



Diplomarbeit

**Erarbeitung eines Konzepts zur Erweiterung  
von Brückenmodellen  
um Informationen zu Semantik und Statik  
auf Basis von IFC-Dateien**

**eingereicht von**

Andreas Ellinger

**verantwortlicher Hochschullehrer**

Prof. Dr.-Ing. Raimar Scherer

**Betreuer**

Dr.-Ing. Peter Katranuschkov

Dipl.-Ing. Ngoc Trung Luu

## Kurzfassung

Schlagworte:	Building Information Modeling (BIM) Brückenbau IFC
Verfasser:	Andreas Ellinger
Titel der Diplomarbeit:	Erarbeitung eines Konzepts zur Erweiterung von Brückenmodellen um Informationen zu Semantik und Statik auf Basis von IFC-Dateien
Standort:	Technische Universität Dresden Fakultät Bauingenieurwesen Institut für Bauinformatik
Bibliographische Angaben:	2017, 94 (122) Seiten, 79 Abbildungen, 49 Tabellen

## **Selbstständigkeitserklärung**

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

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

Dresden, 12.03.2017

.....  
Andreas Ellinger

# Inhaltsverzeichnis

Abbildungsverzeichnis .....	IV
Tabellenverzeichnis .....	VII
Abkürzungsverzeichnis .....	IX
Aufgabenstellung .....	X
1 Einleitung .....	1
1.1 Ziele und Abgrenzung der Arbeit .....	1
1.2 Aufbau der Arbeit .....	2
2 State of the Art .....	3
2.1 Konventionelle Brückenplanung .....	3
2.2 BIM-orientierte Planung .....	3
2.3 Das Datenmodell IFC.....	3
2.4 IFC im Brückenbau .....	4
2.4.1 Das japanische Modell <i>YLPC-BRIDGE</i> .....	4
2.4.2 Das französische Modell <i>IFC-BRIDGE Version 1.0</i> .....	6
2.4.3 Vereinigung der beiden Modelle zu IFC-BRIDGE 2.0 .....	7
2.4.4 Vereinigtes Datenmodell IFC-Bridge (Stand 2014) .....	8
2.5 Parametrische Modellierung .....	9
2.6 Bewertung.....	9
3 Modelldaten für computergestützte Analyse von Tragwerken.....	10
3.1 Materialparameter.....	10
3.2 Stäbe .....	13
3.3 Verbindungen, Relativbewegungen und Freiheitsgrade .....	14
3.3.1 Stabendgelenke .....	14
3.3.2 Freigabe von Knoten, Linien und Flächen .....	15
3.4 Modellierung von Vorspannung .....	16
4 Brückenmodelltypen-Klassifikation .....	17
4.1 Klassifikationskriterien .....	17
4.1.1 Querschnitt des Brückendecks .....	17
4.1.2 Haupttragwerke des Überbaus .....	18
4.1.3 Material .....	22
4.1.4 Funktion .....	22
4.1.5 Lage und Exposition .....	22
4.1.6 Verbindung von Unter- und Oberbau.....	23

4.1.7	Sonderformen.....	23
5	Hilfsschema zur Generalisierung von Brückenbauwerken .....	24
5.1	Allgemeine Struktur des Hilfsschemas.....	25
5.2	Instanziierung des Hilfsschemas als IFC2x3-Datei .....	28
5.3	Ausgewählte Brückentypen .....	30
5.3.1	Balkenbrücke.....	30
5.3.2	Bogenbrücke .....	31
5.3.3	Schräggabelbrücke.....	32
5.4	Gesamtschema – Überblick.....	33
5.5	Hierarchische Struktur der Bauwerkskomponenten.....	35
5.6	Beziehungen und Eigenschaften der Komponenten .....	36
5.6.1	Komponenten des Überbaus .....	36
5.6.2	Elemente von Schrägseiltragwerken .....	42
5.6.3	Elemente von Bogentragwerken.....	44
5.6.4	Bogentragwerke im Hilfsschema .....	46
5.6.5	Elemente des Unterbaus .....	47
5.6.6	Modellierung der Unterbauelemente im Hilfsschema.....	48
5.6.7	Lager.....	49
6	Darstellung der Objekteigenschaften über <i>IfcPropertySet</i> .....	58
6.1	Brückentyp.....	59
6.2	Material und Materialeigenschaften .....	59
6.3	Mechanische Eigenschaften von stabartigen Elementen.....	61
6.3.1	Querschnitte von Stabelementen.....	62
6.4	Fahrbahntragwerk.....	62
6.5	Vorspannung des Fahrbahntragwerks .....	63
6.5.1	Spannglied als konstruktives Element .....	64
6.5.2	Darstellung der Ersatzlasten.....	68
6.6	Ausbauelemente .....	69
6.7	Schrägseiltragwerke .....	71
6.7.1	Seile .....	72
6.7.2	Vorspannung von Seilen .....	73
6.8	Bogentragwerke.....	73
6.9	Unterbau .....	74
6.10	Lager.....	75
6.10.1	Punktlager .....	75

6.10.2	Linienlager.....	82
6.11	Abhängigkeiten von Eigenschaften.....	83
7	Modelldaten in <i>IfcStructural</i> .....	84
7.1	Einführung der Domänenmodellerweiterung <i>IfcStructural</i> .....	84
7.1.1	Verknüpfung von Geometriemodell und Tragwerksmodell.....	85
7.2	Beschreibung der mechanischen Eigenschaften in <i>IfcStructural</i> .....	85
7.2.1	Material .....	85
7.2.2	Lager.....	86
7.2.3	Seile .....	89
7.2.4	Stabartige Bauteile .....	90
7.2.5	Vorspannung.....	90
8	Verifikation des Konzepts anhand einer Beispielinstantz .....	92
9	Schlussbetrachtung.....	93
9.1	Zusammenfassung .....	93
9.2	Ergebnisse und Ausblick.....	93
	Literaturverzeichnis.....	95
	Anlagen .....	98

## Abbildungsverzeichnis

Abbildung 1: Ausschnitt aus dem YLPC-Bridge Schema .....	5
Abbildung 2: Ermittelte Überschneidung von Spanngliedverlauf und Bewehrungsstab .	6
Abbildung 3: Ausschnitt aus dem vereinigten Datenmodell IFC-BRIDGE; Stand 2014..	8
Abbildung 4: Bilineare Spannungs-Dehnungs-Verteilung.....	10
Abbildung 5: Linearer Elastizitätsmodul .....	11
Abbildung 6: Dreidimensionales Modell der Querkontraktion .....	12
Abbildung 7: Freigabe von Knoten.....	16
Abbildung 8: Platten-, Plattenbalken-, Hohlkasten- und Trogquerschnitt.....	17
Abbildung 9: Haupttragwerke der Überbauten; Piktogramm.....	18
Abbildung 10: Dreiländerbrücke; Bogenbrücke mit abgehängtem Fahrbahntragwerk .	19
Abbildung 11: Golden Gate Bridge; San Francisco .....	20
Abbildung 12: Streckträger und Schrägkabel mit einer (links) und zwei Kabelebenen (rechts).....	20
Abbildung 13: Fachwerkbrücken; Bogenform (links) und Hohlkastenform (rechts).....	21
Abbildung 14: Aicherpark-Brücke (Rosenheim); Animation .....	21
Abbildung 15: Klassenhierarchie; Schema-Ausschnitt .....	25
Abbildung 16: Zuordnung der Objekteigenschaften zu Ober- und Unterklassen .....	27
Abbildung 17: Link zwischen Eigenschaftsattributen und IFC-Property-Sets.....	27
Abbildung 18: Instanziierung der Klassen des Hilfsschemas über IFC-Klassen .....	28
Abbildung 19: Beziehung zwischen Hilfsschema-Relationen und IFC-Entitäten.....	29
Abbildung 20: Beispiel-Balkenbrücke; Längsansicht .....	30
Abbildung 21: 3D-Modell der Balkenbrücke .....	31
Abbildung 22: Beispiel-Bogenbrücke; Längsansicht.....	31
Abbildung 23: CAD-Modell der Beispiel-Bogenbrücke .....	32
Abbildung 24: Beispiel-Schräggabelbrücke; Längsansicht .....	32
Abbildung 25: CAD-Modell der Beispiel-Schräggabelbrücke .....	33
Abbildung 26: Vererbungsbeziehungen zwischen den einzelnen Klassen .....	33
Abbildung 27: Überblick über das Gesamtschema; Diagramm .....	34
Abbildung 28: Hierarchische Struktur der Bauwerkskomponenten .....	35
Abbildung 29: Einteilung des Fahrbahntragwerks in Segmente und Elemente.....	36
Abbildung 30: Fahrbahntragwerke im Hilfsschema; Diagramm .....	39
Abbildung 31: Bestandteile der Brückenausrüstung; Querschnitt durch das Brückendeck .....	40

Abbildung 32: Brückenausrüstung im Hilfsschema; Diagramm .....	42
Abbildung 33: Schrägseiltragwerke im Hilfsschema; Diagramm.....	44
Abbildung 34: Aufgeständerte Bogenbrücke; steifer Bogen (links) und Stabbogen (rechts).....	45
Abbildung 35: Bogentragwerke im Hilfsschema; Diagramm .....	46
Abbildung 36: Kastenförmiges Widerlager mit Auflagerbank.....	47
Abbildung 37: Unterbau im Hilfsschema; Diagramm .....	49
Abbildung 38: Straßenbrücke mit Verformungslagern .....	49
Abbildung 39: Verformungsbilder Elastomerlager, unbewehrt und bewehrt .....	50
Abbildung 40: Bewehrtes Elastomerlager .....	50
Abbildung 41: Stählernes Rollenlager .....	51
Abbildung 42: CAD-Modell eines Topflagers.....	52
Abbildung 43: CAD-Modell eines Kalottenlagers.....	52
Abbildung 44: Verformungsbild eines Kalottenlagers; Prinzipskizze.....	53
Abbildung 45: Festes Horizontalkraftlager; Foto (links, Mitte) und Prinzipskizze (rechts).....	54
Abbildung 46: Lagerungsbedingungen als Federsteifigkeiten, 2D-Modell .....	55
Abbildung 47: Momente in den Anschlussfugen aus Horizontalkraft .....	56
Abbildung 48: Brückenlager im Hilfsschema; Diagramm .....	56
Abbildung 49: Vereinfachte Darstellung der Beziehungen zu Lagern in EXPRESS-G.	57
Abbildung 50: Schema zur Anbindung einer <i>IfcLibraryReference</i> .....	60
Abbildung 51: Beschreibung des Spanngliedverlaufs über parametrische Segmente .	66
Abbildung 52: Parametrisches Segment aus <i>RFEM</i> .....	67
Abbildung 53: Referenzpunkte zur Beschreibung der Position in z-Richtung .....	68
Abbildung 54: Separate Anbindung von Vorspannung und Seilstabeigenschaften.....	73
Abbildung 55: Punktlager mit freigegebenem Verbindungsknoten .....	76
Abbildung 56: Schachtelung der Informationen zur Beschreibung der Gelenkeigenschaft .....	82
Abbildung 57: Idealisieretes Structural-Analysis-Model inkl. Verbindungen und Lasten	84
Abbildung 58: Verbindung von <i>IfcElement</i> und <i>IfcStructuralMember</i> .....	85
Abbildung 59: Anbindung von <i>IfcMechanicalMaterialProperties</i> .....	85
Abbildung 60: Verbindung von <i>IfcStructuralMember</i> über <i>IfcStructuralConnection</i> .....	86
Abbildung 61: Anbindung zusätzlicher Gelenkeigenschaften .....	88
Abbildung 62: Beschreibung von Seilen in <i>IfcStructural</i> .....	89
Abbildung 63: Anbindung von Vorspannkraft an Seile.....	89
Abbildung 64: Darstellung von Vorspannung über <i>IfcStructuralAction</i> .....	91

Abbildung 65: Semantische Struktur der IFC-Instanz der Balkenbrücke .....	109
Abbildung 66: Hierarchische Struktur der Komponenten der Beispielinstantz .....	110
Abbildung 67: Eigenschaften der Komponenten der Beispielinstantz.....	111
Abbildung 68: Verlauf des Spannglieds im Träger der Beispielbrücke .....	112
Abbildung 69: Segment-Typ 2.....	112
Abbildung 70: Struktur des Property-Sets Vorspannung .....	113
Abbildung 71: EXPRESS-Spezifikation der abstrakten Oberklassen.....	115
Abbildung 72: EXPRESS-Spezifikation der Brückentypen und des Überbaus .....	115
Abbildung 73: EXPRESS-Spezifikation der Brückenausrüstung .....	116
Abbildung 74: EXPRESS-Spezifikation des Fahrbahntragwerks.....	117
Abbildung 75: EXPRESS-Spezifikation des Schrägkabeltragwerks .....	117
Abbildung 76: EXPRESS-Spezifikation des Bogentragwerks.....	118
Abbildung 77: EXPRESS-Spezifikation des Unterbaus .....	119
Abbildung 78: EXPRESS-Spezifikation der Brückenlager .....	119
Abbildung 79: Segmenttypen in <i>RFEM</i> .....	121

## Tabellenverzeichnis

Tabelle 1: Parameter für das isotrop-linear-elastische Materialmodell .....	11
Tabelle 2: Parameter für orthotrop-elastisches Materialmodell - 2D .....	12
Tabelle 3: Parameter für orthotrop-elastisches Materialmodell - 3D .....	13
Tabelle 4: Querschnittsparameter .....	14
Tabelle 5: Gelenkeigenschaften von Stabendgelenken.....	15
Tabelle 6: Link zwischen Referenzschema-Entitäten und IFC-Entitäten .....	28
Tabelle 7: Link zwischen Referenzschema-Relationen und IFC-Entitäten.....	29
Tabelle 8: Festlegung des Brückentyps .....	59
Tabelle 9: Materialeigenschaften .....	60
Tabelle 10: Materialkennwerte benutzerdefinierter Materialien als IfcComplexProperty.....	61
Tabelle 11: Querschnittswerte .....	62
Tabelle 12: Semantische Eigenschaften des Fahrbantragwerks .....	63
Tabelle 13: Vorspannung als Eigenschaftsset.....	63
Tabelle 14: Container für alle konstruktiven Spannglieder .....	64
Tabelle 15: Spannglied als konstruktives Element .....	64
Tabelle 16: Spannglied-Daten.....	65
Tabelle 17: Container für alle zu einem Verlauf gehörenden Segmente.....	65
Tabelle 18: Verlauf in der xy-Ebene .....	66
Tabelle 19: Anfangs/Endpunkt; Typ C.....	67
Tabelle 20: Ersatzlast .....	68
Tabelle 21: Eigenschaften der Fahrbahnausbildung .....	69
Tabelle 22: Eigenschaften der Fahrbahnübergänge .....	70
Tabelle 23: Eigenschaften der Kappen .....	70
Tabelle 24: Eigenschaften der Lärmschutzanlage.....	71
Tabelle 25: Eigenschaften von Leiteinrichtungen .....	71
Tabelle 26: Semantische Eigenschaften von Schrägseiltragwerken.....	71
Tabelle 27: Semantische Eigenschaften von Schrägseilen .....	72
Tabelle 28: Mechanische Eigenschaften von Seilen .....	72
Tabelle 29: Vorspannung von Seilen .....	73
Tabelle 30: Eigenschaften von Bogentragwerken .....	74
Tabelle 31: Semantische Eigenschaften von Punktlagern.....	75
Tabelle 32: Knotenfreigabe .....	77

Tabelle 33: Nichtlineare Gelenkeigenschaften .....	77
Tabelle 34: Nichtlineare Verschiebungseigenschaften in x-Richtung .....	78
Tabelle 35: Nichtlineare Verdrehungseigenschaften .....	79
Tabelle 36: Spezifikation nichtlinearer Federsteifigkeit.....	80
Tabelle 37: Spezifikation der maximal möglichen Verschiebung .....	80
Tabelle 38: Spezifikation von Fließen und Schlupf.....	81
Tabelle 39: Abhängigkeit möglicher Verschiebungsfreiheitsgrade vom Lagertyp .....	83
Tabelle 40: Allgemeine Materialeigenschaften .....	86
Tabelle 41: <i>IfcBoundaryNodeCondition</i> .....	87
Tabelle 42: <i>IfcFailureConnectionCondition</i> .....	88
Tabelle 43: <i>IfcSlippageConnectionCondition</i> .....	88
Tabelle 44: Auszug der Möglichen Werte von <i>ActionType</i> und <i>ActionSource</i> .....	91
Tabelle 45: Zwischenpunkt Gerade-Parabel; Typ S-P.....	120
Tabelle 46: Zwischenpunkt Parabel-Parabel; Typ P-P .....	120
Tabelle 47: <i>IfcTendon</i> .....	122
Tabelle 48: <i>IfcMechanicalConcreteMaterialProperties</i> .....	122
Tabelle 49: <i>IfcMechanicalSteelMaterialProperties</i> .....	122

## Abkürzungsverzeichnis

BIM:	Building Information Modeling
CAD:	Computer-aided design (rechnerunterstütztes Konstruieren)
DIN:	Deutsche Industrienorm
DXF:	Drawing Exchange Format (Dateiformat)
EN	Europäische Norm
ER-Modell	Entity-Relationship-Modell
EXPRESS	Eigenname; Standardisierte Datenmodellierungssprache
EXPRESS-G	EXPRESS-Graphical; Standardisierte graphische Notationsform
FE-Netz	Finite-Elemente-Netz
IAI	International Alliance for Interoperability
IFC	Industry Foundation Classes
IPE-Profil	Profilreihe; Doppel-T-Trägerprofile nach DIN 1025-5
ISO	International Organization for Standardization
KIT	Karlsruher Institut für Technologie
OPERA	Eigenname; CAD-Programm
PCP	Eigenname; Statikprogramm
PDF	Portable Document Format (Dateiformat)
RFEM	Eigenname; Finite-Elemente-Programm
RF-MAT NL	Eigenname; Produkterweiterung von RFEM
SETRA	Französische Straßenbaubehörde
STEP	Standard for the Exchange of Product model data
XML	Extensible Markup Language (Dateiformat)
YLPC-BRIDGE	Yabuki Laboratory Prestressed Concrete Bridge

## Aufgabenstellung

Name: cand. Ing. Andreas Ellinger  
Vertiefung: Konstruktiver Ingenieurbau

**Thema: Erarbeitung einer Methode zur Erweiterung von Brückenmodellen um semantische und statische Informationen auf Basis von IFC-Daten**  
(Development of a method for the extension of IFC-based Bridge Models by semantical and structural information)

### Zielsetzung:

Building Information Modeling (BIM) ist eine neue modellbasierte Arbeitsmethode für das Planen und Ausführen von Bauvorhaben. Dabei werden die 3D-Bauwerkgeometrie sowie viele weitere projektrelevante Daten in einem zentralen Modell, dem BIM-Modell, verwaltet. Daraus ergeben sich vielfältige Vorteile, wie beispielsweise eine höhere Planungssicherheit und Prozesstransparenz, reduzierte Risiken, unmittelbare Verfügbarkeit von Projektdaten etc. Für die Beschreibung solcher BIM-Modelle wurden die *Industry Foundation Classes* (IFC) als offener Standard (ISO 16739) von *buildingSMART* entwickelt. Mit ihnen lassen sich Gebäudemodelle digital beschreiben. Andere Bauwerktypen, wie beispielsweise Brücken, werden bislang noch nicht vom IFC unterstützt, lassen sich aber mithilfe der *IfcProxy*-Klasse modellieren. Diese Brückenmodelle sind jedoch lediglich Geometriemodelle, die eine sehr geringe semantische Tiefe aufweisen, weswegen die Vorteile der BIM-Methode nicht erzielt werden können.

Im Rahmen der Diplomarbeit soll eine geeignete Methode entwickelt werden, um IFC-Brückenmodellen um geeignete semantische Informationen zu erweitern ohne das bestehende IFC-Schema zu ändern. Zudem sollen statische Daten bereitgestellt werden, um die Transformation zu einem Statikmodell (Analysemodell) zu ermöglichen.

## Arbeitsumfang:

Während der Ausarbeitung sollen folgende Punkte bearbeitet werden:

1. Überblick und Bewertung des Standes der Forschung und Technik im Bereich der Brückenmodellierung mit IFC
2. Erarbeiten einer Brückenmodelltypenklassifikation
3. Erstellen von gewählten typischen Brückenmodellen mithilfe einer CAD-Software inkl. IFC-Export
4. Untersuchung der Möglichkeiten zur Realisierung von Erweiterungen von IFC-Brückenmodellen um semantische und statische Informationen unter Benutzung der *IfcProxy*-Klasse
5. Erarbeitung eines geeigneten Repräsentationskonzepts zur Realisierung der o.g. Erweiterungen von IFC-Brückenmodellen mittels EXPRESS/EXPRESS-G oder XML/OWL Hilfsschemas und Annotationen unter Berücksichtigung der *IfcStructuralAnalysis*- Domänenmodellerweiterung.
6. Manuelle Verifikation und Bewertung des entwickelten Konzepts an einem Beispiel.

Wiss. Betreuer TU Dresden: Dr.-Ing. Peter Katranuschkov  
Dipl.-Ing. Ngoc Trung Luu

ausgehändigt am: 16.09.2016

ursprünglich einzureichen am: 16.01.2017

verlängert bis zum: 16.03.2017

Prof. Dr.-Ing. Raimar Scherer  
Verantwortlicher Hochschullehrer

# 1 Einleitung

## 1.1 Ziele und Abgrenzung der Arbeit

Im Rahmen dieser Arbeit wird ein Konzept entwickelt, um geometrische CAD-Modelle von Brückenbauwerken um nicht-geometrische Informationen zu Semantik und Statik zu erweitern.

„Semantische Information“ wird dabei als Überbegriff für die hierarchische Struktur des Gesamtbauwerks, verallgemeinerte Beziehungen zwischen den einzelnen Bauwerkskomponenten sowie konkrete (nicht-mechanische) Eigenschaften der Bauteile verwendet.

Für die Statik relevante Daten werden unter der Bezeichnung „mechanische Informationen“ zusammengefasst, diese beinhalten alle nicht-geometrischen Daten, die zur Umwandlung eines Geometriemodells in ein Tragwerksmodell benötigt werden. Darunter Materialeigenschaften, Freiheitsgrade von Verbindungen sowie die Festlegung idealisierter Tragstrukturen wie Pendel- oder Seilstäbe.

Das in dieser Arbeit entwickelte Konzept verfolgt den Ansatz, die BIM-Daten um Informationen zu Brückenbauwerken zu erweitern ohne das bestehende BIM-Schema – in diesem Fall das IFC-Schema – zu verändern.

Zu diesem Zweck wird ein auf der Daten-Modellierungssprache *EXPRESS* basierendes Hilfsschema entwickelt. Die innerhalb dieses Schemas eingeführten Objekte und Beziehungen sollen mittels bestehender *IFC2x3*-Klassen instanziiert werden, es werden keine neuen IFC-Klassen im Stil einer Product-Extension eingeführt.

### Gewählter Detailierungsgrad

Bei der Modellbildung von Bauwerksmodellen werden drei Ebenen unterschieden:

- Die Makroebene, also die Bauwerksebene (Träger, Stützen, Platten etc.)
- Die Mesoebene, also die Ebene eines einzelnen Bauteils (schlaufe Bewehrung, Verankerungspunkte etc.)
- Die Mikroebene, die mechanische Details im Milli- und Mikrometerbereich erfasst (z.B. den Einfluss der Rippengeometrie eines Bewehrungsstabes auf dessen Verbundeigenschaften).

Die Modellbildung im Brückenbau erfolgt in der Regel auf der Makroebene; für bestimmte Detailbereiche wie die Verankerungspunkte von Schrägkabeln ist jedoch eine Modellierung auf der Mesoebene erforderlich (Kreuser & Purainer, 2003), (Mehlhorn & Curbach, 2014).

Im Rahmen dieser Arbeit wird nur die Modellbildung auf der Makroebene berücksichtigt, auf Verankerungspunkte von Schrägseilen, auf schlaufe Bewehrung etc. wird nicht eingegangen. Vorspannung stellt in diesem Zusammenhang einen Grenzfall dar; sie wird in dieser Arbeit der Makroebene zugeordnet und in das entworfene Brückenmodell miteinbezogen.

## Berücksichtigte Domain-Bereiche

Unabhängig vom gewählten Detaillierungsgrad kann die Modellbildung klassischer Tragwerksmodelle in vier Domain-Bereiche gegliedert werden (Mehlhorn & Curbach, 2014): Das mechanische Modell, das geometrische Modell, das Materialmodell und das Lastmodell.

Das im Folgenden entworfene Modell enthält die Informationen zu den Domain-Bereichen Geometrie, Mechanik und Materialeigenschaften. Die auf das Tragwerk einwirkenden äußeren Lasten werden nicht abgebildet.

Zu diesen drei klassischen Bereichen hinzu kommt die Beschreibung des Bauwerks auf der semantischen Ebene (siehe oben).

## 1.2 Aufbau der Arbeit

Die Arbeit gliedert sich in neun Abschnitte:

- Kapitel 2 gibt einen Überblick über die bisherigen Ansätze, semantische Brückenmodelle unter Verwendung des IFC-Schemas zu entwickeln.
- Darauf folgt in Kapitel 3 eine Zusammenfassung der nicht-geometrischen Modellparameter, die zur Beschreibung von Tragwerken benötigt werden. Zu diesem Zweck wurde das Finite-Elemente-Programm *RFEM* als Beispielprogramm herangezogen.
- Im 4. Kapitel wird zusammengefasst, in welche Kategorien sich Brückenbauwerke untergliedern lassen und welche Unterscheidungsmerkmale dabei maßgeblich sind. Aus diesen Kategorien werden im weiteren Verlauf drei Brückentypen ausgewählt, auf deren Beschreibung sich das entwickelte Hilfsschema – um den Rahmen der Arbeit nicht zu sprengen – beschränkt.
- Kapitel 5 befasst sich mit der allgemeinen Struktur des in *EXPRESS* formulierten Hilfsschemas, dessen Verlinkung mit dem *IFC2x3*-Schema sowie den im Hilfsschema eingeführten Klassen und Relationen. Zudem werden die CAD-Modelle von drei typischen Beispielmolellen der im Hilfsschema beschriebenen Brückentypen vorgestellt.
- Die Eigenschaften der im Hilfsschema eingeführten Klassen werden nicht im Schema selbst, sondern über Links zu in IFC definierten Eigenschafts-Sets beschrieben. Diese Property-Sets werden in Kapitel 6 gesammelt vorgestellt. Die Darstellung der mechanischen Eigenschaften orientiert sich dabei an der Struktur der in Kapitel 3 behandelten Modelleigenschaften.
- In Kapitel 7 wird die Verwendbarkeit der bestehenden Modellerweiterung *IfcStructural* für die Beschreibung von Brückentragwerken untersucht. Dabei werden die in *IfcStructural* darstellbaren statischen Systemeigenschaften mit den in Kapitel 6 eingeführten mechanischen Eigenschaften verglichen.
- Die Verifikation des in den Kapiteln 6 und 7 entwickelten Repräsentationskonzepts anhand einer Beispielinstantz erfolgt in Kapitel 8. Die Beispielinstantz basiert dabei auf dem IFC-Export von einem der in Kapitel 5 vorgestellten CAD-Modelle.
- Abschließend werden die Ergebnisse der Arbeit in Kapitel 9 zusammengefasst.

## 2 State of the Art

### 2.1 Konventionelle Brückenplanung

Brücken werden traditionell in erster Linie mit Hilfe von zweidimensionalen Plänen entworfen und ausgeführt. Der Einsatz von Computern beschränkt sich auf klassische CAD-Programme zur Erstellung der geometrischen Pläne sowie auf Statikprogramme zur Berechnung idealisierter Tragstrukturen.

Der Datenaustausch erfolgt hauptsächlich in Papierform bzw. mittels gängiger Dateiformaten wie DXF (Drawing Interchange File Format) oder dem ISO-Standard PDF (Ji, 2014). Umfassendere Datenmodelle, welche neben der geometrischen Repräsentation auch die semantische Struktur eines Modells beschreiben, kommen in der Brückenplanung bislang kaum zum Einsatz. Auch komplexere dreidimensionale Entwurfsmethoden, wie das parametrische Modellieren, finden in den meisten Fällen noch keine Anwendung.

Die Übergabe der in CAD-Software erstellten Modelldaten an die unterschiedlichen Berechnungsprogramme erfolgt weitestgehend von Hand, was sowohl zeitaufwendig als auch stark fehleranfällig ist.

### 2.2 BIM-orientierte Planung

Im Gegensatz zum Brückenbau kommt im klassischen Hochbau seit längerem eine modellbasierte Planungsmethode zum Einsatz, die neben der geometrischen Beschreibung des geplanten Bauwerks eine Vielzahl weiterer, nicht-geometrischer Informationen in einem zentralen Modell zusammenfasst und verwaltet: dem BIM-Modell (Building Information Modeling).

Diese Art der Planung erleichtert die Kommunikation zwischen den unterschiedlichen Projektparteien, führt zu einer erhöhten Planungssicherheit und damit zu einer größeren Kosten- und Terminalsicherheit.

Ein zentraler Gedanke dieser Planungsmethode besteht in der Idee „A change anywhere is a change everywhere“ – eine Veränderung an einem der Domain-Teilmodelle führt immer zu einer Änderung des Gesamtmodells. Auf diese Weise können durch inkonsistente Teilmodelle entstehende Planungsfehler vermieden werden.

### 2.3 Das Datenmodell IFC

*Industry Foundation Classes (IFC)* ist ein standardisiertes Datenmodell für das Bauwesen, das von der *Internationalen Allianz für Interoperabilität (IAI)* entwickelt wurde (Ji, 2014, S.50). *IFC* basiert auf dem hauptsächlich im Maschinen- und Anlagenbau gebräuchlichen Datenmodell *STEP*, von dem insbesondere die objektorientierte Datenmodellierungssprache *EXPRESS*, die graphische Darstellungsform *EXPRESS-G*, das physische Datenaustauschformat *STEP Physical File* sowie die Methoden zur Beschreibung von Geometrie übernommen wurden.

Ein entscheidender Vorteil des IFC-Datenmodells gegenüber herkömmlichen Dokumentformaten wie DXF oder PDF besteht in der Trennung von geometrischer und semantischer Information. Während geometrische Information die Form von Objekten sowie deren Lage im Raum definiert, beschreibt semantische Information die Bedeutung der einzelnen Objekte sowie die Beziehungen der Objekte zueinander.

Ein Beispiel für die semantische Strukturierung des IFC-Modells ist die Gliederung von Objekten in Ober- und Unterklassen. So ist ein Bauelement in IFC eine Unterklasse der Klasse Element; die Klasse Bauelement wiederum besitzt u. a. die Unterklassen Stütze, Träger, Wand etc. Jede Klasse erbt alle Attribute ihrer jeweiligen Oberklassen. Auf dieser, von jeglicher geometrischen Repräsentation entkoppelten semantischen Ebene ist eine sehr abstrakte und allgemeine Beschreibung von Bauwerksmodellen möglich. Diesen Modellen können weitere nicht-geometrische Informationen wie Material oder Verbindungseigenschaften angehängt werden, wodurch eine Datengrundlage für weitere Analysen und Berechnungen geschaffen wird (Ji, 2014).

## 2.4 IFC im Brückenbau

Während für den Hochbau bereits 2013 ein IFC-Datenmodell zur Beschreibung von Gebäudemodellen als ISO-Standard eingeführt wurde, existiert momentan noch kein genormtes und international anerkanntes IFC-Schema zur Beschreibung von Brückenbauwerken (Ji, 2014). Zwar entwickelten verschiedene Firmen, Länder und Organisationen unabhängig voneinander eigene Datenmodelle und Anwendungssysteme wie beispielsweise *Japan Highway Product Model* und *TransXML*. Auf Grund mangelnder Absprachen zwischen den Entwicklern ist eine Interoperabilität der einzelnen Modelle und Systeme jedoch – wenn überhaupt – nur stark eingeschränkt möglich (Yabuki, Lebeque u. a., 2006).

Erste Versuche ein auf dem international anerkannten IFC-Standard aufbauendes Produktmodell zur Beschreibung von Brückenbauwerken zu entwickeln, unternahmen im Jahr 2002 zwei Forscherteams in Japan und Frankreich unabhängig voneinander. Obwohl Grundgedanke und Herangehensweisen beider Gruppen sehr ähnlich waren, wussten beide anfänglich nichts von dem Projekt des jeweils anderen.

Nachdem beide Forscherteams voneinander erfahren hatten, entschieden sie, ihre jeweiligen Modellansätze in einem gemeinsamen Modell zusammenzuführen und auf diese Weise die Basis für ein in Zukunft international anerkanntes Standardmodell zu schaffen.

Nachfolgend werden zunächst die beiden unabhängig voneinander entwickelten Ausgangsmodelle erläutert, im Anschluss daran wird das daraus entstandene aktuelle Modell *IFC-Bridge* vorgestellt.

### 2.4.1 Das japanische Modell *YLPC-BRIDGE*

Im Jahr 2002 entwickelte ein japanisches Forscherteam unter Nobuyoshi Yabuki am *Muroran Institute of Technology* eine Produkterweiterung mit dem Namen *YLPC-BRIDGE* (*Yabuki Laboratory Prestressed Concrete Bridge*). Dabei handelte es sich um eine Erweiterung des bestehenden IFC2x-Schemas um neue Klassen und Beziehungen zur Beschreibung vorgespannter Betonbrücken.

Zu diesem Zweck führten sie die Klasse *CivilStructureElement* (sinngemäß *Ingenieurbauwerks-Element*) als neue abstrakte Unterklasse von *IfcElement* ein. Dieser Klasse untergeordnet wurden u. a. die Klassen *Rebar* (*Bewehrungsstab*), *AnchoringDevice* (*Verankerungselement*), *Void* (*Lücke/Hohlraum*), *PrestressingStrand* (*Spannglied*) sowie *Sheath* (*Hüllrohr*). Des Weiteren wurden die Klassen *ConcreteStructureElement* (sinngemäß *Betonbauwerks-Element*) sowie *SlabOfBridge* (sinngemäß *Brückendeck*) als neue Unterklassen von *IfcSpatialStructureElement* eingeführt (vgl. grau hinterlegte Klassen in Abbildung 1).

Ein wesentliches Merkmal dieses Modells besteht darin, dass Bewehrungs- und Vorspannelemente als „im Beton enthalten“ dargestellt werden. Dies geschieht, indem die Elemente *Rebar*, *AnchoringDevice*, *PrestressingStrand* etc. als Unterklasse von *IfcElement* beschrieben werden, während *ConcreteStructureElement* und *SlabOfBridge* als Unterklasse von *IfcSpatialStructureElement* definiert sind.

Die Beziehung „*IfcSpatialStructureElement* enthält *IfcElement*“ ist über die IFC-Entität *IfcRelContainedInSpatialStructure* festgelegt.

Des Weiteren wurden Property-Klassen eingeführt, die die Eigenschaften der neu eingeführten Klassen beschreiben, darunter *ConcreteProperties*, *RebarProperties* und *PrestressingStrandProperties*. Die Trennung von Objektklassen und deren Eigenschaften durch die Verwendung von Property-Klassen führt den Entwicklern zu Folge zu einer flexibleren Modellstruktur.

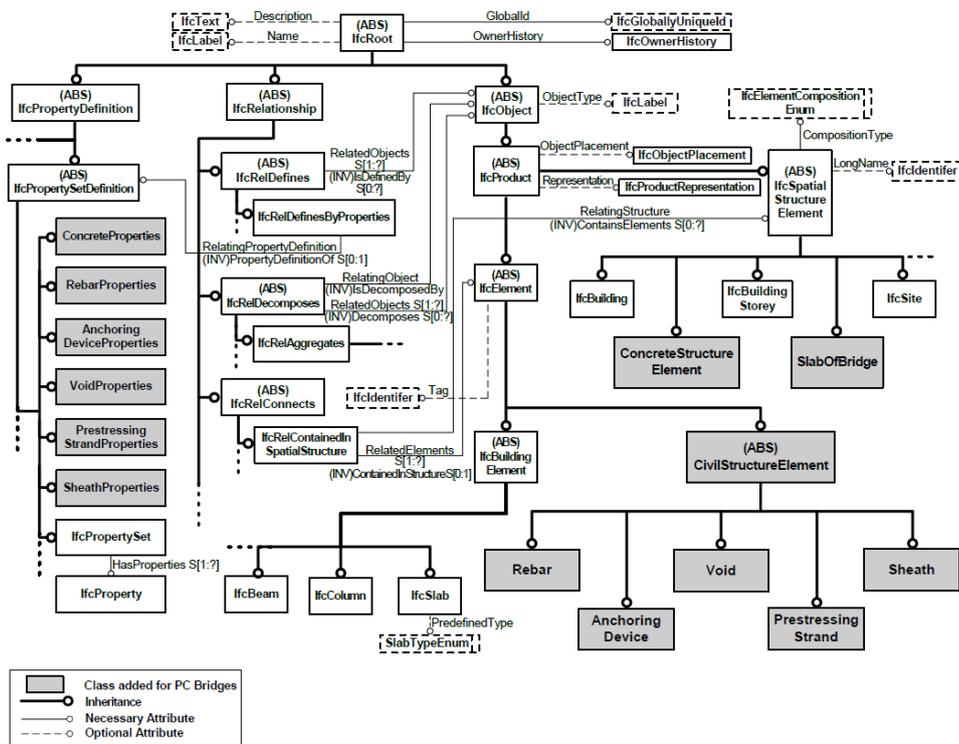


Abbildung 1: Ausschnitt aus dem YLPC-Bridge Schema

Quelle: (Corchado, Hujun Yin u. a., 2006)

Um das entwickelte Modell auf Validität und praktische Anwendbarkeit zu prüfen, integrierten sie u. a. zwei Konverter-Programme sowie einen *interference checking agent* in das Modell. Mit Hilfe des Konverter-Programms CAD2PM (CAD to Product Model) konnten Objekte aus dem YLPC-Bridge-Modell instanziiert und in Form von ifcXML-Dateien exportiert werden. Über das Konverter-Programm PC2CAD (Product Model to CAD) war es möglich, auf Basis dieser ifcXML-Dateien automatisch 3D-Modelle der Modell-Instanzen in *AutoCAD* zu erzeugen.

Mit Hilfe des *interference checking agent* konnte u. a. geprüft werden, ob sich im Modell enthaltene schlaife Bewehrungsstäbe mit dem Verlauf von Spanngliedern bzw. deren Hüllrohren überschneiden. Im Fall einer solchen „Interferenz“ wurde automatisch eine entsprechende Meldung ausgegeben und die betroffenen Bewehrungsstäbe bzw. Spannglieder markiert. Zusätzlich wurde aus der Schnittmenge der sich überschneidenden Objekte ein 3D-Objekt generiert (siehe nachfolgende Abbildung).

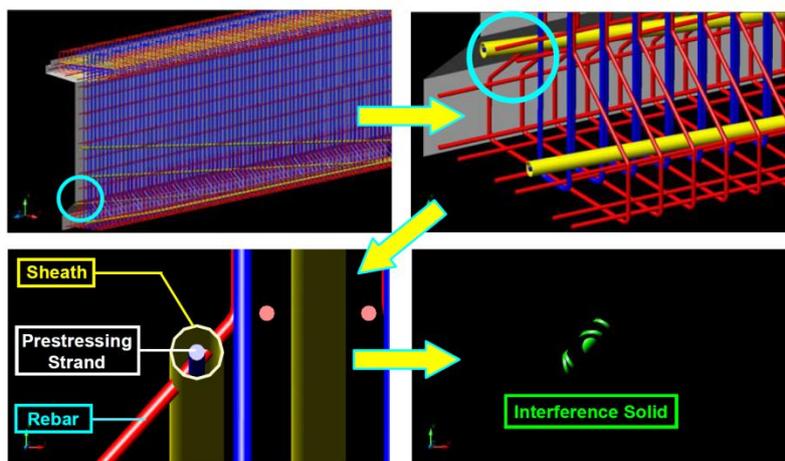


Abbildung 2: Ermittelte Überschneidung von Spanngliedverlauf und Bewehrungsstab

Quelle: (Yabuki, Lebegue u.a., 2006)

#### 2.4.2 Das französische Modell *IFC-BRIDGE Version 1.0*

Das 2002 entwickelte Modell *IFC-BRIDGE Version 1.0* basiert auf dem ursprünglich von der französischen Straßenbaubehörde SETRA in Zusammenarbeit mit mehreren Baufirmen entwickelten Datenmodell OA\_EXPRESS. Im Jahr 2000 zeigten die Ingenieure, dass das OA\_EXPRESS-Modell geeignet war, Daten zwischen dem CAD-Programm OPERA und dem Berechnungsprogramm PCP auszutauschen. Beide Programme entwickelte SETRA eigens für den Einsatz im Brückenbau. Um das Modell einem größeren Anwenderkreis zugänglich zu machen, wurde es 2002 an die Struktur des IFC-Datenmodells angepasst (Yabuki, Lebegue u. a., 2006).

Das Grundkonzept von *IFC-Bridge 1.0* entspricht weitestgehend dem japanischen Datenmodell; auch *IFC-Bridge 1.0* stellt eine Erweiterung des bestehenden IFC-Modells um neue Klassen und Beziehungen dar. So führte es u. a. die Klasse *IfcBridgeElement* als Komplementär zur bestehenden IFC-Klasse *IfcBuildingElement* ein, dieser wurden

typische Brückenelemente wie Deck, Seil oder Pylon untergeordnet. Zur Gliederung der hierarchischen Struktur des Gesamtbauwerks führten die Entwickler zusätzlich die Klassen *IfcBridge* und *IfcBridgeStructureElement* ein.

Ebenfalls analog zum japanischen Modell wurde das Brückenmodell in mehrere „prismatische Elemente“ aufgeteilt, die sich ihrerseits wiederum aus ein- oder mehreren Segmenten zusammensetzen.

Eine Besonderheit des französischen Modells liegt in der Einführung einer neuen Möglichkeit zur Geometriebeschreibung. Zusätzlich zu der in IFC2x standartmäßig vorgesehenen Geometrierepräsentation ermöglicht das Modell über seine neue Klasse *IfcBridgeSectionedSpine* den Verlauf eines prismatischen Brückenelements über mehrere Segmente, die entlang einer Referenzlinie verlaufen, darzustellen.

Zur Beschreibung dieser Referenzlinie fügten sie die in der Verkehrswegeplanung zur Beschreibung von Kurven übliche Klothoide zur Geometrie-Bibliothek des Modells hinzu (Yabuki, Lebegue u. a., 2006).

### 2.4.3 Vereinigung der beiden Modelle zu IFC-BRIDGE 2.0

Im Jahr 2004 beschlossen die japanische und die französische Entwicklergruppe ihre beiden Modelle in einem gemeinsamen Modell zusammenzuführen. Zu diesem Zweck überarbeitete das japanische Team das ursprüngliche *IFC-Bridge 1.0*-Modell sowie eine von den Franzosen weiterentwickelte Version 2.0 und legten einige Änderungsvorschläge vor. Gleichermäßen wurden einige Aspekte des japanischen Ausgangsmodell *YLPC-Bridge* auf Anraten der französischen Entwickler abgeändert.

#### Änderungen im Vergleich zum französischen Ausgangsmodell

1. Die Klasse *IfcBridgeElement* wurde um die Unterklasse *IfcBridgeElementComponent* erweitert. Dieser wiederum ordnen sich die zur Beschreibung von Bewehrung und Vorspannung nötigen Elemente aus dem japanischen Modell unter, darunter die Klassen *Rebar*, *Sheath*, *PrestressingStrand* etc.
2. Die überarbeitete französische Version des Modells enthielt ursprünglich *IfcBridgeSingularInternalPoint* als obligatorisches Attribut der Klasse *IfcBridgeElementComponent*, um Objekte wie Spannglieder mittels der in IFC-Bridge neu eingeführte Referenzlinie geometrisch zu beschreiben. Der Änderungsvorschlag der japanischen Ingenieure bestand darin, das Attribut nicht als obligatorisch, sondern als optional zu definieren. Dies ermöglichte es, die Unterklassen von *IfcBridgeElementComponent* alternativ auch über die in IFC standartmäßig vorgesehenen Methoden zur Geometrierepräsentation zu beschreiben.
3. Die Klassennamen für Bewehrungsstab und Spannglied wurden umbenannt von *Rebar* und *PrestressingStrand* zu *IfcBridgeReinforcingBar* und *IfcBridgeTendon*.
4. Während die französischen Ingenieure in der Weiterentwicklung ihres ursprünglichen Modells die Klasse *IfcBridgeElement* als Unterklasse von *IfcBuildingElement* definiert hatten, wurde sie auf den Vorschlag der Japaner hin wieder – wie in der ursprünglichen Version 1.0 – als direkte Unterklasse von *IfcElement* und somit als Komplementär zu *IfcBuildingElement* festgelegt.

## Änderungen im Vergleich zum japanischen Ausgangsmodell

1. Die im japanischen Modell enthaltenen Klassen *SlabOfBridge* und *ConcreteStructureElement* – beides Unterklassen der IFC-Klasse *IfcSpatialStructureElement* – wurden durch die Klasse *IfcBridgeStructureElement* ersetzt.
2. Das YLPC-Bridge-Modell enthielt ursprünglich eigene, neu eingeführte Klassen zur Beschreibung der Eigenschaften von Brückenelementen wie beispielsweise *RebarProperties* und *ConcreteProperties*. Um die Grundstruktur des bestehenden IFC-Modells zu erhalten, wurden diese neuen Klassen aus der Modellerweiterung entfernt. Stattdessen verwendete man die bestehende Klasse *IfcPropertySet*.

Durch die Einarbeitung der oben beschriebenen Modifikationen entstand das Produktmodell *IFC-BRIDGE 2.0*. Das Modell wird nach Aussage der Entwickler noch einmal gründlich überarbeitet, bevor es in naher Zukunft durch einen IFC-Experten in das offizielle IFC-Modell implementiert werden soll (Yabuki, Lebegue u. a., 2006).

### 2.4.4 Vereinigtes Datenmodell IFC-Bridge (Stand 2014)

Ein Ausschnitt des IFC-Bridge-Modells ist in Abbildung 3 dargestellt. Die Abbildung verdeutlicht, wo die Modellerweiterung im bestehenden IFC-Datenmodell angesiedelt ist.

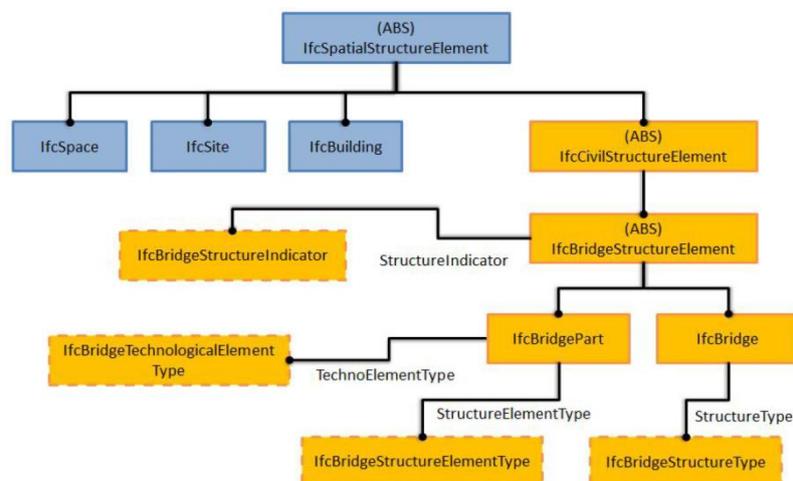


Abbildung 3: Ausschnitt aus dem vereinigten Datenmodell IFC-BRIDGE; Stand 2014

Quelle: (Ji, 2014)

Als Unterklasse von *IfcCivilStructureElement* stellt *IfcBridgeStructureElement* die oberste Element-Klasse dar, die sich konkret auf Brückenbauwerke bezieht. Über die Beziehung zu *IfcBridgeStructureIndicator* kann beispielsweise angegeben werden, ob es sich bei einem Brückenbauwerk um eine Verbundbrücke oder um eine Brücke mit Auflagerungen handelt.

In der Unterklasse *IfcBridge* wird das statische System der Gesamtbrücke genauer definiert. Über die Relation *StructureType* kann u. a. zwischen den Systemen Plattenbrücke, Rahmenbrücke, Bogenbrücke und Schrägseilbrücke ausgewählt werden.

In der Klasse *IfcBridgePart* werden die einzelnen Komponenten eines Brückenbauwerks genauer beschrieben. Zum einen wird festgelegt, um welche Art von Brückenkomponenten es sich handelt, zum anderen können den Komponenten bestimmte bautechnische Eigenschaften zugewiesen werden. So kann die Komponente „Brückendeck“ beispielsweise aus bautechnischer Sicht als „Hohlkasten“ oder als „Massivplatte“ realisiert werden.

Dieser Struktur folgend werden das statische System sowie die bautechnischen Eigenschaften des gesamten Brückenbauwerks auf semantischer Ebene beschrieben.

## 2.5 Parametrische Modellierung

Neben der semantischen Beschreibung von Brückenbauwerken lag auch die Entwicklung parametrischer Modelle, bei denen die geometrischen Abmessungen nicht fix, sondern als variable, von anderen Parametern abhängige Größen beschrieben werden, im Fokus der Entwicklungen der letzten Jahre.

Hierfür bietet die im *IFC-Bridge*-Modell eingeführte Methode, die Geometrie der Brücke bezogen auf eine gekrümmte Referenzlinie zu beschreiben, eine vielversprechende Grundlage.

Da parametrische Modellierung jedoch nicht Bestandteil dieser Arbeit ist, wird auf die Details dieser Methode sowie auf deren Probleme an dieser Stelle nicht eingegangen. Es sei lediglich gesagt, dass die parametrische Beschreibung von Brückenbauwerken auf Basis des *IFC-Bridge*-Modells zwar möglich ist, auf Grund mangelnder Möglichkeiten der im Bauwesen gebräuchlichen CAD-Programme jedoch noch nicht praktikabel umgesetzt werden kann (Lukas, Ji, Obergießer, & Borrmann, o.J. ).

## 2.6 Bewertung

Das gemeinsame Modell *IFC-Bridge* der französischen und japanischen Entwickler stellt einen vielversprechenden Ansatz dar, das IFC-Schema für den Einsatz im Brückenbau zu erweitern. Besonders die Detailmodellierung von Bewehrungselementen auf der Mesoebene und die daraus resultierende Möglichkeit, Überschneidungen von Vorspannung und schlaffer Bewehrung automatisch feststellen zu können, erscheint sehr nützlich – und das nicht nur für den Einsatz im Brückenbau.

Auch zur Beschreibung parametrischer Brückenmodelle existieren gute Ansätze, diese lassen sich auf Grund der im Bauwesen gebräuchlichen CAD-Programme jedoch – wie bereits gesagt – noch nicht praktikabel umsetzen.

