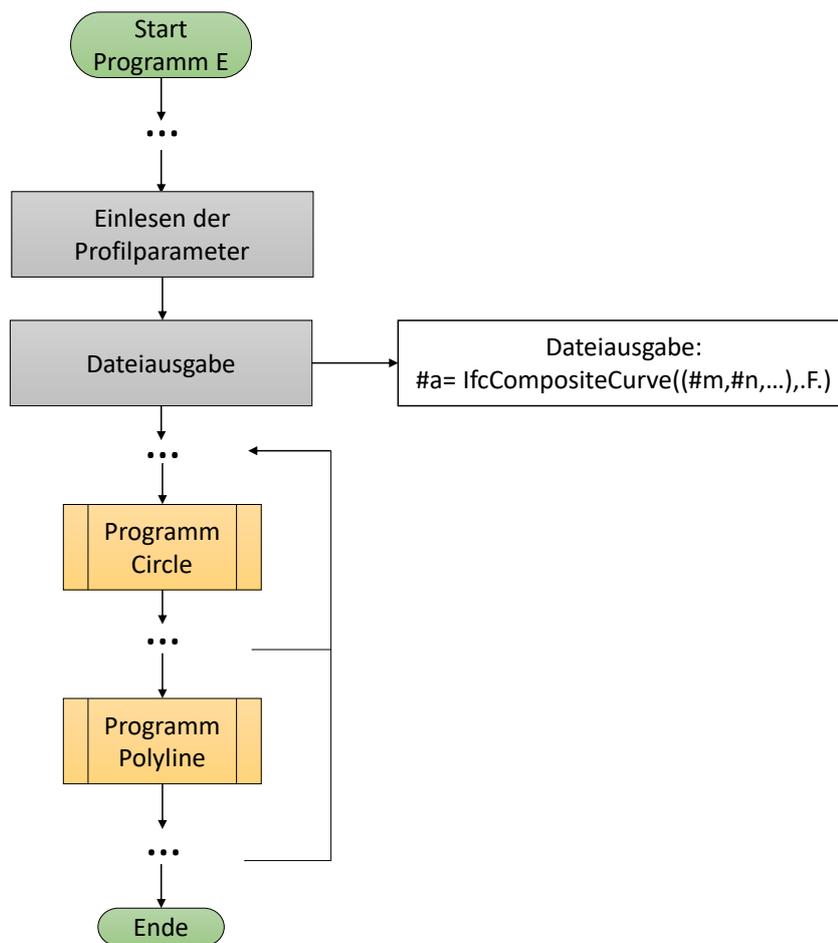


Abbildung 4.4: Flussdiagramm des Unterprogramms E zur Umformung von parametrischen Profilen in Teilkurven

	Polyline	Circle
Eingabe	P1 - Startpunkt (x,y) P2 - Endpunkt (x,y)	P1 - Startpunkt in Grad P2 - Endpunkt in Grad r - Radius K - Ursprung des lokalen Koordinatensystems v - Richtung der lokalen X-Achse
Ausgabe	#n= IfcCompositeCurveSegment(.CONTINUOUS.,.T.,#a); #a= IfcPolyline((#b,#c)); #b= IfcCartesianpoint(P1); #c= IfcCartesianpoint(P2);	#m= IfcCompositeCurveSegment(.CONTINUOUS.,.T.,#a); #a= IfcTrimmedCurve(#b, P1 , P2 ,.F.,.PARAMETER); #b= IfcCircle(#c,r); #c= IfcAxis2Placement2D(#d,#e); #d= IfcCartesianpoint(K); #e= IfcDirection(v);

Tabelle 4.2: Eingabewerte und Dateiausgabe der Unterprogramme Polyline und Circle

<pre> 1 #84= IFCEXTRUDEDAREASOLID(#76,#85,#92,3000.); 2 #76= IFCRECTANGLEPROFILEDEF(.AREA.,\$,#79,10000.,115.); 3 #79= IFCAXIS2PLACEMENT2D(#80,#82); 4 #80= IFCCARTESIANPOINT((5000.,57.5)); 5 #82= IFCDIRECTION((1.,0.)); 6 #85= IFCAXIS2PLACEMENT3D(#86,#88,#90); 7 #86= IFCCARTESIANPOINT((0.,0.,0.)); 8 #88= IFCDIRECTION((0.,0.,1.)); 9 #90= IFCDIRECTION((1.,0.,0.)); 10 #92= IFCDIRECTION((0.,0.,1.)); </pre>	<pre> 1 #84= IFCEXTRUDEDAREASOLID(#192,#85,#92,3000.); 2 #192= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,\$,#193); 3 #193= IFCCOMPOSITECURVE((#194,#195,#196,#197),.F.); 4 #194= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#198); 5 #198= IFCPOLYLINE((#199,#200)); 6 #199= IFCCARTESIANPOINT((0.,115.)); 7 #200= IFCCARTESIANPOINT((10000.,115.)); 8 #195= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#201); 9 #201= IFCPOLYLINE((#202,#203)); 10 #202= IFCCARTESIANPOINT((10000.,115.)); 11 #203= IFCCARTESIANPOINT((10000.,0.)); 12 #196= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#204); 13 #204= IFCPOLYLINE((#205,#206)); 14 #205= IFCCARTESIANPOINT((10000.,0.)); 15 #206= IFCCARTESIANPOINT((0.,0.)); 16 #197= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.F.,#207); 17 #207= IFCPOLYLINE((#208,#209)); 18 #208= IFCCARTESIANPOINT((0.,0.)); 19 #209= IFCCARTESIANPOINT((0.,115.)); </pre>
(a) Original Repräsentation	(b) Händisch angepasste Repräsentation

Abbildung 4.5: Transformation eines Rechtecks in eine *IfcCompositeCurve*

oder Unterprogramm *E* verwendet wird. Das Unterprogramm *E* muss dann um diese Ursprungsprofile erweitert werden.

Das Vorgehen wurde händisch an einem IFC-Modell einer Wand getestet (siehe Abb. 4.5). Die originale Repräsentation wurde aus dem Beispiel *Wand* entnommen und entsprechend dem Unterprogramm *E* umgeschrieben. Es fällt auf, dass die Punkte sich wiederholen, also mehrere Instanzen mit dem gleichen Objekt auftreten (z.B.: Abb. 4.5b, Zeile 7 und 10). Das umgeformte Profil erzeugt im Viewer das gleiche Ergebnis.

4.2.2 Transformation der SweptSolid-Geometrie

Die Oberflächendarstellung BRep ist sehr vielfältig einsetzbar und wird besonders zur Visualisierung genutzt. CSG und Verschiebegeometrien (bzw. Volumenmodelle) müssen für grafische Aufgaben in Oberflächenmodelle umgewandelt werden, da für diese Aufgabe hauptsächlich Flächeninformationen erforderlich sind (vgl. [Schiele, 2012], S.55). Die Transformation kann außerdem, wie zuvor erwähnt, nötig sein, wenn die Software die Konstruktionsschritte des Körpers nicht deuten und nachbilden kann.

Für die Transformation von einem SweptSolid-Körper in einen BRep Körper stehen im IFC zwei geometrische Repräsentation zur Verfügung. Im Kontext dieser Arbeit soll die einfache BRep mit planaren Flächen und geraden Kanten untersucht werden. Ein Volumenkörper mit geraden Flächen kann in eine einfache BRep-Struktur umgewandelt werden, indem die Eckpunkte des Ursprungsobjektes gesammelt werden.

Wenn nur einfache BRep-Methoden betrachtet werden, können für die *SweptArea* des SweptSolid-Körpers nur Profile mit geraden Außenkanten genutzt werden. Profile mit gekrümmten Außenkanten würden, durch die Extrusion entlang einer Achse, gekrümmte Flächen erzeugen. Diese müssten für die einfach BRep-Struktur in Polylinien gewandelt werden. Auch gekrümmte Extrusionspfade und Rotationen führen zwangsläufig zur erweiterten BRep Repräsentation und gehören daher aus dem selben Grund nicht zu dieser Transformation.

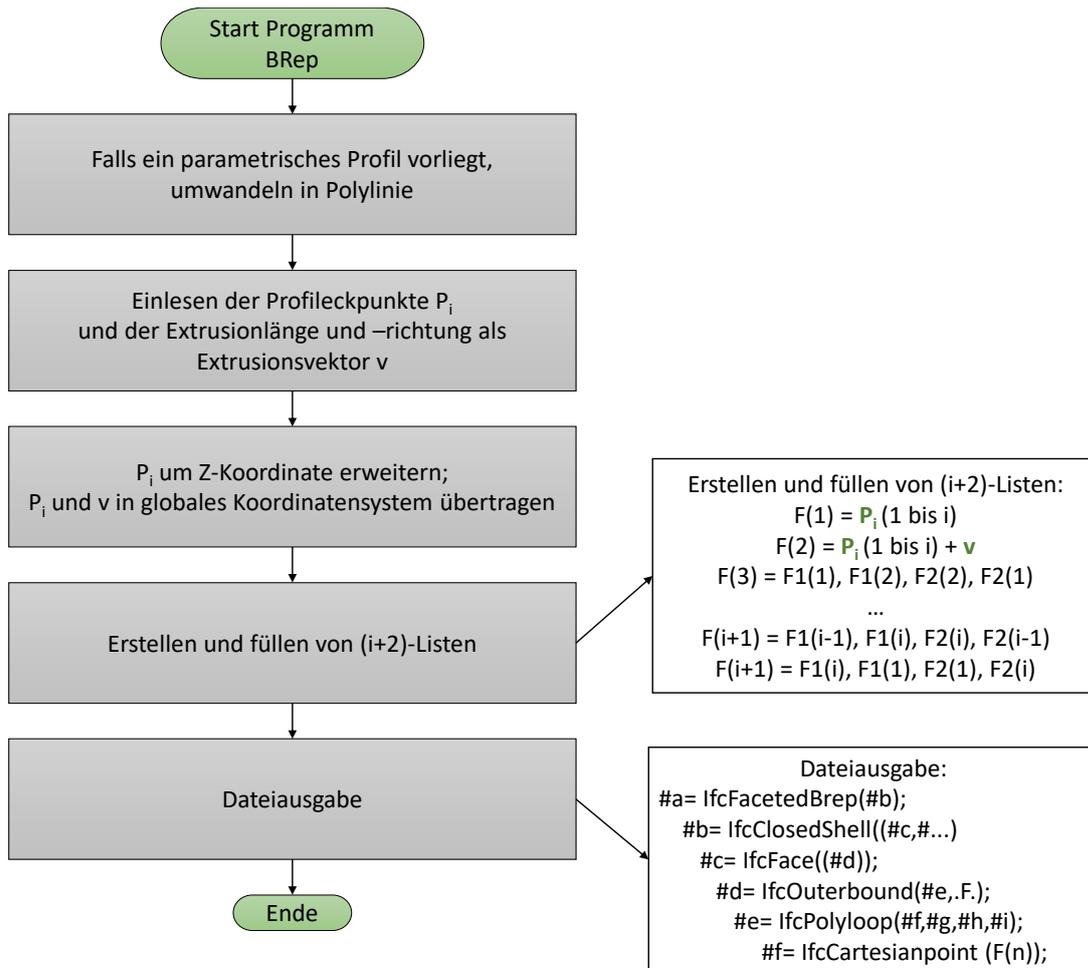


Abbildung 4.6: Flussdiagramm des Hauptprogramms BRep

Die BRep-Objekte beinhalten explizit alle Eckpunkte als Koordinaten in ihrem Modell. Diese sind aus der SweptSolid-Datenstruktur ablesbar, wenn das Profil aus Polylinien besteht. Wenn ein parametrisches Profil vorliegt, kann dieses im ersten Schritt in Polylinien umgewandelt werden (siehe Abschnitt 4.2.1). Die benötigte Punktmenge besteht zur Hälfte aus den referenzierten *IfcCartesianpoints* der *IfcPolyline* des Profils. Die andere Hälfte muss durch eine Verschiebung der Punkte um die Extrusionslänge entlang der Extrusionsrichtung ermittelt und als *IfcCartesianpoints* initiiert werden. Die Grundmenge an Punkten muss anschließend in Listen einsortiert werden. Jeder Oberfläche werden die Eckpunkte in einer Reihenfolge zugeordnet, in der sie abgelaufen werden können.

