



Lehrstuhl für Computergestützte Modellierung und Simulation

Technische Universität München

Entwicklung eines IFC-basierenden Datenaustauschstandards für den Unterbau von Brückenbauwerken

Valery Kuloyants

Masterthesis

im Studiengang Bauingenieurwesen

Prüfer: Prof. Dr.-Ing. André Borrmann

Betreuer: M. Sc. Julian Amann

eingereicht: 31.01.14

Inhalt

INHALT	2
EINLEITUNG	7
ABSTRACT	8
ABKÜRZUNGSVERZEICHNISS	9
1 BUILDING INFORMATION MODELLING UND IFC	10
1.1 BUILDINGSMART	12
1.2 IFC-SCHEMA (EXPRESS-DATEI)	12
1.2.1 <i>Entitäten und Typen</i>	13
1.2.2 <i>Aufzählung und Auswahl</i>	14
1.2.3 <i>Funktionen und Regeln</i>	15
1.3 IFC-MODEL (STEP-DATEI)	16
2 IFC-BRIDGE	18
2.1 ENTWICKLUNG VON IFCBRIDGE UND AKTUELLER STAND	18
2.2 BRÜCKE MIT MASSIVPLATTENQUERSCHNITT IN IFC	18
2.3 DARSTELLUNG VON ÜBERBAUPLATTEN MIT VERÄNDERLICHEN QUERSCHNITTEN IN IFCBRIDGE	20
3 UNTERBAUTEN EINER BRÜCKE	24
3.1 ELEMENTE DES BRÜCKENUNTERBAUS	24
3.2 WIDERLAGER	24
3.3 PFEILER UND STÜTZEN	28
3.4 GRÜNDUNGEN	31
4 BESCHREIBUNG DES BRÜCKENUNTERBAUS IN IFC	33
4.1 STRUKTUR DES UNTERBAUS	33
4.2 GEOMETRISCHE DARSTELLUNG DES UNTERBAUS	36
4.2.1 <i>Überblick über geometrische Darstellungsmethoden</i>	36
4.2.2 <i>Darstellung von Widerlagern</i>	44
4.2.3 <i>Abspeicherung der Pfeiler und Stützen</i>	49
4.2.4 <i>Abspeicherung der Gründung</i>	53
5 IFC-IMPORT NACH SIEMENS NX	56
5.1 ALLGEMEINES ZU SIEMENS NX	56
5.2 ERWEITERUNG DES PROGRAMMS IFCBRIDGE TOOL	56
6 PARAMETRISCHES MODELLIEREN	62

6.1	EINFÜHRUNG UND MOTIVATION	62
6.2	VERFAHREN ZUR SPEICHERUNG DER PARAMETRIK IN IFC.....	63
6.2.1	<i>Referenzenmodell</i>	63
6.2.2	<i>Skizzenparametrikmodell</i>	69
6.2.3	<i>Beschreibung der Überbauplatte mit veränderlicher Querschnittshöhe</i>	74
6.3	MODELLIERUNG EINES PARAMETRISCHEN BRÜCKENUNTERBAUS IN IFC	77
7	ZUSAMMENFASSUNG UND FAZIT	84
	LITERATURVERZEICHNIS	86
	VERWENDETE SOFTWAREANWENDUNGEN.....	89
	ANHANG.....	90
	ERKLÄRUNG	91

Inhalt

Abbildung 1: BIM-Model speichert 2D- und 3D-Geometrie und zusätzliche Bauwerks-Information [2]	10
Abbildung 2: Arbeiten am gemeinsamen Modell mit Hilfe von BIM [4].....	11
Abbildung 3: BIM fördert kosteneffizientes Bauen [5]	12
Abbildung 4:Ausschnitt aus dem IFC-Diagramm	13
Abbildung 5: Entität IfcDoor (Tür) beschrieben mit EXPRESS-Schema [8].....	14
Abbildung 6: Beispiele für ENUMERATION (Aufzählung) und (SELECT) Auswahl [9]	15
Abbildung 7: Funktionen und Regeln innerhalb der Entität IfcCartesianPoint [8].....	15
Abbildung 8: Definition des Headers einer STEP-Datei.....	16
Abbildung 9: Polygonzug (IfcPolyline) in der IFC-Datei (STEP-Format)	17
Abbildung 10: Massivplatte [11]	18
Abbildung 11: Aufbau einer Brücke in IFC. IfcBridge-Elemente (violett), IFC-Standard (weiß)	19
Abbildung 12: Gewutete Balkenbrücke, Donaubrücke Pöchlarn [12].....	20
Abbildung 13: Darstellung des Überbaus mit Hilfe von IfcReferencedSectionedSpine	21
Abbildung 14: IfcReferencedSectionedSpine dient zur Darstellung des Überbaus [13].....	21
Abbildung 15: Konstruktionsprinzip des Hohlkastenquerschnitts mit drei Zellen in IFC [11].....	22
Abbildung 16: Kastenförmiges Widerlager - Prinzipdarstellung [14].....	24
Abbildung 17: Anschluss Überbau an Widerlager mit Kammerwand, Auflagerbank und Lagerung [17]	25
Abbildung 18: Anordnung von Widerlagern im Böschungsbereich [14]	26
Abbildung 19: Einfache Winkelstützwand [12] [15].....	26
Abbildung 20: Regelform eines kastenförmigen Widerlagers [17].....	27
Abbildung 21: Alternative Widerlagerform bei hohen Talbrücken [17]	27
Abbildung 22: Ein kastenförmiger Pfeiler im Bauzustand [19].....	28
Abbildung 23: System der Pfeilerscheibe (links) [14] und eine Autobahnunterführung [20].....	29
Abbildung 24: Treskowbrücke über die Spree in Berlin [21]	29
Abbildung 25: Unterstützung des Überbaus mit Einzelsützen. System [14] und Autobahnbrücke [20]	30
Abbildung 26: Rahmenstütze mit Zugband [14]	30
Abbildung 27: Hammerkopfausbildung [15]	31
Abbildung 28: Pfahlgründung mit schrägen Pfählen [9].....	32
Abbildung 29: Erweiterung des Strukturtyps um den Eintrag SUBSTRUCTURE	33
Abbildung 30: Definition des Brückenunterbaus in der IFC-Datei	34
Abbildung 31: Hierarchie IfcElement	34
Abbildung 32: Definition der Art des Unterbauelementes mittels IfcBridgeSubstructureElementType.....	34
Abbildung 33: Definition des Objektes IfcBridgeSubstructureElement in der IFC-STEP-Datei.....	35
Abbildung 34: Beispiel einer Aggregation der Unterbauelemente unter einem IfcBridgePart-Objekt.....	35
Abbildung 35: Prinzip des Randdarstellung-Modells (Boundary Representation) [22].....	37
Abbildung 36: Beschreibung einer rechteckigen Wand mit IfcFacetedBrep.....	38
Abbildung 37: Erzeugung eines Zylinders infolge der Extrusion eines 2D-Kreises	39
Abbildung 38: Aufbau eines IFC-Raumobjektes, das mit Extrusion beschrieben wird.....	40
Abbildung 39: Boolesche Operation [19]	41

Inhalt

Abbildung 40: Anschauliches Beispiel für die Konstruktion von CSG- und Extrusions-Körpern.....	42
Abbildung 41: IfcBoundingBox [23].....	43
Abbildung 42: Regelform (links) und Variantenform (rechts) eines kastenförmigen Widerlagers	44
Abbildung 43: Konstruktion des Basiskörpers mit Extrusion [23]	45
Abbildung 44: Basiskörper des Widerlagers in IFC [23]	46
Abbildung 45: Konstruktion der Flügelwände durch Abziehen eines Hilfskörpers [23]	47
Abbildung 46: Extrusion der massive Widerlagerstirnwand [23]	47
Abbildung 47: Flügelwände des Widerlagers und die Vereinigung aller Bauteile [23]	48
Abbildung 48: Pfeiler und Stützen mit konstanten Querschnitten [23].....	49
Abbildung 49: Darstellung eines Pfeilers mit veränderlichem Querschnitt.....	50
Abbildung 50: variable Extrusion (links) und der fertige Pfeiler (rechts)	51
Abbildung 51: Extrusion eines Rechtecksquerschnitts entlang eines Kreisbogen mit fester Orientierung (Siemens NX) [23].....	52
Abbildung 52: Konstruktion des Bogenquerschnitts mit IfcReferencedSectionedSpine [23].....	53
Abbildung 53: Fundamentmodellierung für ein Pfeiler (links) und ein kastenförmiges Widerlager (rechts) [23]	54
Abbildung 54: Extrusionsrichtungen bestimmen den Neigungswinkel der Bohrpfähle [23]	55
Abbildung 55: IFC-Hohlkasten-Brücke importier ins Siemens NX [13].....	56
Abbildung 56: Benutzeroberfläche des Programms IfcBridgeTool	57
Abbildung 57: ins Siemens NX importierte Plattenbrücke mit Unterbauelemente [26] [24]	58
Abbildung 58: Imports eines IFC-Modells mit der Umsetzung der Parametrik innerhalb der Schnittstelle ...	59
Abbildung 59: Hergang des Imports eines IFC-Modells mit der Integration der Parametrik im IFC-Modell	60
Abbildung 60: IfcParametricBinding EXPRESS-Schema [24]	64
Abbildung 61: Abhängigkeit der Extrusionstiefe der Widerlagerwand von anderen Parametern [24].....	65
Abbildung 62: Hierarchischer Weg zur z-Koordinate der Fundamentposition [24].....	66
Abbildung 63: Aufbau des Objektes IfcParametricConstant [24]	67
Abbildung 64: gängigsten Operatoren sortiert nach Kategorie.....	68
Abbildung 65: Berechnung der Widerlagerwand mit IfcParametricFormula	69
Abbildung 66: Vorschlag der Änderung der Klasse IfcParametricGeometryElement [24].....	71
Abbildung 67: topologische Zwangsbedingungen [27].....	72
Abbildung 68: dimensionale Zwangsbedingungen [27]	73
Abbildung 69: IfcParametricProfileDef im EXPRESS-Schema [29].....	74
Abbildung 70: Parametrische Außenkontur eines Querschnitts der Hohlkastenbrücke und ihre Beschreibung in IFC [30] [24].....	75
Abbildung 71: variable Höhe der Querschnitte [30].....	77
Abbildung 72: parametrische Brücke, dargestellt mit im Programm "Parametric IFC Editor" [31] [24]	79
Abbildung 73: Änderung der Brückenlänge [31] [24].....	80
Abbildung 74: Kreuzungsmöglichkeiten der über- und unterführten Straße [16]	80
Abbildung 75: Änderung der Achse der unterführten Straße [31] [24].....	81

Inhalt

<i>Abbildung 76: Änderung der Böschungsneigung [31] [24].....</i>	<i>82</i>
<i>Abbildung 77: Änderung der Fundamentversatzes [31] [24].....</i>	<i>83</i>
<i>Abbildung 78: Änderung der Fundamentoberkante [31] [24].....</i>	<i>83</i>

Einleitung

Seit dem Jahr 2007 werden 94% aller Daten digital abgespeichert [1]. Auch die Bauindustrie ist keine Ausnahme. Wenn früher die Bauzeichnung als wichtigster Informationsträger galt, wird heute nach digitalen, dreidimensionalen Bauwerksmodellen gestrebt, die nicht nur geometrische und statische Informationen, sondern auch den zeitlichen Ablauf der Ausführung, Massenermittlung, Kostenkalkulation und weitere Angaben beinhalten.

3D-Modellierungssoftwares werden von vielen namhaften CAD-Herstellern angeboten. Strategie, Umsetzung und Bezeichnungen unterscheiden sich jedoch von Hersteller zu Hersteller. Die internationale Organisation „buildingSMART“ entwickelte ein einheitliches, neutrales Basisdatenmodell – die Industry Foundation Classes (IFC) für den modellbasierten Datenaustausch im Bauwesen. Dadurch soll die Informationsweitergabe qualitäts-, termin- und kostensicherer gestaltet werden. Das semantische Modell wurde hauptsächlich für den Hochbau erarbeitet und enthält deswegen Objekte, die für dieses Teilgebiet des Bauwesens typisch sind. IfcBridge ist ein Projekt mit dem Ziel, spezielle Elemente eines Brückenbauwerkes in IFC zu integrieren, um die Vorteile, wie z. B. Datenneutralität von IFC, auch im Brückenbau nutzen zu können. Die Entwicklung im Rahmen dieses Projektes beschränkte sich bisher auf die Beschreibung der Überbauelemente einer Brücke wie z.B. der Fahrbahnplatte. Der Brückenunterbau mit seinen zahlreichen speziellen Bauelementen blieb bisher außer Achtung.

Im Rahmen dieser Masterarbeit wird ein Vorschlag erarbeitet, wie man Widerlager, Pfeiler und Fundamente als semantische Objekte im IFC-Format abspeichern kann. Dabei wird großer Wert auf die parametrischen Abhängigkeiten innerhalb eines Brückenmodells gelegt und anschließend ein Vorschlag erarbeitet, wie die Parametrik ins IFC-Modell integriert werden kann. Schlussendlich wird ein Plug-In entwickelt, das eine IFC-Brücke inklusive der Unterbauelemente in eine parametrische Modellierungssoftware übertragen kann.

Abstract

Since 2007, 94% of all data is stored in a digital form [1]. The construction industry is no exception. Whilst in the past the architectural drawing were the main information holder, now there will be strived for digital, three-dimensional building models, which include not only geometric and static informations, but also the timetable for the construction work, computation of the material quantities, cost calculations and further information.

3D modeling software is offered by many well-known CAD suppliers. However, the strategy, implementation and designation vary from manufacturer to manufacturer. The international organization „buildingSMART” developed a unified, neutral data model – the **Industry Foundation Classes (IFC)** for the model-based data exchange in civil engineering. This is intended to ensure a high degree of quality, reliable on-time delivery and cost certainty in the information transfer. The semantic model has been mainly developed for the building construction and consequently contains objects that are typical of this branch of the civil engineering. IfcBridge is a project with the aim to integrate specific elements of bridge constructions into IFC in order to benefit from its advantages such as data neutrality of IFC. The developments within this project so far were limited to the description of the bridge superstructures such as the bridge deck. The bridge substructure with its many special components remained until now unnoticed.

In the context of this master thesis, a proposal is being developed, how abutments, piers and foundations can be saved as semantic objects in the IFC format. A great emphasis is attached on the parametric dependencies within a bridge model and subsequently a proposal was developed how the parameterization can be integrated into the IFC-model. Finally, a plug-in is developed, which can import an IfcBridge-model including the substructure elements into a parametric modeling software.

Abkürzungsverzeichnis

2D	zweidimensional
3D	dreidimensional
ABS	abstract
API	Application Programming Interface
BIM	Building Information Modelling
BRep	Boundary Representation
bspw.	Beispielsweise
ca.	circa
CAD	Computer-Aided Design
CSG	Constructive Solid Geometry
d.h.	das heißt
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
m	meter
OPT	OPTIONAL
SDAI	Standard Data Access Interface
STEP	Standard for the Exchange of Product model data
usw.	und so weiter
z.B.	zum Beispiel

1 Building Information Modelling und IFC

Brückenbauzeichnungen werden heute typischerweise mit CAD-Programmen erstellt. Die klassische Planung eines Ingenieurbauwerks, z.B. einer Brücke, beginnt mit einem ersten CAD-Entwurf des Architekten. Um die Kosten für die Errichtung schätzen zu können, wird eine Massenermittlung auf Basis des Architekten-Entwurfs erstellt. Brandschutzgutachter, Energie- und Umwelttechniker, Behörden und weitere Fachingenieure benötigen die Baupläne zur Ermittlung der Kosten aus den jeweiligen Gewerken. Aufgrund der Komplexität eines Bauwerks sind Änderungen der Planung unumgänglich. Tritt z.B. eine Geometrie-Änderung auf, müssen die CAD-Zeichnungen geändert und die Massenermittlungen angepasst werden. Folglich müssen auch alle Beteiligten ihre Fachplanung mit den neuen Zeichnungen abgleichen und gegebenenfalls anpassen. Dass dieser Prozess einen hohen Koordinierungs-, Arbeits- und Kostenaufwand erfordert, belegen zahlreiche Studien [2].

Building Information Modeling (BIM) beschreibt ein 3D-Modellierungs-Verfahren, das diesem Problem entgegenwirkt. In einem Building Information Model wird zusätzlich zur Geometrie eine Vielzahl anderer relevanter Bauwerksdaten digital im 3D-Modell gespeichert.

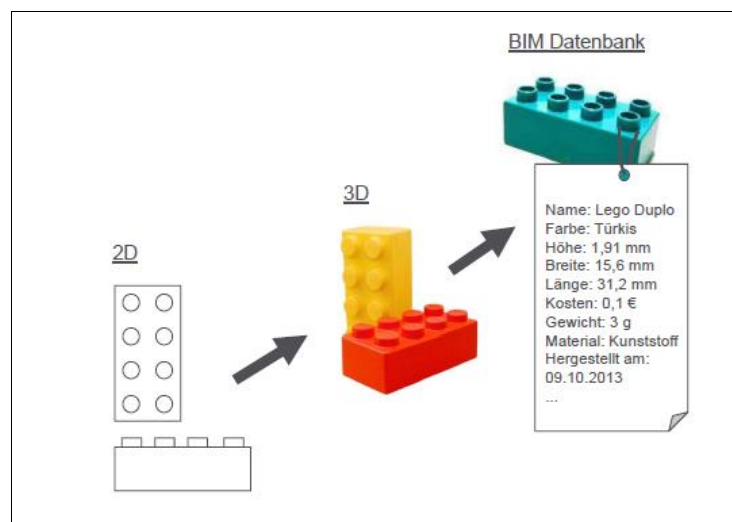


Abbildung 1: BIM-Model speichert 2D- und 3D-Geometrie und zusätzliche Bauwerks-Information [2]

Die hohe Komplexität der Bauwerksprodukte wird von mehreren, auf bestimmte Fachgebiete spezialisierten Teams erarbeitet [3]. Das BIM-Modell bietet die Möglichkeit für die beteiligten Architekten und Fachplaner, an einer gemeinsamen Datenbasis zu arbeiten. Potentielle Fehler, die häufig infolge eines unsorgfältigen Datenabgleichs entstehen, werden somit weitgehend vermieden.

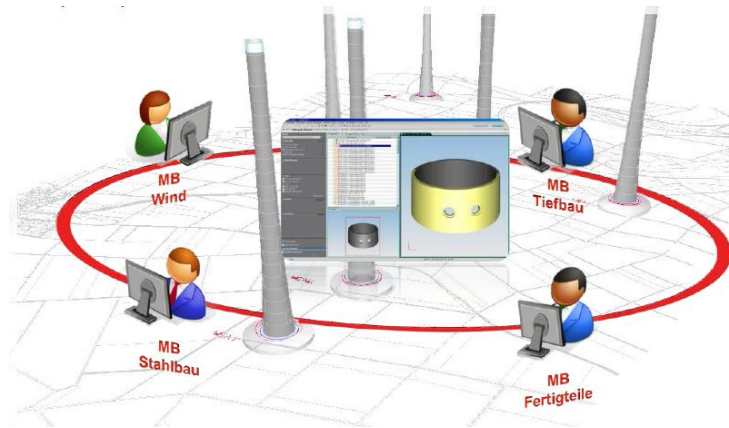


Abbildung 2: Arbeiten am gemeinsamen Modell mit Hilfe von BIM [4]

Idealerweise läuft der Prozess der Planungsänderung unter Anwendung von BIM folgendermaßen ab. Der Architekt nimmt eine Geometrie-Änderung vor, die Massenermittlung wird aufgrund der Verknüpfung mit der Bauwerks-Geometrie automatisch angepasst. Die Kostenermittlung wird aufgrund der Verknüpfung mit der Massenermittlung angepasst. Somit bedeuten das Arbeiten an einem einzigen Modell und die ständige Synchronisation dieses Modells hohe Qualität der Daten, verbesserten Informationsaustausch zwischen Planungsbeteiligten und kontinuierliche Verfügbarkeit der relevanten Daten während des gesamten Lebenszyklus eines Gebäudes [5]. Somit optimiert BIM die drei wichtigsten Säulen des Bauplanungsprozesses: Kosten, Termine und Qualität. In der Praxis ist man jedoch von so einem Arbeitsverlauf noch weit entfernt, da es beispielsweise für die Brückenbauwerke noch kein standardisiertes Austauschformat gibt.

Weitere Vorteile, die der Einsatz von BIM erkennen lässt, ist eine frühere Entscheidungsfindung im Projektverlauf mit kostengünstigeren Änderungen, Vermeidung von Mehrfacharbeit, höhere Planungssicherheit, einheitliche Projektdokumentation und zukünftig keine Abhängigkeit von bestimmten Software-Produkten [5]. Dies ermöglicht die Errichtung nachhaltiger und kosteneffizienter Bauwerke (Abbildung 3).

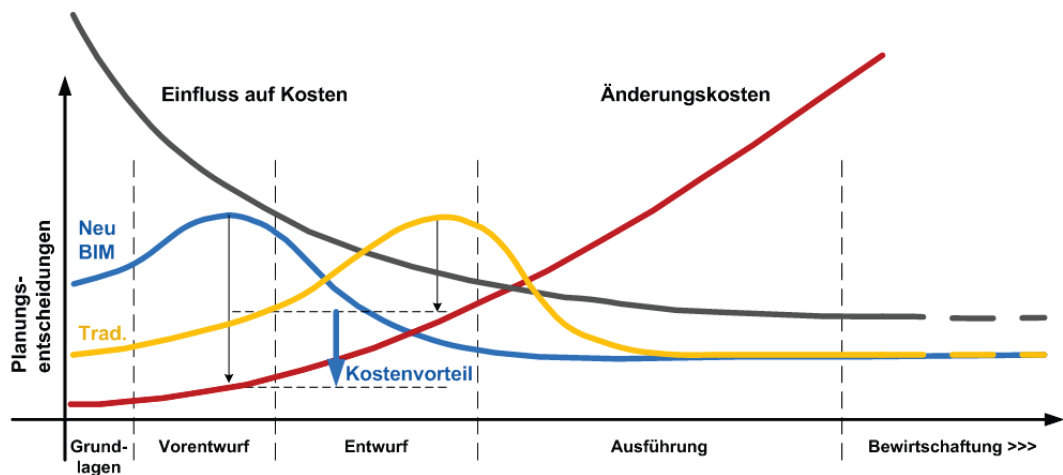


Abbildung 3: BIM fördert kosteneffizientes Bauen [5]

1.1 buildingSMART

„buildingSMART International“ ist eine internationale Organisation, die sich auf die zukunftsorientierte Modellierungsmethode BIM konzentriert. Es ist ein Verband aus Hochschulen, Firmen, aber auch Privatpersonen aus den Bereichen des Bauwesens, die daran interessiert sind, BIM weiterzuentwickeln und immer effizientere Methoden der Planung und Projektabwicklung zu erschaffen. Jährlich werden Veranstaltungen von buildingSMART organisiert, um neue Ergebnisse zu präsentieren und praktische Erfahrungen auszutauschen. Als Grundlage der Entwicklungen von buildingSMART gilt das Datenformat IFC (Industry Foundation Classes), das zum Austausch von digitalen Bauwerksdaten in der Baubranche dient. Bei IFC handelt es sich um einen offenen und somit für jeden zugänglichen Standard für die Datennutzung innerhalb der Wertschöpfungskette Bau [5]. Dieses Format sorgt dafür, dass ein und dasselbe Bauwerksmodell in verschiedenen Softwareanwendungen geladen und entsprechend verändert werden kann. Jedoch müssen diese Anwendungen IFC-kompatibel sein. Folgendes Kapitel gibt einen Überblick über die wichtigsten Fakten des IFC-Standards.

1.2 IFC-Schema (EXPRESS-Datei)

EXPRESS ist eine Sprache zur Beschreibung von produktspezifischen Informationen, die von der ISO-Norm 10303 definiert ist. In der EXPRESS-Datei werden Datentypen, Entitäten, Regeln, Funktionen und andere Objekte definiert [6]. IFC-Schema, das in EXPRESS-Sprache geschrieben ist, ermöglicht eine neutrale Beschreibung eines Bauwerksmodells und seine Umsetzung in unterschiedlichen Softwareanwendungen.

Mittels spezieller, hierarchisch aufgebauter Objekte lassen sich komplexe Bauwerke vollständig und detailliert beschreiben. Die Objekte werden in folgenden Gruppen kategorisiert:

- Entitäten (ENTITY)
- Typen (TYPE)
- Aufzählungen (ENUMERATION)
- Auswahllisten (SELECT)
- Funktionen (FUNCTION)
- Regeln (RULE)

Diese Objekte des IFC-Standards werden ständig aktualisiert und erweitert. Die aktuelle Version des IFC-Schemas - IFC4 - erschien im März 2013. Nachfolgend werden am Beispiel einer Brücke die einzelnen Objekte des IFC-Schemas erläutert.

1.2.1 Entitäten und Typen

Die Entitäten sind hierarchisch strukturiert (siehe Abbildung 4). Das EXPRESS-Schema unterstützt eine Mehrfachvererbung zwischen den definierten Objekten [6]. Die Entitäten `IfcPoint` (Punkt) und `IfcCurve` (Kurve) sind Unterklassen der abstrakten Entität `IfcGeometricRepresentationItem` (geometrierepräsentierendes Element) und erben somit ihre Eigenschaften. Der Punkt besitzt mehrere ihm untergeordnete Entitäten wie z.B.: `IfcCartesianPoint` (Definition über kartesische Koordinaten), `IfcPointOnCurve` (Punkt auf der Kurve) usw. In gleicher Weise gliedert sich die Kurve unter anderem in `IfcBoundedCurve` (begrenzte Kurve) und `IfcOffsetCurve` (Offsetkurve) (Abbildung 4).

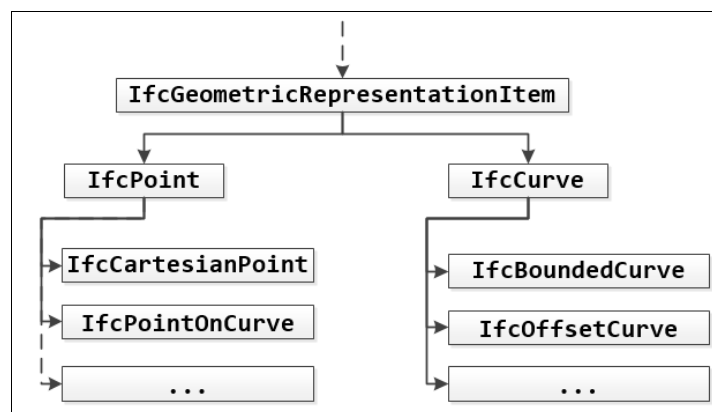


Abbildung 4: Ausschnitt aus dem IFC-Diagramm

Eigenschaften einer Entität (ENTITY) werden mit Attributen definiert. Abbildung 5 zeigt bspw. das EXPRESS-Gerüst für die Tür, die mit zwei Attributen – Höhe (OverallHeight) und Breite (OverallWidth) in IFC beschrieben wird. Die beiden Attribute sind jeweils vom Typ IfcPositiveLengthMeasure, das eine numerische Zahl mit positiver Längeneinheit definiert. Die Entität IfcPositiveLengthMeasure besitzt wiederum eigene Attribute, die ihre Eigenschaften beschreiben.

```
ENTITY IfcDoor
  SUBTYPE OF (IfcBuildingElement);
    OverallHeight : OPTIONAL IfcPositiveLengthMeasure;
    OverallWidth  : OPTIONAL IfcPositiveLengthMeasure;
END_ENTITY;
```

Abbildung 5: Entität IfcDoor (Tür) beschrieben mit EXPRESS-Schema [8]

Das Schlüsselwort OPTIONAL bedeutet, dass die Definition des betroffenen Attributs innerhalb des produktspezifischen IFC-STEP-Modells (Kapitel 1.3) optional, also nicht verpflichtend, ist.

1.2.2 Aufzählung und Auswahl

ENUMERATION (Aufzählung) und SELECT (Auswahlliste) sind Datentypen mit einer endlichen Wertemenge. Die Aufzählungen beinhalten Werte mit konstanten Namen. Die Auswahl dagegen bietet eine Selektion aus weiteren Typen oder Entitäten.

In der Abbildung 6 (links) ist der Eintrag im EXPRESS für den Aufzählungstyp IfcBridgeStructureType (Tragwerkstyp der Brücke) dargestellt. Die englischen Bezeichnungen für die Tragwerkstypen Hohlkasten, Bogenbrücke, Fachwerkbrücke usw. sind nacheinander aufgelistet. Die rechte Seite der Abbildung 6 zeigt den Auswahl-Datentyp IfcSimpleValue (einfacher Wert), das weitere elementare Datentypen beinhaltet, wie z.B. Wahrheitswerte, Datum, Ganzzahl, Gleitkommazahl usw.