Inwiefern das *IFC-Bridge*-Modell für den Einsatz im Brückenbau tatsächlich geeignet ist, lässt sich jedoch erst beurteilen, wenn es als Teil der IFC-Norm in der Praxis bei realen Projekten auf den Prüfstand gekommen ist.

### 3 Modelldaten für computergestützte Analyse von Tragwerken

Im Folgenden wird untersucht, welche Informationen zur Beschreibung eines digitalen Tragwerksmodells notwendig sind, um dessen Analyse mit Hilfe gängiger Statikprogramme zu ermöglichen.

Struktur und Benennung der Modelldaten orientieren sich dabei weitgehend an dem zu diesem Zweck herangezogenen Beispielprogramm *Dlubal RFEM 5*.

Nachfolgende Ausführungen beschränken sich auf jene Modelldaten, die im späteren Verlauf der Arbeit bei der Beschreibung von Brückenmodellen verwendet werden.

#### 3.1 Materialparameter

Materialparameter fließen in die Steifigkeit von Fläche, Querschnitten und Volumenkörpern mit ein und sind somit ein notwendiger Teil der für statische Berechnungen eines Modells benötigten Modelldaten (GmbH, Dlubal Software, 2016).

##### Materialmodelle

Welche Parameter zur Berechnung von Spannung, Verformung oder Schwingung eines Systems mit Hilfe von Statik-Software benötigt werden, hängt von dem bei der Berechnung verwendeten Materialmodell ab. Während zur Beschreibung linear-elastischer Materialmodelle die Parameter Elastizitätsmodul, Schubmodul, Querdehnzahl, spezifisches Gewicht, Wärmedehnzahl sowie ein Teilsicherheitsbeiwert ausreichen, werden für die Berechnung nach dem bilinear-elastischen Modell zusätzlich die Parameter Streckgrenze und Verfestigungsmodul benötigt (siehe Abbildung 4).

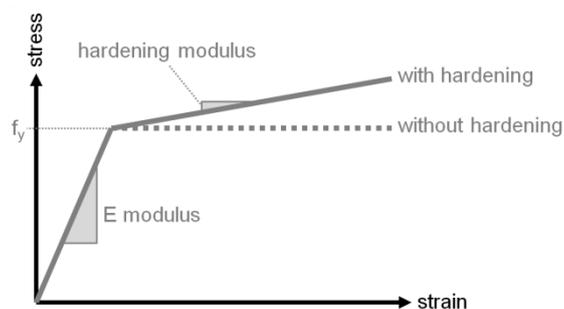


Abbildung 4: Bilineare Spannungs-Dehnungs-Verteilung

Quelle: Nemetschek SCIA

In RFEM kann zwischen mehreren Materialmodellen ausgewählt werden. Ohne das kostenpflichtige Zusatzmodul *RF-MAT NL* stehen folgende Materialmodelle zur Verfügung:

- Isotrop linear elastisch
- Orthotrop elastisch 3D
- Orthotrop elastisch 2D

Mit dem Zusatzmodul *RF-MAT NL* kann zusätzlich unter folgenden Materialmodellen ausgewählt werden:

- Isotrop nichtlinear elastisch 1D
- Isotrop plastisch 1D
- Isotrop nichtlinear elastisch 2D/3D
- Isotrop plastisch 2D/3D
- Orthotrop elastisch 2D
- Orthotrop elastisch 3D
- Isotrop thermisch-elastisch
- Isotropes Mauerwerk 2D

Für die Beschreibung der Materialmodelle des Zusatzmoduls *RF-MAT NL* werden zusätzliche Materialparameter benötigt. Im Rahmen dieser Arbeit werden jedoch nur die für die standartmäßig verfügbaren Modelle notwendigen Materialparameter berücksichtigt.

### Parameter für das isotrop-linear-elastische Materialmodell

Beim isotropen Materialmodell sind die Steifigkeitseigenschaften des Materials richtungsunabhängig. Linear-elastisch bedeutet, dass der Elastizitätsmodul, der Schubmodul und somit auch die Querdehnzahl als konstant und damit unabhängig von der Größe der Spannung bzw. Verformung eines Körpers angenommen werden (vgl. Abbildung 5). Die zu Anfang des Kapitels bereits erwähnten, für die Berechnung nach dem isotrop-linear-elastische Materialmodell notwendigen Materialparameter sind an dieser Stelle der Übersichtlichkeit halber noch einmal in Form einer Tabelle aufgeführt:

Tabelle 1: Parameter für das isotrop-linear-elastische Materialmodell

Isotrop-linear-elastisches Materialmodell		
Parameter	Symbol	Einheit
Elastizitätsmodul	E	[N/mm <sup>2</sup> ]
Schubmodul	G	[N/mm <sup>2</sup> ]
Querdehnzahl	$\nu$	[-]
Spezifisches Gewicht	$\gamma$	[kN/m <sup>3</sup> ]
Wärmedehnzahl	$\alpha$	[1/°C]
Teilsicherheitsbeiwert	$\gamma_M$	[-]

Quelle: (GmbH, Dlubal Software, 2016)

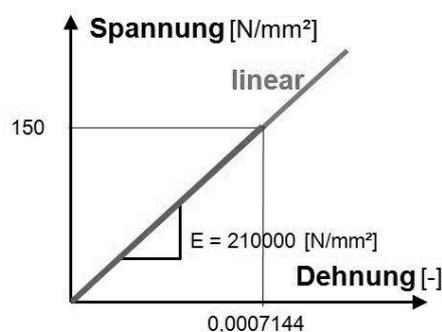


Abbildung 5: Linearer Elastizitätsmodul

Quelle: CADFEM Wiki

## Parameter für das orthotrop-elastische Materialmodell – 2D

Beim orthotrop-elastischen Materialmodell können die Steifigkeiten eines Körpers getrennt für die jeweiligen Koordinatenrichtungen angegeben werden. Dadurch lassen sich z.B. die Festigkeitseigenschaften von Holz darstellen.

Für diese Form des Materialmodells werden folgende Parameter benötigt (Tabelle 2):

Tabelle 2: Parameter für orthotrop-elastisches Materialmodell - 2D

Orthotrop-elastisches Materialmodell - 2D			
Parameter	Orientierung	Symbol	Einheit
Elastizitätsmodul	x	$E_x$	[N/mm <sup>2</sup> ]
	y	$E_y$	
Schubmodul	yx	$G_{yz}$	[N/mm <sup>2</sup> ]
	xz	$G_{xz}$	
	xy	$G_{xy}$	
Hauptquerdehnzahl	xy	$\nu_{xy}$	[-]
Nebenquerdehnzahl	yx	$\nu_{yx}$	[-]
Spezifisches Gewicht	-	$\gamma$	[kN/m <sup>3</sup> ]
Wärmedehnzahl	-	$\alpha$	[1/°C]
Teilsicherheitsbeiwert	-	$\gamma_M$	[-]

Quelle: in Anlehnung an (GmbH, Dlubal Software, 2016)

Die Querdehnzahl beschreibt das Verhältnis der relativen Längenänderung quer zur Belastungsrichtung zur relativen Längenänderung in Belastungsrichtung (siehe Abbildung 6). Zwischen Haupt- und Nebenquerdehnzahl besteht folgender Zusammenhang:

$$\frac{\nu_{yx}}{E_y} = \frac{\nu_{xy}}{E_x}$$

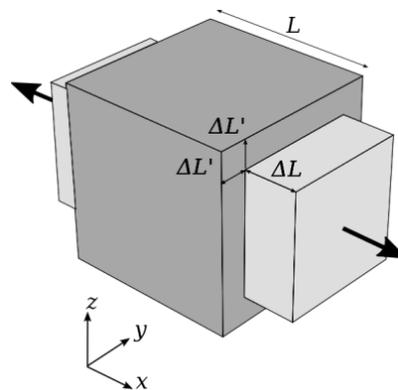


Abbildung 6: Dreidimensionales Modell der Querkontraktion

Quelle: Wikipedia

## Parameter für das orthotrop-elastische Materialmodell – 3D

Beim dreidimensionalen orthotrop-elastischen Materialmodell kommen der Elastizitätsmodul in z-Richtung und somit auch die sich auf ihn beziehenden Querdehnrzahlen hinzu. Benötigt werden folgende Parameter (siehe Tabelle 3):

Tabelle 3: Parameter für orthotrop-elastisches Materialmodell - 3D

Orthotrop-elastisches Materialmodell - 3D			
Parameter	Orientierung	Symbol	Einheit
Elastizitätsmodul	x	$E_x$	[N/mm <sup>2</sup> ]
	y	$E_y$	
	z	$E_z$	
Schubmodul	yx	$G_{yz}$	[N/mm <sup>2</sup> ]
	xz	$G_{xz}$	
	xy	$G_{xy}$	
Hauptquerdehnrzahl	yz	$\nu_{yz}$	[-]
	xz	$\nu_{xz}$	
	xy	$\nu_{xy}$	
Nebenquerdehnrzahl	zy	$\nu_{zy}$	[-]
	zx	$\nu_{zx}$	
	yx	$\nu_{yx}$	
Spezifisches Gewicht	-	$\gamma$	[kN/m <sup>3</sup> ]
Wärmedehnrzahl	-	$\alpha$	[1/°C]
Teilsicherheitsbeiwert	-	$\gamma_M$	[-]

Quelle: in Anlehnung an (GmbH, Dlubal Software, 2016)

## 3.2 Stäbe

In *Dlubal RFEM* werden Stäbe als „Eigenschaften von Linien“ verstanden und bilden eines der zentralen Elemente zur Modellierung von Tragwerken (vgl. Dlubal Software, 2016, S.140). Jedem Stab muss ein Querschnitt zugeordnet werden, worüber er seine Steifigkeit erhält. Die geometrische Lage von Stäben kann dem Stab über Eingabe von Start- und Endpunkt sowie ggf. weiteren Hilfspunkten entweder direkt zugewiesen werden oder durch Zuweisung zu einer bereits bestehenden Linie bzw. Kurve festgelegt werden. Die Informationen zum Material sind über die Daten des zugewiesenen Querschnitts bereits enthalten.

### Stabquerschnitte

In *RFEM 5* steht bereits eine große Auswahl vordefinierter Querschnitte zur Verfügung. Diese enthalten bereits alle mechanischen Informationen wie Trägheitsmomente, Querschnittsflächen, Hauptachsenneigungen etc. Man unterscheidet vollständig vordefinierte Querschnitte wie beispielsweise IPE-Profile, welche bereits mit allen geometri-

schen Abmessungen versehen sind, und parametrische Querschnitte, bei denen die geometrischen Abmessungen variabel sind und einzeln festgelegt werden müssen. Zudem ist jedem vordefinierten Querschnitt bereits standardmäßig ein Material zugeordnet; dieses kann jedoch durch ein beliebiges Material aus der Bibliothek oder auch durch ein benutzerdefiniertes Material ersetzt werden.

Neben der Verwendung von vordefinierten Querschnitten besteht die Möglichkeit, selbst Querschnitte zu definieren. Die zur Beschreibung eines benutzerdefinierten, rechteckförmigen Querschnittes notwendigen Parameter sind in Tabelle 4 zusammengefasst. Für komplexere Formen, wie beispielsweise I-Profile, ist die Angabe von deutlich mehr Parametern notwendig. Der Übersichtlichkeit halber wird an dieser Stelle jedoch das vergleichsweise einfache Beispiel eines Rechteckquerschnittes gewählt.

Tabelle 4: Querschnittsparameter

Querschnittsparameter			
Parameter		Symbol	Einheit
Trägheitsmomente	Torsion	$I_T$	[cm <sup>4</sup> ]
	Biegung	$I_y$	
		$I_z$	
Querschnittsfläche	axial	A	[cm <sup>2</sup> ]
	Schub	$A_y$	
		$A_z$	
Hauptachsenneigung		$\alpha$	[°]
Mantelfläche		$A_s$	[m <sup>2</sup> /m]

### 3.3 Verbindungen, Relativbewegungen und Freiheitsgrade

Wenn sich in einem Modell zwei getrennte Körper an einer Fläche, Linie oder in einem Punkt berühren, ohne dabei monolithisch und biegesteif miteinander verbunden zu sein, ist die Modellierung von Freiheitsgraden notwendig. Auf diese Weise können z. B. die Lager zwischen Unter- und Überbau sowie die Aufhängung von Seilen beschrieben werden. In *Diubal RFEM* stehen verschiedene Möglichkeiten zur Modellierung dieser Verbindungsstellen zur Verfügung.

#### 3.3.1 Stabendgelenke

Stabendgelenke sind Eigenschaften von Stäben und legen fest, welche Schnittkräfte über den Endknoten eines Stabes übertragen werden können und welche nicht. Sie ermöglichen die Einführung aller sechs linearen Freiheitsgrade sowie weitere nicht-lineare Gelenkeigenschaften (siehe Tabelle 5). Gelenke können nur den Stabenden zugewiesen werden, also zum Beispiel nicht in der Mitte eines Stabes angebracht werden.

Translatorische Freiheitsgrade werden in *RFEM* durch Querkraft- bzw. Normalkraftgelenke und rotatorische Freiheitsgrade durch Momentengelenke dargestellt. Jedem der

sechs linearen Freiheitsgrade ist eine Federkonstante zugeordnet, über die z. B. elastische Verbindungen modelliert werden können.

Neben den linearen Federsteifigkeiten können jedem Freiheitsgrad auch nichtlineare Eigenschaften zugeordnet werden. Diese sind entweder mit der linearen Federsteifigkeit kombinierbar, z.B. „*Federsteifigkeit  $c_{ux} = 5 \text{ kN/m}$  bei Zug, fest bei Druck*“; oder aber sie ersetzen die lineare Federsteifigkeit, wie es z. B. bei der Nichtlinearität „Diagramm“ der Fall ist. Hier wird der über die Federkonstante beschriebene, lineare Zusammenhang zwischen Verschiebung und Rückstellkraft durch einen in einem Diagramm beschriebenen, nichtlinearen Zusammenhang ersetzt.

Tabelle 5: Gelenkeigenschaften von Stabendgelenken

Gelenkeigenschaften von Stabendgelenken		
Gelenk	Federkonstante	Nichtlinearität
$u_x$	$c_{ux}$	keine; fest bei Druck; fest bei Zug; Reibung; teilweise Wirkung; Diagramm;
$u_y$	$c_{uy}$	
$u_z$	$c_{uz}$	
$\varphi_x$	$c_{\varphi x}$	keine; fest falls Moment negativ; fest falls Moment positiv; teilweise Wirkung; Diagramm;
$\varphi_y$	$c_{\varphi y}$	
$\varphi_z$	$c_{\varphi z}$	

Quelle: in Anlehnung an (GmbH, Dlubal Software, 2016)

### 3.3.2 Freigabe von Knoten, Linien und Flächen

#### Knotenfreigabe

Durch die Freigabe eines Knotens wird ein Modell an einer bestimmten Stelle entkoppelt. Dabei wird der Knoten von *RFEM* verdoppelt und das Modell am Knotenpunkt geteilt. Der ursprüngliche Knoten und seine Kopie gehören zu verschiedenen Teilen des Modells, die nun nicht mehr notwendigerweise starr miteinander verbunden sind (siehe Abbildung 7). Für die Freigabe eines Knotens müssen Gelenkeigenschaften definiert werden, die die Freiheitsgrade der Verbindungsstelle beschreiben. Es können auch nichtlineare Verbindungseigenschaften definiert werden, wie beispielsweise ein Kontaktverlust im Falle von Zugkräften.

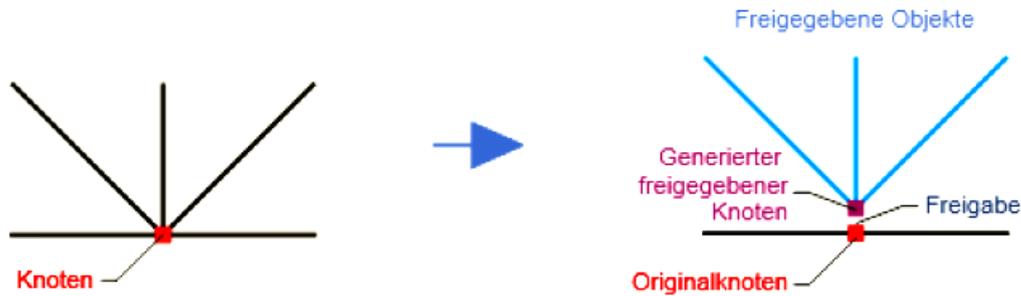


Abbildung 7: Freigabe von Knoten

Quelle: in Anlehnung an (GmbH, Dlubal Software, 2016)

Zwischen dem Originalknoten und seiner entkoppelten Kopie wird ein unsichtbarer Stab angelegt. Die Festlegung der Gelenkeigenschaften der freigegebenen Knotenverbindung erfolgt über die Zuweisung von Stabendgelenken zu diesem unsichtbaren Stab (vgl. Abschnitt „Stabendgelenke“).

### Linienfreigabe und Flächenfreigabe

Linien- und Flächenfreigabe entsprechen prinzipiell der Knotenfreigabe. Der Unterschied besteht darin, dass die durch sie entkoppelten Modellbereiche nicht punktförmig, sondern linienförmig bzw. flächig miteinander verbunden sind.

Auf Grund dieser unterschiedlichen Verbindungsgeometrie ändert sich auch die Anzahl der durch die Freigabe einführbaren Verdrehungsfreiheitsgrade. So können Linienfreigaben nur Drehungen um die Längsachse der freigegebenen Linie ermöglichen, die Anzahl insgesamt möglicher Freiheitsgrade reduziert sich somit im Vergleich zur Knotenfreigabe von sechs auf vier.

Flächenfreigaben werden an dieser Stelle nicht genauer ausgeführt, da sie in dem im Folgenden entwickelten Brückenmodell nicht verwendet werden.

## 3.4 Modellierung von Vorspannung

Die aus Vorspannung resultierenden Ersatzlasten sind der einzige Lastfall, der in dem im weiteren Verlauf der Arbeit entwickelten Brückenmodell abgebildet wird. Vorspannlasten werden dabei nicht als äußere Einwirkung, sondern als mechanische Eigenschaft des Tragsystems betrachtet.

In *RFEM* kann Vorspannung über das Zusatzmodul *RF-TENDON* modelliert und berechnet werden. Die Beschreibung der Spannglieder sowie ihre Bauteileigenschaften und die aus der Vorspannkraft resultierenden Ersatzlasten werden dabei getrennt dargestellt. (Eine Ausführliche Beschreibung dieser Darstellungsweise findet sich in Kapitel 6.5)

## 4 Brückenmodelltypen-Klassifikation

### 4.1 Klassifikationskriterien

Unterschiedliche Arten von Brücken besitzen unterschiedliche Bauelemente, Eigenschaften und Beziehungen zwischen den beteiligten Konstruktionselementen.

Um ein allgemeingültiges Datenmodell zur Beschreibung von Brückenbauwerken entwickeln zu können, muss zunächst geklärt werden, in welche Unterklassen sich der Oberbegriff „Brücke“ aufgliedern lässt und welche Klassifikationskriterien zur eindeutigen Definition eines bestimmten Brückentyps notwendig sind.

Daher wird in diesem Abschnitt behandelt, anhand welcher Merkmale eine solche Klassifikation erfolgen kann.

Mögliche Kriterien zur Einteilung von Brücken in Unterkategorien sind u.a.: Querschnittsform des Brückendecks, Form und Konstruktion des Haupttragwerks, Material, Funktion, Lage sowie Art des Verbundes zwischen Ober- und Unterbau (Weidemann, 1982), (Mehlhorn & Curbach, 2014), (Stetter, 2010).

Im Folgenden werden diese Kategorien erläutert und die ihnen zugehörigen Brückentypen kurz beschrieben. Der Fokus liegt hierbei auf den jeweiligen Unterscheidungsmerkmalen, nicht auf einer vollständigen Beschreibung von Konstruktion, Funktionsweise oder Eigenschaften der jeweiligen Unterklassen.

#### 4.1.1 Querschnitt des Brückendecks

Nachfolgend werden Brücken lediglich nach der Querschnittsgestaltung des Brückendecks – auch Fahrbahntragwerk genannt – unterschieden.

Allgemein und vom Baustoff unabhängig lassen sich drei Grundtypen von Querschnitten unterscheiden: Der Platten-, der Plattenbalken- und der Hohlkastenquerschnitt (siehe Abbildung 8). Der Trogquerschnitt stellt eine Sonderform des Plattenbalkenquerschnitts dar (Mehlhorn & Curbach, 2014).

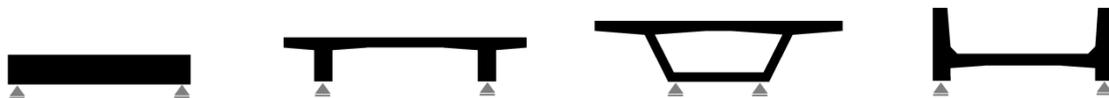


Abbildung 8: Platten-, Plattenbalken-, Hohlkasten- und Trogquerschnitt

#### Platte

Die Platte zählt zu den Flächentragwerken, der Lastabtrag erfolgt zweiachsig. Bei hinreichender Dünnwandigkeit kann ein Ebenbleiben der Querschnitte angenommen sowie die Schubverzerrung vernachlässigt werden. Durchbiegungen werden gegenüber der Plattendicke als sehr klein vorausgesetzt.

Plattenquerschnitte kommen bei geringen bis mittleren Spannweiten zum Einsatz und eignen sich besonders für schiefwinklige Fahrbahnquerungen sowie für Auffahrten und Überleitungen von Hochstraßen mit kreis- oder parabelförmiger Trassierung.

### Plattenbalken

Der Plattenbalken verbindet die Tragwirkung von Platte und Balken bzw. Scheibe, wodurch sich ein effektiveres Tragsystem als bei der Vollplatte ergibt.

Ein oder mehrere Balken, in diesem Zusammenhang oft auch als „Stege“ bezeichnet, sind schubfest mit der Platte verbunden. Die Stege dienen der Erhöhung der Querkrafttragfähigkeit, während die Platte als Untergrund für Fahrbahn, Schienen etc. dient und die angreifenden Lasten durch Plattentragwirkung auf das gesamte Tragsystem verteilt (Mehlhorn & Curbach, 2014).

### Hohlkasten

Der Hohlkasten ist eine Erweiterung des Plattenbalkens, bei dem die beiden Außenstege an ihrer Unterseite über eine weitere Platte biegesteif miteinander verbunden werden, was vor allem die Torsionssteifigkeit des Querschnitts im Vergleich zum Plattenbalken stark erhöht. Die Stege können gerade oder schräg angeordnet sein, wobei im Falle schräger Anordnung zusätzliche Druckkräfte in die untere Platte eingeleitet werden.

### Trogquerschnitt

Trogbrücken sind eine Spezialform der Plattenbalkenbrücken und weisen einen wannenförmigen Querschnitt auf, bei dem die Hauptlängsträger teilweise oder vollständig oberhalb der Fahrbahnplatte liegen. Hauptsächlich kommt der Trogquerschnitt bei Kanalbrücken zur Anwendung, bei denen das wannenförmige Brückendeck mit Wasser geflutet wird und somit z. B. als Fahrrinne für Schiffe dienen kann.

## 4.1.2 Haupttragwerke des Überbaus

Die Einteilung nach Art und Konstruktion des Haupttragwerks des Überbaus bildet im Hinblick auf das zugrunde liegende statische System und die an der jeweiligen Konstruktion beteiligten Elemente das aussagekräftigste Unterscheidungskriterium.

Man unterscheidet Balkenbrücke, Rahmenbrücke, Bogenbrücke, Hängebrücke, Schrägkabelbrücke, Fachwerkbrücke, Pendelpfeilerbrücke, Extradosed-Brücke und Spannbandbrücke (siehe folgende Abbildung).

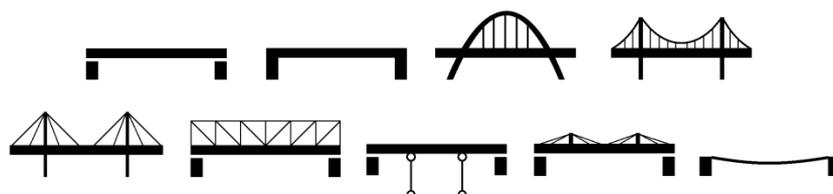


Abbildung 9: Haupttragwerke der Überbauten; Piktogramm

## **Balkenbrücke, Rahmenbrücke und Pendelpfeilerbrücke**

Diese Konstruktionsformen sind dadurch gekennzeichnet, dass sie neben dem Fahrbahntragwerk keine zusätzliche Tragstruktur wie Bogen- oder Seiltragwerk besitzen. Dabei kann das Fahrbahntragwerk neben den Widerlagern auf einem oder mehreren Stützpfeilern aufliegen.

Der Unterschied zwischen Balken- und Rahmenbrücken besteht darin, dass der Überbau im Fall von Balkenbrücken über Lager beweglich mit dem Unterbau verbunden ist, während Unter- und Überbau im Fall von Rahmenbrücken monolithisch und biegesteif miteinander verbunden sind (vgl. Abschnitt 4.1.6).

Pendelpfeilerbrücken können als eine Sonderform der mehrfeldrigen Balkenbrücken betrachtet werden, bei der die Stützpfeiler sowohl mit ihrem Fundament als auch mit dem Fahrbahntragwerk gelenkig verbunden sind. Auf diese Weise werden weder Horizontalkräfte noch Biegemomente in die Pfeiler bzw. Stützen eingeleitet.

## **Bogenbrücke**

Bei Bogenbrücken wird das Fahrbahntragwerk zwischen zwei Auflagerpunkten durch eine bogenförmige Tragstruktur zusätzlich gestützt, wodurch größere lichte Spannweiten erzielt werden können. Der Bogen kann dabei entweder direkt mit dem Fahrbahntragwerk verbunden sein oder indirekt über Aufhängungen bzw. Aufständerungen.



Abbildung 10: Dreiländerbrücke; Bogenbrücke mit abgehängtem Fahrbahntragwerk

Quelle: (baulinks.de, 2007)

Bogenbrücken können ihrerseits wiederum in mehrere Kategorien untergliedert werden. Ein Kriterium stellt dabei die Lage des Fahrbahntragwerks in Bezug auf den Bogen dar. Man unterscheidet zwischen obenliegender Fahrbahn, bei der das Brückendeck mittels Stützen auf dem Bogen „aufgeständert“ ist, und untenliegender Fahrbahn, bei der das Deck über Seile oder Stahlstreben am Bogen „aufgehängt“ wird. Mischformen, bei denen die Fahrbahn teils oberhalb, teils unterhalb des Bogens verläuft, sind ebenfalls möglich.

Ein weiteres Kriterium besteht in der Art der Ableitung der Horizontalkräfte aus dem Bogen. Hier wird zwischen „echten Bogenbrücken“ und „unechten Bogenbrücken“ unterschieden. Bei der „echten Bogenbrücke“ werden die Druckkräfte aus dem Bogen allein von den Widerlagern, in diesem Zusammenhang auch als „Kämpferstützen“ be-

zeichnet, aufgenommen. Im Falle der „unechten Bogenbrücke“ wird ein Teil der Horizontalkräfte aus dem Bogen ans Fahrbahntragwerk abgegeben. Das Fahrbahntragwerk fungiert also als eine Art Zugband.

Des Weiteren unterscheidet man Systeme mit steifen Bögen und Systeme mit sogenannten Stabbögen (vgl. hierzu Abschnitt 5.6.3).

## Hängebrücken

Bei Hängebrücken wird der Fahrbahnträger an senkrechten Seilen aufgehängt, die ihrerseits an zwischen den Pylonen gespannten Tragseilen befestigt sind. Das Tragsystem ähnelt dem einer Bogenbrücke mit aufgehängter Fahrbahn.



Abbildung 11: Golden Gate Bridge; San Francisco

Quelle: (kid-lit-reviews.com, o.J.)

## Schräggabelbrücken

Bei Schräggabelbrücken – oftmals auch Schrägseilbrücken genannt – ist das Fahrbahntragwerk über Kabel direkt an einem oder mehreren Pylonen aufgehängt. Der Lastabtrag erfolgt also nicht, wie bei der Hängebrücke, indirekt über ein zwischen zwei Pylonen gespanntes Tragseil. Daraus ergibt sich automatisch ein schräger, also in der Vertikalebene geneigter Verlauf der einzelnen Kabel.

Durch diesen geneigten Verlauf der Tragkabel werden Drucknormalkräfte ins Fahrbahntragwerk eingeleitet. Jene Teile des Fahrbahntragwerks, die diese Drucknormalkräfte aufnehmen, werden daher bei Schräggabelbrücken auch als Streckträger bezeichnet (beispielsweise Längsträger oder Hohlkastensysteme).

Man unterscheidet Schräggabelbrücken mit einer oder mehreren Kabelebenen. Diese Unterscheidung ist deshalb von Bedeutung, da Brücken mit einer Kabelebene ein torsionssteifes Fahrbahntragwerk erfordern. Bei Brücken mit zwei oder mehr Kabelebenen besitzt das Gesamtsystem eine wesentlich höhere räumliche Steifigkeit, weshalb auch Fahrbahntragwerke mit torsionsweichen Querschnitten möglich sind (siehe Abbildung 12).

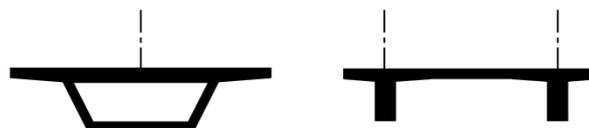


Abbildung 12: Streckträger und Schräggabel mit einer (links) und zwei Kabelebenen (rechts)

## Fachwerkbrücken

Die Fachwerkbrücke gleicht von ihrer Tragstruktur her am ehesten einer Balkenbrücke mit Hohlkasten- oder Plattenbalkenquerschnitt. Die Querschnittsbestandteile „Platte“ und „Balken“ sind jedoch nicht massiv, sondern bestehen aus vielen einzelnen Streben, die ein Netz aus dreieckigen Strukturen bilden – das Fachwerk. Die einzelnen Streben werden dabei vorwiegend auf Druck bzw. Zug beansprucht und bestehen in aller Regel aus Stahl oder Holz.

Häufig ähneln Fachwerkbrücken auch der sogenannten „unechten Bogenbrücke“, wobei der Bogen anstelle von Seilen oder Pendelstützen über fachwerkartig angeordnete Verstreben mit dem Fahrbahntragwerk verbunden ist.



Abbildung 13: Fachwerkbrücken; Bogenform (links) und Hohlkastenform (rechts)

Quellen: links: (archiexpo.de, o.J.); rechts: (structurae.de, o.J.)

## Extradosed-Brücken

Extradosed-Brücken kombinieren die Tragsysteme einer vorgespannten Hohlkastenbrücke und einer Schrägseilbrücke. Die Pylone sind im Vergleich zur Schrägkabelbrücke wesentlich niedriger, die Schrägkabel verlaufen daher viel flacher (siehe Abbildung 14). Die Pylone werden dabei nicht, wie bei der Schrägkabelbrücke, als separate, vom Brückendeck getrennte Komponenten behandelt. Stattdessen betrachtet man sie als eine spezielle Form von externer Vorspannung und somit als einen Teil des Fahrbahntragwerks.



Abbildung 14: Aicherpark-Brücke (Rosenheim); Animation

Quelle: (mangfall24.de, o.J.)

## **Spannbandbrücken**

Das Tragsystem einer Spannbandbrücke entspricht dem eines flach gespannten Seils (Mehlhorn & Curbach, 2014). Dabei werden mehrere Spannglieder zwischen zwei Widerlager gespannt und in ein relativ dünnes Betonband eingebettet. Der Beton übernimmt dabei keine primäre Tragfunktion. Er dient lediglich zur Versteifung bzw. im Falle schwach durchhängender Spannbandbrücken auch gleichzeitig als Fahrbahn. Bei stärker durchhängenden Spanngliedern wird die Fahrbahn – ähnlich wie bei Bogenbrücken mit oberliegender Fahrbahn – auf dem Spannband aufgeständert.

Bei Mehrfeldbrücken werden die Stützpfiler in aller Regel als Pendelstützen ausgeführt.

### **4.1.3 Material**

Ein weiteres Klassifikationskriterium stellt das Material dar, aus dem die Tragstrukturen eines Brückenbauwerks vorwiegend bestehen. Man unterscheidet Stein-, Holz-, Stahl- und Stahlbetonbrücken sowie Brücken in Verbundbauweise.

Verbundbrücken bestehen klassischerweise aus Stahlbeton- und Stahlkomponenten, wobei die Querschnitte der einzelnen Komponenten räumlich klar voneinander getrennt sind. (Stahlbeton wird in diesem Zusammenhang als eigenständige Materialklasse behandelt.) Weitere, seltener zur Anwendung kommende, Verbundbauweisen sind der Stahl-Holz-Verbund oder der Holz-Beton-Verbund (Bletz & Bathon, 2008).

Eine relativ neue, hinsichtlich des Tragverhaltens und der Dauerhaftigkeit vielversprechende, Bauweise bietet der Einsatz von Karbonbeton im Brückenbau (Kempkens, 2016).

### **4.1.4 Funktion**

Brücken dienen stets dem Zweck, eine kontinuierliche Fortsetzung einer Verbindung über ein Hindernis hinweg herzustellen. Bei der Klassifikation nach Funktion werden Brückentypen danach unterschieden, welche Funktion die durch die Brücke fortgesetzte Verbindungsstrecke erfüllt. Eine Brücke, die einen Fußgängerweg über einen Fluss führt, wird als Fußgängerbrücke bezeichnet.

Geläufige Brückentypen sind demnach Straßenbrücken, Eisenbahnbrücken, Fußgängerbrücken, Kanalbrücken und Wildbrücken. Im Anlagenbau kommen zudem Förderband- und Leitungsbrücken zum Einsatz. Im militärischen Bereich werden Pionierbrücken zur behelfsmäßigen Überwindung von Hindernissen eingesetzt.

### **4.1.5 Lage und Exposition**

Die geografische und topologische Lage einer Brücke kann ebenfalls als Klassifikationskriterium gewählt werden. Entscheidend für das Design der Brückenkonstruktion sind dabei die aus den örtlichen Gegebenheiten resultierenden Expositionsklassen. Daher wurden im Rahmen dieser Arbeit die drei Kategorien „Festland“, „Süßwasser“ und „Salzwasser“ eingeführt. Die Expositionsklassen sind für die einzelnen Brücken-

komponenten unterschiedlich und werden daher nicht der Brücke als Gesamtbauwerk zugeordnet.

#### **4.1.6 Verbindung von Unter- und Oberbau**

Je nach Art des Verbunds zwischen Fahrbahntragwerk und Unterbau unterscheidet man zwischen Balken- und Rahmenbrücken sowie semi-integralen Brücken (Mehlhorn & Curbach, 2014):

##### **Aufgelagerte Brücken**

Bei dieser Konstruktionsform sind Unter- und Überbau über Lager beweglich miteinander verbunden. Dadurch werden Zwangsspannungen im Fahrbahntragwerk und an den Verbindungspunkten zum Unterbau verhindert, wie sie beispielsweise durch Temperaturveränderung oder durch von bremsenden Fahrzeugen in die Fahrbahn eingeleitete Schubkräfte entstehen können.

##### **Rahmenbrücken**

Diese Bauart besitzt weder Fugen noch Lager, die eine Verschieblichkeit des Brückendecks gegenüber dem Unterbau ermöglichen und somit Zwangsspannungen abbauen könnten.

##### **Semi-integrale Brücken**

Semi-integrale Brücken stellen eine Mischform aus den beiden oben genannten Typen dar, bei der Deck und Unterbau nicht an allen Verbindungspunkten monolithisch verbunden sind oder bei der das Deck Bewegungsfugen aufweist.

#### **4.1.7 Sonderformen**

Besondere Brückentypen wie Schwimmbrücken und bewegliche Brücken lassen sich in keine gängigen Kategorien einordnen. Sie bestehen zum Teil aus gänzlich anderen Bauteilen als die zuvor beschriebenen Brückentypen und werden in dieser Arbeit nicht weiter thematisiert.

## 5 Hilfsschema zur Generalisierung von Brückenbauwerken

Im Folgenden wird ein Hilfsschema entwickelt, das Brückenbauwerke in Form eines Entity-Relationship-Modells beschreibt. Das Schema lässt sich unter Verwendung bestehender IFC2x3-Klassen instanziiieren; es stellt keine Erweiterung des IFC-Schemas dar.

### Brücken als Entity-Relationship-Modell

Ein Entity-Relationship-Modell (ER-Modell) bildet einen für ein bestimmtes Problem relevanten Ausschnitt der Realität ab. Es besteht aus Entitäten und Beziehungen zwischen den Entitäten. Mit Hilfe der Entitäten werden Objekte der Realität dargestellt, sie beschreiben jedoch keine Individuen, sondern Gruppen von Individuen mit gleichen Eigenschaften (Anderl & Trippner, 2000).

Um einen Sachverhalt in Form eines ER-Modells abbilden zu können, muss dieser zuerst in einzelne, klar voneinander abgegrenzte und eindeutig unterscheidbare Entitäten gegliedert werden. Diese können anschließend mit Relationen zueinander in Beziehung gesetzt werden.

Um ein ER-Modell zur Beschreibung von Brückenkonstruktionen erstellen zu können, muss man also zunächst untersuchen, in welche Komponenten sich Brückenkonstruktionen aufgliedern lassen, wie diese zueinander in Beziehung stehen und welche Eigenschaften die einzelnen Komponenten besitzen.

### Gewählter Modellierungsstil

Grundsätzlich unterscheidet man zwischen dem abhängigkeitsgetriebenen Modellierungsstil und dem informationsgetriebenen Modellierungsstil.

Der grundlegende Unterschied der beiden Stile besteht darin, dass bei der abhängigkeitsgetriebenen Modellierung die Relationen vom beschreibenden Objekt ausgehend hin zu dessen Eigenschaften gerichtet sind, während sie bei der informationsgetriebenen Modellierung genau anderes herum verlaufen, also von der Eigenschaft hin zum Objekt.

Die abhängigkeitsgetriebene Modellierung hat den Vorteil, dass sie sowohl übersichtlich als auch eindeutig ist, während die informationsgetriebene Modellierung sich durch eine hohe Flexibilität des Modells auszeichnet, dafür jedoch komplexer und unübersichtlicher erscheint.

Bei der Modellierung des Hilfsschemas wurde ein abhängigkeitsgetriebener Ansatz gewählt, da das Schema eine möglichst eindeutige Definition der drei Brückentypen liefern soll. Eine hohe Flexibilität des Modells wäre daher kontraproduktiv.

## Aufbau des Kapitels – Entwicklung des Hilfsschemas

Nach der Beschreibung der allgemeinen Struktur des Hilfsschemas folgt eine Übersicht des entwickelten Gesamtschemas in Form eines EXPRESS-G-Diagramms.

Da eine Beschreibung des Schemas ausgehend vom Gesamtschema zu unübersichtlich wäre, wird es im darauf folgenden Abschnitt ausgehend von den einzelnen Komponenten seiner Teilbereiche entwickelt.

In den einzelnen Abschnitten werden zunächst die realen Brückenkomponenten und ihre Eigenschaften erläutert, darauf folgt eine Verallgemeinerung der Elemente und ihrer wechselseitigen Beziehungen.

Die EXPRESS-Spezifikationen der eingeführten Klassen finden sich in Anlage 3.

### 5.1 Allgemeine Struktur des Hilfsschemas

#### Eingeführte Klassen und Klassenhierarchie

Als oberste Klasse wird die abstrakte Superklasse *Entität* eingeführt. Sie bildet das Pendant zur IFC-Klasse *IfcRoot*. Ihr untergeordnet sind die beiden abstrakten Superklassen *Objekt* und *Eigenschaft*.

Die Klasse *Objekt* gliedert sich ihrerseits in die abstrakten Klassen *Element* und *Komponente*, wobei Elemente nicht weiter untergliederte Körper repräsentieren (beispielsweise *Seil*), während Komponenten aus einem oder mehreren Elementen und/oder Komponenten zusammengesetzt sein können (z.B. *Seiltragwerk*).

Alle konkreten, instanzierbaren Eigenschaften sind der abstrakten Klasse *Eigenschaft* untergeordnet, so beispielsweise die Klasse *Material*. Abbildung 15 verdeutlicht die Klassenhierarchie an einem Ausschnitt des Gesamtschemas.

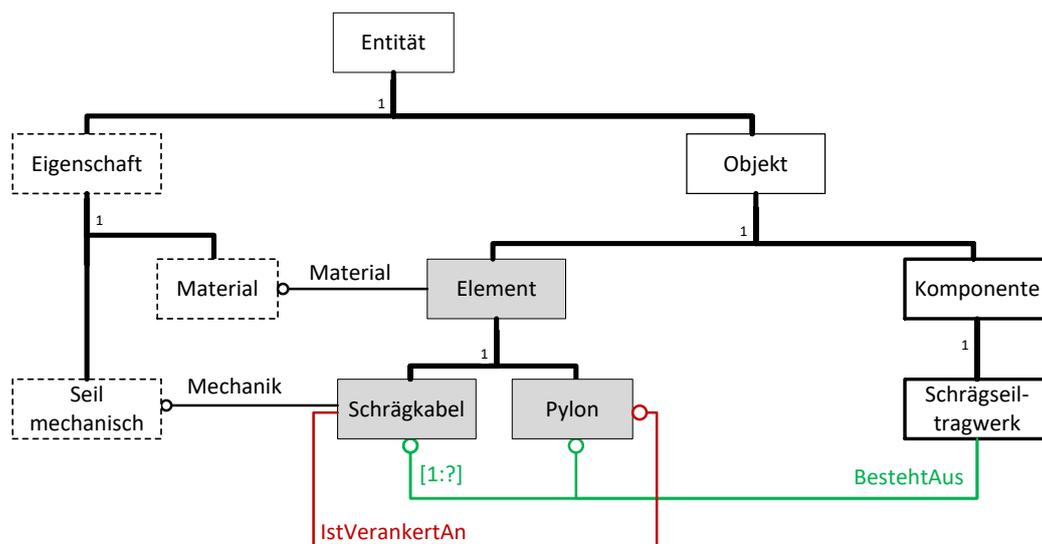


Abbildung 15: Klassenhierarchie; Schema-Ausschnitt

## Eingeführte Relationen

Das Hilfsschema enthält fünf Grundtypen von Relationen. Der erste Typ beschreibt eine „Das-Ganze-und-seine-Teile“-Beziehung. Beziehungen dieses Typs haben die Struktur  $A \text{ BestehtAus} B: B$  und sind in den EXPRESS-G Diagrammen grün dargestellt. Der zweite Typ legt fest, dass zwei Objekte räumlich miteinander verbunden sind und beschreibt, welche Rollen die verbundenen Objekte beim Lastabtrag spielen. Beziehungen dieses Typs können die Formen  $A \text{ Trägt} B: B$ ;  $A \text{ Stützt} B: B$ ; oder  $A \text{ RuhtAuf} B: B$  annehmen und sind rot dargestellt. Die Beziehung *Trägt* bedeutet dabei, dass ein Objekt alle Lasten aus dem verbundenen Objekt aufnimmt, während bei der Beziehung *Stützt* nur ein Teil der Lasten aus dem gestützten Objekt aufgenommen wird.

Typ drei ähnelt Typ zwei, diese Beziehung besagt, dass zwei Objekte räumlich miteinander verbunden sind und ihre Verbindung Gelenkeigenschaften besitzt, also nicht starr ist. Die Struktur entspricht  $A \text{ BeweglichVerbundenMit} B: B$ , dargestellt ist sie violett.

Der vierte Typ verbindet Objekte mit den zugehörigen Objekteigenschaften. Er hat die Struktur  $A \text{ Eigenschaft} B: B$  und ist schwarz dargestellt.

## Besonderheiten bei der Darstellung in EXPRESS-G

Die im Hilfsschema eingeführten Klassen werden in der Datenmodellierungssprache *EXPRESS* definiert; die entsprechenden *EXPRESS*-Spezifikationen enthalten alle direkten und inversen Relationen sowie die zugehörigen Kardinalitäten (siehe Anlage 3). Wie sich die in einem Abschnitt erläuterten Komponenten in den Gesamtkontext des Hilfsschemas einfügen, wird am Ende der jeweiligen Abschnitte anhand von *EXPRESS-G*-Diagrammen verdeutlicht. Um die Diagramme übersichtlich zu halten, werden die Bezeichnungen der Relationen dort in abgekürzter Form angegeben. So wird an Stelle der Beziehung *Schrägseiltragwerk BestehtAusPylon: Pylon* nur die verkürzte Form *Schrägseiltragwerk BestehtAus: Pylon* dargestellt. Darüber hinaus zeigen die Diagramme keine inversen Relationen.

Wenn für eine Beziehung keine Kardinalität angegeben ist, gilt für obligatorische Relationen immer die Kardinalität „genau eins“, für optionale Relationen gilt „Null oder Eins“. Die gestrichelte Umrandung der Eigenschaftsklassen dient zur besseren optischen Unterscheidbarkeit von den Objektklassen, es handelt sich dabei nicht um das in *EXPRESS-G* gebräuchliche Symbol für einen „Defined-Typ“.

Um die Diagramme übersichtlich zu halten, sind nur jene Eigenschaftsattribute grafisch dargestellt, die eine besondere statische Systemeigenschaft beschreiben.

## Zuordnung der Eigenschaften

Abbildung 16 zeigt am Beispiel der Klasse *Schräggabel*, wie die Objekteigenschaften innerhalb des Hilfsschemas den einzelnen Über- und Unterklassen zugeordnet werden.

Die Eigenschaft *Material* wird über die abstrakte Superklasse *Element* referenziert, semantische und mechanische Eigenschaften einer konkreten Unterklasse von *Element* werden direkt in der jeweiligen Unterklasse definiert (in nachfolgendem Beispiel in der Klasse *Schräggabel*).

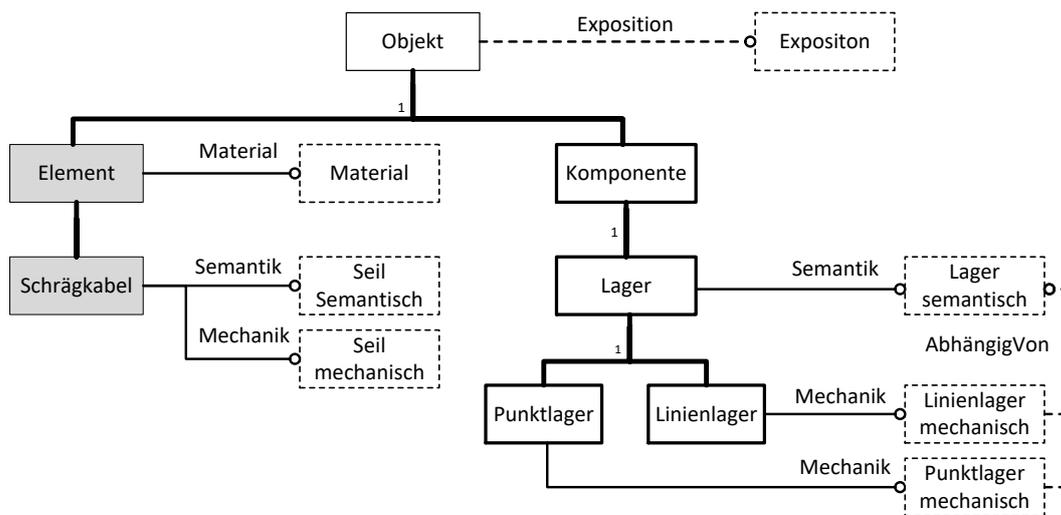


Abbildung 16: Zuordnung der Objekteigenschaften zu Ober- und Unterklassen

Die Eigenschaften der einzelnen Komponenten werden innerhalb des Hilfsschemas nicht explizit definiert. Stattdessen erhalten die Entitäten ein Attribut, über das sie mit einem, ihre jeweiligen Eigenschaften beschreibenden, *IfcPropertySet* verlinkt werden. So verweist die Eigenschaftsklasse *Seil\_semantisch* auf ein gleichnamiges *IfcPropertySet* (siehe Abbildung 17).

Die zu diesem Zweck definierten Property-Sets sind in Kapitel 6 ausführlich beschrieben. Diese Aufteilung wurde gewählt, um unnötige Wiederholungen zu vermeiden, da sich eine allgemeine Zusammenfassung der Eigenschaften in Tabellenform nur geringfügig von der Darstellung in Form eines *IfcPropertySet* unterscheiden würde.

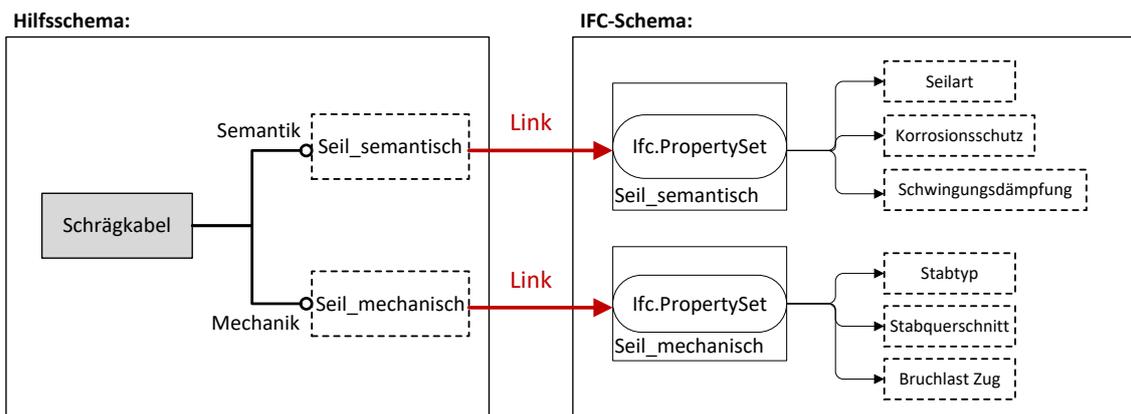


Abbildung 17: Link zwischen Eigenschaftsattributen und IFC-Property-Sets

## 5.2 Instanziierung des Hilfsschemas als IFC2x3-Datei

### Instanziierung der Entitäten

Alle im Hilfsschema enthaltenen Entitäten können mittels bestehender IFC-Klassen instanziiert werden. Welche Entitäten des Hilfsschemas über welche IFC-Klassen instanziiert werden, ist in der nachfolgenden Tabelle zusammengefasst.

Tabelle 6: Link zwischen Referenzschema-Entitäten und IFC-Entitäten

Hilfsschema-Entität	IFC-Schema-Entität
Element	IfcBuildingElementProxy
Komponente	IfcProxy
Eigenschaft	IfcPropertySet

### Verknüpfung der Entitäten des Hilfsschemas mit ihren IFC-Instanzen

Im Fall von Objekten und Elementen kann die Verknüpfung über das Attribut *ObjectType* der Klassen *IfcBuildingElementProxy* bzw. *IfcProxy* erfolgen. Das Attribut nimmt dabei den Namen der entsprechenden Entität des Hilfsschemas an. So wird ein Objekt vom Typ *Pylon* aus dem Hilfsschema über ein *IfcBuildingElementProxy* instanziiert, dessen Attribut *ObjectType* den Wert „Pylon“ annimmt.