Aus diesen Überlegungen lässt sich schließen, dass eine automatische Transformation für den eingegrenzten Fall ermöglicht werden kann. In Abb. 4.6 ist das Hauptprogramm der Transformation abgebildet. Im ersten Schritt wird das Profil in eine einzelne Polylinie umgewandelt, wenn dieses als parametrisches Profil vorliegt. Anschließend können die Punkte der Polylinie ausgelesen werden.

Aus der Richtung und der Länge der Extrusion wird ein Verschiebungsvektor berechnet. Die Punkte liegen als *IfcCartesianpoints* vor und die Extrusionslänge und -richtung

sind Attribute von *IfcExtrudedAreaSolid*. Die Richtung ist als *IfcDirection* referenziert. Zu beachten ist, dass die Referenzierungsreihenfolge der Punkte in der Polylinie beibehalten werden muss.

Für die BRep sind keine lokalen Koordinatensysteme nötig und alle Punkte sind im dreidimensionalen globalen Koordinatensystem platziert. In der SweptSolid-Repräsentation liegen die Punkte der Profile jedoch in einem lokalen zweidimensionalen Koordinatensystem, genauer in einer X-Y-Ebene. Deshalb ist eine Ergänzung um eine dritte Dimension nötig, sowie anschließend eine Transformation der Profilmunkte und des Verschiebungsvektors in das globale Koordinatensystem.

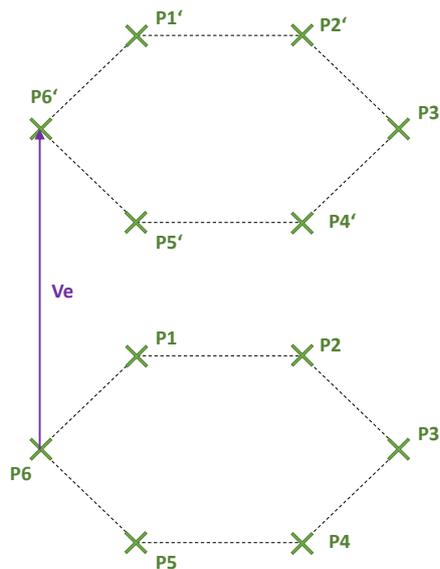


Abbildung 4.7: Erhaltene Punktmenge P_i aus der *SweptArea* und zu erzeugende Punktmenge P_i' durch Verschieben der Punktmenge

Die gesamte Punktmenge ist die Grundlage der BRep-Hierarchie. Aus je vier Eckpunkten, zusammengefasst in einer *IfcPolyloop*, muss eine Außengrenze *IfcOuterbound* pro Außenfläche *IfcFace* der gesamten Außenschale *IfcClosedShell* erstellt werden. Im Gegensatz zur *IfcPolyline* ist in einer *IfcPolyloop* der Startpunkt nicht gleich dem Endpunkt. Durch die Beibehaltung der Reihenfolge der Punkte aus der *IfcPolyline* können die Punkte aus der gesamten Punktmenge (vgl. Abb. 4.7) in Listen entsprechend der Außenflächen sortiert werden.

Es gibt immer zwei "horizontale" Flächen und genauso viele Seitenflächen wie es Eckpunkte im Basis-Profil gibt. In Abbildung 4.6 ist das Schema zur Sortierung der Punkte in Listen je Seitenfläche notiert. Die Liste F(1) beinhaltet alle Punkte aus der *SweptArea* und die Liste F(2) alle verschobenen Punkte. Die restlichen Flächen beziehen die Eckpunkte aus diesen zwei Listen nach einem einheitlichen Schema und halten dadurch die gleiche Reihenfolge der Punkte ein.

Mit den sortierten Listen kann dann abschließend die Geometrie als BRep instanziiert werden (wie in Abb. 4.6 in der Ausgabe). Außerdem muss die Originaldatei noch umgeändert werden, indem die Instanz *IfcShaperepresentation* angepasst wird, wie im Unterpunkt Umformen von STEP-Files beschrieben.

Die händische Überführung einer SweptSolid-Darstellung in eine BRep erwies sich als zu komplex und konnte in dieser Arbeit nicht zielbringend händisch nachvollzogen werden.

5 Generischer Leitfaden

5.1 Überblick

In einem BIM-Projekt arbeiten verschiedene Parteien miteinander. Die Nutzung eines gemeinsamen IFC-Modells erfordert gemeinsame Absprachen im Vorfeld. Zwischen den Projektbeteiligten müssen Vereinbarungen über die Modellinhalte, die Modellqualität und die Prozessabläufe geklärt werden (vgl. [Borrmann *et al.*, 2015], S.11). Der Austausch des Modells, insbesondere der Geometrie, kann mit IFCs auf unterschiedliche Arten verwirklicht werden. Dieser Aspekt erfordert eine Übereinkunft über die passenden Repräsentationen.

In dieser Arbeit wurde vorab ein Überblick über einige der gängigen geometrischen Repräsentationen gewonnen. Es wurde einerseits eine explizite Geometriebeschreibung verwendet, welche den Vorteil bieten alle geometrischen Informationen, wie der Koordinaten und die Form der Kurven- und Ebenen, direkt zu beschreiben. Das Einlesen des IFC-Modells kann mit diesen Darstellungsformen ohne weitere Berechnungen vorstatten gehen. Das STEP-File wird aber in Projekten mit der Zeit unübersichtlich und benötigt nach und nach mehr Speicherplatz.

Andererseits wurden zwei implizite Repräsentationen verwendet, welche den Vorteil haben leicht nachvollziehbar und einfach anpassbar zu sein. Für den Import sind Berechnungen nötig, um die Geometrie im CAD-Programm erzeugen.

Die Achsgeometrie (vgl. 3.3.1) ist eine Mischung aus expliziter und impliziter Geometrie. Darin liegen die geometrischen Informationen der Achse explizit vor und in Kombination mit einem Materialprofil kann daraus eine dreidimensionale Geometrie generiert werden.

Allgemein ist neben den geometrischen Repräsentationen aus gängigen CAD-Programmen im Bauwesen ein Trend zur parametrischen Modellierung zu erkennen. So nennt Borrmann *et al.* die Parametrik als zu erwartende Optimierung der Modellkonsistenz (vgl. [Borrmann *et al.*, 2015], S.126). In einigen Projekten wird bereits gezielt Software aus dem Maschinenbau eingesetzt, welche die parametrische Modellierung nutzen (vgl. [Lukas *et al.*, 2008] S. 6-8; [Scherer & Schapke, 2014], S.172f). Bei dieser Modellierung werden Geometrien und Eigenschaften mit Parametern beschrieben.

Es empfiehlt sich eine Einigung aller Beteiligten über die geometrische Repräsentation. Im Anschluss können diese detailliert in ihrer Form festgelegt werden. Aus diesem Grund werden in den folgenden Abschnitten 5.2 und 5.3 Hilfestellungen für Absprachen erläutert.

5.2 Eignung der Darstellungsformen

Mit dem Standard IFC4 konnten die in dieser Arbeit untersuchten Bauelemente korrekt dargestellt werden. Die Unterschiede zwischen den Exports aus den zwei gewählten CAD-Programmen waren nicht gravierend, da in den meisten Fällen das gleiche Element entstand. Die dabei verwendeten gängigen geometrischen Repräsentationen sind

	BRep	Achsgeometrie	SweptSolid-Geometrie	Clipping-Geometrie
Art	explizit	explizit/implizit	implizit	implizit
Informationsgehalt	geometrische Informationen direkt enthalten	geometrische Information über Achse direkt enthalten	geometrische Information durch Profil und Extrusionspfad indirekt enthalten	geometrische Information durch SweptSolid und Schnittebene indirekt enthalten
Darstellbare Körper	jegliche Formen	Körper mit Achse	Körper mit Achse und Profil	Körper mit Achse und Profil

Tabelle 5.1: Übersicht über die gängigen geometrischen Repräsentationen

in Tabelle 5.1 zur Übersicht zusammengefasst. Die Festlegung der Darstellung kann sich an der benötigten Informationen oder den Softwareprodukten der Projektpartner orientieren.

Je nach benötigten Informationen kann die Anwendung der Achsgeometrie sinnvoll sein. Die SweptSolid-Geometrie kann jedoch mehr Körperformen repräsentieren als die Achsgeometrie. Daher kann die SweptSolid-Geometrie bevorzugt werden. Die in den Untersuchungen beobachtete Kombination aus beiden Geometrien kann durchaus von Nutzen sein. Den Objekten wird durch die Kombination eine korrekte Körperform zugeordnet und dieser mit der Achsgeometrie eine Referenzlinie bzw. Ausrichtung hinzugefügt. Beide Repräsentationen sind aufgrund ihrer Struktur (vgl. Abschnitte 3.3.1,3.3.2) nicht für komplexere Körperformen anwendbar. Für einfache Geometrien wie beispielsweise Wände, Geschossdecken, Stützen oder Unterzüge sind beide Repräsentationen gut geeignet.

Mit der Clipping-Geometrie wird die Darstellungsmöglichkeit von SweptSolid-Körper erweitert, indem SweptSolids durch Halbräume ein oder mehrfach beschnitten werden (vgl. Abschnitt 3.3.3). Diese sollte daher zusätzlich zur SweptSolid-Geometrie benutzt werden, um zusätzliche Formen zu erzeugen. Komplexere Formen mit gekrümmten Oberflächen, wie zum Beispiel NURBS-Flächen, sind jedoch trotzdem nicht zu erzeugen. Es können Geometrien für angepasste Bauelemente erstellt werden, wie zum Beispiel eine Giebelwand.

Die BRep bleibt aus diesem Grund für die Abbildung komplexerer Körper notwendig. Sie könnte jedoch von parametrischen Repräsentationen abgelöst werden, sobald diese dem Datenmodell hinzugefügt werden. Diese Repräsentation eignet sich neben komplexen Formen auch in der Entwurfsphase oder für visuelle Zwecke.

In der Betrachtung der Exports wurde bei der Wahl der Repräsentation eine Abhängigkeit von dem CAD-Produkt aufgezeigt. Die angewendete Repräsentation ist abhängig davon, welche von den Herstellern unterstützt wird. Genauso verhält es sich während

des Importvorgangs.

Die Untersuchung der Transformationseigenschaften in Kapitel 4 hat gezeigt, dass einige geometrische Repräsentationen prinzipiell überführt werden können. Dies eröffnet den Projektteilnehmern die Möglichkeit sich nicht auf eine geometrische Repräsentation beschränken zu müssen. Die geometrische Repräsentation kann nach dem Export in eine andere überführt werden, somit wird der Anwender unabhängiger von den Softwareprodukten. Dafür ist zu untersuchen, ob das IFC-Modell um die Transformation erweitert werden könnte oder ob ein Überschreiben der ursprünglichen Repräsentation nötig wird. Bei mehreren Repräsentationen des gleichen Objekts in einem Modell wird der Aufwand der Modellbearbeitung größer.

5.3 Empfehlungen für Absprachen

Bei der Untersuchung an konkreten Bauelementen wurden bei der Darstellung Variationen festgestellt (vgl. Abschnitt 3.2). Trotz gleicher geometrischer Repräsentation wurden Unterschiede in den IFC-Modellen identifiziert. Vertragspartner in einem BIM-Projekt sollten sich diese bewusst machen und im Vorfeld diskutieren. Folgende Empfehlungen sollen dabei als Leitfaden dienen:

- Krümmungen können unterschiedlich dargestellt werden, entweder explizit als Kurve durch mathematischen Funktionen oder approximiert als Polylinie. Es muss eine Darstellungsart gewählt werden:
 - immer explizit als Kurve
 - immer approximiert durch Polylinie
- Bei der Achsgeometrie kann die Lage der Achse selbst variieren. Je nach Softwareprodukt kann die Lage im Programm beeinflusst werden. Für die Lage gibt es z.B. folgende Absprachen:
 - immer in der Mitte des Elements
 - bei mehrschichtigen Wänden immer an der Außenkante der innersten Schicht
 - bei Außenwänden immer an der Innenseite und Innenwänden immer mittig
- Die Achsgeometrie wird nicht nur für Standardbauteile genutzt, sondern z.B. auch für eine abgeschrägte Wand. Aus der Achse und dem Materialprofil kann nur ein Standardbauteil generiert werden. Die zu repräsentierenden Elemente sind festzulegen z.B. durch folgende Absprachen:
 - alle raumbegrenzenden Elemente unabhängig ihrer 3D-Form
 - alle möglichen Elemente
 - nur 3D-Achsgeometrien (mit Achse in der Mitte des Elements)
- Das Extrusionsprofil von SweptSolid-Geometrien kann mit Teilkurven oder parametrisch (wenn es als Subtyp von *IfcParameterizedProfileDef* vorliegt) definiert werden. Es muss eine Vereinbarung getroffen werden:

- immer Teilkurven
- immer paramterisch wenn möglich
- Die BRep kann mit der einfachen (nur Eckpunkte explizit enthalten) oder mit einer erweiterten Methode (auch Kanten und Ebenen explizit enthalten) ausgeführt werden. Es sollte eine Vereinbarung getroffen werden:
 - immer die einfache BRep (gegebenenfalls durch approximieren von Krümmungen)
 - immer die erweiterte BRep

6 Ergebnisse

6.1 Zusammenfassung und Diskussion

Zielsetzung der vorliegenden Arbeit war es einen generischen Leitfaden für den geometrischen Export von Gebäudeelementen in IFC zu erarbeiten. Bestandteil der Arbeit waren auf diesem Weg drei Themenbereiche. Erstens eine Übersicht über die Grundlagen der Geometrie und deren Darstellung, sowie eine Einarbeitung in die IFC.