<pre> TYPE IfcBridgeStructureType = ENUMERATION OF (BOX_GIRDER_BRIDGE, ARCHED_BRIDGE, SUSPENSION_BRIDGE, CABLE_STAYED_BRIDGE, GIRDER_BRIDGE, SLAB_BRIDGE, SLAB_BRIDGE_WITH_BROAD_CANTILEVER, BOW_STRING_BRIDGE, LADDER_BRIDGE, FRAMEWORK_BRIDGE, GISCLARD_BRIDGE, PORTAL_BRIDGE); END_TYPE; </pre>	<pre> TYPE IfcSimpleValue = SELECT (IfcBoolean, IfcDate, IfcDateTime, IfcDuration, IfcIdentifier, IfcInteger, IfcLabel, IfcLogical, IfcReal, IfcText, IfcTime, IfcTimeStamp); END_TYPE; </pre>
---	---

Abbildung 6: Beispiele für ENUMERATION (Aufzählung) und (SELECT) Auswahl [9]

1.2.3 Funktionen und Regeln

Am Beispiel der Entität des kartesischen Punktes (IfcCartesianPoint) aus der Abbildung 7 lässt sich die Aufgabe von Funktionen (FUNCTION) und Regeln (RULE) erklären. Der kartesische Punkt besitzt als erstes Attribut seine kartesischen Koordinaten (Coordinates). Diese werden in einer Liste (LIST) angegeben. Das zweite Attribut DIM (Dimension des Punktes) wird über eine Funktion HIINDEX hergeleitet. Die Dimension des Punktes ist gleich der Anzahl der angegeben Koordinaten (2D oder 3D). Die Regel WR1 (WHERE) definiert die Mindestanzahl der kartesischen Koordinaten auf 2.

```

ENTITY IfcCartesianPoint
  SUBTYPE OF (IfcPoint);
  Coordinates : LIST [1:3] OF IfcLengthMeasure;
  DERIVE
    Dim : IfcDimensionCount := HIINDEX(Coordinates);
  WHERE
    WR1 : HIINDEX(Coordinates) >= 2;
END_ENTITY;

```

Abbildung 7: Funktionen und Regeln innerhalb der Entität IfcCartesianPoint [8]

1.3 IFC-Model (STEP-Datei)

Produktbezogene Informationen zu einem individuellen Bauwerk werden im IFC-Model gespeichert. Es ist eine Text-Datei, die im internationalen STEP-Format geschrieben ist. STEP steht für **ST**andard for the **E**xchange of **P**roduct data und ist durch ISO 10303 genormt. Der Aufbau einer im STEP-Format verfassten Textdatei erlaubt den Austausch von produktspezifischen Informationen zwischen Softwareanwendungen.

Im Header der STEP-Datei werden allgemeine IFC-Model-Informationen wie Dateiname, Erstellungsdatum, Autor usw. erfasst. Unter dem Eintrag FILE_SCHEMA wird das zugehörige EXPRESS-Schema erwähnt. Im Beispiel aus der Abbildung 8 handelt es sich um IFC4.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('IFC4'), '2; 1');
FILE_NAME ('slabBridge.stp', '2013-08-30T09:14:44', (''), ('', ''), '', 'IfcProject', '');
FILE_SCHEMA (('IFC4'));
```

Abbildung 8: Definition des Headers einer STEP-Datei

Im Hauptteil der STEP-Datei werden konkrete Objekte instanziiert. Die Klasse eines Objektes entspricht einer bestimmten Entität, die im zugehörigen EXPRESS-Schema definiert ist. Alle Objekte besitzen eine eindeutige Nummerierung und sind somit inhaltlich voneinander abgegrenzt, können jedoch über ihre Attribute miteinander verknüpft werden.

In der Abbildung 9 werden vier kartesische Punkte (IFCCARTESIANPOINT) definiert. Wie wir aus dem vorigen Kapitel erfahren haben, besitzt der kartesische Punkt als Attribut seine kartesischen Koordinaten (Coordinates). In der STEP-Datei können entweder zwei oder drei Koordinaten definiert werden. Werden nur zwei eingetragen, wird die z-Koordinate automatisch zu 0 gesetzt. An der Position #5 wird ein Polygonzug (IFCPOLYLINE) instanziiert. Unter dem Listenattribut Points werden die Polygonpunkte definiert. In unserem konkreten Beispiel entsprechen sie den unter #1 bis #4 instanziierten kartesischen Punkten. Infolge des Referenzierens dieser Punkte innerhalb des Polygonzuges entsteht ein Rechteck mit Abmessungen 14 (Länge) und 3 (Breite).


```
#1=IFCCARTESIANPOINT((0.0, 7.0));  
#2=IFCCARTESIANPOINT((3.0, 7.0));  
#3=IFCCARTESIANPOINT((3.0, -7.0));  
#4=IFCCARTESIANPOINT((0.0, -7.0));  
#5=IFCPOLYLINE((#1, #2, #3, #4));
```

Abbildung 9: Polygonzug (IfcPolyline) in der IFC-Datei (STEP-Format)

Durch Polygonzüge entstehen Flächen, aus denen räumliche Objekte erzeugt werden können. Ein hierarchischer Aufbau der geometrischen Elemente und weitere Operationen mit räumlichen Objekten, die in späteren Kapiteln detaillierter beschrieben werden, ermöglichen die Abspeicherung komplizierter Geometrien im IFC-Model.

2 IFC-Bridge

2.1 Entwicklung von IfcBridge und aktueller Stand

Der IFC-Standard (die aktuelle Version von buildingSMART IFC2x4) ist für Gebäudebauwerke konzipiert. Eine detaillierte Beschreibung der speziellen Brückenelemente ist aus diesem Grund nicht möglich. Die französische Organisation CSTB und ein japanisches Forschungsteam entwickelten zunächst unabhängig voneinander ein eigenes, auf IFC aufbauendes Produktmodell für Brücken [9]. Die Kooperation der beiden Teams ergab, dass ihre Entwicklungen auf ähnlichen Ansätzen basierten waren, und man entschloss sich, die Erweiterung des Formats IfcBridge, das die Beschreibung von IFC-Brücken ermöglichen sollte, gemeinsam auszuarbeiten. Neue IFC-Bridge-Entitäten (ENTITY) und IFC-Bridge-Typen (TYPE) wurden eingefügt. Anhand eines einfachen Beispiels werden im nachfolgenden Kapitel die wichtigsten IFC-Bridge-Elemente erläutert.

2.2 Brücke mit Massivplattenquerschnitt in IFC

In diesem Kapitel wird anhand einer einfachen Brücke erläutert, wie diese mit den Ansätzen des bisher entwickelten IfcBridge-Formats abgespeichert werden kann. Es handelt sich um eine Rahmenbrücke mit Massivplattenquerschnitt (Abbildung 10), die gewöhnlich bei Autobahnunterführungen mit kleinen Stützweiten ihre Anwendung findet.



Abbildung 10: Massivplatte [11]

In der Abbildung 11 sieht man den ersten Teil des hierarchischen Aufbaus der IFC-Brücke. IfcProject ist das übergeordnete Objekt, das aus einer beliebigen Anzahl an Bauobjekten bestehen kann. In IFC werden Beziehungen zwischen Objekten über die Klasse IfcRelationship abgebildet. Dies erlaubt, die spezifische Eigenschaft bestimmter Beziehungen durch ein konkretes „Beziehungs“-Objekt zu definieren und macht es somit möglich, diese Eigenschaften in Zukunft zu verändern, ohne dass die Objekte selbst betroffen sind. In unserem Beispiel sorgt IfcRelAggregates (eine Kindklasse von

IfcRelation) für die Beziehung der Objekte IfcBridge (die Brücke selbst) und IfcProject. Das Projekt kann mehrere Brücken beinhalten.

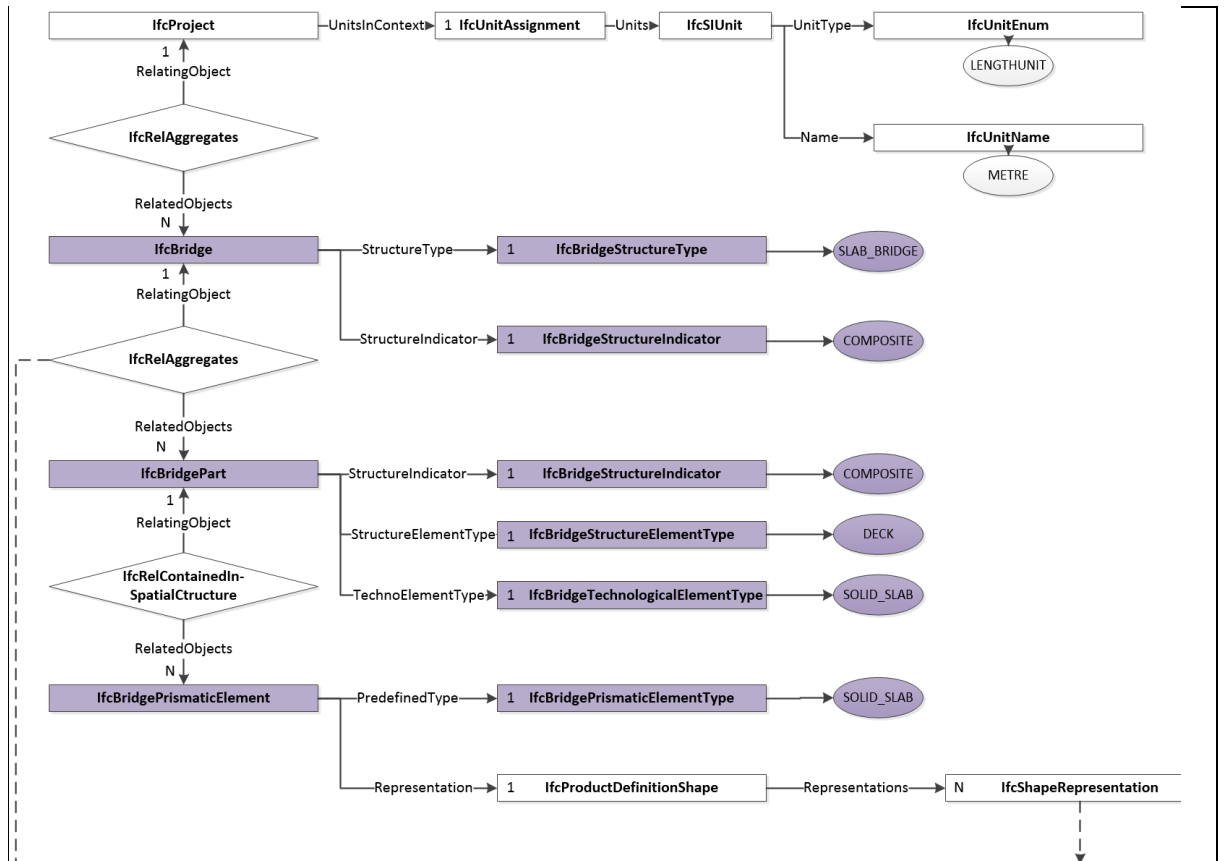


Abbildung 11: Aufbau einer Brücke in IFC. IfcBridge-Elemente (violett), IFC-Standard (weiß)

Das Tragsystem der Brücke wird durch das Attribut **StructureType** charakterisiert. Der Aufzählungstyp **IfcBridgeStructureType** bietet eine Reihe von Brücken-Tragsystemen: z.B. Platten-, Fachwerk-, Bogen-, Hängebrücke usw. Da die Beispielsbrücke eine Massivplatte als Querschnitt besitzt, wird dem Attribut **StructureType** der Wert **SLAB_BRIDGE** zugewiesen. Die Klasse **IfcBridge** erbt von ihrer abstrakten Elternklasse **IfcBridgeStructureElement** das Attribut **StructureIndicator**. Dieses beschreibt die verwendete Werkstoffart der Brücke. Im vorliegenden Beispiel handelt es sich um eine Stahlbetonbrücke, deswegen wird hier der Eintrag **COMPOSITE** (Verbundwerkstoff) gewählt.

Die Brücke (**IfcBridge**) besteht aus mehreren Brückenteilen (**IfcBridgePart**). Diese Beziehung wird ebenfalls in einem **IfcRelAggregates**-Objekt gespeichert (Abbildung 11). Die **IfcBridgePart** beschreibt eine bestimmte Brückenkomponente wie bspw. die

Tragfläche (DECK), ein Pylon (PYLON), ein Tragseil (CABLE), den Vorbauschnabel (LAUNCHING_NOSE), eine Hilfsstütze (TEMPORARY_BENT) usw. Da das `IfcBridgePart`-Objekt ebenso wie `IfcBridge` von der `IfcBridgeStructureElement` ihre Eigenschaften erbt, kann die Werkstoffart der `IfcBridgePart` mit dem Attribut `StructureIndicator` definiert werden.

2.3 Darstellung von Überbauplatten mit veränderlichen Querschnitten in `IfcBridge`

Das tragende Bauteil der Brücke ist die Überbauplatte (DECK). Diese wird über ein `IfcBridgePart`-Objekt beschrieben. Um größere Stützmomente aufnehmen zu können, wird der Untergurt der Überbauplatte mit einer Krümmung (Voute) gebaut. Es entsteht eine veränderliche Plattenhöhe entlang der Brücke (siehe Abbildung 12). Um die veränderliche Höhe des Überbaus (Abbildung 12) darstellen zu können, benötigt man mehrere Objekte der Klasse `IfcBridgePrismaticElement`, die mit Hilfe des Objektes `IfcRelAggregates` der Überbauplatte (`IfcBridgePart`) untergeordnet werden.



Abbildung 12: Gewutete Balkenbrücke, Donaubrücke Pöchlarn [12]

Den einzelnen Brückenabschnitten (`IfcBridgePrismaticElement`) wird jeweils ein geometrisches Repräsentationsobjekt zugewiesen. Im Rahmen des `IfcBridge`-Projektes wurde eine neue geometrische Repräsentationsmethode entwickelt, die sich für die Beschreibung von beliebigen Brückenüberbauten optimal eignet – `IfcReferencedSectionedSpine`. Die damit verbundenen Ansätze ermöglichen die Beschreibung von Objekten mit veränderlichen Querschnitten. Die Hierarchie der Überbauplatte, die über das Objekte `IfcReferencedSectionedSpine` beschrieben ist, zeigt die Abbildung 13.

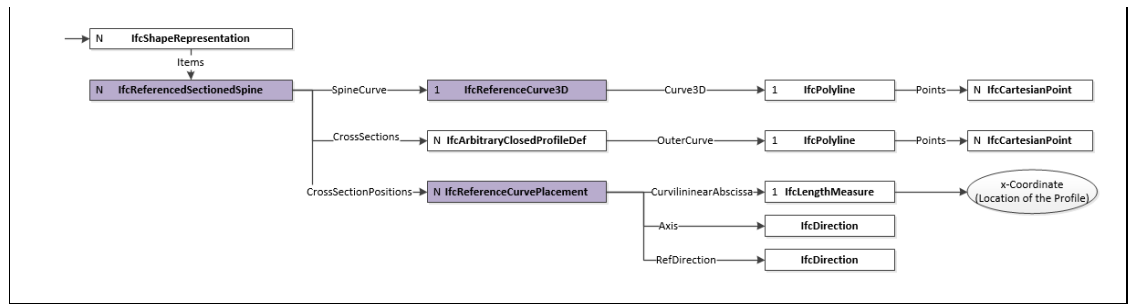


Abbildung 13: Darstellung des Überbaus mit Hilfe von IfcReferencedSectionedSpine

Mehrere Querschnitte (IfcProfileDef oder IfcArbitraryClosedProfileDef), die eine unterschiedliche 2D-Geometrie besitzen können, werden auf einer Referenzkurve (IfcReferenceCurve3D) an bestimmten Stellen (IfcReferenceCurvePlacement) positioniert und entlang der Referenzkurve extrudiert (siehe Abbildung 14).

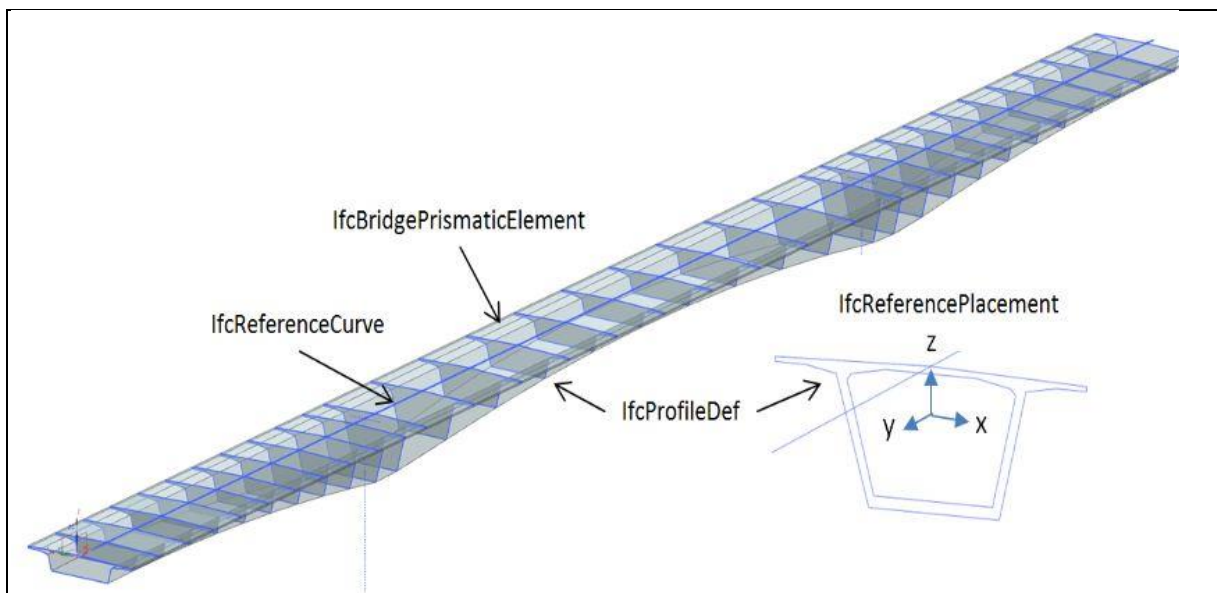


Abbildung 14: IfcReferencedSectionedSpine dient zur Darstellung des Überbaus [13]

Mit IfcArbitraryClosedProfileDefintion kann ein Vollquerschnitt dargestellt werden. Dabei definiert das Attribut OuterCurve die äußere Randkurve des Querschnitts. Möchte man einen Hohlkastenquerschnitt darstellen, bietet die untergeordnete Klasse IfcArbitraryClosedProfileDefWithVoids an, zusätzlich eine oder mehrere innere Kurven über das Listenattribut InnerCurve festzulegen (Abbildung 15).

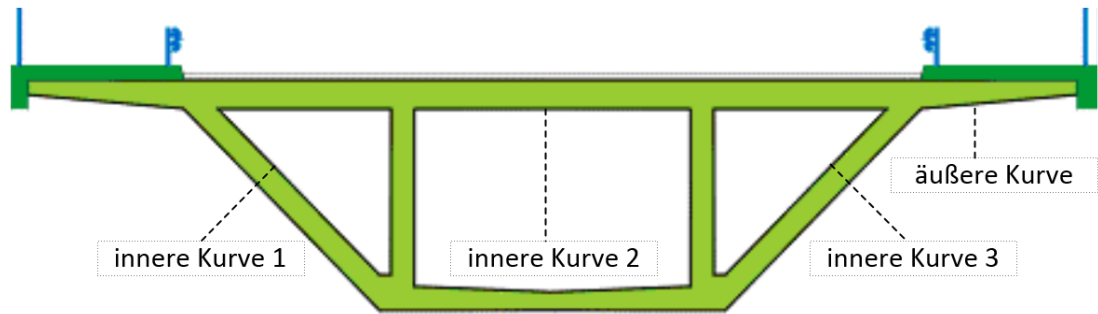


Abbildung 15: Konstruktionsprinzip des Hohlkastenquerschnitts mit drei Zellen in IFC [11]

Die Klasse `IfcReferenceCurvePlacement` definiert die eindeutige Position des jeweiligen Querschnitts über folgende Attribute:

- **CurvilinearAbscissa:**
X-Koordinate entlang der Referenzkurve bei gerader Trassierung oder die Bogenlänge bei gekrümmter Trassierung. An dieser Stelle wird der entsprechende Querschnitt positioniert.
- **Axis:**
der Richtungsvektor der lokalen z-Achse
- **RefDirection:**
der Richtungsvektor der lokalen x-Achse

Somit kann auch ein gekrümmter Brückenüberbau relativ leicht in IFC modelliert werden. Dazu muss lediglich die Referenzkurve einen gekrümmten Verlauf aufweisen. Die Extrusion entlang der Referenzkurve erzeugt die räumliche Struktur des Überbaus. Aufgrund der Interpolation zwischen zwei Profilen mit unterschiedlichen Querschnitten entsteht eine flüssige Veränderung des Querschnitts, und man spricht von der variablen Extrusion [13].

Das `IfcBridge`-Team hat einen Vorschlag entwickelt, wie man den Brückenüberbau in IFC beschreiben kann. Dies ermöglicht die Darstellung der überbauspezifischen Brückenelemente wie `DECK`, `CABLE`, etc. Ein Konzept für die Speicherung des Unterbaus einer Brücke in IFC existiert aktuell noch nicht. Im Rahmen der vorliegenden Masterarbeit wird zunächst eine Möglichkeit vorgeschlagen, die Brückenunterbauelemente wie Widerlager, Pfeiler, Fundamente usw. in IFC abzuspeichern und geometrisch bzw.

topologisch darzustellen. Anschließend wird ein Konzept erarbeitet, wie die Parametrik einer Brücke in IFC-Dateien integriert werden kann.

3 Unterbauten einer Brücke

3.1 Elemente des Brückenunterbaus

Bei kleinen bis mittleren Brücken beträgt der Aufwand für die Unterbauten 60% des Gesamtaufwandes [14]. Das begründet die Notwendigkeit einer gründlichen Planung dieser Elemente. Die Hauptaufgabe der Unterbaukonstruktion ist die Lagerung des Überbaus und die Aufnahme der Belastung aus Eigengewicht und Verkehr. Dabei müssen alle Einwirkungen, sowohl horizontale wie vertikale, als auch die aus den Verformungen des gesamten Bauwerks in die Gründung übertragen werden. Die Unterbaukonstruktion besteht aus Widerlagern, Stützen, Pfeilern und deren Gründungen. In den folgenden Kapiteln wird die konstruktive Ausbildung der Unterbauelemente näher erläutert. Kapitel 4 stellt anschließend Methoden vor, wie die Elemente des Brückenunterbaus in IFC abgespeichert werden können.

3.2 Widerlager

Als abschließende Brückenbauwerke sollen die Widerlager horizontale und vertikale Lasten sicher in den Baugrund einleiten [14]. Die Abbildung 16 veranschaulicht ein kastenförmiges Widerlager mit seinen Bestandteilen.

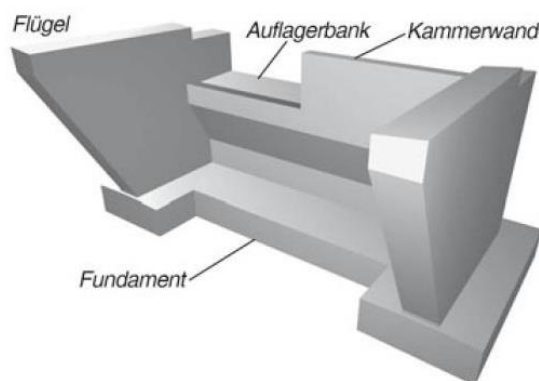


Abbildung 16: Kastenförmiges Widerlager - Prinzipdarstellung [14]

Die horizontalen Kräfte aus dem Erddruck übernimmt die massive Wand an der Stirnseite des Widerlagers. Die Widerlagerwand gilt gleichzeitig als Stützbauwerk für den beginnenden Verkehrsdamm. Die hinter dem Überbauende angeordnete Kammerwand schützt den Überbau gegen das Eindringen des Hinterfüllungsmaterials in das Innere des

Widerlagers. Die Tiefe der Auflagerbank hängt von der Konstruktion des Überbaus und der erforderlichen Längsbeweglichkeit der Brücke ab [15] (Abbildung 17).

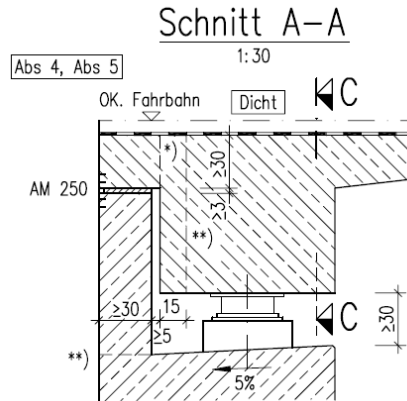


Abbildung 17: Anschluss Überbau an Widerlager mit Kammerwand, Auflagerbank und Lagerung [17]

Bei größeren Stützweiten und damit größeren Überbauhöhen, werden spezielle Übergangskonstruktionen benötigt, um die Verformungen aus z.B. Temperaturlasten aufzunehmen. Diese Konstruktion wird in der Kammerwand verankert.

Die Flügelwände sind wandartige Bauelemente des Widerlagers, die das Hinterfüllungsmaterial im Böschungsbereich seitlich abfangen [15]. Der Abstand zwischen den Flügelwänden entspricht im Normalfall der Fahrbahnbreite. Die Höhe der Flügelwände hängt von der Einbindung des Widerlagers in die Landschaft ab. Am höchsten sind die Flügelwände dann, wenn die Widerlagerstirnwand direkt am Böschungsfuß ansteht [17]. In diesem Fall erhalten die Flügelwände ihre eigene Gründung und es entsteht ein kastenförmiges Widerlager. Wenn das Widerlager in die Böschung zurückgesetzt ist, können relativ niedrige Flügelwände ausgeführt werden. Hohe Flügelwände erwecken die Torwirkung, niedrige dagegen unterstreichen die Landschaft und verbessern die Sicht für die Autofahrer (Abbildung 18).

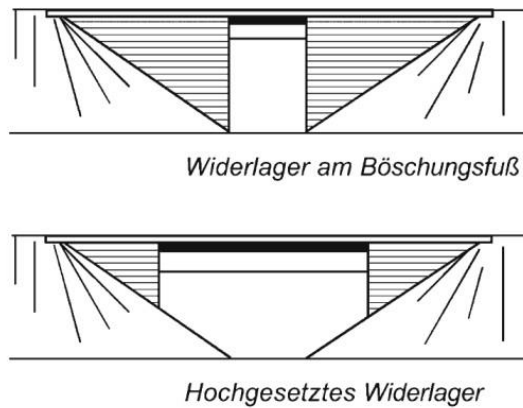


Abbildung 18: Anordnung von Widerlagern im Böschungsbereich [14]

Wird das Widerlager weit genug in die Böschung zurückgesetzt, können die Flügelwände sogar komplett entfallen und es bleibt eine einfache Widerlagerstützwand (Abbildung 19).

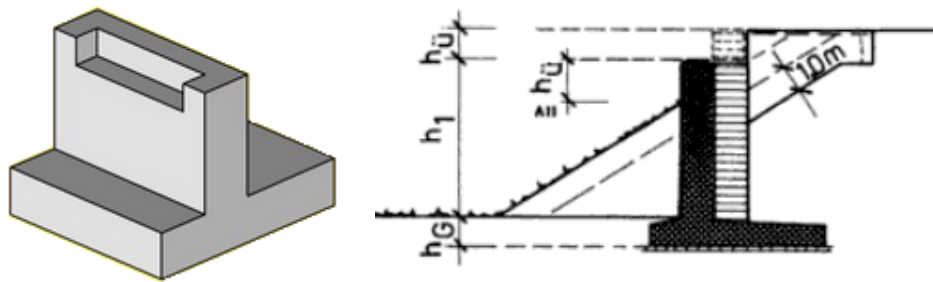


Abbildung 19: Einfache Winkelstützwand [12] [15]

Das Widerlager besteht also aus einer Winkelstützwand, an die seitlich kleine Kragflügel angehängt sind. Diese dienen der seitlichen Abschirmung der Hinterfüllung. Die am häufigsten verwendete Widerlagerform ist ein kastenförmiges Widerlager. Nach den Richtzeichnungen für Ingenieurbauten [17] stellt das in Abbildung 20 dargestellte Widerlager eine Regelform dar.

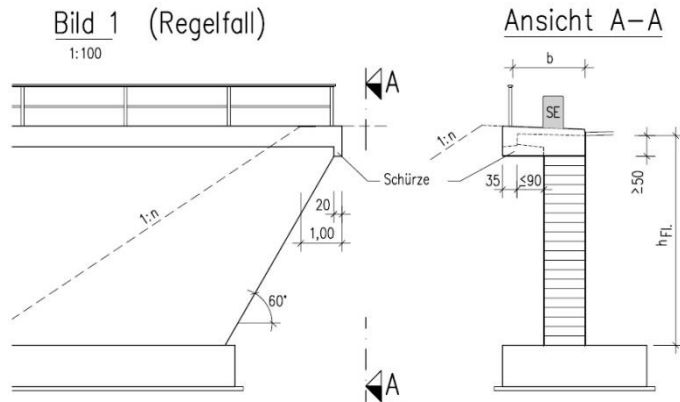


Abbildung 20: Regelform eines kastenförmigen Widerlagers [17]

Das schräge Flügelwandende trägt zur stabilen Verankerung des Widerlagers in dem anstehenden Damm bei. In der Regelausführung aus der Abbildung 20 beträgt der Winkel der Schräge über die komplette Widerlagerhöhe 60° , unabhängig von der Böschungsneigung. Die Breite der Flügelwand verjüngt sich also mit der Widerlagertiefe. Bei großen Talbrücken mit hohen Überbauquerschnitten wird die Regelausführung aus wirtschaftlichen Gründen unbrauchbar [15]. Ab einer bestimmten Widerlagerhöhe ist es wirtschaftlicher, die durchgehenden schrägen Flügelwände aufzulösen. Dabei entsteht die in der Abbildung 21 dargestellte Widerlagerform.

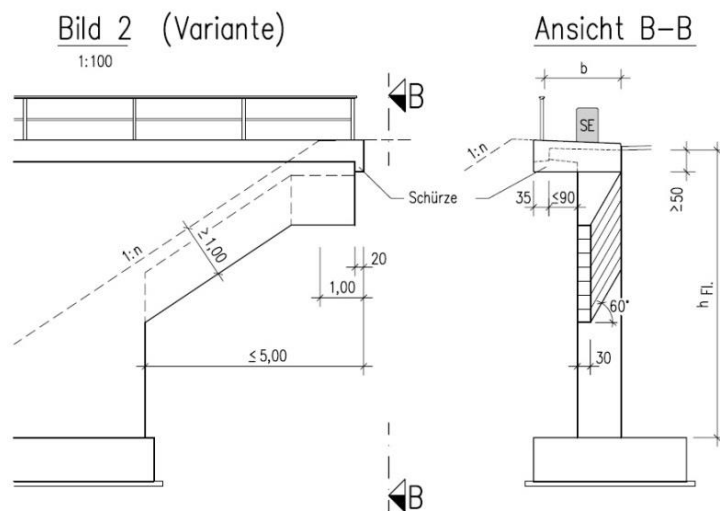


Abbildung 21: Alternative Widerlagerform bei hohen Talbrücken [17]

Bei dieser Variante gleicht der Winkel der schrägen Unterkante der Böschungsneigung. Als Regelböschungsneigung ist das Verhältnis 1:1,5 vorgeschrieben. Eine andere Böschungsneigung als die Regelneigung kann vorgesehen werden [17]:

- aus erdstatischen Gründen
- zur Einpassung der Straße in das umgebende Gelände
- aus Gründen des Immissionsschutzes

Das Flügelende befindet sich 1,0 m hinter dem Schnittpunkt der Straßenböschung und dem Planum der überführten Straße. Die Abbildung 22 zeigt ein kastenförmiges Widerlager, das sich noch im Bauzustand befindet. Bei der fertigen Konstruktion wird ein Großteil der Flügelwände unter der Erde verschwinden und nicht mehr sichtbar sein.



Abbildung 22: Ein kastenförmiger Pfeiler im Bauzustand [19]

3.3 Pfeiler und Stützen

Um die Belastung bei langen Brücken abzutragen, werden zusätzlich zu den Widerlagern Pfeiler und Stützen aufgestellt. Einwirkungen aus dem Eigengewicht des Überbaus und aus den Verkehrslasten werden über die Pfeiler auf die Fundamente und anschließend in den Baugrund eingeleitet.

Pfeiler oder Pfeilerscheiben sind wandartige Bauelemente, die sich über die komplette Breite des Überbaus erstrecken. Solchen Konstruktionen begegnet man oft bei

Autobahnüberquerungen. Meist einfache, rechteckige Querschnittsabmessungen werden mit gerundeten oder gebrochenen Ecken angeschlossen (Abbildung 23).



Abbildung 23: System der Pfeilerscheibe (links) [14] und eine Autobahnunterführung [20]

Die Breite des Pfeilers ist eng mit der Form des Überbaus verbunden. Bei massiven und breiten Fahrbahnplatten können mehrere Pfeiler in der Reihe aufgestellt werden [14].

Massivere Pfeiler trifft man bei Brücken über Seen oder Flüssen an. In der Regel besitzt diese Variante niedrige Höhen, dafür aber große Querschnitte. Kreisförmige oder elliptische Abschlüsse werden vorgesehen, um den Pfeiler besser in den Flusstrom einzupassen [15] (Abbildung 24).



Abbildung 24: Treskowbrücke über die Spree in Berlin [21]

Bei Straßen-Brücken schränkt die massive Pfeilerform die Sichtverhältnisse unter der Brücke ein. Eine bessere Lösung in dieser Hinsicht bieten einzelne Stützen, die im

gleichen Abstand in Reihe aufgestellt werden (Abbildung 25). Die Querschnittsform der Einzelstützen kann quadratisch, rechteckig oder rund sein.

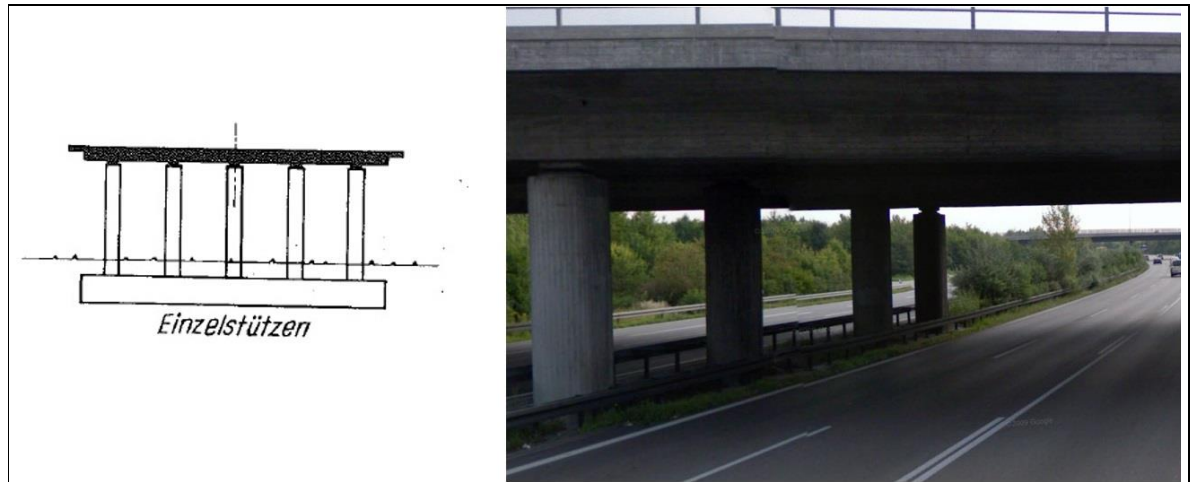


Abbildung 25: Unterstützung des Überbaus mit Einzelstützen. System [14] und Autobahnbrücke [20]

Eine weitere häufig eingesetzte Form der Brückenunterstützung ist die Rahmenstütze mit Zugband. Diese Ausführung findet meistens bei Brückenüberbauten aus Fertigteilträgern ihre Anwendung [14] (Abbildung 26).

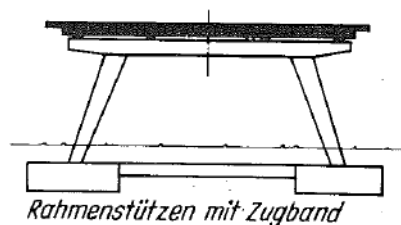


Abbildung 26: Rahmenstütze mit Zugband [14]

Muss der Stützenkopf mehrere Lager aufnehmen, werden größere Stützenquerschnitte benötigt, um die Lasten dennoch aufnehmen zu können. Ein konstanter Querschnitt konkurriert in diesem Fall mit dem Ziel der Wirtschaftlichkeit. Aus diesem Grund werden Stützen mit veränderlichen Querschnitten gebaut, die sich dem Schnittgrößenverlauf anpassen. Eine weitere Möglichkeit, um den hohen Lasten entgegenzuwirken, ist die Ausbildung eines sogenannten Hammerkopfes am oberen Ende der Stützen (Abbildung 27).

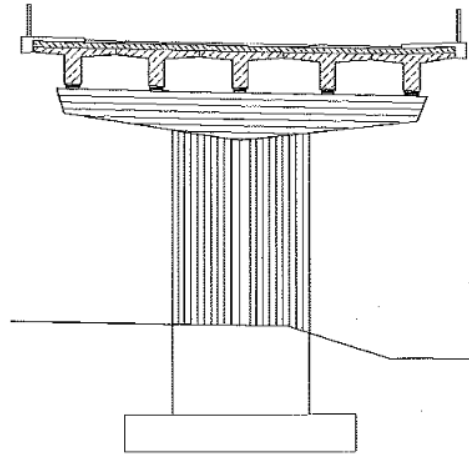


Abbildung 27: Hammerkopfausbildung [15]

3.4 Gründungen

Die Gründung der Widerlager und Pfeiler oder Stützen soll die Lasten aus dem gesamten Brückenbauwerk sicher und ohne schädliche Setzungen in den tragfähigen Boden übertragen. Die Gründung der Widerlager erfolgt entweder über Fundamentplatten (Flachgründung) oder über Bohrpfähle (Tiefgründung) [15]. Die Wahl der Gründungsart hängt hauptsächlich von der bodenmechanischen Situation des jeweiligen Standortes ab.

Flachgründungen sind nur dort möglich, wo ein tragfähiger Baugrund bereits in geringer Tiefe ansteht. Der Grundriss der Fundamente besitzt meist eine ähnliche Form wie der untere Querschnitt des über ihm stehenden Bauteils. Wenn die tragfähigen Bodenschichten dagegen erst in großen Tiefen vorliegen, kommt die Tiefgründung zum Einsatz. Dabei werden die Lasten über die Ramm- oder Bohrpfähle in die tiefliegenden, tragfähigen Schichten übertragen.

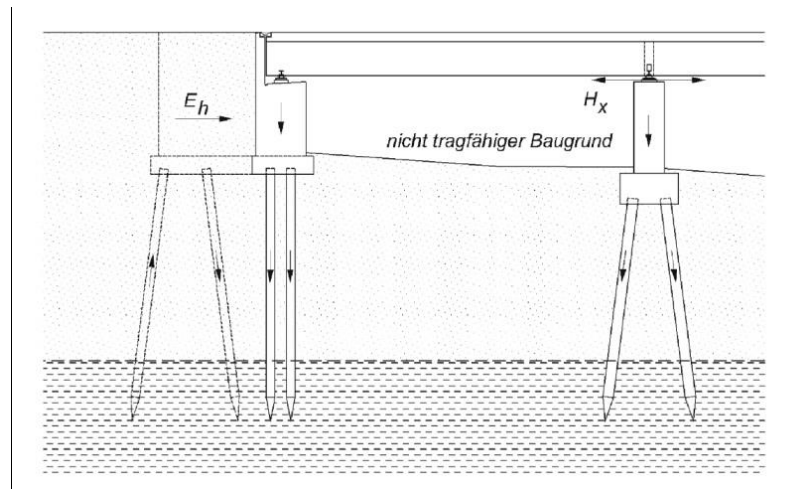


Abbildung 28: Pfahlgründung mit schrägen Pfählen [9]

Durch die schräge Stellung der Pfähle können horizontale Kräfte aus dem Erddruck und aus den Anfahr- und Bremslasten besser abgetragen werden [15].

4 Beschreibung des Brückenunterbaus in IFC

4.1 Struktur des Unterbaus

Der Unterbau einer Brücke ist ein Sammelbegriff für mehrere Brückenelemente wie Widerlager, Pfeiler und deren Gründungen. Wie in den vorigen Kapiteln erläutert wurde, kann jedes von diesen Elementen unterschiedliche Strukturen annehmen, abhängig von diversen technischen, funktionellen oder topologischen Rahmenbedingungen. Analog zu den Längs- und Quersystemen des Überbaus muss auch die Struktur der Unterbauelemente in IFC festgehalten werden. Dazu wird das EXPRESS-Schema von `IfcBridge` (siehe Kapitel 2) um diverse Elemente erweitert. Dazu zählen sowohl neue Entitäten als auch neue Eintragungen in den Aufzählungen und Auswahllisten. Im Folgenden werden diese Änderungen im Detail erläutert.

Um dem hierarchischen Aufbau des IFC-Formats gerecht zu werden, ist ein übergeordnetes „Kopf“-Objekt notwendig, das weitere untergeordnete Objekte des Unterbaus zusammenfasst. Dazu ist keine neue Entität notwendig, sondern es kann die bereits in `IfcBridge` vorhandene Entität `IfcBridgePart` verwendet werden. `IfcBridgePart` ist der Sammelbegriff für jegliche Unterelemente der Brücke, wozu auch der Brückenunterbau zählt. Die Charakterisierung der `IfcBridgePart` erfolgt über das Attribut Struktur-Element-Typ (`StructureElementType`). Dieses beinhaltet eine Aufzählung von diversen Brückenteilen wie Überbauplatte (`DECK`), Pylon (`PYLON`), Vorbauschnabel (`LAUNCHING_NOSE`) usw. Dieser Liste wird die Option Unterbau (`SUBSTRUCTURE`) hinzugefügt (Abbildung 29). Das ermöglicht eine Abspeicherung des Objektes `IfcBridgePart` vom Typ `SUBSTRUCTURE` (Abbildung 30).

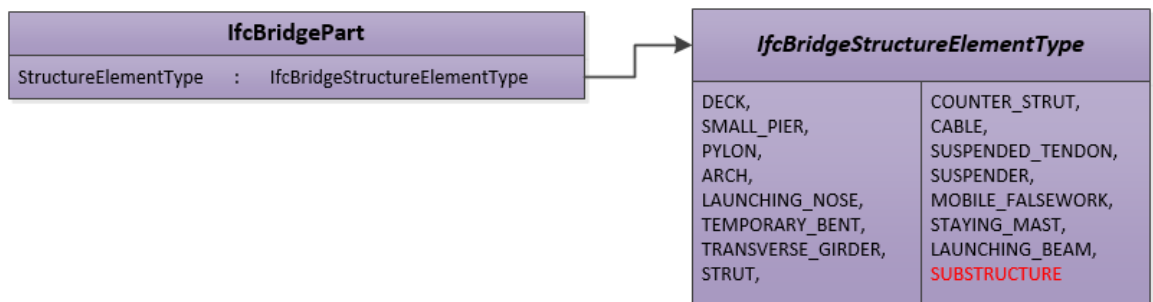


Abbildung 29: Erweiterung des Strukturtyps um den Eintrag SUBSTRUCTURE

Einen möglichen Eintrag im IFC-Model zeigt die Abbildung 30:

```
#8 = IFCBRIDGEPART ($, $, $, $, $, $, $, $, $, $, $, $, .SUBSTRUCTURE., $);
```

Abbildung 30: Definition des Brückenunterbaus in der IFC-Datei

Die einzelnen Elemente des Brückenunterbaus repräsentiert die Klasse `IfcBridgeSubstructureElement`, die auf der gleichen Hierarchieebene wie die Klasse `IfcBridgePrismaticElement` angeordnet ist, und erbt die Eigenschaften der übergeordneten, abstrakten Klasse `IfcBridgeElement`. Die Abbildung 31 zeigt den hierarchischen Weg zum Unterbauelement.

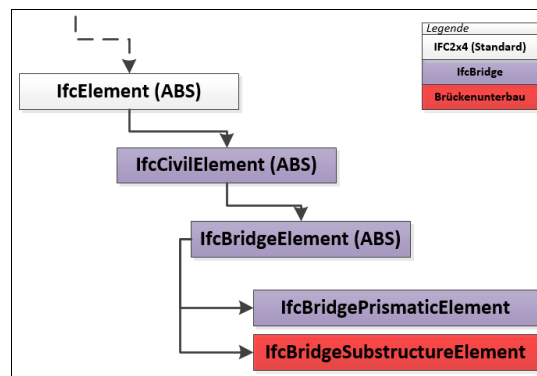


Abbildung 31: Hierarchie IfcElement

Die eingefügte Klasse `IfcBridgeSubstructureElement` verfügt über ein Attribut Unterbautyp (`SubstructureType`), welches die Art des jeweiligen Unterbauelementes beschreibt. `SubstructureType` ist vom Typ `IfcBridgeSubstructureElementType` (Unterbautyp), worin diverse Arten der Brückenunterbauelemente in Form einer Aufzählung (ENUMERATION) aufgelistet sind (Abbildung 32).

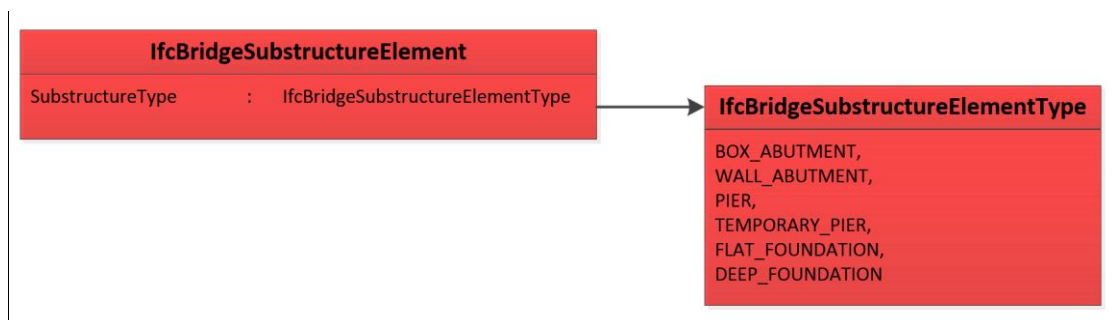


Abbildung 32: Definition der Art des Unterbauelementes mittels `IfcBridgeSubstructureElementType`

Die neu eingeführten Klassen ermöglichen die Beschreibung von folgenden Unterbauelementen:

- Kastenförmiges Widerlager (BOX_ABUTMENT)
- Wandwiderlager (WALL_ABUTMENT)
- Pfeiler / Stütze (PIER)
- Temporäre Stütze im Bauzustand (TEMPORARY_PIER)
- Flachgründung (FLAT_FOUNDATION)
- Tiefgründung (DEEP_FOUNDATION)

Der entsprechende Eintrag in der IFC-Datei könnte beispielsweise so umgesetzt werden:

```
#71 = IFCBRIDGESUBSTRUCTUREELEMENT ($, $, $, $, $, $, #70, $, .BOX_ABUTMENT.);
```

Abbildung 33: Definition des Objektes IfcBridgeSubstructureElement in der IFC-STEP-Datei

Mit Hilfe der IFC-Aggregations-Klasse IfcRelContainedInSpatialStructure können mehrere Unterbauelemente einer einzigen IfcBridgePart zugeordnet werden. Die Abbildung 34 zeigt, wie die Aggregation von 6 verschiedenen Unterbauelementen unter einem Objekt der Klasse IfcBridgePart im IFC-Model umgesetzt wird.