*IfcPropertySets* werden über ihr Namensattribut mit der zugehörigen Eigenschaft aus dem Hilfsschema verknüpft. Nachfolgende Abbildung verdeutlicht dieses Prinzip am Beispiel der Bestandteile eines Seiltragwerks.

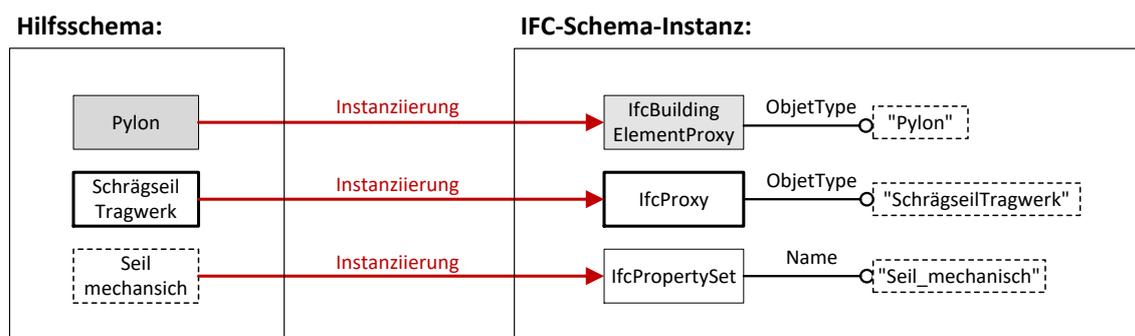


Abbildung 18: Instanziierung der Klassen des Hilfsschemas über IFC-Klassen

## Instanziierung der Relationen

Die im Hilfsschema beschriebenen Relationen werden in IFC über sogenannte relationsbildende Entitäten dargestellt, also über Unterklassen von *IfcRelationship*. Nachfolgende Tabelle fasst die Beziehungen zwischen den Hilfsschema-Relationen und den IFC-Entitäten zusammen.

Tabelle 7: Link zwischen Referenzschema-Relationen und IFC-Entitäten

Hilfsschema-Relation	IFC-Schema-Entität
BestehtAus	IfcRelAggregates
Trägt/RuhtAuf/Stützt etc.	(optional) IfcRelConnectsElements
BeweglichVerbundenMit	IfcRelConnectsElements
Eigenschaft	IfcRelDefinesByProperties

Relationen vom Typ *A BestehtAusB: B* können über die IFC-Entität *IfcRelAggregates* abgebildet werden. Über deren Beziehungen *RelatingObject* und *RelatedObjects* kann die Richtung der Relation aus dem Hilfsschema wie folgt im IFC-Schema dargestellt werden:

Die Hilfsschema-Beziehung *A BestehtAusB: B* wird innerhalb des IFC-Schemas beschrieben als „*IfcRelAggregatesRelatingObject:A* und *RelatedObjects:B*“. Auf diese Weise bleibt die Richtung der „Das-Ganze-und-seine-Teile-Beziehung“ erhalten (siehe Abbildung 19).

Die Abbildung der vier anderen oben beschriebenen Beziehungstypen erfolgt nach dem gleichen Prinzip, wobei die Klasse *IfcRelConnectsElements* noch zusätzlich das optionale Attribut *ConnectionGeometry* zur Beschreibung der Verbindungsgeometrie besitzt.

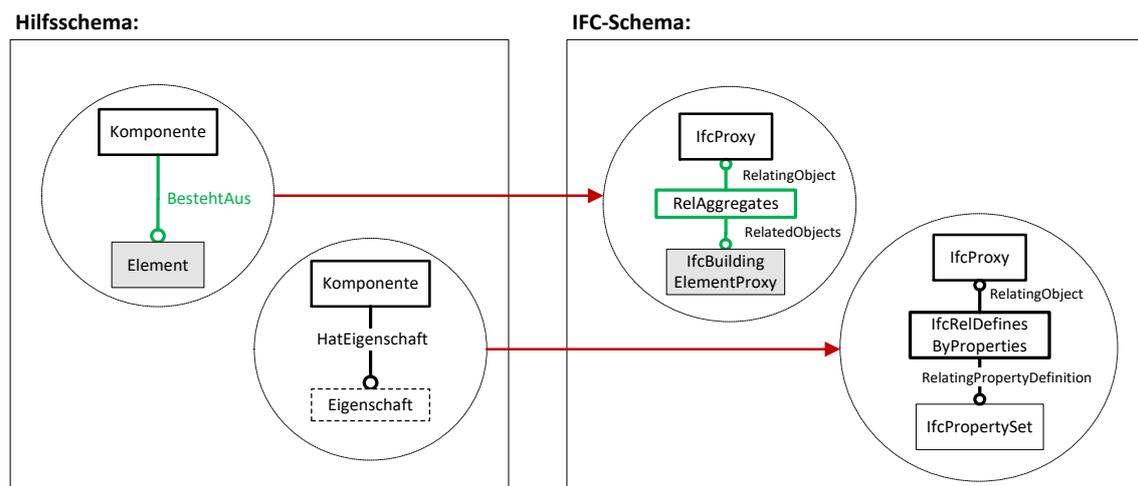


Abbildung 19: Beziehung zwischen Hilfsschema-Relationen und IFC-Entitäten

## 5.3 Ausgewählte Brückentypen

Das im Rahmen dieser Arbeit entwickelte Hilfsschema beschränkt sich auf die Beschreibung von drei konkreten Brückentypen.

Als maßgebliches Auswahlkriterium wird die Konstruktionsform der Haupttragwerke gewählt, da diese aus Sicht eines Tragwerksplaners das aussagekräftigste Unterscheidungsmerkmal darstellt.

Aus dieser Kategorie ausgesucht werden die Typen Balkenbrücke, Bogenbrücke und Schrägkabelbrücke, weil diese sich sowohl in Bezug auf ihre charakteristischen Bauelemente als auch hinsichtlich der ihnen zugrunde liegenden statischen Systeme deutlich voneinander unterscheiden.

### CAD-Modelle der drei Brückentypen

Von jedem der drei Typen wurde in *Autodesk Revit* ein CAD-Modell erstellt, das die wesentlichen Merkmale des jeweiligen Brückentyps enthält. Die Geometrie dieser Modellbeispiele ist bewusst sehr einfach gehalten, die Abmessungen der einzelnen Bauteile spielen keine Rolle.

Die aus *Revit* exportierte IFC2x3-Datei der Balkenbrücke dient später zur Verifizierung, dass sich das in den folgenden Kapiteln beschriebenen Hilfsschemas unter Verwendung des bestehenden IFC2x3-Schemas instanziiieren lässt.

#### 5.3.1 Balkenbrücke

Das Fahrbahntragwerk der Beispiel-Balkenbrücke besteht aus einer Fahrbahnplatte auf zwei vorgespannten Längsträgern. Jeder Längsträger ist an beiden Seiten über ein Punktlager auf dem Brückenwiderlager aufgelagert, die Widerlager ruhen ihrerseits jeweils auf einem separaten Fundament (siehe Abbildung 20 und Abbildung 21). Das Fahrbahntragwerk ist freitragend, d.h. es wird zwischen den beiden Widerlagern nicht durch zusätzliche Stützkonstruktionen wie beispielsweise Pfeiler abgestützt.

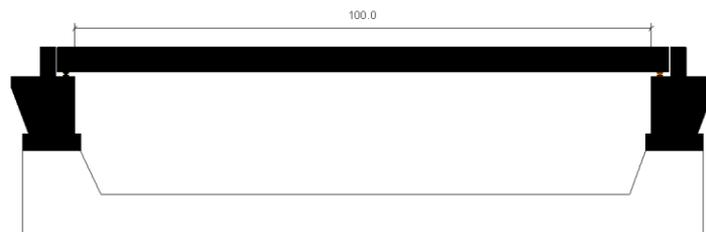


Abbildung 20: Beispiel-Balkenbrücke; Längsansicht

Über- und Unterbau sind in Stahlbetonbauweise ausgeführt, die Lagerung des Überbaus erfolgt über bewehrte Elastomerlager. Die Eigenschaften der Lager sowie die detaillierte Beschreibung der Vorspannung werden der IFC-Datei in Anhang 1 händisch hinzugefügt.

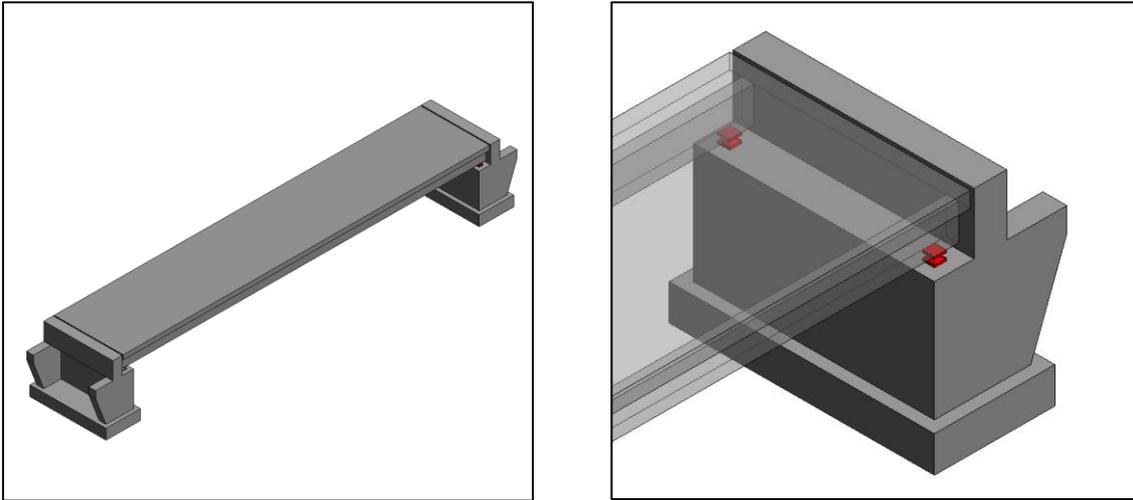


Abbildung 21: 3D-Modell der Balkenbrücke  
Erstellt in *Autodesk Revit*

### 5.3.2 Bogenbrücke

Die Beispielinstantz vom Typ Bogenbrücke entspricht in der Konstruktion des Überbaus, der Fundamente und Widerlager sowie in der Anordnung der Brückenlager der zuvor gezeigten Balkenbrücke, lediglich die Vorspannung der Längsträger entfällt.

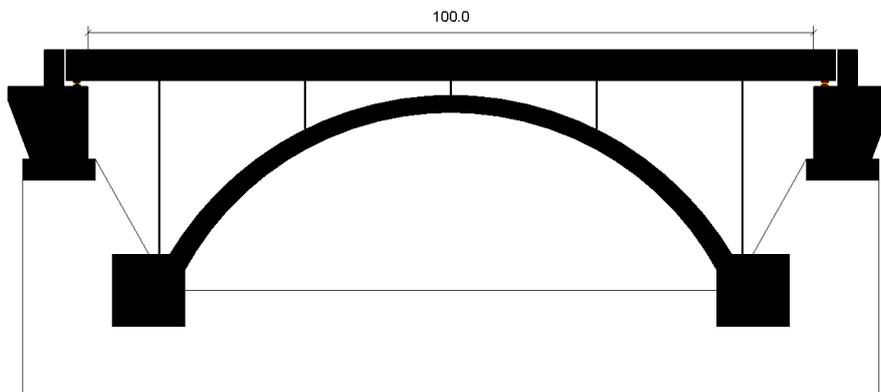


Abbildung 22: Beispiel-Bogenbrücke; Längsansicht

Zusätzlich ist das Fahrbahntragwerk auf zwei separaten, steifen Bögen aufgeständert. Die Bögen sind mit den Kämpferstützen direkt und starr verbunden. Bögen und Kämpferstützen bestehen aus Stahlbeton, auf ihnen wird das Fahrbahntragwerk über stählerne Stützen aufgeständert (siehe nachfolgende Abbildung).

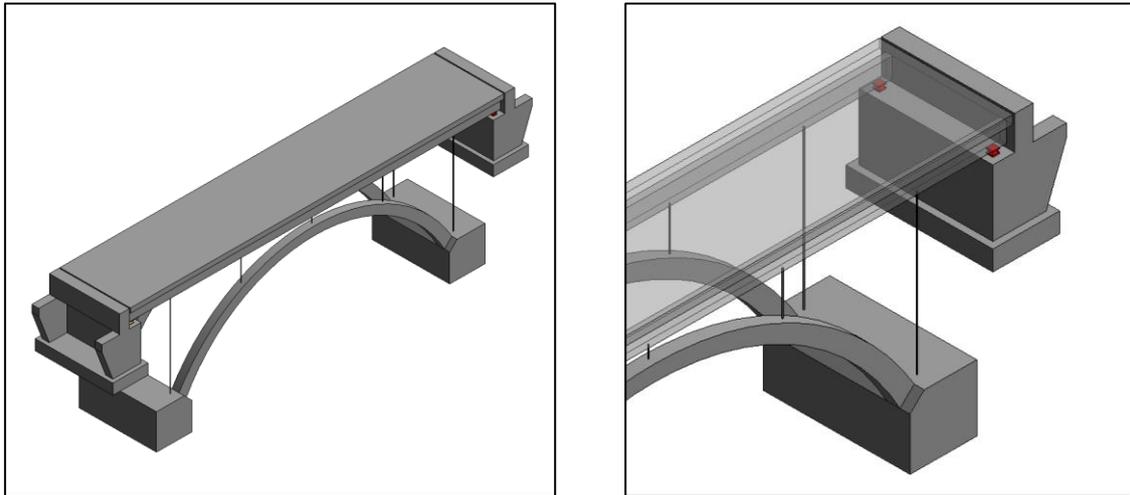


Abbildung 23: CAD-Modell der Beispiel-Bogenbrücke  
Erstellt in *Autodesk Revit*

### 5.3.3 Schrägkabelbrücke

Analog zur Bogenbrücke stellt auch das Modell der Schrägkabelbrücke eine Erweiterung der zu Anfang gezeigten Balkenbrücke dar. Hinzu kommt das Schrägkabeltragwerk, dafür entfällt die Vorspannung der Längsbalken.

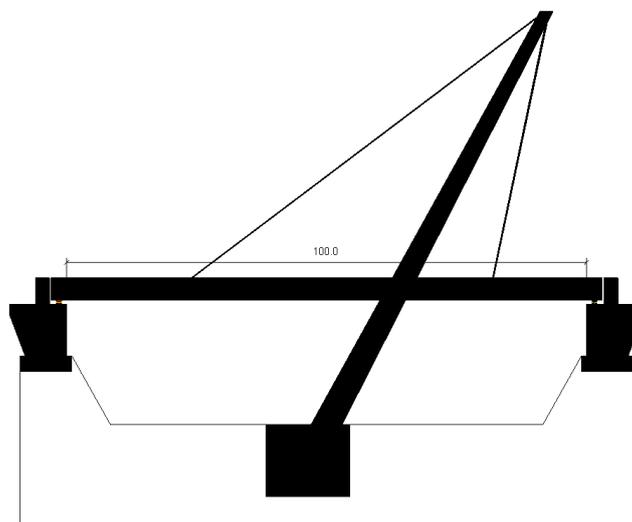


Abbildung 24: Beispiel-Schrägkabelbrücke; Längsansicht

Das Fahrbahntragwerk ist über vier Schrägkabel in zwei Kabelebenen an einem freistehenden Pylon aufgehängt, der Pylon ruht seinerseits auf einem eigenen Fundament (siehe nachfolgende Abbildung).

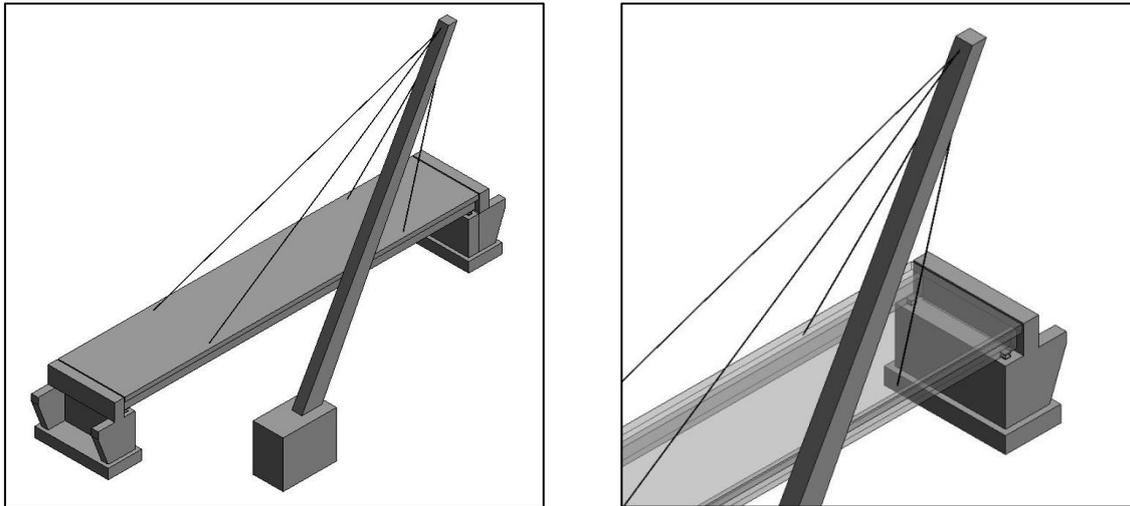


Abbildung 25: CAD-Modell der Beispiel-Schrägkabelbrücke  
Erstellt in Autodesk Revit

## 5.4 Gesamtschema – Überblick

Das im folgenden EXPRESS-G-Diagramm dargestellte Gesamtschema dient nur als Überblick. Die genaue Beschreibung des Schemas erfolgt im nachfolgenden Kapitel ausgehend von seinen einzelnen Komponenten und ist in mehrere Abschnitte aufgeteilt.

In den EXPRESS-G-Diagrammen zu den im Folgenden gezeigten Hilfsschema sind die abstrakten Oberklassen *Element*, *Komponente* und *Eigenschaft* sowie die Vererbungsbeziehung zu ihren Unterklassen aus Gründen der Übersichtlichkeit nicht abgebildet.

Stattdessen ist die Zugehörigkeit der Klassen zu ihrer jeweiligen Oberklasse über die grafische Gestaltung der (die Klassennamen einrahmenden) Felder festgelegt. Grau hinterlegte Felder stehen für Unterklassen der Klasse *Element*, Felder mit dickem, durchgezogenem schwarzen Rand für Unterklassen von *Komponente* und Felder mit gestricheltem Rand für Unterklassen von *Eigenschaft*.

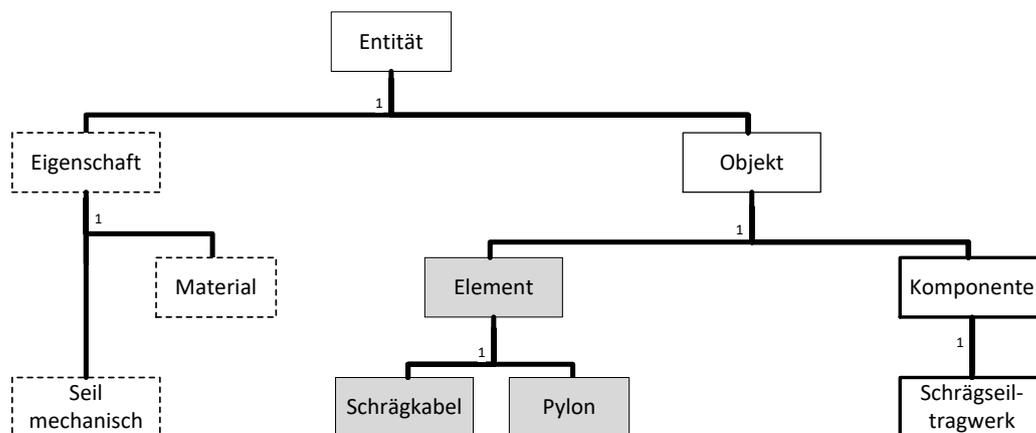


Abbildung 26: Vererbungsbeziehungen zwischen den einzelnen Klassen



## 5.5 Hierarchische Struktur der Bauwerkskomponenten

Grundsätzlich wird bei Brücken zwischen Unter- und Überbau unterschieden. Widerlager, Gründung und Pfeiler bilden den Unterbau; Fahrbahntragwerk und Ausbauelemente den Überbau. Beide sind entweder direkt und monolithisch oder verschieblich über Lager miteinander verbunden (Stetter, 2010).

Spezielle Stützkonstruktionen wie Bogen- oder Seiltragwerke, die höhere Stützweiten bei gleichzeitig geringeren Konstruktionshöhen des Brückendecks ermöglichen, werden laut Weidemann in *Brückenbau: Stahlbeton- u. Spannbetonbrücken* (Weidemann, 1982) ebenfalls dem Oberbau zugerechnet, in den meisten Fällen jedoch als eine separate Komponente betrachtet (Mehlhorn & Curbach, 2014). In dem in dieser Arbeit entwickelten Hilfsschema wurde die Darstellung als separate Komponenten gewählt.

### Modellierung der hierarchischen Struktur im Hilfsschema

Im Hilfsschema steht die abstrakte Oberklasse *Brücke* bzw. ihre drei instanzierbaren Unterklassen *Balken-*, *Bogen-* und *Schräggabelbrücke* (siehe Abbildung 28) in der hierarchischen Struktur der Gesamtkonstruktion an oberster Stelle.

Jede dieser Klassen setzt sich aus den Komponenten Überbau, Unterbau sowie optional beliebig vielen Lagerungen zusammen.

Schräggabelbrücken bestehen zusätzlich aus mindestens einem Schräggabeltragwerk, Bogenbrücken aus mindestens einem Bogentragwerk.

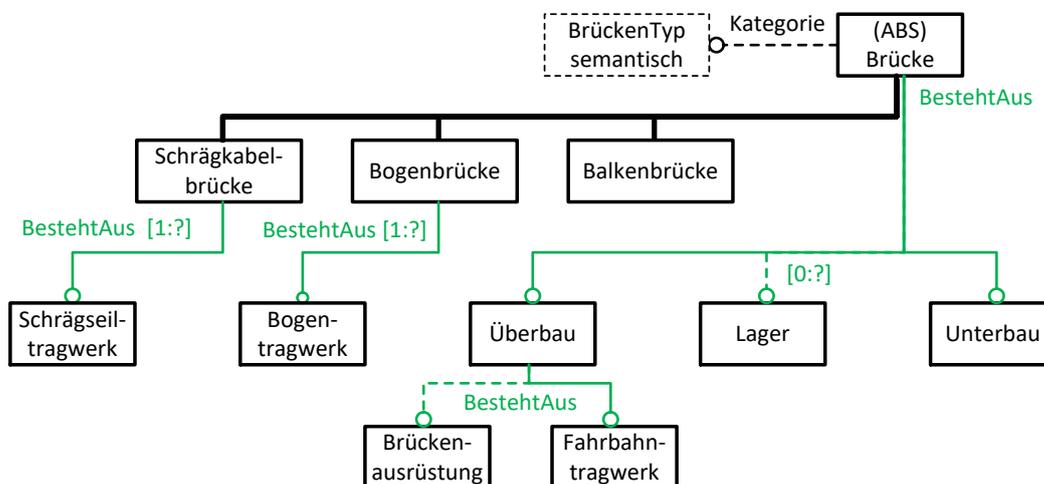


Abbildung 28: Hierarchische Struktur der Bauwerkskomponenten

Der Überbau besteht im Modell immer aus dem Fahrbahntragwerk, optional kann ihm auch die Brückenausrüstung zugerechnet werden. Über das Attribut *BrückenTyp* wird das Bauwerk als Ganzes durch die Einordnung in verschiedene Kategorien genauer definiert. Die entsprechenden Klassifikationskriterien wurden im vorangegangenen Kapitel erläutert.

## 5.6 Beziehungen und Eigenschaften der Komponenten

### 5.6.1 Komponenten des Überbaus

#### 5.6.1.1 Fahrbahntragwerk

Das Fahrbahntragwerk bildet die zentrale Komponente jedes Brückenbauwerks. Der Begriff wird in diesem Zusammenhang als Überbegriff für die tragenden Komponenten des Brückendecks von Straßen-, Eisenbahn-, Fußgängerbrücken etc. gleichermaßen verwendet; er bezieht sich also nicht allein auf die Fahrbahn von Straßenbrücken.

Fahrbahntragwerke können quer zur Brückenlängsachse in ein oder mehrere Segmente gegliedert sein. Abhängig von der Querschnittsform des Brückendecks besteht jedes Segment wiederum aus verschiedenen Elementen. Nahezu alle Querschnittsformen lassen sich dabei über die Elemente Platte, Längsbalken und Querbalken darstellen. Allerdings müssen dabei weder die Platten eben, noch die Balken gerade und senkrecht angeordnet sein (siehe Abbildung 29). Aus diesen vier Elementen können alle in Kapitel 4.1.1 beschriebenen Querschnittsformen gebildet werden.

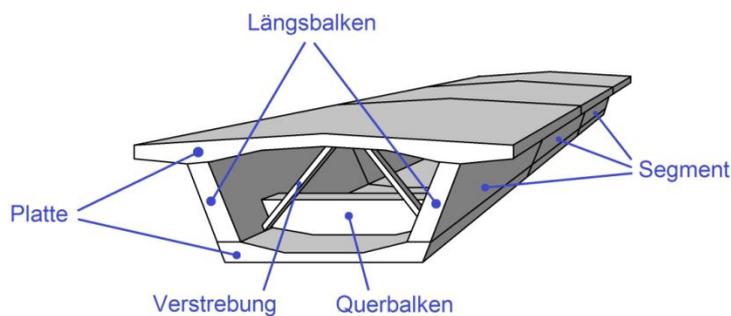


Abbildung 29: Einteilung des Fahrbahntragwerks in Segmente und Elemente

Ein Vorteil dieser Aufteilung des Querschnitts in Platte, Balken etc. besteht darin, dass interne Vorspannung auch auf der semantischen Ebene den entsprechenden Querschnittsbereichen zugeordnet werden kann, z. B. den Längsbalken.

Externe Vorspannung hingegen kann dem Fahrbahntragwerk als Ganzes zugewiesen werden.

Aus den am Fahrbahntragwerk beteiligten Elementen selbst lässt sich nicht automatisch der Querschnittstyp des Tragwerks ableiten. So kann beispielsweise ein aus einer Platte und zwei Längsbalken zusammengesetztes Tragwerk sowohl einen Plattenbalken- als auch einen Trogquerschnitt ergeben.

Daher scheint es sinnvoll, diesen in der Klasse Fahrbahntragwerk explizit zu definieren. Zur Auswahl stehen dabei die Typen Vollplatte, Trägerrost, orthotrope Platte, Plattenbalken und Hohlkasten.

## **Platte**

Platten untergliedern sich nach ihrer Funktion in Fahrbahnplatte und aussteifende Platten. Aussteifende Platten verbinden beispielsweise die unteren Enden der Stege von Hohlkastenquerschnitten miteinander.

Abhängig von ihrer bautechnischen Struktur unterscheidet man Vollplatten, Hohlplatten, Zellenkästen, Trägerroste und orthotrope Platten.

Vollplatten sind – wie der Name besagt – massive Stahlbetonplatten, die sich nicht weiter in Elemente oder Querschnittsbereiche gliedern lassen. Sie entsprechen in ihrer Tragwirkung einem scheibenförmigen Flächentragwerk.

Bei Hohlplatten werden Hohlkörper aus Kunststoff in die geringer belasteten Bereiche der Platte eingelegt, wodurch sich das Verhältnis von Eigengewicht zu Tragfähigkeit verbessert.

Unter Zellenkästen versteht man breite Hohlkästen mit annähernd rechteckigen Hohlräumen. Die Breite der Hohlräume liegt in der Regel zwischen zwei und drei Metern, ihre lichte Höhe beträgt im Allgemeinen mindestens 1,20 Meter. Das sich um die Hohlräume bildende Netz aus Längs- und Querstegen aus Beton kann als Trägerrost betrachtet werden.

Bei Trägerrosten wird eine Platte von einem Netz aus Stegen und Querträgern unterzogen. Das so entstandene Plattenbalkensystem besitzt eine so große Quersteifigkeit, dass eine näherungsweise flächenhafte Lastverteilung erreicht wird. Daher werden die Stege und Querträger als Teil des plattenförmigen Flächentragwerks betrachtet und nicht, wie bei den Stegen von Plattenbalkenbrücken, als separate Verstärkungen.

Orthotrope Platten sind Baustahlplatten mit an der Unterseite in Längs- und Querrichtung angeschweißten Querverstrebungen und weisen eine hohe Steifigkeit bei relativ geringem Gewicht auf.

## **Balken**

Balken sind schubfest mit der Fahrbahnplatte verbunden; sie dienen der Erhöhung der Querkrafttragfähigkeit und der Biegesteifigkeit. Je nachdem, ob Balken parallel oder quer zur Brückenlängsrichtung angeordnet sind, unterscheidet man Längs- und Querbalken.

## **Verstrebung**

Unter dem Begriff Verstrebungen werden in diesem Zusammenhang stabartige, vorwiegend durch Normalkräfte belastete und in der Regel diagonal angeordnete, Elemente zusammengefasst, die der zusätzlichen Aussteifung von Querschnitten dienen.

### 5.6.1.2 Vorspannung des Fahrbahntragwerks

Das Ziel von Vorspannung ist es, in einem Bauteil Spannungen zu erzeugen, die den aus den Eigen- und Auflasten entstehenden Biegezugspannungen entgegenwirken. Zu diesem Zweck werden in ein Bauteil Spannglieder eingelegt. Deren – meist abschnittsweise parabolischer – Verlauf ist komplementär zum Verlauf der durch die statischen Lasten erzeugten Biegemomente ausgerichtet.

Die Spannglieder werden an den beiden Enden mit dem vorzuspannenden Bauteil über Ankerplatten verbunden. Das Spannglied wird an einem oder auch an beiden Enden angespannt, wodurch zum einen Zugspannungen im Spannglied und zum anderen Druckspannungen im Bauteil entstehen.

Durch den gekrümmten Verlauf des Spannglieds entstehen Umlenkkräfte; diese wiederum erzeugen Biegemomente, die den aus den Eigen- und Nutzlasten resultierenden Momenten entgegenwirken.

Dies führt dazu, dass sich das Tragwerk entgegen seiner aus der Belastung resultierenden Durchbiegung verformt bzw. dass jene Bereiche des Tragwerks unter Druck gesetzt werden, in denen es ohne Vorspannung durch die Belastung zu Zugspannungen kommen würde. Auf diese Weise kann der für eine Belastung auf Druck ausgelegte Baustoff Beton wesentlich besser ausgenutzt werden; zudem werden Durchbiegung und somit Rissbildung reduziert, was der Gebrauchstauglichkeit und der Dauerhaftigkeit zu Gute kommen kann (Mehmel, 1973).

### Geometrisch-konstruktive Darstellung vs. Ersatzlasten

Die Beschreibung von Vorspannung kann auf zwei grundlegend unterschiedliche Arten erfolgen: Zum einen kann sie direkt dargestellt werden, indem der geometrische Spanngliedverlauf, die Art der Vorspannung, der Durchmesser von Spannglied und Hüllrohr etc. sowie die an den Ankerpunkten aufgebrauchte Vorspannkraft (Normalkraft tangential zum Spannglied) beschrieben werden.

Zum anderen können die aus der Vorspannung resultierenden Schnittgrößen in Form von Ersatzlasten dargestellt werden. Informationen zu Anzahl, Geometrie und Verlauf der Spannglieder sind bei dieser Art der Darstellung nicht enthalten.

Letzteres könnte bei nachträglichen Änderungen der Trägerquerschnitte zu Problemen führen, da der den Ersatzlasten zugrunde liegende Spanngliedverlauf bestimmte Mindesthöhen bzw. –breiten des Trägers voraussetzt. Diese Mindestmaße sind aus den Ersatzlasten allein jedoch nicht ersichtlich.

Sinnvoll erscheint daher eine Kombination aus beiden Varianten, bei der beide Informationen voneinander getrennt dargestellt werden, also sowohl der tatsächliche dreidimensionale Verlauf des Spannglieds im Volumenelement als auch die aus der Vorspannung resultierenden Ersatzlasten.

Denkbar wäre, die in Kapitel 6.5.1 beschriebenen Informationen zur Spanngliedgeometrie automatisch in ein CAD-Objekt umzuwandeln und in das Gesamtmodell zu integrieren. Dies könnte besonders für die detaillierte Planung der (schlaffen) Bewehrung des vorgespannten Bauteils von Vorteil sein, da so ein versehentliches Kreuzen von Spanngliedverlauf und Bewehrungsführung leichter vermieden werden könnte. Auch

eine automatische Überprüfung auf Kollisionspunkte von Bewehrungsstäben und Spannglied wäre möglich (vgl. Yabuki, Lebegue u.a., 2006).

### 5.6.1.3 Modellierung des Fahrbahntragwerks im Hilfsschema

Ein Fahrbahntragwerk besteht aus mindestens einem Segment und kann vorgespannt sein. Es ist mit maximal zwei Widerlagern sowie optional mit beliebig vielen Pfeilern direkt verbunden. Zudem kann es auf beliebig vielen beweglichen Auflagern aufliegen. Ein Segment besteht aus mindestens einer Platte sowie optional beliebig vielen Längsbalken, Querbalken und Verstrebungen; alle diese Elemente sind Unterklassen der Klasse Deckelement, jedes Deckelement kann vorgespannt sein.

Spannglieder können als Teil des gesamten Fahrbahntragwerks betrachtet werden oder aber als Teil einzelner Elemente.

Vorspannung wird sowohl als Eigenschaft von Fahrbahntragwerken bzw. Bauelementen als auch optional in Form von eigenständigen Spanngliedern mit geometrischer Repräsentation beschrieben.

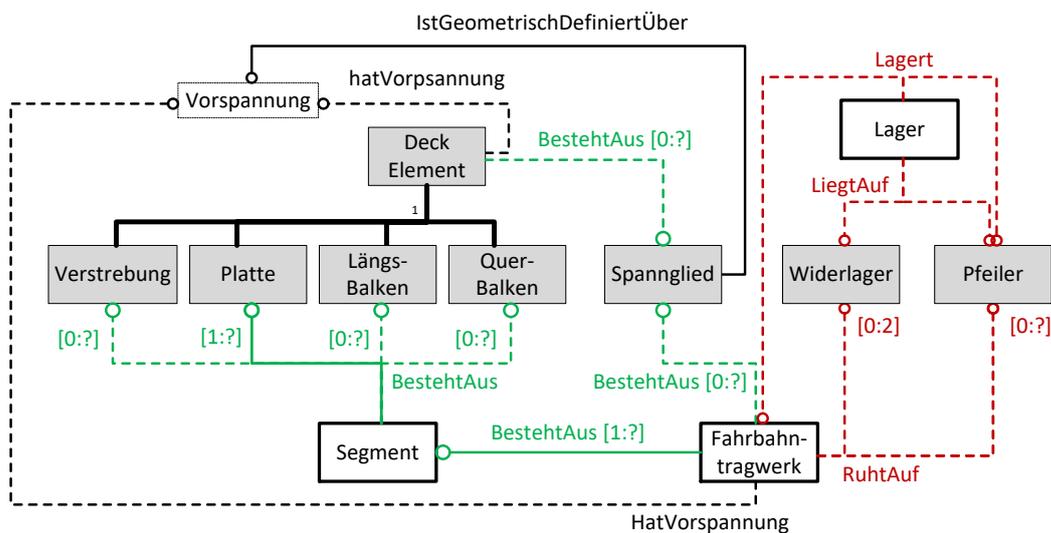


Abbildung 30: Fahrbahntragwerke im Hilfsschema; Diagramm

#### 5.6.1.4 Brückenausrüstung

Zur Brückenausrüstung werden alle Bestandteile einer Brücke gezählt, die für den Betrieb und die Gebrauchstauglichkeit von Brücke eine essentielle Rolle spielen, aber kein Bestandteil des Tragwerks sind, sich also nicht am Lastabtrag beteiligen (Kracke & Lodde, 2011).

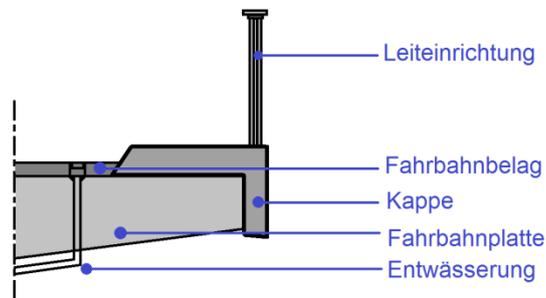


Abbildung 31: Bestandteile der Brückenausrüstung; Querschnitt durch das Brückendeck

#### Fahrbahn

Der Begriff „Fahrbahn“ wird in diesem Zusammenhang als Überbegriff für die Beläge von Straßen- und Fußgängerbrücken sowie für das Gleisbett und die Gleisanlage von Eisenbahnbrücken verwendet.

Diese Fahrbahnkonstruktionen sind meistens in mehreren Schichten aufgebaut; Dichtungs-, Verschleiß- und Ausgleichsschichten werden ebenfalls dazu gezählt.

#### Kappen

Kappen erfüllen mehrere Funktionen. Zum einen dienen sie als Bordstein, der abirrende Fahrzeuge vor dem Absturz von der Brücke schützen soll, zum anderen fungieren sie im Falle von Straßen- und Eisenbahnbrücken als (u. U. nur für Notfälle gedachter) Fuß- und Radweg. Ihre dritte Aufgabe besteht im konstruktiven Schutz der Außenkante des Brückendecks vor Witterung, Tausalzen etc.

#### Fahrbahnübergänge

„Fahrbahnübergänge haben die Aufgabe, die Fuge zwischen der Brücke und der anschließenden Fortsetzung der Fahrbahn zu schließen und die dort stattfindenden Bewegungen infolge Temperaturänderung und veränderlicher Lasten auszugleichen.“ (siehe Mehlhorn & Curbach, 2014; S.907). Grundsätzlich können diese Bewegungen sowohl in Brückenlängs- und Querrichtung als auch in vertikaler Richtung oder in Form von Verdrehungen erfolgen. Es sind jedoch nicht alle Fahrbahnübergänge in der Lage, Verschiebungen in Querrichtung oder in vertikaler Richtung aufzunehmen. Zudem sind alle Bewegungen nur in einem begrenzten Rahmen möglich.

Da Fahrbahnübergänge nicht dazu gedacht sind, diese Bewegungen zu verhindern bzw. die dabei auftretenden Kräfte aufzunehmen, müssen sie immer zusammen mit den Brückenlagern dimensioniert werden.

Fahrbahnübergänge werden in verschiedenen Konstruktionsformen ausgeführt, darunter elastische Fugen, Kissenkonstruktionen, Fingerübergänge, Lammellenübergänge und Gleitplattenübergänge (vgl. Mehlhorn & Curbach, 2014; S.909 ff).

### **Leiteinrichtungen, Beleuchtung, Entwässerung und Versorgungsleitungen**

Die Funktion dieser vier Komponenten der Brückenausrüstung ist selbsterklärend. Auf Grund des geringen Gewichts werden die durch sie entstehenden Ausbaulasten in dieser Arbeit vernachlässigt. Lediglich die durch Geländer- und Leiteinrichtungen sowie Beleuchtung eventuell erhöhten Windangriffsflächen werden berücksichtigt.

### **Lärmschutzanlagen**

Lärmschutzanlagen dienen dazu, die durch Straßen- oder Zugverkehr entstehende Lärmbelästigung für die Umgebung zu mindern. Sie bestehen in der Regel aus wandartigen Konstruktionen; die häufigste Konstruktionsform besteht aus Stahlpfosten, zwischen denen austauschbare Scheibenelemente angebracht werden. Für die Tragwerksplanung sind sie besonders auf Grund ihrer großen Windangriffsfläche von Bedeutung.

### **Modellierung im Hilfsschema**

Ausbauelemente unterscheiden sich von den restlichen Komponenten des Brückenmodells insofern, dass sie nicht Teil des Tragwerks sind. Daher werden sie vom Rest des Modells wie folgt entkoppelt:

Alle Ausbauelemente werden über die Relation *BestehtAus* als Teil der Klasse *Brückenausrüstung* definiert. Dies hat den Vorteil, dass dargestellt werden kann, welche Ausbauelemente Teil einer jeden Brückenausrüstung sein müssen, ohne dass diese Elemente gleichzeitig auch ein obligatorischer Bestandteil des Gesamtmodells sein müssen.

Zudem werden sie als Unterklasse von *Komponente* modelliert, d.h. sie besitzen selbst keine geometrische Repräsentation.

Die einzelnen Ausbauelemente können aus konkreten körperlichen Elementen bestehen, müssen dies aber nicht. So bezeichnet beispielsweise die Klasse *Entwässerung* ein abstraktes System, dem über die Relation *BestehtAusAusbauelement\_SET\_[0:?]\_OF\_Ausbauelement* optional konkrete Rohre, Schächte etc. zugeordnet werden können.

Durch die beschriebene Entkoppelung der Brückenausrüstung und der optionalen Möglichkeit, geometrische Elemente den einzelnen Ausbauelementen zuzuordnen, lässt das Modell drei unterschiedliche Detaillierungsgrade zu:

Erstens das reine Tragwerksmodell, in dem nichttragende Komponenten überhaupt nicht dargestellt werden; zweitens die Darstellung des Tragwerksmodells samt Brückenausrüstung, allerdings ohne geometrische Beschreibung konkreter Ausbauelemente und drittens die vollständige Modellierung der Brückenausrüstung samt geometrischer Repräsentation.

Die Beziehungen zwischen den Klassen sind wie folgt: Alle Ausbauelemente, darunter der Fahrbahnbelag, die Entwässerung, die Leiteinrichtungen etc. werden als Unterklasse der Klasse *Ausbauelemente* eingeführt. Über sie erben sie das optionale Attribut *BestehtAusAusbauelement*. Jede Brückenausrüstung besteht aus dem Fahrbahnbelag, der Entwässerung und den Leiteinrichtungen und den Kappen. Optional kann sie zudem aus Beleuchtung, Versorgungsleitungen, Schallschutzanlagen und Fahrbahnübergängen bestehen. Die Brückenausrüstung selbst wird über die inverse Relation zur Klasse *Überbau* als optionaler Bestandteil des Gesamtmodells definiert (siehe nachfolgende Abbildung). Die Klassen *Leiteinrichtungen*, *Fahrbahnausbildung*, *Kappen*, *Fahrbahnübergang* und *Schallschutzanlage* besitzen semantische Eigenschaften.

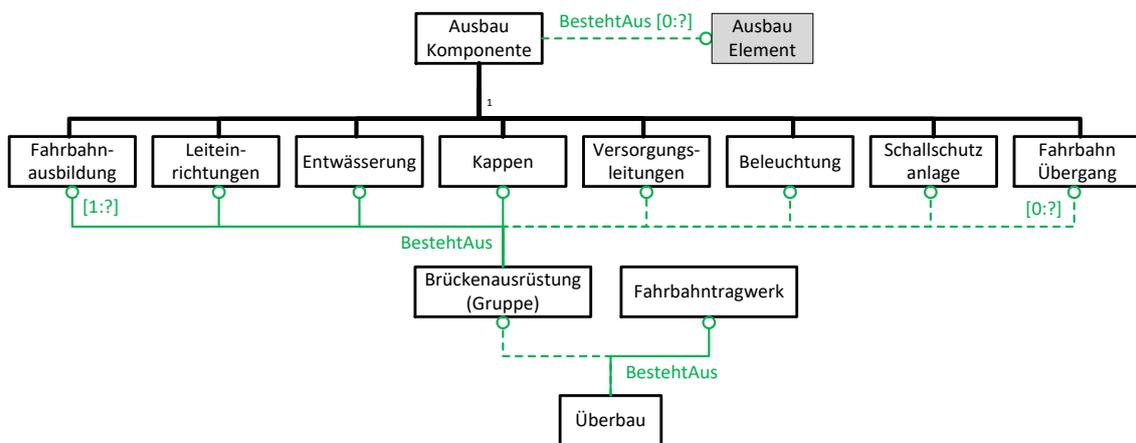


Abbildung 32: Brückenausrüstung im Hilfsschema; Diagramm

## 5.6.2 Elemente von Schrägseiltragwerken

### Seile

Seile – oft auch Kabel genannt – sind Zugglieder, die erst bei großer transversaler Auslenkung Querkräfte übertragen. Daher muss die Berechnung von Seilen nach Theorie III. Ordnung erfolgen, um diesen, erst bei großen Verformungen auftretenden, Effekt zu berücksichtigen. Die Aufgabe von Schrägseilen im Brückenbau besteht in erster Linie darin, die Lasten aus dem Brückendeck in die Pylone zu übertragen und somit die im Brückendeck auftretenden Biegemomente zu reduzieren. Zudem haben Seile und die auf sie aufgebrachten Vorspannkräfte einen erheblichen Einfluss auf die Steifigkeit und das Schwingungsverhalten des Gesamtsystems.

### Pylone

Die Aufgabe von Pylonen ist es, die über Seile in die Pylone übertragenen Lasten aus dem Brückendeck in den Baugrund abzuleiten. Obwohl die Hauptbelastung in den meisten Fällen in Form von Normalkräften entlang der vertikalen Achse der Pylone auftritt, müssen Pylone auch in der Lage sein, die aus Wind, Anfahr- und Bremslasten

sowie ggf. aus asymmetrischer Anordnung der Seile entstehenden Querkräfte und Biegemomente abzutragen.

Optional können Pylone zusätzlich die Rolle von Stützfeilern übernehmen, wobei das Fahrbahntragwerk über Lager auf dem Pylon aufliegt.

Man unterscheidet freistehende Pylone, A-förmige und H-förmige Pylone.

### **Streckträger**

Als Streckträger wird bei Schrägseilbrücken jener Teil des Fahrbahntragwerks bezeichnet, der die auftretenden Lasten aus dem Brückendeck in die Verankerung der Schrägseile überträgt sowie lokal konzentrierte Lasten auf mehrere benachbarte Seile verteilt. Umgekehrt betrachtet müssen Streckträger die über die Kabel in das Tragwerk eingeleiteten Punktlasten auf das Fahrbahntragwerk verteilen, wobei sie in erster Linie durch horizontale Druckkräfte in Brückenlängsrichtung belastet werden. Oft wird das gesamte Plattenbalken- bzw. Hohlkastensystem des Fahrbahntragwerks als Streckträger bezeichnet. Daher wird in dem im Rahmen dieser Arbeit entworfenen Hilfsschema keine gesonderte Objektklasse zur Beschreibung von Streckträgern eingeführt, diese werden stattdessen über die Klasse *Fahrbahntragwerk* abgebildet.

### **Verankerungen**

Verankerungen haben die Aufgabe, die Kräfte aus den Schrägseilen in die Pylone bzw. in das Fahrbahntragwerk einzuleiten. Dabei ist es entscheidend, den Kraftereinleitungsbereich so zu gestalten, dass die lokalen Spannungen nicht zu groß werden.

Obwohl die Konstruktion der Verankerungselemente bei der Planung von Schrägseilbrücken eine wichtige Detailaufgabe darstellt, werden Verankerungselemente in dem in dieser Arbeit entworfenen Brückenschema nicht als separate Elemente dargestellt.

Sie als eigenständige Bauteile in das Hilfsschema zur Beschreibung von Brücken aufzunehmen, entspricht in etwa dem Detaillierungsgrad, bei dem auch schlaffe Bewehrungselemente als eigenständiges Objekt dargestellt würden, was den gewählten Detaillierungsgrad des Schemas übersteigt. Stattdessen werden Schrägseile an ihren Enden direkt und punktförmig mit dem Fahrbahntragwerk bzw. den Pylonen verbunden.

### **Schrägseiltragwerke im Hilfsschema**

Ein Schrägseiltragwerk besteht aus mindestens einem Schrägseil und genau einem Pylon. Ein Schrägseil hängt an genau einem Pylon und stützt genau ein Fahrbahntragwerk (siehe Abbildung 33).

Jeder Pylon gründet auf genau einer Gründung. An ihm ist mindestens ein Schrägseil verankert, zudem kann er als Auflagerbank für Brückenlager dienen. Die Klassen *Schrägseiltragwerk*, *Seil* und *Pylon* besitzen Attribute zur Beschreibung ihrer semantischen Eigenschaften. Die speziellen mechanischen Eigenschaften von Seilen werden über das Attribut *Seil\_mechanisch* referenziert.

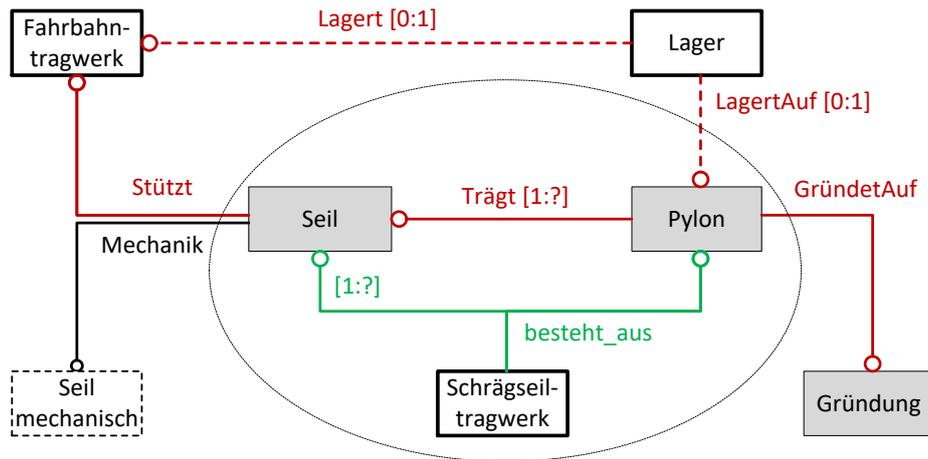


Abbildung 33: Schrägseiltragwerke im Hilfsschema; Diagramm

### 5.6.3 Elemente von Bogentragwerken

Beim Bogentragwerk folgt die Form des Bogens der sogenannten Stützlinie. Die Stützlinie bezeichnet dabei den Verlauf der Resultierenden aller Druckspannungen in einem Körper.

Im Idealfall entstehen in einem Bogentragwerk daher keine Biegemomente, der Lastabtrag erfolgt ausschließlich durch Normalkräfte tangential zur Schwerachse des Bogens. In der Realität gilt dies jedoch immer nur näherungsweise. Verändert sich die Verteilung der auf ein Tragwerk wirkenden Lasten (z. B. durch Verkehrslasten), so verändert sich auch die Stützlinie. Da die Bogenform nur an eine – meist aus den ständigen Einwirkungen resultierende – Stützlinie angepasst werden kann, entstehen durch veränderte Lastbilder immer auch Biegemomente im Bogentragwerk (Weidemann, 1982), (Mehlhorn & Curbach, 2014).