Zweitens, darauf aufbauend, fand eine Untersuchung des geometrischen Exports mit dem aktuellen IFC4-Format statt. Dabei wurden Arbeiten über Exportuntersuchungen vergangener IFC-Formate betrachtet. Es entstand ein Untersuchungsaufbau in dessen Fokus der Export von ausgewählten Gebäudeelementen aus zwei gängigen CAD-Programmen und die resultierenden STEP-Files standen. Die darin verwendeten geometrischen Repräsentationen wurden anhand der expliziten Gebäudeelemente auf Unterschiede und eventuelle Fehler geprüft und auf ihren Informationsgehalt untersucht. Die geometrischen Exports konnten in dieser Untersuchung gute Ergebnisse vorweisen, jedoch mit kleineren Variationen in der Nutzung der Darstellungsarten.

Im dritten und letzten Themenbereich wurden die geometrischen Repräsentationen auf Transformationseigenschaften untersucht, indem mögliche Transformationen diskutiert wurden. An zwei der genannten Transformationen wurde die Durchführbarkeit der Transformation durch den Entwurf von Algorithmen betrachtet. In diesem Teil der Arbeit wurde deutlich, dass Transformationen im IFC-Format prinzipiell möglich sind. Einige basieren jedoch auf der Entwicklung von Methoden zur Erzeugung zusätzlicher geometrischer Informationen, die zuvor nicht explizit vorlagen.

Die durchlaufenen Arbeitsschritte führten zu einem Leitfaden, welcher von Vertragspartnern in BIM-Projekten oder von CAD-Entwicklern zu Rate gezogen werden kann. Der Leitfaden analysiert und diskutiert die gängigen geometrischen Repräsentationen, führt jedoch auch an, dass die gängigen bereits als die geeignetsten Darstellungformen gesehen werden können. Abschließend wurde auf die Variationen in der Anwendung der Repräsentationen hingewiesen, einschließlich der Empfehlungen für Vereinbarungen.

In der Arbeit wurde sich hauptsächlich auf die, von den gewählten CAD-Programmen genutzten, geometrischen Repräsentationen beschränkt. Dies simuliert die Einschränkung der Softwareanwender in der Praxis, da diese an die Wahl der exportierten Repräsentation gebunden sind. Trotzdem wurden alle standardisierten geometrischen Repräsentationen kurz erläutert, jedoch nicht vertieft. Ein Vergleich mit weiteren möglichen geometrischen Repräsentationen, welche von den gewählten CAD-Programmen nicht verwendet wurden, können den Leitfaden erweitern. Auf eine zukünftige parametrische Darstellungsform wurde hingewiesen.

Die Analyse der geometrischen Repräsentation beschränkte sich auf die erzeugten IFC-Modelle. So konnte der Export ohne Veränderungen durch Importvorgänge betrachtet werden. Für das Mapping während der Importvorgänge wurden nur Annahmen getroffen. Die gewählten CAD-Programme verfügten zum Zeitpunkt März 2017 noch nicht

über die Importfunktion für IFC4-Modelle. Hier besteht noch Potenzial für eine umfangreichere Untersuchung.

Die Transformationsentwürfe wurden im Rahmen dieser Arbeit nicht implementiert, da sie in der Bearbeitungszeit nicht zu bewerkstelligen waren. Die Durchführbarkeit der entwickelten Überführungsalgorithmen blieb dadurch eine Annahme und eine automatische Transformation von geometrischen Repräsentationen bleibt daher noch offen.

6.2 Ausblick

Die im Rahmen dieser Diplomarbeit durchgeführten Untersuchungen konnten erste Erkenntnisse über die Nutzung des IFC4-Formats liefern. Weitere Untersuchungen über den Export, mit komplexeren Gebäudeelementen, sollten jedoch an diese Arbeit angeschlossen werden. Ein größerer Untersuchungsrahmen könnte weiterhin den realen Einsatz simulieren, indem ein IFC-Modell eines ganzen Projekts erzeugt wird. Dies ist mit einem größeren Zeitaufwand verbunden. Teil einer solchen Arbeit könnte ein Experiment sein, in dem mehrere Projektpartner und deren Interaktion miteinander simuliert werden. Der Leitfaden dieser Arbeit kann für diese Projektpartner als Handlungsanweisung oder -grundlage dienen.

Die Einführung der Importschnittstellen von IFC4 steht noch aus. Die gute Qualität des Exports während der Untersuchung in dieser Arbeit lässt vermuten, dass der Import von ähnlicher Qualität ist. Wünschenswert wäre eine Prüfung der Importqualität im Zusammenhang mit den zuvor genannten Überlegungen. Im Zuge dessen kann geprüft werden, ob mit Transformationen der geometrischen Repräsentationen zwischen Export und Import die Interoperabilität verbessert werden kann. Zu diesem Zweck müssen automatische Transformationen konkret implementiert werden.

Weiterhin ist die Entwicklung der parametrischen Modellierung im Bauwesen zu beobachten. buildingSmart und Partner sind bereits seit geraumer Zeit an der Entwicklung einer parametrischen Erweiterung der IFC. In Pilotprojekten konnten bereits parametrische IFC entwickelt werden. Wenn diese zum Standard gehören ist eine weitere Betrachtung denkbar. Darin kann geprüft werden, ob eine Ablösung der BRep als geeignete Darstellungsform in einigen Anwendungsfällen auftreten kann.

Literatur

- [bui, 2008-2017] . 2008-2017. *buildingSMART, International home of openBIM*. Website. Online erhältlich unter <http://www.buildingsmart-tech.org>, zuletzt abgerufen am 14.03.2017.
- [Ste, 2015] . 2015. *STEP Tools, Schema IFC4*. Website. Online erhältlich unter <http://www.steptools.com/stds/ifc/html/>, zuletzt abgerufen am 18.03.2017.
- [Anderl, 2000] ANDERL, REINER. 2000. *STEP STandard for the Exchange of Product Model Data - Eine Einführung in die Entwicklung, Implementierung und industrielle Nutzung der Normenreihe ISO 10303 (STEP)*. 2000. Aufl. edn. Wiesbaden: Vieweg+Teubner Verlag.
- [Borrmann et al., 2015] BORRMANN, ANDRÉ, KÖNIG, MARKUS, KOCH, CHRISTIAN, & BEETZ, JAKOB. 2015. *Building Information Modeling - Technologische Grundlagen und industrielle Praxis*. 1. Aufl. edn. Wiesbaden: Springer Fachmedien Wiesbaden.
- [Gronau et al., 2008-2016] GRONAU, NORBERT, BECKER, JÖRG, SINZ, ELMAR J., SUHL, LEENA, & LEIMEISTER, JAN MARCO. 2008-2016. *Enzyklopädie der Wirtschaftsinformatik - Online-Lexikon*. Lehrstuhl für Wirtschaftsinformatik (insb. Prozesse und Systeme), Universität Potsdam, Website. Online erhältlich unter <http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon>, zuletzt abgerufen am 14.02.2017.
- [Hausknecht & Liebich, 2016] HAUSKNECHT, KERSTIN, & LIEBICH, THOMAS. 2016. *BIM-Kompendium. - Building Information Modeling als neue Planungsmethode*. Stuttgart: Fraunhofer Irb Stuttgart.
- [Leemhuis, 2005] LEEMHUIS, HELEN. 2005. Funktionsgetriebene Konstruktion als Grundlage verbesserter Produktentwicklung.
- [Liebich et al., 2015-2017] LIEBICH, THOMAS, ADACHI, YOSHINOBU, FORESTER, JAMES, HYVARINEN, JUHA, RICHTER, STEFAN, CHIPMAN, TIM, WEISE, MATTHIAS, & WIX, JEFFREY. 2015-2017. *buildingSMART, International home of openBIM*. Website by buildingSMART International Limited. Online erhältlich unter <http://www.buildingsmart-tech.org/ifc/IFC4/final/html/>, zuletzt abgerufen am 14.11.2016.
- [Lukas et al., 2008] LUKAS, KATHARINA, JI, YANG, OBERGRIESSER, MATHIAS, BORRMANN, ANDRÉ, BAUINFORMATIK, FACHBEREICH BAUINGENIEURWESEN, & CAD, HOCHSCHULE REGENSBURG. 2008. Parametrische Modellierung von Brückenbauwerken. In: *Proc., 20. Forum Bauinformatik (Tagungsband des 20. Forum Bauinformatik)*. Dresden, Germany.

- [Ma *et al.*, 2006] MA, HOMAN, HA, KWAN MEI ELSA, CHUNG, CHUN KIT JACKIE, & AMOR, ROBERT. 2006. Testing semantic interoperability. *Pages 1216–1225 of: Proc. of the Joint International Conference on Computing and Decision Making in Civil and Building Engineering (ICCCBE)*.
- [Merziger *et al.*, 2010] MERZIGER, GERHARD, MÜHLBACH, GÜNTER, WILLE, DETLEF, & WIRTH, THOMAS. 2010. *Formeln + Hilfen höhere Mathematik* -. 6. aufl. edn. Barsinghausen: Binomi Verlag.
- [Monteiro & Martins, 2013] MONTEIRO, ANDRÉ, & MARTINS, JOÃO POÇAS. 2013. A survey on modeling guidelines for quantity takeoff-oriented BIM-based design. *Automation in Construction*, **35**, 238–253.
- [Mortenson, 1997] MORTENSON, MICHAEL E. 1997. *Geometric Modeling*. Second edition edn. Wiley.
- [Niedermaier & Bäck, 2015] NIEDERMAIER, ANKE, & BÄCK, ROBERT. 2015 (11). *Allplan; BIM-Kompendium; Theorie und Praxis*. 2 edn. Allplan GmbH, München.
- [Pazlar & Turk, 2008] PAZLAR, TOMAŽ, & TURK, ŽIGA. 2008. Interoperability in practice: geometric data exchange using the IFC standard. *Journal of Information Technology in Construction (ITcon)*, **13**(24), 362–380.
- [Sabelka, 2001] SABELKA, RAINER. 2001. *Dreidimensionale Finite Elemente Simulation von Verdrahtungsstrukturen auf Integrierten Schaltungen*. Website. Online erhältlich unter <http://www.iue.tuwien.ac.at/phd/sabelka/diss.html>, zuletzt abgerufen am 17.02.2017.
- [Scherer & Schapke, 2014] SCHERER, RAIMAR J., & SCHAPKE, SVEN-ERIC. 2014. *Informationssysteme im Bauwesen 1 - Modelle, Methoden und Prozesse*. 1. aufl. edn. Berlin Heidelberg New York: Springer-Verlag.
- [Schiele, 2012] SCHIELE, HANS-GÜNTER. 2012. *Computergrafik für Ingenieure - Eine anwendungsorientierte Einführung*. 2012. aufl. edn. Berlin Heidelberg New York: Springer-Verlag.