```
#10 = IFCBRIDGEPART($,$,$,$,$,$,$,$,$,$, .SUBSTRUCTURE., $);  
#20 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #169, $, .BOX_ABUTMENT.);  
#30 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #158, $, .BOX_ABUTMENT.);  
#40 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #184, $, .PIER.);  
#50 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #195, $, .FLAT_FOUNDATION.);  
#60 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #234, $, .FLAT_FOUNDATION.);  
#70 = IFCBRIDGESUBSTRUCTUREELEMENT($,$,$,$,$,$,$,$, #257, $, .FLAT_FOUNDATION.);  
#80 = IFCRELCONTAINEDINSPATIALSTRUCTURE($,$,$,$,$, (#20, #30, #40, #50, #60, #70), #10);
```

Abbildung 34: Beispiel einer Aggregation der Unterbauelemente unter einem IfcBridgePart-Objekt

Alle zuvor genannten Ergänzungen des IfcBridge-Models besitzen einen rein informativen Charakter. Sie werden dazu benutzt, um einerseits die Art der Brücke (Platten-, Hänge-, Bogenbrücke usw.) zu beschreiben und andererseits festzustellen, aus welchen weiteren Unterelementen (Überbauplatte, Pylone, Widerlager, Pfeiler usw.) die Brücke konstruiert ist. Im folgenden Kapitel geht es um die geometrische Beschreibung der Elemente des Brückenunterbaus. Es werden geeignete geometrische Repräsentationsmethoden

vorgestellt und angesichts der Anwendbarkeit für die Darstellung der Unterbauelemente miteinander verglichen.

4.2 Geometrische Darstellung des Unterbaus

Der aktuelle IFC-Standard verfügt über herkömmliche Methoden, mit deren Hilfe sich 2D- und 3D-Geometrien der Bauwerke und Bauwerksteile modellieren lassen. 2D-Elemente wie z.B. Punkt, Kurve oder Fläche können über die IFC-Objekte `IfcPoint`, `IfcCurve` bzw. `IfcSurface` dargestellt werden. Zur Darstellung von 3D-Komponenten existieren Techniken wie Randdarstellung, Extrusion, CSG-Körper usw. Über die Attribute der jeweiligen IFC-Objekte (`IfcFacetedBrep`, `IfcExtrudedAreaSolid`, `IfcCsgSolid`, usw.) werden speziellen Variablen Werte zugewiesen, die diverse Informationen, bspw. dimensionale Maße oder Verknüpfungsarten, beinhalten. Die Visualisierung des Bauwerks erfolgt durch ein Darstellungsprogramm (z.B. IFC-Viewer), das die vorher genannten Angaben aus der IFC-Datei entnimmt und IFC-Objekte in Bilder umsetzt.

4.2.1 Überblick über geometrische Darstellungsmethoden

Im Folgenden werden unterschiedliche Modellierungswege miteinander verglichen und anschließend wird analysiert, welche Vorgehensweise sich am besten dafür eignet, Brückenunterbauten darzustellen. Folgende Aspekte werden bei der Wahl der passenden Modellierungsmethode berücksichtigt:

- Anzahl der Objekte, die in der IFC-STEP-Datei erfasst werden (Speichervolumen).
- Interoperabilität mit anderen Modellierungssoftwares. Je einfacher die Gestaltung der Elemente, desto weniger aufwändig ist die Entwicklung der Schnittstellen in Softwareprodukte, in die die Brücken-Daten importiert werden sollen.
- Realisierung der parametrischen Abhängigkeiten zwischen den einzelnen Bauelementen innerhalb der IFC-Datei.

Boundary Representation

Die Boundary Representation ist eine Form der geometrischen Darstellung von Raumobjekten, bei der 3D-Objekte mit Hilfe der umhüllenden Flächen konstruiert werden [6]. Die Repräsentation erfolgt in einem hierarchischen Aufbau, in dem die Flächen von ihren Kanten umrandet und diese von ihren Ecken (Punkten) begrenzt werden. In IFC wird die Randdarstellungsmethode mit Hilfe der Klasse `IfcFacetedBrep` realisiert. Das übergeordnete Element – die Schale (`IfcClosedShell`) – erfasst die Menge aller Flächen (`IfcFace`) in ihrem Attribut `CfsFaces`. Die einzelnen Flächen (`IfcFace`) setzen sich aus Kanten zusammen, die bspw. über einzelne Polylinien (`IfcPolyline`) oder in Form eines Polygonzuges (`IfcPolyloop`) beschrieben werden. Es kann aber auch eine andere kurvenbeschreibende Klasse verwendet werden, die der abstrakten Klasse `IfcCurve` untergeordnet ist. Der Polygonzug wird durch die Verbindung seiner kartesischen Polygonpunkte (`IfcCartesianPoint`) konstruiert. In der Abbildung 35 wird das Prinzip der Boundary Representation anhand der Konstruktion eines Würfels veranschaulicht.

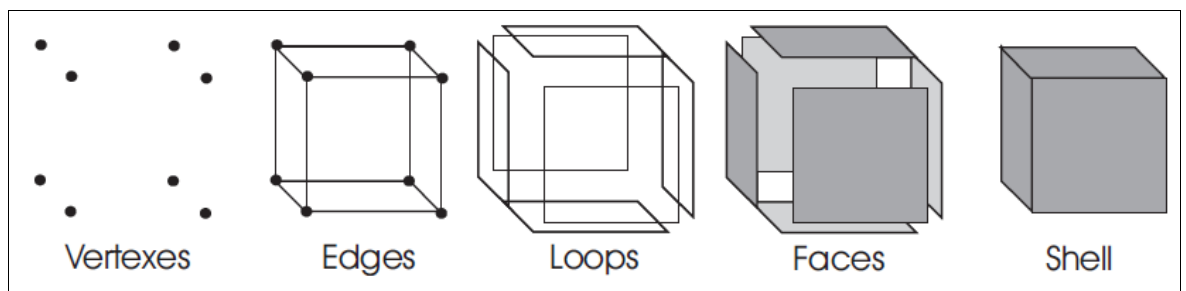


Abbildung 35: Prinzip des Randdarstellung-Modells (Boundary Representation) [22]

Das Beispiel aus der Abbildung 36 veranschaulicht anhand einer einfachen Wand mit rechteckigem Querschnitt und den Abmessungen 300x750x1400 (BreitexLängexHöhe), wie die `IfcFacetedBrep` (geometrische Repräsentation mit Randdarstellung) im IFC-Model umgesetzt wird.

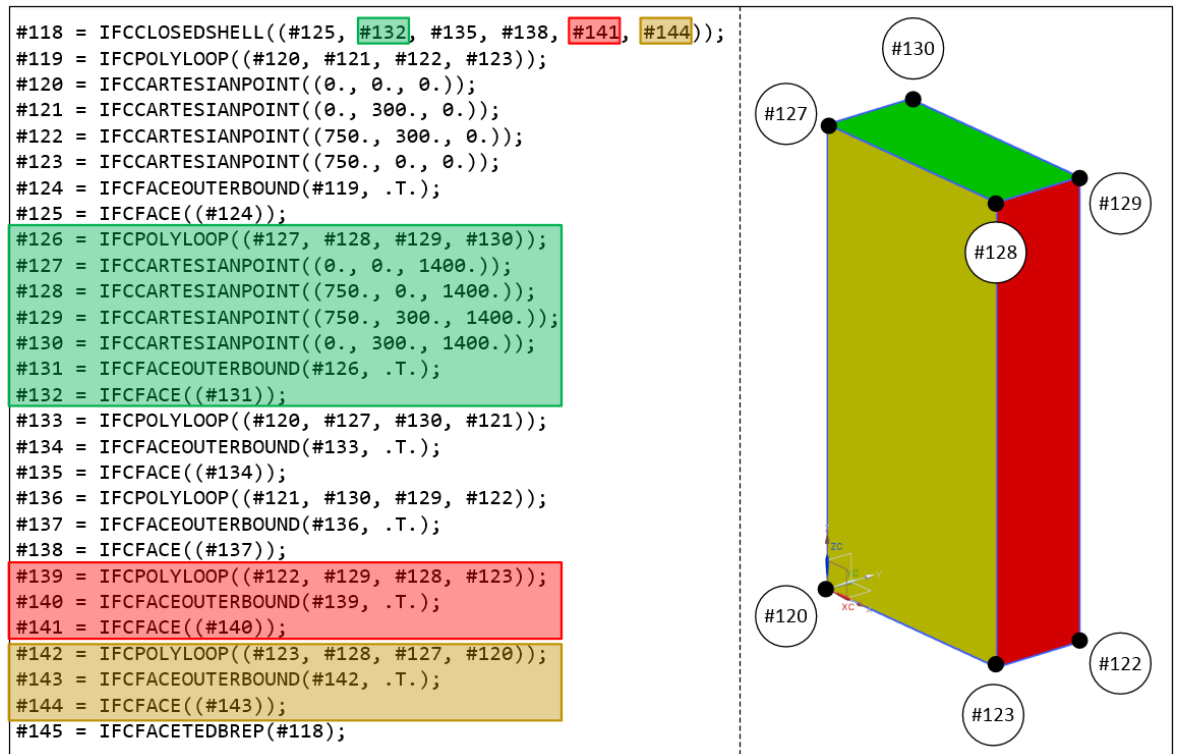


Abbildung 36: Beschreibung einer rechteckigen Wand mit IcfacedBrep

Ein großer Vorteil der BRep-Darstellung ist die Möglichkeit, Freiformflächen über beliebige Polygonformen zu konstruieren. Mit der Komplexität der darzustellenden Körper steigt jedoch der Aufwand für die Randdarstellungsmethode. Es werden immer mehr Flächen, Kanten und somit Punkte benötigt. Die Randdarstellung stößt an ihre Grenzen, sobald die Visualisierung gerundeter Bauteile notwendig wird. Da die umhüllenden Flächen aus dem BRep-Modell eine ebene Geometrie besitzen, kann ein Volumenkörper mit gerundeten Oberflächen nur durch eine Approximation dargestellt werden. Dabei erhalten die meist dreieckförmigen Randflächen (relativ) kleine Abmessungen, um die Abbildung des Bauteils genau und somit realistisch genug darzustellen. Je kleiner die Dreiecke, desto höher ist deren Anzahl, was die Zunahme an Kanten und Punkten zur Folge hat, die in der IFC-Datei erfasst werden müssen. Die dadurch entstehende Datenmenge erhöht den Speicherbedarf und steigert somit den Aufwand bei der Programmierung von komplexen Schnittstellen zu anderen Softwareanwendungen. Auch im Hinblick auf die parametrische Modellierung bringt die hohe Anzahl der BRep-Flächenelemente einen Nachteil mit sich, da jeder von diesen Teilelementen parametrische Abhängigkeiten erhalten muss und somit die Übersichtlichkeit des Modells darunter leidet.

Extrusion

Ein weiteres Werkzeug zur Generierung komplexer 3D-Körper stellt die Extrusion dar. Dabei wird eine 2D-Fläche mit einer beliebigen Form entlang einer bestimmten Richtung (Raumvektor) parallel verschoben [6]. Ein Zylinder entsteht, wenn ein Kreis senkrecht nach oben extrudiert wird (Abbildung 37).

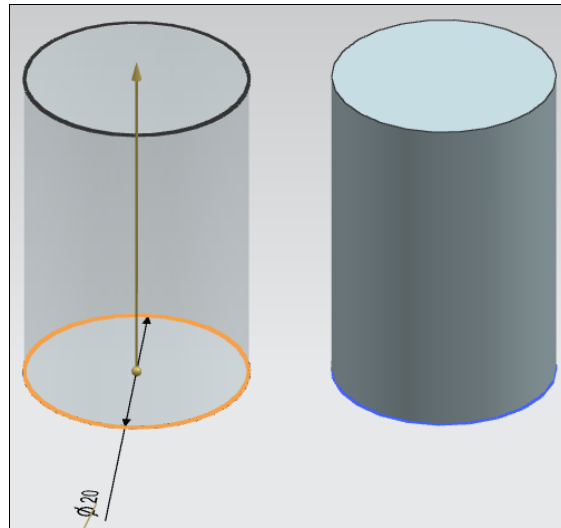


Abbildung 37: Erzeugung eines Zylinders infolge der Extrusion eines 2D-Kreises

In IFC ist die Klasse `IfcExtrudedAreaSolid` für die Extrusion zuständig. Das Attribut `SweptArea` beschreibt die zu extrudierende 2D-Fläche. Um z.B. geschlossenes Profil darzustellen, wird `IfcArbitraryClosedProfileDef` verwendet. `Position` ist ein weiteres Attribut von `IfcExtrudedAreaSolid`, das die exakte Position und Ausrichtung des zu extrudierenden 2D-Profiles festlegt. Dafür ist die Klasse `IfcAxis2Placement3D` geeignet, die unter anderem auch zur Definition des globalen dreidimensionalen Koordinatensystems verwendet wird. Im vorliegenden Fall jedoch speichert `IfcAxis2Placement3D` die Angaben zum lokalen Koordinatensystem der `SweptArea` (Extrusionsfläche) mit Hilfe der Attribute `Location`, `RefDirection` und `Axis`. Das Attribut `Location` definiert dabei den Ursprung des Koordinatensystems, `RefDirection` enthält die Informationen über die lokale x-Achse und `Axis` über die lokale z-Achse. Die Achsen werden in IFC mit Hilfe der Klasse `IfcDirection` beschrieben. Diese definiert drei Vektorkoordinaten der jeweiligen Achse. Schließlich werden in der übergeordneten Klasse `IfcExtrudedAreaSolid` die Extrusionsrichtung (`ExtrudedDirection`) und die Extrusionstiefe (`Depth`) festgehalten. Das Schema aus der Abbildung 38 veranschaulicht

einen möglichen Aufbau eines 3D-Extrusionskörpers. Im nachfolgenden Abschnitt dieses Kapitels wird die Beschreibung von 3D-Körpern mit `IfcExtrudedAreaSolid` anhand eines anschaulichen Beispiels näher erklärt.

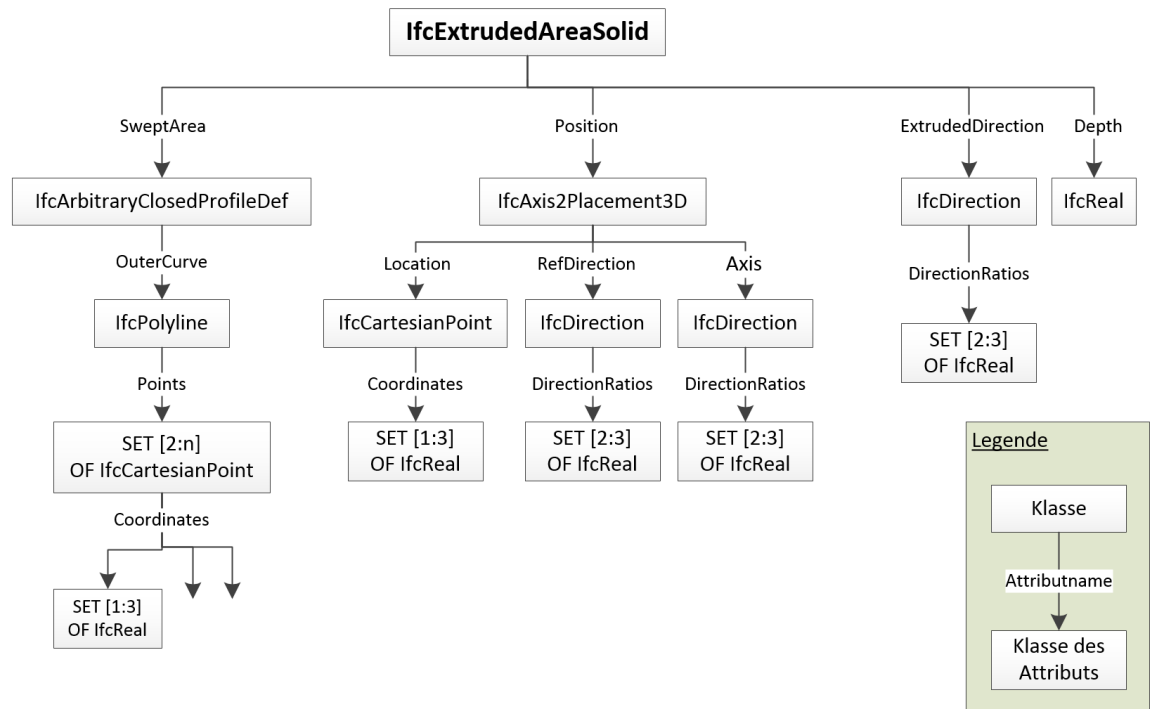


Abbildung 38: Aufbau eines IFC-Raumobjektes, das mit Extrusion beschrieben wird

Die Extrusion erweist sich als vorteilhaft bezüglich des geringen Datenvolumens, das benötigt wird, um komplexe 3D-Geometrien darzustellen. Auch Parametrik lässt sich mit nur relativ geringer Anzahl der parametrischen Abhängigkeiten ermöglichen.

CSG Darstellung

Constructive Solid Geometry (CSG) basiert auf der Verknüpfung von sogenannten 3D-Primitiven (Quader, Kugeln, Zylinder usw.) zur Darstellung von komplexen 3D-Objekten [6]. Die 3D-Primitiven werden mit Hilfe von booleschen Operationen wie Vereinigung, Differenz und Schnitt zu komplexen 3D-Geometrien verknüpft (Abbildung 39). In der Regel werden CSG-Modelle in einem CSG-Baum illustriert, an dem zu erkennen ist, welche primitiven Objekte durch welche booleschen Operationen miteinander verknüpft sind und wie letztendlich der Zielkörper entsteht [6].

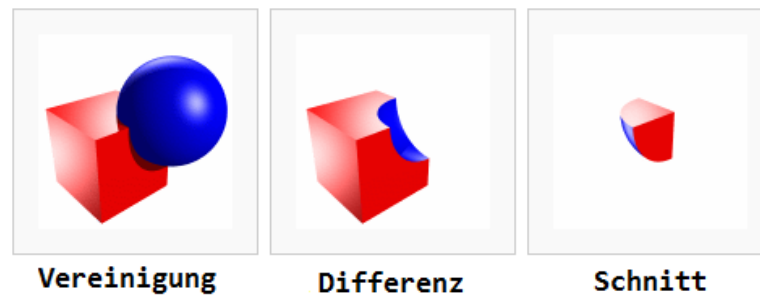


Abbildung 39: Boolesche Operation [19]

Die IFC-Klasse `IfcCsgSolid` gilt als das übergeordnete Objekt für die Darstellung von CSG-Körpern in IFC. Ihr einziges Attribut `TreeRootExpression` vom Typ `IfcBooleanResult` beschreibt den CSG-Baum. `IfcBooleanResult` erzeugt einen 3D-Körper aus zwei anderen Körpern, die unter den Attributen `FirstOperand` bzw. `SecondOperand` gespeichert sind. Das besondere am `IfcCsgSolid`-Verfahren ist, dass die booleschen Operationen nicht nur auf primitive 3D-Körper (`IfcCsgPrimitive3D`), sondern auch auf jede beliebige 3D-Geometrie angewendet werden können. Das ermöglicht die Verknüpfung von 3D-Körpern, die infolge der Methoden `Boundary Representation`, `Extrusion` oder `CSG-Körper` entstanden sind. Die Klasse `IfcBooleanOperator` speichert einen der drei booleschen Operatoren, die in IFC als `UNION` (Vereinigung), `DIFFERENCE` (Differenz) und `INTERSECTION` (Schnitt) bezeichnet werden.

Möchte man für die Konstruktion von 3D-Körpern primitive 3D-Geometrien verwenden, dann kommt die Klasse `IfcCsgPrimitive3D` zum Einsatz. Folgende primitive 3D-Körper stehen in IFC zur Verfügung:

- Spat (IfcBlock)
- Rechteckige Pyramide (IfcRectangularPyramid)
- Kreiszyylinder (IfcRightCircularCylinder)
- Kreiskegel (IfcRightCircularCone)
- Kugel (IfcSphere)

Die Abbildung 40 veranschaulicht die Vorgehensweise der CSG-Methode im IFC-Model. Der erste Operand des `IfcBooleanResults` ist ein Quader, der infolge einer Extrusion entstanden ist. Dabei wurde ein Quadrat mit einer Seitenlänge von 3,0 als eine 2D-Fläche

gezeichnet und anschließend in Richtung der globalen z-Achse um 3,0 extrudiert. Der zweite Operand ist eine Kugel, die als ein primitiver Körper – IfcSphere (erbt von IfcCsgPrimitive3D) – mit Radius 1,5 dargestellt wird. Die Schnittmenge der beiden 3D-Objekte ergibt den rot-blauen-Körper.

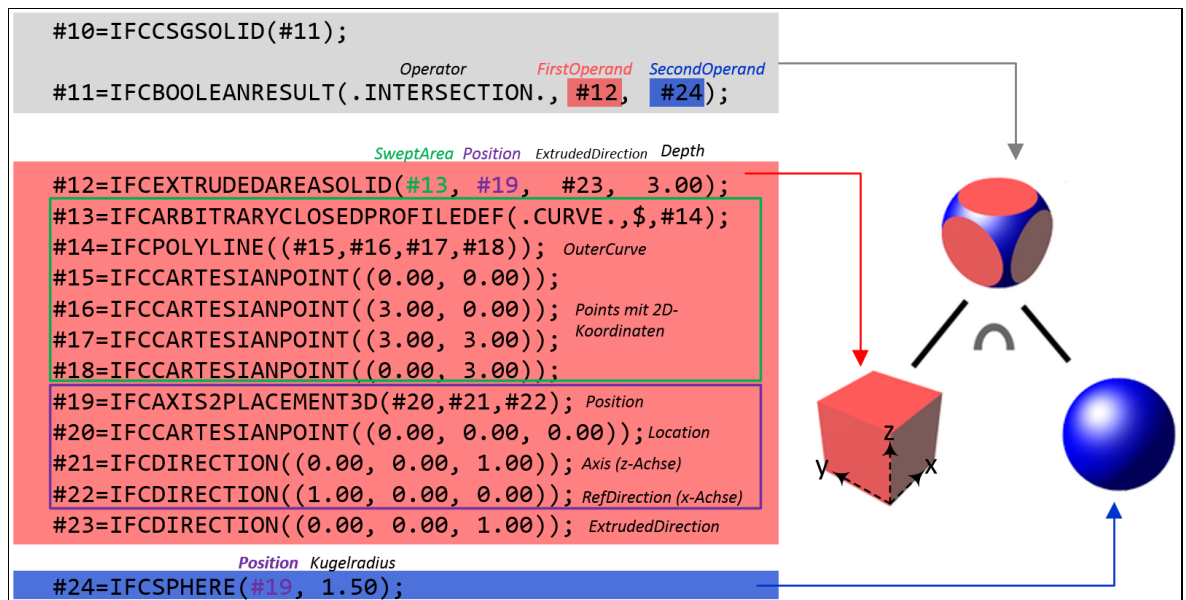


Abbildung 40: Anschauliches Beispiel für die Konstruktion von CSG- und Extrusions-Körpern

Das benötigte Datenvolumen, um komplexe 3D-Geometrien mit der CSG-Methode speichern zu können, ist im Vergleich zum entstehenden Datenvolumen aus Boundary Repräsentation wesentlich geringer. Das führt zum geringeren Aufwand bei der Entwicklung von IFC-Schnittstellen zu anderen Softwares.

Die Modellierung von komplexen, verketteten CSG-Körpern kann sich nachteilig auf die Parametrik dieser Körper auswirken. Hält sich jedoch die Komplexität und die Verkettung der booleschen Verknüpfungen im Rahmen, lässt sich auch die Parametrik in IFC mit einem brauchbaren Aufwand erzeugen und abspeichern. Anhand des Brücken-Unterbau-Beispiels in späteren Kapiteln wird dies näher erläutert.

Gegenüberstellung der geometrischen Darstellungs-Methoden

Vergleicht man die drei erwähnten Modellierungsmethoden bezüglich des benötigten Datenvolumens, so geht die Extrusion als die effizienteste Methode hervor. Jedoch wird es kaum möglich sein, komplexe Geometrien ausschließlich mit dem Extrusionsverfahren zu konstruieren. Meistens kommt es zur Verschmelzung zwischen den unterschiedlichen

geometrischen Modellierungsmethoden. Betrachtet man die Boundary Representation, so bietet diese Methode zwar einfache und leicht nachvollziehbare Ansätze, um die Körper-Geometrie darzustellen, jedoch steigt die Anzahl der benötigten Kanten und Punkte sehr rasch, wenn die Komplexität der zu modellierenden Körper erhöht wird. Somit steigt das Datenvolumen, was einen Nachteil im Bereich der Performance und der Übersichtlichkeit darstellt. Außerdem stößt die Randdarstellung mit ebenen Begrenzungsflächen an ihre Grenzen, wenn gerundete Bauteile modelliert werden müssen. Dafür bietet die BRep nur eine angenäherte Lösung an.

Die CSG-Methode in IFC wendet boolesche Operationen auf jegliche 3D-Körper an. Darunter fallen auch Boundary Representation und Extrusion. D.h, die Vor- und Nachteile der CSG-Methode sind mit anderen 3D-Modellierungs-Methoden sehr eng verknüpft. Um die Nachteile der Boundary Representation bezüglich des hohen Datenvolumens zu umgehen, entschied man sich im Rahmen dieser Masterarbeit, CSG und Extrusion zu verwenden. Die Brückenbauteile werden überwiegend mit diesen beiden Verfahren geometrisch konstruiert.

Zusätzlich zur CSG und Extrusion werden weitere Verfahren angewendet, um kleinere Bauteile mit einfacher Geometrie darzustellen. So kann z.B. ein Elastomerlager mit Hilfe der Klasse `IfcBoundingBox` erzeugt werden. Dabei wird ein einfacher Spat anhand der Position (Corner) und drei Dimensionen – Länge (XDim), Breite (YDim), Tiefe (ZDim) – konstruiert.

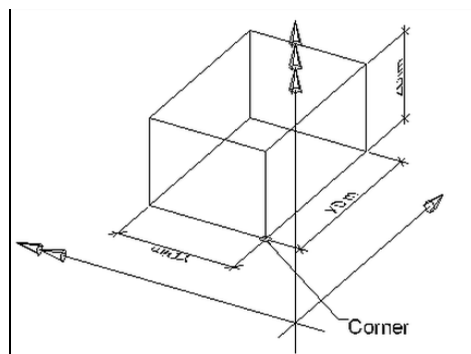


Abbildung 41: `IfcBoundingBox` [23]

Im nachfolgenden Kapitel wird erläutert, wie konkrete Bauteile des Brückenunterbaus geometrisch dargestellt werden können.

4.2.2 Darstellung von Widerlagern

An dieser Stelle soll darauf hingewiesen werden, dass in den folgenden Ausführungen unter der Bezeichnung Widerlager nur die massiven Widerlagerbauteile gemeint sind und nicht die Detailelemente wie z.B. Lager, Wartungsbereiche, Vogelschutz usw.

Die geometrische Form der Brückenwiderlager hängt von vielen Faktoren, wie Umgebungssituation, Bodenverhältnisse, Ästhetik usw., ab (Kapitel 3.2, s. 24). Von der einfachen massiven Widerlagerwand bis zum kastenförmigen Widerlager mit seitlichen Flügelwänden kann die Geometrie sehr variabel sein. Während eine alleinstehende Widerlagerwand eher einen seltenen Fall darstellt, kommt das kastenförmige Widerlager häufiger zum Einsatz [15]. Deshalb richtet sich die IFC-Modellierung im vorliegenden Kapitel nach dieser Widerlagerform.

Kastenförmiges Widerlager besitzt laut den Richtzeichnungen der Bundesanstalt für Straßenwesen [17] eine Regelform und eine Variantenform (Kapitel 3.2, s. 24). Der Unterschied zwischen diesen beiden Ausführungen besteht in Hinsicht auf die Form ihrer seitlichen Flügelwände. Bei der Regelausführung verläuft die nach unten gerichtete Abschrägung der Flügelwände unter einem konstanten Winkel von 60° (Abbildung 42-links).

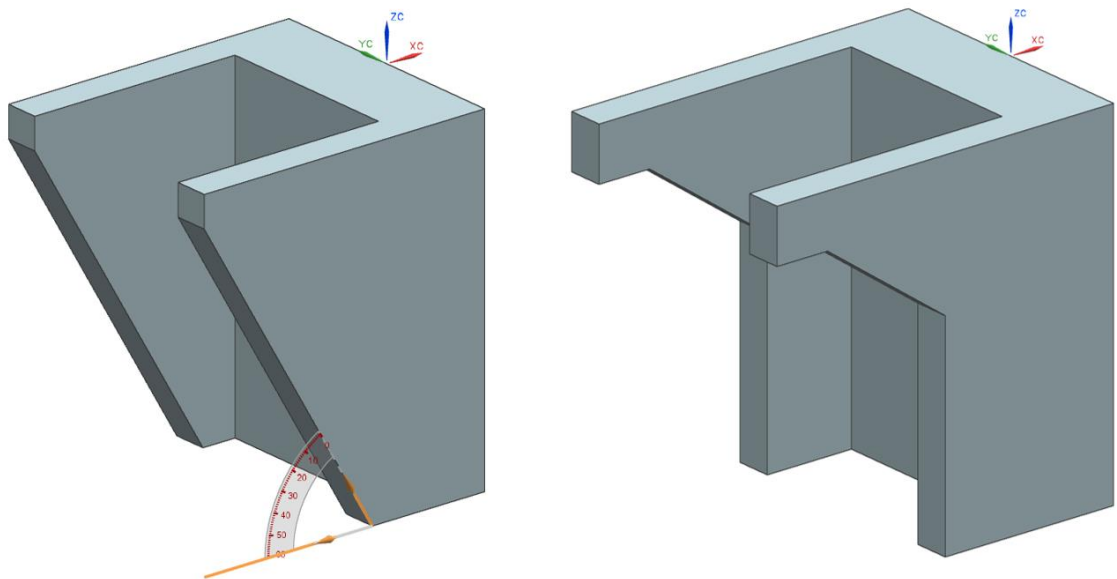


Abbildung 42: Regelform (links) und Variantenform (rechts) eines kastenförmigen Widerlagers [24]

Sind größere Widerlagerhöhen notwendig, wird die Regelausführung unwirtschaftlich, da ein Teil der Flügelwandflächen, statisch gesehen, nicht tragend und somit überflüssig wird [16]. Aus diesem Grund konstruiert man ein Widerlager mit einer nur teilweise abschrägenden Flügelwand und auskragenden Flügelarmen – die sogenannte Variantenform (Abbildung 42-rechts). Die Neigung der Flügelwände ist in diesem Fall identisch mit der Böschungsneigung. In den nachfolgenden Erläuterungen wird die Variante als Vorzeigebispiel für die Beschreibung in IFC hergenommen. Die Regelausführung kann daraus analog abgeleitet werden. Zur Speicherung des kastenförmigen Widerlagers im IFC-Model, wurden zwei Methoden ausgearbeitet. Das Widerlager kann als ein komplettes Bauteil oder aus mehreren wandartigen Elementen konstruiert werden.

Darstellung des Widerlagers als komplettes Bauteil

Extrusion (`IfcExtrudedAreaSolid`) und CSG (`IfcCsgSolid` / `IfcBooleanResult`) sind die beiden Darstellungsverfahren die zum Einsatz kommen werden. Zunächst wird mittels der Klasse `IfcExtrudedAreaSolid` ein Basiskörper des Widerlagers konstruiert. Dieser enthält durchgängige Flügel, die anschließend mittels einer booleschen Operation ihre richtige Form erhalten. Die Abbildung 43 veranschaulicht den Prozess der Extrusion.

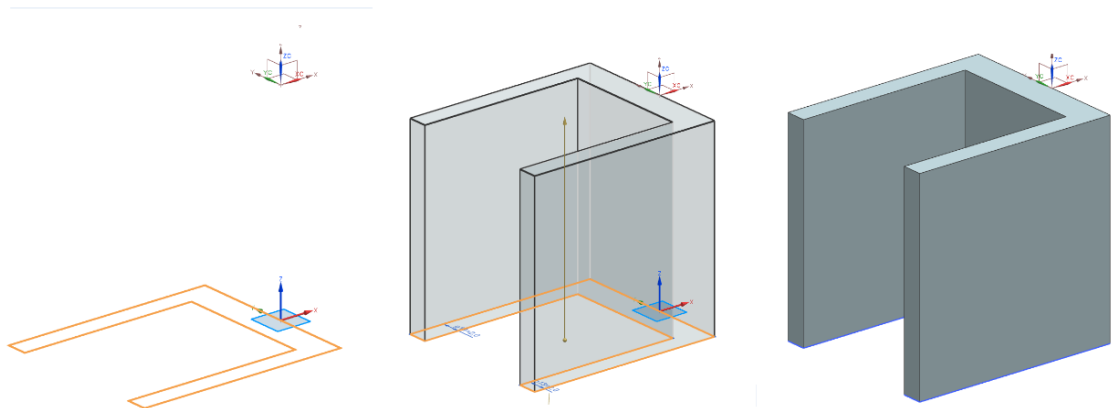


Abbildung 43: Konstruktion des Basiskörpers mit Extrusion [24]

Der Ursprung des globalen Koordinatensystems befindet sich an der oberen Kante des Widerlagers. Der Grundriss des Basiskörpers des Widerlagers wird auf einer 2D-Ebene gezeichnet (orange). In IFC entspricht die Grundrisszeichnung dem Attribut `SweptArea`, das der Klasse `IfcExtrudedAreaSolid` angehört. Die an der orangenen Fläche aufgezeichneten Vektoren repräsentieren das lokale Koordinatensystem der `SweptArea`.

Die Richtungen der jeweiligen Vektoren und der Ursprung des Koordinatensystems werden in `IfcAxis2Placement3D` gespeichert. Der Ursprung des lokalen Systems befindet sich global an der Oberkante des späteren Widerlagerfundaments und wird in IFC durch das Attribut `Location` festgelegt. Anschließend wird die Grundrisszeichnung in die globale positive z-Richtung (nach oben) extrudiert. Die Extrusionstiefe (`Depth`) entspricht der Höhe des Widerlagers. Wie die Konstruktion des extrudierten Basiskörpers im IFC-Model umgesetzt wird, zeigt die Abbildung 44.