#### Bögen

Je nach Verhältnis der Steifigkeiten von Bogen und Fahrbahntragwerk werden Bögen in „steife Bögen“ und in „Stabbögen“ eingeteilt (siehe nachfolgende Abbildung).

Steife Bögen tragen sich selbst und das Fahrbahntragwerk, sie bilden für sich genommen ein unverschiebliches System, das auch ohne Verbindung zum Fahrbahntragwerk eine ausreichende Steifigkeit besitzt.

Stabbögen hingegen sind in mehrere Abschnitte unterteilt, die je nach Konstruktionsart als starr oder näherungsweise gelenkig verbunden betrachtet werden. Diese „Gelenke“ befinden sich jeweils dort, wo der Bogen mit den Ständern bzw. Hängern verbunden ist. Stabbögen können sich nicht selbst tragen, sie dienen lediglich zur Aussteifung des Überbaus.

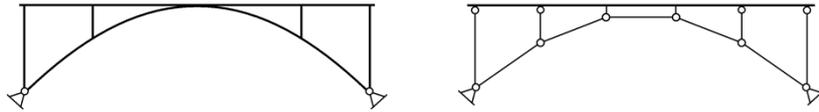


Abbildung 34: Aufgeständerte Bogenbrücke; steifer Bogen (links) und Stabbogen (rechts)

Bögen können entweder beidseitig starr oder beidseitig gelenkig mit den Kämpferstützen verbunden sein. Eingelenkbögen scheiden wegen ihres schlechten Kriech-, Schwind- und Stabilitätsverhaltens aus (Mehlhorn & Curbach, 2014).

### Hänger

Hänger verbinden bei aufgehängten Bogenbrücken bzw. Brückenabschnitten den Bogen mit dem darunterliegenden Fahrbahntragwerk. Je nach Konstruktionsform und Verankerungsart können sie entweder nur Normalkräfte entlang ihrer Achse oder auch zusätzlich Querkräfte und Biegemomente aufnehmen. Alternativ können auch Seile die Funktion von Hängern übernehmen.

### Stützen

Stützen verbinden den Bogen mit der darüber liegenden Fahrbahn, sie bilden das Pendant zu Hängern für aufgeständerte Brückenabschnitte. Stützen können sowohl als Pendelstützen als auch als starre, monolithisch verbundene Elemente modelliert werden.

### Verankerungen

Ähnlich den Verankerungen bei Seiltragwerken, können auch Bogentragwerke Verankerungselemente besitzen, über die Hänger bzw. Ständer im Bogen und im Fahrbahntragwerk verankert werden. Wie zuvor schon bei den Seiltragwerken werden Verankerungen in dieser Arbeit jedoch nicht als separate Komponenten modelliert. Hänger bzw. Ständer werden stattdessen direkt mit Bogen- oder Fahrbahntragwerk verbunden. Die meist näherungsweise gelenkigen Eigenschaften dieser Verbindungen werden durch die Zuweisung von Druck- bzw. Zugstab-Eigenschaften zu den Ständern bzw. Hängern abgebildet.

### Kämpferstützen

Kämpferstützen werden im Rahmen dieser Arbeit als Teil des Unterbaus betrachtet und nicht als Teil des Bogentragwerks.

Diese Einteilung wurde gewählt, da Kämpferstützen in ihrer Funktion den Pylonfundamenten von Schrägseiltragwerken entsprechen, welche ebenfalls dem Unterbau zugerechnet werden. Die Beschreibung der Kämpferstütze erfolgt daher im Abschnitt „Elemente des Unterbaus“.

## 5.6.4 Bogentragwerke im Hilfsschema

Die Klasse *Bogen* ist eine abstrakte Unterklasse von *Komponente* und gliedert sich in die beiden Unterklassen *SteiferBogen* und *Stabbogen* (Siehe Abbildung 35). Ein steifer Bogen besteht aus einem einzigen Bogensegment, ein Stabbogen hingegen aus mindestens drei Bogensegmenten, welche untereinander starr oder gelenkig verbunden sein können. Die gelenkige Verbindung kann über das optionale Attribut *Stabeigenschaften\_mechanisch* definiert werden. Das Attribut verknüpft das Bogensegment mit einem Property-Set, über das ihm *Stabeigenschaften* zugewiesen werden.

Auf diese Weise können Stabbogentragwerke modelliert werden, bei denen der Lastabtrag nur über Druck- und Zugkräfte in den einzelnen Stäben erfolgt.

Steife Bögen können das Fahrbahntragwerk entweder direkt stützen oder über Seile, Hänger oder Stützen mit ihm verbunden sein. Stabbögen hingegen sind immer indirekt über mindestens zwei Hänger und/oder Stützen mit dem Fahrbahntragwerk verbunden. Sowohl steife Bögen als auch Stabbögen können entweder auf dem Fahrbahntragwerk oder auf den Kämpferstützen fußen.

Die Verbindung „*Bogen – Kämpferstütze*“ kann direkt oder über Lager erfolgen, die Verbindung „*Bogen – Fahrbahntragwerk*“ hingegen erfolgt immer direkt.

Dies ist im Hilfsschema dadurch geregelt, dass Lager generell nicht auf dem Fahrbahntragwerk aufliegen können, somit ist die Verbindung „*Bogen\_FußAuf\_Lager* und *Lager\_LiegtAuf\_Fahrbahntragwerk*“ nicht möglich. Die umgekehrte Verbindung, bei der das Fahrbahntragwerk über Lager auf dem Bogen aufliegt, ist hingegen möglich.

Die mechanischen Eigenschaften von Hängern und Stützen werden über das Attribut *Stab\_mechanisch* referenziert.

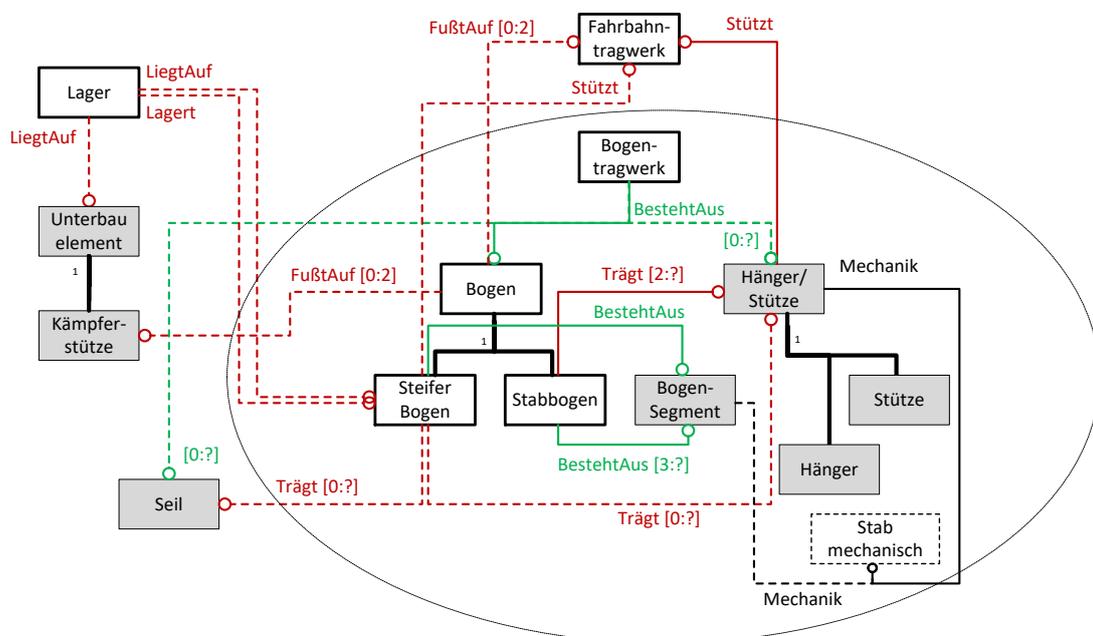


Abbildung 35: Bogentragwerke im Hilfsschema; Diagramm

## 5.6.5 Elemente des Unterbaus

### Widerlager

Als Widerlager bezeichnet man die Abschlussbauwerke an den Brückenden; sie bilden gleichzeitig das Endauflager der Brücke und den Übergang zum anschließenden Verkehrsdamm. Widerlager nehmen sowohl die Vertikal- und Horizontallasten aus dem Überbau als auch die eventuell anfallenden Erddrücke aus dem anschließenden Gelände auf.

Es existiert eine Vielzahl unterschiedlicher konstruktiver Formen. Dennoch lassen sich fast alle Widerlager räumlich in die zwei Bereiche, „Widerlagerwand“ und „Flügel“ gliedern. Die Widerlagerwand verläuft quer zur Brückenlängsrichtung; auf ihr liegt das Brückendeck auf. Die Flügel verlaufen entweder parallel oder in einem Winkel zur Widerlagerwand und nehmen die ggf. anfallenden Erddrücke des anschließenden Geländes auf. Bei Brücken mit beweglich gelagerten Überbauten gliedert sich die Widerlagerwand ihrerseits in Kammerwand und Auflagerbank, wie in der folgenden Abbildung zu sehen ist (Mehlhorn & Curbach, 2014).

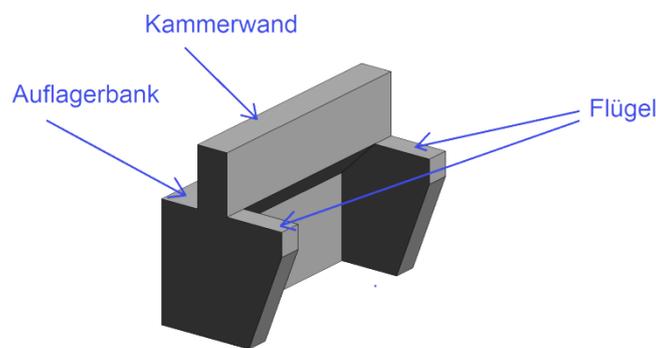


Abbildung 36: Kastenförmiges Widerlager mit Auflagerbank

Bei kastenförmigen Widerlagern mit parallel verlaufenden Flügeln sind die Flügel monolithisch mit der Widerlagerwand verbunden; das gesamte Widerlager kann als ein einziges kompaktes Bauteil betrachtet werden.

Bei Widerlagern mit sehr langen oder nicht parallel zueinander verlaufenden Flügeln werden die Flügel häufig durch eine Raumfuge von der Widerlagerwand getrennt; folglich müssten sie als separate Bauteile modelliert werden.

Da der Fokus dieser Arbeit jedoch auf dem statischen System des Brückenbauwerks und nicht auf der Interaktion des Bauwerks mit dem Baugrund liegt, wird dieses Detail im Modell vernachlässigt. Sämtliche Widerlager werden als nicht weiter untergliederte Bauteile und somit als Unterklasse der Klasse *Element* dargestellt.

### Pfeiler und Stützen

Pfeiler und Stützen bilden zusätzliche Auflagerpunkte für das Fahrbantragwerk von Mehrfeldbrücken zwischen den beiden Widerlagern. In jedem Fall nehmen sie Vertikal-

kräfte aus dem Fahrbahntragwerk auf; je nach Art der Verbindung zu Fahrbahntragwerk und Fundament können auch zusätzlich Querkräfte und/oder Biegemomente übertragen werden.

Es gibt keine allgemein anerkannte Definition, nach der sich Pfeiler und Stützen voneinander abgrenzen ließen. Beide Begriffe gehen vielmehr fließend ineinander über. In der Regel versteht man unter Pfeilern kompakte, wandartige Stützelemente, wie beispielsweise die Strompfeiler von Steinbrücken, während mit Stützen meist eher schlanke, säulenartige Bauteile gemeint sind. Einen grundlegenden Unterschied zwischen den beiden Bezeichnungen gibt es jedoch nicht.

Um sie klar von den Stützen zu trennen, die bei Bogentragwerken den Bogen mit dem Fahrbahntragwerk verbinden, wird im Folgenden nur noch der Begriff „Pfeiler“ verwendet.

Eine Spezialform des Pfeilers bildet der Pendelpfeiler. Er ist per Definition sowohl mit seinem Fundament als auch mit dem Fahrbahntragwerk gelenkig gelagert.

## Gründungen

Aufgabe von Gründungen ist es, alle Lasten der Brücke sowie oft auch Teile der Last des angrenzenden Erddammes in tragfähigen Baugrund zu übertragen.

Man unterscheidet Flach- und Tiefgründungen. Tiefgründungen sind noch einmal untergliedert in Brunnengründung, Senkkasten und Pfahlgründung (Weidemann, 1982).

Die einzelnen Gründungstypen werden im Hilfsschema nicht als Unterklassen der Klasse *Gründung* modelliert. Stattdessen wird der Gründungstyp über das Attribut *Gründung\_semantisch* definiert.

## Kämpferstützen

Kämpferstützen nehmen die Kräfte aus den Bögen von Bogenbrücken auf und leiten diese in den Baugrund ab. In der Regel besitzen Kämpferstützen keine separaten Fundamente; in ihrer Funktion entsprechen sie daher in etwa den Pylon-Fundamenten von Seiltragwerken (vgl. Abschnitt 6.8).

### 5.6.6 Modellierung der Unterbauelemente im Hilfsschema

Der Unterbau einer Brücke besteht aus mindestens zwei Widerlagern und ihren jeweiligen Gründungen. Widerlager und Gründung sind dabei starr verbunden.

Zudem kann der Unterbau aus beliebig vielen Pfeilern und Kämpferstützen bestehen. Über das optionale Attribut *Stab\_mechanisch* können die mechanischen Eigenschaften stabartiger Pfeiler definiert werden.

Alle Unterbauelemente werden der Klasse *Unterbauelement* untergeordnet. Über sie erben sie die inverse Relation, die sie als mögliche Auflagerbank von Brückenlagern definiert.

Pfeiler können sowohl mit ihrem Fundament als auch mit dem Fahrbahntragwerk direkt und monolithisch oder beweglich über Lager verbunden sein. So befindet sich die Lagerungsebene bei Rahmenbrücken beispielsweise oft zwischen den Pfeilern und ihren Fundamenten, während das Fahrbahntragwerk starr mit den Pfeilern verbunden ist.



## Verformungslager

Im Brückenbau werden als Verformungslager immer bewehrte Elastomerlager verwendet. Diese bestehen aus mehreren Elastomerschichten, die von horizontalen Stahlblechen getrennt werden. Diese als Bewehrung fungierenden Stahlbleche behindern ein seitliches Ausbauchen des Elastomers, wie es durch Pressung in vertikaler Richtung hervorgerufen wird. Eine Schubverformung, wie sie durch horizontales Verschieben des Überbaus gegenüber dem Unterbau entsteht, behindern die Bleche nicht (siehe Abbildung 39).

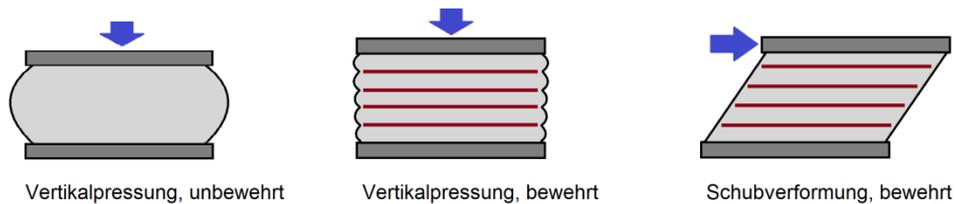


Abbildung 39: Verformungsbilder Elastomerlager, unbewehrt und bewehrt

Daher ist die Steifigkeit des Lagers in horizontaler Richtung ca. 1000-mal geringer als in vertikaler Richtung. Die so ermöglichte elastische Verschieblichkeit in der Horizontalen erzeugt eine Rückstellkraft, die der Verformung entgegenwirkt. Die Größe dieser Kraft ist abhängig von der Größe der Verformung, von der Größe und Geometrie des Lagers sowie von dem (temperaturabhängigen) Schubmodul des Elastomers.

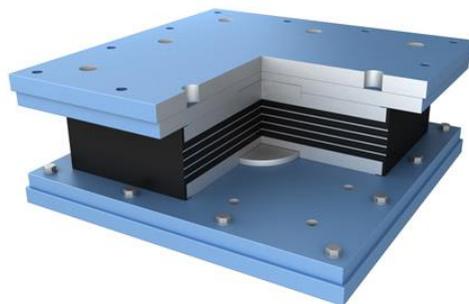


Abbildung 40: Bewehrtes Elastomerlager

Quelle: (oiles corporation, o.J.)

Das gleiche Prinzip führt auch zu einer elastischen Verdrehbarkeit des Lagers um die Vertikalachse; auch hier wird eine gegenseitige Verdrehung der Elastomerschichten durch die Stahlbleche nicht behindert. Eine Verdrehung um eine horizontale Achse ist ebenfalls möglich. Diese führt jedoch zu einer ungleichmäßigen Pressung des Lagers in vertikaler Richtung, wobei die Wirkung der Stahlblech-Bewehrung wieder zum Tragen kommt.

Um Verschiebungen bzw. Verdrehungen zwischen Über- und Unterbau in das Lager einzuleiten, muss ein schubfester Verbund zwischen den Bauteilen des Tragwerks und dem Lager vorhanden sein.

Bei ausreichenden vertikalen Druckspannungen wird dieser Verbund durch die Reibung zwischen Lager und Tragwerk erzeugt. Ist der Druck in vertikaler Richtung zu gering, sind konstruktive Maßnahmen zur Sicherung des Verbundes notwendig. Daher unterscheidet man bei Verformungslagern zwischen unverankerten und verankerten Elastomerlager.

Die durch Schubverformung des Lagers ermöglichte Horizontalverschieblichkeit ist relativ gering und erzeugt, wie bereits erwähnt, eine Rückstellkraft. Falls größere, freie Verformungen in der Horizontalebene erwünscht sind, können zusätzlich Gleitteile eingeführt werden. Verformungslager mit zusätzlichen Gleitteilen werden auch als Verformungs-Gleitlager bezeichnet.

## Stahllager

Stahllager gibt es als Punkt- und Linienlager. Sie bestehen aus zwei Stahlplatten, die über einen kugel- oder zylinderförmigen Stahlkörper miteinander verbunden sind. Man unterscheidet zudem Kipplager und Rollenlager.



Abbildung 41: Stählernes Rollenlager

Quelle: (Feller, 2013)

Bei Kipplagern ist der kugel- bzw. zylinderförmige Stahlkörper fest mit einer der Platten verbunden, wodurch Verdrehungen zwar möglich, Verschiebungen jedoch verhindert sind. Im Fall von Rollenlager sind zwei ebene Platten über eine stählerne Walze verbunden, über die die obere Platte auf der unteren „abrollen“ kann. Dadurch wird sowohl einachsige Verdrehung als auch einachsige Verschiebung ermöglicht.

Bei Stahllagern findet, im Gegensatz zu den Elastomerlagern, keine elastische Verformung statt. Folglich entstehen auch keine Rückstellkräfte bzw. –momente.

Analog zu den Elastomerlagern können auch Stahllager durch zusätzliche Gleiteinrichtungen einseitig bzw. allseitig beweglich gemacht werden.

## Topflager

Beim Topflager wird die Vertikallast durch Pressung der Elastomereinlagen flächenhaft übertragen; ein Ausquetschen des Elastomers wird durch den Topfrand verhindert (siehe Abbildung 42). Das Lager ist um alle drei Raumachsen elastisch verdrehbar, die Größe der resultierenden Rückstellmomente hängt u. a. von der Geometrie des Lagers, dem (temperaturabhängigen) Schubmodul des Elastomers sowie konstruktiven Details wie Schmierung und Dichtung ab. Zudem wird es beeinflusst von äußeren Faktoren wie der Größe und Geschwindigkeit der Verdrehung selbst sowie der von der angreifenden Vertikallast abhängigen Pressung des Elastomers.

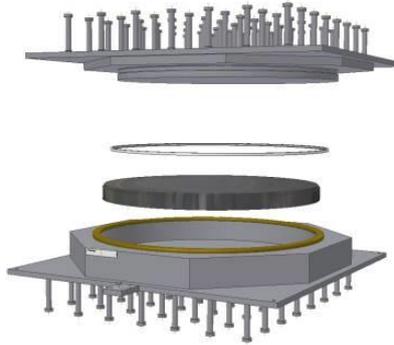


Abbildung 42: CAD-Modell eines Topflagers

Quelle: (BT Bautechnik GmbH, o.J.)

Um eine freie Verschieblichkeit zu ermöglichen, kann auch hier ein zusätzliches Gleitteil eingeführt werden. Diese flächig zwischen Lagertopf und Lagerdeckel angebrachte, sog. „Gleitpaarung“ ermöglicht ein freies Gleiten des Lagerdeckels auf dem Lagertopf in alle Richtungen entlang der Horizontalebene. Ist die Verschieblichkeit nur in eine Richtung gewünscht, kann zusätzlich eine Führungsleiste zwischen Topf und Deckel angebracht werden, die das Gleiten quer zur Leiste verhindert. Die Relativverschiebung des Topfes gegenüber dem Deckel führt zu einer exzentrischen Einleitung der Last in den Lagerdeckel. Je nach Einbaurichtung muss diese bei der Berechnung des Über- bzw. Unterbaus berücksichtigt werden.

Die maximal zulässige Verdrehung eines Topflagers ist von Bauart und Größe des Lagers abhängig, gleiches gilt für die maximal zulässigen Vertikalkräfte.

### Kalottenlager

Kalottenlager entsprechen einer allseitig gelenkigen Lagerung. Eine konvexe Edelstahlplatte, die Kalotte, lagert in einer konkaven Stahlplatte, Pfanne genannt. Zwischen beiden ist ein (ebenfalls sphärisch gekrümmtes) Gleitblech angebracht, wodurch sich die Kalotte in der Pfanne bewegen kann (siehe Abbildung 43).

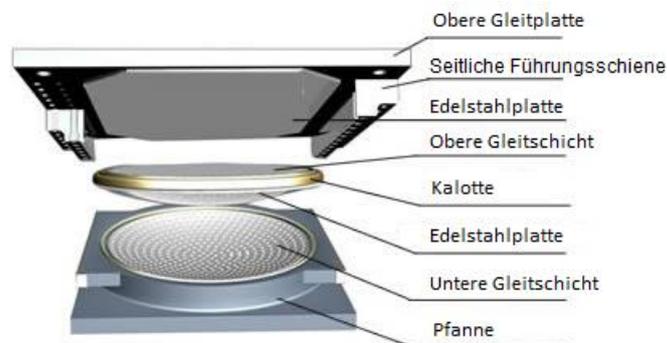


Abbildung 43: CAD-Modell eines Kalottenlagers

Quelle: in Anlehnung an (Dacheng Rubber Co., Ltd, 2014)

Die Oberseite der Kalotte ist eben geformt und über ein zweites Gleitbleich mit einer dritten – ebenen – Stahlplatte verbunden. Die zweite Gleitlage zwischen der oberen Stahlplatte und der Kalotte ist von entscheidender Bedeutung. Erst sie ermöglicht es, dass es bei einer Verdrehung des Lagers um eine horizontale Achse nicht auch zu einer horizontalen Verschiebung des Überbaus relativ zum Unterbau kommt. Das Prinzip ist in Abbildung 44 verdeutlicht. Die mittlere Zeichnung zeigt das Verformungsbild, wie es bei einem Kalottenlager mit zwei Gleitschichten entsteht, die rechte Zeichnung beschreibt, wie sich das Lager ohne die zweite Gleitschicht verformen würde.

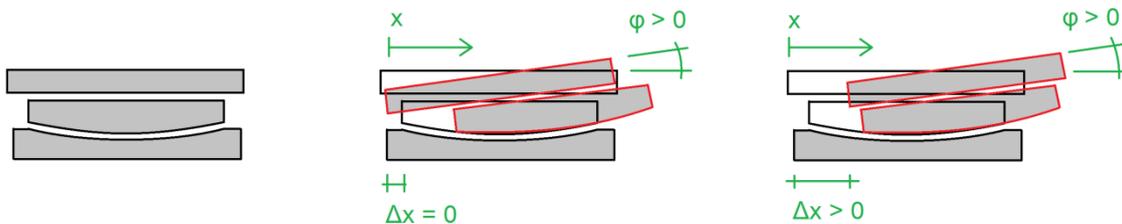


Abbildung 44: Verformungsbild eines Kalottenlagers; Prinzipskizze

Bei der Verdrehung eines Kalottenlagers um eine horizontale Achse entstehen ebenfalls Rückstellmomente. Anders als bei den Elastomerlagern resultieren sie jedoch nicht aus der elastischen Schubverformung des Lagers, sondern aus der bei der Verdrehung entstehenden Exzentrizität der Vertikalkraft (Mehlhorn & Curbach, 2014). Der Reibungswiderstand in den Gleitflächen führt dazu, dass die horizontale Verschiebung, die durch die obere, ebene Gleitfläche verhindert werden soll, in der Realität nicht exakt Null ist. Daher liegt die reale Verschiebung zwischen der in der Mitte und der rechts dargestellten Skizze in Abbildung 44. Die Größe der so entstandenen Exzentrizität der resultierenden Vertikalkraft im Lager ist abhängig von dem (temperaturabhängigen) Reibbeiwert der Gleitflächen und dem Radius der Kalotte. Die maximal aufnehmbare Horizontalkraft ist von der Höhe der vorhandenen vertikalen Belastung abhängig. Um die Kalotte in horizontaler Richtung unverschieblich zu machen, werden an den Seiten der oberen Stahlplatte Führungen angebracht, die die Pfanne seitlich umgreifen. Das Anbringen einer solchen Führungsschiene führt gleichzeitig zu einer Verhinderung der Verdrehung um die vertikale Achse.

### Horizontalkraftlager

Horizontalkraftlager sind Führungslager, die eine Verschiebung des Überbaus gegenüber dem Unterbau in der Horizontalebene verhindern.

Man unterscheidet punktförmige und linienförmige Horizontalkraftlager. Punktförmige Lager sind unverschieblich, linienförmige Lager lassen die Horizontalverschiebung in eine Richtung zu; sie entsprechen vom Prinzip einer waagerechten Führungsschiene. Durch die Lager werden weder Vertikalkräfte noch Biegemomente übertragen. Es ist jedoch zu beachten, dass durch den exzentrischen Lastangriff der Horizontalkräfte dennoch Biegemomente in den Anschlussfugen zwischen Lager und Über- bzw. Unterbau entstehen.



Abbildung 45: Festes Horizontalkraftlager; Foto (links, Mitte) und Prinzipskizze (rechts)

Quelle: (mageba-germany.de, o.J.)

### Besonderheiten bei der Modellierung von Brückenlagern

Vor der Einführung von computergestützten Berechnungen – und zu einem gewissen Grad auch noch heute – basierten die Modelle zur Berechnung der Interaktion von Über- und Unterbau über Lager auf teilweise unzulässigen Vereinfachungen (Eggert & Kauschke, 2002). An den Auflagerpunkten wurden die Lasten aus dem Überbau wie für einen durchgehenden Balken berechnet; danach wurde der Unterbau vom Überbau losgelöst entworfen. Die Auflagerkräfte wurden als äußere Kräfte sowohl auf den Unterbau als auch auf den Überbau angesetzt. Diese Methode liefert nur dann realitätsnahe Ergebnisse, wenn der Unterbau nahezu starr ist.

In Bezug auf Zwangsspannungen liegen die Ergebnisse für das Gesamtmodell auf der sicheren Seite, da die Flexibilität des Unterbaus – und die damit verbundene Fähigkeit, Zwangsspannungen abzubauen – vernachlässigt wird.

Problematisch ist, dass die oft nicht zu vernachlässigenden Effekte von Auflagerverchiebungen nicht abgebildet werden. Durch translatorische Verschiebung der Auflagerpunkte entstandene Lasten zweiter Ordnung können das Kräftegleichgewicht der Unterbauelemente stark verändern, insbesondere im Falle hoher und schlanker Bauteile wie beispielsweise Brückenpfeiler.

Aus diesen Gründen ist es sinnvoll, ein Gesamtmodell aus Überbau, Unterbau und realitätsnaher Lagerung zu bilden, bei dessen Berechnung alle oben genannten Effekte berücksichtigt werden.

#### 5.6.7.1 Verallgemeinerte Darstellung im Modell

Alle im Vorangegangenen beschriebenen Lagertypen werden im Folgenden in zwei verallgemeinerten Modellen zusammengefasst, dem Punktlager-Modell und dem Linienlager-Modell. Diese Modelle besitzen nicht mehr die konkreten Eigenschaften und Einflussfaktoren der verschiedenen Lagertypen, sondern bilden lediglich die für das statische System relevanten Auswirkungen der einzelnen Faktoren ab.

Alle systemrelevanten Eigenschaften einer Lagerung lassen sich über sechs räumliche Freiheitsgrade beschreiben. Jedem Freiheitsgrad wird eine Federsteifigkeit zugeordnet, die Werte zwischen Null und einer beliebig großen Zahl  $n$  annehmen kann. „Null“ steht für eine freie Verschieblichkeit bzw. Verdrehbarkeit, jede Zahl größer Null steht für eine tatsächliche Steifigkeit. Abbildung 46 zeigt dieses Prinzip anhand des 2D-Modells einer Punktlagerung.

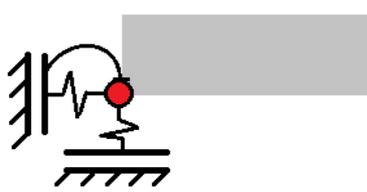


Abbildung 46: Lagerungsbedingungen als Federsteifigkeiten, 2D-Modell

Um nichtlineare Spannungs-Dehnungsbeziehungen abzubilden, kann die Steifigkeit auch als eine von der Verschiebung abhängige Funktion angegeben werden.

Ist eine Relation zu einem bestimmten Freiheitsgrad nicht vorhanden, gilt die entsprechende Verschiebung bzw. Verdrehung automatisch als vollständig verhindert. Das Modell geht also von einer starren, unverschieblichen Lagerung aus, dem einzelne Freiheitsgrade explizit zugewiesen werden müssen. Dies hat den Vorteil, dass nicht in jedem Fall alle sechs Freiheitsgrade einzeln definiert werden müssen, sondern nur diejenigen, deren Steifigkeit kleiner unendlich ist. Wie dies innerhalb des IFC-Schemas über die entsprechenden Property-Sets umgesetzt wird, wird in Kapitel 6.10 ausführlich beschrieben.

Auf die Ursachen der einzelnen Steifigkeiten wird dabei nicht eingegangen, da sich die Faktoren, welche beispielsweise die Rückstellkräfte und –momente der verschiedenen Lagertypen beeinflussen, nicht verallgemeinern lassen. Wie im Vorangegangenen erläutert, hängen diese im Fall der Elastomerlager vom Schubmodul des Elastomers ab, während sie bei Kalottenlagern von den Reibbeiwerten der Gleitschichten beeinflusst werden.

Auch der Einfluss der Temperatur auf die Steifigkeiten bestimmter Lager wird in den mechanischen Lagereigenschaften nicht berücksichtigt.

Neben den linearen Federsteifigkeiten müssen auch eventuelle nichtlineare Eigenschaften, wie ein Ausfall der Kraftübertragung beim Auftreten vertikaler Zugkräfte oder ein Blockieren des Lagers, ab einer bestimmten Auslenkung beschrieben werden können.

Einzuhaltende Grenzwerte, wie z. B. maximal zulässige Kräfte und Verschiebungen, können dem Lager zusätzlich als semantische Information angehängt werden. Warum dies notwendig sein kann, ist ebenfalls in Abschnitt Kapitel 6.10 erläutert.

### Lager als Volumenkörper

Prinzipiell können Lager auf zwei verschiedene Arten modelliert werden. Zum einen können sie lediglich als die Eigenschaft einer Verbindung zwischen zwei Körpern dargestellt werden, zum anderen können sie als ein physikalischer Körper mit geometrischer Repräsentation modelliert werden, der aus zwei beweglich miteinander verbundenen Teilen besteht. Die mechanischen Eigenschaften des Lagers werden dabei der Verbindung zwischen den beiden Lagerhälften zugewiesen.

Letztere Variante hat den Vorteil, dass Form und Größe des Lagers bereits festgelegt sind, es muss lediglich noch an einer bestimmten Stelle des Modells platziert werden. Dies bietet sich besonders deshalb an, da es sich bei Brückenlagern in der Regel um Fertigteile mit fester Größe handelt.

Ein weiterer Vorteil dieser Herangehensweise besteht darin, dass das durch den vertikalen Abstand der horizontalen Kräfteinleitung zur Anschlussfuge des Lagers entstandene Moment nicht als eine zusätzliche Eigenschaft des Lagers beschrieben werden muss. Bei einem als Volumenkörper modellierten Lager ergibt sich dieses Moment automatisch aus dem tatsächlichen vertikalen Abstand von Lagerfuge zum Verbindungspunkt der beiden Lagerhälften (siehe Abbildung 47).

Zudem kann auf das bei punktförmigen Lagerungen häufig auftretende Problem der lokalen Singularitäten auf diese Weise bereits am Lager selbst eingegangen werden, indem beispielsweise die Steifigkeit des verwendeten Materials sowie die Maschen des FE-Netzes entsprechend angepasst werden. Das Lager kann einmalig detailliert modelliert werden und danach an beliebig vielen Stellen unterschiedlicher Brückenmodelle platziert werden.

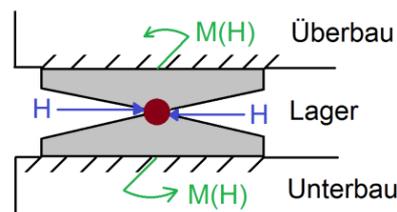


Abbildung 47: Momente in den Anschlussfugen aus Horizontalkraft

### Definition im Hilfsschema

Die Komponente *Lager* setzt sich aus den beiden Elementen Lagerunterteil und Lageroberteil zusammen; diese sind als „beweglich miteinander verbunden“ definiert. Die Klasse der Lager untergliedert sich in die beiden Klassen *Punktlager* und *Linienlager*; deren mechanische Eigenschaften werden über unterschiedliche Property-Sets festgelegt. Über zwei Select-Types ist festgelegt, dass jedes Lager entweder ein Fahrbahntragwerk, einen Bogen oder einen Pfeiler lagert und entweder auf einem Unterbauelement, einem Pylon oder einem Bogen aufliegt.

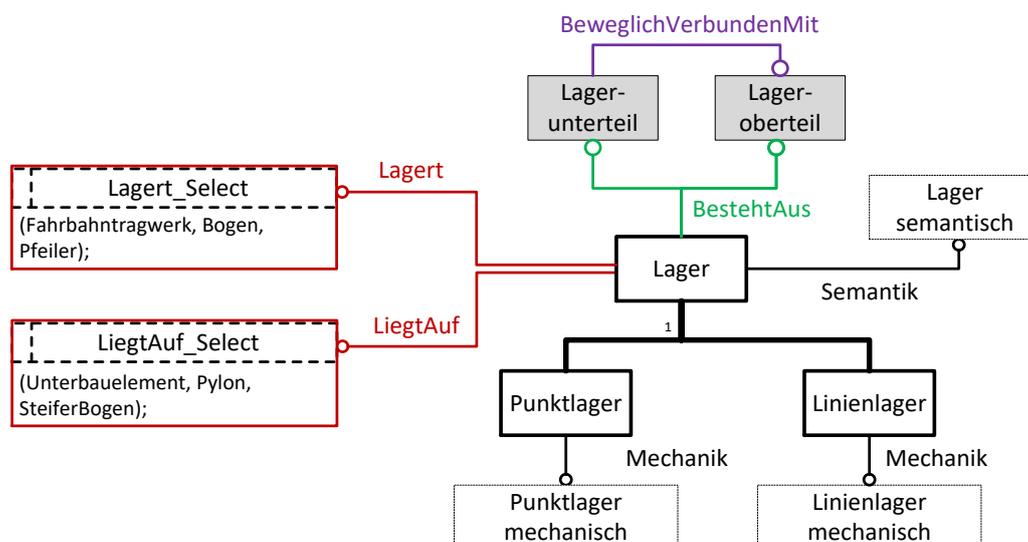


Abbildung 48: Brückenlager im Hilfsschema; Diagramm

## Vereinfachte Darstellung in den Kontext-Diagrammen

Beziehungen zu Lagern sind in den *EXPRESS-G*-Diagrammen der meisten anderen Komponenten enthalten. Der Übersichtlichkeit halber werden die entsprechenden Relationen dort in vereinfachter Form dargestellt. An Stelle der präzisen Darstellung der Verbindung über die Select-Types werden vereinfacht optionale Relationen zu den im Select-Typ zur Auswahl stehenden Klassen dargestellt.



Abbildung 49: Vereinfachte Darstellung der Beziehungen zu Lagern in EXPRESS-G

## 6 Darstellung der Objekteigenschaften über *IfcPropertySet*

Im diesem Kapitel werden alle semantischen und mechanischen Eigenschaften der im Hilfsschema beschriebenen Klassen definiert. Dafür wird die dynamisch erweiterbare IFC-Klasse *IfcPropertySet* genutzt.

Informationen zur Semantik und zur Mechanik werden jeweils in separaten Property-Sets beschrieben; dies soll einen gezielten Zugriff auf die jeweiligen Informationen erleichtern. Property-Sets mit dem Präfix „mechanisch“ enthalten dabei alle Informationen, die zur Umwandlung des Geometriemodells in ein Tragwerksmodell benötigt werden; alle anderen Informationen werden den Property-Sets mit dem Präfix „semantisch“ untergeordnet.

### *IfcPropertySet* und *IfcProperty*

Die Klasse *IfcPropertySet* dient als eine Art Container; über sie werden mehrere Eigenschaften in einem „Set“ zusammengefasst die sich alle auf die gleichen Objekte beziehen. Die entsprechenden Objekte werden über *IfcRelDefinesByProperties* mit dem Property-Set verknüpft.

*IfcProperty* untergliedert sich in die beiden Klassen *IfcComplexProperty* und *IfcSimpleProperty*.

*IfcComplexProperty* beschreibt eine komplexe Eigenschaft, d.h. eine Eigenschaft, die ihrerseits aus mehreren einfachen Eigenschaften besteht. So setzt sich beispielsweise die komplexe Eigenschaft „Material“ aus den einfachen Eigenschaften „Dichte“, „E-Modul“, „Schubmodul“ etc. zusammen.

In IFC2x3 stehen sechs Arten von Simple-Propertys, also einfachen Eigenschaften, zur Verfügung, die im Folgenden kurz beschrieben werden. Die Bedeutung der jeweiligen Eigenschaft wird immer über das Namensattribut definiert (also z.B. „Stadt“). Die zu einer Eigenschaft gehörenden Werte können beschreibender oder numerischer Natur sein.

Über *IfcPropertySingleValue*, kann ein einfacher Wert und optional eine zugehörige Einheit beschrieben werden.

Mittels *IfcPropertyEnumeratedValue* können ein oder mehrere Werte aus einer Liste von Werten ausgewählt werden; die Werte der Liste können ebenfalls frei definiert werden, sie müssen lediglich alle vom gleichen Datentyp sein (also z.B. alle vom Typ STRING oder REAL).

*IfcPropertyTableValue* beschreibt eine Reihe von Werten, die paarweise voneinander abhängig sind. Auf diese Weise lässt sich eine Art Tabelle mit zwei Spalten darstellen, wobei die Werte der zweiten Spalte von den Werten der ersten Spalte abhängen. Jeder „Spalte“ kann eine eigene Einheit zugewiesen werden. Über das Attribut *Description* wird in einigen Fällen festgelegt, wie die einzelnen Positionen der Liste zu interpretieren sind. So wird z.B. bei der Beschreibung dreidimensionaler Translationsfedern festgelegt, dass sich der erste Eintrag auf die Steifigkeit in x-Richtung bezieht und der zweite und dritte Eintrag dementsprechend auf die Koordinatenrichtungen y und z.

Über *IfcPropertyListValue* kann eine Liste von Werten gleichen Datentyps dargestellt werden, der Gesamtliste kann optional eine Einheit zugewiesen werden.

In einigen Fällen wird in der *Description* festgelegt, wie die einzelnen Positionen der Liste zu interpretieren sind. So wird z.B. bei der Beschreibung dreidimensionaler Translationsfedern festgelegt, dass sich der erste Eintrag auf die Steifigkeit in x-Richtung und der zweite und dritte Eintrag dementsprechend auf die Koordinatenrichtungen y und z beziehen.

*IfcPropertyReferenceValue* ermöglicht den Verweis aus einem Property-Set auf eine externe Information, beispielsweise auf ein Material aus einer externen Materialbibliothek.

Über *IfcPropertyBoundedValue* können Ober- und Untergrenze eines Intervalls festgelegt werden. Dem Intervall kann ebenfalls eine Einheit zugewiesen werden.

## 6.1 Brückentyp

Über das Property-Set *Brückentyp\_semantisch* wird festgelegt, um welche Art von Brücke es sich bei einer Modellinstanz handelt.

Über Enumerated-Values kann ausgewählt werden, welche Funktion die Brücke erfüllt, ob sie über Süßwasser, Salzwasser oder über Land führt und ob das Brückendeck verschieblich oder starr mit dem Unterbau verbunden ist. Das Kriterium *Material* beschreibt, welcher Baustoff für die Tragstruktur der Brücke in erster Linie verwendet wurde, es ersetzt nicht die Zuweisung von Materialeigenschaften zu den einzelnen Bauelementen.

Tabelle 8: Festlegung des Brückentyps

IfcPropertySet „BrückenTyp_semantisch“		
Name	Typ	Value
Material	Enumerated	„Stein“, „Holz“, „Stahl“, „Stahlbeton“, „Karbonbeton“, „Stahlverbund“, „Holz-Beton-Verbund“, „Holz-Stahl-Verbund“;
Funktion	Enumerated	„StraßenB“, „FußgängerB“, „EisenbahnB“, „KanalB“, „WildB“, „FörderbandB“, „LeitungsB“, „PionierB“;
Lage	Enumerated	„Süßwasser“, „Salzwasser“, „Festland“;
Verbundart	Enumerated	„Aufgelagert“, „Rahmen“, „Semi-Integral“

## 6.2 Material und Materialeigenschaften

Materialparameter fließen in die Steifigkeit von Flächen, Querschnitten und Volumenkörpern mit ein und sind somit ein notwendiger Teil der für statische Berechnungen eines Modells benötigten Modelldaten (vgl. Kapitel 3.1). Materialien können entweder aus Materialbibliotheken ausgewählt oder vom Benutzer selbst definiert werden.

Tabelle 9: Materialeigenschaften

IfcPropertySet „Material_mechanisch“			
Name	Typ	Value	Description
Nutzerdefiniert	SingleValue	Boolean	
Materialkennwerte	Complex	<i>Siehe Tabelle 10</i>	Entfällt wenn Nutzerdefiniert = FALSE
VordefiniertesMaterial	ReferenceValue	<i>Siehe Abbildung 50</i>	Entfällt wenn Nutzerdefiniert = TRUE

### Referenzierung von Materialien aus Materialbibliotheken von Statikprogrammen

Falls ein vordefiniertes Material aus der Bibliothek ausgewählt werden soll, kann dies über ein *IfcPropertyReferenceValue* referenziert werden. Dazu wird dem Property-Set *Material* das entsprechende *IfcPropertyReferenceValue* angefügt. Über dessen Relation zu *IfcObjectReferenceSelect* kann die abstrakte Klasse *IfcExternalReference* bzw. deren Unterklasse *IfcLibraryReference* referenziert werden (siehe Abbildung 50). Diese besitzt die drei optionalen Attribute *Location*, *ItemReference* und *Name*. Über *Location* kann der Ort, an dem die Informationen zu finden sind, z.B. die RFEM-Materialbibliothek, angegeben werden. Über *ItemReference* kann ein konkretes Material in der Bibliothek ausgewählt werden, das *Name*-Attribut erlaubt dabei eine für Menschen verständliche Beschreibung des Materials.

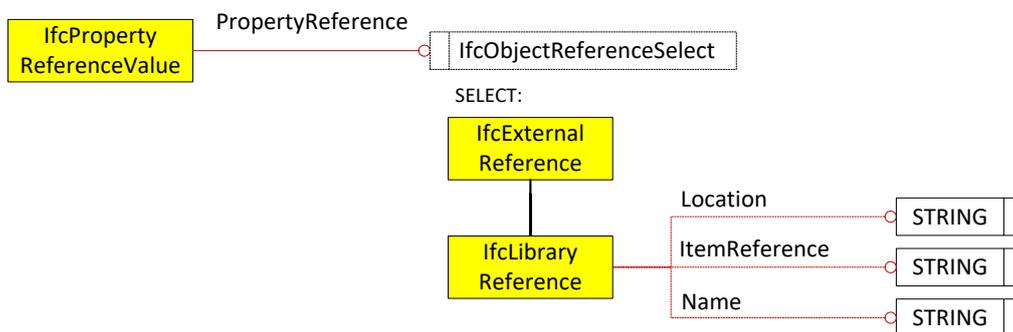


Abbildung 50: Schema zur Anbindung einer *IfcLibraryReference*

### Beschreibung benutzerdefinierter Materialien

Die Anzahl der zur Beschreibung eines Materials notwendigen Parameter ist vom gewählten Materialmodell abhängig. Tabelle 10 zeigt alle Informationen, die zur Beschreibung eines Materials nach dem orthotrop-elastischen Materialmodell (3D) benötigt werden.

Bei der Wahl des isotrop-linear-elastischen Materialmodells entfällt die separate Definition von E-Modul und Schubmodul für die drei Raumdimensionen; im Falle von "orthotrop-elastisch-2D" entfallen in z-Richtung sowohl der E-Modul als auch die Querdehnzahl (vgl. Kapitel 3.1).

Tabelle 10: Materialkennwerte benutzerdefinierter Materialien als IfcComplexProperty

IfcComplexProperty „Materialkennwerte“				
Name	Typ	Value	Unit	Description
Bezeichnung	SingleValue	String		
E <sub>x</sub>	SingleValue	Real	N/m <sup>2</sup>	E-Modul
E <sub>y</sub>				
E <sub>z</sub>				
G <sub>x</sub>	SingleValue	Real	N/m <sup>2</sup>	Schubmodul
G <sub>y</sub>				
G <sub>z</sub>				
v <sub>yz</sub>	SingleValue	Real		Querdehnzahl
v <sub>xz</sub>				
v <sub>xy</sub>				
γ	SingleValue	Real		Spezifisches Gewicht
α	SingleValue	Real		Wärmedehnzahl
γ <sub>M</sub>	SingleValue	Real		Teilsicherheitsbeiwert
Materialmodell	Enumerated	“Isotrop-linear-elastisch”; “orthotrop-elastisch-2D”; “orthotrop-elastisch-3D”;		

### 6.3 Mechanische Eigenschaften von stabartigen Elementen

Stabbogensegmente, Hänger und Stützen von Bogentragwerken sowie schlanke Pfeiler können in Tragwerksmodellen durch Stabelemente repräsentiert werden. Die mechanischen Eigenschaften der Bauelemente werden dabei über die Stabendgelenke der jeweiligen Stäbe definiert.

Im hier definierten Property-Set werden die beiden Stabtypen *Fachwerkstab* und *Balkenstab* zur Auswahl angegeben. Während im Falle des Fachwerkstabs alle Stabendgelenke bereits vordefiniert sind, können die Gelenkbedingungen für den Balkenstab einzeln festgelegt werden. Die mechanischen Eigenschaften von Seilstäben werden über ein separates Property-Set definiert, vgl. hierzu Abschnitt 6.7.

Da die über Stäbe modellierten Stützen, Hänger etc. zwar an den Verbindungsstellen als frei oder elastisch drehbar angenommen werden können, sich die Stabenden jedoch nicht voneinander lösen dürfen, wird über das Property-Set nur die Beschreibung rotatorischer Gelenkeigenschaften ermöglicht

IfcPropertySet „Stab_mechanisch“			
Name	Typ	Value	Description
Stabtyp	Enumerated	„Balkenstab“ „Fachwerkstab“	
Querschnittswerte	Complex / ReferenceValue	Siehe Tabelle 11 / Abbildung 50	
Lineare_Gelenkbedingungen_Rotation	ListValue	Real	1. Eintrag = $c_{\varphi x}(\varphi_x)$ , 2. Eintrag = $c_{\varphi y}(\varphi_y)$ , 3. Eintrag = $c_{\varphi z}(\varphi_z)$ , \$ = starr, 0 = frei drehbar

### 6.3.1 Querschnitte von Stabelementen

Um die mechanischen Eigenschaften von stabartigen Elementen wie Seilen, Hängern oder Stabbogensegmenten zu definieren, ist u. a. die Festlegung des Stabquerschnittes notwendig. Analog zum Material können Querschnitte entweder aus einer Querschnittsbibliothek ausgewählt oder vom Nutzer selbst definiert werden.

Tabelle 11 fasst die zur Definition eines nutzerdefinierten Standardquerschnitts notwendigen Informationen als Complex-Property zusammen (auf Sonderformen, wie beispielsweise zusammengesetzte Querschnitte sowie spezielle Eigenschaften bestimmter Querschnittstypen, wird an dieser Stelle nicht eingegangen.).

Benötigt werden die Trägheitsmomente sowie die Querschnitts- und Mantelfläche. Die Parameter  $A_y$  und  $A_z$  zur Berücksichtigung der Schubverformung gedrungener Stäbe können zur Beschreibung von Stabbogensegmente notwendig sein; bei schlanken Stäbe spielen sie in der Regel keine Rolle (Dlubal Software GmbH, 2016). Jedem benutzerdefinierten Querschnitt wird zudem eine frei wählbare Bezeichnung gegeben.

Tabelle 11: Querschnittswerte

IfcComplexProperty „Querschnittswerte“				
Name	Typ	Value	Unit	Description
Bezeichnung	SingleValue	String		
$I_T$	SingleValue	Real	$m^4$	Trägheitsmomente
$I_Y$				
$I_Z$				
A	SingleValue	Real	$M^2$	Querschnittsfläche
$A_Y$				Schubflächen
$A_Z$				
$A_S$	SingleValue	Real	$m^2/m$	Mantelfläche

Das IFC-Schema zur Referenzierung von Querschnitten aus einer Querschnittsbibliothek entspricht dem in Kapitel 0 beschriebenen Schema zur Referenzierung vordefinierter Materialien.

## 6.4 Fahrbahntragwerk

### Semantische Eigenschaften

Im Property-Set *Fahrbahntragwerk\_semantisch* werden die Querschnittsform des Fahrbahntragwerks und die bautechnische Struktur der Fahrbahnplatte festgelegt. (siehe Tabelle 12).