A Anhang

A.1 Geometrische Repräsentationen in IFC4

	Identifer	Typ	kurze Erläuterung	Elemente
Box Geometry	Box	Bounding-Box	vereinfachte Darstellung; umhüllender Quader als Platzhalter	<i>IfcBoundingBox</i>
Annotation 2D Geometry	Annotation	Annotation 2D	Darstellung von Punkten, Kurven, Schraffuren, Text; ohne topologischen Zusammenhang	<i>IfcPoint, IfcCurve, IfcAnnotationFillArea, IfcTextLiteral</i>
Annotation 3D Geometry	Annotation	Geometric Set	zusätzlich zur 2D-Version sind Oberflächen darstellbar	<i>IfcPoint, IfcCurve, IfcSurface</i>
Axis 2D Geometry	Axis	Curve2D	für Elemente die einer Bahn folgen: Darstellung als Achse, mit assoziiertem Profil kann ein Körper generiert werden; Profil wird an der Achse ausgerichtet; nutzbar für Standardwände	<i>IfcBoundedCurve</i>
Axis 3D Geometry	Axis	Curve3D	für Elemente, deren Profil entlang einer Achse extruiert wird, z.B.: standard Träger, Stützen, Rohre, Leitungen	<i>IfcBoundedCurve</i>
FootPrint Geometry	Foot Print	Geometric CurveSet/ Annotation2D	Darstellung der Grundfläche eines Körpers mit Begrenzungslinien; mit assoziiertem Profil kann ein Körper generiert werden	<i>IfcGeometricCurveSet /IfcPoint, IfcCurve, IfcAnnotationFillArea, IfcTextLiteral</i>
Profile 3D Geometry	Profile	Curve3D	Repräsentation für die Außenlinie von Objekten, welche eine Öffnung füllen, z.B: Türen oder Fenster; beschreibt die Form der Öffnung	<i>IfcBoundedCurve</i>
Surface 3D Geometry	Surface	Surface3D	Repräsentation durch Oberflächen, z.B. für thermische Grenzen	<i>IfcBoundedSurface, IfcSweptSurface</i>

Tabelle A.1: Übersicht über geometrische Repräsentationen des Standards IFC4, Teil 1 (vgl. [Liebich *et al.*, 2015-2017])

	Identifizier	Typ	kurze Erläuterung	Elemente
Body SurfaceOr SolidModel Geometry	Body	SurfaceOr SolidModel	gemischte Repräsentation aus SolidModel und Surface	<i>IfcTessellatedItem, IfcShellBasedSurfaceModel, IfcFaceBasedSurfaceModel, IfcSolidModel</i>
Body SurfaceModel Geometry	Body	Surface Model	beschreibt ein 3D-Objekt nur durch Oberflächen	<i>IfcTessellatedItem, IfcShellBasedSurfaceModel, IfcFaceBasedSurfaceModel</i>
Body Tessellation Geometry	Body	Tessellation	beschreibt die Form eines 3D-Objekts durch triangulierte Oberflächen	<i>IfcTessellatedItem</i>
Body SweptSolid Geometry	Body	SweptSolid	beschreibt die Form eines 3D-Objekts durch extrudieren oder rotieren eines 2D-Profiles	<i>IfcExtrudedAreaSolid, IfcRevolvedAreaSolid</i>
Body Advanced SweptSolid Geometry	Body	Advanced SweptSolid	erweiterte SweptSolid-Repräsentation; weitere Sweep-Operatoren möglich: z.B: geschwungene Pfade, Verschlingung des Profils	<i>IfcSweptAreaSolid, IfcSweptDiskSolid</i>
Body Brep Geometry	Body	Brep	beschreibt die Form eines 3D-Objekts durch ebene Begrenzungsflächen	<i>IfcFacetedBrep</i>
Body Advanced Brep Geometry	Body	Advanced Brep	beschreibt die Form eines 3D-Objekts durch ebene und gekrümmte Begrenzungsflächen	<i>IfcAdvancedBrep, IfcFacetedBrep</i>
Body CSG Geometry	Body	CSG	beschreibt die Form eines 3D-Objekts durch Primitive und deren Vereinigung, Schnitt und Differenz	<i>IfcCSGSolid</i>
Body Clipping Geometry	Body	Clipping	beschreibt die Form eines 3D-Objekts als Differenz eines SweptSolids mit einem Halbraum	<i>IfcBooleanClippingResult</i>

Tabelle A.2: Übersicht über geometrische Repräsentationen des Standards IFC4, Teil 2 (vgl. [Liebich *et al.*, 2015-2017])

A.2 STEP-Files

A.2.1 Wand

```
#132= IFCWALLSTANDARDCASE('3XjfkwiE9EDRoXpKvYw$_d',$,'',$,
    $,#64,#65,$,$);
#64= IFCLOCALPLACEMENT($,#61);
#61= IFCAXIS2PLACEMENT3D(#62,$,$);
#62= IFCCARTESIANPOINT((0.,0.,0.));
#65= IFCPRODUCTDEFINITIONSHAPE($,$,(#69,#105));
#69= IFCSHAPEREPRESENTATION(#19,'Body','SweptSolid',(#84));
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,
    *,*,*,#15,$,.MODEL_VIEW.,$)
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
    1.0000000000000000E-5,#10,$);
#10= IFCAXIS2PLACEMENT3D(#13,$,$);
#13= IFCCARTESIANPOINT((0.,0.,0.));
#84= IFCEXTRUDEDAREASOLID(#76,#85,#92,3000.);
#76= IFCRECTANGLEPROFILEDEF(.AREA.,$,#79,10000.,115.);
#79= IFCAXIS2PLACEMENT2D(#80,#82);
#80= IFCCARTESIANPOINT((5000.,57.5));
#82= IFCDIRECTION((1.,0.));
#85= IFCAXIS2PLACEMENT3D(#86,#88,#90);
#86= IFCCARTESIANPOINT((0.,0.,0.));
#88= IFCDIRECTION((0.,0.,1.));
#90= IFCDIRECTION((1.,0.,0.));
#92= IFCDIRECTION((0.,0.,1.));
#105= IFCSHAPEREPRESENTATION(#21,'Axis','Curve2D',(#108));
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis','Model',
    *,*,*,*,#15,$,.MODEL_VIEW.,$);
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
    1.0000000000000000E-5,#10,$);
#10= IFCAXIS2PLACEMENT3D(#13,$,$);
#13= IFCCARTESIANPOINT((0.,0.,0.));
#108= IFCPOLYLINE((#110,#112));
#110= IFCCARTESIANPOINT((0.,0.));
#112= IFCCARTESIANPOINT((10000.,0.));
```

Abbildung A.1: *Wand*-Export aus Allplan; Referenzlinie innen

```

#132= IFCWALLSTANDARDCASE( '3XjfkwiE9EDRoXpKvYw$_d' , $ , ' ' , $ ,
    $ , #64 , #65 , $ , $ );
#64= IFCLOCALPLACEMENT( $ , #61 );
#61= IFCAXIS2PLACEMENT3D( #62 , $ , $ );
#62= IFCCARTESIANPOINT( ( 0 . , 0 . , 0 . ) );
#65= IFCPRODUCTDEFINITIONSHAPE( $ , $ , ( #69 , #105 ) );
#69= IFCSHAPEREPRESENTATION( #19 , 'Body' , 'SweptSolid' , ( #84 ) );
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Body' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW . , $ )
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0 . , 0 . , 0 . ) );
#84= IFCEXTRUDEDAREASOLID( #76 , #85 , #92 , 3000 . );
#76= IFCRECTANGLEPROFILEDEF( .AREA . , $ , #79 , 10000 . , 115 . );
#79= IFCAXIS2PLACEMENT2D( #80 , #82 );
#80= IFCCARTESIANPOINT( ( 5000 . , 57.5 ) );
#82= IFCDIRECTION( ( 1 . , 0 . ) );
#85= IFCAXIS2PLACEMENT3D( #86 , #88 , #90 );
#86= IFCCARTESIANPOINT( ( 0 . , 0 . , 0 . ) );
#88= IFCDIRECTION( ( 0 . , 0 . , 1 . ) );
#90= IFCDIRECTION( ( 1 . , 0 . , 0 . ) );
#92= IFCDIRECTION( ( 0 . , 0 . , 1 . ) );
#105= IFCSHAPEREPRESENTATION( #21 , 'Axis' , 'Curve2D' , ( #108 ) );
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Axis' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW . , $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0 . , 0 . , 0 . ) );
#108= IFCPOLYLINE( ( #110 , #112 ) );
#110= IFCCARTESIANPOINT( ( 0 . , 115 . ) );
#112= IFCCARTESIANPOINT( ( 10000 . , 115 . ) );

```

Abbildung A.2: Wand-Export aus Allplan; Referenzlinie außen

```

#136= IFCWALLSTANDARDCASE( '2kbT9kqU59lBvW$E6vuvPr' , $ , ' ' ,
    $ , $ , #68 , #69 , $ , $ );
#68= IFCLOCALPLACEMENT( $ , #61 );
#61= IFCAXIS2PLACEMENT3D( #62 , #64 , #66 );
#62= IFCCARTESIANPOINT( ( 10000. , 115. , 0. ) );
#64= IFCDIRECTION( ( 0. , 0. , 1. ) );
#66= IFCDIRECTION( ( -1. , 0. , 0. ) );
#69= IFCPRODUCTDEFINITIONSHAPE( $ , $ , ( #73 , #109 ) );
#73= IFCSHAPEREPRESENTATION( #19 , 'Body' , 'SweptSolid' , ( #88 ) );
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Body' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW. , $ )
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#88= IFCEXTRUDEDAREASOLID( #80 , #89 , #96 , 3000. );
#80= IFCRECTANGLEPROFILEDEF( .AREA. , $ , #83 , 10000. , 115. );
#83= IFCAXIS2PLACEMENT2D( #84 , #86 );
#84= IFCCARTESIANPOINT( ( 5000. , 57.5 ) );
#86= IFCDIRECTION( ( 1. , 0. ) );
#89= IFCAXIS2PLACEMENT3D( #90 , #92 , #94 );
#90= IFCCARTESIANPOINT( ( 0. , -0. , 0. ) );
#92= IFCDIRECTION( ( 0. , 0. , 1. ) );
#94= IFCDIRECTION( ( 1. , 0. , 0. ) );
#96= IFCDIRECTION( ( 0. , 0. , 1. ) );
#109= IFCSHAPEREPRESENTATION( #21 , 'Axis' , 'Curve2D' , ( #112 ) );
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Axis' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW. , $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#112= IFCPOLYLINE( ( #114 , #116 ) );
#114= IFCCARTESIANPOINT( ( 0. , 57.5 ) );
#116= IFCCARTESIANPOINT( ( 10000. , 57.5 ) );

```

Abbildung A.3: *Wand*-Export aus Allplan, Referenzlinie mitte

```

#195= IFCWALLSTANDARDCASE('2zBRdou2XC5eXf_C5JAgU9',#43,
    'Basiswand:KS 11.5:388895',,$,'Basiswand:KS 11.5:3922',#156,
    #189,'388895',.NOTDEFINED.);
#43= IFCOWNERHISTORY(#40,#5,$,.NOCHANGE.,,$,$,0);
#156= IFCLOCALPLACEMENT(#138,#155);
#138= IFCLOCALPLACEMENT(#33,#137);
#33= IFCLOCALPLACEMENT(#268,#32);
#268= IFCLOCALPLACEMENT($,#267);
#267= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#32= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#137= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#155= IFCAXIS2PLACEMENT3D(#153,$,$);
#153= IFCCARTESIANPOINT((0.,0.0575,0.));
#189= IFCPRODUCTDEFINITIONSHAPE($,$,(#162,#186));
#162= IFCSHAPEREPRESENTATION(#100,'Axis','Curve2D',(#160));
#100= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis','Model',* ,* ,
    * ,* ,#96,$,.GRAPH_VIEW.,$);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
    1.000000000000000E-5,#93,#94);
#93= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#94= IFCDIRECTION((6.12303176911189E-17,1.));
#160= IFCPOLYLINE((#10,#158));
#10= IFCCARTESIANPOINT((0.,0.));
#158= IFCCARTESIANPOINT((10.,0.));
#186= IFCSHAPEREPRESENTATION(#102,'Body','SweptSolid',(#176));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',* ,* ,
    * ,* ,#96,$,.MODEL_VIEW.,$);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
    1.000000000000000E-5,#93,#94);
#93= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#94= IFCDIRECTION((6.12303176911189E-17,1.));
#176= IFCEXTRUDEDAREASOLID(#172,#175,#20,3.);
#172= IFCRECTANGLEPROFILEDEF(.AREA.,$,#171,10.,0.115);
#171= IFCAXIS2PLACEMENT2D(#169,#26);
#169= IFCCARTESIANPOINT((5.,-3.46944695195361E-18));
#26= IFCDIRECTION((-1.,0.));
#175= IFCAXIS2PLACEMENT3D(#6,$,$);
#20= IFCDIRECTION((0.,0.,1.));