```
#39=IFCCARTESIANPOINT((0.0000,0.0000));
#40=IFCCARTESIANPOINT((0.0000,7.500));
#41=IFCCARTESIANPOINT((-20.00,7.500));
#42=IFCCARTESIANPOINT((-20.00,6.405));
#43=IFCCARTESIANPOINT((-5.000,6.405));
#44=IFCCARTESIANPOINT((-5.000,-6.405));
#45=IFCCARTESIANPOINT((-20.00,-6.405));
#46=IFCCARTESIANPOINT((-20.00,-7.500));
#47=IFCCARTESIANPOINT((0.0000,-7.500));
#48=IFCPOLYLINE((#39,#40,#41,#42,#43,#44,#45,#46,#47));
#49=IFCARBITRARYCLOSEDPROFILEDEF(.CURVE.,$,#48);
#50=IFCCARTESIANPOINT((5.000,0.0000,-20.00));
#51=IFCDIRECTION((0.0000,0.0000,1.000));
#52=IFCDIRECTION((1.000,0.0000,0.0000));
#53=IFCAXIS2PLACEMENT3D(#50,#51,#52);
#54=IFCDIRECTION((0.0000,0.0000,1.000));
#55=IFCEXTRUDEDAREASOLID(#49,#53,#54,20.00);
```

Abbildung 44: Basiskörper des Widerlagers in IFC [24]

Der konstruierte Basiskörper besitzt noch keine Flügelwände mit Kragflügeln, sondern durchgängige Wände. Um die erwünschte Form der seitlichen Widerlagerwände zu erhalten, wird mit Hilfe einer CSG-Verknüpfung der überschüssige Körper abgezogen. Zunächst wird der Abzugskörper mit Hilfe der Extrusion erzeugt. Die `SweptArea` ist in diesem Fall eine in der x-z-Ebene gezeichnete 2D-Fläche. Der Ursprung des lokalen Koordinatensystems für den Abzugskörper befindet sich an der unteren rechten Ecke der Flügelwand (Abbildung 45-1). Die Extrusionsrichtung ist die positive y-Achse und die Extrusionstiefe entspricht der Breite des Widerlagers.

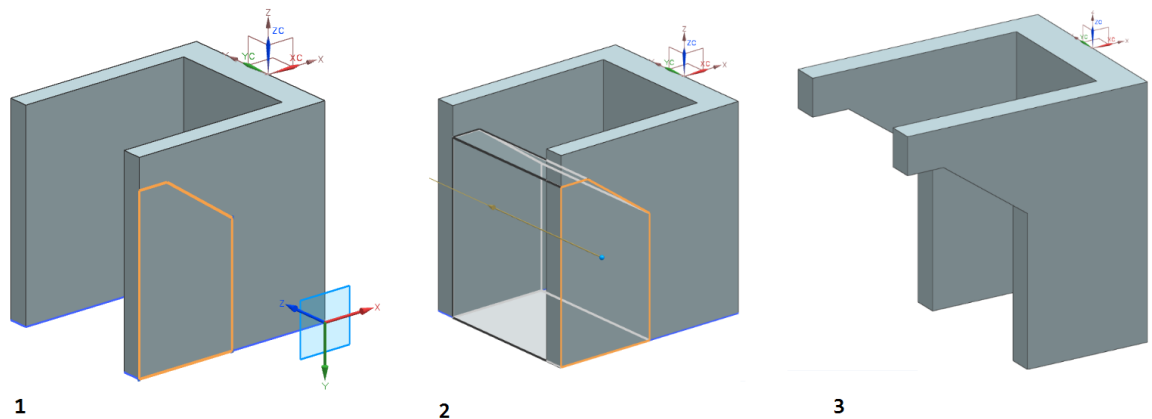


Abbildung 45: Konstruktion der Flügelwände durch Abziehen eines Hilfskörpers [24]

Schließlich wird der Abzugskörper vom Basiskörper abgezogen. In IFC ist dafür die Klasse `IfcBooleanResult` zuständig. Dabei wird der Basiskörper als `FirstOperand` und der Abzugskörper als `SecondOperand` bezeichnet. Der boolesche Operator ist die Differenz (`DIFFERENCE`). Als Resultat entsteht der in der Abbildung 45-3 dargestellte Widerlager mit Kragflügeln.

Darstellung des Widerlagers aus einzelnen Bauteilen

Die nachfolgend vorgestellte Methode konstruiert das Widerlager aus mehreren einzelnen Bauteilen. Dabei spaltet man das Widerlager in drei Hauptteile: Stirnwand und die beiden Flügelwände. Alle drei Bauteile werden mit der Extrusionsmethode dargestellt. Zunächst wird die massive Widerlagerstirnwand mit einem rechteckigen Querschnitt konstruiert. Dabei zeichnet man den Grundriss des Rechtecks auf eine 2D-Ebene und extrudiert sie um die Höhe des Widerlagers in die Richtung der globalen negativen z-Achse (nach unten) (Abbildung 46).

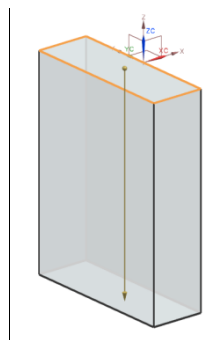


Abbildung 46: Extrusion der massiven Widerlagerstirnwand [24]

Anschließend konstruiert man die beiden Flügelwände. Dabei extrudiert man jeweils den Querschnitt der Wand in die negative bzw. positive y-Richtung. Die Extrusionstiefe in diesem Fall entspricht der Dicke der Flügelwand. Anschließend erfolgt eine Vereinigung der drei Bauteile mit Hilfe von `IfcBooleanResult` und dem Operator `UNION` (Schnitt) (Abbildung 47).

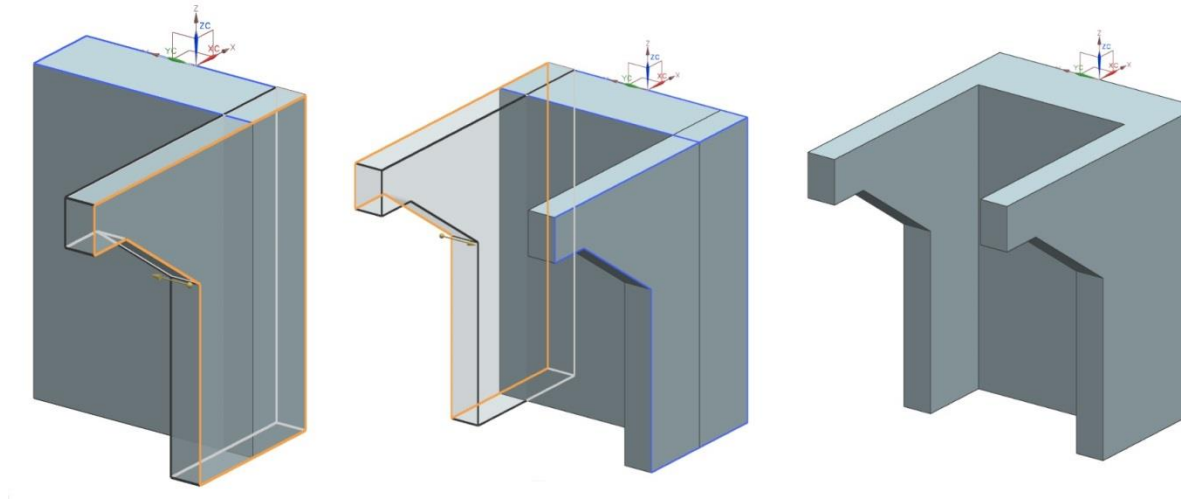


Abbildung 47: Flügelwände des Widerlagers und die Vereinigung aller Bauteile [24]

Die oben beschriebene Form des Widerlagers ist nur für eine bestimmte Umgebungssituation relevant. Verläuft die über- und unterführte Straße nicht senkrecht zueinander, ändert sich der Kreuzungswinkel und das Brückenbauwerk wird schiefwinklig. Folglich muss die Geometrie des Widerlagers an den Verlauf der Straßen angepasst werden. Aufgrund der separaten Konstruktion der einzelnen Widerlagerteile, lässt sich die Anpassung an die veränderte Situation übersichtlicher realisieren als bei einer ganzheitlichen Konstruktion. Weitere Aspekte um das Thema der veränderlichen Randbedingungen und welche Auswirkung diese auf die Geometrie des Widerlagers haben, werden im Kapitel 6 genauer erläutert, wenn es darum geht, parametrische Abhängigkeiten zwischen den Unterbauelementen einer Brücke im IFC-Model abzuspeichern.

4.2.3 Abspeicherung der Pfeiler und Stützen

Brücken-Pfeiler sind massive wandartige Bauteile, die meistens einen konstanten Querschnitt über die Höhe aufweisen. Auch die Stützen besitzen in der Regel einen konstanten runden Durchschnitt. In solchen Fällen bietet sich die Extrusionsmethode an. Ein rechteckiger Pfeilerquerschnitt wird auf die gleiche Weise wie die Widerlagerwand aus vorigem Kapitel modelliert (Abbildung 48).

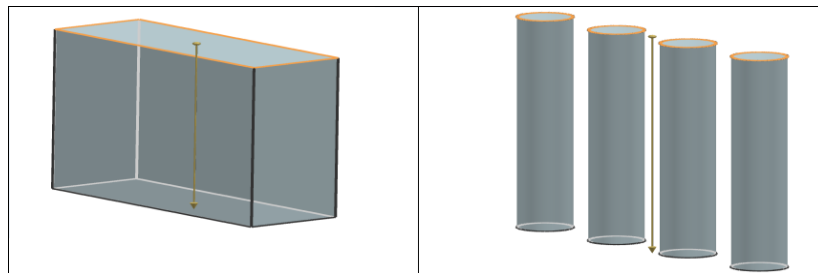


Abbildung 48: Pfeiler und Stützen mit konstanten Querschnitten [24]

Um optimale Abtragung der Lasten zu ermöglichen, kommt es häufig zum Einsatz von Stützen und Pfeilern mit veränderlichem Querschnitt oder zur Ausbildung eines Hammerkopfes (s. Kapitel 3.3). `IfcReferencedSectionedSpine` eignet sich optimal dafür, solche Pfeiler-Konstruktionen zu beschreiben. Bereits im Kapitel 2.3 (s. 20) wurde erläutert wie eine Überbauplatte mit veränderlichem Querschnitt mit Hilfe dieser IFC-Klasse erzeugt werden kann. Die Konstruktion der Pfeiler verläuft analog. Die Abbildung 49 veranschaulicht anhand eines Beispiels, wie ein Pfeiler mit veränderlichem Querschnitt in IFC abgespeichert werden kann und welcher Zusammenhang zu einzelnen Pfeilerelementen besteht.

Beschreibung des Brückenunterbaus in IFC

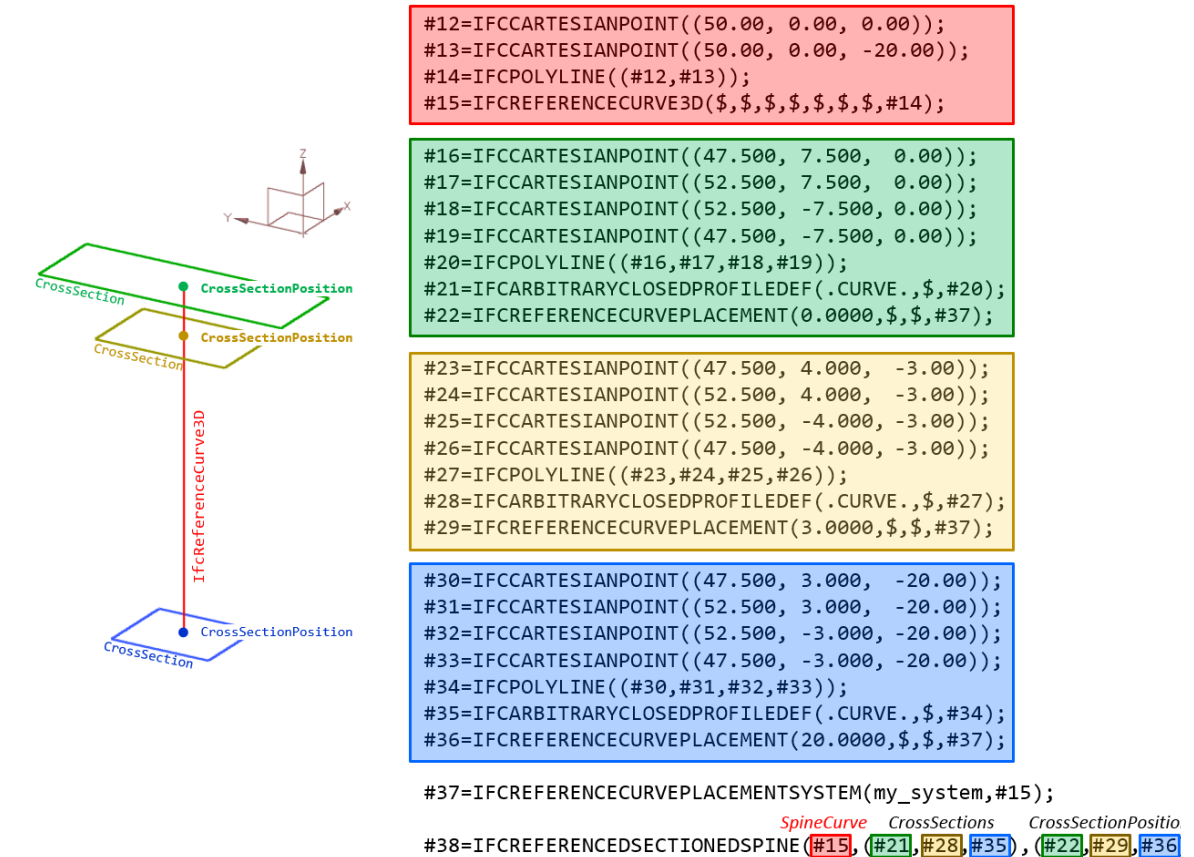


Abbildung 49: Darstellung eines Pfeilers mit veränderlichem Querschnitt [24]

Es wird angenommen dass die x-Koordinate der Pfeilerposition im globalen Koordinatensystem 50,00 beträgt. Darauf sind die restlichen kartesischen Koordinaten der Punkte basiert. Das erste Attribut der `IfcReferencedSectionedSpine` ist die `SpineCurve`, die von der Klasse `IfcReferencedCurve3D` repräsentiert wird. Diese verläuft vom oberen bis zum unteren Rand des Pfeilers und gilt als die Führungslinie für die variable Extrusion [3]. Das zweite Attribut der `IfcReferencedSectionedSpine` ist eine Liste aus Pfeilerquerschnitten (Attribut: `CrossSections`), geordnet nach dessen Position von oben nach unten, da diese Richtung dem Verlauf der Referenzlinie entspricht. Die Breite der Querschnitte bleibt konstant und beträgt 5,00. Die Längen der Querschnitte verkleinern sich von oben nach unten. Die Positionen der jeweiligen Querschnitte (Attribut: `CrossSectionPositions`) sind ebenfalls in einer Liste innerhalb des Objektes `IfcReferencedSectionedSpine` festgelegt und von oben nach unten geordnet. Somit muss die Anzahl der Querschnitte gleich der Anzahl der Querschnitt-Positionen sein. Anhand dieser Informationen kann ein Pfeiler visualisiert werden, indem seine Form zwischen den gegebenen Querschnitten mit Hilfe der variablen Extrusion interpoliert wird.

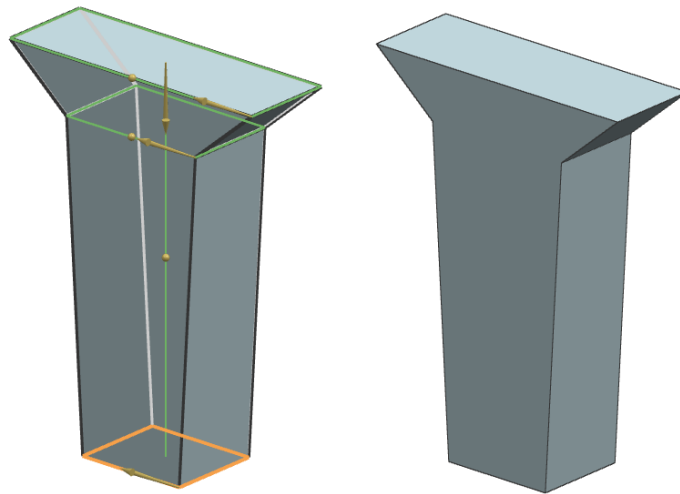


Abbildung 50: variable Extrusion (links) und der fertige Pfeiler (rechts) [24]

Ein Sonderfall der Brückenunterstützung ist der Bogen. Bogenbrücken kommen meist in tiefen Tälern vor, wo die hohen massiven Pfeiler unwirtschaftlich werden [16]. Theoretisch müsste sich die Bogenform ebenfalls mit `IfcReferencedSectionedSpine` beschreiben lassen. In den bekannten IFC-Visualisierungsanwendungen ist es jedoch nicht gelungen eine `IfcReferencedSectionedSpine`, die auf einer gekrümmten Referenzlinie basiert, darzustellen. An dieser Stelle sollte auf die ähnliche Darstellungsmethode verwiesen werden, die im Modellierungsprogramm Siemens NX ihre Anwendung findet.

In Siemens NX lassen sich Querschnitte entlang beliebiger Kurven, die als Leitlinien bezeichnet werden, mit Hilfe der Funktion `Swept` extrudieren. Dabei können diverse Einstellung getroffen werden, die das Verhalten der Querschnitte beeinflussen. Darunter existiert die Einstellung „Orientierung der Querschnitte“, die mehrere Auswahlmöglichkeiten bietet. Standardmäßig ist die Option „Fixed“ eingestellt. Dabei behält der Querschnitt seine Ausgangsorientierung entlang des gesamten Extrusionswegs. Die daraus resultierende Form ist verfälscht, da sich der Bogen zum Scheitelpunkt hin verjüngt (Abbildung 51).

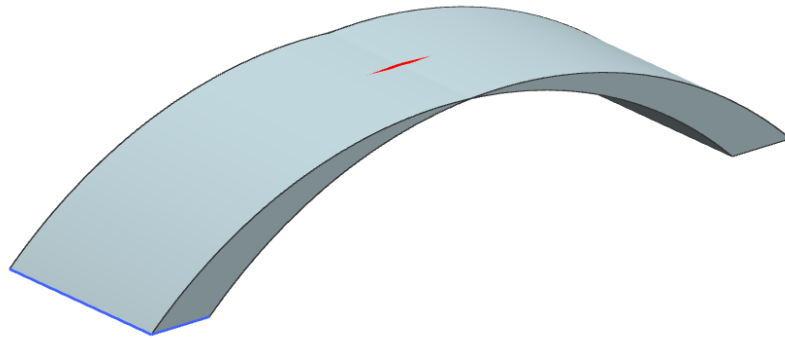


Abbildung 51: Extrusion eines Rechtecksquerschnitts entlang eines Kreisbogens mit fester Orientierung (Siemens NX) [24]

Die richtige Form erhält man, indem die Orientierung der Querschnittsnormalen und die Tangente des Kreisbogens an der jeweiligen Querschnittsposition gleichgesetzt werden. Um dies zu erzwingen, wird in NX bei der Einstellung der Orientierung die Option „Vector Direction“ gewählt. In IFC ist es jedoch aktuell nicht möglich, solch eine Angabe zu speichern. Um einen Bogentragwerk mit Hilfe der `IfcReferencedSectionedSpine` zu beschreiben, könnte man jedoch ähnliche Vorgehensweise wie in Siemens NX anwenden. Dafür erhält die `IfcReferencedSectionedSpine` ein weiteres Attribut mit der Bezeichnung `CrossSectionOrientation`, das die entsprechende Auskunft über die Orientierung festlegt. Der Bogenkörper wird schließlich mit der bereits bekannten Vorgehensweise konstruiert. Die Referenzlinie entspricht dabei einem Kreisbogen. Am Anfang und am Ende der Referenzlinie wird jeweils ein rechteckiger Querschnitt gezeichnet. Die Positionen der Querschnitte werden im Bogenmaß angegeben. Anhand dieser Angaben formt der IFC-Viewer den Bogenkörper durch die Interpolation zwischen den beiden Querschnitten. Die Abbildung 52 zeigt eine mögliche Visualisierung des Bogens mit Hilfe der `IfcReferencedSectionedSpine`.

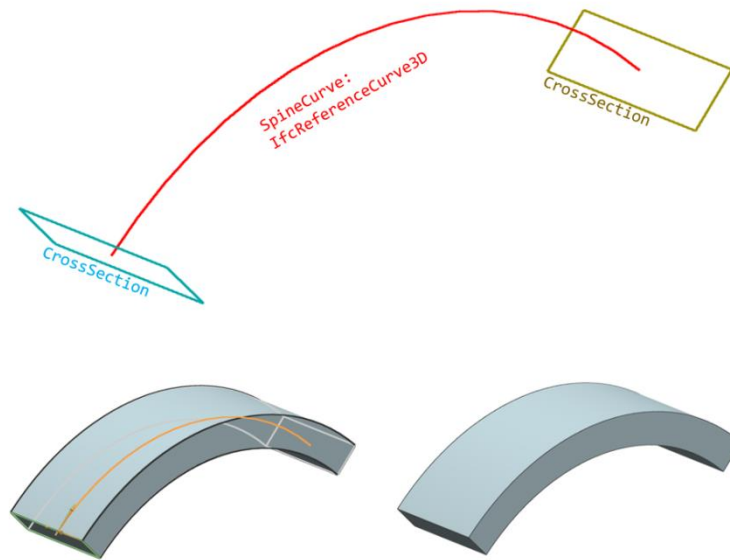


Abbildung 52: Konstruktion des Bogenquerschnitts mit IfcReferencedSectionedSpine [24]

4.2.4 Abspeicherung der Gründung

Bei der Gründung der Widerlager und Pfeiler unterscheidet man zwischen Flach- und Tiefgründung (siehe Kapitel 3.4, s. 31). Flachgründungen sind massive Bodenplatten, die in der Regel konstante Dicke über die gesamte Fläche aufweisen [15]. Die Grundrissform der Fundamente lässt sich aus der entsprechenden Unterseite des Widerlagers oder Pfeilers ableiten. Die einfachste Praktik ist die Anwendung der Klasse `IfcOffsetCurve2D`. Durch ihre Attribute `BasisCurve` und `Distance` wird die Ausgangskurve bzw. deren Abstand zur neuen Kurve erfasst. Um `IfcOffsetCurve2D` anwenden zu können, benötigt man die Querschnittsskizze der Unterseite des Pfeilers bzw. des Widerlagers. Bei rechteckigen Pfeilern oder Widerlagerwänden lässt sich diese Methode ohne Probleme einsetzen. Dabei gilt die `SweptArea`, die im Laufe der Darstellung des Pfeilers mit der Klasse `IfcExtrudedAreaSolid` definiert wurde, als Ausgangskurve für die Offsetkurve des Fundamentgrundrisses (Abbildung 53-links).

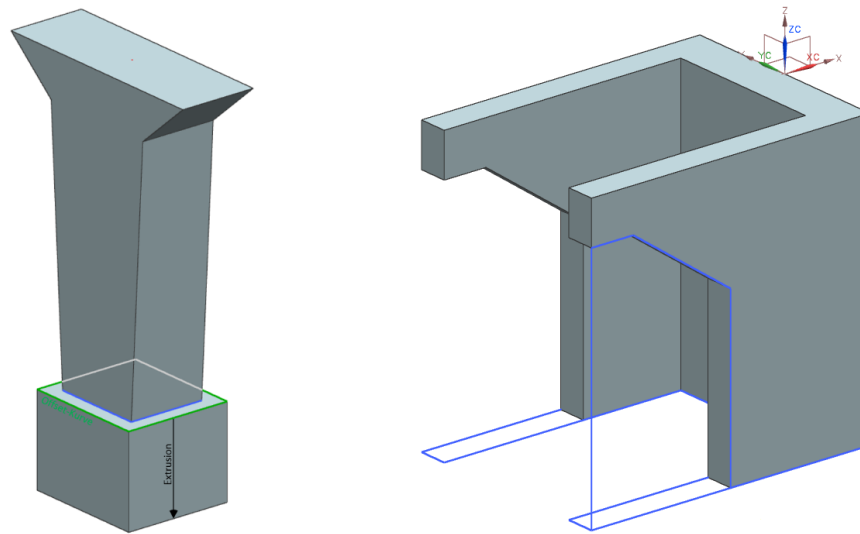


Abbildung 53: Fundamentmodellierung für ein Pfeiler (links) und ein kastenförmiges Widerlager (rechts) [24]

Bei kastenförmigen Widerlagern ist die Konstruktion des Fundamentgrundrisses über eine Offsetkurve in IFC ohne weiteres nicht möglich. Im Kapitel 4.2.2 (s.44) wurden Methoden zur geometrischen Konstruktion von Widerlagern behandelt. Die erste Variante sieht ein Abziehen eines Hilfskörpers vom Basiskörper vor. Dabei besitzt der gezeichnete Grundriss die Form der Unterseite des Basiskörpers. Die Flügelwände des eigentlichen Widerlagers werden jedoch kürzer, nachdem der Abzugskörper vom Basiskörper subtrahiert wurde. D.h. die Grundrisszeichnung des Fundamentes lässt sich vom Widerlager nicht direkt ableiten (Abbildung 53-rechts). Bei der zweiten Variante werden die einzelnen Widerlagerteile getrennt voneinander modelliert. Die Widerlagerstirnwand wird in die z-Richtung extrudiert und besitzt somit eine rechteckige Unterseite, die für das Fundament als Ausgangszeichnung benutzt werden kann. Bei den Flügelwänden erfolgt die Extrusion dagegen in die y-Richtung, was die Beziehung zur Geometrie des Fundaments wieder nichtig macht.

Eine mögliche Lösung wäre die Erfassung der Fundamentgeometrie unabhängig von der Widerlagerform. Da die Flachgründungen in der Regel keine komplexen Formen aufweisen, verursacht dies keinen großen Aufwand. Außerdem besteht die Möglichkeit, die Koordinatenpunkte der Gründung über parametrische Abhängigkeiten mit den Widerlagerabmessungen zu verknüpfen. Dies hat eine automatische Generierung der

Fundamentgeometrie, aufgrund der aufgestellten parametrischen Regeln, zur Folge. Die Parametrik innerhalb der Brückenunterbauten wird ausführlich im Kapitel 6 behandelt.

Ist die Vorkommnis der tragfähigen Bodenschichten erst in größeren Tiefen anzutreffen, müssen die Methoden der Tiefgründung angewendet werden [16]. Im Brückenbau ist die Konstruktion von Pfahlgründungen verbreitet (vergl. Kapitel 3.4). Die gängigste Form ist ein runder Bohrpfehl mit konstantem Durchmesser von $\varnothing.3$ bis 3.0 m. Analog zu den runden Stützen, die in den vorigen Erläuterungen dieses Kapitels behandelt wurden lassen sich auch die Bohrpfähle mit Hilfe der Extrusion konstruieren. Über die Extrusionsrichtung kann die Neigung der Bohrpfähle definiert werden. Die Abbildung 54 zeigt eine Systemskizze mit jeweils einem geraden und einem geneigten Bohrpfehl.

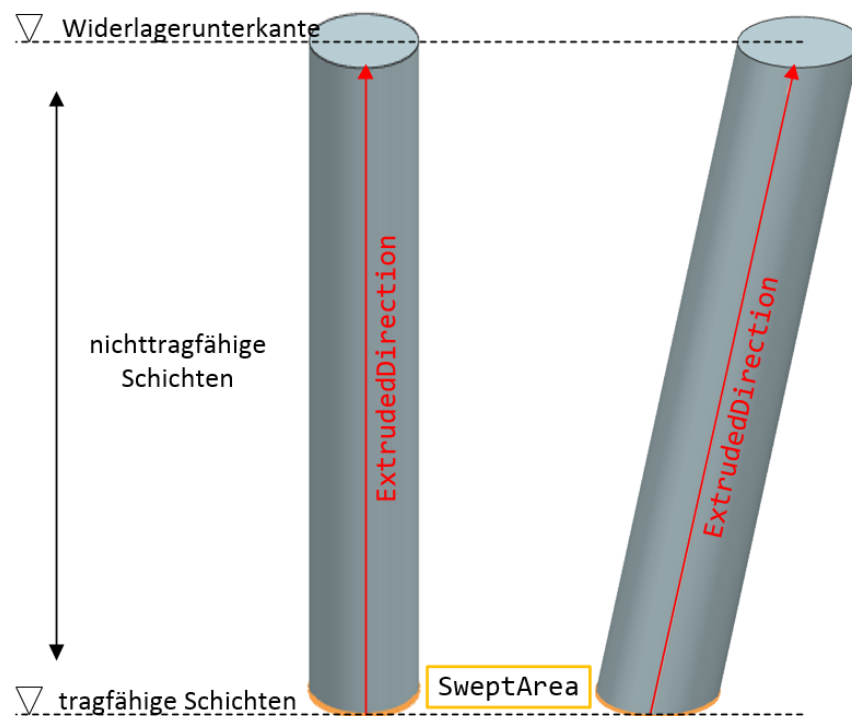


Abbildung 54: Extrusionsrichtungen bestimmen den Neigungswinkel der Bohrpfähle [24]

5 IFC-Import nach Siemens NX

5.1 Allgemeines zu Siemens NX

Bei NX handelt es sich um ein parametrisierbares 3D-Darstellungs-Programm „für Konstruktion, Styling, Zeichnungserstellung, Simulation und Fertigung sowie weitere Anwendungsgebiete“ [3]. 3D-Modellierung unter Nutzung von Grundkörpern, Formelementen, Skizzen, booleschen Operationen zählen zu den Standardmethoden.

Ein NX-Addin – das IfcBridgeTool, entwickelt vom Lehrstuhl für Computergestützte Modellierung und Simulation an der TU München, ermöglicht den Import von IfcBridge-Models ins Programm Siemens NX. Die Erweiterungen der NX-Schnittstelle, die im Rahmen der vorliegenden Arbeit stattgefunden haben, ermöglichen den Import von einer IFC-Brücke einschließlich der Unterbauelemente.

5.2 Erweiterung des Programms IfcBridgeTool

IfcBridgeTool importiert IfcBridge-Dateien in parametrische Modellierungssoftware Siemens NX und ermöglicht die Weiterverarbeitung des Brückenmodells. Außerdem kann das Programm IfcBridgeTool parametrische Beziehungen zwischen bestimmten Bauteilen der Brücke erzeugen und diese ins NX-Brücken-Modell aufnehmen. IfcBridgeTool basiert auf dem IfcBridge-Schema und war somit nur für den Import von Brückenüberbauten anwendbar. Die Abbildung 56 zeigt eine ins NX importierte Überbauplatte mit veränderlichem Querschnitt.

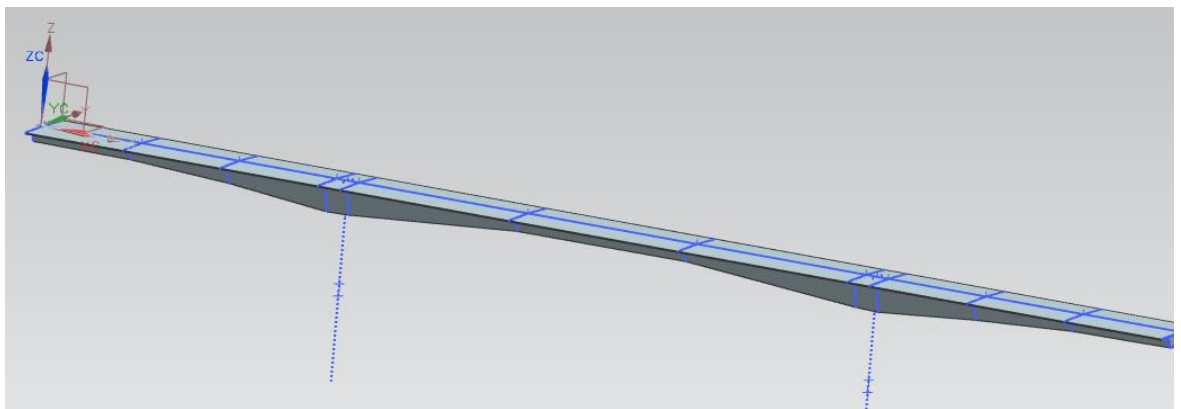


Abbildung 55: IFC-Hohlkasten-Brücke importiert ins Siemens NX [13]

Im Rahmen der vorliegenden Masterarbeit erweiterte man die Schnittstelle IfcBridgeTool um die Möglichkeiten auch Brückenunterbauten ins Siemens NX zu importieren. Im folgenden Kapitel wird aufgezeigt, welche Prinzipien und Strategien verwendet wurden, um den Import zu realisieren.

Die Abbildung 56 zeigt die Benutzeroberfläche des mit der Programmiersprache C# entwickelten IfcBridgeTool's. Über die Schaltflächen „Browse Schema“ und „Browse File“ wird das IFC-Schema (EXPRESS-Format) bzw. die zu importierende IFC-Datei (STEP-Format) aus einem lokalen Verzeichnis ausgewählt.

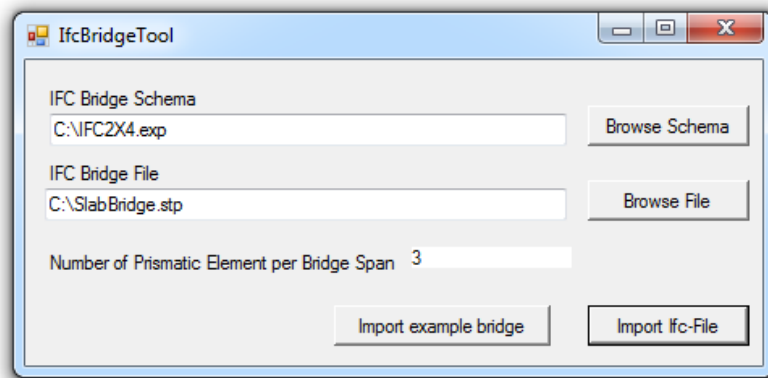


Abbildung 56: Benutzeroberfläche des Programms IfcBridgeTool [25]

Die Betätigung der Schaltfläche „Import Ifc File“ löst innerhalb der Schnittstelle einen Prozess aus, der sich in 3 Hauptschritte aufteilen lässt:

- 1) Lesen des IFC-Brückenmodells aus der angegebenen STEP-Datei.
- 2) Laden der einzelnen IFC-Objekte
- 3) Darstellung des Brückenmodells in NX anhand der geladenen Informationen

Das Lesen der STEP-Datei übernimmt die externe dynamische Bibliothek IfcEngine, die seit 2001 von der Firma RDF entwickelt wird. Die meisten Programme, die auf Visualisierung, Überprüfung oder Umsetzung der IFC-Dateien spezialisiert sind, sind mit IFC-Engine ausgestattet. Zur Steuerung der IFC-Daten benutzt die IfcEngine eine abstrakte Anwendungsschnittstelle **Standard Data Access Interface (SDAI)**. Digitale Daten, die nach dem EXPRESS-Modell aufgebaut und gespeichert sind, können mit Hilfe der Methoden, die SDAI zur Verfügung stellt, manipuliert werden [26]. Dabei kann die IFC-STEP-Datei

geladen, gespeichert und wieder geschlossen werden. Die einzelnen Entitäten können mit Hilfe von SDAI abgerufen und ihre Attribute abgefragt werden.

IfcBridgeTool geht über jede Entität der IFC-Datei und speichert diese mit Hilfe der IfcEngine in einen Zwischenspeicher. Anschließend wird das gesamte IFC-Modell in objektorientierter Form gespeichert. Einzelne C#-Klassen entsprechen jeweils der Entitäten aus der IFC-Datei und ihre Attribute werden in den privaten Variablen dieser C#-Klassen gespeichert. Im nächsten Schritt wird jedes einzelne Objekt der oben erwähnten Klassen instanziiert. Somit erhält man den Zugriff auf die Attribute der vorher geladenen IFC-Klassen. Der Import der Brückendaten ins Siemens NX kann beginnen. Im dritten und letzten Schritt wird die Brückengeometrie anhand der Informationen, die die jeweiligen Instanzen bereitstellen ins NX geladen. Dafür bietet NX ein Framework namens NX Open an, das Methoden enthält, mit deren Hilfe geometrische Körper innerhalb des Programms Siemens NX erstellt und geändert werden können. Die Abbildung 57 zeigt eine Brücke mit zwei Widerlagern und zwei Pfeilern, die in IFC beschrieben war und über die Schnittstelle IfcBridgeTool nach Siemens NX importiert wurde.

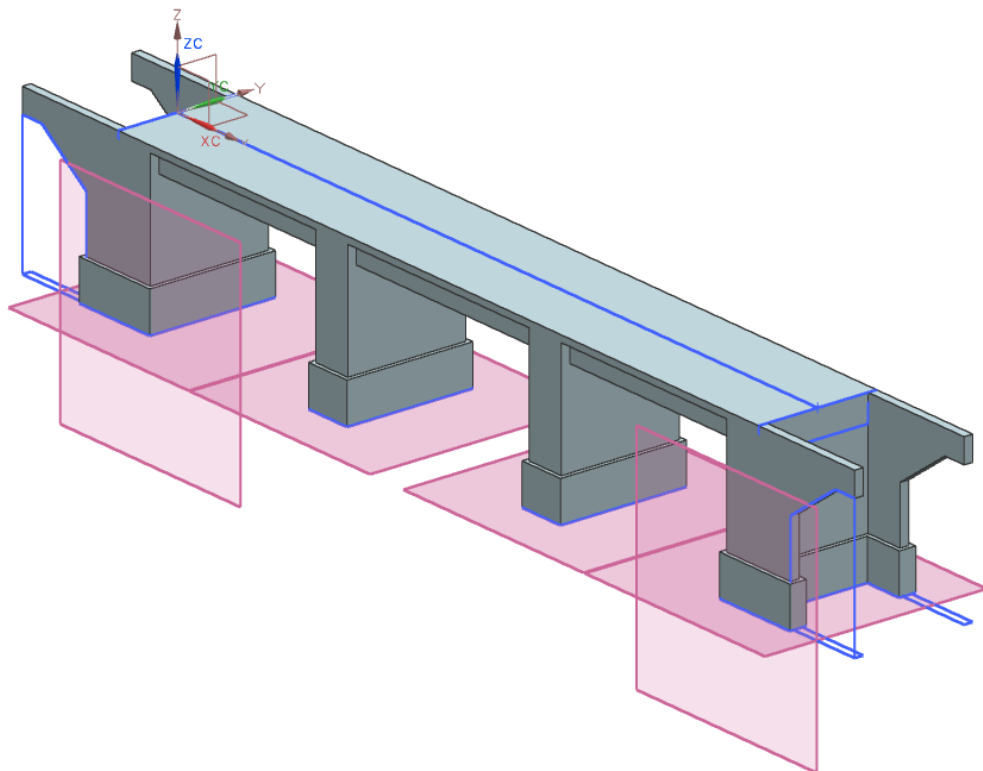


Abbildung 57: ins Siemens NX importierte Plattenbrücke mit Unterbauelemente [25] [24]

Zusätzlich zur Geometrie aus der IFC-Datei erzeugt IfcBridgeTool parametrische Abhängigkeiten innerhalb des importierten NX-Modells. Eine besondere Beachtung soll der Tatsache geschenkt werden, dass die parametrischen Regeln des erzeugten NX-Modells ausschließlich innerhalb der Schnittstelle IfcBridgeTool aufgestellt wurden. Somit leistet die eigentliche Datenquelle, also das IFC-Modell, keinen Beitrag zur Parametrik des Brückenmodells. In den folgenden Erläuterungen wird der Import von IFC-Dateien ins Siemens NX mit parametrischen Abhängigkeiten näher beleuchtet und analysiert.

Import der Parametrik ins NX

Die Planungsgesellschaft Obermeyer nutzt Siemens NX als Werkzeug zur Modellierung von parametrischen Brücken. Aufgrund relativ hoher Anzahl der Planungsprojekte im Brückenbau, bringt die Firma Obermeyer große, praxisbezogene Erfahrung im Bereich der Modellierung der Brückenbauwerke. Die Vorgehensweisen und Methoden, die die Firma Obermeyer im Zuge der Brücken-Modellierung mit Siemens NX verwendet, wurden im Rahmen der vorliegenden Arbeit als Grundlage zur parametrischen Darstellung von Brückenunterbauten genutzt.

Die große Stärke des Programms NX beläuft sich auf die parametrische Darstellung der Elemente. Für den Brückenüberbau wurden parametrische Regeln ausgearbeitet. Diese werden innerhalb des IfcBridgeTool beim Import auf das NX-Brücken-Modell übertragen. D. h. alle Brücken, die über die Schnittstelle IfcBridgeTool ins NX importiert werden, erhalten dieselben parametrischen Abhängigkeiten (siehe Abbildung 58).

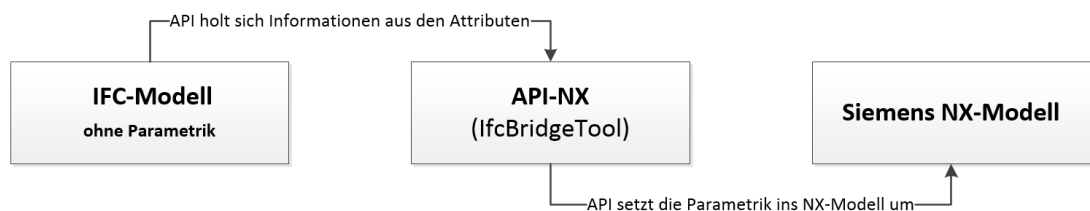


Abbildung 58: Imports eines IFC-Modells mit der Umsetzung der Parametrik innerhalb der Schnittstelle [24]

Die parametrischen Regeln können jedoch bei zwei unterschiedlichen Brücken voneinander abweichen. Am Beispiel der Widerlagerbreiten lässt sich dieser Unterschied erklären. Es sollen zwei Brücken mit Hilfe des IfcBridgeTool's ins NX importiert werden. Das Widerlager der Brücke 1 besitzt eine parametrische Abhängigkeit zur Überbauplatte,

und zwar sollen diese beiden Elemente identische Breiten besitzen. Diese Parametrik wird innerhalb des IfcBridgeTools festgelegt und ins NX-Modell importiert. Dagegen besitzt das Widerlager der Brücke 2, aufgrund bestimmter Umgebungsrandbedingungen eine andere Definition der Breite. Um den Import der Brücke 2 zu realisieren, muss entweder die Schnittstelle (IfcBridgeTool) angepasst werden oder eine neue Schnittstelle entwickelt werden, die speziell für die Brücke 2 den Import ermöglicht. Angesichts der Tatsache, dass jedes Brückenbauwerk ein einzigartiges Objekt darstellt, können die parametrischen Regeln unterschiedlicher Brücken sehr variabel sein. Die Entwicklung oder die Anpassung der Programmierschnittstellen für jede einzelne Brücke würde deswegen zu einem enormen Programmieraufwand führen.

Ein denkbarer Ansatz, um dieser Problematik auszuweichen, ist die Integration der parametrischen Regeln direkt im IFC-Modell. Dabei liefert das IFC-Modell die Informationen über die konkreten parametrischen Abhängigkeiten, die nur diese spezielle Brücke betreffen, an die NX-Schnittstelle. Infolgedessen ist die Schnittstelle befähigt, Brücken mit beliebig definierten parametrischen Regeln ins NX zu importieren.

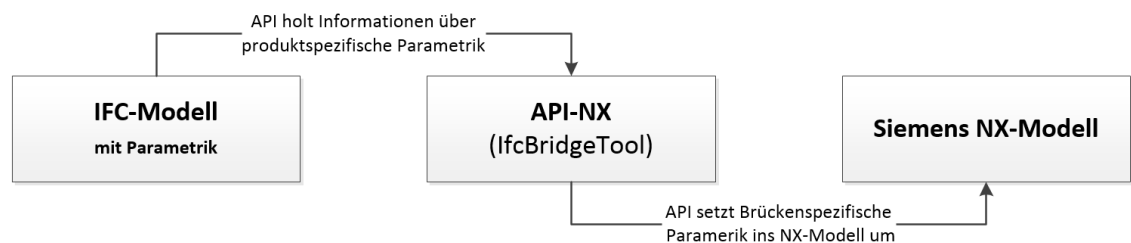


Abbildung 59: Hergang des Imports eines IFC-Modells mit der Integration der Parametrik im IFC-Modell [24]

Die parametrischen Regeln innerhalb der IFC-Datei werden analog zu den anderen bisher erwähnten IFC-Entitäten gespeichert. Sie werden im EXPRESS-Schema aufgenommen und in der IFC-Datei durch Nummerierung eindeutig referenziert. Nachfolgende Kapitel erläutern unterschiedliche Konzepte, die sich dafür eignen, parametrische Geometrie der Bauteile direkt in der IFC-Datei zu speichern.

An dieser Stelle soll erwähnt werden, dass der Export der IFC-Dateien aus Modellierungsanwendungen wie bspw. Siemens NX eine dringende Voraussetzung für den praktikablen Einsatz von IFC-Modellen in der Bauindustrie darstellt. Die Entwicklung einer Export-Schnittstelle oder die Integration der IFC-Exportmöglichkeiten ins IfcBridgeTool

konnte aufgrund der begrenzten Bearbeitungszeit im Rahmen der vorliegenden Arbeit nicht umgesetzt werden.

6 Parametrisches Modellieren

6.1 Einführung und Motivation

Die Suche nach optimaler Form eines Bauwerksmodells ist ein schwerer und langwieriger Prozess, da jedes einzelne Teil innerhalb des Gesamtmodells nicht einzelstehend gebaut wird, sondern meistens eine konstruktive, geometrische Beziehung zu anderen Bauteilen besitzt. Die Parametrik definiert die Abhängigkeiten zwischen geometrischen Elementen [27] und hilft somit die notwendige Flexibilität und Leistungsfähigkeit, die die heutige Bauindustrie von einem digitalen komplexen Bauwerksmodell verlangt, zu realisieren.