Tabelle 12: Semantische Eigenschaften des Fahrbahntragwerks

IfcPropertySet „Fahrbahntragwerk_semantisch“		
Name	Typ	Value
Querschnittsform	Enumerated	“Platte”, “Plattenbalken”, “Hohlkasten”, “Trog”;
BautechnischeStruktur Fahrbahnplatte	Enumerated	“Vollplatte”, “Hohlplatte”, “OrthotropePlatte”, “Zellenkasten”;

## 6.5 Vorspannung des Fahrbahntragwerks

### Vorspannung als Eigenschaft eines Bauteils

Im Hilfsschema ist Vorspannung nicht nur als eigenständiges Objekt, sondern in erster Linie als optionale Eigenschaft von Bauteilen definiert. Aus den über diese Eigenschaft referenzierten Informationen können bei Bedarf dreidimensionale Spannglieder als CAD-Objekte generiert werden. Hierfür wäre allerdings ein spezielles Konverter-Programm notwendig, das die über Parameter beschriebene Geometrie der Spannglieder automatisch in ein CAD-Objekt umwandelt.

Im Folgenden wird gezeigt, wie sich Vorspannung als eine Eigenschaft von Bauteilen beschreiben lässt. Struktur und Bezeichnung der geometrisch-konstruktiven Informationen orientieren sich dabei an den erforderlichen Eingabedaten des Beispielprogramms *RFEM*.

### Aufteilung in geometrisch-konstruktive und mechanische Eigenschaften

Die Beschreibung der Vorspannung gliedert sich in die beiden Teile „geometrisch-konstruktive Informationen“ zu den Spanngliedern sowie „mechanisch-statische Information“ zu den aus der Vorspannung resultierenden Ersatzlasten (siehe nachfolgende Tabelle).

Tabelle 13: Vorspannung als Eigenschaftsset

IfcPropertySet „Vorspannung“			
Name	Typ	Value	Description
Spannglieder	Complex	<i>Tabelle 15</i>	Pseudocode: Contains SET [1:?] OF IfcComplexProperty WHERE: Name of ComplexProperty (1,2,...,n) = (Spannglied_1, Spannglied_2,...,Spannglied_n)
Ersatzlasten	Complex	<i>Siehe Tabelle 20</i>	

Existieren in einem Bauteil mehrere Spannglieder, so wird jedes Spannglied als eigenes Complex-Property der Complex-Property „Spannglieder“ untergeordnet. Jede in einer komplexen Eigenschaft enthaltene Eigenschaft muss einen innerhalb dieses Complex-Property einzigartigen Namen haben (BuildingSmart, o.J.). Daher wird festge-

legt, dass der Name eines Spannglieds eine innerhalb des Property-Sets einmalige Nummer enthalten muss, über die die Spannglieder voneinander unterschieden werden können, also „Spannglied\_1“, Spannglied\_2“ etc. Das Complex-Property „Spannglied“ wird dabei als eine Art Container verwendet, der eine beliebige Anzahl weiterer komplexer Eigenschaften gleichen Typs enthalten kann (siehe Tabelle 14).

Alternativ wäre es auch möglich, alle Spannglieder direkt dem Property-Set „Vorspannung“ unterzuordnen; der Übersichtlichkeit halber wurde jedoch die Zusammenfassung in einem Complex-Property gewählt.

Tabelle 14: Container für alle konstruktiven Spannglieder

IfcComplexProperty „Spannglieder“		
Name	Typ	Value
Spannglied_1	Complex	<i>Siehe Tabelle 15</i>
Spannglied_2	Complex	
Spannglied_3	Complex	

### 6.5.1 Spannglied als konstruktives Element

Die Beschreibung eines Spannglieds als körperliches Bauteil enthält zum einen Informationen zum Spannglied selbst, also dessen Durchmesser, Material etc.

Zum anderen enthält sie die Darstellung des räumlichen Spanngliedverlaufs im vorgespannten Bauteil; dieser wird über seine Projektionen auf die xy-Ebene und die xz-Ebene dargestellt.

Tabelle 15: Spannglied als konstruktives Element

IfcComplexProperty „Spannglied_1“			
Name	Typ	Value	Description
SpanngliedDaten	Complex	<i>Siehe Tabelle 16</i>	
Verlauf_XY	Complex	<i>Siehe Tabelle 18</i>	Pseudocode: Contains SET [1:?] OF IfcComplexProperty WHERE: Name of ComplexProperty (1,2,...,n) = (Segment_1, Segment_2,...,Segment_n)
Verlauf_XZ	Complex	<i>Analog Tabelle 18</i>	Analog Verlauf_XY

### Spannglieddaten

Die Spannglieddaten enthalten alle Informationen zum Spannglied selbst, darunter die Anzahl der Litzen, den Durchmesser und das Material des Hüllrohrs sowie die Art der Vorspannung, aufgebrauchte Vorspannkraft und maximal zulässige Zugkraft im Spannglied.

Tabelle 16: Spannglied-Daten

IfcComplexProperty „SpanngliedDaten“			
Name	Typ	Value	Unit
LitzenAnzahl	SingleValue	Integer	
d_Hüllrohr	SingleValue	Real	mm
Material_Hüllrohr	Enumerated	“Metall”, “Kunststoff”;	
Vorspannkraft_vorh	SingleValue	Real	kN
Max_Zugkraft_zul	SingleValue	Real	kN
ArtDerVorspannung	Enumerated	“SofortigerVerbund”, “NachträglicherVerbund”;	

Zwar existiert in IFC2x3 zur Beschreibung von Spanngliedern bereits die Klasse *IfcTendon* als Unterklasse von *IfcReinforcingElement*; deren Attribute stimmen jedoch nicht vollständig mit den im Beispielprogramm *RFEM* benötigten Modellparametern überein (vgl. Anlage 6).

Zudem ist *IfcTendon* keine Unterklasse von *IfcProperty* und kann demnach nicht Teil eines Property-Sets sein. Auch eine Verbindung über *IfcPropertyReferenceValue* ist nicht möglich, da über diese Klasse keine Unterklassen von *IfcElement* referenziert werden können. *IfcTendon* wird daher im Rahmen dieser Arbeit nicht verwendet.

### Beschreibung des geometrischen Verlaufs

Der auf eine Ebene projizierte geometrische Verlauf wird über ein oder mehrere parametrische Segmente beschrieben. Da sich der Verlauf aus beliebig vielen Segmenten zusammensetzen kann, wird ebenfalls ein „Container“ eingeführt, in dem alle zu einem Spanngliedverlauf in der xy- bzw. xz-Ebenen gehörenden Segmente zusammengefasst werden. Die im Folgenden beschriebenen Segmenttypen sowie deren Parameter wurden aus dem Beispielprogramm *RFEM* übernommen.

Tabelle 17: Container für alle zu einem Verlauf gehörenden Segmente

IfcComplexProperty „Verlauf XY“		
Name	Typ	Value
Segment_1	Complex	<i>Siehe Tabelle 18</i>
Segment_2	Complex	
Segment_3	Complex	

Zunächst wird festgelegt, aus welchen Segmenttypen sich der Verlauf zusammensetzt, dazu stehen in *RFEM* sieben Typen zur Auswahl. Der jeweilige Typ kann über ein Enumerated-Value angegeben werden (Zur Definition der einzelnen Segmenttypen in *RFEM* siehe Anlage 5).

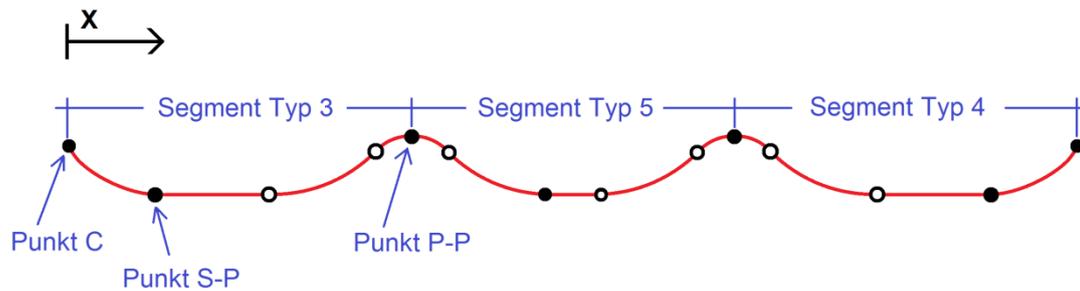


Abbildung 51: Beschreibung des Spanngliedverlaufs über parametrische Segmente  
 Quelle: in Anlehnung an (Diubal Software GmbH, 2016)

Position und Länge eines Segments, projiziert auf die lokale x-Achse des Trägers, werden über die x-Koordinaten seiner Start- und Endpunkte beschrieben. Diese werden über ein List-Value angegeben.

Der geometrische Verlauf der Segmente selbst ergibt sich aus dem jeweiligen Segmenttyp und der Festlegung mehrerer geometrischer Parameter.

Diese Parameter sind bestimmten Punkten auf den Segmenten zugeordnet, die Änderungen im Segmentverlauf markieren, also z. B. den Übergang von einem geraden Bereich in einen parabolischen.

Es existieren drei unterschiedliche Arten von Punkten:

- Anfangs- bzw. Endpunkte eines Segments (Punkt-Typ C)
- Verbindungspunkte zwischen einem geraden und einem parabolischen Segmentabschnitt (Punkt-Typ S-P)
- Verbindungspunkte zwischen zwei parabolischen Abschnitten (Punkt-Typ P-P)

Über die diesen Punkten zugeordneten Parameter lässt sich der Verlauf jedes Segmenttyps eindeutig beschreiben. Die Punkte enthalten dabei nicht nur Informationen zu ihrer eigenen Position im Sinne von x- und y-Koordinaten, über sie werden auch die Parameter zur Beschreibung des Verlaufs der angrenzenden Segmentbereiche festgelegt, also z.B. die Länge eines an einen Punkt angrenzenden geraden Abschnittes.

Tabelle 18: Verlauf in der xy-Ebene

IfcComplexProperty „Segment_1“			
Name	Typ	Value	Description
Position	List	Real	1.Eintrag=x-Koord.Anfang, 2.Eintrag=x-Koord.Ende
Segmenttyp	Enumerated	“1”,“2”,“3”,“4”,“5”,“6”,“7”;	
Anfangspunkt C	Complex	<i>Siehe Tabelle 19</i>	
Endpunkt C	Complex	<i>Analog Tabelle 19</i>	
Gerade-Parabel_S-P	Complex	<i>Siehe Anlage 4</i>	
Parabel-Parabel_P-P	Complex	<i>Siehe Anlage 4</i>	

## Beschreibung der Punkte und Verlaufparameter

Allgemein enthalten alle drei Punkttypen die Beschreibung der Lage des Punktes in vertikaler Richtung (y- bzw. z-Richtung) sowie die Festlegung eines Referenzpunktes, auf den sich diese Angabe bezieht.

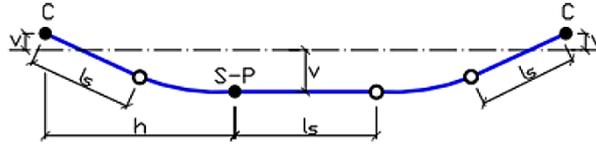


Abbildung 52: Parametrisches Segment aus RFEM

(Dlubal Software GmbH, 2016)

Je nach Punkt-Typ unterschiedlich beschrieben werden Segmentverlauf, Lage des Punktes bezogen auf die x-Achse sowie entsprechende Bezugspunkte.

### Anfangs- und Endpunkt Typ C

Im Folgenden wird exemplarisch beschrieben, wie sich die zur Beschreibung eines Punktes vom Typ C (Anfangs- bzw. Endpunkt) benötigten Parameter als komplexe Eigenschaft darstellen lassen.

Die Beschreibung der Zwischenpunkte P-S und P-P erfolgt nach dem gleichen Prinzip und findet sich in Anhang 4.

Die vertikale Position des Punktes wird über den Abstand zu einem Referenzpunkt als Single-Value angegeben. Zusätzlich muss die Länge des direkt an den Punkt C anschließenden geraden Segmentabschnitts festgelegt werden, was ebenfalls über ein Single-Value erfolgt (siehe Tabelle 19).

Tabelle 19: Anfangs/Endpunkt; Typ C

IfcComplexProperty „Anfangs/Endpunkt-C“			
Name	Typ	Value	Unit
V-Referenzpunkt	Enumerated	„Maximum_Z+“, „Randdurchdringung_Ze+“, „Bezugsachse_v“, „Schwerpunkt_Zcg“, „Randdurchdringung_Ze-“, „Minimum_Z-“;	
V-Abstand_RefP	SingleValue	Real	m
H-Gerade_Länge_L <sub>s,c</sub>	SingleValue	Real	m

Der Referenzpunkt zur Beschreibung der vertikalen Position kann aus sechs Typen ausgewählt werden: Der Punkt „Maximum\_Z+“ bzw. „Minimum\_Z-“ bezeichnet den höchsten bzw. niedrigsten Punkt eines Querschnitts. Der Punkt „Randdurchdringung\_Ze+“ die Stelle, an der eine durch das Spannglied verlaufende, senkrechte Linie die Oberseite des Querschnitts schneiden würde, „Randdurchdringung\_Ze-“ bezeichnet dementsprechend den Schnittpunkt dieser Linie mit der Unterseite des Querschnitts. Weitere Möglichkeiten sind der Schwerpunkt des Querschnitts sowie ein Punkt auf Höhe einer frei wählbaren Bezugsachse.

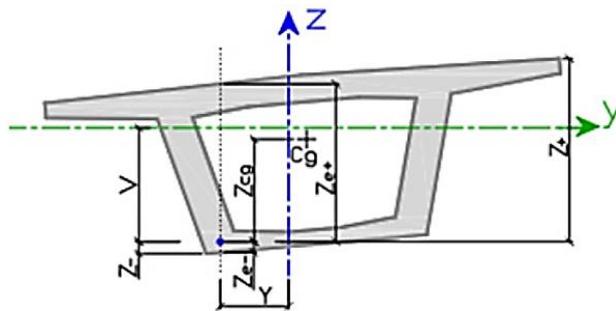


Abbildung 53: Referenzpunkte zur Beschreibung der Position in z-Richtung

Quelle: (Dlubal Software GmbH, 2016)

### 6.5.2 Darstellung der Ersatzlasten

Die aus der Berechnung der Vorspannung resultierenden Ersatzlasten können unabhängig von der konstruktiven Beschreibung der Spannglieder dargestellt werden.

Normalkraft in x-Richtung, Querkraft in y- und z-Richtung sowie Biegemomente um die y- und z-Achse werden jeweils über separate Table-Values dargestellt, ihr tatsächlicher Verlauf wird dabei über die Angabe mehrerer Punkte angenähert.

In der *Description* der Table-Value ist festgelegt, dass über die Liste der Defining-Values die lokale x-Koordinate eines Punktes angegeben wird; die Liste der Defined-Value beschreibt die zu den jeweiligen x-Koordinaten gehörenden Lastgrößen.

Tabelle 20: Ersatzlast

IfcComplexProperty „Ersatzlasten“						
Name	Typ	Defining Values	Defined Values	Defining Unit	Defined Unit	Description
$N_x$	TableValue	Real	Real	m	kN	DefiningValue = x-Koord. (lokal), DefinedValue = Lastgröße
$Q_y$	TableValue	Real	Real	m	kN	
$Q_z$	TableValue	Real	Real	m	kN	
$M_y$	TableValue	Real	Real	m	kNm	
$M_z$	TableValue	Real	Real	m	kNm	

## 6.6 Ausbauelemente

Da Ausbauelemente im Hilfsschema nicht notwendiger Weise eine geometrische Repräsentation besitzen müssen, sind in den im Folgenden beschriebenen Property-Sets auch einige Informationen zu deren Geometrie und Materialeigenschaften enthalten. Diese Angaben sollen es ermöglichen, die durch die Ausbauelemente entstehenden Lasten zumindest näherungsweise im Tragwerksmodell zu berücksichtigen. Zudem enthalten alle Property-Sets ein Single-Value, über das das entsprechende Ausbauelement in Textform beschrieben werden kann.

### Fahrbahn

Das Property-Set *Fahrbahn\_semantisch* beschreibt die Position der jeweiligen Fahrbahn in Bezug auf die Brückenlängsachse sowie deren Gesamtbreite über zwei Single-Values.

Die Beschreibung der in der schichtweise aufgebauten Fahrbahnkonstruktionen erfolgt über drei List-Values. Das erste List-Value enthält die Bezeichnung der Schichten, das zweite die zugehörigen Schichtdicken und über das dritte werden die entsprechenden Wichten angegeben. Die Bedeutung der einzelnen Listenpositionen ist in der jeweiligen *Description* festgelegt.

Tabelle 21: Eigenschaften der Fahrbahnausbildung

IfcPropertySet „Fahrbahn_semantisch“				
Name	Typ	Value	Unit	Description
Beschreibung	SingleValue	String		[0:1]
Position	SingleValue	Real	m	Abstand Mittelachse Fahrbahn zu Mittelachse Brücke
Breite	SingleValue	Real	m	
Bezeichnung-Schichten	ListValue	String		Bezeichnung Schichten 1 bis n
SchichtDicken	ListValue	Real	m	Dicke Schichten 1 bis n
SchichtWichten	ListValue	Real	kN/m <sup>3</sup>	Wichte Schichten 1 bis n

### Fahrbahnübergänge

Die semantischen Informationen zu Fahrbahnübergängen umfassen zum einen die möglichen Konstruktionsformen – auswählbar über ein Enumerated-Value – und zum anderen die für die Übergangskonstruktion maximal möglichen Verschiebungen und Verdrehungen. Diese Werte müssen bei der Dimensionierung der Brückenlager berücksichtigt werden.

Tabelle 22: Eigenschaften der Fahrbahnübergänge

IfcPropertySet „Fahrbahnübergang_semantisch“				
Name	Typ	Value	Unit	Description
Typ	Enumerated	„ElastischeFuge“, „Kissenkonstruktion“, „Fingerübergang“, „Lammellenübergang“, „Gleitplattenübergang“;		[1:1]
MaxVerschiebung	ListValue	Real	m	1.Eintrag = $U_{\max,X}$ , 2.Eintrag = $U_{\max,Y}$ , 3.Eintrag = $U_{\max,Z}$
MaxVerdrehung	ListValue	Real	Grad	1.Eintrag = $\varphi_{\max,X}$ , 2.Eintrag = $\varphi_{\max,Y}$ , 3.Eintrag = $\varphi_{\max,Z}$

### Kappen

Ob die Kappen eines Brückendecks als Notweg, als Fußgängerweg oder als Fuß- und Radweg genutzt werden, wird über ein Enumerated-Value angegeben.

Durch die Angabe von Breite, Höhe und Wichte sind alle Informationen gegeben, um die Kappen im Tragwerksmodell näherungsweise als Ausbaulasten zu berücksichtigen. Es wird dabei davon ausgegangen, dass die Kappen rechts und links von der Fahrbahn symmetrisch aufgebaut sind.

Tabelle 23: Eigenschaften der Kappen

IfcComplexProperty „Kappen_semantisch“			
Name	Typ	Value	Unit
Beschreibung	SingleValue	String	
Funktion	Enumerated	„Notweg“, „Fußweg“, „FußUndRadweg“;	
Breite	SingleValue	Real	m
Höhe	SingleValue	Real	m
Wichte	SingleValue	Real	kN/m <sup>3</sup>

### Lärmschutzanlage

Das Property-Set zur Beschreibung der Lärmschutzanlagen enthält Informationen zum Schallabsorptionsvermögen der Anlage, angegeben in Dezibel.

Zudem sind die durch die Anlage entstehenden Ausbaulasten über eine äquivalente Linienlast sowie den Abstand der Anlage zur Längsachse der Brücke gegeben, dabei wird von einem geraden Verlauf der Schallschutzkonstruktion sowie von einem symmetrischen Aufbau auf beiden Brückenlängsseiten ausgegangen. Über die Angabe der Höhe kann die durch die Konstruktion erhöhte Windangriffsfläche näherungsweise berücksichtigt werden.

Tabelle 24: Eigenschaften der Lärmschutzanlage

IfcComplexProperty „Lärmschutzanlage_semantisch“			
Name	Typ	Value	Unit
Beschreibung	SingleValue	String	
Schallabsorptionsvermögen	SingleValue	Real	dB
ÄquivalenteLinienlast	SingleValue	Real	kN/m
AbstandZurBrückenlängsachse	SingleValue	Real	m
Höhe	SingleValue	Real	m

### Leiteinrichtungen

Da das Gewicht von Leiteinrichtungen im Verhältnis zum Rest des Brückendecks in der Regel sehr gering ist, wird es an dieser Stelle nicht berücksichtigt. Lediglich die je nach Konstruktionsform nicht unerhebliche Windangriffsfläche kann über ein Single-Value in „Quadratmeter Angriffsfläche pro Meter in Brückenlängsrichtung“ angegeben werden.

Tabelle 25: Eigenschaften von Leiteinrichtungen

IfcComplexProperty „Leiteinrichtungen_semantisch“				
Name	Typ	Value	Unit	Description
Beschreibung	SingleValue	String		
Windangriffsfläche	SingleValue	Real	m <sup>2</sup> /m	Erhöhung der Windangriffsfläche in m <sup>2</sup> je m bezogen auf die Brückenlängsrichtung

### Beleuchtung

Beleuchtungsanlagen können zwar geometrisch modelliert und über ihre Beziehung zur Brückenausrüstung in der Hierarchie des Gesamtbauwerks eingeordnet werden, sie besitzen jedoch keine für das hier entworfene Brückenmodell relevanten Eigenschaften.

## 6.7 Schrägseiltragwerke

Die für die Torsionssteifigkeit einer Seilbrücke entscheidende Anzahl der Seilebenen wird über ein Single-Value angegeben, die Anordnung der Schrägkabel wird in dem Enumerated-Value „Kabelsystem“ festgelegt.

Da zu jedem Schrägseiltragwerk nach der Definition des Hilfsschemas genau ein Pylon gehört und Pylonen als einzige Information der „Pylontyp“ zugewiesen wird, wird dieser bereits im Property-Set des Schrägseiltragwerks festgelegt.

Tabelle 26: Semantische Eigenschaften von Schrägseiltragwerken

IfcPropertySet „Schrägseiltragwerk_semantisch“		
Name	Typ	Value
AnzahlSeilebenen	SingleValue	Integer
Kabelsystem	Enumerated	„Bündelsystem“, „Harfensystem“, „Fächersystem“;
Pylontyp	Enumerated	„Freitragend“, „A-Pylon“, „H-Pylon“;

## 6.7.1 Seile

### Semantische Eigenschaften

Das Property-Set zur Beschreibung der semantischen Eigenschaften von Schrägseilen umfasst die Seilart, den vorhandenen Korrosionsschutz sowie eventuell mit dem Seil verbundene Schwingungsdämpfungs-Systeme.

Tabelle 27: Semantische Eigenschaften von Schrägseilen

IfcPropertySet „Seil_semantisch“			
Name	Typ	Value	Descr.
Seilart	Enumerated	„StabbündelAusEinzellitzen“, „Einzelstab“, „Parallellitzenseil“, „Paralleldrahtseil“, „Rundlitzenseil“, „VollverschlossenesSpiralseil“, „OffenesSpiralseil“, „Nutzerdefiniert“, „NichtDefiniert“,	
Korrosionsschutz	Enumerated	„verzinkt“, „Anstrich“, „Hüllrohr“, „gefettet(temporär)“, „Nutzerdefiniert“, „NichtDefiniert“;	[0:?]
Schwingungsdämpfung	Enumerated	„keine“, „ZementmörtelHüllrohr“, „Neoprene-Ringe“, „Stoßdämpfer“, „Stabilisierungsseile“, „Nutzerdefiniert“, „NichtDefiniert“;	[0:?]

### Mechanische Eigenschaften

In Statikprogrammen können Seile modelliert werden, indem dem Geometrieobjekt „Seil“ ein Stab vom Typ „Seilstab“ zugewiesen wird. Der „Seilstab“ enthält bereits alle notwendigen Informationen zu Freiheitsgraden, nichtlinearen Gelenkeigenschaften sowie Berechnungsmodellen (vgl. Kapitel 3.3). Dem Stab wird ein Querschnitt zugewiesen und dem Querschnitt wiederum ein Material.

Tabelle 28 fasst alle Informationen zusammen, die zur Beschreibung der mechanischen Eigenschaften eines Seil-Elements in einem Statikprogramm benötigt werden. Der auszuwählende Stabtyp „Seilstab“ wird als einziger Wert eines Enumerated-Values angegeben. Die Informationen zu Querschnitt und Material können jeweils entweder in Form einer Complex-Property explizit gegeben werden oder es kann über ein Reference-Value auf vordefinierte Querschnitte bzw. Materialien verwiesen werden (vgl. Abschnitt 0 bzw. 6.3.1). Die maximal zulässige Belastung wird als Single-Value angegeben.

Tabelle 28: Mechanische Eigenschaften von Seilen

IfcPropertySet „Seil_mechanisch“		
Name	Typ	Value
Stabtyp	Enumerated	„Seilstab“
Querschnittswerte	Complex/ ReferenceValue	<i>Siehe Tabelle 11</i>
Bruchlast	SingleValue	Real

## 6.7.2 Vorspannung von Seilen

Vorspannkraft von Seilen kann durch die Zuweisung einer Stabkraft vom Typ „Vorspannung“ auf den zuvor erzeugten Seilstab modelliert werden.

Anders als Stabendgelenke, Material und Querschnitt werden Stabkräfte im Beispielprogramm *RFEM* nicht als Attribute der Entität „Stab“ dargestellt. Sie bilden vielmehr eigenständige Entitäten, die einem oder mehreren Stäben zugeordnet werden können. Daher wird die Vorspannkraft im Folgenden nicht als eine weitere komplexe Eigenschaft des Property-Sets „Stab“ beschrieben, sondern als ein selbstständiges Property-Set mit dem Namen *Vorspannung\_Seil* (siehe Tabelle 29).

Die Beschreibung von Seileigenschaften und Seilvorspannung in getrennten Property-Sets hat zudem den Vorteil, dass Seilinstanzen mit unterschiedlicher Vorspannkraft, aber ansonsten identischen Eigenschaften, mit denselben Instanzen von *Seil\_semantisch* und *Seil\_mechanisch* verbunden werden können. Dadurch wird die mehrfache Beschreibung identischer Information vermieden (siehe nachfolgende Abbildung).

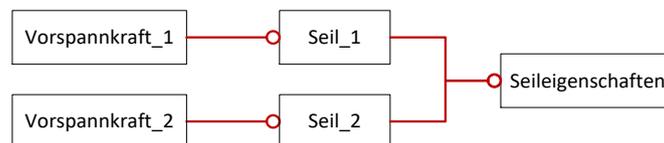


Abbildung 54: Separate Anbindung von Vorspannung und Seilstabeigenschaften

Lastverteilung und Lastrichtung müssen nicht explizit angegeben werden, da Vorspannung von Stäben in *RFEM* als eine konstante Last entlang der lokalen x-Achse definiert ist.

Tabelle 29: Vorspannung von Seilen

IfcPropertySet „Vorspannung_Seil_mechanisch“			
Name	Typ	Value	Unit
Lastart	Enumerated	„Anfangsvorspannung“ „Endvorspannung“,	
Betrag	SingleValue	Real	kN

## 6.8 Bogentragwerke

Ob es sich bei einem Bogentragwerk um ein „echtes Bogentragwerk“ handelt oder ob das Fahrbantragwerk als Zugband fungiert und einen Teil der Horizontalkräfte aufnimmt, wird über ein Single-Value angegeben, das die Werte „wahr“ und „falsch“ annehmen kann. Der Wert „FALSE“ bedeutet dabei, dass es sich um ein Bogentragwerk mit Zugband handelt.

Tabelle 30: Eigenschaften von Bogentragwerken

IfcPropertySet „Bogentragwerk_semantisch“			
Name	Typ	Value	Description
EchtesBogentragwerk	SingleValue	Boolean	Wenn FALSE dann Bogen mit Zugband

### Mechanische Eigenschaften von Stabbogensegmenten, Hängern und Stützen

Stabbogensegmente, Hänger und Stützen von Bogentragwerken können in Tragwerksmodellen durch Stabelemente repräsentiert werden. Die mechanischen Eigenschaften werden über das in Kapitel 6.3 beschriebene Property-Set *Stab\_mechanisch* festgelegt.

Falls das Fahrbahntragwerk über Seile am Bogen aufgehängt ist, werden deren Eigenschaften über das im Vorangegangenen erläuterte Property-Set *Seil\_mechanisch* beschrieben.

## 6.9 Unterbau

### Pfeiler

Im Property-Set zu den semantischen Eigenschaften des Unterbaus wird zum einen festgelegt, ob es sich bei einem Pfeiler um ein schlankes Bauteil handelt und daher im Tragwerksmodell hinreichend genau als Stabelement modelliert werden kann. Zum anderen wird angegeben, ob es sich bei dem Pfeiler um einen Pendelpfeiler handelt oder nicht. Beide Informationen werden als Single-Value mit dem Wertebereich TRUE oder FALSE angegeben.

IfcPropertySet „Pfeiler_semantisch“			
Name	Typ	Value	Description
Stabartig	SingleValue	Boolean	Wenn FALSE dann „Kompakt“
IstPendelpfeiler	SingleValue	Boolean	

Falls es sich bei dem Pfeiler um ein stabartiges Bauteil handelt, werden dessen mechanische Eigenschaften über das zuvor beschriebene Property-Set *Stab\_mechanisch* definiert.

### Gründung

Ob es sich bei einer Gründung um eine Flachgründung oder eine bestimmte Art von Tiefgründung handelt, wird über das Enumerated-Value *Gründungstyp* angegeben.

IfcPropertySet „Gründung_semantisch“		
Name	Typ	Value
Gründungstyp	Enumerated	„Flachgründung“, „Pfahlgründung“, „Brunnengründung“, „Senkkasten“;

## 6.10 Lager

Da sich Punkt- und Linienlager sowohl in den zur Auswahl stehenden Konstruktionsformen als auch in den möglichen Verdrehungsfreiheitsgraden voneinander unterscheiden, werden für die beiden Klassen Punkt- und Linienlager jeweils getrennte Property-Sets definiert.

### 6.10.1 Punktlager

#### Semantische Eigenschaften

Über die semantischen Eigenschaften von Lagern kann festgelegt werden, um welche Art von Punktlager es sich handelt und ob das Lager an dem angrenzenden Bauteil fest verankert ist. Zur Auswahl stehen die in Kapitel 5.6.7 erläuterten Konstruktionsformen.

Zudem werden die Grenzwerte für Belastung und Verformung auch im Property-Set *Punktlager\_semantisch* beschrieben, da sie sich auf der Ebene der zur automatischen Interpretation von Statikprogrammen wie *RFEM* gedachten mechanischen Eigenschaften nicht vollständig darstellen lassen. Dieser Punkt wird im folgenden Abschnitt zu den mechanischen Lagereigenschaften noch genauer erläutert.

Tabelle 31: Semantische Eigenschaften von Punktlagern

IfcPropertySet „Punktlager_semantisch“				
Name	Typ	Value	Unit	Description
Lagertyp	Enumerated	„Verformungslager“, „Stahllager“, „Topflager“, „Kalottenlager“, „Horizontalkraftlager“;		
Verankert	SingleValue	Boolean		
Max_Kräfte	ListValue	Real	kN	1. Eintrag = $P_x$ , 2. Eintrag = $P_y$ , 3. Eintrag = + $P_z$ , 4. Eintrag = - $P_z$ ;
Max_Verschiebungen	ListValue	Real	m	1. Eintrag = $U_x$ , 2. Eintrag = $U_y$ , 3. Eintrag = + $U_z$ , 4. Eintrag = - $U_z$ ;
Max_Verdrehungen	ListValue	Real	rad	1. Eintrag = $\varphi_x$ , 2. Eintrag = $\varphi_y$ , 3. Eintrag = $\varphi_z$ ;

## Mechanische Eigenschaften

Die mechanischen Eigenschaften von Punktlagern umfassen die durch das Lager ermöglichten linearen und nichtlinearen Bewegungsfreiheitsgrade. Struktur und Benennung der Eigenschaften orientieren sich dabei – wie schon in den vorangegangenen Abschnitten – weitgehend am Beispielprogramm *RFEM*.

### Knotenfreigabe

Die Beschreibung einer beweglichen Verbindung von zwei Körpern erfolgt in *RFEM* über die Einführung einer sogenannten „Knotenfreigabe“. Die Freiheitsgrade des freigegebenen Knotens werden dabei als Gelenkbedingungen bezeichnet.

Dabei muss zunächst angegeben werden, welche zwei Körper „freigegeben“ werden sollen und über welchen Verbindungsknoten die Freigabe erfolgt. Diese Informationen sind über die im Hilfsschema festgelegten Beziehungen zwischen dem Punktlager und den beiden Lagerhälften, aus denen es sich zusammensetzt, wie folgt gegeben (vgl. Abschnitt 5.6.7):

Das Property-Set *Punktlager\_mechanisch* ist über *RelDefinesByProperties* mit der Punktlager-Instanz verbunden. Die beiden Lagerhälften sind über *IfcRelAggregates* als Bestandteile des Punktlager-Proxys definiert und über die Klasse *IfcRelConnectsElements* explizit miteinander verbunden. Das Attribut *ConnectionPointGeometry* definiert dabei den Verbindungsknoten.

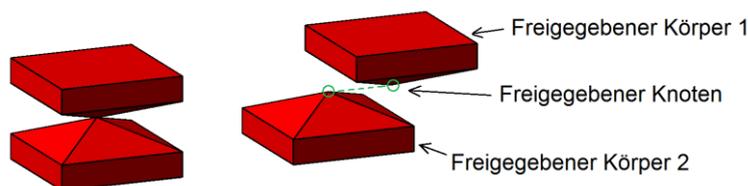


Abbildung 55: Punktlager mit freigegebenem Verbindungsknoten

### Lineare Gelenkbedingungen

Die Beschreibung der linearen Freiheitsgrade erfolgt über die Klasse *IfcPropertyListValue*. Welche Position der Listen welche Information repräsentiert, wird ebenfalls über die Relation *Description* angegeben (siehe Tabelle 32).

Auf diese Weise wird festgelegt, dass Position eins in der Werte-Liste von *Lineare\_Gelenkbedingungen\_Translation* die Federsteifigkeit bei Verschiebung in x-Richtung, Position zwei und drei dem entsprechend die Federsteifigkeiten in Richtung y und z beschreibt.

Des Weiteren wird definiert, dass „kein Eintrag“ für eine starre Verbindung, der Wert „Null“ für freie Beweglichkeit und alle Werte größer Null für eine reale Federsteifigkeit stehen. Über die Relation *Unit* wird die zugehörige Einheit kN/m festgelegt.

Die Beschreibung aller sechs linearen Freiheitsgrade und deren zugehöriger Einheit über ein einziges *IfcPropertyListValue* ist nicht möglich, da jeder Instanz nur eine Ein-

heit zugewiesen werden kann. Translatorische und rotatorische Federsteifigkeiten besitzen jedoch unterschiedliche Einheiten; daher werden die Verdrehungsfreiheitsgrade über eine zweite Instanz von *IfcPropertyListValue* beschrieben.

Tabelle 32: Knotenfreigabe

IfcPropertySet „Punktlager_mechanisch“				
Name	Typ	Value	Unit	Description
Bezugsachsensystem	Enumerated	„lokal“, „global“		
Lineare_Gelenkbedingungen_Translation	ListValue	Real	kN/m	1. Eintrag = $c_x(u_x)$ , 2. Eintrag = $c_y(u_y)$ , 3. Eintrag = $c_z(u_z)$ , \$ = starr
Lineare_Gelenkbedingungen_Rotation	ListValue	Real	kNm/rad	1. Eintrag = $c_{\varphi_x}(\varphi_x)$ , 2. Eintrag = $c_{\varphi_y}(\varphi_y)$ , 3. Eintrag = $c_{\varphi_z}(\varphi_z)$ , \$ = starr
Nichtlineare_Gelenkbedingungen	Complex	Siehe Tabelle 33		[optional]

### Nichtlineare Gelenkbedingungen über *IfcComplexProperty*

Für die Darstellung einiger Lager kann auch die Beschreibung nichtlinearer Gelenkeigenschaften notwendig sein. Dies ist über *IfcPropertyListValue* nicht mehr auf übersichtliche Weise möglich, eine Darstellung über ein *IfcComplexValue* ist hierfür besser geeignet.

Jedem Verschiebungs- bzw. Verdrehungsfreiheitsgrad kann eine nichtlineare Gelenkbedingung zugewiesen werden. In einigen Fällen ergänzt diese Bedingung die entsprechende lineare Federsteifigkeit. So kann beispielsweise zusätzlich zu einer linearen translatorischen Federsteifigkeit in x-Richtung eine nichtlineare Bedingung festgelegt werden, die besagt, dass die Verbindung bei Druck aus negativer x-Richtung starr bleibt. Somit wäre die Verbindung entlang der x-Achse nur noch in eine Richtung verschieblich.

Andere nichtlineare Verbindungseigenschaften ersetzen die linearen Steifigkeiten vollständig. So beschreibt z. B. die im Beispielprogramm *RFEM* mit *Diagramm* bezeichnete Eigenschaft einen nichtlinearen Zusammenhang zwischen der Größe der Verschiebung und der daraus resultierende Rückstellkraft.

Tabelle 33: Nichtlineare Gelenkeigenschaften

IfcComplexProperty „Nichtlineare_Gelenkbedingungen“			
Name	Typ	Value	Description
NL_u <sub>x</sub>	Complex	Siehe Tabelle 34	[optional]
NL_u <sub>y</sub>	Complex	Analog NL_u <sub>x</sub>	[optional]
NL_u <sub>z</sub>	Complex	Analog NL_u <sub>x</sub>	[optional]
NL_φ <sub>x</sub>	Complex	Siehe Tabelle 35	[optional]
NL_φ <sub>y</sub>	Complex	Analog NL_φ <sub>x</sub>	[optional]
NL_φ <sub>z</sub>	Complex	Analog NL_φ <sub>x</sub>	[optional]

Über die komplexe Eigenschaft *Nichtlineare\_Gelenkbedingungen* kann jedem Freiheitsgrad optional eine nichtlineare Eigenschaft zugewiesen werden (siehe Tabelle 33), wobei jede dieser Eigenschaften selbst wiederum als komplexe Eigenschaft dargestellt wird.

### Nichtlineare Verschiebungseigenschaften

Alle zur Beschreibung der translatorischen Eigenschaften definierten Complex-Propertys haben die gleiche Struktur. Voneinander unterschieden werden sie lediglich durch ihren Namen, über den sie dem jeweiligen Freiheitsgrad zugeordnet werden.

In der zu einem bestimmten Freiheitsgrad gehörenden komplexen Eigenschaft wird über ein Enumerated-Value festgelegt, um welche Art von nichtlinearer Eigenschaft es sich handelt.

Einige nichtlineare Gelenkbedingungen müssen über zusätzliche Informationen noch genauer definiert werden. In der jeweiligen *Description* der dafür eingeführten Complex-Properties ist festgelegt, dass nur die zum instanziierten Gelenkbedingungen-Typ passende Spezifikation instanziiert werden kann (siehe nachfolgende Tabelle). Dieser Umweg über optionale komplexe Eigenschaften ist notwendig, da über den Wert eines Enumerated-Values nicht direkt auf ein Complex-Property verwiesen werden kann. (*IfcPropertyEnumeratedValue* lässt nur Werte vom Typ *IfcValue* zu, also einfache Typen wie String oder Real.)

Tabelle 34: Nichtlineare Verschiebungseigenschaften in x-Richtung

IfcComplexProperty „NL_u <sub>x</sub> “			
Name	Typ	Value	Description
Typ	Enumerated	„FestBeiDruck“; „FestBeiZug“; „Diagramm“; „MaxVerschiebung“ „FließenUndSchlupf“	
Spezifikation_ Diagramm	Complex		Entfällt wenn Typ ≠ Diagramm
Spezifikation_ MaxVerschiebung	Complex		Entfällt wenn Typ ≠ Max- Verschiebung
Spezifikation_ FließenUndSchlupf	Complex		Entfällt wenn Typ ≠ Fließen- UndSchlupf

### Nichtlineare Verdrehungseigenschaften

Die Property-Sets zur Beschreibung der nichtlinearen Verdrehungsfreiheitsgrade unterscheiden sich geringfügig von denen zur Beschreibung der Verschiebungsfreiheitsgrade, da die zur Auswahl stehenden Eigenschafts-Typen teilweise anders benannt sind. Aus Zug- und Druckkräften werden positive und negative Momente, aus Verschiebungen werden Verdrehungen, wie in der folgenden Tabelle zu sehen ist.

Tabelle 35: Nichtlineare Verdrehungseigenschaften

IfcComplexProperty „NL_φx“			
Name	Typ	Value	Description
Typ	Enumerated	„FestWennMomentPositiv“; „FestWennMomentNegativ“; „Diagramm“; „MaxVerdrehung“ „FließenUndSchlupf“	
Spezifikation_ Diagramm	Complex		Entfällt wenn Typ ≠ Diagramm
Spezifikation_ MaxVerdrehung	Complex		Entfällt wenn Typ ≠ Max- Verdrehung
Spezifikation_ FließenUndSchlupf	Complex		Entfällt wenn Typ ≠ Fließen- UndSchlupf

### Spezifikation bestimmter Gelenkeigenschaften

Im Folgenden werden die zur Spezifikation der nichtlinearen Verschiebungseigenschaften notwendigen Property-Sets vorgestellt. Die zur Beschreibung der nichtlinearen Verdrehungseigenschaften benötigten Property-Sets sind analog aufgebaut, der Unterschied besteht lediglich darin, dass sie sich auf Momente an Stelle von Kräften und auf Verdrehung anstelle von Verschiebung beziehen; sie werden daher nicht gesondert dargestellt.

### Diagramm

Durch die nichtlineare Gelenkbedingung *Diagramm* können nichtlineare Federsteifigkeiten beschrieben werden, wobei der tatsächliche Kraft-Verschiebungs-Verlauf über die Angabe mehrerer Verschiebungspunkte und deren zugehöriger Rückstellkräfte angenähert wird.

Die hierfür notwendige Menge von Punkten kann über *IfcPropertyTableValue* angegeben werden. Auch wenn diese Klasse in erster Linie dafür gedacht ist, Wertebereiche zu definieren, kann sie auch genutzt werden, um eine Reihe von Punkten in einem zweidimensionalen Koordinatensystem darzustellen. Der Wert in der Spalte *Defining-Value* steht für die x-Koordinate, der Wert in der Spalte *DefinedValue* für die zugehörige y-Koordinate.

Zusätzlich zum Verschiebungs-Kraft-Verlauf selbst muss angegeben werden, was nach Erreichen der größten, über einen Diagrammpunkt definierten, Verschiebung passiert. Die zur Auswahl stehenden Möglichkeiten „Reißen“, „Fließen“, „Fortlaufend“ und „Anschlag“ können in einem *IfcPropertyEnumeratedValue* gegeben werden.

Tabelle 36: Spezifikation nichtlinearer Federsteifigkeit

IfcComplexProperty „Spezifikation_Diagramm“			
Name	Typ	Values	Description
Verlauf_nach_letztem_Schritt	Enumerated	„Reißen“; „Fließen“; „Fortlaufend“; „Anschlag“;	Ersetzt lineare Freiheitsgrade
Diagramm_Werte	TableValue		DefiningValues = Auslenkung, DefinedValues = Rückstellkraft

### Konflikt von *Diagramm* mit linearen Federsteifigkeiten

*Diagramm* ist die einzige hier aufgeführte nichtlineare Eigenschaft, die die lineare Federsteifigkeit nicht ergänzt, sondern vollständig ersetzt. Dies wird in der *Description* des zugehörigen Enumerated-Values festgelegt, damit es nicht zu Unklarheiten bei der Interpretation kommt. Folgendes Beispiel soll den andernfalls entstehenden Widerspruch verdeutlichen:

Die Werte des List-Values, das die linearen Steifigkeiten beschreibt, können nur Werte vom Typ REAL oder den Wert „\$“ annehmen. In der zugehörigen *Description* ist jedoch bereits festgelegt, dass „\$“ für eine unendliche hohe Steifigkeit, also für „starr“ steht. Demnach kann dieser Wert nicht mehr im Sinne von „keine Angabe“ interpretiert werden. Keine Angabe über einen bestimmten linearen Freiheitsgrad zu machen, ist in dieser Form somit nicht möglich.

Nimmt das List-Value *Lineare\_Gelenkbedingungen\_Translation* z.B. die Werte (\$,\$,\$) an, würde das für sich genommen bedeuten, dass die Verbindung in alle Richtungen unverschieblich ist. Wird nun durch die komplexe Eigenschaft „NL\_u<sub>x</sub>“ festgelegt, dass die Verbindung in x-Richtung gegen eine nichtlineare Rückstellkraft verschieblich ist, würde diese Angabe im Widerspruch zu der Angabe stehen, die Verbindung sei allseitig starr. Daher muss explizit festgelegt werden, welche der beiden Angaben gültig ist.

### Maximal mögliche Verschiebung

Diese Eigenschaft kann verwendet werden, um abzubilden, dass ein Lager nur über eine begrenzte Strecke verschieblich ist; darüber hinaus ist es starr. Da ein Lager nicht notwendigerweise in beide Richtungen gleich weit verschieblich sein muss bevor es anschlägt, können die maximal möglichen Verschiebungen in positiver und negativer Koordinatenrichtung getrennt angegeben werden. Verwendet wird dafür der Typ Single-Value.

Tabelle 37: Spezifikation der maximal möglichen Verschiebung

IfcComplexProperty „Spezifikation_MaxVerschiebung“			
Name	Typ	Values	Unit
InPositiveKoordinatenrichtung	SingleValue	Real	m
InNegativeKoordinatenrichtung	SingleValue	Real	m

## Fließen und Schlupf

Über „Fließen und Schlupf“ können zwei voneinander unabhängige Eigenschaften angegeben werden. Da einem Freiheitsgrad im Beispielprogramm *RFEM* jedoch beide Eigenschaften gleichzeitig zugewiesen werden können, werden sie in einer gemeinsamen Complex-Property aufgeführt.

„Fließen“ beschreibt, ab welcher Belastung eine Verbindung versagt; „Schlupf“ hingegen gibt an, wie weit eine Verbindung in positive- bzw. negative Koordinatenrichtung frei bewegt werden kann, bevor sie einen Widerstand durch die Rückstellkräfte der zugehörigen linearen Federsteifigkeit erfährt.

Tabelle 38: Spezifikation von Fließen und Schlupf

IfcComplexProperty „Spezifikation_FlößenUndSchlupf“			
Name	Typ	Values	Unit
FließenAbDruck	SingleValue	Real	kN
FließenAbZug	SingleValue	Real	kN
SchlupfinPositiveRichtung	SingleValue	Real	m
SchlupfinNegativeRichtung	SingleValue	Real	m

### Separate Angabe der Grenzwerte in den semantischen Eigenschaften

Da im Beispielprogramm *RFEM* jedem Freiheitsgrad nur eine der oben beschriebenen fünf nichtlinearen Eigenschaften zugewiesen werden kann, ist es nicht möglich, auf diese Weise gleichzeitig die maximal zulässigen Kräfte und die maximal möglichen Verformungen eines Lagers festzulegen. Daher werden diese Informationen noch einmal separat als Teil der semantischen Lagereigenschaften aufgeführt.

### Schachtelung der Gelenkeigenschaften – Zusammenfassung

Zur besseren Übersichtlichkeit ist die Schachtelung der zur Beschreibung der Gelenkeigenschaften notwendigen Informationen in Abbildung 56 noch einmal zusammengefasst.

Das Property-Set *Punktlager\_mechanisch* enthält alle linearen Gelenkeigenschaften in Form von zwei List-Values, eventuelle nichtlineare Eigenschaften sind unter der komplexen Eigenschaft *Nichtlineare\_Gelenkbedingungen* zusammengefasst.

Obwohl es auch möglich wäre, alle sechs möglichen nichtlinearen Freiheitsgrade direkt dem Property-Set unterzuordnen, wurden, um eine klare Struktur zu wahren, alle in einer komplexen Eigenschaft zusammengefasst. Die konkreten Eigenschaften eines nichtlinearen Freiheitsgrades werden in den jeweiligen, auf einen bestimmten Freiheitsgrad bezogenen, Complex-Properties definiert.

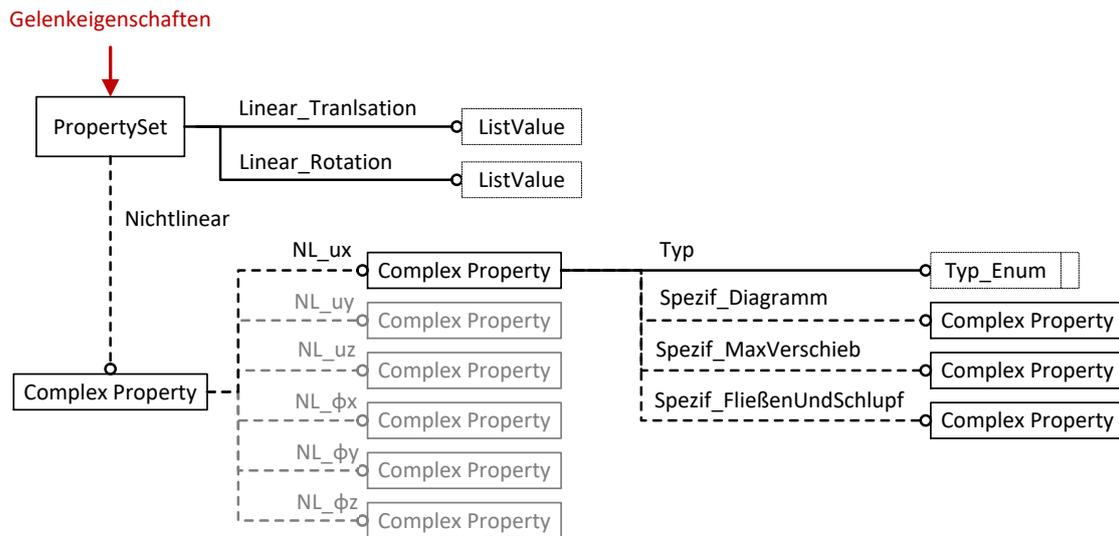


Abbildung 56: Schachtelung der Informationen zur Beschreibung der Gelenkeigenschaft

## 6.10.2 Linienlager

Die Beschreibung von Linienlagern erfolgt nach dem gleichen Prinzip wie die der Punktlager. Der grundlegende Unterschied besteht darin, dass die beiden Lagerhälften nicht punkt- sondern linienförmig miteinander verbunden sind, weshalb die Beschreibung der mechanischen Eigenschaften im Fall von Linienlagern über die Zuweisung einer Linienfreigabe anstelle einer Knotenfreigabe erfolgt.