```

Abbildung A.4: Wand-Export aus Revit, Referenzlinie mitte

```

#380= IFCWALLSTANDARDCASE( '17Dt7Z8ODA9f$HFzqVQ_LL' , $ , ' ' , $ ,
    $ , #64 , #65 , $ , $ );
#64= IFCLOCALPLACEMENT( $ , #61 );
#61= IFCAXIS2PLACEMENT3D( #62 , $ , $ );
#62= IFCCARTESIANPOINT( ( 5000. , 0. , 0. ) );
#65= IFCPRODUCTDEFINITIONSHAPE( $ , $ , ( #69 , #350 ) );
#69= IFCSHAPEREPRESENTATION( #19 , 'Body' , 'SweptSolid' , ( #329 ) );
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Body' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW. , $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#329= IFCEXTRUDEDAREASOLID( #76 , #330 , #337 , 3000. );
#76= IFCARBITRARYCLOSEDPROFILEDEF( .AREA. , $ , #79 );
#79= IFCPOLYLINE( ( 124 x IFCCARTESIANPOINT [ #81 - #327 ] ) );
#330= IFCAXIS2PLACEMENT3D( #331 , #333 , #335 );
#331= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#333= IFCDIRECTION( ( 0. , 0. , 1. ) );
#335= IFCDIRECTION( ( 1. , 0. , 0. ) );
#337= IFCDIRECTION( ( 0. , 0. , 1. ) );
#350= IFCSHAPEREPRESENTATION( #21 , 'Axis' , 'Curve2D' , ( #353 ) );
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Axis' , 'Model' , * ,
    * , * , #15 , $ , .MODEL_VIEW. , $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $ , 'Model' , 3 ,
    1.0000000000000000E-5 , #10 , $ );
#10= IFCAXIS2PLACEMENT3D( #13 , $ , $ );
#13= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#353= IFCTRIMMEDCURVE( #361 , ( IFCPARAMETERVALUE( 0. ) ) ,
    ( IFCPARAMETERVALUE( 180. ) ) , .T. , .PARAMETER. );
#361= IFCCIRCLE( #356 , 5000. );
#356= IFCAXIS2PLACEMENT2D( #357 , #359 );
#357= IFCCARTESIANPOINT( ( 0. , 0. ) );
#359= IFCDIRECTION( ( 1. , 0. ) );

```

Abbildung A.5: Halbkreiswand-Export aus Allplan, Referenzlinie innen

```

#232= IFCWALLSTANDARDCASE( '2v0KorP1DFR93Rlf73j6nc', #43,
    'Basiswand:KS 11.5:388785', $, 'Basiswand:KS 11.5:3922', #156,
    #226, '388785', .NOTDEFINED.);
#43= IFCOWNERHISTORY(#40, #5, $, .NOCHANGE., $, $, $, 0);
#156= IFCLOCALPLACEMENT(#138, #155);
#138= IFCLOCALPLACEMENT(#33, #137);
#33= IFCLOCALPLACEMENT(#305, #32);
#305= IFCLOCALPLACEMENT($, #304);
#304= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#32= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#137= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#155= IFCAXIS2PLACEMENT3D(#153, $, $);
#153= IFCCARTESIANPOINT((0.05749999999999999, 0., 0.));
#226= IFCPRODUCTDEFINITIONSHAPE($, $, (#165, #223));
#165= IFCSHAPEREPRESENTATION(#100, 'Axis', 'Curve2D', (#162));
#100= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis', 'Model', *, *,
    *, *, #96, $, .GRAPH_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#162= IFCTRIMMEDCURVE(#161, (IFCPARAMETERVALUE(360.)),
    (IFCPARAMETERVALUE(180.)), .T., .PARAMETER.);
#161= IFCCIRCLE(#160, 4.9425);
#160= IFCAXIS2PLACEMENT2D(#158, #24);
#158= IFCCARTESIANPOINT((4.9425, 0.));
#24= IFCDIRECTION((1., 0.));
. . .

```

Abbildung A.6: *Halbkreiswand*-Export aus Revit, Referenzlinie innen, Teil 1

```

. . .

#223= IFCSHAPEREPRESENTATION(#102,'Body','SweptSolid',(#213));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,
*,*,#96,$,.MODEL_VIEW.,$);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
1.000000000000000E-5,#93,#94);
#93= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#94= IFCDIRECTION((6.12303176911189E-17,1.));
#213= IFCEXTRUDEDAREASOLID(#209,#212,#20,3.);
#209= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#203);
#203= IFCCOMPOSITECURVE((#179,#187,#195,#202),.F.);
#179= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#176);
#176= IFCTRIMMEDCURVE(#175,(IFCPARAMETERVALUE(360.)),
(IFCPARAMETERVALUE(180.)),.T.,.PARAMETER.);
#175= IFCCIRCLE(#174,5.);
#174= IFCAXIS2PLACEMENT2D(#172,#24);
#172= IFCCARTESIANPOINT((4.9425,0.));
#24= IFCDIRECTION((1.,0.));
#187= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#185);
#185= IFCPOLYLINE((#181,#183));
#181= IFCCARTESIANPOINT((-0.0574999999999999,0.));
#183= IFCCARTESIANPOINT((0.0574999999999999,0.));
#195= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.F.,#192);
#192= IFCTRIMMEDCURVE(#191,(IFCPARAMETERVALUE(360.)),
(IFCPARAMETERVALUE(180.)),.T.,.PARAMETER.);
#191= IFCCIRCLE(#190,4.885);
#190= IFCAXIS2PLACEMENT2D(#188,#24);
#188= IFCCARTESIANPOINT((4.9425,0.));
#24= IFCDIRECTION((1.,0.));
#202= IFCCOMPOSITECURVESEGMENT(.CONTINUOUS.,.T.,#200);
#200= IFCPOLYLINE((#196,#198));
#196= IFCCARTESIANPOINT((9.8275,0.));
#198= IFCCARTESIANPOINT((9.9425,-0.));
#212= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#20= IFCDIRECTION((0.,0.,1.));

```

Abbildung A.7: Halbkreiswand-Export aus Revit, Referenzlinie innen, Teil 2

```

#142= IFCWALLSTANDARDCASE( '2JSPqWyKb7dB9308tSUzqu', $, ' ',
    $, $, #64, #65, $, $ );
#64= IFCLOCALPLACEMENT( $, #61 );
#61= IFCAXIS2PLACEMENT3D( #62, $, $ );
#62= IFCCARTESIANPOINT( ( 0., 0., 0. ) );
#65= IFCPRODUCTDEFINITIONSHAPE( $, $, ( #69, #115 ) );
#69= IFCSHAPEREPRESENTATION( #19, 'Body', 'Clipping', ( #103 ) );
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Body', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $, 'Model', 3,
    1.0000000000000000E-5, #10, $ );
#10= IFCAXIS2PLACEMENT3D( #13, $, $ );
#13= IFCCARTESIANPOINT( ( 0., 0., 0. ) );
#103= IFCBOOLEANCLIPPINGRESULT( .DIFFERENCE., #84, #102 );
#84= IFCEXTRUDEDAREASOLID( #76, #85, #92, 3000.000000000001 );
#76= IFCRECTANGLEPROFILEDEF( .AREA., $, #79, 10000., 115. );
#79= IFCAXIS2PLACEMENT2D( #80, #82 );
#80= IFCCARTESIANPOINT( ( 5000., 57.5 ) );
#82= IFCDIRECTION( ( 1., 0. ) );
#85= IFCAXIS2PLACEMENT3D( #86, #88, #90 );
#86= IFCCARTESIANPOINT( ( 0., 0., 0. ) );
#88= IFCDIRECTION( ( 0., 0., 1. ) );
#90= IFCDIRECTION( ( 1., 0., 0. ) );
#92= IFCDIRECTION( ( 0., 0., 1. ) );
#102= IFCHALFSPACESOLID( #94, .T. );
#94= IFCPLANE( #95 );
#95= IFCAXIS2PLACEMENT3D( #96, #98, #100 );
#96= IFCCARTESIANPOINT( ( 0., 0., 3000.000000000001 ) );
#98= IFCDIRECTION( ( -0.09950371902099894, 0.,
    -0.9950371902099892 ) );
#100= IFCDIRECTION( ( 0.9950371902099893, 0.,
    -0.09950371902099896 ) );
#115= IFCSHAPEREPRESENTATION( #21, 'Axis', 'Curve2D', ( #118 ) );
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Axis', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $, 'Model', 3,
    1.0000000000000000E-5, #10, $ );
#10= IFCAXIS2PLACEMENT3D( #13, $, $ );
#13= IFCCARTESIANPOINT( ( 0., 0., 0. ) );
#118= IFCPOLYLINE( ( #120, #122 ) );
#120= IFCCARTESIANPOINT( ( 0., 0. ) );
#122= IFCCARTESIANPOINT( ( 10000., 0. ) );

```

Abbildung A.8: *Abgeschrägte Wand*-Export aus Allplan, Referenzlinie innen

```

#203= IFCWALLSTANDARDCASE( '1 QD8ehWkL5LBQAuuxuY06d', #43,
    'Basiswand:KS 11.5:388684', $, 'Basiswand:KS 11.5:3922',
    #156, #197, '388684', .NOTDEFINED.);
#43= IFCOWNERHISTORY(#40, #5, $, .NOCHANGE, $, $, $, 1482850066);
#40= IFCPERSONANDORGANIZATION(#36, #39, $);
#36= IFCPERSON($, 'Lang', 'Simone', ('Maria'), $, $, $, $);
#39= IFCORGANIZATION($, '', '$, $);
#156= IFCLOCALPLACEMENT(#138, #155);
#138= IFCLOCALPLACEMENT(#33, #137);
#33= IFCLOCALPLACEMENT(#276, #32);
#276= IFCLOCALPLACEMENT($, #275);
#275= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#32= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#137= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#155= IFCAXIS2PLACEMENT3D(#153, $, $);
#153= IFCCARTESIANPOINT((0., 0.0575, 0.));
#197= IFCPRODUCTDEFINITIONSHAPE($, $, (#162, #194));
#162= IFCSHAPEREPRESENTATION(#100, 'Axis', 'Curve2D', (#160));
#100= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis', 'Model',
    *, *, *, *, #96, $, .GRAPH_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#160= IFCPOLYLINE((#10, #158));
#10= IFCCARTESIANPOINT((0., 0.));
#158= IFCCARTESIANPOINT((10., 0.));
#194= IFCSHAPEREPRESENTATION(#102, 'Body', 'Clipping', (#184));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model',
    *, *, *, *, #96, $, .MODEL_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#184= IFCBOOLEANCLIPPINGRESULT(.DIFFERENCE., #176, #183);
#176= IFCEXTRUDEDAREASOLID(#172, #175, #20, 3.);
#172= IFCRECTANGLEPROFILEDEF(.AREA., $, #171, 10., 0.115);
#171= IFCAXIS2PLACEMENT2D(#169, #26);
#169= IFCCARTESIANPOINT((5., 6.93889390390723E-18));
#26= IFCDIRECTION((-1., 0.));
#175= IFCAXIS2PLACEMENT3D(#6, $, $);
#20= IFCDIRECTION((0., 0., 1.));
. . .