Im Hochbau werden parametrische Gebäudemodelle bereits von zahlreichen Bauunternehmen und Architektenbüros erfolgreich im Planungsprozess eingesetzt. Das Programm Revit von der Firma Autodesk bietet die Möglichkeit, semantische und parametrische Gebäudemodelle zu erzeugen. Dabei werden im Zuge der Modellierung Randbedingungen und Parameter von einzelnen Bauteilen festgelegt, die die geometrische Abhängigkeit zu anderen Bauteilen beschreiben. Wird einer dieser Parameter geändert, folgt eine automatische Anpassung des neuen Modells an die festgelegten Randbedingungen und eine Aktualisierung der Geometrie aller betroffenen Elemente. Somit wird der Prozess der iterativen Annäherung an das optimale Ergebnis mit Hilfe der Parametrik erleichtert und beschleunigt.

Auch für die Brückenbauwerke ist die parametrische Modellierung aufgrund der komplexen Konstruktion der Über- und Unterbauten dringend notwendig, um den Aufwand für den Planungsprozess zu reduzieren. Doch die Entwicklung der parametrischen Brückenmodellierung ist im Rückstand im Vergleich zur Hochbauindustrie. Die Planungsgesellschaft Obermeyer hat eine allgemeine Vorgehensweise entwickelt, mit der man Brückenmodelle mit parametrischen Abhängigkeiten im Programm Siemens NX modellieren kann. Dieses Verfahren wird von der Firma Obermeyer erfolgreich in der Praxis für die Brückenplanung angewendet.

Es ist wünschenswert diese parametrische Brückenbeschreibung softwareunabhängig zu machen. IFC4-Standard besitzt noch keine Parametrik, bietet sich jedoch aufgrund seines Erfolges an. Es existieren Entwicklungen und Vorschläge zur Integration parametrischer Abhängigkeiten direkt in eine IFC-Datei. Im folgenden Kapitel werden diese erläutert und

anschließend anhand einer Beispielsbrücke gezeigt, wie eine mögliche Beschreibung des parametrischen Brückenunterbaus in IFC realisiert werden kann.

6.2 Verfahren zur Speicherung der Parametrik in IFC

6.2.1 Referenzenmodell

Ein IFC-Modell wird über die einzelnen Entitäten innerhalb einer IFC-Datei beschrieben, die im STEP-Dateiformat gespeichert wird. Die Eigenschaften der einzelnen Entitäten und somit die geometrische Form der Bauteile werden durch die Entitäts-Attribute charakterisiert, die im STEP-Format nur als konstante, alphanumerische Werte gespeichert werden können. Um die automatische Generierung der Geometrie zu ermöglichen, müssen die Attribute als Variablen gespeichert werden, was im STEP-Format jedoch nicht erlaubt ist. Ein Team geleitet von Peter Bonsma von der Firma RDF [28] schlug eine Methode vor, um parametrische Abhängigkeiten direkt in die IFC-Datei zu integrieren und somit der oben beschriebenen Problematik entgegenzuwirken. Für die Verwirklichung dieses Verfahrens sorgen hauptsächlich 3 neu eingeführte Entitäten:

- `IfcParametricBinding`
- `IfcParametricConstant` und
- `IfcParametricFormula`

Anhand der Modellierung einer einfachen Brückenwiderlagerwand in IFC wird das Verfahren in den folgenden Ausführungen erläutert.

Die Höhe der Widerlagerwand soll parametrisiert werden und über die Formel 1 ermittelt werden:

Formel 1: Berechnung der Widerlagerhöhe

$$\text{Widerlagerhöhe} = z\text{Koordinate der Überbauplatte} - \text{Fundamentoberkante} - 0.5$$

Zunächst wird die linke Seite der Gleichung, in unserem Fall nur die Widerlagerhöhe, aus bestehenden Objekten des IFC-Modells referenziert. Die geometrische Beschreibung der Widerlagerwand erfolgt mit dem Extrusionsverfahren. Der rechteckige Querschnitt wird senkrecht nach unten extrudiert. Die Entität `IfcExtrudedAreaSolid` enthält das Attribut `Depth`, das der Extrusionstiefe und somit der Höhe der Widerlagerwand entspricht. Man

referenziert das Attribut `Depth` in einem separaten Objekt `IfcParametricBinding`, das seinerseits über eigene Attribute charakterisiert ist. Die

Abbildung 60 zeigt die Attribute des Referenzobjektes (`IfcParametricBinding`) und des Zielobjektes (`IfcExtrudedAreaSolid`). Anhand farbiger Markierungen lässt sich der Zusammenhang zwischen diesen beiden Objekten anschaulich darstellen.

#407 = IfcExtrudedAreaSolid (Zielobjekt)		
Attributname	Wert des Attributs	Eintrag in der IFC-Datei
SweptArea	#405	#407=IFCEXTRUDEDAREASOLID (#405, #411, #415, 20.00);
Position	#411	
ExtrudedDirection	#415	
Depth	20.00	

#423 = IfcParametricBinding (Referenzobjekt)		
Attributname	Wert des Attributs	Eintrag in der IFC-Datei
Name	'abutExtrusionDepth'	#423=IFCPARAMETRICBINDING ('abutExtrusionDepth', 'height of the abutment wall', #407, 'Depth', \$, .T., .F., .T., .T.);
Description	'height of the abutment wall'	
BoundObject	#407	
BoundAttribute	Depth	
BoundIndex	\$	
UserRead	.T.	
UserWrite	.F.	
ModelRead	.T.	
ModelWrite	.T.	

referenziert

Abbildung 60: `IfcParametricBinding` EXPRESS-Schema [24]

Das Attribut `Name` definiert den Variablennamen für das Referenzobjekt, in unserem Beispiel `abutExtrusionDepth`. Anhand dieser Bezeichnung wird das Referenzobjekt in den mathematischen Formeln (`IfcParametricFormula`) identifiziert. Das zweite Attribut `Description` enthält ausführliche Erläuterungen und Hinweise zum Zielobjekt oder -attribut. Im `BoundObject` wird das Zielobjekt (`IfcExtrudedAreaSolid`) und im `BoundAttribute` das Zielattribut (`Depth`) referenziert (siehe

Abbildung 60). Falls das Zielattribut mehrere Werte in einer Liste zusammenfasst, wird unter dem Referenzattribut `BoundIndex` eine Ganzzahl angegeben, die die Position des zu referenzierenden Elements innerhalb der Liste definiert. Andernfalls wird ein Platzhalter (\$) eingetragen. Die letzten vier Attribute des Referenzobjektes `IfcParametricBinding` sind alle vom Typ `IfcBoolean`, können also entweder wahr (.T.) oder falsch (.F.) sein. Diese legen einerseits fest, ob der Benutzer den referenzierten Wert sehen bzw. verändern

darf (UserRead, UserWrite) und andererseits, ob diese Rechte für das aktuelle Ausführungstool, indem das IFC-Model bearbeitet wird, gelten (ModelRead, ModelWrite). Ändert der Benutzer im Zuge der Modellierung ein Parameter, das mit einem Attribut verknüpft ist, bei dem ModelWrite auf .F. eingestellt ist, erfolgt eine Warnung über die Verletzung der definierten Regeln. Diese informiert den Benutzer darüber, dass seine Änderungen die Aktualisierung eines anderen Parameters erfordert, dessen Zwangsbedingungen dies jedoch nicht erlauben.

Das Referenzobjekt für die Widerlagerhöhe wurde somit erzeugt. Die rechte Seite der Formel 1 besteht aus drei Komponenten: z-Koordinate der Überbauposition, Fundamentoberkante und einem konstanten Wert, der den Überstand der Überbauplatte bezüglich der Widerlagerwandoberkante beschreibt. Die Abbildung 61 veranschaulicht die Abhängigkeit der Widerlagerhöhe von der Position des Fundamentes und der Überbauplatte.

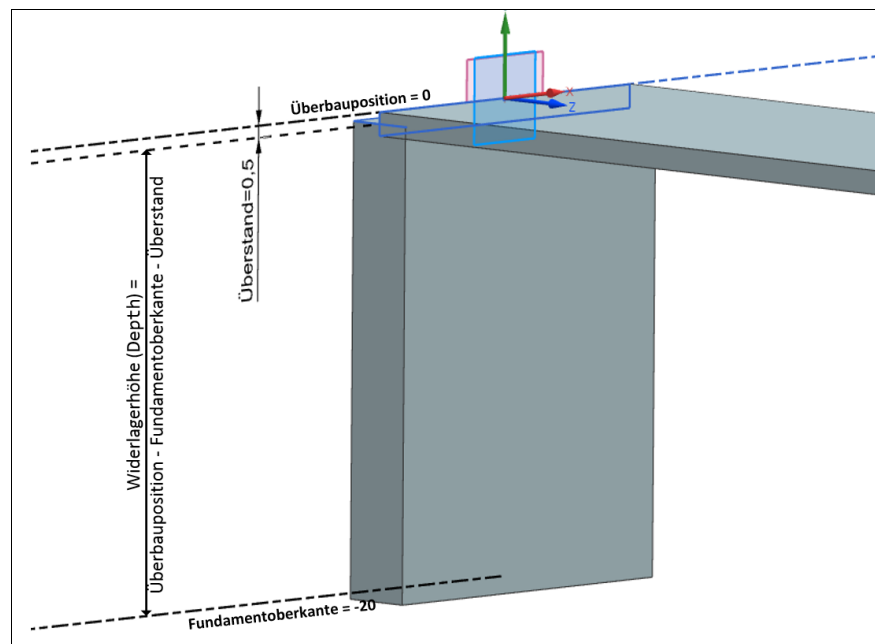


Abbildung 61: Abhängigkeit der Extrusionstiefe der Widerlagerwand von anderen Parametern [24]

Das Widerlagerfundament wird ebenfalls mit Extrusion beschrieben, indem der rechteckige Querschnitt von oben nach unten um die Fundamentdicke extrudiert wird. Die globale z-Koordinate der Fundamentoberkante, die wir für die Berechnung der Widerlagerhöhe benötigen, lässt sich aus dem Ursprung des lokalen Koordinatensystems des zu extrudierenden Fundamentquerschnitts referenzieren. Die Abbildung 62 zeigt den

hierarchischen Pfad ausgehend vom übergeordneten Extrusionsobjekt (IfcExtrudedAreaSolid) bis zum benötigten Attribut, der die erforderliche z-Koordinate der Fundamentoberkante speichert. Das Zielobjekt der Klasse IfcCartesianPoint enthält die x-, y- und z-Koordinate des Ursprungs des lokalen Koordinatensystems des Fundamentquerschnitts. Die Koordinaten werden innerhalb des Attributes Coordinates in einer Liste festgehalten. Das Attribut BoundIndex des erzeugten Referenzobjektes (IfcParametricBinding) greift auf das erforderliche Listenelement. Da das erste Element den Index 0 besitzt, erhält man mit BoundIndex gleich 2 das dritte Listenelement und somit die gesuchte z-Koordinate der Fundamentoberkante.

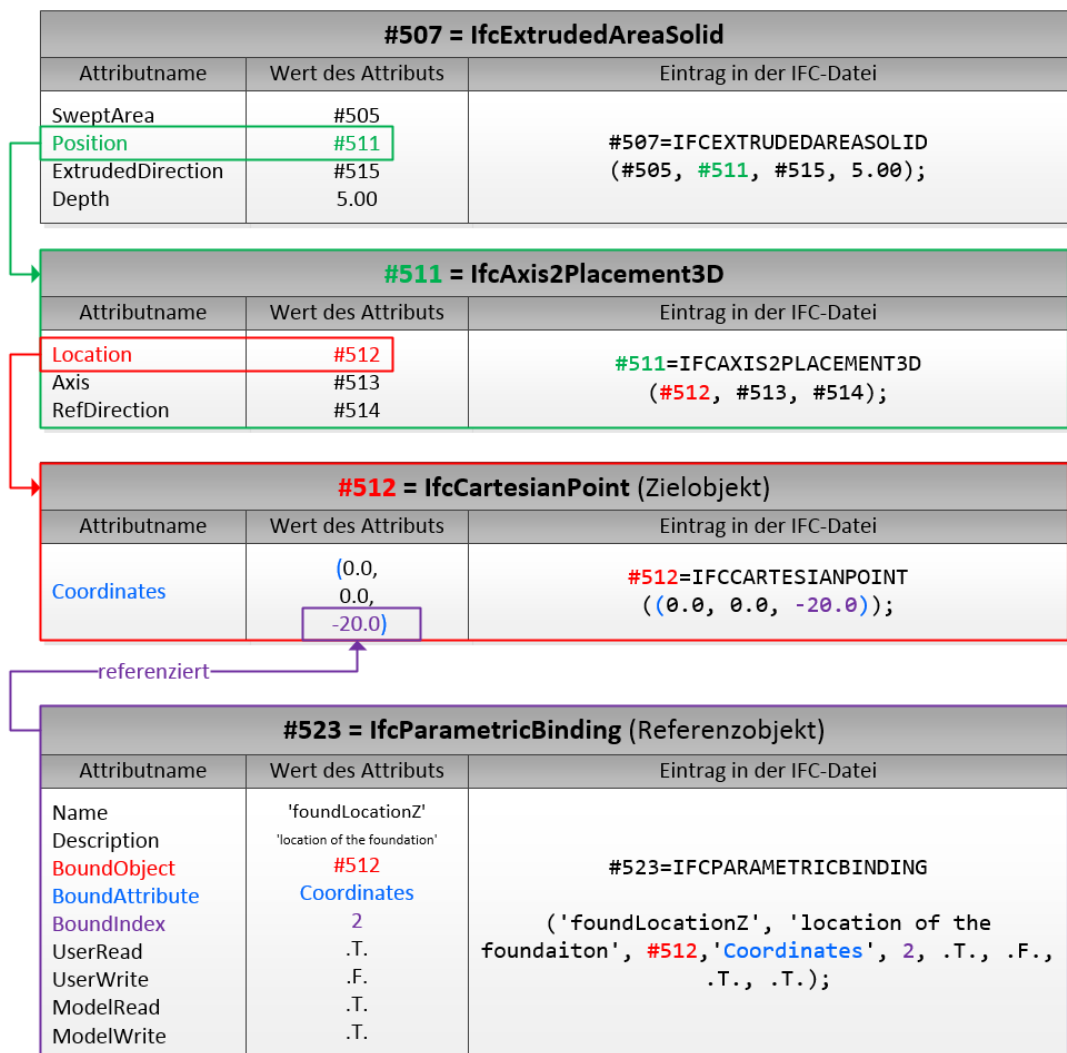


Abbildung 62: Hierarchischer Weg zur z-Koordinate der Fundamentposition [24]

Die Überbauplatte wird über die `IfcReferencedSectionedSpine` beschrieben. Die maßgebliche Stelle, an der sich die benötigte z-Koordinate der Überbauposition befindet, ist der Schnittpunkt der Referenzkurve (`IfcReferenceCurve3D`) mit der Widerlagerachse. Im IFC-Modell lässt sich der entsprechende Punkt aus der Definition der Referenzkurve ableiten. Im vorliegenden Beispiel wird die z-Koordinate analog zur Fundamentposition aus dem Objekt `IfcCartesianPoint` referenziert und ist gleich 0.

Die letzte fehlende Komponente der Formel zur Berechnung der Widerlagerhöhe (Formel 1) ist der Überstand der Überbauplatte zur oberen Kante der Widerlagerwand. Dieser ist ein konstanter Wert und beträgt 0.5. Zur Beschreibung der Konstanten wurde ein neues Objekt der Klasse `IfcParametricConstant` erstellt, das in die parametrischen Formeln einfließen kann. Über das Attribut `NominalValue` wird der Wert der Konstante festgehalten. Den vollständigen Aufbau des Objektes `IfcParametricConstant` zeigt die Abbildung 63.

#607 = IfcParametricConstant		
Attributname	Wert des Attributs	Eintrag in der IFC-Datei
Name	'overhangDeckAbut'	#607=IFCPARAMETRICCONSTANT
Description	'overhang between deck and abutment'	('overhangDeckAbut',
NominalValue	0.5	'overhang between deck and abutment',
Unit	\$	0.5, \$);

Abbildung 63: Aufbau des Objektes `IfcParametricConstant` [24]

Nachdem alle notwendigen Komponenten referenziert wurden, kann die Formel 1 von der Seite 63 mit Hilfe der neu eingeführten Klasse `IfcParametricFormula` aufgestellt werden und die Höhe der Widerlagerwand errechnet werden. Anschauliche Darstellung der Verknüpfungen der Formelattribute und der Referenzobjekte zeigt die Abbildung 65.

Der errechnete Ausgabewert wird allen Referenzobjekten zugewiesen, die im Listenattribut `DefinedOutputs` erwähnt sind. In unserem Beispiel betrifft dies nur das Referenzobjekt der Extrusionstiefe der Widerlagerwand (#423). Die Liste der Eingabewerte enthält dagegen drei Referenzobjekte – Überbauposition (#571), Fundamentoberkante (#523) und den Überstand (#607). Diese werden im Listenattribut `DefiningInputs` gespeichert. Die Art der Verknüpfung der Eingabewerte wird unter dem Attribut `Operation` festgelegt. Alle Operatoren werden in IFC unter dem Typen-Objekt `IfcParametricOperatorEnum` zusammengefasst. Es werden drei Operatorarten unterschieden: numerische, logische und

Textoperatoren. Die letzteren können Zeichenketten miteinander verknüpfen und in bestimmte andere Formate umwandeln, werden jedoch im Rahmen der Geometriemodellierung nicht benötigt. Die booleschen Operatoren liefern einen Wahrheitswert und die numerischen definieren primitive mathematische Rechenoperationen wie Addieren, Subtrahieren usw. Besteht bspw. die Inputliste aus 3 Eingabewerten und der Operator ADD wird definiert, entspricht die Ausgabe der Summe aus allen drei Elementen. Die Abbildung 64 zeigt eine erweiterte Auflistung der Operatoren, die verwendet werden um Eingabewerte unterschiedlicher Typen miteinander zu verknüpfen.

IfcParametricOperatorEnum		
Numerische Operatoren	Logische Operatoren	Textoperatoren
ADD SUBTRACTION MULTIPLY DIVIDE EQUALS MINIMUM MAXIMUM AVERAGE SINE, COSINE, TANGENT	EQUALS NOTEQUALS GREATERTHAN LESSTHAN GREATERTHANOREQUALTO LESSTHANOREQUALTO MATCHES DIFFERS	WORD PHRASE LIST SENTENCE PARAGRAPH

Abbildung 64: gängigsten Operatoren sortiert nach Kategorie [28] [24]

Setzt man das Attribut `Operation` zu `.USERDEFINED.`, ermöglicht das die Eintragung der kompletten rechten Seite der Formel unter dem Attribut `UserDefinedOperation`. Genau diese Technik wird im Rahmen des vorliegenden Beispiels verwendet. Dabei entsprechen die einzelnen Komponenten der Formel den definierten Variablennamen der Referenzobjekte (`IfcParametricBinding`), Konstanten (`IfcParametricConstant`) oder anderer Formeln (`IfcParametricFormula`). Für die mathematischen Ausdrücke werden bekannte arithmetische Zeichen verwendet (+, -, *, /). Mit dem Attribut `IsActive`, das durch ein logisches Wert definiert ist, lässt sich die Ausführung der Formel aktivieren oder deaktivieren. Bei Eintragung eines Platzhalters \$ wird der Standardwert `.T.` (wahr) verwendet. In IFC ist eine Verkettung der Formeln zugelassen, d.h. sowohl Ausgabe- als auch Eingabewerte der Formeln können wiederum andere Formeln sein.

<code>#423=IFCPARAMETRICBINDING('abutExtrusionDepth','height of the abutment wall',#407,'Depth',\$.T.,.F.,.T.,.T.);</code>
<code>#523=IFCPARAMETRICBINDING('foundLocationZ','location of the foundaiton', #512,'Coordinates',2,.T.,.F.,.T.,.T.);</code>
<code>#571=IFCPARAMETRICBINDING('deckLocationZ','location of the deck', #570,'Coordinates',2,.T.,.F.,.T.,.T.);</code>
<code>#607=IFCPARAMETRICCONSTANT('overhangDeckAbut','overhang between deck and abutment',0.5,\$);</code>

#700 = IfcParametricFormula		
Attributname	Wert des Attributs	Eintrag in der IFC-Datei
Name	'calcAbutHeight'	#700=IFCPARAMETRICFORMULA
Description	'calculation of the abutment height'	('calcAbutHeight','calculation
DefinedOutputs	#423	of the abutment height',
DefiningInputs	#523, #571, #607	(#423), (#523, #571, #607),
Operation	.USERDEFINED.	.USERDEFINED., 'deckLocationZ –
UserDefinedOperation	'deckLocationZ – foundLocationZ – overhangDeckAbut'	foundLocationZ – overhangDeckAbut',
IsActive	\$	\$);

Abbildung 65: Berechnung der Widerlagerwand mit IfcParametricFormula [24]

Die Höhe der Widerlagerwand errechnet sich, indem die Werte der jeweiligen Referenzobjekte in die „Formel 1: Berechnung der Widerlagerhöhe“ eingesetzt werden:

$$abutExtrusionDepth = deckLocationZ - foundLocationZ - overhangDeckAbut$$

$$abutExtrusionDepth = 0 - (-20) - 0.5 = 19.5$$

Somit besitzt die Höhe des Widerlagers parametrische Abhängigkeiten zu anderen geometrischen dimensional GröÙen und ist unabhängig von dem eigentlichen Eintrag der Extrusionstiefe innerhalb des IFC-Models. Ändert sich der Wert einer der Parameter, die in die Formel einfließen, wird die Widerlagerhöhe neu ermittelt und aktualisiert.

Die vorgestellte Methode ist imstande jedes Attribut, das im IFC-Model definiert ist, zu referenzieren und diesem einen variablen Wert zuzuweisen, der seinerseits wiederum von anderen beliebigen Attributen des IFC-Models abhängig gemacht werden kann. Im nachfolgenden Kapitel wird eine Methode erläutert, die es ermöglicht IFC-Objekte über geometrische Zwangsbedingung direkt zu verknüpfen.

6.2.2 Skizzenparametrikmodell

Die Technische Universität München entwickelte in Zusammenarbeit mit der TU Regensburg im Rahmen des Projektes ForBAU eine Vorgehensweise, parametrische Skizzen zu erzeugen. Die vorgeschlagene Methode basiert auf den geometrischen

Darstellungstechniken, die von Modellierungsprogrammen wie Siemens NX und Autodesk Inventor angewendet werden.

Die Erstellung eines Raumkörpers basiert oft auf einer 2D-Skizze. Beispielsweise benötigt die Extrusionsmethode einen 2D-Querschnitt, um diesen in eine bestimmte Richtung zu extrudieren. Auch die Beschreibungsmethode der Überbauplatten im Rahmen des IfcBridge (IfcReferencedSectionedSpine) basiert auf der Definition mehrerer 2D-Querschnitte entlang einer Referenzkurve. Daraus folgt, dass Skizzen auf Ebenen eine wichtige Rolle bei der Modellierung komplexer Raumobjekte spielen. Um eine 2D-Zeichnung parametrisch zu gestalten, wurden Zwangsbedingungen zwischen primitiven geometrischen 2D-Objekten innerhalb der Skizze erzeugt. Wird ein Teil der Skizze geändert, bewirkt diese intelligente Kopplung eine automatische Anpassung der Skizzengeometrie, falls keine Konflikte mit der betreffenden Zwangsbedingung hervorgerufen wurden. Ist das nicht der Fall, wird eine Warnung ausgelöst, die den Nutzer über die konkurrierende Zwangsbedingung informiert. Außerdem wurden dimensionale Zwangsbedingungen eingeführt, wodurch bestimmte geometrische Elemente innerhalb der Skizze feste Abmessungen, Winkel oder Abstände erhalten. Auch diese Regeln sollen zu einer automatischen Anpassung bzw. einer Warnung führen.

Das übergeordnete IFC-Objekt für die parametrischen Beziehungen ist die Klasse IfcParametricConstraint. Darin werden die beiden Arten der Zwangsbedingungen festgelegt: topologischen (IfcGeometricConstraint) und dimensionalen (IfcDimensionalConstraint), die in den folgenden Ausführungen detailliert erläutert werden.

Geometrische Zwangsbedingungen

Der Vorteil von topologischen Zwangsbedingungen ist die Erzeugung der parametrischen Abhängigkeiten, ohne komplizierte mathematische Formeln aufstellen zu müssen. Das erleichtert die Lesbarkeit des Modells und beschleunigt somit den Planungsprozess.

Primitive, geometrische 2D-Elemente, zwischen denen die topologischen Zwangsbedingungen erzeugt werden, sind in der IFC-Klasse IfcParametricGeometryElement aufgelistet. Im aktuellen Schema sind folgende drei Klassen aufgeführt: der Punkt (IfcPoint), die Linie (IfcLine) und die getrimmte Kurve

(IfcTrimmedCurve). Die Linie (IfcLine) und die getrimmte Kurve (IfcTrimmedCurve) erben von der allgemeinen Kurve (IfcCurve) und gelten somit als spezielle Linien. Eine Erweiterung der oben erwähnten Auflistung um zusätzliche Elemente wie Polylinie (IfcPolyline), Splinekurve (IfcBSplineCurve), Kreis (IfcCircle), Ellipse (IfcEllipse) und einige mehr, hat sich als sinnvoll herausgestellt und wird deswegen als Nachtrag zur Skizzenparametrik-Methode vorgeschlagen.

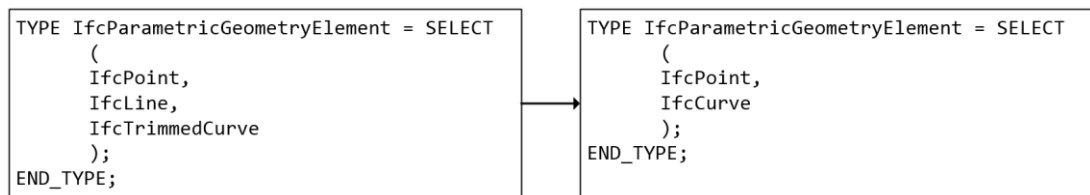


Abbildung 66: Vorschlag der Änderung der Klasse IfcParametricGeometryElement [24]

Wie die Abbildung 66 zeigt, genügt lediglich die Aufnahme des Elternelementes Kurve (IfcCurve) in die Auflistung IfcParametricGeometryElement innerhalb des EXPRESS-Schemas, um alle vorhandenen Kurvenspezifikationen in IFC abzudecken. Auf die gleiche Weise werden auch spezielle Punkte, wie beispielsweise ein kartesischer Punkt (IfcCartesianPoint) oder ein Punkt auf einer Kurve (IfcPointOnCurve) automatisch abgedeckt, da diese vom Objekt IfcPoint erben, das in den Auswahlmöglichkeiten des Typen IfcParametricGeometryElement enthalten ist. Im Zuge der Skizzenerstellung können diverse topologische Zwangsbedingungen zwischen geometrischen 2D-Objekten hergestellt werden. Die acht gängigsten Verknüpfungsmöglichkeiten, die im Rahmen der Skizzenparametrikmethode verwendet werden, illustriert die Abbildung 67.

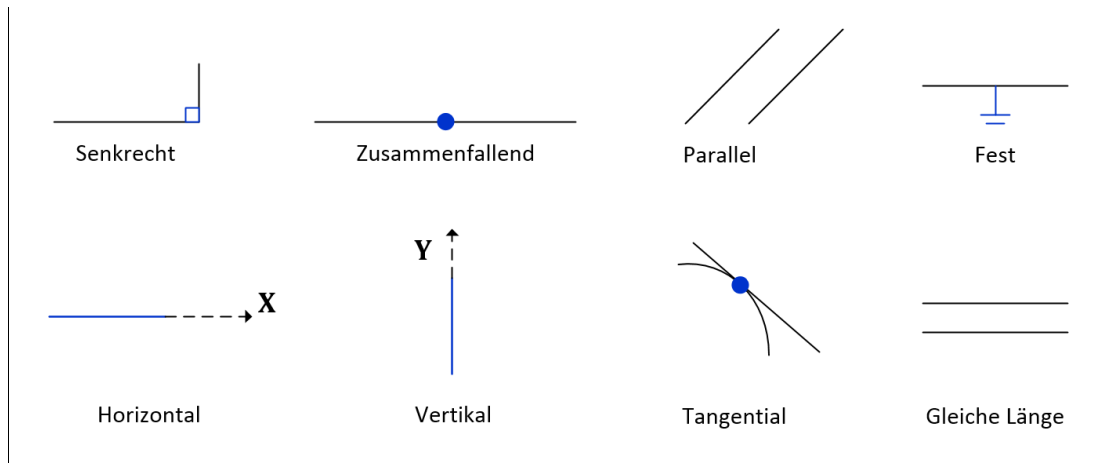


Abbildung 67: topologische Zwangsbedingungen [27]

Für jede topologische Zwangsbedingung existiert eine IFC-Klasse, die alle im IFC-EXPRESS-Schema aufgeführt sind. Innerhalb jeder Klasse werden die Anzahl und der Typ der geometrischen Objekte explizit vorgeschrieben, die über die entsprechende Zwangsbedingung miteinander verknüpft werden können. Diese variiert in Abhängigkeit von der Art der Zwangsbedingung:

- `IfcPerpendicularGeometricConstraint` (senkrecht)
Zwei gerade Linien schließen einen rechten Winkel ein.
- `IfcCoincidentGeometricConstraint` (zusammenfallend)
Zwei Punkte nehmen die gleiche geometrische Position ein.
- `IfcParallelGeometricConstraint` (parallel)
Zwei gerade Linien schneiden sich nicht.
- `IfcFixedGeometricConstraint` (fest, unveränderbar)
Ein und mehrere Elemente besitzen feste Position in der Skizze
- `IfcHorizontalGeometricConstraint` (horizontal)
Eine oder mehrere gerade Linien besitzen eine waagrechte Lage
- `IfcVerticalGeometricConstraint` (vertikal)
Eine oder mehrere gerade Linien besitzen eine lotrechte Lage
- `IfcTangentGeometricConstraint` (tangential)
Eine gerade Linie berührt eine Kurve in einem bestimmten Punkt
- `IfcEqualLengthGeometricConstraint` (gleiche Länge)
Zwei Linien besitzen identischen Betrag der Länge

Im IFC-Model (STEP-Datei) werden die zu verknüpfenden geometrischen Objekte (Punkt, Linie, Kurve) über ihre Objektnummern referenziert. Am Ende dieses Kapitels wird anhand eines Beispiels gezeigt, wie topologische und dimensionale Zwangsbedingungen bei Erstellung eines Querschnitts zum Einsatz kommen.

Dimensionale Zwangsbedingungen

Bei dimensionalen Zwangsbedingungen, handelt es sich um die Definition fester Abmessungen der geometrischen Objekte innerhalb einer Skizze. Es wurden fünf unterschiedliche Dimensionsarten eingeführt: horizontale, vertikale, parallele, Winkel- und Radiusdimension (Abbildung 68).

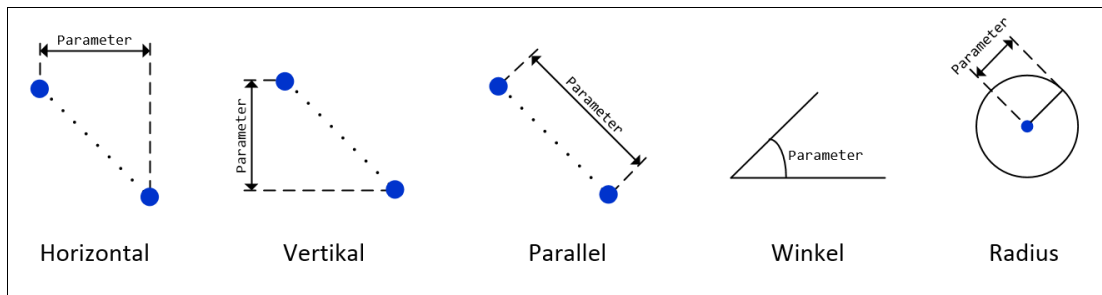


Abbildung 68: dimensionale Zwangsbedingungen [27]

Für jede dieser Zwangsbedingungen wurden separate IFC-Klassen eingeführt. In gleicher Weise, wie bei den topologischen Zwangsbedingungen, besitzen diese jeweils Attribute, die die Anzahl bzw. den Typ der miteinander verknüpften Objekte vorschreiben:

- `IfcHorizontalDimensionalConstraint` (horizontal)
Fester horizontaler Abstand zweier Punkte
- `IfcVerticalDimensionalConstraint` (vertikal)
Fester vertikaler Abstand zweier Punkte
- `IfcParallelDimensionalConstraint` (parallel)
Fester direkter Abstand zweier Punkte
- `IfcAngularDimensionalConstraint` (Winkel)
Fester Winkel zwischen zwei geraden Linien
- `IfcRadialDimensionalConstraint` (Radius)
Fester Radius eines Kreises

Die übergeordnete Klasse `IfcDimensionalConstraint` enthält das Attribut `Parameter`, der den Dimensionswert speichert. In der ersten Version des Skizzenparametrikmodells erfolgte die Repräsentation dieses Attributs als eine Zeichenkette [30]. Nach der Fusion mit dem parametrischen Referenzenmodell, das im Kapitel 6.2.1, Seite 63 erläutert wurde, wird der Dimensionswert über eine mathematische Formel (`IfcParametricFormula`) repräsentiert und kann somit von jedem beliebigen Objekt innerhalb des IFC-Modells abhängig gemacht werden.

6.2.3 Beschreibung der Überbauplatte mit veränderlicher Querschnittshöhe

Anhand eines konkreten Beispiels, das auf den Dokumentationen der beiden parametrischen Modelle (Referenzenmodell, Skizzenparametrik) basiert, wird im Folgenden die Erzeugung der Parametrik mit Hilfe der topologischen und dimensional Zwangsbedingungen in IFC näher erläutert.

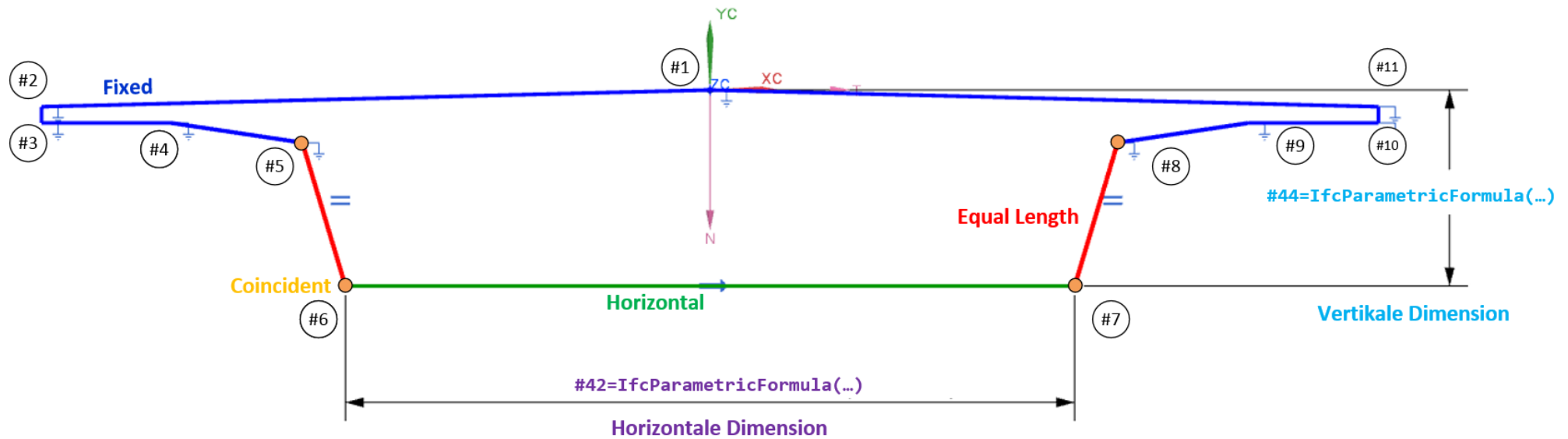
Es handelt sich um eine Hohlkastenbrücke mit veränderlicher Höhe der Überbauplatte über die Brückenlänge. Die Beschreibung der Tragfläche in IFC erfolgt mit der geometrischen Repräsentation `IfcReferencedSectionedSpine`. Mehrere Skizzen mit unterschiedlichen Querschnitten werden entlang der Referenzkurve positioniert. Zur parametrischen Beschreibung der Querschnitte wird die neu eingefügte Klasse `IfcParametricProfileDef` verwendet. Die Eintragung im EXPRESS-Schema zeigt die Abbildung 69.