Hinsichtlich der Gelenkeigenschaften unterscheiden sich Linienlager von Punktlagern durch die Reduzierung der möglichen Verdrehungsfreiheitsgrade von drei auf einen sowie durch die veränderten Einheiten der Steifigkeiten. Die Einheiten beziehen sich bei Linienfreigaben neben der Größe der Verschiebung bzw. Verdrehung auch auf die Länge der entsprechenden Linie, aus kN/m wird (kN/m)/m, aus kNm/rad wird (kNm/rad)/m. Die über *Unit* referenzierte Einheit muss dementsprechend angepasst werden.

Zur Darstellung des einen verbliebenen linearen Rotationsfreiheitsgrades bietet sich an Stelle des List-Value ein Single-Value an, da nur noch ein Zahlenwert festgelegt werden muss.

Auch auf der semantischen Ebene entfällt die Möglichkeit der Drehung um die lokalen y- und z-Achse. Die zur Auswahl stehenden Lagertypen reduzieren sich daher auf die Typen Stahllager und Horizontalkraftlager.

## 6.11 Abhängigkeiten von Eigenschaften

In der Realität können sich einige der im Vorangegangenen beschriebenen Eigenschaften gegenseitig bedingen. Über die Klasse *IfcPropertyDependencyRelationship* können diese Abhängigkeiten im Modell abgebildet werden. Diese Klasse gibt an, welche Eigenschaft von welcher anderen Eigenschaft abhängt und wie diese Abhängigkeit aussieht.

Die Zuordnung der „abhängigen“ und der „steuernden“ Eigenschaft erfolgt über die Relationen *DependantProperty* und *DependingProperty*, über die auf konkrete Instanzen von *IfcProperty* verwiesen wird.

Nachfolgend wird eine solche Abhängigkeit exemplarisch am Beispiel von zwei Lager-eigenschaften aufgeführt. Die Art der Abhängigkeit wird dabei einmal in Textform, über das Attribut *Description*, und einmal in Form von Pseudocode, über das Attribut *Expression*, beschrieben. Um einen allgemeinen schematischen Zusammenhang darzustellen, wird in dem hier gezeigten Beispiel nicht auf konkrete Instanzen, sondern auf den Namen der Eigenschaft und den des Property-Sets, zu dem sie gehört, verwiesen. (In der Form *NamePropertySet.NameProperty*.)

### Abhängigkeiten von Eigenschaften am Beispiel von Brückenlagern

Bei Brückenlagern hängen die mechanischen Eigenschaften teilweise von den semantischen Eigenschaften ab.

So grenzt der Lagertyp ein, welche Freiheitsgrade ein Lager haben kann. Ein Lager vom Typ *Horizontalkraftlager* beispielsweise ist per Definition in der Horizontalebene starr, während es entlang der Vertikalen frei verschieblich ist (vgl. Kapitel 5.6.7). Diese Abhängigkeit ist in Tabelle 39 schematisch beschrieben.

Tabelle 39: Abhängigkeit möglicher Verschiebungsfreiheitsgrade vom Lagertyp

IfcPropertyDependencyRelationship	
DependingProperty:	Punktlager_semantisch.Lagertyp
DependantProperty:	Punktlager_mechanisch.Lineare_Gelenkbedingungen_Translation
Description:	Wenn ein Punktlager vom Typ Horizontalkraftlager ist, dann muss das List-Value zur Beschreibung der linearen Verschiebungsfreiheitsgrade automatisch die Werte (\$,\$,0), also (starr, starr, frei beweglich) annehmen;
Expression:	Pseudocode: IF Lagertyp = „Horizontalkraftlager“ THEN Lineare_Gelenkbedingungen_Translation = (\$,\$,0);

## 7 Modelldaten in *IfcStructural*

Im vorangegangenen Kapitel wurde gezeigt, wie sich die statischen Systemeigenschaften von Brückenmodellen über Property-Sets darstellen lassen, ohne dafür das bestehende IFC-Schema erweitern zu müssen.

Eine alternative Möglichkeit besteht darin, hierfür die Domänenmodellerweiterung *IfcStructuralAnalysisModel* zu nutzen, welche speziell zur Beschreibung von statischen Systemen entwickelt wurde. Diese Modellerweiterung wurde jedoch für den Einsatz im klassischen Hochbau konzipiert und nicht für die Beschreibung von Ingenieurbauwerken.

In wieweit sich die zur Beschreibung von Brückenmodellen notwendigen Informationen unter Verwendung von *IfcStructural* (Version 2x3) darstellen lassen, wird in diesem Kapitel untersucht.

### 7.1 Einführung der Domänenmodellerweiterung *IfcStructural*

Die Domänenmodellerweiterung *IfcStructuralAnalysisModel* erlaubt die Beschreibung von Verbindungs- und Auflagereigenschaften, aufgebrachten Lasten sowie die von Ergebnissen statischer Berechnungen (siehe Abbildung 57).

Hierfür wird ein Element aus dem allgemeinen 3D-Modell mit einem sogenannten *Structural Member*, also einem Element aus dem Analysemodell, verknüpft. Jedes Element des physischen Modells kann mit beliebig vielen Elementen aus dem Structural-Analysis-Modell verknüpft werden und umgekehrt (buildingSMART). Die statischen Systemeigenschaften werden über die Relationen des jeweiligen *Structural Member* definiert. Das Analysemodell enthält nur die Verknüpfungen zum allgemeinen Geometriemodell, nicht das Geometriemodell selbst (Lehtinen & Hietanen, o.J.).

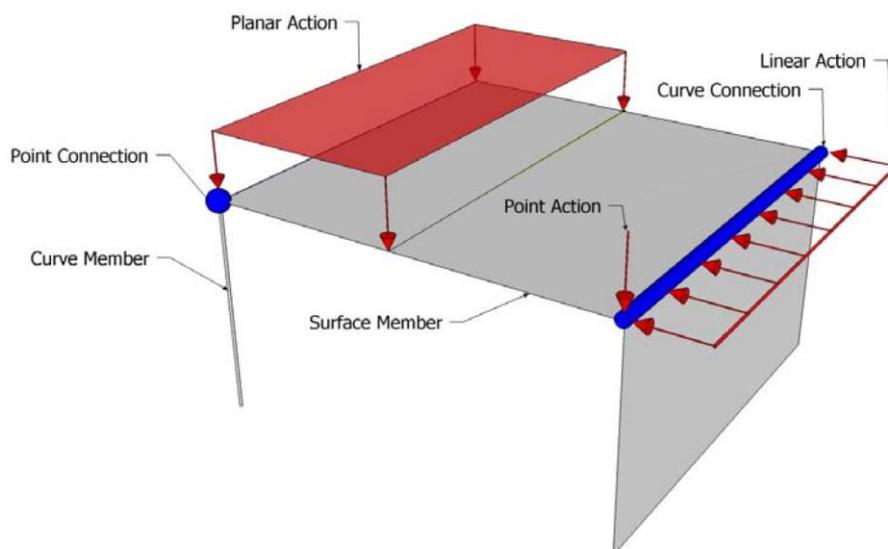


Abbildung 57: Idealisiertes Structural-Analysis-Modell inkl. Verbindungen und Lasten

Quelle: (Lehtinen & Hietanen, o.J.)

## 7.1.1 Verknüpfung von Geometriemodell und Tragwerksmodell

Über die relationsbildende Entität *IfcRelConnectsStructuralElement* wird ein *IfcElement* aus dem Geometriemodell mit seinem entsprechenden Vertreter im Tragwerksmodell, dem *IfcStructuralMember*, verknüpft.

Die abstrakte Klasse *IfcStructuralMember* gliedert sich in die beiden instanzierbaren Unterklassen *IfcCurveMember*, welche zur Beschreibung einer Linie bzw. Kurve im Raum dient, und *IfcSurfaceMember*, mittels derer Oberflächen beschrieben werden. Die Unterklasse *IfcPointMember*, durch welche sich ein Punkt im Raum beschreiben ließe, existiert in IFC2x3 noch nicht. Auf ihre geplante Einführung in zukünftigen Versionen wird in der Dokumentation zu IFC2x3 jedoch bereits hingewiesen (buildingSMART).

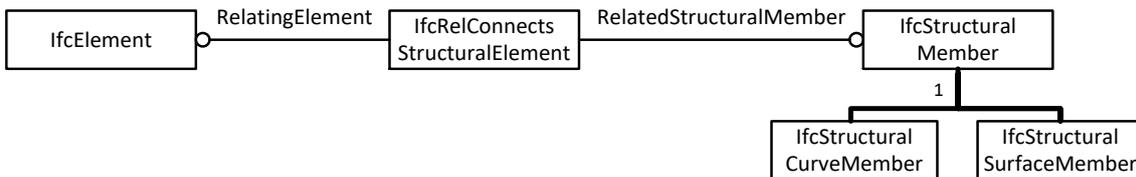


Abbildung 58: Verbindung von *IfcElement* und *IfcStructuralMember*

## 7.2 Beschreibung der mechanischen Eigenschaften in *IfcStructural*

### 7.2.1 Material

Allgemeine Materialparameter, wie sie zur Beschreibung eines isotrop-linear-elastischen Materialmodells benötigt werden, können innerhalb der Domänenmodellerweiterung *IfcStructural* über die Klasse *IfcMechanicalMaterialProperties* beschrieben werden. Diese kann über *IfcMaterial* und *IfcRelAssociatesMaterial* an alle Unterklassen von *IfcRoot*, also sowohl an ein *IfcBuildingElementProxy* als auch an einen *IfcStructuralMember*, angebunden werden (siehe Abbildung 59). Über die beiden Unterklassen *IfcMechanicalConcreteMaterialProperties* und *IfcMechanicalSteelMaterialProperties* können zusätzlich die Eigenschaften von Beton und Stahl detaillierter beschrieben werden.

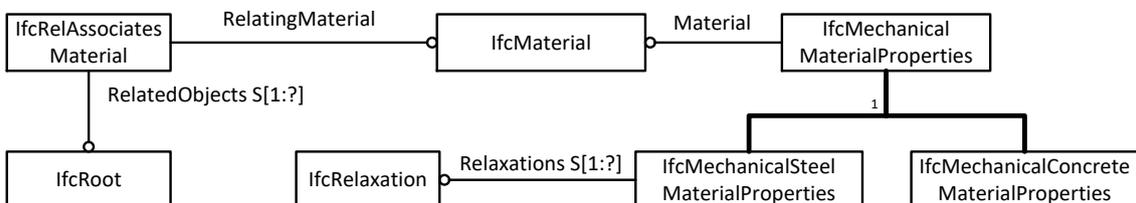


Abbildung 59: Anbindung von *IfcMechanicalMaterialProperties*

Die über *IfcMechanicalMaterialProperties* gegebenen allgemeinen Materialeigenschaften umfassen Viskosität, Elastizitätsmodul, Schubmodul, Querdehnzahl und thermi-

schen Ausdehnungskoeffizient (siehe Tabelle 40). Eine Darstellung der speziellen Eigenschaften von Beton und Stahl findet sich in Anlage 7.

Tabelle 40: Allgemeine Materialeigenschaften

IfcMechanicalMaterialProperties	
Attribut	Value
DynamicViscosity	Real
YoungModulus	Real
ShearModulus	Real
PoissonRatio	IfcRatioMeasure
ThermalExpansionCoefficient	Real

## Einschränkungen

Die Beschreibung plastischer oder orthotrop-elastischer Materialmodelle ist nicht möglich, da hierfür die entsprechenden Attribute wie z.B. zur richtungsabhängigen Beschreibung von E-Modul, Schubmodul etc. fehlen.

## 7.2.2 Lager

Die zur Modellierung von Lagern notwendige Beschreibung einer Verbindung mit sechs räumlichen Freiheitsgraden ist in *IfcStructural* mit Hilfe der Klasse *IfcStructuralConnection* möglich. Diese wird über *IfcRelConnectsStructuralMember* mit den die beiden Lagerhälften repräsentierenden Instanzen von *IfcStructuralMember* verknüpft.

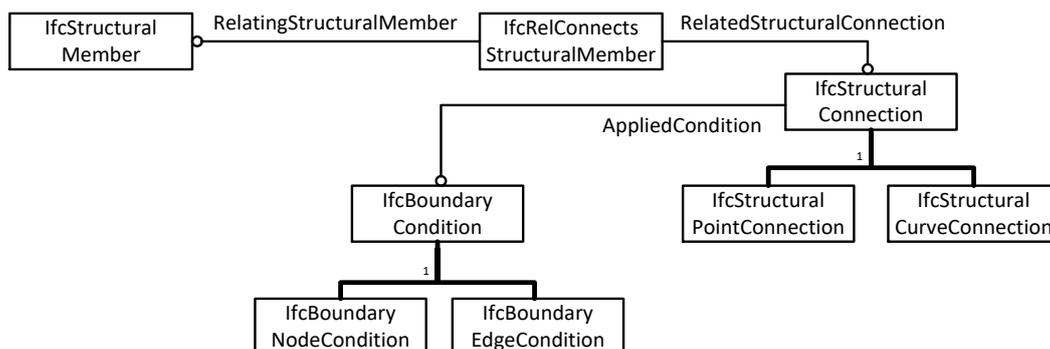


Abbildung 60: Verbindung von *IfcStructuralMember* über *IfcStructuralConnection*

Punktverbindungen werden über die Unterklasse *IfcStructuralPointConnection* beschrieben, Linienverbindungen über *IfcStructuralCurveConnection*. Den so definierten Verbindungen können nun über die Relation *AppliedCondition* Verbindungseigenschaften zugewiesen werden; *Boundary Conditions* – zu Deutsch „Randbedingungen“ – genannt.

Darin besteht ein Unterschied zur Beschreibung der Eigenschaften über Property-Sets: Das Property-Set Punktlager\_mechanisch wird über *RelDefinesByProperties* mit dem Proxy-Element verknüpft, in dem beide Lagerhälften zusammengefasst sind, da eine Anbindung von Property-Sets an Unterklassen von *IfcRelation* nicht möglich ist. Zur Beschreibung der mechanischen Lagereigenschaften in *IfcStructural* werden die Gelenkeigenschaften direkt der Verbindung selbst zugewiesen und nicht dem Lager als Gesamtkomponente.

## Lineare Steifigkeiten

*IfcBoundaryConditions* gliedert sich in die beiden Unterklassen *IfcBoundaryNodeCondition* und *IfcBoundaryEdgeCondition*, über die Eigenschaften von Punkt- bzw. Linien-gelenken definiert werden können. Beide besitzen sechs Attribute zur Beschreibung der linearen Verschiebungs- und Verdrehungseigenschaften bzw. deren zugehörigen Steifigkeiten (siehe Tabelle 41). Die Steifigkeiten können die Werte „Null“, „minus Eins“ einen beliebigen Wert größer Null annehmen. „Null“ steht hierbei für einen vollständigen Freiheitsgrad, also freie Beweglichkeit, „minus Eins“ steht für eine unendlich hohe Steifigkeit bzw. eine, bezogen auf einen bestimmten Freiheitsgrad, starre Verbindung. Alle Werte größer Null beschreiben eine lineare Federsteifigkeit.

Tabelle 41: *IfcBoundaryNodeCondition*

<b>IfcBoundaryNodeCondition</b>	
<b>Attribut</b>	<b>Value</b>
LinearStiffnessX	Real
LinearStiffnessY	
LinearStiffnessZ	
RotationalStiffnessX	Real
RotationalStiffnessY	
RotationalStiffnessZ	

## Nichtlineare Gelenkeigenschaften

Weitere nichtlineare Eigenschaften können über die Klasse *IfcStructuralConnectionCondition* beschrieben werden, bzw. über deren Unterklassen *IfcSlippageConnectionCondition* und *IfcFailureConnectionCondition*. Diese werden nicht über die Verbindungsklasse *IfcStructuralConnection* selbst, sondern über ein Attribut der relationsbildenden Entität *IfcRelConnectsStructuralMember* referenziert (siehe folgende Abbildung).

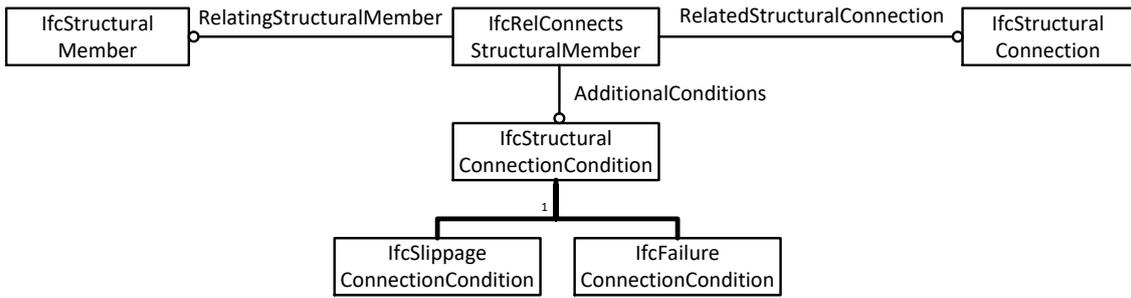


Abbildung 61: Anbindung zusätzlicher Gelenkeigenschaften

## Grenzwerte für Druck- und Zugbelastung

Über *IfcFailureConnectionCondition* können die maximal zulässigen Druck- und Zugkräfte definiert werden. Hierfür stehen je drei Attribute zur Beschreibung der Druck- bzw. Zugspannungen in x-, y- und z-Richtung zur Verfügung.

Tabelle 42: *IfcFailureConnectionCondition*

IfcFailureConnectionCondition	
Attribut	Value
TensionFailureX	Real
TensionFailureY	Real
TensionFailureZ	Real
CompressionFailureX	Real
CompressionFailureY	Real
CompressionFailureZ	Real

## Schlupf

Über *IfcSlippageConnectionCondition* kann eine freie Verschieblichkeit innerhalb einer begrenzten Strecke festgelegt werden, jeweils getrennt für die x-, y- und z-Richtung (siehe Tabelle 43).

Tabelle 43: *IfcSlippageConnectionCondition*

IfcSlippageConnectionCondition	
Attribut	Value
SlippageX	IfcLenthMeasure
SlippageY	IfcLenthMeasure
SlippageZ	IfcLenthMeasure

Auf diese Weise kann beispielsweise modelliert werden, dass manche Lager keine vertikalen Zugkräfte übertragen können. Der Schlupf muss dabei auf eine „nach oben“ gerichtete lokale Koordinatenachse bezogen werden.

## Einschränkungen

Nichtlineare Beziehungen zwischen Auslenkung und resultierender Rückstellkraft sowie maximal mögliche Relativverschiebungen und -verdrehungen können in *IfcStructural* nicht dargestellt werden. Auch eine differenzierte Beschreibung dessen, was beim Überschreiten der maximal zulässigen Gelenkkräfte passiert, ist nicht möglich (Fließen, Reißen, etc.).

### 7.2.3 Seile

Die in Statikprogrammen gebräuchliche Beschreibung von Seilen als Stabelement vom Typ „Seilstab“ kann in *IfcStructural* nach dem gleichen Prinzip erfolgen, wie bei der in Kapitel 6.7.1 gezeigten Darstellung über *IfcPropertySet*.

Hierfür wird das Geometrieobjekt „Seil“ mit einem *IfcStructuralCurveMember* verknüpft, dem über die Relation *PredefinedType* der Typ „CABLE“ zugewiesen wird. Dieser Typ ist innerhalb der IFC-Dokumentation festgelegt als „Zugglied, das nur unter großer Auslenkung transversale Lasten übertragen kann“ (BuildingSmart, o.J.)

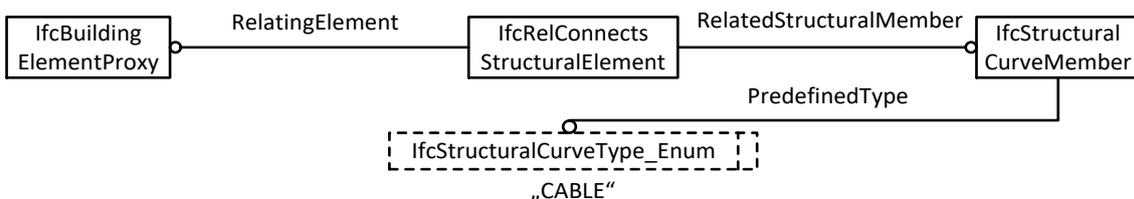


Abbildung 62: Beschreibung von Seilen in *IfcStructural*

### Vorspannung von Seilen

Vorspannung von Seilen kann in *IfcStructural* beschrieben werden, indem dem das Seil repräsentierende *IfcStructuralCurveMember* eine konstante positive Normalkraft zugeordnet wird. Dies geschieht über die Entität *IfcStructuralLoadLinearForce*. Sie besitzt sechs optionale Attribute, über die konstante Lasten bzw. Momente jeweils auf die drei Raumachsen bezogen definiert werden können.

Das Schema zur Anbindung von *IfcStructuralLoadLinearForce* über *IfcRelConnectsStructuralActivity* an das *BuildingElement-Proxy* ist in Abbildung 63 dargestellt.

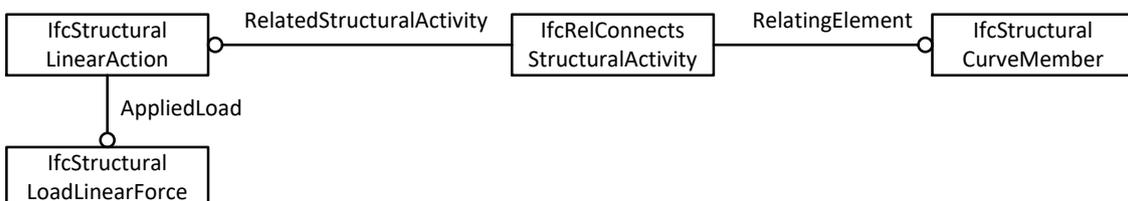


Abbildung 63: Anbindung von Vorspannkraft an Seile

## 7.2.4 Stabartige Bauteile

Hänger, Stützen und Stabbogensegmente sowie stabartige Pfeiler können nach dem gleichen Prinzip wie Seile modelliert werden, sofern sich ihre Eigenschaften über die in *IfcStructural* vordefinierte Stabtypen „Balkenstab“, „Pendelstab“, „Zugstab“ und „Druckstab“ beschreiben lassen.

Durch diese Stabtypen nicht abgedeckte spezielle Gelenkeigenschaften der Stabenden können – analog zur Beschreibung von Lagern – durch Zuweisung von Randbedingungen zu den Verbindungen der Stabenden zu angrenzenden Bauteilen modelliert werden.

## 7.2.5 Vorspannung

Zur Darstellung von Vorspannung in *IfcStructural* bieten sich grundsätzlich die gleichen zwei Varianten wie schon zuvor bei der Darstellung über Property-Sets. Zum einen können die Spannglieder als Geometrieobjekte modelliert werden, denen eine tangentiale Zugkraft zugeordnet wird, zum anderen können die aus der Vorspannung resultierenden Schnittkräfte als Ersatzlasten auf das vorgespannte Bauteil aufgebracht werden. In beiden Fällen ergeben sich bei der Verwendung bestehender *IfcStructural*-Klassen gewisse Probleme.

### Linienlast tangential zur Spanngliedachse

Variante eins, also die Zuordnung einer tangentialen Zugkraft zu einem geometrisch modellierten Spannglied, bringt das Problem mit sich, dass IFC keine Möglichkeit vorsieht, Kräfte einer gekrümmt verlaufenden Bezugsachse zuzuweisen.

Ein gekrümmt verlaufendes Spannglied könnte über *IfcSurfaceCurveSweptAreaSolid* beschrieben werden. Dazu wird eine Fläche entlang einer Raumkurve extrudiert; der Verlauf der Kurve ist dabei über eine Unterklasse von *IfcCurve* definiert, die sogenannte *Directrix*.

Die Zugkraft kann über *IfcStructuralLoadLinearForce* angegeben und über *IfcStructuralAction* und *IfcRelConnectsStructuralActivity* mit dem jeweiligen Spannglied verbunden werden. *IfcStructuralLoadLinearForce* besitzt sechs optionale Attribute, über die konstante Lasten und Momente jeweils bezogen auf die x-, y- und z-Achse definiert werden können.

Über das Attribut *IfcGlobalOrLocalEnum* wird angegeben, ob sich diese Lasten auf die globalen Koordinaten oder auf die lokalen Koordinaten des Structural-Member – also des Spanngliedes – beziehen.

Das Problem dabei ist nun, dass das lokale Koordinatensystem eines Swept-Area-Solids nur die Ausgangsrichtung der Kurve angibt.

Eine Beschreibung der Krafrichtung im Sinne von „verläuft an jedem Punkt tangential zur *Directrix* des Swept-Area-Solids“ steht nicht zur Verfügung.

Inwiefern diese Information über andere Relationen von *IfcStructuralAction* angegeben werden kann, z.B. über *Description* oder *Repräsentation* müsste in der Zukunft noch genauer untersucht werden.

## Veränderliche Größe der Ersatzlasten

Bei der zweiten Variante – der Beschreibung der auf das vorgespannte Bauteil wirkenden Ersatzlasten – ergibt sich das Problem, dass die aus Vorspannung resultierenden Schnittkräfte über die Bauteillänge nicht konstant verlaufen.

Die Attribute der Klassen von *IfcStructuralLoadLinearForce* ermöglichen jedoch nur die Beschreibung konstanter Lastgrößen, z. B. 5 kN/m.

Bei einer detaillierten Modellierung von Vorspannung würde dies zudem bei der oben beschriebenen Variante (Modellierung geometrischer Spannglieder mit zugewiesener Vorspannkraft) zu Problemen führen, da die Zugkräfte entlang eines Spannglieds in der Realität auch nicht konstant verlaufen. Stattdessen nehmen sie auf Grund von Reibungsverlusten immer mehr ab, je weiter man sich von der Verankerungsstelle, an der die Zugkraft aufgebracht wurde, entfernt.

## Zusammenfassung mehrerer Lasten zur Lastgruppe „Vorspannung“

Unabhängig von den oben beschriebenen Problemen ist es möglich, mehrere Lasten in einer Lastgruppe zusammenzufassen und diese Gruppe semantisch als „Vorspannlasten“ zu definieren. So könnten aus der Vorspannung resultierende Lasten über *IfcRelAssignsToGroup* in einer Instanz von *IfcStructuralLoadGroup* zusammengefasst werden (siehe Abbildung 64). Über deren Attribute *IfcActionTypeEnum* und *IfcActionSourceTypeEnum* kann die Art der Belastung sowie deren Ursache definiert werden (siehe Tabelle 44). So kann beispielsweise dargestellt werden, dass es sich bei allen Lasten einer Gruppe um permanente, aus Vorspannung resultierende Lasten handelt.

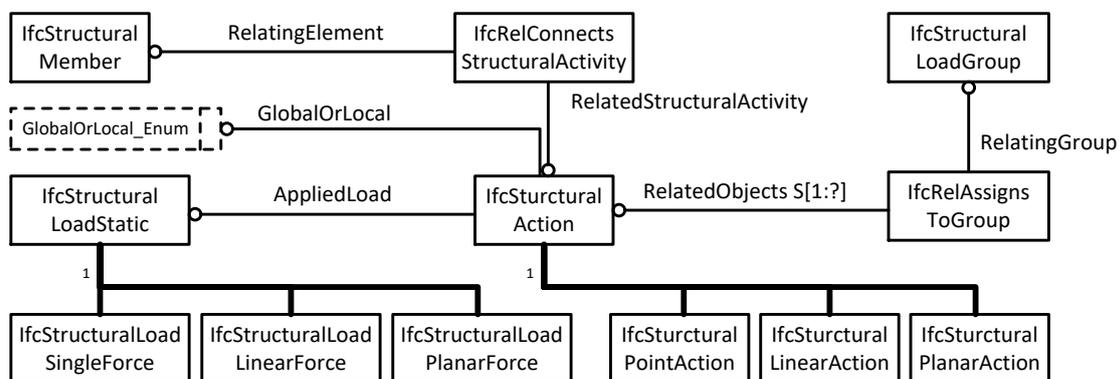


Abbildung 64: Darstellung von Vorspannung über *IfcStructuralAction*

Tabelle 44: Auszug der Möglichen Werte von *ActionType* und *ActionSource*

IfcStructuralLoadGroup	
Attribut	Value (enumerated)
ActionType	PERMANENT_G, VARIABLE_Q, EXTRAORDINARY_A, USERDEFINED, NOTDEFINED;
ActionSource	SNOW_S, WIND_W, PRESTRESSING_P, SETTLEMENT_U, TEMPERATURE_T, EARTHQUAKE_E,

## 8 Verifikation des Konzepts anhand einer Beispielinstantz

Um die Möglichkeit der Instanziierung des Hilfsschemas unter Verwendung bestehender IFC2x3-Klassen zu verifizieren, wird im Folgenden eine Beispielinstantz der vorge-spannten Balkenbrücke auf Validität geprüft.

Hierfür wird zunächst ein CAD-Modell der Brücke (vgl. 5.3.1) aus *Autodesk Revit* als IFC2x3-Datei exportiert.

Die hierarchische Struktur der einzelnen Bauwerkskomponenten, die explizite Verbindung der Lagerelemente sowie die über Property-Sets dargestellten Eigenschaften der Komponenten werden dem exportierten Dateicode händisch annotiert. Von *Revit* automatisch erstellte, nicht zum Modell passende Eigenschaften und Aggregationen werden manuell aus dem Dateicode entfernt. Der vollständige Code der Beispielinstantz findet sich in Anhang 1.

Die Validitätsprüfung des manuell veränderten Codes erfolgt über das Programm *KFZ-Viewer* des *KIT*. Das Programm ermöglicht sowohl die grafische Darstellung der Modellgeometrie als auch die Darstellung der auf der semantischen Ebene festgelegten Hierarchie der Bauwerkskomponenten und ihrer Eigenschaften. Dabei wurden alle Annotationen erkannt und korrekt dargestellt.

Die separate Validierungsfunktion des *FZK-Viewers* gibt dennoch einige Fehlermeldungen vom Typ *integrity constraint violations* zurück. Diese werden jedoch größten Teils auf die manuelle Entfernung der von *Revit* automatisch erstellten Instanzen von *IfcBuilding* und *IfcSite* sowie weiterer automatisch erstellter Annotationen und Aggregationen zurückgeführt. Da diese Fehler keinen erkennbaren Einfluss auf die korrekte Darstellung des Modells haben, wurden sie nicht weiter behandelt.

Eine ausführliche Beschreibung der Beispielinstantz sowie der mittels *FZK-Viewer* durchgeführten Verifikation inklusive Screenshots findet sich in Anlage 2.

Die Diagramme bieten zum einen eine Übersicht über die Struktur der Beispielinstantzen, zum anderen dienen sie als Verifikation, dass die händisch veränderte IFC2x3-Instanz mit Hilfe gängiger Programme gelesen werden kann und dass ihre Struktur mit der des Hilfsschemas übereinstimmt.

## 9 Schlussbetrachtung

### 9.1 Zusammenfassung

Abschließend lässt sich sagen, dass es möglich ist, Brückenbauwerke in generalisierter Form in einem *EXPRESS*-Hilfsschema abzubilden und dieses Schema unter Verwendung des bestehenden *IFC2x3*-Schemas zu instanziiieren.

Die zur statischen Berechnung von Brückenbauwerken notwendigen Modelldaten, die hierarchischen Struktur des Gesamtbauwerks sowie weitere, nicht-mechanische Eigenschaften können mit Hilfe der *IfcProperty*- und *IfcProxy*-Klassen dargestellt und den geometrischen Modelldaten händisch hinzugefügt werden.

Auf diese Weise erweiterten Modellinstanzen lassen sich von einem gängigen *IFC-Viewer*-Programm einlesen und korrekt darstellen.

Auf Grund fehlender Schnittstellen zu konkreten Statikprogrammen kann jedoch nur die syntaktische Validität der im *IFC2x3*-Format instanziierten Modelldatei gezeigt werden. Inwieweit das Konzept zur automatischen Generierung von Analysemodellen geeignet ist, müsste in der Zukunft gesondert untersucht werden.

Ein Vergleich der benötigten statischen Modelldaten mit den Möglichkeiten der Domainmodellerweiterung *IfcStructural* zeigt, dass *IfcStructural* zwar die Beschreibung eines Großteils der statischen Systemeigenschaften von Brückenbauwerken erlaubt, bei detaillierteren Beschreibung einzelner Teilbereiche jedoch an ihre Grenzen gelangt.

### 9.2 Ergebnisse und Ausblick

Die in *IFC2x3 (Koordination View)* zur Verfügung stehenden *PropertySets* bzw. die in ihnen zusammengefassten Unterklassen von *IfcProperty* sind variabel genug einsetzbar, um alle nötigen Parameter zur Beschreibung statischer Systemeigenschaften sowie weitere nicht-geometrische Daten bereitzustellen.

Mit Hilfe der *IfcProxy*-Klassen in Kombination mit den Unterklassen von *IfcRelation*, insbesondere durch die Klasse *IfcRelAggregates*, lässt sich auch die hierarchische Struktur der Komponenten eines Brückenbauwerks ausführlich beschreiben.

Die im Hilfsschema formulierten, verallgemeinerten Beziehungen zur Beschreibung der geometrischen Verbindungen sowie die der Rolle, die die verbundenen Komponenten beim Lastabtrag spielen, ließe sich mit Hilfe der IFC-Klasse *IfcRelConnectsElements* prinzipiell auch abbilden.

Der Zweck dieser Relationen liegt jedoch in erster Linie darin, eine generalisierte Beschreibung der Brückenkonstruktionen innerhalb des Hilfsschemas zu ermöglichen.

Da sich diese Informationen aus der Geometrie einer Modellinstanz bereits weitestgehend ergeben, erscheint eine zusätzliche Darstellung dieser Information auf der semantischen Ebene innerhalb einer Instanz nicht sinnvoll.

In wieweit Informationen wie z. B. die explizite Beschreibung von Verbindungen und Verbindungsgeometrie für die korrekte Interpretation der Modelle durch Analyseprogramme dennoch notwendig ist, müsste in der Zukunft gesondert untersucht werden.

Ein Vergleich der von *RFEM* benötigten Modelldaten mit den in der Domainmodellerweiterung *IfcStructural* vorhandenen Klassen zeigt, dass mit Hilfe von *IfcStructural* zwar eine Vielzahl der Eigenschaften von Brückenkonstruktionen dargestellt werden können; eine vollständige Beschreibung möglicher nichtlinearer Gelenkeigenschaften von Brückenlagern, die Darstellung von Vorspannung oder die Beschreibung anisotropen Materialeigenschaften ist jedoch nicht möglich.

### **Einschränkungen bei der Modellierung mit *IfcProperty*-Klassen**

Property-Sets können in *IFC2x3* nicht an Unterklassen von *IfcRelation* angehängt werden, da sie nur eine Relation zur Klasse *IfcObjekt* besitzen. Daher ist es nicht möglich, Verbindungseigenschaften in *Ifc Coordination View* direkt einer auf der semantischen Ebene definierten Verbindung zuzuweisen, sondern nur den verbundenen Objekten selbst.

Die direkte Zuweisung von Eigenschaften zu einer Verbindung ist bisher nur in *IfcStructural* möglich.

### **Geometrische Beschreibung idealisierter Tragwerke**

Da in *Autodesk Revit* nur dreidimensionale Volumenkörper als IFC-Objekte exportiert werden können, wurde eine mögliche Idealisierung der Tragwerksgeometrie über Linien- und Flächenelemente in dieser Arbeit nicht weiter behandelt. Prinzipiell ist es jedoch möglich, in *IfcStructural* jedem Struktural-Member eine eigene geometrische Repräsentation zuzuweisen, wodurch sich die Komponenten eines geometrisch idealisierten Tragwerksmodells mit denen des ursprünglichen Geometriemodells verknüpfen lassen.

Inwiefern eine separate Geometrierepräsentation des Tragwerkmodells im Brückenbau sinnvoll ist, welche Probleme sie mit sich bringt und ob sie sich ggf. sogar automatisieren lässt, könnte ebenfalls in der Zukunft weiter untersucht werden.

## Literaturverzeichnis

Anderl, R., & Trippner, D. (2000). *STEP - Eine Einführung in die Entwicklung, Implementierung und industrielle Nutzung der Normenserie ISO 10303 (STEP)*. Leipzig : B. G. Teubner.

archiexpo.de. (o.J.). *archiexpo.de*. Abgerufen am 01. Februar 2017 von <http://www.archiexpo.de/prod/us-bridge/product-133011-1550483.html>

baulinks.de. (10. Juni 2007). *baulinks.de*. Abgerufen am 5. Januar 2017 von <http://www.baulinks.de/webplugin/2007/0905.php4>

Bletz, O., & Bathon, L. (2008). Holz-Beton-Verbund-Verkehrsbrücken. *HOLZBAU*.

BT Bautechnik GmbH. (o.J.). *bt-bautechnik-gmbh.de*. Abgerufen am 25. November 2016 von [http://www.bt-bautechnik-gmbh.de/Lager\\_Tf.html](http://www.bt-bautechnik-gmbh.de/Lager_Tf.html)

*buildingSMART*. (kein Datum). Abgerufen am 4. September 2016 von <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcstructuralanalysisdomain/lexical/ifcstructuralanalysisismodel.htm>

*buildingSMART*. (kein Datum). Abgerufen am 4. September 2016 von <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcstructuralanalysisdomain/lexical/ifcstructuralmember.htm>

*buildingSMART*. (o.J.). *buildingSMART*. Von <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/> abgerufen

*BuildingSmart*. (o.J.). *BuildingSmart*. Von <http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcstructuralanalysisdomain/lexical/ifcstructuralcurvetypeenum.htm> abgerufen

Corchado, E., Hujun Yin, V. B., & Fyfe, C. (2006). *Intelligent Data Engineering and Automated Learning*. (S. 1285 ff). Burgos, Spain: Springer.

Dacheng Rubber Co., Ltd. (2014). *bridgebearings.org*. Abgerufen am 28. November 2016 von <http://www.bridgebearings.org/product/spherical-bridge-bearing.html>

Dehn, F., Holschemacher, K., & Tue, N. V. (2004). *Neue Entwicklungen im Brückenbau*. Berlin: Bauwerk Verlag GmbH.

Dlubal Software GmbH. (2016). *Programm Beschreibung RF-TENDON*. Tiefenbach: Dlubal Software GmbH.

Dywidag-Systems International. (2016). *dywidag-systems.de*. Von [https://www.dywidag-systems.de/uploads/pics/DSI\\_Japan\\_Neak-Loeung-Cambodia\\_2.jpg](https://www.dywidag-systems.de/uploads/pics/DSI_Japan_Neak-Loeung-Cambodia_2.jpg) abgerufen

- Eggert, H., & Kauschke, W. (2002). *Structural Bearings*. Berlin: Ernst & Sohn Verlag für Architektur und technische Wissenschaften GmbH.
- Feller, P. (30. October 2013). *commons.wikimedia.org*. Abgerufen am 27. November 2016 von [https://commons.wikimedia.org/wiki/File:Roller\\_Bearing,\\_Jensen\\_Drive\\_\(Hill\\_Street\)\\_Bridge\\_over\\_Buffalo\\_Bayou,\\_Houston,\\_Texas\\_1310261117\\_\(10577360824\).jpg](https://commons.wikimedia.org/wiki/File:Roller_Bearing,_Jensen_Drive_(Hill_Street)_Bridge_over_Buffalo_Bayou,_Houston,_Texas_1310261117_(10577360824).jpg)
- Fu, G. (o.J.). *king-of-offshore.blogspot.com*. Abgerufen am 24. November 2016 von <http://king-of-offshore.blogspot.com/tag/Bridge-Bearing>
- GmbH, Dlubal Software. (2016). *RFEM 5 - Räumliche Tragwerke nach der Finite Elemente Methode - Programm-Beschreibung*. Tiefenbach: Dlubal Software GmbH.
- Hamburger Tauwerk Fabrik GmbH & Co.KG. (o.J.). *Hamburger Tauwerk Fabrik*. Abgerufen am 7. November 2016 von <https://www.htf-hh.com/media/images/org/Drahtseil7x19.jpg>
- Ji, D.-I. Y. (17. 07 2014). Durchgängige Trassen- und Brückenplanung auf Basis eines. München: Technische Universität München; Ingenieur fakultät Bau Geo Umwelt.
- Kempkens, W. (01. November 2016). *ingenieur.de*. Abgerufen am 03. Februar 2017 von <http://www.ingenieur.de/Themen/Werkstoffe/Schluss-rostenden-Autobahnbruecken>
- kid-lit-reviews.com. (o.J.). *kid-lit-reviews.com*. Von <https://kid-lit-reviews.com/2015/05/10/692-building-the-golden-gate-bridge-an-interactive-engineering-adventure-by-blake-hoena/> abgerufen
- Kracke, E.-A., & Lodde, K. (2011). *Leitfaden Straßenbrücken - Entwurf, Baudurchführung, Erhaltung*. Ernst & Sohn.
- Kreuser, M., & Purainer, R. (2003). *Zur wirklichkeitsnahen Berechnung von Tragwerken aus Stahlbeton*. Springer Verlag.
- Lehtinen, S., & Hietanen, J. (o.J.). *Structural Design to Structural Analysis*. Tampere University of Technology.
- Lukas, K., Ji, Y., Obergießer, M., & Borrmann, A. (o.J. ). *Parametrische Modellierung von Brückenbauwerken*. München: Lehrstuhl für Computational Engineering, Technische Universität München.
- mageba-germany.de. (o.J.). *mageba-germany.de*. Abgerufen am 28. November 2016 von <http://www.mageba-germany.de/de/794/Infrastruktur/Bauwerkslager/Horizontal-Querkraftlager/RESTON-FORCE/Detail.htm>
- Malcorp. (o.J.). *malcorp.net*. Abgerufen am 25. November 2016 von <http://www.malcorp.net/our-products/bridge-bearings-expansion-joint/>

- mangfall24.de. (o.J.). *mangfall24.de*. Abgerufen am 01. Februar 2017 von <https://www.mangfall24.de/region/mangfalltal/kolbermoor-ort45141/kolbermoor-stadtrat-moniert-aicherpark-bruecke-5559236.html>
- Mehlhorn, G., & Curbach, M. (2014). *Handbuch Brücken*. Wiesbaden: Springer Fachmedien.
- Mehmel, A. (1973). *Vorgespannter Beton: Grundlagen der Theorie, Berechnung und Konstruktion*. Berlin/Heidelberg: Springer.
- Nemetscheck SCIA. (kein Datum). *SCIA Resource Centre*. Abgerufen am 21. Oktober 2016 von [http://resources.scia.net/de/factsheets/analysis/analyzer\\_plasticanalysisofsurfacemembers.htm](http://resources.scia.net/de/factsheets/analysis/analyzer_plasticanalysisofsurfacemembers.htm)
- oiles corporation. (o.J.). *oiles.co*. Abgerufen am 27. November 2016 von <http://www.oiles.co.jp/en/menshin/bridge/product/rb/>
- Schuler, R. W. (2001). *The Application of ISO 10303-11 (the EXPRESS Language) in Defining Data Models for Software Design and Implementation*. Maryland: University of Maryland University College.
- Stetter, O. (Februar 2010). *Leonhardt2009.de*. Abgerufen am 20. September 2016 von <http://www.leonhardt2009.de/allgemeine-informationen-zum-brueckenbau/bestandteile-und-elemente-einer-bruecke.html>
- structurae.de. (o.J.). *structurae.de*. Abgerufen am 01. Februar 2017 von <https://structurae.de/bauwerke/rhein-herne-kanal-bruecke-a3>
- Weidemann, H. (1982). *Brückenbau: Stahlbeton- u. Spannbetonbrücken*. Düsseldorf: Werner-Verlag GmbH.
- Wikipedia.de*. (kein Datum). Abgerufen am 22. September 2016 von [https://de.wikipedia.org/wiki/Orthotrope\\_Platte](https://de.wikipedia.org/wiki/Orthotrope_Platte)
- Wikipedia.de*. (kein Datum). Abgerufen am 20. September 2016 von <https://de.wikipedia.org/wiki/Br%C3%BCcke#.C3.9Cberbau>
- Yabuki, N., Lebegue, E., Gual, J., Shitani, T., & Zhanatao, Z. (2006). *International collaboration for developing the bridge product model "IFC-BRIDGE"*. Montréal, Canada: Joint International Conference on Computing and Decision Making in Civil and Building Engineering.