```

Abbildung A.9: *Abgeschrägte Wand*-Export aus Revit, Referenzlinie innen, Teil 1

```

#183= IFCHALFSPACESOLID(#182,.F.);
#182= IFCPLANE(#181);
#181= IFCAXIS2PLACEMENT3D(#177,#179,#16);
#177= IFCCARTESIANPOINT((0.,-0.0575,3.));
#179= IFCDIRECTION((0.0995037190209989,0.,0.995037190209989));
#16= IFCDIRECTION((0.,1.,0.));

```

Abbildung A.10: *Abgeschrägte Wand-Export* aus Revit, Referenzlinie innen, Teil 2

```

#247= IFCRELVOIDSELEMENT('1Vdckg2NfF3xLEWO5ssUPy',,$,$,$,
#132,#220);
#132= IFCWALLSTANDARD_CASE('3F4hnKRUTB7Pshw6NGADkb',,$,' ',
,$,$,#64,#65,$,$);
#220= IFCOPENINGSTANDARD_CASE('2cQaGpW2z1HObdFpDirdxJ',,$,
',',,$,'Opening',#190,#192,$,$);
#190= IFCLOCALPLACEMENT(#64,#183);
#64= IFCLOCALPLACEMENT($,#61);
#61= IFCAXIS2PLACEMENT3D(#62,$,$);
#62= IFCCARTESIANPOINT((0.,0.,0.));
#183= IFCAXIS2PLACEMENT3D(#184,#186,#188);
#184= IFCCARTESIANPOINT((3000.,0.,1000.));
#186= IFCDIRECTION((0.,1.,0.));
#188= IFCDIRECTION((0.,0.,1.));
#192= IFCPRODUCTDEFINITIONSHAPE($,$,(#194));
#194= IFCSHAPEREPRESENTATION(#19,'Body','SweptSolid',(#209));
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,
*,*,*,#15,$,MODEL_VIEW.,$);
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
1.0000000000000000E-5,#10,$);
#10= IFCAXIS2PLACEMENT3D(#13,$,$);
#13= IFCCARTESIANPOINT((0.,0.,0.));
#209= IFCEXTRUDEDAREASOLID(#196,#210,#217,115.);
#196= IFCARBITRARYCLOSEDPROFILEDEF(.AREA.,$,#197);
#197= IFCPOLYLINE((#199,#201,#203,#205,#199));
#199= IFCCARTESIANPOINT((0.,4000.));
#201= IFCCARTESIANPOINT((1000.,4000.));
#203= IFCCARTESIANPOINT((1000.,0.));
#205= IFCCARTESIANPOINT((0.,0.));
#210= IFCAXIS2PLACEMENT3D(#211,#213,#215);
#211= IFCCARTESIANPOINT((0.,0.,0.));
#213= IFCDIRECTION((0.,0.,1.));
#215= IFCDIRECTION((1.,0.,0.));
#217= IFCDIRECTION((0.,0.,1.));

```

Abbildung A.11: *Öffnung in Wand-Export* aus Allplan, Referenzlinie innen

```

#246= IFCRELVOIDSELEMENT( '1Y9s$PK0jAEeQi50ZHFZHj', #43, $, $,
    #195, #241);
#43= IFCOWNERHISTORY(#40, #5, $, .NOCHANGE., $, $, $, 1480955771);
#195= IFCWALLSTANDARDCASE( '2tDv47frDFre_QFc6Z9rAA', #43,
    'Basiswand:KS 11.5:388762', $, 'Basiswand:KS 11.5:3922', #156,
    #189, '388762', .NOTDEFINED.);
#241= IFCOPENINGELEMENT( '2tDv47frDFre_QFc6Z9rBu', #43,
    'Basiswand:KS 11.5:388762', $, 'Opening', #240, #234, $, .OPENING.);
#43= IFCOWNERHISTORY(#40, #5, $, .NOCHANGE., $, $, $, 1480955771);
#240= IFCLOCALPLACEMENT(#156, #239);
#156= IFCLOCALPLACEMENT(#138, #155);
#138= IFCLOCALPLACEMENT(#33, #137);
#33= IFCLOCALPLACEMENT(#307, #32);
#307= IFCLOCALPLACEMENT($, #306);
#306= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#32= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#137= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#155= IFCAXIS2PLACEMENT3D(#153, $, $);
#153= IFCCARTESIANPOINT((0., 0.0575, 0.));
#239= IFCAXIS2PLACEMENT3D(#237, $, $);
#237= IFCCARTESIANPOINT((3., -0.057500000000000129, 1.));
#234= IFCPRODUCTDEFINITIONSHAPE($, $, (#230));
#230= IFCSHAPEREPRESENTATION(#102, 'Body', 'SweptSolid', (#220));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model', *,
    *, *, *, #96, $, .MODEL_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.0000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#220= IFCEXTRUDEDAREASOLID(#216, #219, #20, 0.115);
#216= IFCRECTANGLEPROFILEDEF(.AREA., $, #215, 1., 4.);
#215= IFCAXIS2PLACEMENT2D(#213, #24);
#213= IFCCARTESIANPOINT((5.55111512312578E-17,
    4.44089209850063E-16));
#24= IFCDIRECTION((1., 0.));
#219= IFCAXIS2PLACEMENT3D(#217, #16, #22);
#217= IFCCARTESIANPOINT((2., 0., 0.5));
#16= IFCDIRECTION((0., 1., 0.));
#22= IFCDIRECTION((0., 0., -1.));
#20= IFCDIRECTION((0., 0., 1.));

```

Abbildung A.12: *Öffnung in Wand-Export aus Revit, Referenzlinie innen*

A.2.2 Geschossdecke

```
#122= IFCSLABSTANDARDCASE( '0 eE72dTefBSPYSzzDuZSHh ', $, ' ', $,
    $, #64, #65, $, .FLOOR.);
#64= IFCLOCALPLACEMENT($, #61);
#61= IFCAXIS2PLACEMENT3D(#62, $, $);
#62= IFCCARTESIANPOINT((0., 0., 0.));
#65= IFCPRODUCTDEFINITIONSHAPE($, $, (#69));
#69= IFCSHAPEREPRESENTATION(#19, 'Body', 'SweptSolid', (#84));
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $);
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.0000000000000000E-5, #10, $);
#10= IFCAXIS2PLACEMENT3D(#13, $, $);
#13= IFCCARTESIANPOINT((0., 0., 0.));
#84= IFCEXTRUDEDAREASOLID(#76, #85, #92, 150.);
#76= IFCRECTANGLEPROFILEDEF(.AREA., $, #79, 10000., 7000.);
#79= IFCAXIS2PLACEMENT2D(#80, #82);
#80= IFCCARTESIANPOINT((5000., 3500.));
#82= IFCDIRECTION((1., 0.));
#85= IFCAXIS2PLACEMENT3D(#86, #88, #90);
#86= IFCCARTESIANPOINT((0., 0., 0.));
#88= IFCDIRECTION((0., 0., 1.));
#90= IFCDIRECTION((1., 0., 0.));
#92= IFCDIRECTION((0., 0., 1.));
```

Abbildung A.13: *Geschossdecke*-Export aus Allplan

```

#189= IFCSLAB('105aUTPRX43BgXzPIXzH7r',#43,
'Geschossdecke:STB 15.0:389019',$, 'Geschossdecke:STB 15.0',
#157,#184,'389019',.FLOOR.);
#43= IFCOWNERHISTORY(#40,#5,$,.NOCHANGE.,$,$,$,1480689090);
#40= IFCPERSONANDORGANIZATION(#36,#39,$);
#36= IFCPERSON($,'Lang','Simone',('Maria'),$,$,$,$);
#39= IFCORGANIZATION($,'','',$,$);
#5= IFCAPPLICATION(#1,'2017','Autodesk Revit 2017 (DEU)',
'Revit');
#157= IFCLOCALPLACEMENT(#138,#156);
#138= IFCLOCALPLACEMENT(#33,#137);
#33= IFCLOCALPLACEMENT(#261,#32);
#261= IFCLOCALPLACEMENT($,#260);
#260= IFCAXIS2PLACEMENT3D(#6,$,$);
#32= IFCAXIS2PLACEMENT3D(#6,$,$);
#137= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#156= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCCARTESIANPOINT((0.,0.,0.));
#184= IFCPRODUCTDEFINITIONSHAPE($,$,(#177));
#177= IFCSHAPEREPRESENTATION(#102,'Body','SweptSolid',(#167));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,
*,*,*,#96,$,.MODEL_VIEW.,$);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
1.000000000000000E-5,#93,#94);
#93= IFCAXIS2PLACEMENT3D(#6,$,$);
#94= IFCDIRECTION((6.12303176911189E-17,1.));
#167= IFCEXTRUDEDAREASOLID(#161,#166,#20,0.15);
#161= IFCRECTANGLEPROFILEDEF(.AREA.,'STB 15.0',#160,7.,10.);
#160= IFCAXIS2PLACEMENT2D(#158,#24);
#158= IFCCARTESIANPOINT((0.,0.));
#24= IFCDIRECTION((1.,0.));
#166= IFCAXIS2PLACEMENT3D(#164,#22,#16);
#164= IFCCARTESIANPOINT((5.,3.5,0.15));
#22= IFCDIRECTION((0.,0.,-1.));
#16= IFCDIRECTION((0.,1.,0.));
#20= IFCDIRECTION((0.,0.,1.));

```

Abbildung A.14: *Geschossdecke*-Export aus Revit

A.2.3 Stütze

```
#138= IFCCOLUMNSTANDARD_CASE('31RLC5mbf7ouyoLSKH2H95', $,
    ', $, $, #64, #65, $, $);
#64= IFCLOCALPLACEMENT($, #61);
#61= IFCAXIS2PLACEMENT3D(#62, $, $);
#62= IFCCARTESIANPOINT((0., 0., 0.));
#65= IFCPRODUCTDEFINITIONSHAPE($, $, (#69, #105));
#69= IFCSHAPEREPRESENTATION(#19, 'Body', 'SweptSolid', (#84));
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $);
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.0000000000000000E-5, #10, $);
#10= IFCAXIS2PLACEMENT3D(#13, $, $);
#13= IFCCARTESIANPOINT((0., 0., 0.));
#84= IFCEXTRUDEDAREASOLID(#76, #85, #92, 3000.);
#76= IFCRECTANGLEPROFILEDEF(.AREA., $, #79, 250., 250.);
#79= IFCAXIS2PLACEMENT2D(#80, #82);
#80= IFCCARTESIANPOINT((125., 125.));
#82= IFCDIRECTION((1., 0.));
#85= IFCAXIS2PLACEMENT3D(#86, #88, #90);
#86= IFCCARTESIANPOINT((0., 0., 0.));
#88= IFCDIRECTION((0., 0., 1.));
#90= IFCDIRECTION((1., 0., 0.));
#92= IFCDIRECTION((0., 0., 1.));
#105= IFCSHAPEREPRESENTATION(#21, 'Axis', 'Curve3D', (#108));
#21= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Axis', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $);
#15= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.0000000000000000E-5, #10, $);
#10= IFCAXIS2PLACEMENT3D(#13, $, $);
#13= IFCCARTESIANPOINT((0., 0., 0.));
#108= IFCPOLYLINE((#110, #112));
#110= IFCCARTESIANPOINT((0., 0., 0.));
#112= IFCCARTESIANPOINT((0., 0., 3000.));
```

Abbildung A.15: *Stütze*-Export aus Allplan

```

#224= IFCCOLUMN( '0DVyerIbHDlu47vMWsL5VT', #43,
    'STB St\X2\00FC\X0\tze - rechteckig:STB 25 x 25:389496',
    $, 'STB 25 x 25', #222, #214, '389496', .COLUMN.);
#43= IFCOWNERHISTORY(#40, #5, $, .NOCHANGE., $, $, $, 0);
#40= IFCPERSONANDORGANIZATION(#36, #39, $);
#36= IFCPERSON($, 'Lang', 'Simone', ('Maria'), $, $, $, $);
#39= IFCORGANIZATION($, '', '', $, $);
#5= IFCAPPLICATION(#1, '2017', 'Autodesk Revit 2017 (DEU)',
    'Revit');
#222= IFCLOCALPLACEMENT(#138, #221);
#138= IFCLOCALPLACEMENT(#33, #137);
#33= IFCLOCALPLACEMENT(#264, #32);
#264= IFCLOCALPLACEMENT($, #263);
#263= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#32= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#137= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#221= IFCAXIS2PLACEMENT3D(#219, $, $);
#219= IFCCARTESIANPOINT((0.1250000000000004, 0.125, 0.));
#214= IFCPRODUCTDEFINITIONSHAPE($, $, (#212));
#212= IFCSHAPEREPRESENTATION(#102, 'Body',
    'MappedRepresentation', (#210));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model', *,
    *, *, *, #96, $, .MODEL_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#210= IFCMAPPEDITEM(#180, #209);
#180= IFCREPRESENTATIONMAP(#179, #172);
#179= IFCAXIS2PLACEMENT3D(#6, $, $);
#6= IFCCARTESIANPOINT((0., 0., 0.));
#172= IFCSHAPEREPRESENTATION(#102, 'Body', 'SweptSolid', (#162));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body', 'Model', *,
    *, *, *, #96, $, .MODEL_VIEW., $);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($, 'Model', 3,
    1.000000000000000E-5, #93, #94);
#93= IFCAXIS2PLACEMENT3D(#6, $, $);
#94= IFCDIRECTION((6.12303176911189E-17, 1.));
#162= IFCEXTRUDEDAREASOLID(#158, #161, #20, 3.);
#158= IFCRECTANGLEPROFILEDEF(.AREA., 'STB 25 x 25', #157,
    0.25, 0.25);
#157= IFCAXIS2PLACEMENT2D(#155, #24);
#155= IFCCARTESIANPOINT((0., 0.));
. . .