```

ENTITY IfcParametricProfileDef
  SUBTYPE OF (IfcProfileDef);
  GeometricElements:      LIST [1:?] OF IfcParametricGeometryElement;
  DimensionalConstraints: LIST [1:?] OF IfcDimensionalConstraint;
  GeometricConstraints:   LIST [1:?] OF IfcGeometricConstraint;
END_ENTITY;
    
```

Abbildung 69: `IfcParametricProfileDef` im EXPRESS-Schema [28]

Das Attribut `GeometricElements` speichert alle verwendeten geometrischen Elemente innerhalb der Querschnittsskizze. `DimensionalConstraints` und `GeometricConstraints` beinhalten die dimensional bzw. die topologischen Zwangsbedingungen, die innerhalb der Skizze zum Einsatz kommen. Eine anschauliche Darstellung des Frontquerschnitts der Hohlkastenbrücke und die Beschreibung des parametrischen Querschnitts in IFC bietet die Abbildung 70.



geometrischen Elemente	
Kartesische Punkte	Linien
#1=IFCCARTESIANPOINT(...)	#20=IFCPOLYLINE(#1,#2);
#2=IFCCARTESIANPOINT(...)	#21=IFCPOLYLINE(#2,#3);
#3=IFCCARTESIANPOINT(...)	#22=IFCPOLYLINE(#3,#4);
#4=IFCCARTESIANPOINT(...)	#23=IFCPOLYLINE(#4,#5);
#5=IFCCARTESIANPOINT(...)	#24=IFCPOLYLINE(#5,#6);
#6=IFCCARTESIANPOINT(...)	#25=IFCPOLYLINE(#6,#7);
#7=IFCCARTESIANPOINT(...)	#26=IFCPOLYLINE(#7,#8);
#8=IFCCARTESIANPOINT(...)	#27=IFCPOLYLINE(#8,#9);
#9=IFCCARTESIANPOINT(...)	#28=IFCPOLYLINE(#9,#10);
#10=IFCCARTESIANPOINT(...)	#29=IFCPOLYLINE(#10,#11);
#11=IFCCARTESIANPOINT(...)	#30=IFCPOLYLINE(#11,#1);

topologischen Zwangsbedingungen
#31=IFCFIXEDGEOMETRICCONSTRAINT((#20,#21,#22,#23,#27,#28,#29,#30));
#32=IFCEQUALLENGTHGEOMETRICCONSTRAINT((#24,#26);
#33=IFCHORIZONTALGEOMETRICCONSTRAINT((#25));

dimensionale Zwangsbedingungen
#41=IFCHORIZONTALDIMENSIONALCONSTRAINT(#42,(#6,#7);
#42=IFCPARAMETRICFORMULA(...);
#43=IFCVERTICALDIMENSIONALCONSTRAINT(#44,(#1,#6);
#44=IFCPARAMETRICFORMULA(...);