## Anlagen

Anlage 1: Quellcode der Balkenbrücke inkl. Annotationen .....	99
Anlage 2: Verifikation der Beispielinstantz (Ergänzung zu Kapitel 8).....	109
Anlage 3: EXPRESS-Spezifikationen der Hilfsschema-Klassen.....	115
Anlage 4: Punkttypen S-P und P-P (Ergänzung zu Kapitel 6.5.1) .....	120
Anlage 5: Segment-Typen in <i>RFEM</i> (Ergänzung zu Kapitel 6.5.1).....	121
Anlage 6: Eigenschaften von <i>IfcTendon</i> .....	122
Anlage 7: Spezielle Eigenschaften von Beton und Stahl in <i>IfcStructural</i> .....	122

# Anlage 1: Quellcode der Balkenbrücke inkl. Annotationen

ISO-10303-21;  
HEADER;

```
/*
*****
* STEP Physical File produced by: The EXPRESS Data Manager Version 5.02.0100.07 : 28 Aug 2013
* Module: EDMstepFileFactory/EDMstandAlone
* Creation date: Mon Feb 27 17:24:27 2017
* Host: DESKTOP-5QEEMIM
* Database: C:\Users\Andreas\AppData\Local\Temp\{CD88C0F7-493C-40A3-9D2E-DB4A0F7FA92B}\ifc
* Database version: 5507
* Database creation date: Mon Feb 27 17:24:27 2017
* Schema: IFC2X3
* Model: DataRepository.ifc
* Model creation date: Mon Feb 27 17:24:27 2017
* Header model: DataRepository.ifc_HeaderModel
* Header model creation date: Mon Feb 27 17:24:27 2017
* EDMuser: sdai-user
* EDMgroup: sdai-group
* License ID and type: 5605 : Permanent license. Expiry date:
* EDMstepFileFactory options: 020000
*****
FILE_DESCRIPTION(('ViewDefinition [CoordinationView]'),'2;1');
FILE_NAME('0001','2017-02-27T17:24:27','(',')','The EXPRESS Data Manager Version 5.02.0100.07 : 28 Aug
2013','20140606_1530(x64) - Exporter 2015.0.2014.0606 - Default UI','');
FILE_SCHEMA(('IFC2X3'));
ENDSEC;
```

DATA;

**/\* \*\*\*\*\* ANNOTATIONEN \*\*\*\*\* \*/**

**/\*Hierarchische Struktur der Bauwerkskomponenten\*/**

```
/*Fahrbahntragwerk-Proxy zusammengefasst aus (Segment aus (Längsträger 1 und 2 + Fahrbahnplatte) */
#4999= IFCPROXY ('19s4b2W4b8XxQ4ZKiC4999', #41, 'Segment', $, 'ZusammengesetzterBauwerksteil', $, $, .PRODUCT., $);
#4998= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH4998', #41, 'AggregatesSegment', $, #4999, (#933, #814, #870));
#5000= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5000', #41, 'Fahrbahntragwerk', $, 'ZusammengesetzterBauwerksteil', $, $, .PRODUCT., $);
#5001= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEHAa1', #41, 'AggregatesFahrbahntragwerk', $, #5000, (#4999));
/*Lager-Proxy aus den beiden Lagerhälften*/
/*Lager 1*/
#5013= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5013', #41, 'Lager1', $, 'Komponente', $, $, .PRODUCT., $);
#5014= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH5014', #41, 'AggregatesLager1', $, #5013, (#1316, #1082));
/*Lager 2*/
#5015= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5015', #41, 'Lager2', $, 'Komponente', $, $, .PRODUCT., $);
#5016= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH5016', #41, 'AggregatesLager2', $, #5015, (#1379, #1432));
/*Lager 3*/
#5017= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5017', #41, 'Lager3', $, 'Komponente', $, $, .PRODUCT., $);
#5018= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH5018', #41, 'AggregatesLager3', $, #5017, (#1482, #1535));
/*Lager 4*/
#5019= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5019', #41, 'Lager4', $, 'Komponente', $, $, .PRODUCT., $);
#5020= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH5020', #41, 'AggregatesLager4', $, #5019, (#1585, #1638));
/*Unterbau-Proxy aus Fundamenten und Widerlagern*/
#5002= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5002', #41, 'Unterbau', $, 'ZusammengesetzterBauwerksteil', $, $, .PRODUCT., $);
#5003= IFCRELAGGREGA-
TES ('2xmOiD17X4RAYWPjEH5003', #41, 'AggregatesUnterbau', $, #5002, (#399, #726, #518, #152)); /*(Fundamente, Widerlager)*/
/*Brücke-Proxy aus zusammengefasst aus (Fahrbahntragwerk, Unterbau und Lagern)*/
#5004= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5004', #41, 'Balkenbruecke', $, 'ZusammengesetzterBauwerksteil', $, $, .PRODUCT., $);
#5005= IFCRELAGGREGA-
TES ('2xmOiD17X4RAYWPjEH5005', #41, 'AggregatesBruecke', $, #5004, (#5011, #5002, #5016, #5013, #5015, #5017, #5019));
/*Brückenausrüstung-Proxy aus (Fahrbahnausbildung, Entwässerung, Leiteinrichtungen, Fahrbahnübergang, Kappen)*/
#5006= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5006', #41, 'BrueckenAusruistung', $, 'Komponente', $, $, .PRODUCT., $);
#5007= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5007', #41, 'Fahrbahnausbildung', $, 'Ausbaukomponente', $, $, .PRODUCT., $);
#5008= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5008', #41, 'Etnwaesserung', $, 'Ausbaukomponente', $, $, .PRODUCT., $);
#5009= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5009', #41, 'Leiteinrichtungen', $, 'Ausbaukomponente', $, $, .PRODUCT., $);
#90000= IFCPROXY ('19s4b2W4b8XxQ4ZKi90000', #41, 'Fahrbahneuebergang', $, 'Ausbaukomponente', $, $, .PRODUCT., $);
#90001= IFCPROXY ('19s4b2W4b8XxQ4ZKi90001', #41, 'Kappen', $, 'Ausbaukomponente', $, $, .PRODUCT., $);
#5010= IFCRELAGGREGA-
TES ('2xmOiD17X4RAYWPjEH5010', #41, 'AggregatesBrueckenAusruistung', $, #5006, (#5007, #5008, #5009, #90000, #90001));
/*Überbau-Proxy aus (Brückenausrüstung und Fahrbahntragwerk)*/
#5011= IFCPROXY ('19s4b2W4b8XxQ4ZKiC5011', #41, 'Ueberbau', $, 'Komponente', $, $, .PRODUCT., $);
#5012= IFCRELAGGREGATES ('2xmOiD17X4RAYWPjEH5012', #41, 'AggregatesUeberbau', $, #5011, (#5006, #5000));
/*Verbindung der Beiden Lagerhälften*/
#5021= IFCRELCONNECTSELEMENTS ('19s4b2W4b8XxQ4ZKiC5021', #41, 'BeweglVerbLager1', $, $, #1316, #1082);
#5022= IFCRELCONNECTSELEMENTS ('19s4b2W4b8XxQ4ZKiC5022', #41, 'BeweglVerbLager2', $, $, #1379, #1432);
#5023= IFCRELCONNECTSELEMENTS ('19s4b2W4b8XxQ4ZKiC5023', #41, 'BeweglVerbLager3', $, $, #1482, #1535);
#5024= IFCRELCONNECTSELEMENTS ('19s4b2W4b8XxQ4ZKiC5024', #41, 'BeweglVerbLager4', $, $, #1585, #1638);
```

**/\*Annotation der PropertySets\*/**

```
/*BrückenTyp_semantisch*/
#6000= IFCREDEFINESBYPROPERTIES ('19s4b2W4b8XxQ4ZKiC6000', #41, $, $, (#5004, #6001);
#6001= IFCPROPERTYSET ('19s4b2W4b8XxQ4ZKiC6001', #41, 'BrueckenTyp_semantisch', $, (#6003, #6004, #6005, #6006, #6007));
#6003= IFCPROPERTYENUMERATEDVALUE ('KonstruktionsformHaupttragwerk', $, ('Balkenbruecke'), $);
#6004= IFCPROPERTYENUMERATEDVALUE ('Material', $, ('Stahlbeton'), $);
#6005= IFCPROPERTYENUMERATEDVALUE ('Funktion', $, ('StraßenB'), $);
```

```

#6006= IFCPROPERTYENUMERATEDVALUE ('Lage', $, ('Festland'), $);
#6007= IFCPROPERTYENUMERATEDVALUE ('Verbundart', $, ('Aufgelagert'), $);
/*Material_mechanisch "Beton", zugewiesen zu allen Bauteilen außer Lagern*/
#6008= IFCRELDEFINESBYPROPERTIES ('19s4b2W4b8XxQ4ZKiC6008', #41, $, $, (#933, #814, #870, #399, #726, #518, #152), #6009);
#6009= IFCPROPERTYSET ('19s4b2W4b8XxQ4ZKiC6009', #41, 'Material_Mechanisch', $, (#6010));
#6010= IFCPROPERTYREFERENCEVALUE ('Beton-referenziert', $, $, #6011);
#6011= IFCLIBRARYREFERENCE ('RFEM-Materialbibliothek', 'Beton_C70/85', $);
/*Material_mechanisch "Stahl", zugewiesen zu den Lagern*/
#6012= IFCRELDEFINESBYPROPERTIES ('19s4b2W4b8XxQ4ZKiC6012', #41, $, $, (#1316, #1082, #1379, #1432, #1482, #1535, #1585, #1638), #6013);
#6013= IFCPROPERTYSET ('19s4b2W4b8XxQ4ZKiC6013', #41, 'Material_Mechanisch', $, (#6014));
#6014= IFCPROPERTYREFERENCEVALUE ('Stahl-referenziert', $, $, #6015);
#6015= IFCLIBRARYREFERENCE ('RFEM-Materialbibliothek', 'S_480_1.4462', 'RostfreierStahl');
/*Fahrbahntragwerk_semantisch*/
#6016= IFCRELDEFINESBYPROPERTIES ('19s4b2W4b8XxQ4ZKiC6016', #41, $, $, (#5000), #6017);
#6017= IFCPROPERTYSET ('19s4b2W4b8XxQ4ZKiC6017', #41, 'Fahrbahntragwerk_semantisch', $, (#6018, #6019));
#6018= IFCPROPERTYENUMERATEDVALUE ('Querschnittsform', $, ('Plattenbalken'), $);
#6019= IFCPROPERTYENUMERATEDVALUE ('BautechnischeStrukturFahrbahnplatte', $, ('Vollplatte'), $);
/*Vorspannung der Längsträger*/
#6020= IFCRELDEFINESBYPROPERTIES ('19s4b2W4b8XxQ4ZKiC6020', #41, $, $, (#814, #870), #6021);
#6021= IFCPROPERTYSET ('19s4b2W4b8XxQ4ZKiC6021', #41, 'Vorspannung', $, (#6022, #6100));
#6022= IFCCOMPLEXPROPERTY ('Spannglieder', 'Pseudocode:Contains_SET_{1:?}_OF_IfcComplexProperty_
WHERE: NameOfComplexProperty(1,2,...,n)=(Spannglied_1, Spannglied_2,..., Spannglied_n)',
'VorspannungLängsbalken', (#6023));
/*Spannglied_1*/
#6023= IFCCOMPLEXPROPERTY ('Spannglied_1', $, 'VerwendetInTrägerlund2', (#6024, #6040, #6070));
/*SpanngliedDaten*/
#6024= IFCCOMPLEXPROPERTY ('SpanngliedDaten', $, 'BeschreibtSpannglied_1',
(#6027, #6028, #6029, #6030, #6031, #6032));
#6027= IFCPROPERTYINGLEVALUE ('Litzenanzahl', $, 7.0, $);
#6028= IFCPROPERTYINGLEVALUE ('d_Huellrohr', $, 150.0, 'mm');
#6029= IFCPROPERTYENUMERATEDVALUE ('Material_Huellrohr', $, ('Kunststoff'), $);
#6030= IFCPROPERTYINGLEVALUE ('Vorspannkraft_vorh', $, 50.0, 'kN');
#6031= IFCPROPERTYINGLEVALUE ('Max_Zugkraft_zul', $, 70.0, 'kN');
#6032= IFCPROPERTYENUMERATEDVALUE ('ArtDerVorspannung', $, ('SofortigerVerbund'), $);
/*Verlauf XY*/
#6040= IFCCOMPLEXPROPERTY ('Verlauf_XY', $, 'BeschreibtVerlaufXYSpannglied_1', (#6041));
#6041= IFCCOMPLEXPROPERTY ('Segment_1', $, 'BeschreibtVerlaufXYSpannglied_1',
(#6042, #6043, #6044, #6045, #6046));
#6042= IFCPROPERTYLISTVALUE ('Position', '1. Eintrag=x-Koord. Anfang,
2. Eintrag=x-Koord. Ende', (0.0, 106.0), 'm');
#6043= IFCPROPERTYENUMERATEDVALUE ('Segmenttyp', $, ('2'), $);
/*Anfangspunkt C*/
#6044= IFCCOMPLEXPROPERTY
('Anfangspunkt_C', $, 'BeschreibtAnfangSegment_1', (#6047, #6048, #6049));
#6047= IFCPROPERTYENUMERATEDVALUE ('V-Referenzpunkt', $,
('Schwerpunkt_Zcg'), $);
#6048= IFCPROPERTYINGLEVALUE ('V-Abstand_RefP', $, 1.0, 'm');
#6049= IFCPROPERTYINGLEVALUE ('H-Gerade_Laenge_Ls,c', $, 0.0, 'm');
/*Endpunkt C*/
#6045= IFCCOMPLEXPROPERTY ('Endpunkt_C', $, 'BeschreibtEndeSegment_1',
(#6047, #6048, #6049));
/*Punkt Gerade-Parabel_S-P*/
#6046= IFCCOMPLEXPROPERTY ('Gerade-Parabel_S-P', $, 'BeschreibtMitteSegment_1',
(#6051, #6052, #6053, #6054, #6055, #6056));
#6051= IFCPROPERTYENUMERATEDVALUE ('V-Referenzpunkt', $,
('Schwerpunkt_Zcg'), $);
#6052= IFCPROPERTYINGLEVALUE ('V-Abstand_RefP', $, -1.0, 'm');
#6053= IFCPROPERTYENUMERATEDVALUE ('H-Referenzpunkt', $,
('SegmentAnfang'), $);
#6054= IFCPROPERTYINGLEVALUE ('H-Laenge_realtiv', $, .FALSE, $);
#6055= IFCPROPERTYINGLEVALUE ('H-Abstand_h_S-P', $, 43.0, 'm');
#6056= IFCPROPERTYINGLEVALUE ('H-Gerade_Laenge_L_S-P', $, 20.0, 'm');
/*Verlauf XZ*/
#6070= IFCCOMPLEXPROPERTY ('Verlauf_XZ', $, 'BeschreibtVerlaufXZSpannglied_1', (#6071));
#6071= IFCCOMPLEXPROPERTY ('Segment_1', $, 'BeschreibtVerlaufXZSpannglied_1',
(#6072, #6073, #6074, #6080));
#6072= IFCPROPERTYLISTVALUE ('Position', '1. Eintrag=x-Koord. Anfang,
2. Eintrag=x-Koord. Ende', (0.0, 106.0), 'm');
#6073= IFCPROPERTYENUMERATEDVALUE ('Segmenttyp', $, ('1'), $);
/*Anfangspunkt C*/
#6074= IFCCOMPLEXPROPERTY ('Anfangspunkt_C', $,
'BeschreibtAnfangSegment_1', (#6075, #6076, #6077));
#6075= IFCPROPERTYENUMERATEDVALUE ('V-Referenzpunkt', $,
('Schwerpunkt_Zcg'), $);
#6076= IFCPROPERTYINGLEVALUE ('V-Abstand_RefP', $, 0.0, 'm');
#6077= IFCPROPERTYINGLEVALUE ('H-Gerade_Laenge_Ls,c', $, 0.0, 'm');
/*Endpunkt C*/
#6080= IFCCOMPLEXPROPERTY ('Endpunkt_C', $,
'BeschreibtEndeSegment_1', (#6075, #6076, #6077));
/*Ersatzlasten*/
#6100= IFCCOMPLEXPROPERTY ('Ersatzlasten', $,
'BeschreibtSchnittkräfteAusVorspannung', (#6101, #6102, #6103, #6104, #6105));
#6101= IFCPROPERTYTABLEVALUE ('N_x', $, (0.0, 53.0, 106.0), (-60.0, -70.0, -60.0), $, 'm', 'kN');
#6102= IFCPROPERTYTABLEVALUE ('Q_y', $, (0.0, 53.0, 106.0), (0.0, 0.0, 0.0), $, 'm', 'kN');
#6103= IFCPROPERTYTABLEVALUE ('Q_z', $, (0.0, 53.0, 106.0), (-10.0, 0.0, 10.0), $, 'm', 'kN');
#6104= IFCPROPERTYTABLEVALUE ('M_y', $, (0.0, 53.0, 106.0), (-30.0, -50.0, -30.0), $, 'm', 'kNm');
#6105= IFCPROPERTYTABLEVALUE ('M_z', $, (0.0, 53.0, 106.0), (0.0, 0.0, 0.0), $, 'm', 'kNm');

```

/\*Brückenausrüstung\*/

```
/*Fahrbahn_semantisch*/
#6200= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6200',#41,$,$,(#5007),#6201);
#6201= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6201',#41,'Fahrbahn_semantisch',$,(#6203,#6204,#6205,#6206,#6207));
#6203= IFCPROPERTYSINGLEVALUE('Beschreibung',$,'Asphaltfahrbahn',$);
#6204= IFCPROPERTYSINGLEVALUE('Breite',$,16.0,'m');
#6205= IFCPROPERTYLISTVALUE('BezeichnungSchichten',$,('Dichtungsschicht','Schutzschicht','Deckschicht'),$);
#6206= IFCPROPERTYLISTVALUE('SchichtDicken',$,(0.01,0.05,0.05),'m');
#6207= IFCPROPERTYLISTVALUE('SchichtWichten',$,(1.7,1.5,0.8),'kN/m^3');
#6199= IFCPROPERTYSINGLEVALUE('Position','AbstandM_FahrbahnZu_M_Brücke',0.0,'m');
/*Fahrbahnübergang_semantisch*/
#6208= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6208',#41,$,$,(#90000),#6209);
#6209= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6209',#41,'Fahrbahnübergang_semantisch',$,(#6210,#6211,#6212));
#6210= IFCPROPERTYENUMERATEDVALUE('Typ',$,('Fingerübergang'),$);
#6211= IFCPROPERTYLISTVALUE('MaxVerschiebung',$,(0.30,0.05,0.05),'m');
#6212= IFCPROPERTYLISTVALUE('MaxVerdrehung',$,(0.0,10.0,1.0),'Grad');
/*Kappen_semantisch*/
#6215= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6215',#41,$,$,(#90001),#6216);
#6216= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6216',#41,'Kappen_semantisch',$,(#6217,#6218,#6219,#6220,#6221));
#6217= IFCPROPERTYSINGLEVALUE('Beschreibung',$,$);
#6218= IFCPROPERTYENUMERATEDVALUE('Funktion',$,('Notweg'),$);
#6219= IFCPROPERTYSINGLEVALUE('Breite',$,1.0,'m');
#6220= IFCPROPERTYSINGLEVALUE('Höhe',$,0.15,'m');
#6221= IFCPROPERTYSINGLEVALUE('Wichte',$,1.7,'kN/m^3');
/*Leiteinrichtungen*/
#6230= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6230',#41,$,$,(#5009),#6231);
#6231= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6231',#41,'Leiteinrichtungen_semantisch',$,(#6232,#6233));
#6232= IFCPROPERTYSINGLEVALUE('Beschreibung',$,'LeitplankeStandardTyp+GeländerHöhe1.30m',$);
#6233= IFCPROPERTYSINGLEVALUE('Windangriffsfläche',$,0.7,'m^2');
/*Punktlager_semantisch*/
#6240= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6240',#41,$,$,(#5013,#5015,#5017,#5019),#6241);
#6241= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6241',#41,'Punktlager_semantisch',$,(#6242,#6243,#6244,#6245,#6246));
#6242= IFCPROPERTYENUMERATEDVALUE('Lagertyp',$,('Verformungslager'),$);
#6243= IFCPROPERTYSINGLEVALUE('Verankert',$,.TRUE.,$);
#6244= IFCPROPERTYLISTVALUE('Max_Kraefte',$,(70.0,70.0,500.0,0.0),'kN');
#6245= IFCPROPERTYLISTVALUE('Max_Verschiebungen',$,(0.1,0.1,0.2,0.0),'m');
#6246= IFCPROPERTYLISTVALUE('Max_Verdrehungen',$,(5.0,5.0,5.0),'Grad');
/*Punktlager_mechanisch*/
#6250= IFCRELDEFINESBYPROPERTIES('19s4b2W4b8XxQ4ZKiC6250',#41,$,$,(#5013,#5015,#5017,#5019),#6251);
#6251= IFCPROPERTYSET('19s4b2W4b8XxQ4ZKiC6251',#41,'Punktlager_mechanisch',$,(#6252,#6253,#6254,#6255));
#6252= IFCPROPERTYENUMERATEDVALUE('Bezugsachsensystem',$,('global'),$);
#6253= IFCPROPERTYLISTVALUE('Lineare_Gelenkbedingungen_Translation',$,(700.0,700.0,50000.0,0.0),'kN/m');
#6254= IFCPROPERTYLISTVALUE('Lineare_Gelenkbedingungen_Rotation',$,(5000.0,5000.0,300.0,0.0),'kNm/rad');
#6255= IFCCOMPLEXPROPERTY('Nichtlineare_Gelenkbedingungen',$,'EntältAlleNLGBbezogenAuf6Freiheitsgrade',(#6260));
#6260= IFCCOMPLEXPROPERTY('NL_Ux',$,'BeschreibtNLGBallerLagerinGlobaleX_Richtung',(#6270,#6271));
#6270= IFCPROPERTYENUMERATEDVALUE('Typ',$,('FließenUndSchlupf'),$);
#6271= IFCCOMPLEXPROPERTY('Spezifikation_FließenUndSchlupf',$,
'SpezifiziertSchlupfAllerLager',(#6273,#6274));
#6273= IFCPROPERTYSINGLEVALUE('SchlupfInPositiveRichtung',$,0.1,'m');
#6274= IFCPROPERTYSINGLEVALUE('SchlupfInNegativeRichtung',$,0.1,'m');
```

/\*\*\*\*\*\* ENDE ANNOTATIONEN \*\*\*\*\*\*/

/\*Aus Autodesk Revit exportierte Daten\*/

```
#1= IFCORGANIZATION($,'Autodesk Revit 2015 (ENU)',,$,$);
#5= IFCAPPLICATION(#1,'2015','Autodesk Revit 2015 (ENU)',,'Revit');
#6= IFCCARTESIANPOINT((0.,0.,0.));
#9= IFCCARTESIANPOINT((0.,0.));
#11= IFCDIRECTION((1.,0.,0.));
#13= IFCDIRECTION((-1.,0.,0.));
#15= IFCDIRECTION((0.,1.,0.));
#17= IFCDIRECTION((0.,-1.,0.));
#19= IFCDIRECTION((0.,0.,1.));
#21= IFCDIRECTION((0.,0.,-1.));
#23= IFCDIRECTION((1.,0.));
#25= IFCDIRECTION((-1.,0.));
#27= IFCDIRECTION((0.,1.));
#29= IFCDIRECTION((0.,-1.));
#31= IFCAXIS2PLACEMENT3D(#6,$,$);
#32= IFCLOCALPLACEMENT(#1680,#31);
#35= IFCPERSON($,,'Andreas',,$,$,$);
#37= IFCORGANIZATION($,,'',$,$);
#38= IFCPERSONANDORGANIZATION(#35,#37,$);
#41= IFCOWNERHISTORY(#38,#5,$,.NOCHANGE.,,$,$,1488212656);
#42= IFCSIUNIT(*,.LENGTHUNIT.,$,METRE.);
#43= IFCSIUNIT(*,.AREAUNIT.,$,SQUARE_METRE.);
#44= IFCSIUNIT(*,.VOLUMEUNIT.,$,CUBIC_METRE.);
#45= IFCSIUNIT(*,.PLANEANGLEUNIT.,$,RADIANT.);
#46= IFCDIMENSIONALEXONENTS(0,0,0,0,0,0);
#47= IFCMEASUREWITHUNIT(IFCRATIOMEASURE(0.0174532925199433),#45);
#48= IFCCONVERSIONBASEDUNIT(#46,.PLANEANGLEUNIT.,'DEGREE',#47);
#49= IFCSIUNIT(*,.MASSUNIT.,.KILO.,.GRAM.);
#50= IFCSIUNIT(*,.TIMEUNIT.,$,SECOND.);
#51= IFCSIUNIT(*,.FREQUENCYUNIT.,$,HERTZ.);
#52= IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,KELVIN.);
#53= IFCSIUNIT(*,.THERMODYNAMICTEMPERATUREUNIT.,$,DEGREE_CELSIUS.);
```



```

#264= IFCFACE (#263);
#266= IFCPOLYLOOP (#223,#221,#248,#246);
#268= IFCFACEOUTERBOUND (#266,.T.);
#269= IFCFACE (#268);
#271= IFCCARTESIANPOINT ((11.,0.,0.));
#273= IFCCARTESIANPOINT ((3.,0.,0.));
#275= IFCCARTESIANPOINT ((3.,2.,0.));
#277= IFCCARTESIANPOINT ((8.,2.,0.));
#279= IFCPOLYLOOP (#250,#248,#221,#237,#271,#273,#275,#277);
#281= IFCFACEOUTERBOUND (#279,.T.);
#282= IFCFACE (#281);
#284= IFCPOLYLOOP (#225,#244,#254,#227);
#286= IFCFACEOUTERBOUND (#284,.T.);
#287= IFCFACE (#286);
#289= IFCCARTESIANPOINT ((11.,0.,10.));
#291= IFCCARTESIANPOINT ((7.6999999999999996,0.,10.));
#293= IFCCARTESIANPOINT ((7.6999999999999995,0.,15.1));
#295= IFCCARTESIANPOINT ((5.,0.,15.1));
#297= IFCCARTESIANPOINT ((5.,0.,10.));
#299= IFCCARTESIANPOINT ((0.,0.,10.));
#301= IFCCARTESIANPOINT ((0.,0.,8.));
#303= IFCPOLYLOOP (#271,#289,#291,#293,#295,#297,#299,#301,#273);
#305= IFCFACEOUTERBOUND (#303,.T.);
#306= IFCFACE (#305);
#308= IFCCARTESIANPOINT ((8.,2.,7.));
#310= IFCPOLYLOOP (#250,#277,#308,#252);
#312= IFCFACEOUTERBOUND (#310,.T.);
#313= IFCFACE (#312);
#315= IFCPOLYLOOP (#271,#237,#235,#289);
#317= IFCFACEOUTERBOUND (#315,.T.);
#318= IFCFACE (#317);
#320= IFCCARTESIANPOINT ((0.,2.,8.));
#322= IFCCARTESIANPOINT ((0.,2.,10.));
#324= IFCCARTESIANPOINT ((5.,2.,10.));
#326= IFCPOLYLOOP (#277,#275,#320,#322,#324,#308);
#328= IFCFACEOUTERBOUND (#326,.T.);
#329= IFCFACE (#328);
#331= IFCPOLYLOOP (#301,#299,#322,#320);
#333= IFCFACEOUTERBOUND (#331,.T.);
#334= IFCFACE (#333);
#336= IFCPOLYLOOP (#273,#301,#320,#275);
#338= IFCFACEOUTERBOUND (#336,.T.);
#339= IFCFACE (#338);
#341= IFCPOLYLOOP (#229,#227,#254,#324,#297,#295);
#343= IFCFACEOUTERBOUND (#341,.T.);
#344= IFCFACE (#343);
#346= IFCPOLYLOOP (#231,#229,#295,#293);
#348= IFCFACEOUTERBOUND (#346,.T.);
#349= IFCFACE (#348);
#351= IFCPOLYLOOP (#291,#233,#231,#293);
#353= IFCFACEOUTERBOUND (#351,.T.);
#354= IFCFACE (#353);
#356= IFCPOLYLOOP (#254,#252,#308,#324);
#358= IFCFACEOUTERBOUND (#356,.T.);
#359= IFCFACE (#358);
#361= IFCPOLYLOOP (#322,#299,#297,#324);
#363= IFCFACEOUTERBOUND (#361,.T.);
#364= IFCFACE (#363);
#366= IFCPOLYLOOP (#289,#235,#233,#291);
#368= IFCFACEOUTERBOUND (#366,.T.);
#369= IFCFACE (#368);
#371= IFCCLOSEDSHELL (#242,#259,#264,#269,#282,#287,#306,#313,#318,#329,#334,#339,#344,#349,#354,#359,#364,#369);
#373= IFCFACETEDBREP (#371);
#374= IFCCOLORRGB ($,0.498039215686275,0.498039215686275,0.498039215686275);
#375= IFCSURFACESTYLERENDERING (#374,0.,$, $, $, $, IFCNORMALISEDRTIOMEASURE (0.5), IFCSPECULAREXPONENT (64.),.NOTDEFINED.);
#376= IFCSURFACESTYLE ('Default',.BOTH.,(#375));
#378= IFCPRESENTATIONSTYLEASSIGNMENT ((#376));
#380= IFCSTYLEDITEM (#373, (#378), $);
#383= IFCSHAPEREPRESENTATION (#87, 'Body', 'Brep', (#373));
#385= IFCCARTESIANPOINT ((-8.,-20.,10.));
#387= IFCAXIS2PLACEMENT3D (#6, $, $);
#388= IFCREPRESENTATIONMAP (#387, #383);
#389= IFCMAPPEDITEM (#388, #138);
#391= IFCSHAPEREPRESENTATION (#87, 'Body', 'MappedRepresentation', (#389));
#393= IFCPRODUCTDEFINITIONSHAPE ($, $, (#391));
#395= IFCCARTESIANPOINT ((-2.77765649706652,-8.55890328535934,3.00964918888403));
#397= IFCAXIS2PLACEMENT3D (#395, $, $);
#398= IFCLOCALPLACEMENT (#110, #397);
/*Widerlager 1*/
#399= IFCBUILDINGELEMENTPROXY ('3naYnuw3X5h9HNJmVvFxfh', #41, 'Widerlager rechts:Widerlager rechts:310375', $, 'Widerlager
rechts', #398, #393, '310375',.ELEMENT.);
#402= IFCMATERIAL ('Default');
#405= IFCPRESENTATIONSTYLEASSIGNMENT ((#376));
#407= IFCSTYLEDITEM ($, (#405), $);
#409= IFCSTYLEDREPRESENTATION (#82, 'Style', 'Material', (#407));
#412= IFCMATERIALDEFINITIONREPRESENTATION ($, $, (#409), #402);
#461= IFCAXIS2PLACEMENT3D (#6, $, $);
#462= IFCLOCALPLACEMENT (#117, #487);
#464= IFCCARTESIANPOINT ((0.,11.));
#466= IFCCARTESIANPOINT ((1.,11.));

```

```

#468= IFCPOLYLINE ((#464,#466));
#470= IFCGEOMETRICCURVESET ((#468));
#472= IFCCOLOURRGB ($,0.,0.650980392156863,0.);
#473= IFCDRAUGHTINGPREDEFINEDCURVEFONT ('continuous');
#474= IFCCURVESTYLE ('Lines',#473,$,#472);
#475= IFCPRESENTATIONSTYLEASSIGNMENT ((#474));
#477= IFCSTYLEDITEM (#470, (#475), $);
#480= IFCSHAPEREPRESENTATION (#91, 'Annotation', 'Annotation2D', (#470));
#483= IFCPRODUCTDEFINITIONSHAPE ($, $, (#480));
#485= IFCCARTESIANPOINT ((-2.77765649706649,11.4410967146407,9.00964918888408));
#487= IFCAxis2PLACEMENT3D (#485, #19, #17);
#488= IFCANNOTATION ('1psFknQK5C7v56S2_W6cnY', #41, $, $, $, #462, #483);
#492= IFCAxis2PLACEMENT3D (#504, $, $);
#494= IFCCARTESIANPOINT ((0.800000000000009,2.20000000000002));
#496= IFCAxis2PLACEMENT2D (#494, #23);
#497= IFCRECTANGLEPROFILEDEF (.AREA., 'Fundament rechts', #496,9.99999999999998,22.0000000000002);
#498= IFCCARTESIANPOINT ((4.20000000000002,8.80000000000007,0.));
#500= IFCAxis2PLACEMENT3D (#498, $, $);
#501= IFCEXTRUDEDAREASOLID (#497, #500, #19, 3.);
#502= IFCSHAPEREPRESENTATION (#87, 'Body', 'SweptSolid', (#501));
#504= IFCCARTESIANPOINT ((102., -1., 7.));
#506= IFCAxis2PLACEMENT3D (#6, $, $);
#507= IFCREPRESENTATIONMAP (#506, #502);
#508= IFCMAPPEDITEM (#507, #138);
#510= IFCSHAPEREPRESENTATION (#87, 'Body', 'MappedRepresentation', (#508));
#512= IFCPRODUCTDEFINITIONSHAPE ($, $, (#510));
#514= IFCCARTESIANPOINT ((117.221343502934,12.4420967146406,0.0096491888854547));
#516= IFCAxis2PLACEMENT3D (#514, #19, #13);
#517= IFCLocalPLACEMENT (#110, #516);
/*Fundament 2*/
#518= IFCBUILDINGELEMENTPROXY ('10SYrHTOfEE9laepSUCdG7', #41, 'Fundament rechts:Fundament rechts:313810', $, 'Fundament
rechts', #517, #512, '313810', .ELEMENT.);
#552= IFCAxis2PLACEMENT3D (#712, $, $);
#554= IFCCARTESIANPOINT ((3., 0., 0.));
#556= IFCCARTESIANPOINT ((11., 0., 0.));
#558= IFCCARTESIANPOINT ((11., 0., 10.));
#560= IFCCARTESIANPOINT ((7.70000000000002, 0., 10.));
#562= IFCCARTESIANPOINT ((7.70000000000001, 0., 15.1));
#564= IFCCARTESIANPOINT ((5., 0., 15.1));
#566= IFCCARTESIANPOINT ((5., 0., 10.));
#568= IFCCARTESIANPOINT ((0., 0., 10.));
#570= IFCCARTESIANPOINT ((0., 0., 8.));
#572= IFCPOLYLOOP ((#554, #556, #558, #560, #562, #564, #566, #568, #570));
#574= IFCFACEOUTERBOUND (#572, .T.);
#575= IFCFACE ((#574));
#577= IFCCARTESIANPOINT ((0., 2., 10.));
#579= IFCCARTESIANPOINT ((5., 2., 10.));
#581= IFCCARTESIANPOINT ((8., 2., 7.));
#583= IFCCARTESIANPOINT ((8., 2., 0.));
#585= IFCCARTESIANPOINT ((3., 2., 0.));
#587= IFCCARTESIANPOINT ((0., 2., 8.));
#589= IFCPOLYLOOP ((#577, #579, #581, #583, #585, #587));
#591= IFCFACEOUTERBOUND (#589, .T.);
#592= IFCFACE ((#591));
#594= IFCPOLYLOOP ((#568, #577, #587, #570));
#596= IFCFACEOUTERBOUND (#594, .T.);
#597= IFCFACE ((#596));
#599= IFCPOLYLOOP ((#570, #587, #585, #554));
#601= IFCFACEOUTERBOUND (#599, .T.);
#602= IFCFACE ((#601));
#604= IFCCARTESIANPOINT ((8., 18., 0.));
#606= IFCCARTESIANPOINT ((3., 18., 0.));
#608= IFCCARTESIANPOINT ((3., 20., 0.));
#610= IFCCARTESIANPOINT ((11., 20., 0.));
#612= IFCPOLYLOOP ((#583, #604, #606, #608, #610, #556, #554, #585));
#614= IFCFACEOUTERBOUND (#612, .T.);
#615= IFCFACE ((#614));
#617= IFCPOLYLOOP ((#568, #566, #579, #577));
#619= IFCFACEOUTERBOUND (#617, .T.);
#620= IFCFACE ((#619));
#622= IFCCARTESIANPOINT ((0., 20., 8.));
#624= IFCCARTESIANPOINT ((0., 20., 10.));
#626= IFCCARTESIANPOINT ((5., 20., 10.));
#628= IFCCARTESIANPOINT ((5., 20., 15.1));
#630= IFCCARTESIANPOINT ((7.69999999999995, 20., 15.1));
#632= IFCCARTESIANPOINT ((7.69999999999996, 20., 10.));
#634= IFCCARTESIANPOINT ((11., 20., 10.));
#636= IFCPOLYLOOP ((#610, #608, #622, #624, #626, #628, #630, #632, #634));
#638= IFCFACEOUTERBOUND (#636, .T.);
#639= IFCFACE ((#638));
#641= IFCCARTESIANPOINT ((8., 18., 7.));
#643= IFCPOLYLOOP ((#583, #581, #641, #604));
#645= IFCFACEOUTERBOUND (#643, .T.);
#646= IFCFACE ((#645));
#648= IFCPOLYLOOP ((#610, #634, #558, #556));
#650= IFCFACEOUTERBOUND (#648, .T.);
#651= IFCFACE ((#650));
#653= IFCCARTESIANPOINT ((5., 18., 10.));
#655= IFCCARTESIANPOINT ((0., 18., 10.));
#657= IFCCARTESIANPOINT ((0., 18., 8.));

```

```

#659= IFCPOLYLOOP ((#604,#641,#653,#655,#657,#606));
#661= IFCFACEOUTERBOUND(#659,.T.);
#662= IFCFACE((#661));
#664= IFCPOLYLOOP ((#622,#657,#655,#624));
#666= IFCFACEOUTERBOUND(#664,.T.);
#667= IFCFACE((#666));
#669= IFCPOLYLOOP ((#608,#606,#657,#622));
#671= IFCFACEOUTERBOUND(#669,.T.);
#672= IFCFACE((#671));
#674= IFCPOLYLOOP ((#564,#628,#626,#653,#579,#566));
#676= IFCFACEOUTERBOUND(#674,.T.);
#677= IFCFACE((#676));
#679= IFCPOLYLOOP ((#562,#630,#628,#564));
#681= IFCFACEOUTERBOUND(#679,.T.);
#682= IFCFACE((#681));
#684= IFCPOLYLOOP ((#632,#630,#562,#560));
#686= IFCFACEOUTERBOUND(#684,.T.);
#687= IFCFACE((#686));
#689= IFCPOLYLOOP ((#579,#653,#641,#581));
#691= IFCFACEOUTERBOUND(#689,.T.);
#692= IFCFACE((#691));
#694= IFCPOLYLOOP ((#655,#653,#626,#624));
#696= IFCFACEOUTERBOUND(#694,.T.);
#697= IFCFACE((#696));
#699= IFCPOLYLOOP ((#634,#632,#560,#558));
#701= IFCFACEOUTERBOUND(#699,.T.);
#702= IFCFACE((#701));
#704= IFCCLOSEDSHELL((#575,#592,#597,#602,#615,#620,#639,#646,#651,#662,#667,#672,#677,#682,#687,#692,#697,#702));
#706= IFCFACETEDBREP(#704);
#707= IFCSTYLEDITEM(#706, (#378), $);
#710= IFCSHAPEREPRESENTATION(#87, 'Body', 'Brep', (#706));
#712= IFCCARTESIANPOINT((-8.,0.,10.));
#714= IFCAXIS2PLACEMENT3D(#6,$,$);
#715= IFCREPRESENTATIONMAP(#714,#710);
#716= IFCMAPPEDITEM(#715,#138);
#718= IFCSHAPEREPRESENTATION(#87, 'Body', 'MappedRepresentation', (#716));
#720= IFCPRODUCTDEFINITIONSHAPE($,$, (#718));
#722= IFCCARTESIANPOINT((119.222343502934,11.4410967146407,3.00964918888449));
#724= IFCAXIS2PLACEMENT3D(#722,#19,#13);
#725= IFCLocalPLACEMENT(#110,#724);
/*Widerlager 2*/
#726= IFCBUILDINGELEMENTPROXY('10SyrHTOfEE91aepSUCdG6',#41,'Widerlager rechts:Widerlager rechts:313811',$,'Widerlager
rechts',#725,#720,'313811',.ELEMENT.);
#760= IFCAXIS2PLACEMENT3D(#6,$,$);
#761= IFCLocalPLACEMENT(#117,#781);
#762= IFCCARTESIANPOINT((11.,0.));
#764= IFCCARTESIANPOINT((11.,1.));
#766= IFCPOLYLINE((#762,#764));
#768= IFCGEOMETRICCURVESET((#766));
#770= IFCPRESENTATIONSTYLEASSIGNMENT((#474));
#772= IFCSTYLEDITEM(#768, (#770), $);
#775= IFCSHAPEREPRESENTATION(#91, 'Annotation', 'Annotation2D', (#768));
#777= IFCPRODUCTDEFINITIONSHAPE($,$, (#775));
#779= IFCCARTESIANPOINT((119.222343502934,11.4410967146407,9.00964918888453));
#781= IFCAXIS2PLACEMENT3D(#779,#19,#13);
#782= IFCANNOTATION('0ei27klv9DWvn9mlBaBdth',#41,$,$,$,#761,#777);
#785= IFCAXIS2PLACEMENT3D(#800,$,$);
#787= IFCCARTESIANPOINT((-1.38777878078145E-16,0.));
#789= IFCAXIS2PLACEMENT2D(#787,#23);
#790= IFCRECTANGLEPROFILEDEF(.AREA., 'L\X2\00E4\X0\ngstr\X2\00E4\X0\ger',#789,1.,106.);
#791= IFCCARTESIANPOINT((53.,0.500000000000018,0.));
#793= IFCAXIS2PLACEMENT3D(#791,#19,#15);
#794= IFCEXTRUDEDAREASOLID(#790,#793,#19,2.99999999999999);
#795= IFCSTYLEDITEM(#794, (#378), $);
#798= IFCSHAPEREPRESENTATION(#87, 'Body', 'SweptSolid', (#794));
#800= IFCCARTESIANPOINT((0.,-2.,20.79999999999996));
#802= IFCAXIS2PLACEMENT3D(#6,$,$);
#803= IFCREPRESENTATIONMAP(#802,#798);
#804= IFCMAPPEDITEM(#803,#138);
#806= IFCSHAPEREPRESENTATION(#87, 'Body', 'MappedRepresentation', (#804));
#808= IFCPRODUCTDEFINITIONSHAPE($,$, (#806));
#810= IFCCARTESIANPOINT((111.222343502933,-6.55890328535903,9.80964918888444));
#812= IFCAXIS2PLACEMENT3D(#810,#19,#13);
#813= IFCLocalPLACEMENT(#117,#812);
/*Längsträger 1*/
#814= IFCBUILDINGELEMENT-
PROXY('0_I9qgNft2x02XwnqevCRy',#41,'L\X2\00E4\X0\ngstr\X2\00E4\X0\ger:L\X2\00E4\X0\ngstr\X2\00E4\X0\ger:315867',$,'L\X
2\00E4\X0\ngstr\X2\00E4\X0\ger',#813,#808,'315867',.ELEMENT.);
#861= IFCMAPPEDITEM(#803,#138);
#862= IFCSHAPEREPRESENTATION(#87, 'Body', 'MappedRepresentation', (#861));
#864= IFCPRODUCTDEFINITIONSHAPE($,$, (#862));
#866= IFCCARTESIANPOINT((111.222343502933,10.441096714641,9.8096491888845));
#868= IFCAXIS2PLACEMENT3D(#866,#19,#13);
#869= IFCLocalPLACEMENT(#117,#868);
/*Längsträger 2*/
#870= IFCBUILDINGELE-
MENTPROXY('0_I9qgNft2x02XwnqevCNx',#41,'L\X2\00E4\X0\ngstr\X2\00E4\X0\ger:L\X2\00E4\X0\ngstr\X2\00E4\X0\ger:316124',$,'
L\X2\00E4\X0\ngstr\X2\00E4\X0\ger',#869,#864,'316124',.ELEMENT.);
#904= IFCAXIS2PLACEMENT3D(#919,$,$);
#906= IFCCARTESIANPOINT((0.,0.));

```

```

#908= IFXCAXIS2PLACEMENT2D(#906,#23);
#909= IFCRECTANGLEPROFILEDEF(.AREA., 'Fahrbahnplatte', #908, 106., 20.);
#910= IFCCARTESIANPOINT((53., 10., 0.));
#912= IFXCAXIS2PLACEMENT3D(#910, #19, #13);
#913= IFCEXTRUDEDAREASOLID(#909, #912, #19, 1.300000000000004);
#914= IFCSTYLEDITEM(#913, (#378), $);
#917= IFCSHAPEREPRESENTATION(#87, 'Body', 'SweptSolid', (#913));
#919= IFCCARTESIANPOINT((0., -20., 23.8));
#921= IFXCAXIS2PLACEMENT3D(#6, $, $);
#922= IFCREPRESENTATIONMAP(#921, #917);
#923= IFCMAPPEDITEM(#922, #138);
#925= IFCSHAPEREPRESENTATION(#87, 'Body', 'MappedRepresentation', (#923));
#927= IFCPRODUCTDEFINITIONSHAPE($, $, (#925));
#929= IFCCARTESIANPOINT((111.222343502933, 11.4410967146407, 12.8096491888845));
#931= IFXCAXIS2PLACEMENT3D(#929, #19, #13);
#932= IFCLocalPLACEMENT(#117, #931);
/*Fahrbahnplatte*/
#933= IFCBUILDINGELEMENTPROXY('1G1uRq$0f2NqZbhgu$ZkiD', #41, 'Fahrbahnplatte:Fahrbahnplatte:316998', $, 'Fahrbahnplatte', #932, #927, '316998', .ELEMENT.);
#980= IFXCAXIS2PLACEMENT3D(#1068, $, $);
#982= IFCCARTESIANPOINT((1., 1., 0.4));
#984= IFCCARTESIANPOINT((0., 1., 0.4));
#986= IFCCARTESIANPOINT((0., 0.5000000000000003, 0.4));
#988= IFCCARTESIANPOINT((0., 0., 0.4));
#990= IFCCARTESIANPOINT((1., 0., 0.4));
#992= IFCPOLYLOOP((#982, #984, #986, #988, #990));
#994= IFCFACEOUTERBOUND(#992, .T.);
#995= IFCFACE((#994));
#997= IFCCARTESIANPOINT((1., 1., 0.1999999999999998));
#999= IFCCARTESIANPOINT((0., 1., 0.1999999999999998));
#1001= IFCPOLYLOOP((#984, #982, #997, #999));
#1003= IFCFACEOUTERBOUND(#1001, .T.);
#1004= IFCFACE((#1003));
#1006= IFCCARTESIANPOINT((0., 0.5000000000000003, 0.2000000000000002));
#1008= IFCPOLYLOOP((#986, #984, #999, #1006));
#1010= IFCFACEOUTERBOUND(#1008, .T.);
#1011= IFCFACE((#1010));
#1013= IFCCARTESIANPOINT((0., 0., 0.2000000000000002));
#1015= IFCPOLYLOOP((#988, #986, #1006, #1013));
#1017= IFCFACEOUTERBOUND(#1015, .T.);
#1018= IFCFACE((#1017));
#1020= IFCCARTESIANPOINT((1., 0., 0.1999999999999998));
#1022= IFCPOLYLOOP((#990, #988, #1013, #1020));
#1024= IFCFACEOUTERBOUND(#1022, .T.);
#1025= IFCFACE((#1024));
#1027= IFCPOLYLOOP((#982, #990, #1020, #997));
#1029= IFCFACEOUTERBOUND(#1027, .T.);
#1030= IFCFACE((#1029));
#1032= IFCCARTESIANPOINT((0.5000000000000001, 0.5000000000000002, 0.));
#1034= IFCPOLYLOOP((#999, #1032, #1013, #1006));
#1036= IFCFACEOUTERBOUND(#1034, .T.);
#1037= IFCFACE((#1036));
#1039= IFCPOLYLOOP((#1032, #997, #1020));
#1041= IFCFACEOUTERBOUND(#1039, .T.);
#1042= IFCFACE((#1041));
#1044= IFCPOLYLOOP((#1032, #999, #997));
#1046= IFCFACEOUTERBOUND(#1044, .T.);
#1047= IFCFACE((#1046));
#1049= IFCPOLYLOOP((#1013, #1032, #1020));
#1051= IFCFACEOUTERBOUND(#1049, .T.);
#1052= IFCFACE((#1051));
#1054= IFCCLOSEDSHELL((#995, #1004, #1011, #1018, #1025, #1030, #1037, #1042, #1047, #1052));
#1056= IFCFACETEDBREP(#1054);
#1057= IFCCOLORRGB($, 0.784313725490196, 0., 0.);
#1058= IFCSURFACESTYLERENDERING(#1057, 0., $, $, $, $, IFCNORMALISEDRAIOMEASURE(0.5), IFCSPECULAREXPONENT(64.), .NOTDEFINED.);
#1059= IFCSURFACESTYLE('A Bauteil mit Freiheitsgraden', .BOTH., (#1058));
#1061= IFCPRESENTATIONSTYLEASSIGNMENT((#1059));
#1063= IFCSTYLEDITEM(#1056, (#1061), $);
#1066= IFCSHAPEREPRESENTATION(#87, 'Body', 'Brep', (#1056));
#1068= IFCCARTESIANPOINT((1., -2., 20.4));
#1070= IFXCAXIS2PLACEMENT3D(#6, $, $);
#1071= IFCREPRESENTATIONMAP(#1070, #1066);
#1072= IFCMAPPEDITEM(#1071, #138);
#1074= IFCSHAPEREPRESENTATION(#87, 'Body', 'MappedRepresentation', (#1072));
#1076= IFCPRODUCTDEFINITIONSHAPE($, $, (#1074));
#1078= IFCCARTESIANPOINT((7.22234350293348, -6.55890328535933, 9.40964918888396));
#1080= IFXCAXIS2PLACEMENT3D(#1078, #19, #13);
#1081= IFCLocalPLACEMENT(#117, #1080);
/*Lager 1 Oberteil */
#1082= IFCBUILDINGELEMENTPROXY('1nDKsjRzr2FhZL WPHTuUw', #41, 'Punktlager_Oberteil:Punktlager_Oberteil:320304', $, 'Punktlager_Oberteil', #1081, #1076, '320304', .ELEMENT.);
#1085= IFCMATERIAL('A Bauteil mit Freiheitsgraden');
#1086= IFCPRESENTATIONSTYLEASSIGNMENT((#1059));
#1088= IFCSTYLEDITEM($, (#1086), $);
#1090= IFCSTYLEDREPRESENTATION(#82, 'Style', 'Material', (#1088));
#1092= IFCMATERIALDEFINITIONREPRESENTATION($, $, (#1090), #1085);
#1145= IFXCAXIS2PLACEMENT3D(#1249, $, $);

```

```

#1147= IFCCARTESIANPOINT ((0.,1.,0.20000000000000007));
#1149= IFCCARTESIANPOINT ((0.,0.,0.20000000000000007));
#1151= IFCCARTESIANPOINT ((0.50000000000000003,0.50000000000000002,0.40000000000000004));
#1153= IFCPOLYLOOP ((#1147,#1149,#1151));
#1155= IFCFACEOUTERBOUND (#1153,.T.);
#1156= IFCFACE ((#1155));
#1158= IFCCARTESIANPOINT ((1.,1.,0.20000000000000007));
#1160= IFCCARTESIANPOINT ((1.,0.,0.20000000000000007));
#1162= IFCPOLYLOOP ((#1149,#1147,#1158,#1160));
#1164= IFCFACEOUTERBOUND (#1162,.T.);
#1165= IFCFACE ((#1164));
#1167= IFCPOLYLOOP ((#1160,#1158,#1151));
#1169= IFCFACEOUTERBOUND (#1167,.T.);
#1170= IFCFACE ((#1169));
#1172= IFCPOLYLOOP ((#1147,#1151,#1158));
#1174= IFCFACEOUTERBOUND (#1172,.T.);
#1175= IFCFACE ((#1174));
#1177= IFCPOLYLOOP ((#1151,#1149,#1160));
#1179= IFCFACEOUTERBOUND (#1177,.T.);
#1180= IFCFACE ((#1179));
#1182= IFCCARTESIANPOINT ((1.,1.,0.20000000000000007));
#1184= IFCCARTESIANPOINT ((0.,1.,0.20000000000000007));
#1186= IFCCARTESIANPOINT ((0.,0.50000000000000006,0.20000000000000007));
#1188= IFCCARTESIANPOINT ((0.,0.,0.20000000000000007));
#1190= IFCCARTESIANPOINT ((1.,0.,0.20000000000000007));
#1192= IFCPOLYLOOP ((#1182,#1184,#1186,#1188,#1190));
#1194= IFCFACEOUTERBOUND (#1192,.T.);
#1195= IFCFACE ((#1194));
#1197= IFCCARTESIANPOINT ((1.,1.,0.));
#1199= IFCCARTESIANPOINT ((1.,0.,0.));
#1201= IFCCARTESIANPOINT ((0.,0.50000000000000006,0.));
#1203= IFCCARTESIANPOINT ((0.,1.,0.));
#1205= IFCPOLYLOOP ((#1197,#1199,#6,#1201,#1203));
#1207= IFCFACEOUTERBOUND (#1205,.T.);
#1208= IFCFACE ((#1207));
#1210= IFCPOLYLOOP ((#1184,#1182,#1197,#1203));
#1212= IFCFACEOUTERBOUND (#1210,.T.);
#1213= IFCFACE ((#1212));
#1215= IFCPOLYLOOP ((#1186,#1184,#1203,#1201));
#1217= IFCFACEOUTERBOUND (#1215,.T.);
#1218= IFCFACE ((#1217));
#1220= IFCPOLYLOOP ((#1188,#1186,#1201,#6));
#1222= IFCFACEOUTERBOUND (#1220,.T.);
#1223= IFCFACE ((#1222));
#1225= IFCPOLYLOOP ((#1190,#1188,#6,#1199));
#1227= IFCFACEOUTERBOUND (#1225,.T.);
#1228= IFCFACE ((#1227));
#1230= IFCPOLYLOOP ((#1182,#1190,#1199,#1197));
#1232= IFCFACEOUTERBOUND (#1230,.T.);
#1233= IFCFACE ((#1232));
#1235= IFCCLOSEDSHELL ((#1156,#1165,#1170,#1175,#1180));
#1237= IFCFACETDBREP (#1235);
#1238= IFCSTYLELITEM (#1237, (#1061), $);
#1241= IFCCLOSEDSHELL ((#1195,#1208,#1213,#1218,#1223,#1228,#1233));
#1243= IFCFACETDBREP (#1241);
#1244= IFCSTYLELITEM (#1243, (#1061), $);
#1247= IFCSHAPEREPRESENTATION (#87, 'Body', 'Brep', (#1237,#1243));
#1249= IFCCARTESIANPOINT ((1.,-2.,20.));
#1251= IFCCARTESIANPOINT ((1.,0.));
#1253= IFCPOLYLINE ((#1251,#9));
#1255= IFCCARTESIANPOINT ((0.,0.50000000000000006));
#1257= IFCPOLYLINE ((#9,#1255));
#1259= IFCCARTESIANPOINT ((1.,0.));
#1261= IFCPOLYLINE ((#1259,#9));
#1263= IFCCARTESIANPOINT ((1.,1.));
#1265= IFCCARTESIANPOINT ((1.,0.));
#1267= IFCPOLYLINE ((#1263,#1265));
#1269= IFCCARTESIANPOINT ((0.,1.));
#1271= IFCCARTESIANPOINT ((1.,1.));
#1273= IFCPOLYLINE ((#1269,#1271));
#1275= IFCCARTESIANPOINT ((0.,0.50000000000000006));
#1277= IFCCARTESIANPOINT ((0.,1.));
#1279= IFCPOLYLINE ((#1275,#1277));
#1281= IFCCARTESIANPOINT ((1.,0.));
#1283= IFCPOLYLINE ((#1281,#9));
#1285= IFCCARTESIANPOINT ((1.,0.));
#1287= IFCPOLYLINE ((#1285,#9));
#1289= IFCCARTESIANPOINT ((0.,1.));
#1291= IFCCARTESIANPOINT ((1.,1.));
#1293= IFCPOLYLINE ((#1289,#1291));
#1295= IFCGEOMETRICSET ((#1253,#1257,#1261,#1267,#1273,#1279,#1283,#1287,#1293));
#1297= IFCSHAPEREPRESENTATION (#91, 'FootPrint', 'GeometricSet', (#1295));
#1299= IFCAxis2PLACEMENT3D (#6, $, $);
#1300= IFCREPRESENTATIONMAP (#1299,#1247);
#1301= IFCREPRESENTATIONMAP (#1299,#1297);
#1302= IFCMAPPEDITEM (#1300,#138);
#1304= IFCSHAPEREPRESENTATION (#87, 'Body', 'MappedRepresentation', (#1302));
#1306= IFCMAPPEDITEM (#1301,#138);
#1308= IFCSHAPEREPRESENTATION (#91, 'FootPrint', 'MappedRepresentation', (#1306));
#1310= IFCPRODUCTDEFINITIONSHAPE ($, $, (#1304,#1308));

```

```

#1312= IFCCARTESIANPOINT ((7.22234350293351,-6.55890328535934,9.00964918888397));
#1314= IFCAXIS2PLACEMENT3D (#1312,#19,#13);
#1315= IFCLOCALPLACEMENT (#117,#1314);
/*Lager 1 Unterteil*/
#1316= IFCBUILDINGELEMENT-
PROXY ('1nDKsJRzr2FhZL_WPHTu00',#41,'Punktlager_Unterteil:Punktlager_Unterteil:320658',$,'Punktlager_Unterteil',#1315,#
1310,'320658',.ELEMENT.);
#1370= IFCMAPPEDITEM (#1071,#138);
#1371= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1370));
#1373= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1371));
#1375= IFCCARTESIANPOINT ((7.22234350293348,10.4410967146407,9.40964918888391));
#1377= IFCAXIS2PLACEMENT3D (#1375,#19,#13);
#1378= IFCLOCALPLACEMENT (#117,#1377);
/*Lager 2 Oberteil*/
#1379= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xwa',#41,'Punktlager_Oberteil:Punktlager_Oberteil:321326',$,'Punktlager_Oberteil',#1378,#137
3,'321326',.ELEMENT.);
#1420= IFCMAPPEDITEM (#1300,#138);
#1421= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1420));
#1423= IFCMAPPEDITEM (#1301,#138);
#1424= IFCSHAPEREPRESENTATION (#91,'FootPrint','MappedRepresentation',(#1423));
#1426= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1421,#1424));
#1428= IFCCARTESIANPOINT ((7.22234350293351,10.4410967146407,9.00964918888392));
#1430= IFCAXIS2PLACEMENT3D (#1428,#19,#13);
#1431= IFCLOCALPLACEMENT (#117,#1430);
/*Lager 2 Unterteil*/
#1432= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xwb',#41,'Punktlager_Unterteil:Punktlager_Unterteil:321327',$,'Punktlager_Unterteil',#1431,#
1426,'321327',.ELEMENT.);
#1473= IFCMAPPEDITEM (#1071,#138);
#1474= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1473));
#1476= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1474));
#1478= IFCCARTESIANPOINT ((110.222343502934,-6.55890328535933,9.40964918888441));
#1480= IFCAXIS2PLACEMENT3D (#1478,#19,#13);
#1481= IFCLOCALPLACEMENT (#117,#1480);
/*Lager 3 Oberteil*/
#1482= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xus',#41,'Punktlager_Oberteil:Punktlager_Oberteil:321468',$,'Punktlager_Oberteil',#1481,#147
6,'321468',.ELEMENT.);
#1523= IFCMAPPEDITEM (#1300,#138);
#1524= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1523));
#1526= IFCMAPPEDITEM (#1301,#138);
#1527= IFCSHAPEREPRESENTATION (#91,'FootPrint','MappedRepresentation',(#1526));
#1529= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1524,#1527));
#1531= IFCCARTESIANPOINT ((110.222343502934,-6.55890328535934,9.00964918888442));
#1533= IFCAXIS2PLACEMENT3D (#1531,#19,#13);
#1534= IFCLOCALPLACEMENT (#117,#1533);
/*Lager 3 Unterteil*/
#1535= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xut',#41,'Punktlager_Unterteil:Punktlager_Unterteil:321469',$,'Punktlager_Unterteil',#1534,#
1529,'321469',.ELEMENT.);
#1576= IFCMAPPEDITEM (#1071,#138);
#1577= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1576));
#1579= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1577));
#1581= IFCCARTESIANPOINT ((110.222343502934,10.4410967146407,9.40964918888436));
#1583= IFCAXIS2PLACEMENT3D (#1581,#19,#13);
#1584= IFCLOCALPLACEMENT (#117,#1583);
/*Lager 4 Oberteil*/
#1585= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xvi',#41,'Punktlager_Oberteil:Punktlager_Oberteil:321510',$,'Punktlager_Oberteil',#1584,#157
9,'321510',.ELEMENT.);
#1626= IFCMAPPEDITEM (#1300,#138);
#1627= IFCSHAPEREPRESENTATION (#87,'Body','MappedRepresentation',(#1626));
#1629= IFCMAPPEDITEM (#1301,#138);
#1630= IFCSHAPEREPRESENTATION (#91,'FootPrint','MappedRepresentation',(#1629));
#1632= IFCPRODUCTDEFINITIONSHAPE ($,$, (#1627,#1630));
#1634= IFCCARTESIANPOINT ((110.222343502934,10.4410967146407,9.00964918888437));
#1636= IFCAXIS2PLACEMENT3D (#1634,#19,#13);
#1637= IFCLOCALPLACEMENT (#117,#1636);
/*Lager 4 Unterteil*/
#1638= IFCBUILDINGELEMENT-
PROXY ('1xelFVpP192fKmjWF25Xvj',#41,'Punktlager_Unterteil:Punktlager_Unterteil:321511',$,'Punktlager_Unterteil',#1637,#
1632,'321511',.ELEMENT.);
#1849= IFCRELASSOCIATESMATERIAL ('3k6TaxmvzFGg9i0aksDy8d',#41,$,$, (#399,#726,#814,#870,#933),#402);
#1856= IFCRELASSOCIATESMATERI-
AL ('09Wwx7BwbCrxTBT1GVCY4Z',#41,$,$, (#1082,#1316,#1379,#1432,#1482,#1535,#1585,#1638),#1085);
#1932= IFCPRESENTATIONLAYERASSIGNMENT ('A-DETL-____-OTLN',$, (#480,#775), $);
#1934= IFCPRESENTATIONLAYERASSIGNMENT ('A-GENM-____-
OTLN',$, (#130,#141,#383,#391,#502,#510,#710,#718,#798,#806,#862,#917,#925,#1066,#1074,#1247,#1297,#1304,#1308,#1371,#1
421,#1424,#1474,#1524,#1527,#1577,#1627,#1630), $);

```

ENDSEC;

END-ISO-10303-21;

## Anlage 2: Verifikation der Beispielinstantz (Ergänzung zu Kapitel 8)

Im Folgenden werden das Beispielmodell der Balkenbrücke in Diagrammform veranschaulicht und dessen Instanzierbarkeit als *IFC2x3*-Datei mittels Screenshots des *FZK-Viewer* verifiziert.