```

Abbildung A.16: *Stütze*-Export aus Revit, Teil 1

. . . .

```
#24= IFCDIRECTION (( 1. , 0. ));
#161= IFCAxis2Placement3D (#6, #20, #18);
#6= IFCCARTESIANPOINT (( 0. , 0. , 0. ));
#20= IFCDIRECTION (( 0. , 0. , 1. ));
#18= IFCDIRECTION (( 0. , - 1. , 0. ));
#20= IFCDIRECTION (( 0. , 0. , 1. ));
#209= IFCCARTESIANTRANSFORMATIONOPERATOR3D ($, $, #6, 1., $);
#6= IFCCARTESIANPOINT (( 0. , 0. , 0. ));
```

Abbildung A.17: *Stütze*-Export aus Revit, Teil 2

A.2.4 Körpermodell

```
#158= IFCBUILDINGELEMENTPROXY( '0SkciiM8f2HuPT24w0jgo5' ,
    $, ' ', $, $, #64, #65, $, $ );
#64= IFCLOCALPLACEMENT( $, #61 );
#61= IFCAXIS2PLACEMENT3D( #62, $, $ );
#62= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#65= IFCPRODUCTDEFINITIONSHAPE( 'Explicit Shape', $, (#69));
#69= IFCSHAPEREPRESENTATION( #19, 'Body', 'Brep', (#136));
#19= IFCGEOMETRICREPRESENTATIONSUBCONTEXT( 'Body', 'Model',
    *, *, *, *, #15, $, .MODEL_VIEW., $ );
#15= IFCGEOMETRICREPRESENTATIONCONTEXT( $, 'Model', 3,
    1.0000000000000000E-5, #10, $ );
#10= IFCAXIS2PLACEMENT3D( #13, $, $ );
#13= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#136= IFCFACETEDBREP( #134 );
#134= IFCCLOSEDSHELL( (#85, #97, #108, #119, #132));
#85= IFCFACE( (#84) );
#84= IFCFACEOUTERBOUND( #82, .F. );
#82= IFCPOLYLOOP( (#76, #78, #80) );
#76= IFCCARTESIANPOINT( ( 3500. , 3500. , 7000. ) );
#78= IFCCARTESIANPOINT( ( 7000. , 0. , 0. ) );
#80= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#97= IFCFACE( (#96) );
#96= IFCFACEOUTERBOUND( #94, .F. );
#94= IFCPOLYLOOP( (#88, #90, #92) );
#88= IFCCARTESIANPOINT( ( 3500. , 3500. , 7000. ) );
#90= IFCCARTESIANPOINT( ( 7000. , 7000. , 0. ) );
#92= IFCCARTESIANPOINT( ( 7000. , 0. , 0. ) );
#108= IFCFACE( (#107) );
#107= IFCFACEOUTERBOUND( #105, .F. );
#105= IFCPOLYLOOP( (#99, #101, #103) );
#99= IFCCARTESIANPOINT( ( 3500. , 3500. , 7000. ) );
#101= IFCCARTESIANPOINT( ( 0. , 7000. , 0. ) );
#103= IFCCARTESIANPOINT( ( 7000. , 7000. , 0. ) );
#119= IFCFACE( (#118) );
#118= IFCFACEOUTERBOUND( #116, .F. );
#116= IFCPOLYLOOP( (#110, #112, #114) );
#110= IFCCARTESIANPOINT( ( 3500. , 3500. , 7000. ) );
#112= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#114= IFCCARTESIANPOINT( ( 0. , 7000. , 0. ) );
#132= IFCFACE( (#131) );
#131= IFCFACEOUTERBOUND( #129, .F. );
#129= IFCPOLYLOOP( (#121, #123, #125, #127) );
#121= IFCCARTESIANPOINT( ( 0. , 0. , 0. ) );
#123= IFCCARTESIANPOINT( ( 7000. , 0. , 0. ) );
#125= IFCCARTESIANPOINT( ( 7000. , 7000. , 0. ) );
#127= IFCCARTESIANPOINT( ( 0. , 7000. , 0. ) );
```

Abbildung A.18: Körpermodell-Export aus Allplan

```

#293= IFCBUILDINGELEMENTPROXYTYPE('1yJJNtk_XC1R1nycODJZrE',#43,
'Pyramide',,$,$,$,(#289),'389972',$,.NOTDEFINED.);
#43= IFCOWNERHISTORY(#40,#5,$,.NOCHANGE.,,$,$,1483368429);
...
#289= IFCREPRESENTATIONMAP(#288,#281);
#288= IFCAXIS2PLACEMENT3D(#6,$,$);
#6= IFCARTESIANPOINT((0.,0.,0.));
#281= IFCSHAPEREPRESENTATION(#102,'Body','AdvancedBrep',(#280));
#102= IFCGEOMETRICREPRESENTATIONSUBCONTEXT('Body','Model',*,*,*,
*,#96,$,.MODEL_VIEW.,$);
#96= IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,
1.000000000000000E-5,#93,#94);
#93= IFCAXIS2PLACEMENT3D(#6,$,$);
#94= IFCDIRECTION((6.12303176911189E-17,1.));
#280= IFCADVANCEDBREP(#278);
#278= IFCCLOSEDSHELL((#214,#228,#244,#260,#276));
#214= IFCADVANCEDFACE((#205),#213,.T.);
#205= IFCFACEOUTERBOUND(#203,.T.);
#203= IFCEDGELOOP((#200,#201,#202));
#200= IFCORIENTEDEDGE(*,*,#163,.T.);
#163= IFCEDGECURVE(#157,#160,#162,.T.);
#157= IFCVERTEXPOINT(#155);
#155= IFCARTESIANPOINT((7.,0.,0.));
#160= IFCVERTEXPOINT(#158);
#158= IFCARTESIANPOINT((7.,7.,0.));
#162= IFCLINE(#155,#161);
#155= IFCARTESIANPOINT((7.,0.,0.));
#161= IFCVECTOR(#16,0.3048);
#16= IFCDIRECTION((0.,1.,0.));
#201= IFCORIENTEDEDGE(*,*,#171,.T.);
#171= IFCEDGECURVE(#160,#166,#170,.T.);
#160= IFCVERTEXPOINT(#158);
#158= IFCARTESIANPOINT((7.,7.,0.));
#166= IFCVERTEXPOINT(#164);
#164= IFCARTESIANPOINT((3.5,3.499999999999999,7.));
#170= IFCLINE(#158,#169);
#158= IFCARTESIANPOINT((7.,7.,0.));
#169= IFCVECTOR(#167,0.3048);
#167= IFCDIRECTION((-0.408248290463845,-0.408248290463845,
0.816496580927744));
#202= IFCORIENTEDEDGE(*,*,#176,.T.);
#176= IFCEDGECURVE(#166,#157,#175,.T.);
#166= IFCVERTEXPOINT(#164);
#164= IFCARTESIANPOINT((3.5,3.499999999999999,7.));
#157= IFCVERTEXPOINT(#155);
#155= IFCARTESIANPOINT((7.,0.,0.));
#175= IFCLINE(#164,#174);
...

```

Abbildung A.19: Körpermodell-Export aus Revit, Teil 1

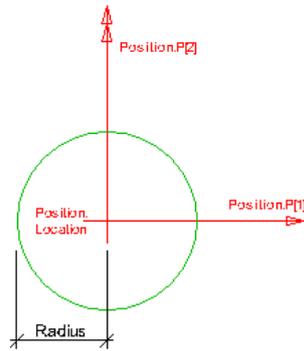
```

. . .
#164= IFCCARTESIANPOINT((3.5,3.499999999999999,7.));
#174= IFCVECTOR(#172,0.3048);
#172= IFCDIRECTION((0.408248290463845,-0.408248290463841,
-0.816496580927746));
#213= IFCPLANE(#212);
#212= IFCAXIS2PLACEMENT3D(#206,#208,#210);
#206= IFCCARTESIANPOINT((5.25,0.,3.4999999999999994));
#208= IFCDIRECTION((0.894427190999928,-0.,0.447213595499934));
#210= IFCDIRECTION((0.447213595499934,0.,-0.894427190999928));
#228= IFCADVANCEDFACE((#223),#227,.T.);
. . .
#244= IFCADVANCEDFACE((#235),#243,.T.);
. . .
#260= IFCADVANCEDFACE((#251),#259,.T.);
. . .
#276= IFCADVANCEDFACE((#267),#275,.T.);
. . .

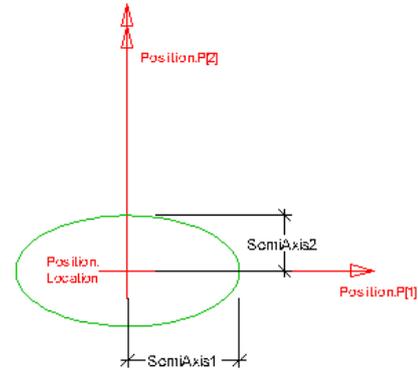
```

Abbildung A.20: Körpermodell-Export aus Revit, Teil 2

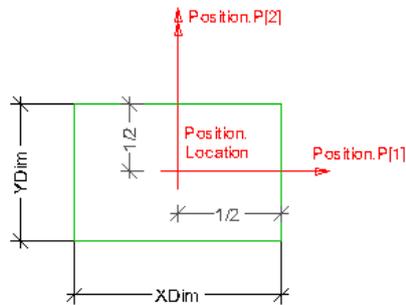
A.3 Parametrische Profile



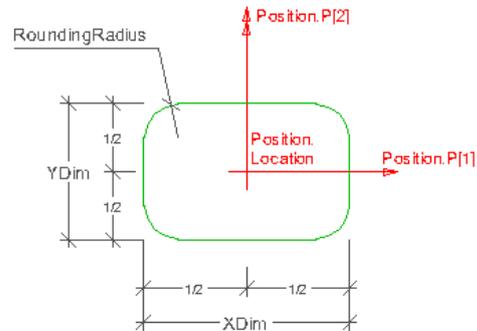
(a) IfcCircleProfileDef
[Liebich *et al.*, 2015-2017]



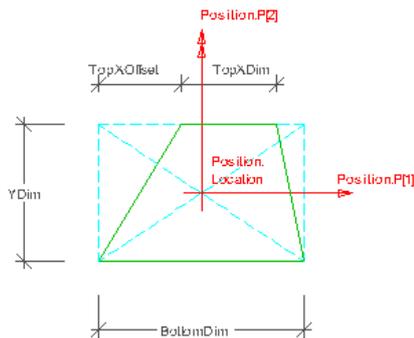
(Quelle: (b) IfcEllipseProfileDef
[Liebich *et al.*, 2015-2017] (Quelle:



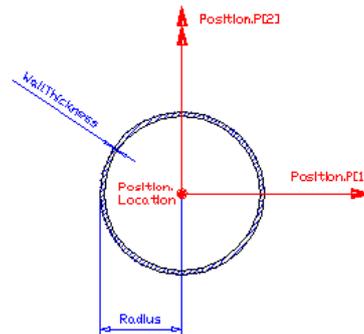
(c) IfcRectangleProfileDef
[Liebich *et al.*, 2015-2017]



(Quelle: (d) IfcRoundedRectangleProfileDef
[Liebich *et al.*, 2015-2017] (Quelle:

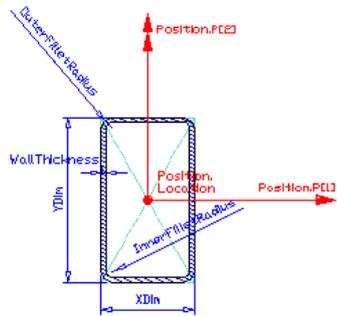


(e) IfcTrapeziumProfileDef
[Liebich *et al.*, 2015-2017]

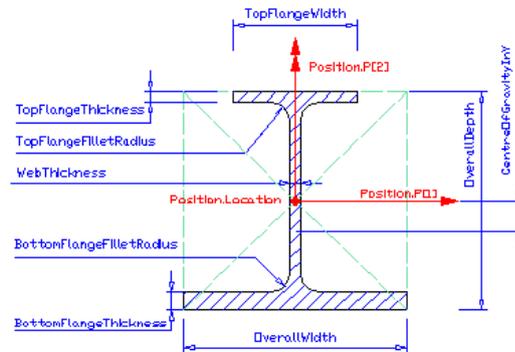


(Quelle: (f) IfcCircleHollowProfileDef
[Liebich *et al.*, 2015-2017] (Quelle:

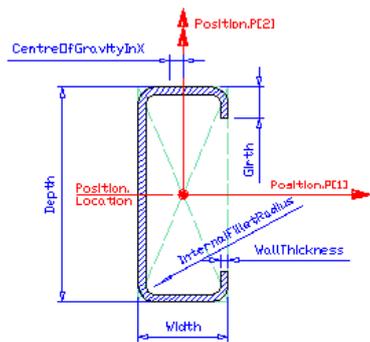
Abbildung A.21: Parametrische Profile des Standards IFC4, Teil 1



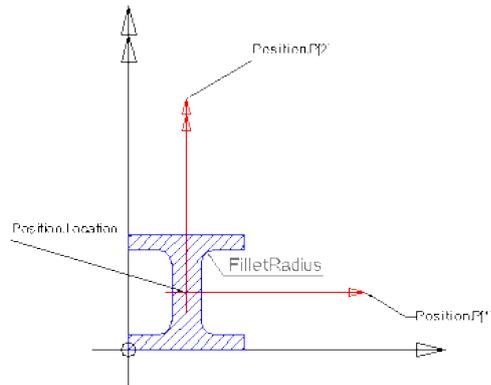
(a) IfcRectangleHollowProfileDef [Liebich *et al.*, 2015-2017]



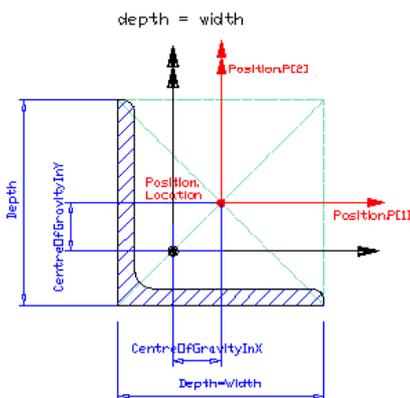
(Quelle: (b) IfcAsymmetricIShapeProfileDef [Liebich *et al.*, 2015-2017])



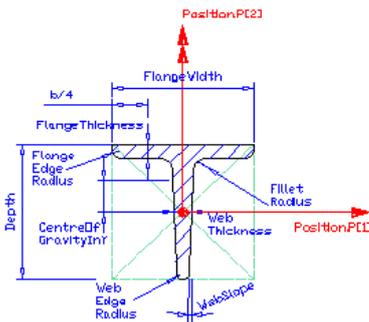
(c) IfcCShapeProfileDef [Liebich *et al.*, 2015-2017]



(Quelle: (d) IfcIShapeProfileDef [Liebich *et al.*, 2015-2017])

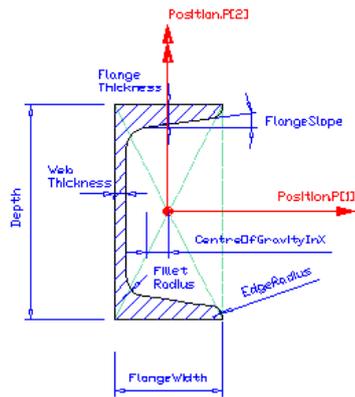


(e) IfcLShapeProfileDef [Liebich *et al.*, 2015-2017]

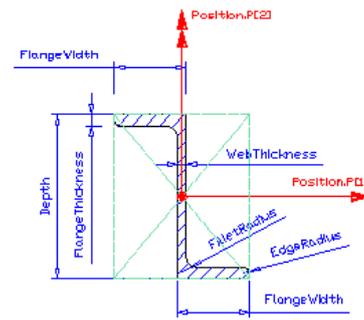


(Quelle: (f) IfcTShapeProfileDef [Liebich *et al.*, 2015-2017])

Abbildung A.22: Parametrische Profile des Standards IFC4, Teil 2



(a) IfcUShapeProfileDef
[Liebich *et al.*, 2015-2017]



(Quelle:(b) IfcZShapeProfileDef
[Liebich *et al.*, 2015-2017] (Quelle:

Abbildung A.23: Parametrische Profile des Standards IFC4, Teil 3

A.4 Flussdiagramme

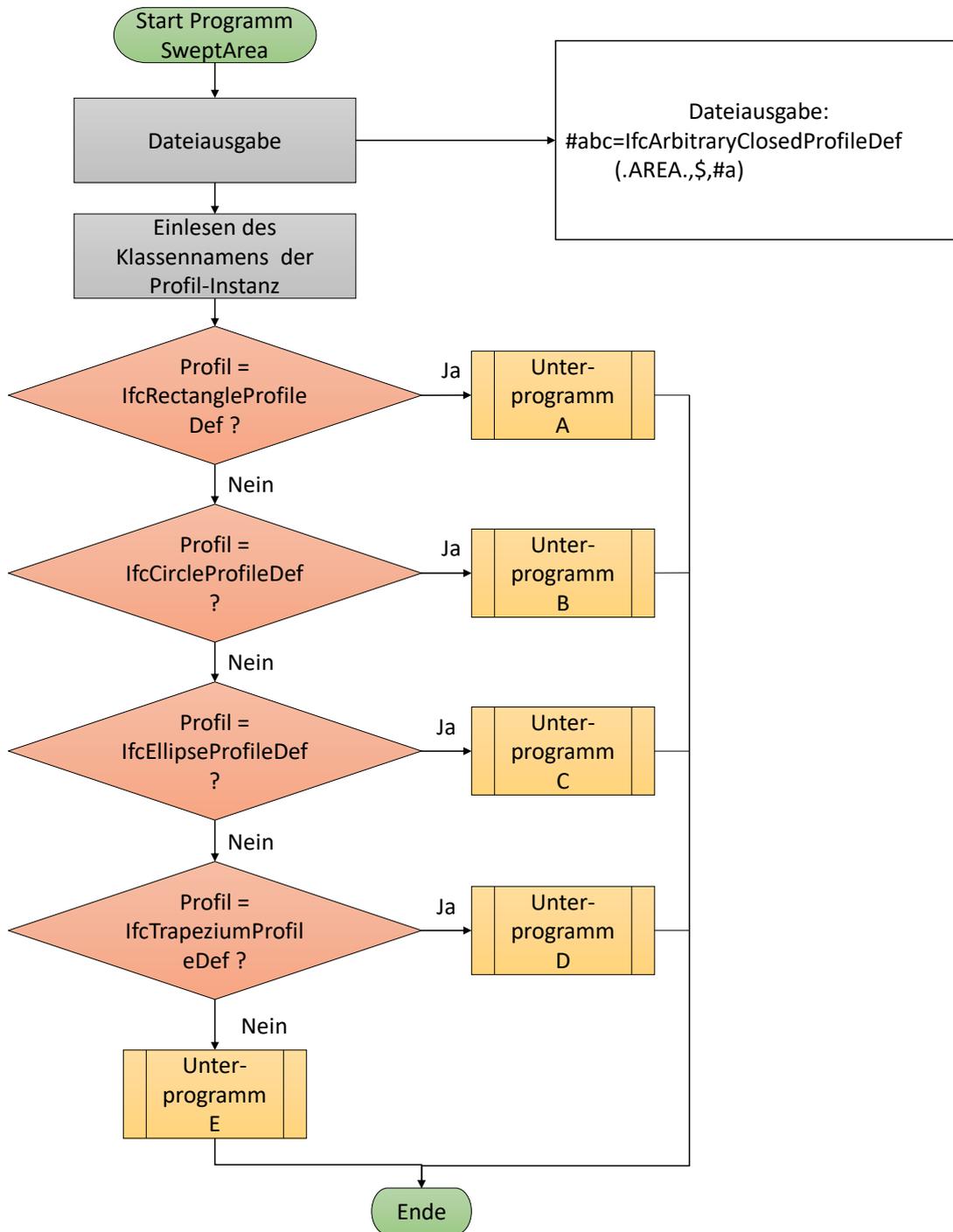


Abbildung A.24: Flussdiagramm des Hauptprogramms `SweptArea`

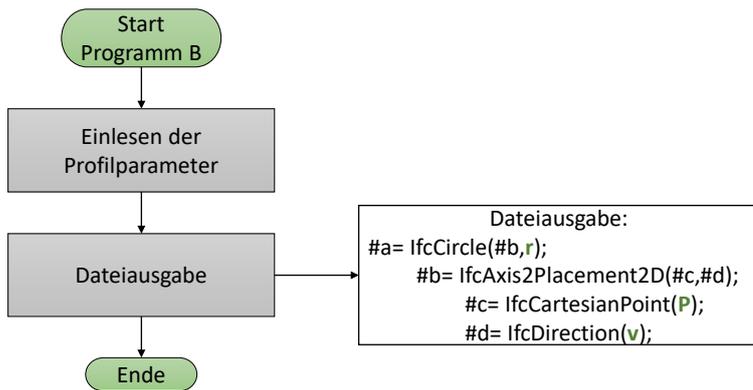


Abbildung A.25: Flussdiagramm des Unterprogramms B zur Umformung einer *IfcCircleProfileDef* inn Kreis

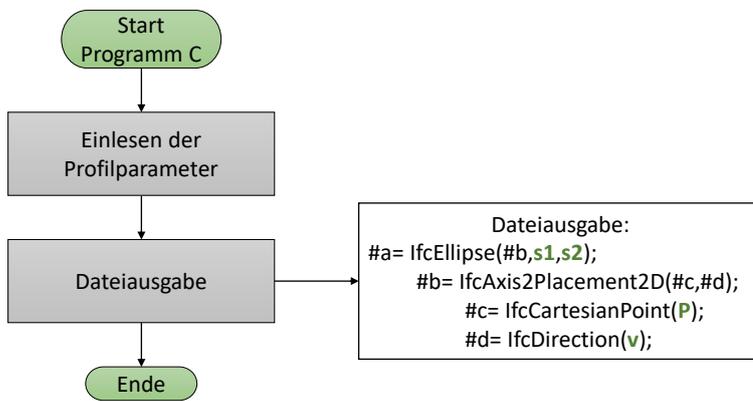


Abbildung A.26: Flussdiagramm des Unterprogramms C zur Umformung einer *IfcEllipseProfileDef* in eine Ellipse

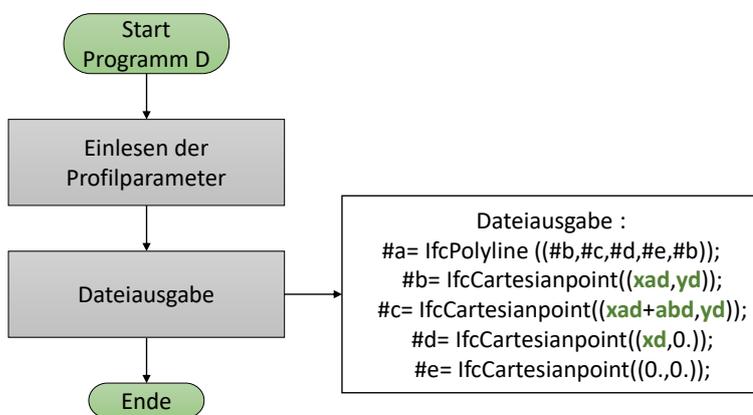


Abbildung A.27: Flussdiagramm des Unterprogramms D zur Umformung einer *IfcTrapeziumProfileDef* in eine Polylinie

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 20. März 2017

Simone Maria Lang