```
#50=IFCPARAMETRICPROFILEDEF((#1,#2,#3,#4,#5,#6,#7,#8,#9,#10,#11,#20,#21,#22,#23,#24,#25,#26,#27,#28,#29,#30),(#41,#43),(#31,#32,#33));
```

Abbildung 70: Parametrische Außenkontur eines Querschnitts der Hohlkastenbrücke und ihre Beschreibung in IFC [30] [24]

Der vorgestellte parametrische Querschnitt gilt als Muster für alle Querschnitte die entlang der Brückenachse stationiert werden.

Zunächst betrachten wir die topologischen Beziehungen innerhalb der Skizze. Die Linien im oberen Skizzenabschnitt (dunkelblau) werden unter der festen Zwangsbedingung `IfcFixedGeometricConstraint` zusammengefasst. Damit wird festgelegt dass die Position und die Abmessungen dieser Skizzenelemente in allen Brückenquerschnittsskizzen unverändert bleiben. Die Zwangsbedingung `IfcEqualLengthGeometricConstraint` enthält die beiden geneigten Stege (rot) und legt fest, dass diese stets eine identische Länge aufweisen, wobei der Betrag dieser Länge variieren kann. Mit `IfcHorizontalGeometricConstraint` wird die waagrechte Lage des Untergurtes (grün) erzwungen, seine Länge kann jedoch unterschiedlich sein. Die zusammenfallende Zwangsbedingung (`IfcCoincidentGeometricConstraint`) zweier Punkte (orange Markierung) wird im vorgestellten Beispiel nicht benötigt, da der Start- und Endpunkt zweier Nachbarlinien ein und derselbe Punkte im IFC-Model sind und die Linien somit automatisch miteinander verknüpft sind.

Die dimensional Zwangsbedingungen des Querschnitts sorgen für die variablen Querschnittsformen entlang der Brückenachse. Die Breite des Untergurtes und die Gesamthöhe des Querschnitts werden jeweils in Abhängigkeit von der Position des Querschnitts auf der Überbauachse berechnet und als `IfcParametricFormula` innerhalb der Objekte `IfcHorizontalDimensionalConstraint` und `IfcVerticalDimensionalConstraint` angegeben. Aufgrund der topologischen Zwangsbedingung, werden die Linien zu einem zusammenhängenden Polygonzug geschlossen und es entsteht die fertige Querschnittsgeometrie (Abbildung 71).

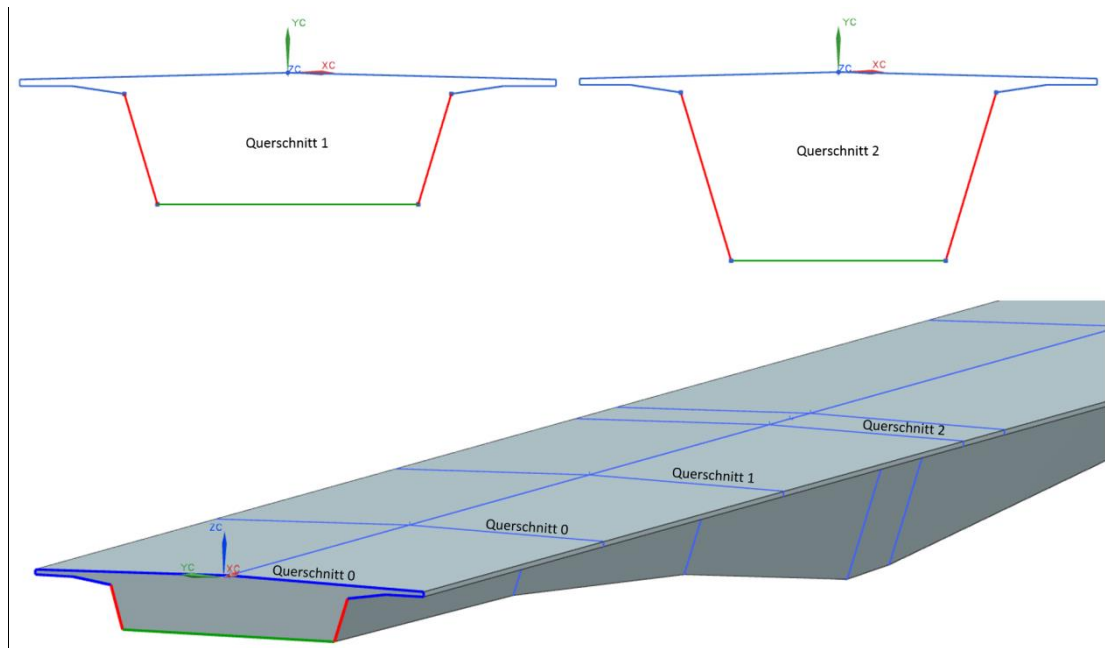


Abbildung 71: variable Höhe der Querschnitte [25]

Im Idealfall gibt der Benutzer lediglich die Anzahl von Querschnitten an, die vom Modellierungsprogramm entlang der Referenzkurve in gleichen Abständen positioniert werden sollen.

Das beschriebene Modell ermöglicht eine automatische Anpassung der Brückengeometrie, falls ein Parameter geändert. Dies führt zum effizienteren und flexibleren Modellierungsvorgang. Die Suche nach der optimalen Form des Bauwerksmodells führt schneller zu einem Ergebnis als bei einem parametrikfreien Modell.

6.3 Modellierung eines parametrischen Brückenunterbaus in IFC

Die Gestaltung des Brückenunterbaus ist an viele Umstände gebunden, die kaum vollständig erfasst werden können. Offensichtliche Faktoren, die die geometrische Form und Position der Widerlager und Pfeiler beeinflussen sind beispielsweise der Überbau, die unterführte Straße und die Umgebungsverhältnisse (Böschungsneigung). Das vorliegende Kapitel soll anhand eines Beispiels aufzeigen, wie einige dieser Faktoren in IFC umgesetzt werden können und welche Vorteile parametrisches IFC gegenüber dem aktuellen, parametrikfreien IFC-Schema bietet. Zur Umsetzung der Parametrik kommt das im Kapitel 6.2.1 erläuterte Referenzenmodell. Der Hauptgrund für die Wahl dieses Modells zur Beschreibung

der Parametrik war das Vorhandensein eines Visualisierungsprogramms, das ein IFC-Model mit dem Referenzenmodell lesen kann und die definierte Parametrik umsetzen kann. Dagegen existiert für das Skizzenparametrikmodell (Kapitel 6.2.2) bisher noch keine Software, die ein mit diesem Modell beschriebenes Bauwerk dreidimensional darstellen kann.

Das Programm „Parametric IFC Editor“ wurde von der Firma RDF entwickelt und kann anhand der Objekte `IfcParametricBinding`, `IfcParametricConstant` und `IfcParametricFormula` Abhängigkeiten zwischen beliebigen IFC-Objekten erzeugen. Außerdem kann man auf der Benutzeroberfläche dieses Tools die Änderung der referenzierten Attribute vornehmen, die bei Abspeicherung direkt im IFC-Model umgesetzt werden. Aufgrund der benutzerfreundlichen Ansicht der Parametric IFC Editor, lässt sich die Anpassung des Modells rationaler als über die Editierung der STEP-Datei gestalten.

Die modellierte Brücke (Abbildung 72) enthält folgende Elemente:

- Überbauplatte mit einem Massivplattenquerschnitt
- Unterführte Straße
- Zwei Widerlager mit unterschiedlichen Ausführungen: Regel und Variante
- Ein Pfeiler in der Mitte der Brücke
- 2 Böschungskörper
- Flachgründungen unter den Widerlagern und dem Pfeiler

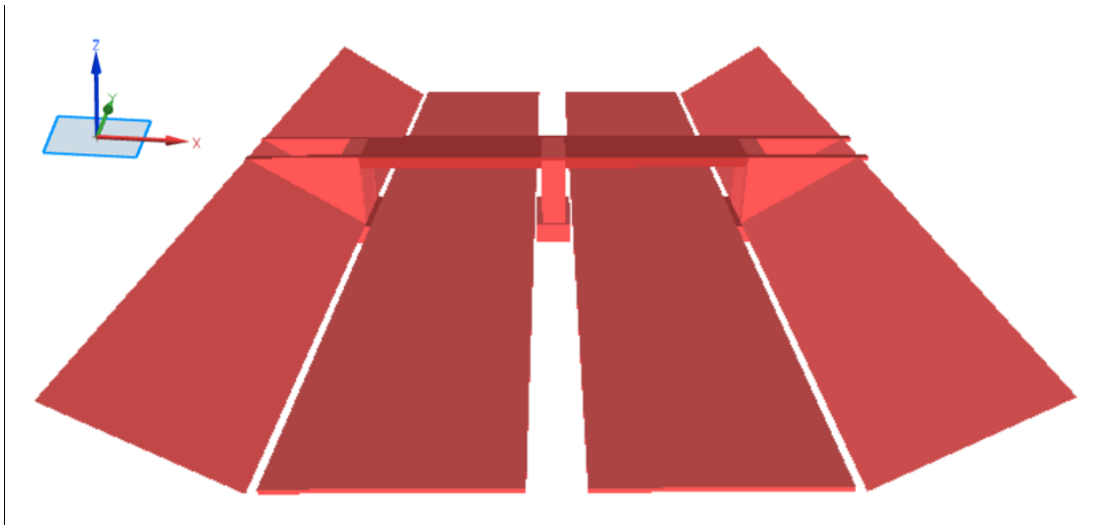


Abbildung 72: parametrische Brücke, dargestellt mit im Programm "Parametric IFC Editor" [31] [24]

Zunächst wurden diverse Objekte mit Hilfe der Klasse `IfcParametricBinding` referenziert, deren Attribute einen variablen Wert erhalten sollen. Weiterhin wurde eine Reihe von Faktoren als `IfcParametricConstant` definiert, die die Form der Brückenelemente beeinflussen. Diese werden in den jeweiligen Abschnitten dieses Kapitels näher erläutert. Schließlich werden anhand der aufgestellten Formeln mit Hilfe der Klasse `IfcParametricFormula` die benötigten Werte ermittelt und den entsprechenden Referenzobjekten zugewiesen.

Globale Parameter der Brücke und der unterführten Straße

Der Ursprung des globalen Koordinatensystems befindet sich am linken Rand des Brückenüberbaus. In diesem Beispiel wurde der Überbau nicht mit Hilfe der Klasse `IfcReferencedSectionedSpine` beschrieben, wie es in `IfcBridge` üblich ist, sondern mit dem Extrusionsverfahren. Aufgrund des geradlinigen und waagrechten Verlaufs des Überbaus, lässt sich die Extrusion realisieren. Der Parameter Brückenlänge (`bridgeLength`), wird als eine Konstante gespeichert und für die Berechnung diverser Attribute verwendet. Beispielsweise beträgt die x-Koordinate der Pfeilerposition die halbe Brückenlänge ($\text{bridgeLength}/2$). Die Position des rechten Widerlagers und der rechten Böschung ist gleich der Brückenlänge. Eine Vergrößerung der Brückenlänge führt zu entsprechenden Anpassungen. Zusätzlich werden die Fahrbahnbreiten der unterführten Straße aktualisiert (Abbildung 73).

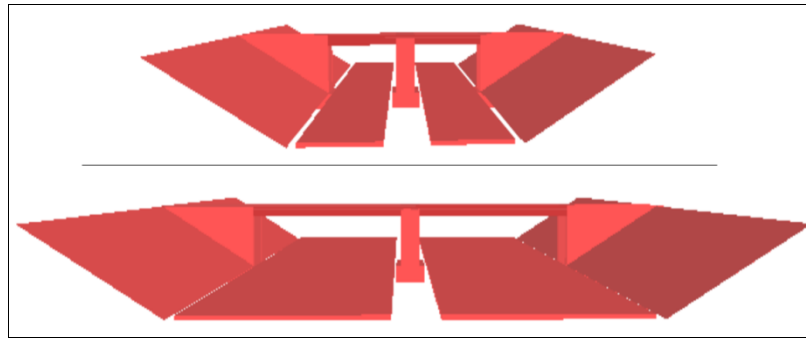


Abbildung 73: Änderung der Brückenlänge [31] [24]

Ein weiterer wichtiger Faktor, der die Konstruktion der Widerlager und der Böschungen beeinflusst, ist der Kreuzungswinkel zwischen der über- und unterführten Straße. Die Abbildung 74 zeigt den Grundriss einer recht- und einer schiefwinkligen Kreuzung. Die letztere kommt heutzutage am häufigsten vor, da die „Trassierungselemente des Straßenbaus, die einen zügigen Verkehrsfluss ermöglichen sollen, die Kreuzung in jedem Winkel verlangen“ [16].

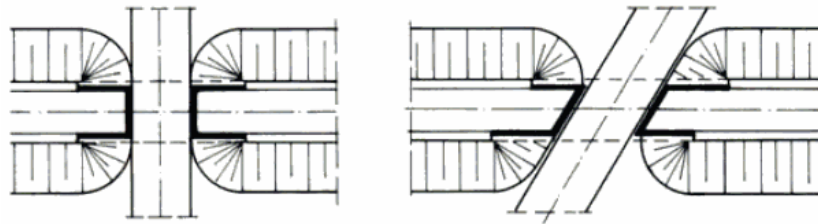


Abbildung 74: Kreuzungsmöglichkeiten der über- und unterführten Straße [16]

Die Beschreibung der unterführten Straße erfolgt mittels einer einfachen Extrusion. Die Extrusionsrichtung lässt sich über das Parameter Axis, das im Rahmen der Extrusionsfläche (SweptArea) definiert wird, steuern. Im rechtwinkligen Kreuzungsfall entspricht Axis der globalen y-Achse. Die entsprechende Definition der Vektorrichtungen in IFC ist $(0.0, 1.0, 0.0)$. Um den Kreuzungswinkel variabel zu gestalten, wird der Eintrag für die x-Achse mit IfcParametricBinding referenziert und der Konstante underpassAxisX gleichgesetzt. Um z.B. einen Kreuzungswinkel von ca. 73° zu erhalten, setzt man underpassAxisX zu 0.3.

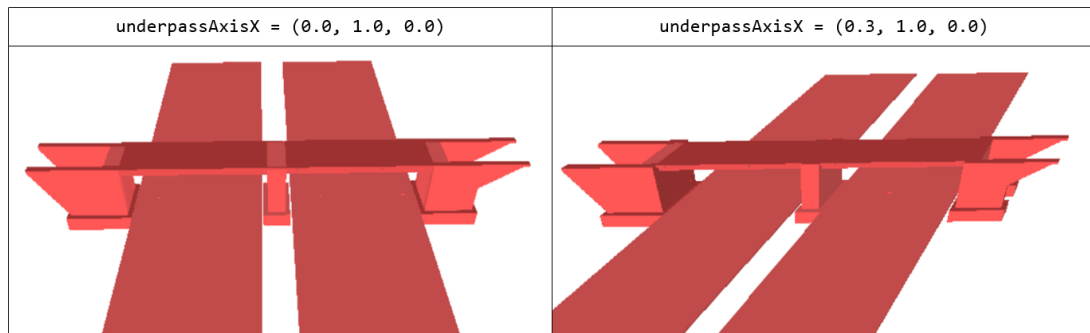


Abbildung 75: Änderung der Achse der unterführten Straße [31] [24]

Wie man aus der Abbildung 75 erkennen kann, bewirkt die Änderung der Achse der Unterführung eine Geometrieangepassung der Überbauplatte und aller Unterbauelemente. Um dies zu ermöglichen, musste jeweils die Grundrissgeometrie der 3D-Objekte (Widerlager, Pfeiler, Fundamente) von der Unterführungsachse abhängig gemacht werden. Dazu wurden alle kartesischen Koordinaten jedes Querschnittspunktes referenziert. Da es sich bei der Extrusionsfläche um einen 2D-Querschnitt handelt, gibt es zu jedem Punkt 2 Koordinaten. Anschließend wurden für jeden Punkt jeweils zwei Formeln aufgestellt, um die x- und y-Koordinate des jeweiligen Punktes in Abhängigkeit von der Unterführungsachse zu berechnen. Die Erzeugung der Parametrik innerhalb einer Skizze mit der Referenzmethode kann somit rasch zu einem hohen Datenvolumen führen. Die Skizzenparametrikmethode aus Kapitel 6.2.2 (Seite 69) eignet sich besser für diese Aufgabe. Infolge der Definition topologischer Zwangsbedingungen zwischen den geometrischen Skizzenelementen, lässt sich mit deutlich weniger IFC-Objekten und höherer Übersichtlichkeit das gewünschte Ergebnis erzielen.

Widerlagerform

Im Kapitel 3.2 (Seite 24) wurden zwei unterschiedliche Formen des Widerlagers erläutert – die Regel- und Variantenausführung. Ihre Hauptunterscheidung liegt in der Ausbildung der schrägen Flügelwände. Bei der Regelausführung verläuft die Schräge über die komplette Widerlagerhöhe in einem 60°-Winkel. Dagegen ist die Schräge der Variantenausführung nicht durchgehend und besitzt einen Winkel, der der Böschungsneigung entspricht.

Die beiden Widerlager der Beispielbrücke besitzen jeweils unterschiedliche Ausführungen. Das rechte Widerlager besitzt die Variantenausführung. Um die Abhängigkeit zwischen der Böschungsneigung und dem schrägen Winkel der Flügelwand herzustellen, wird zunächst das Attribut `Axis` des Böschungskörpers, der über die Extrusionsmethode beschrieben wurde, referenziert. Der Winkel zwischen der Böschungsachse und der globalen x-Achse entspricht der Böschungsneigung und wird über trigonometrische Funktionen ermittelt. Die Flügelwand des Widerlagers wird über die Extrusion in die globale y-Richtung erstellt. Die vorher ermittelte Böschungsneigung fließt in die Berechnungen der Punktkoordinaten für den Flügelwandquerschnitt ein. Die Abbildung 76 zeigt das Ergebnis der automatischen Anpassung der Flügelwandgeometrie bei Änderung der Böschungsneigung.

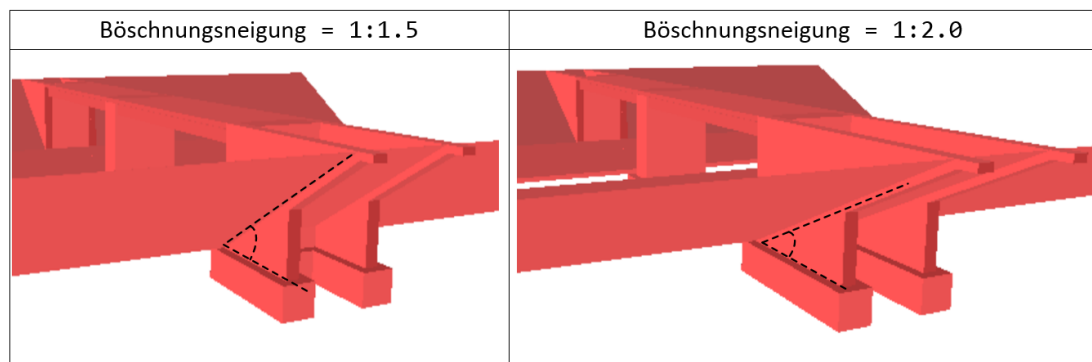


Abbildung 76: Änderung der Böschungsneigung [31] [24]

Außerdem wurden weitere Konstanten, wie Breite und Höhe der Widerlager oder Dicke der Flügelwände, eingeführt, um die Widerlagerform parametrisch zu steuern. All diese Parameter werden bei der Erstellung von Extrusionsflächen der entsprechenden Bauteile verwendet, indem sie in die Berechnungsformeln einfließen.

Pfeiler und Fundamente

Die Erzeugung der Pfeiler wird ebenfalls mit einigen Parametern versehen. Beispielsweise wird der Pfeilerquerschnitt bei schiefwinkligen Brücken angepasst. Aber auch Parameter wie Pfeilerbreite und -länge werden als Konstanten definiert. Bei deren Änderung erfolgt eine entsprechende Reaktion der Geometrieanpassung.

Für die Gründung der Widerlager und Pfeiler im vorgeführten Beispiel kommen Streifenfundamente zum Einsatz. Diese sind von den Bauteilen abhängig, die sie

unterstützen. Die Dicke des Fundamentstreifens unter einer Widerlagerwand besitzt die Dicke der Wand zusätzlich eines Versatzes auf beiden Seiten. Der Wert des Versatzes wurde als Parameter eingeführt. Ändert man diesen, passen sich die Fundamentquerschnitte aller Unterbauelemente automatisch an (Abbildung 77).

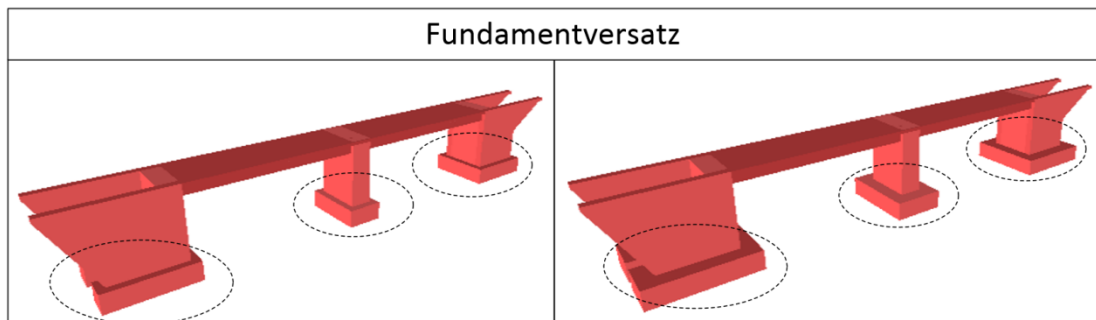


Abbildung 77: Änderung der Fundamentversatzes [31] [24]

Schließlich wird die Fundamentoberkante parametrisiert. Diese fließt in die Berechnung von der Widerlagerhöhe ein und bestimmt somit die Höhe der gesamten Brücke (Abbildung 78).

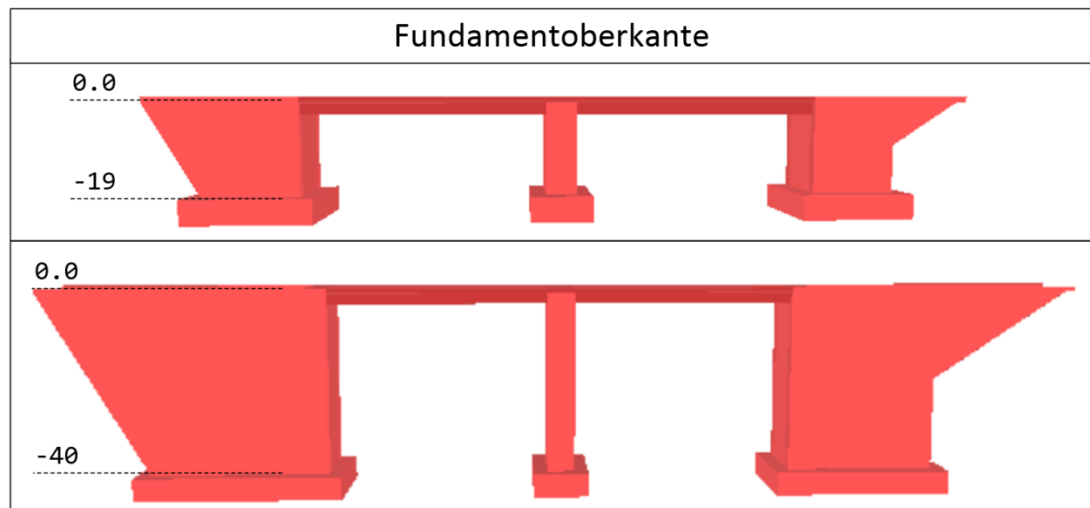


Abbildung 78: Änderung der Fundamentoberkante [31] [24]

7 Zusammenfassung und Fazit

Die Modernisierung von Planungsprozessen im Bauwesen ist eines der strategischen Ziele, die durch die Entwicklung von bauwerksmodellbasierten Ansätzen verfolgt werden [6]. Building Information Modeling und IFC zählen zu den wichtigsten Bausteinen, die zum Fortschritt innerhalb der Baubranche führen sollen. Die 3D-Modellierung von Ingenieurbauwerken befindet sich jedoch im Gegensatz zu den Hochbauten in einer frühen Entwicklungsphase und unterliegt immer noch der herkömmlich CAD-Darstellung.

Das Ziel dieser Masterarbeit war unter anderem die Analyse der bereits existierenden Ansätze zur Entwicklung der Brückenmodelle, basierend auf dem neutralen Datenaustauschformat IFC. Weiterhin sollte die praktische Anwendbarkeit dieser Methoden geprüft und anschließend weitere Vorschläge erarbeitet werden, wie diese vervollständigt werden können. In Anbetracht der fehlenden Ansätze zur Abspeicherung der Unterbauelemente einer Brücke in IFC, wurden Methoden ausgearbeitet, die eine Beschreibung der Elemente Widerlager, Pfeiler und Gründungen mit standardisierter Produktdefinitionssprache STEP ermöglichen.

Den Import von IfcBridge-Models in die Modellierungsanwendung Siemens NX ermöglicht die, von der TU München entwickelte IFC-NX-Schnittstelle, IfcBridgeTool. Infolge diverser Modifikationen, wurde einerseits die objektorientierte Struktur des Programms optimiert und andererseits der Import der zusätzlich eingefügten Unterbauelemente – Widerlager, Pfeiler und Gründungen realisiert.

Eine Steigerung der Effizienz beim Modellieren von Bauwerken kann durch die Erzeugung von parametrischen Abhängigkeiten erreicht werden. Das aktuelle IFC-Standard IFC4 erlaubt jedoch keine parametrische Abspeicherung der Bauelemente, was die Steuerung des Importes des IFC-Modells in eine andere Softwareanwendung unflexibel gestaltet [29]. Bereits existierende Verfahren, die eine Integration der parametrischen Geometriebeziehungen direkt ins IFC-Model ermöglichen, wurden analysiert und miteinander verglichen. Während die Suche nach der optimalen Eigenschaften der Brücke durch manuelle Iterationen extrem schwierig und meistens nicht erfolgsversprechend ist [32], erleichtert die Parametrik durch die

Automatisierung der Modellanpassung den Modellierungs- und vor allem durch den Änderungsvorgang. Diese Tatsache belegt auch die Brücke, die im Rahmen der Masterarbeit mit Hilfe des parametrischen IFC-Referenzenmodells beschrieben wurde. Die Änderungen eines festdefinierten Parameters bewirkte eine automatische Anpassung aller anderen abhängigen Brückenteile. Außerdem verbessert die explizite Festlegung der dimensional Abmessungen die Verständlichkeit der Geometrie des Bauwerkes und verringert somit die Wahrscheinlichkeit, dass Missverständnisse beim Modellaustausch zwischen unterschiedlichen Fachingenieuren auftreten.

Literaturverzeichnis

- [1]. **Hilbert, Martin.** The World's Technological Capacity to Store, Communicate, and Compute Information. 2011.
- [2]. **OBERMEYER Planen + Beraten, Dieter Renth.** *BIM Week & Forum - Potential und Herausforderung.* 2013.
- [3]. **Schmidt, Marcel.** *CAD mit NX 8.* 2012.
- [4]. **MAX BÖGL, Johann Bögl.** *BIM Week & Forum - Trends und Perspektiven in der Bauindustrie im 21.Jahrhundert.* 2013.
- [5]. **Chuck Eastman, Paul Teicholz.** *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors.* 2011.
- [6]. **Thomas Liebich, Erwin Gersching, Patrick MacLeamy.** *buildingSMART Strategiekonzept.* 2010.
- [7]. **Jami J.Shah, Martti Mäntylä.** *Parametric And Feature-Based CAD/CAM.* 1995.
- [8]. **buildingSMART.** *IFC2x4. Standardisiertes IFC - EXPRESS-Schema.* 2013.
- [9]. **IfcBridge, Eric Lebeque.** *IfcBridge. IFC EXPRESS Schema for Bridge structures.*
- [10]. **Nabuyoshi Yabuki, Eric Lebeque, Tomoaki Shitani, Li Zhantao.** *International Collaboration For Developing The Bridge Product Model "Ifc-Bridge".* 2006.
- [11]. **Technische Universität München, Lehrstuhl für Massivbau.** *Vorlesungsunterlagen - Massivbrücken.* 2012.
- [12]. **Gotsch, Karl.** www.karl-gotsch.de. [Online]
- [13]. **Lehrstuhl für Computergestützte Modellierung und Simulation, OpenINFRA Projekt.** *IFC-Bridge parametric extension.* München : s.n., 2011.

- [14]. **Guido Klette, Sándor Vajna.** *UNIGRAPHICS NX5 - kurz und bündig: Grundlagen für Einsteiger.* Wiesbaden : s.n., 2008.
- [15]. **Mehlhorn, Gerhard.** *Handbuch Brücken.* 2007.
- [16]. **Holst, Karl Heinz.** *Brücken aus Stahlbeton und Spannbeton.* 2004.
- [17]. **Bundesministerium für Verkehr, Bau und Stadtentwicklung, Allgemeines Rundschreiben Straßenbau Nr. 09/2013.** Richtzeichnungen für Ingenieurbauten (RIZ-ING). 2012.
- [18]. **Weidemann, Hans.** *Stahlbeton- und Spannbetonbrücken.* 1982.
- [19]. **Enzyklopedie Wikipedia.** www.wikipedia.org. [Online]
- [20]. **Google, Inc.** www.maps.google.com - Street View. [Online]
- [21]. **Plass + Engel Ingenieurgesellschaft für Bauwesen mbH.** www.plass-ingenieure.de. [Online]
- [22]. **Institut für Technische und Betriebliche Informationssysteme (ITI) Magdeburg, Eike Schallehn.** Vorlesungsunterlagen - Informatik für Ingenieure. *Geometrische Modellierungsmethoden.* 2009.
- [23]. **buildingSMART Internation Ltd.** www.buildingsmart-tech.org, www.buildingsmart.de. [Online]
- [24]. **Kuloyants, Valery.** Eigene Darstellungen.
- [25]. **Technische Universität München, Yang Ji.** IfcBridgeTool. *Programmierschnittstelle zu Siemens NX.*
- [26]. **STEP Tools, Inc., Jochen Fritz.** *SDAI C Library Reference Manual.*
- [27]. **Lehrstuhl für Computergestützte Modellierung und Simulation, Building Information Modelling.** Vorlesungsunterlagen. 2012.
- [28]. **IfcBridge-Team, TUM - Yang Ji, RDF - Peter Bonsma.** IfcBridge2x3_Parametric. *Parametrisches IFC EXPRESS-Schema.*

- [29]. **Yang Ji, Jakob Betz, Nicholas Nisbet, Peter Bonsma, Casimir Katz, Andre Borrmann.** Integration of Parametric Geometry into IFC-Bridge.
- [30]. **Yang Ji, Andre Borrmann, Mathias Obergrießer.** Towards the Exchange of Parametric Bridge Models using a Neutral Data Format.
- [31]. **Bonsma, Peter.** Parametric IFC Editor.
- [32]. **Siemens PLM Software, Web-Präsenz.** www.plm.automation.siemens.com.
[Online]
- [33]. **Minyi Wang, M.Eng.** 3D-Planung von Brückenbauwerken mit Siemens NX 7.5. München : s.n., 2012.
- [34]. **Universität Weimar, Dr.-Ing. H. Kirschke.** *Datenintegration mit STEP.*

Verwendete Softwareanwendungen

Microsoft Office

- Microsoft Word 2010
- Microsoft Visio Professional 2013
- Microsoft Excel 2010
- Microsoft PowerPoint 2010

Programmierung

- Entwicklungsumgebung: Visual Studio 2012 / C# / C++
- Versionsverwaltung: TortoiseHg
- Texteditor: Notepad++
- Dynamische Bibliothek: IfcEngine
- Frameworks: IfcGears
- NX-Programmierschnittstelle: NX Open

3D-Visualisierung

- IfcBridgeViewer 1.0
- ParametricIfcEditor
- IFC ObjectViewer
- Siemens NX 8.0

Entwickelte und erweiterte Anwendungen

- IfcBridgeTool
- IfcStepFileGenerator

Sonstige

- Cloud-Storage-Dienst: Wuala

Anhang

Anhang1: Erweitertes IfcBridge-Schema mit Unterbauelementen

Anhang2: Parametrisches IFC-Schema

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Masterarbeit selbstständig, ohne fremde Hilfe und nur unter Zuhilfenahme der angegeben Hilfsmittel angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

München, den 31.01.2014

Valeriy Kuloyants