### Semantische Beziehungen innerhalb der IFC-Datei

Die folgende Grafik ist als Veranschaulichung der semantischen Beziehungen zwischen den Objekten der Modellinstanz zu verstehen, es handelt sich dabei nicht um ein EXPRESS-G-Diagramm eines Schemas. Vielmehr soll verdeutlicht werden, wie die Struktur des Hilfsschemas und die Struktur der IFC-Modellinstanz zusammenhängen. Die in eckigen Klammern angegebenen Nummern geben an, wie oft eine Klasse im Beispielmodell instanziiert wurde.

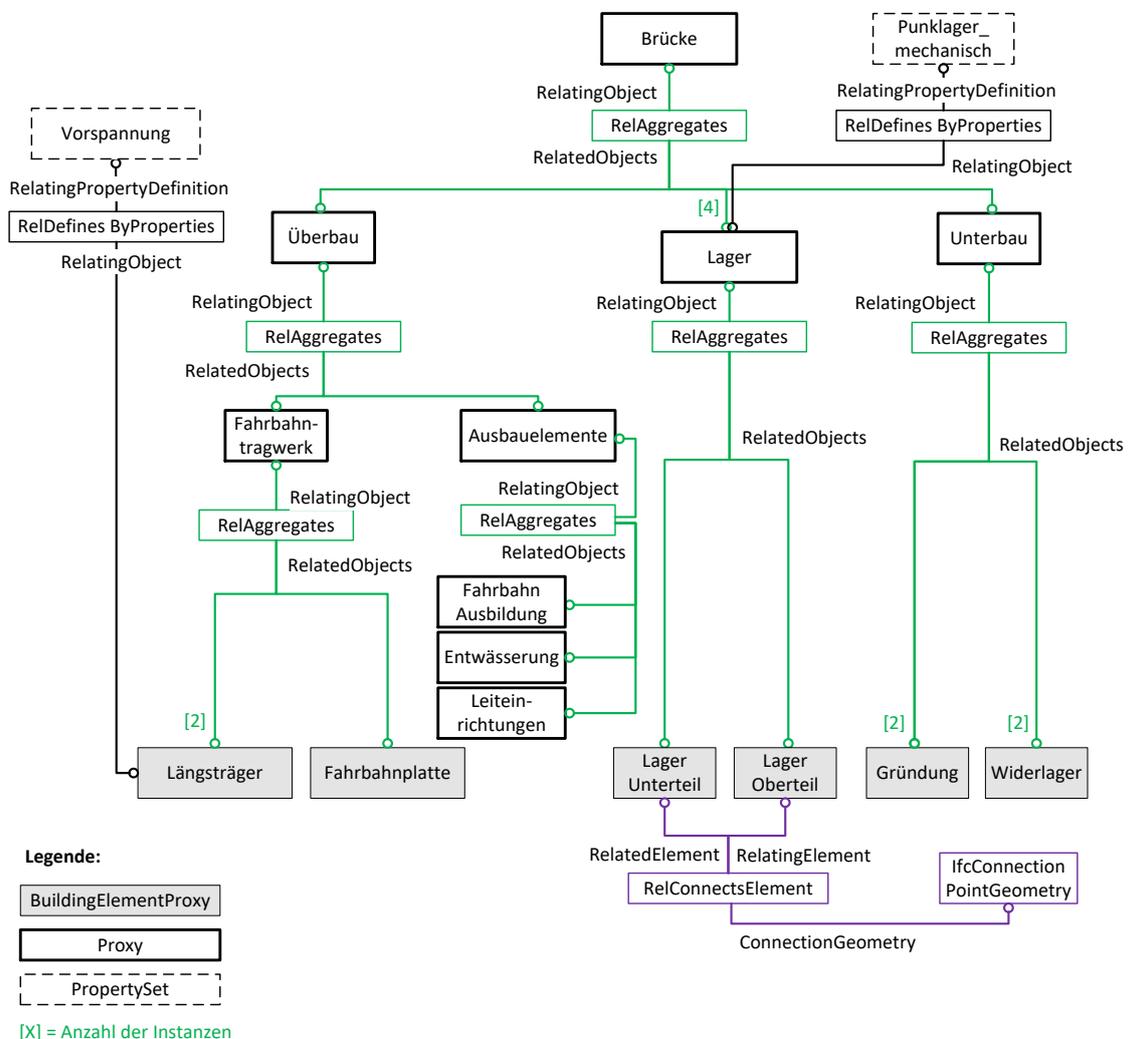


Abbildung 65: Semantische Struktur der IFC-Instanz der Balkenbrücke

## Hierarchische Bauwerksstruktur; FZK-Viewer (Screenshot)

Abbildung 66 zeigt einen Screenshot der *Browser Toolbar* des *KFZ Viewers*. Dort ist die hierarchische Struktur der Bauwerkskomponenten in Form eines Baumdiagramms veranschaulicht.

Das Diagramm zeigt, dass die händisch erweiterte IFC2x3-Datei der Modellinstanz strukturell mit der Struktur des Hilfsschemas übereinstimmt. Die entsprechenden Passagen des Dateicodes finden sich in Anlage 1 auf Seite 99.

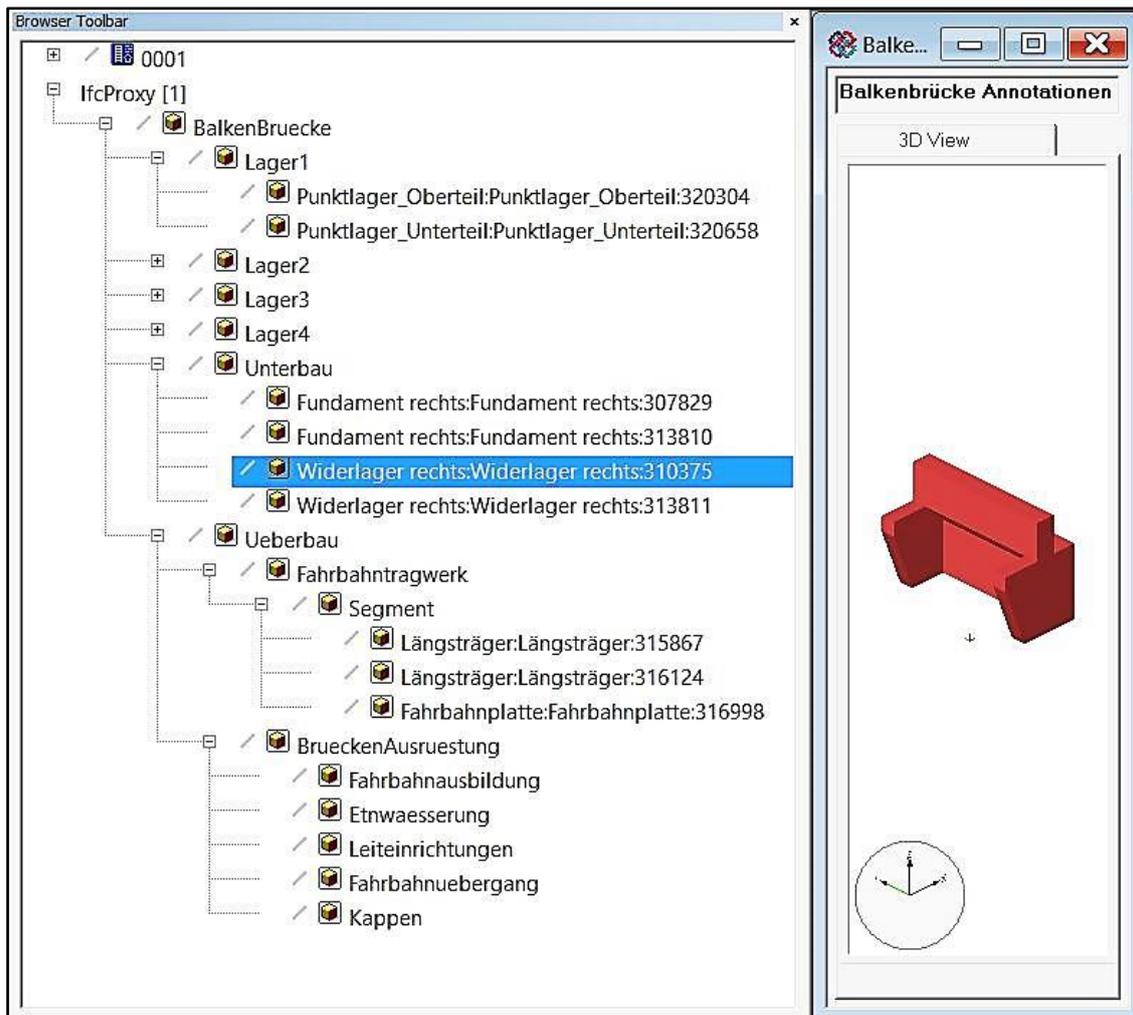


Abbildung 66: Hierarchische Struktur der Komponenten der Beispielinstantz

## Eigenschaften der Komponenten, FZK-Viewer (Screenshot)

Mit Hilfe der Funktion *Query* → *Model Information* → *PropertySets* lassen sich die annotierten Property-Sets sowie die geschachtelte Struktur der enthaltenen Eigenschaften ebenfalls als Baumdiagramm veranschaulichen.

Nachfolgende Abbildung zeigt den Screenshot einer solchen Abfrage. Sie zeigt die zu den als Proxy-Elemente instanziierten Brückenkomponenten gehörenden Eigenschafts-Sets; die Eigenschaften der Punktlager sind dabei in der rechten Spalte gesondert aufgeführt. Die entsprechenden Passagen des Quellcodes finden sich auf den Seiten 99 bis 101.

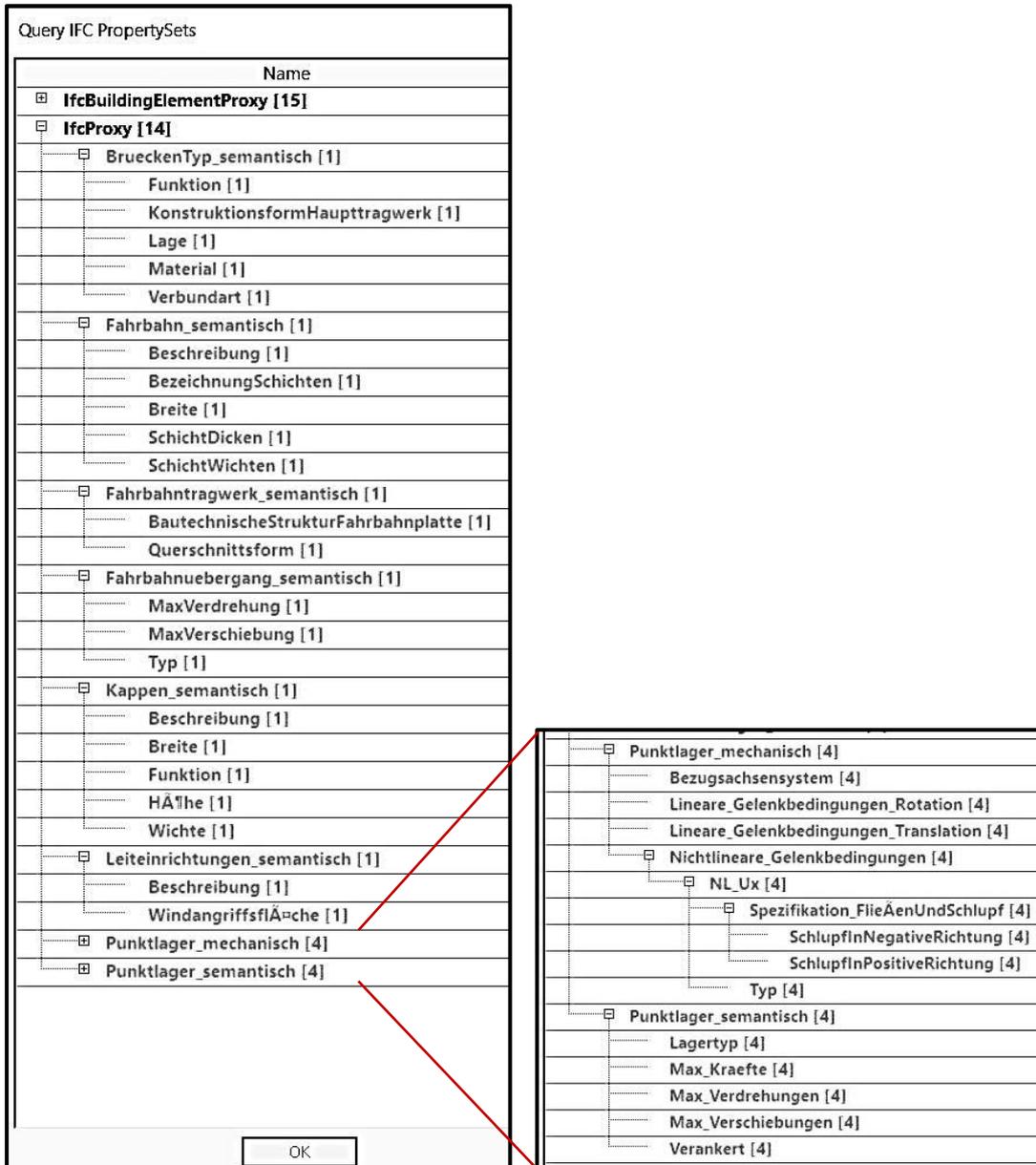


Abbildung 67: Eigenschaften der Komponenten der Beispielinstantz

## Spanngliedverlauf in den Längsträgern der Beispielbrücke

Da die Spannglieder der Beispielbrücke keine geometrische Repräsentation besitzen, wird sie im Folgenden kurz beschrieben.

Für die Längsträger der Beispielbrücke wurde eine Vorspannung bestehend aus je einem Spannglied pro Träger gewählt; der Verlauf des Spanngliedes in der x-z-Ebene ist in Abbildung 68 dargestellt. Er besteht aus drei Abschnitten: Die beiden äußeren Abschnitte verlaufen parabolisch, ihre auf die x-Achse projizierte Länge beträgt jeweils 43,0 Meter. Der mittlere Bereich verläuft horizontal, er besitzt eine Länge von 20,0 Metern.

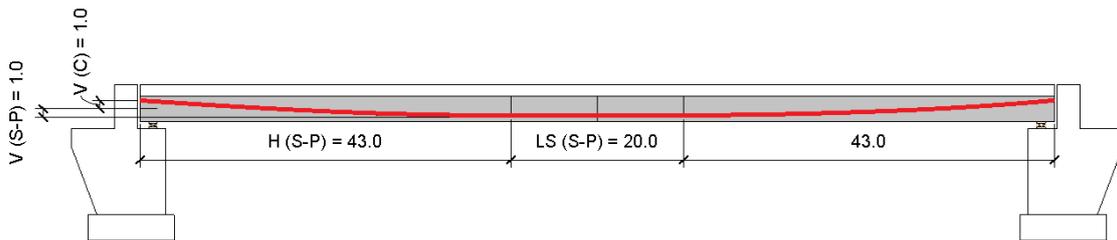


Abbildung 68: Verlauf des Spannglieds im Träger der Beispielbrücke

Dieser Verlauf lässt sich über ein Segment vom Typ 2 darstellen (siehe Abbildung 69). Die über die Randpunkte C beschriebenen geraden Abschnitte besitzen dabei die Länge  $l_s = 0,0$  Meter. Beide Randpunkte besitzen dieselben Parameter, ihre Position bezüglich der x-Achse ist in dem übergeordneten Complex-Property „Segmentgeometrie“ beschrieben. Als Referenzpunkt zur Beschreibung der Lage in z-Richtung wurde die Schwerachse des Trägers in z-Richtung gewählt, der Abstand der Randpunkte C zu dieser Achse beträgt 1,0 Meter.

Über den Zwischenpunkt S-P werden dessen Position sowie die Länge des geraden mittleren Abschnittes beschrieben. Die Position des Punktes in z-Richtung bezieht sich ebenfalls auf die Schwerachse des Trägers, der Abstand dazu beträgt 1,0 Meter. Als Referenzpunkt für die horizontale Position des Punktes wurde der Segmentanfang gewählt, der entsprechende Abstand beträgt 43,0 Meter. Die Länge des mittleren geraden Abschnittes hat die Länge  $l_s = 20,0$  Meter. Alle Abstände wurden als absoluter Wert und in Metern beschrieben und nicht, wie alternativ möglich, in Prozent der Gesamtlänge des Segments.

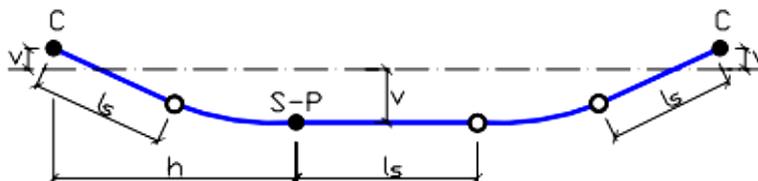


Abbildung 69: Segment-Typ 2

Quelle: (Dlupal Software GmbH, 2016)

## Struktur des Property-Sets Vorspannung, dargestellt mittels *FZK-Viewer*

Nachfolgende Abbildung zeigt die Instanz des Property-Set *Vorspannung*. Die entsprechenden Passagen des Quellcodes finden sich in Anlage 1, Seite 100.

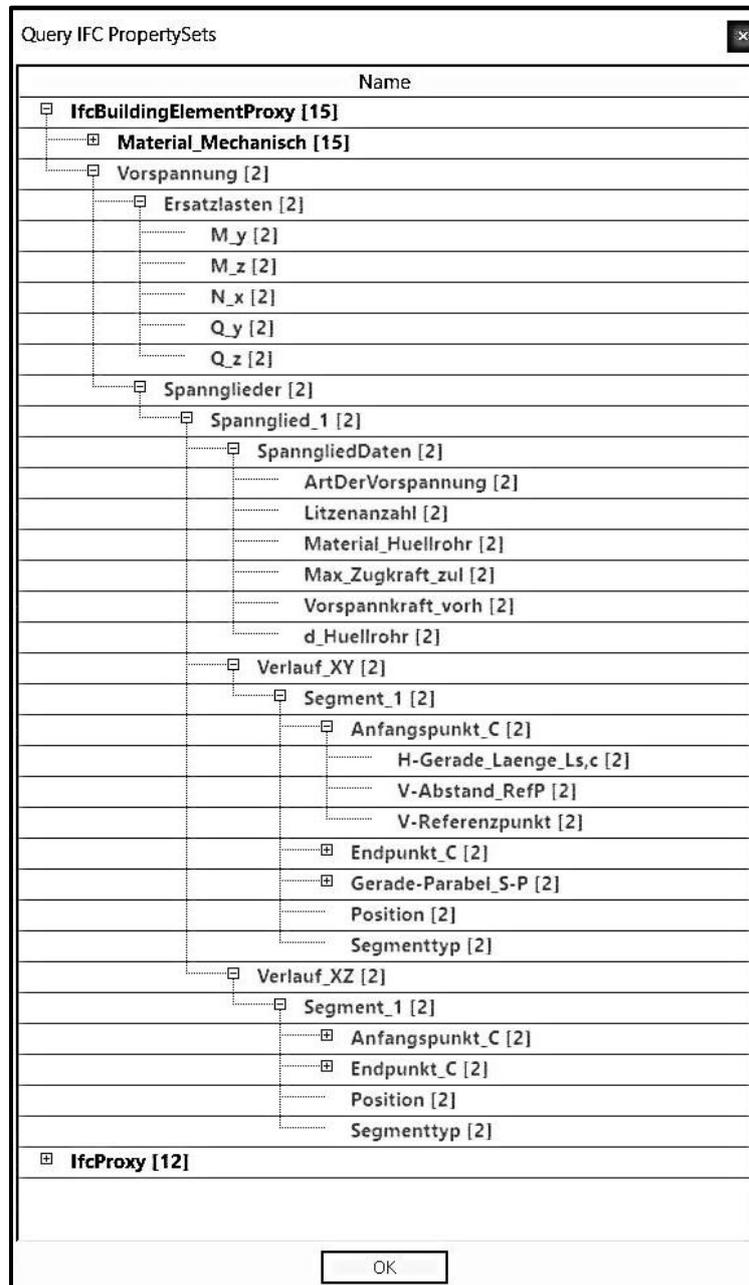


Abbildung 70: Struktur des Property-Sets Vorspannung

## Fehlermeldungen bei der Validierung

Zusätzlich zur Darstellung der Datei über das Programm *FZK-Viewer* wurden die aus *Revit* exportierten und händisch erweiterten *IFC2x3*-Dateien der Beispielbrücke mit Hilfe des *ifcCheckingTool\_Lite V 1.5* des *KIT* auf Validität überprüft. Die dabei aufgetretenen Fehlermeldungen werden im Folgenden kurz stichpunktartig aufgeführt.

- Bereits die unveränderte, aus *Revit* exportierte Originaldatei lieferte mehrere Fehlermeldungen.

Diese bezogen sich hauptsächlich darauf, dass *Revit* „*Representation Identifier*“ ausgab, die in *IFC Koordination View* nicht definiert sind. Auf diese Meldung wurde nicht weiter eingegangen.

- Die aus *Revit* exportierte *IFC*-Datei hatte den einzelnen Element-Proxys automatisch eine Zugehörigkeit zu einem Gebäudelevel zugewiesen.

Bei der händischen Zuweisung der Building-Element-Proxys zu den als Container fungierenden Proxys kam es zu einer Fehlermeldung, da ein Element nicht gleichzeitig einem Gebäudelevel und einem Proxy zugeteilt werden darf. Die Zuweisung zum Gebäudelevel sowie das Gebäudelevel selbst und die mit ihm verbundenen Objekte mussten daher händisch entfernt werden.

- Laut #CV-2x3-142 der *Ifc Implementation Agreements* muss ein Projekt mindestens eine Instanz vom Typ *IfcBuilding* enthalten, die Modellinstanz der Brücke besteht jedoch ausschließlich aus Instanzen von *IfcProxy* und *IfcBuildingElementProxy*.

- *Revit* hat den Building-Element-Proxys automatisch eine Reihe von Property-Sets zugewiesen. Da sich das nicht ohne Weiteres in *Revit* selbst umgehen ließ, wurden diese Property-Sets sowie deren zugehörige Objekte nach dem Export händisch aus der *IFC*-Datei entfernt.

- Das Entfernen dieser Objekte, Beziehungen und Eigenschaften zog weitere Fehlermeldungen nach sich. Da sich diese „*Constraint-Violations*“ jedoch in keiner erkennbaren Weise auf die korrekte Darstellung der Modelldaten mittels *FZK-Viewer* bemerkbar machten, wurde darauf nicht weiter eingegangen.

## Anlage 3: EXPRESS-Spezifikationen der Hilfsschema-Klassen

### Abstrakte Oberklassen

<pre>ENTITY <b>Entität</b>; ABSTRACT SUPERTYPE OF (ONE OF (Eigenschaft, Objekt)); END ENTITY;</pre>	<pre>ENTITY <b>Element</b>; SUBTYPE OF (Objekt); ABSTRACT SUPERTYPE OF (ONE OF (*Alle Elementklassen*)); InstanziiertUeberIfcKlasse: IfcBuildingElementProxy; hatMaterial: Material; WHERE (Pseudocode) Wr2: (InstanziiertUeberIfcKlasse.ObjectType = *NameOfSubtype*); END ENTITY;</pre>
<pre>ENTITY <b>Objekt</b>; SUBTYPE OF (Entität); ABSTRACT SUPERTYPE OF (ONE OF (Element, Komponente)); END ENTITY;</pre>	<pre>ENTITY <b>Komponente</b>; SUBTYPE OF (Objekt); ABSTRACT SUPERTYPE OF (ONE OF (*Alle Komponenteklassen*)); InstanziiertUeberIfcKlasse: IfcProxy; WHERE (Pseudocode) Wr3: (InstanziiertUeberIfcKlasse.ObjectType = *NameOfSubtype*); END ENTITY;</pre>
<pre>ENTITY <b>Eigenschaft</b>; SUBTYPE OF Entität; ABSTRACT SUPERTYPE OF (ONE OF (*Alle Eigenschaftsklassen*)); ReferenziertIfcKlasse: IfcPropertySet; WHERE (Pseudocode) wr1: (ReferenziertesIfcKlasse.Name = *NameOfSubtype*); END ENTITY;</pre>	

Abbildung 71: EXPRESS-Spezifikation der abstrakten Oberklassen

### Brückentypen und Überbau

<pre>ENTITY <b>Brücke</b>; SUBTYPE OF (Komponente); ABSTRACT SUPERTYPE OF (Balkenbrücke, Bogenbrücke, Schräggabelbrücke); BestehtAusLager: SET [0:?] OF Lager; BestehtAusÜberbau: Überbau; BestehtAusUnterbau: Unterbau; Kategorie: BrückenTyp; END ENTITY;</pre>	<pre>BestehtAusSchrägseiltragwerk: SET [1:?] OF Schrägseiltragwerk; END ENTITY;</pre>
<pre>ENTITY <b>Balkenbrücke</b>; SUBTYPE OF (Brücke); END ENTITY;</pre>	<pre>ENTITIIY <b>Bogenbrücke</b>; SUBTYPE OF (Brücke); BestehtAusBogentragwerk: SET [0:?] OF Bogentragwerk; END ENTITIIY,</pre>
<pre>ENTITY <b>Schräggabelbrücke</b>; SUBTYPE OF (Brücke);</pre>	<pre>ENTITY <b>Überbau</b>; SUBTYPE OF (Komponente); BestehtAusFahrbahntragwerk: Fahrbahntragwerk; BestehtAusBrückenausrüstung: OPTIONAL Brückenausrüstung; INVERSE istTeilVonBrücke: Brücke FOR BestehtAus; END ENTITIIY;</pre>

Abbildung 72: EXPRESS-Spezifikation der Brückentypen und des Überbaus

## Brückenausrüstung

```
ENTITY Brückenausrüstung;  
SUBTYPE OF (Komponente);  
EnthältBeleuchtung: OPTIONAL Beleuchtung;  
EnthältLeiteinrichtung: Leiteinrichtung;  
EnthältSchallschutzanlage: OPTIONAL Schallschutzanlage;  
EnthältEntwässerung: Entwässerung;  
EnthältVersorgungsleitungen: OPTIONAL;  
Versorgungsleitungen;  
EnthältKappe: Kappen;  
EnthältFahrbahnausbildung: SET [0:?] OF  
Fahrbahnausbildung;  
INVERSE  
IstTeilVonÜberbau: Überbau  
FOR BestehtAusBrückenausrüstung;  
END ENTITY;  
  
ENTITY Ausbaukomponente;  
SUBTYPE OF (Komponente);  
ABSTRACT SUPERTYPE OF (Fahrbahnausbildung,  
Leiteinrichtungen, Entwässerung, Kappen,  
Versorgungsleitungen, Beleuchtung, Schallschutzanlage,  
Fahrbahnübergang);  
BestehtAus: SET [0:?] OF Ausbauelement;  
END ENTITY;  
  
ENTITY Ausbauelement;  
SUBTYPE OF (Element);  
INVERSE  
istTeilVon: Ausbaukomponente  
FOR BestehtAus;  
END ENTITY;  
  
ENTITY Fahrbahnausbildung;  
SUBTYPE OF (Ausbaukomponente);  
Semantik: Fahrbahnausbildung_semantisch;  
INVERSE  
IstTeilVon: Brückenausrüstung  
FOR BestehtAusFahrbahnausbildung;  
END ENTITY;  
  
/*Alle weiteren Unterklassen von Ausbaukomponente  
analog zu Fahrbahnausbildung*/
```

Abbildung 73: EXPRESS-Spezifikation der Brückenausrüstung

## Fahrbahntragwerk

<p><b>ENTITY DeckElement;</b> SUBTYPE OF (Element); ABSTRACT SUPERTYPE OF (ONE OF (Verstrebung, Platte, LängsBalken, Querbalken)); HatVorspannung: OPTIONAL Vorspannung; BestehtAusSpanngliedern: SET [0:?] OF Spannglied; END ENTITY;</p> <p><b>ENTITY Verstrebung;</b> SUBTYPE OF (DeckElement); INVERSE IstTeilVonSegment: Segment FOR bestehtAusVerstrebung; END ENTITY;</p> <p><b>ENTITY Platte;</b> SUBTYPE OF (DeckElement); INVERSE IstTeilVonSegment: Segment FOR bestehtAusPlatte; END ENTITY;</p> <p><b>ENTITY LängsBalken;</b> SUBTYPE OF (DeckElement); INVERSE IstTeilVonSegment: Segment FOR bestehtAusLängsBalken; END ENTITY;</p> <p><b>ENTITY Querbalken;</b> SUBTYPE OF (DeckElement); INVERSE IstTeilVonSegment: Segment FOR bestehtAusQuerbalken; END ENTITY;</p>	<p><b>ENTITY Segment;</b> SUBTYPE OF (Komponente); BestehtAusVerstrebung: SET [0:?] OF Verstrebung; BestehtAusPlatte: SET [1:?] OF Platte; BestehtAusLängsBalken: SET [0:?] OF LängsBalken; BestehtAusQuerbalken: SET [0:?] OF Querbalken; INVERSE IstTeilVonFahrbahntragwerk: Fahrbahntragwerk FOR bestehtAusSegment; END ENTITY;</p> <p><b>ENTITY Fahrbahntragwerk;</b> SUBTYPE OF (Komponente); BestehtAus: SET [1:?] OF Segment; BestehtAusSpanngliedern: SET [0:?] OF Spannglied; VerbundenMitWiderlager: SET [0:2] OF Widerlager; VerbundenMitPfeiler: SET [0:?] OF Pfeiler; HatVorspannung: OPTIONAL Vorspannung; Semantik: Fahrbahntragwerk_semantisch; INVERSE VerbundenMitLager: SET [0:?] OF Lager FOR Lagert; END ENTITY;</p> <p><b>ENTITY Spannglied;</b> SUBTYPE OF (Element); IstGeometrischDefiniertÜber: Vorspannung; INVERSE IstTeilVon: SET [0:1] OF Deckelement FOR bestehtAusSpannglied; IstTeilVon: SET [0:1] OF Fahrbahntragwerk FOR bestehtAusSpannglied; END ENTITY;</p>
--	---

Abbildung 74: EXPRESS-Spezifikation des Fahrbahntragwerks

## Schräggabeltragwerk

<p><b>ENTITY Schrägseiltragwerk;</b> SUBTYPE OF (Komponente); BestehtAusSchräggabel: SET [1:?] OF Schräggabel; BestehtAusPylon: Pylon; Semantik: Schrägseiltragwerk_semantisch; END ENTITY;</p> <p><b>ENTITY Seil;</b> SUBTYPE OF (Element); StütztFahrbahntragwerk: Fahrbahntragwerk; Mechanik: Seil_mechanisch; Semantik: Seil_semantisch; INVERSE IstTeilVon: SET [0:1] OF Schrägseiltragwerk</p>	<p>FOR bestehtAusSchrägSeil; IstTeilVon: SET [0:1] OF Bogentragwerk FOR bestehtAusSeil; END ENTITY;</p> <p><b>ENTITY Pylon;</b> SUBTYPE OF (Element); GründetAufGründung: Gründung; TrägtSeil: SET [1:?] OF Seil; INVERSE IstAuflagerbankFür: SET [0:?] OF Lager FOR LagertAuf; IstTeilVon: Schrägseiltragwerk FOR bestehtAusPylon; END ENTITY;</p>
--	---

Abbildung 75: EXPRESS-Spezifikation des Schräggabeltragwerks

## Bogentragwerk

```
ENTITY Bogentragwerk;  
SUBTYPE OF (Komponente);  
BestehtAusBogen: Bogen;  
BestehtAusHänger/Ständer: SET [0:?] OF  
Hänger/Stütze;  
BestehtAusSeil: SET [0:?] OF Seil;  
INVERSE  
IstTeilVonBogenbrücke: Bogenbrücke  
FOR BestehtAusBogentragwerk;  
END ENTITY;  
  
ENTITY Bogen;  
SUBTYPE OF (Komponente);  
ABSTRACT SUPERTYPE OF (ONE OF  
(SteiferBogen, Stabbogen));  
FußAufFahrbahntragwerk: OPTIONAL  
Fahrbahntragwerk;  
FußAufKämpferstütze: OPTIONAL  
Kämpferstütze;  
INVERSE  
IstTeilVonBogentragwerk: Bogentragwerk  
FOR BestehtAusBogen;  
END ENTITY;  
  
ENTITY SteiferBogen;  
SUBTYPE OF (Bogen);  
BestehtAusSegment: BogenSegment;  
TrägtHänger/Ständer: SET [0:?] OF  
Hänger/Ständer;  
TrägtSeil: SET [0:?] OF Seil;  
StütztFahrbahntragwerk: OPTIONAL  
Fahrbahntragwerk;  
INVERSE  
IstAuflagerbankFürLager: SET [0:?] OF Lager  
FOR LiegtAuf;  
FußAufLager: SET [0:1] OF Lager  
FOR Lagert;  
END ENTITY;  
  
ENTITY Stabbogen;  
SUBTYPE OF (Bogen);  
BestehtAusSegmenten: SET [3:?] OF BogenSegment;  
TrägtHänger/Stütze: SET [2:?] OF Hänger/Ständer;  
END ENTITY;  
  
ENTITY Hänger/Stütze;  
SUBTYPE OF (Element);  
ABSTRACT SUPERTYPE OF (ONE OF (Hänger, Stütze));  
StütztFahrbahntragwerk: Fahrbahntragwerk;  
Mechanik: Stabeigenschaften_mechanisch;  
INVERSE  
GetragenVonStabbogen: SET [0:1] OF Stabbogen  
FOR TrägtHänger/Ständer;  
GetragenVonSteiferBogen: SET[0:?] OF SteiferBogen  
FOR TrägtHänger/Ständer;  
END ENTITY;  
  
ENTITY Hänger;  
SUBTYPE OF (Hänger/Stütze);  
END ENTITY;  
  
ENTITY Stütze;  
SUBTYPE OF (Hänger/Stütze);  
END ENTITY;  
  
ENTITY BogenSegment;  
SUBTYPE OF (Element);  
Mechanik: OPTIONAL Stabbogen_mechanisch;  
INVERSE  
IstTeilVonStabbogen: Stabbogen  
FOR BestehtAusSegmenten;  
BildetSteifenBogen: SteiferBogen FOR BestehtAus;  
END ENTITY;
```

Abbildung 76: EXPRESS-Spezifikation des Bogentragwerks

## Unterbau

<p>ENTITY <b>Unterbau</b>;  SUBTYPE OF (Komponente);  BestehtAusKämpferstütze: SET [0:?] OF Kämpferstütze;  BestehtAusWiderlager: SET [2:?] OF Widerlager;  BestehtAusGründung: SET [2:?] OF Gründung;  BestehtAusPfeiler: SET [0:?] OF Pfeiler;  INVERSE  IstTeilVon: Brücke FOR BestehtAusUnterbau;  END ENTITY;</p> <p>ENTITY <b>Unterbauelement</b>;  SUBTYPE OF (Element);  ABSTRACT SUPERTYPE OF (ONE OF  (Kämpferstütze, Widerlager, Gründung, Pfeiler));  INVERSE  IstAuflagerbankFür: SET [0:?] Lager FOR LiegtAuf;  END ENTITY;</p> <p>ENTITY <b>Kämpferstütze</b>;  SUBTYPE OF (Unterbauelement);  INVERSE  IstTeilVon: Unterbau FOR BestehtAusKämpferstütze;  StütztBogen: SET [0:?] OF Bogen FOR FußtAuf;  END ENTITY;</p> <p>ENTITY <b>Widerlager</b>;  SUBTYPE OF (Unterbauelement);</p>	<p>GründetAuf: Gründung;  INVERSE  IstTeilVon: Unterbau FOR BestehtAusWiderlager;  IstAuflagerbankFür: Fahrbahntragwerk  FOR RuhtAuf;  END ENTITY;</p> <p>ENTITY <b>Gründung</b>;  SUBTYPE OF (Unterbauelement);  Semantik: Gründung_semantisch;  INVERSE  IstTeilVon: Unterbau FOR BestehtAusGründung;  GründungVonWiderlager: SET [0:1] OF Widerlager  FOR GründetAuf;  GründungVonPfeiler: SET [0:?] OF Pfeiler  FOR GründetAuf;  END ENTITY;</p> <p>ENTITY <b>Pfeiler</b>;  SUBTYPE OF (Unterbauelement);  GründetAuf: OPTIONAL Gründung;  Mechanik: OPTIONAL Stabeigenschaften_mechanisch;  Semantik: OPTIONAL Pfeiler_semantisch;  INVERSE  StütztFahrbahntragwerk: SET [0:1] OF Fahrbahntragwerk  FOR RuhtAuf;  IstTeilVon: Unterbau FOR BestehtAusPfeiler;  END ENTITY;</p>
---	---

Abbildung 77: EXPRESS-Spezifikation des Unterbaus

## Brückenlager

<p>ENTITY <b>Lager</b>;  SUBTYPE OF (Komponente);  ABSTRACT SUPERTYPE OF (ONE OF  (Punktlager, Linienlager);  LiegtAuf: LiegtAuf_Select;  Lagert: Lagert_Select;  BestehtAusUnterteil: Lagerunterteil;  BestehtAusOberteil: Lageroberteil;  SemantEigensch: Lager_semantisch;  END ENTITY;</p> <p>TYPE <b>LiegtAuf_Select</b> = SELECT  (Unterbauelement, Pylon, Bogen);  END TYPE;</p> <p>TYPE <b>LagertSelect</b> = SELECT  (Fahrbahntragwerk, Bogen, Pfeiler);  END TYPE;</p> <p>ENTITY <b>Punktlager</b>;  SUBTYPE OF (Lager);</p>	<p>Mechanik: Punktlager_mechanisch  END ENTITY;</p> <p>ENTITY <b>Linienlager</b>;  SUBTYPE OF (Lager);  Mechanik: Linienlager_mechanisch;  END ENTITY;</p> <p>ENTITY <b>Lagerunterteil</b>;  SUBTYPE OF (Element);  BeweglichVerbundenMit: Lageroberteil;  INVERSE  IstTeilVon: Lager FOR BestehtAus;  END ENTITY;</p> <p>ENTITY <b>Lageroberteil</b>;  SUBTYPE OF (Element);  INVERSE  BeweglichVerbundenMit: Lagerunterteil  FOR BeweglichVerbundenMit;  IstTeilVon: Lager FOR BestehtAus;  END ENTITY;</p>
--	---

Abbildung 78: EXPRESS-Spezifikation der Brückenlager

## Anlage 4: Punkttypen S-P und P-P (Ergänzung zu Kapitel 6.5.1)

### Zwischenpunkt Gerade-Parabel

Das Complex-Property, das Verbindungspunkte von geraden mit parabolischen Abschnitten beschreibt, enthält zusätzlich zur vertikalen Position des Punktes seine Lage in lokaler x-Richtung – beschrieben entweder relativ zum Segmentanfang – zur Segmentmitte oder zum Segmentende, und die Länge des angrenzenden geraden Abschnitts. Beide Werte können entweder als absoluter Wert oder relativ, bezogen auf die Gesamtlänge des Segments, angegeben werden.

Tabelle 45: Zwischenpunkt Gerade-Parabel; Typ S-P

IfcComplexProperty „ZwischenPunkt_Gerad-Parabel_S-P“		
Name	Typ	Value
V-Referenzpunkt	Enumerated	„Maximum_Z+“, „Randdurchdringung_Ze+“, „Bezugsachse_v“, „Schwerpunkt_Zcg“, „Randdurchdringung_Ze-“, „Minimum_Z-“;
V-Abstand_RefP	SingleValue	Real
H-Referenzpunkt	Enumerated	„SegmentAnfang“, „SegmentMitte“, „SegmentEnde“;
H-Länge_relativ	SingleValue	Boolean
H-Abstand_h <sub>S,P</sub>	SingleValue	Real
H-Gerade_Länge_L <sub>S,S-P</sub>	SingleValue	Real

### Zwischenpunkt Parabel-Parabel

Über Punkte, die zwei parabelförmige Abschnitte miteinander verbinden, wird der minimale Radius der angrenzenden Parabeln sowie die (absolute) Länge der angrenzenden geraden Abschnitte links und rechts des Punktes definiert. Während der minimale Radius für beide angrenzenden Parabeln identisch sein muss, können die geraden Bereiche links und rechts unterschiedlich lang sein.

Tabelle 46: Zwischenpunkt Parabel-Parabel; Typ P-P

IfcComplexProperty „ZwischenPunkt_Parabel-Parabel_P-P“		
Name	Typ	Value
V-Referenzpunkt	Enumerated	„Maximum_Z+“, „Randdurchdringung_Ze+“, „Bezugsachse_v“, „Schwerpunkt_Zcg“, „Randdurchdringung_Ze-“, „Minimum_Z-“;
V-Abstand_RefP	SingleValue	Real
H-Minimaler_Radius	SingleValue	Real
H-Gerade_Länge_L <sub>Links,P-P</sub>	SingleValue	Real
H-Gerade_Länge_L <sub>Rechts,P-P</sub>	SingleValue	Real

## Anlage 5: Segment-Typen in *RFEM* (Ergänzung zu Kapitel 6.5.1)

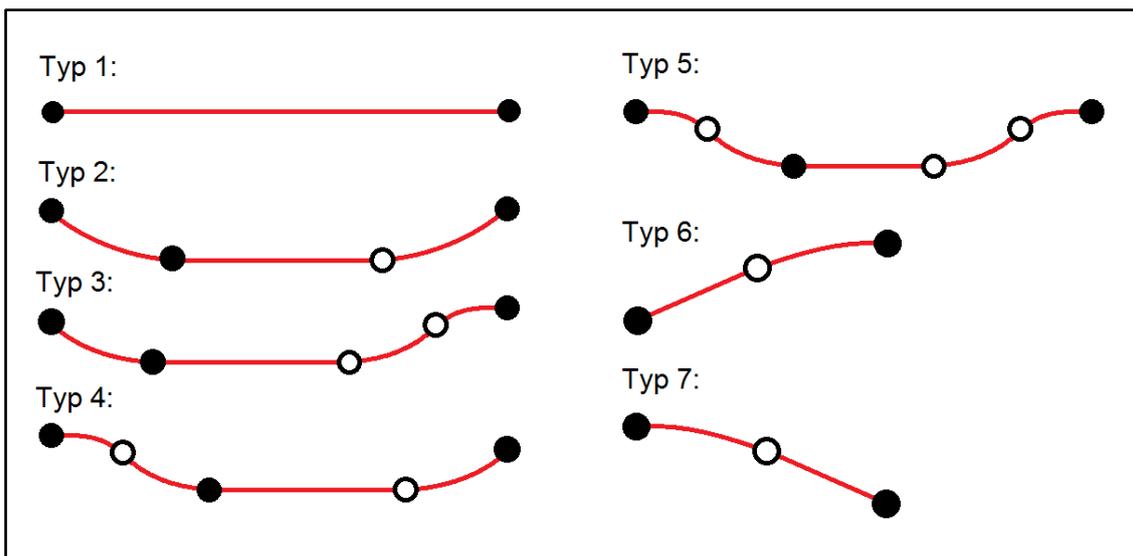


Abbildung 79: Segmenttypen in *RFEM*

Quelle: in Anlehnung an (Dlubal Software GmbH, 2016)

## Anlage 6: Eigenschaften von *IfcTendon*

Tabelle 47: *IfcTendon*

<b>IfcTendon</b>		
<b>Attribute</b>	<b>Type</b>	<b>Value</b>
PredefinedType	IfcTendonTypeEnum	STRAND, WIRE, BAR, COATED, USERDEFINED, NOTDEFINED;
NominalDiameter	IfcPositiveLengthMeasure	IfcLengthMeasure
CrossSectionArea	IfcAreaMeasure	REAL
TensionForce [opt]	IfcForceMeasure	REAL
TensionForce [opt]	IfcPressureMeasure	REAL
FrictionCoefficient [opt]	IfcNormalisedRatioMeasure	IfcRatioMeasure
AnchorageSlip [opt]	IfcPositiveLengthMeasure	IfcLengthMeasure
MinCurvatureRadius [opt]	IfcPositiveLengthMeasure	IfcLengthMeasure

## Anlage 7: Spezielle Eigenschaften von Beton und Stahl in *IfcStructural*

Tabelle 48: *IfcMechanicalConcreteMaterialProperties*

<b>IfcMechanicalConcreteMaterialProperties</b>	
<b>Attribut</b>	<b>Value</b>
CompressiveStrength	Real
MaxAggregateSize	IfcLengthMeasure
AdmixturesDescription	String
Workability	String
ProtectivePoreRatio	IfcRatioMeasure
WaterImpermeability	String

Tabelle 49: *IfcMechanicalSteelMaterialProperties*

<b>IfcMechanicalSteelMaterialProperties</b>	
<b>Attribut</b>	<b>Value</b>
YieldStress	Real
UltimateStress	IfcLengthMeasure
UltimateStrain	String
HardeningModule	String
ProportionalStress	IfcRatioMeasure
PlasticStrain	String
<b>IfcRelaxation</b>	
<b>Attribut</b>	<b>Value</b>
RelaxationValue	IfcRatioMeasure
InitialStress	IfcRatioMeasure