Technische Universität München

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Computergestützte Modellierung und Simulation

Erstellung von IfcBridge-Modellen mithilfe visueller Programmierung

Bachelorthesis

für den Bachelor of Science Studiengang Bauingenieurwesen

Autor: Korbinian Aicher

Matrikelnummer: [redacted]

1. Betreuer: Prof. Dr.-Ing. André Borrmann

2. Betreuer: M.Sc. Sebastian Esser

Ausgabedatum: 11. Mai 2019

Abgabedatum: 27. August 2019

Abstract

In the ever-increasing digitalization of the construction industry, methods of Building Information Modeling (BIM) and neutral data exchange play an essential role. There are already numerous BIM software applications and corresponding standardizations for the above-ground construction sector. In contrast to above-ground construction projects, where BIM methods are already a standard, the utilization of BIM in infrastructure planning is currently about to beginn. However, one can already observe a growing interest in BIM in the infrastructure sector. The application-neutral data exchange format Industry Foundation Classes (IFC), developed by buildingSMART, has introduced a concept whereby data from BIM-models can be transferred between proprietary software applications. The IFC exchange format, which was originally optimized for above-ground construction projects, introduced extensions that extend the area of application to include the infrastructure sector. With the release of the ifcBridge extension, bridge models can be exchanged via a defined standard in the course of a construction project.

The following thesis focuses on an export tool that creates IFC files of BIM bridge models using visual programming. For this purpose, bridge models are first constructed in a pre-planning program and afterwards transferred to an established BIM software application. Potential problems and exportability constraints are explored during the bridge modeling process. Moreover, the above-mentioned BIM models serve as a basis to test the implemented export tool.

In order to create ifcBridge models using visual programming, this thesis developed a .Net-based application that generates an ifcBridge file and transfers components of an example bridge to the IFC format. The software applications used to create the bridge BIM-model and the visual programming language are part of the portfolio of the company Autodesk.

Zusammenfassung

In der immer weiter fortschreitenden Digitalisierung des Bauwesens besitzen Methoden des Building Information Modeling (BIM) und neutrale Datenaustauschformate eine zentrale Rolle. Für den Hochbausektor existieren bereits zahlreiche BIM-Softwareanwendungen und Standardisierungen. Obwohl im Gegensatz zu Hochbauprojekten die Anwendung von BIM-Methoden für Infrastrukturplanung erst beginnen, ist ein steigende Interesse für BIM in diesem Bereich zu erkennen. Das von buildingSMART entwickelte herstellerneutrale Datenaustauschformat Industry Foundation Classes (IFC) hat ein Konzept geschaffen, wodurch Daten von BIM-Modellen zwischen proprietären Anwendungen transferiert werden können. Das IFC Austauschformat, das ursprünglich für Hochbauprojekte optimiert ist, ergänzt durch Erweiterungen des Schemas den Einsatzbereich auf den Infrastruktursektor. Mit der Einführung von IfcBridge können Brückenmodelle über einen definierten Standard im Laufe eines Bauprojekts ausgetauscht werden.

Die vorliegende Arbeit beschäftigt sich mit einer Exportmöglichkeit, die BIM-Brückenmodelle mithilfe visueller Programmierung als IFC-Datei erstellt. Hierfür werden zunächst Brückenmodelle in einem Vorplanungsprogramm konstruiert und in eine etablierte BIM-Softwareanwendung übertragen. Während des Erstellungsprozesses werden Problemantiken und Einschränkungen für die Exportmöglichkeit untersucht. Die BIM-Modelle dienen außerdem als Grundlage, um die erfolgreiche Nutzung der implementierten Exportmöglichkeit zu testen.

Für die Erstellung von IfcBridge-Modellen durch eine visuelle Programmiersprache wurde im Rahmen der Arbeit eine .Net-basierte Applikation entwickelt, die eine IfcBridge-Datei generiert und Bauteilkomponenten der Beispielbrücken in das IFC-Schema übergibt. Jegliche genutzte Software zur Erstellung der Brücke und die genutzte visuelle Programmiersprache stammt aus dem Portfolio der Firma Autodesk.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AEC-Sektor	Architecture, Engineering and Construction Sektor
API	Application Programm Interface
ARV	Alignment-based Bridge Reference View
BIM	Building Information Modelling
BMVI	Bundesministerium für Verkehr und digitale Infrastruktur
BRep	Boundary Representation
CSG	Constructive Solid Geometry
C#	C-Sharp
DTV	Bridge Design Transfer View
gKRS	Geodätisches Koordinaten Referenzsystem
IFC	Industry Foundation Classes
MVS	Microsoft Visual Studio
PKS	Projiziertes Koordinatensystem
RV	Bridge Reference View
xBIM	eXtensible Building Information Modeling
ZTN	Zero Touch Nodes

1 Motivation und Aufbau der Arbeit

1.1 Motivation und Idee

Das Themenfeld Digitalisierung hat in den vergangenen Jahren in der deutschen Bauindustrie immer mehr an Relevanz gewonnen. Besonders Building Information Modeling (BIM) spielt hierbei eine zentrale Rolle, vor allem in der Entwicklung und im Anwendungsbereich.

Der Ursprung der BIM-Technologie liegt im Hochbau und wurde im Laufe der Entwicklung speziell für diesen Bereich optimiert. Durch die Erweiterung von Bauwerksmodellen, um nicht geometrische Parameter, wird das Erzeugen eines digitalen Zwillings von Bauwerken möglich, welcher ein hohes Maß an Detailtreue besitzt. Die Verwendung des BIM-Modells findet über den gesamten Bauwerkszyklus statt und wird im Zuge dessen nicht als Softwareanwendung verstanden, sondern als Gesamtkonzept, welches die Zusammenarbeit unterschiedlicher Fachdisziplinen vereinfacht.

Durch den vom Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) entwickelten Stufenplan „Planen und Bauen“ und dem Masterplan „Bauen 4.0“ wird die Bedeutung von BIM-Technologien für die fortschreitende Digitalisierung dieses Industriezweiges hervorgehoben. Besonders die großflächige Integration von BIM für infrastrukturelle Bauvorhaben ist von großer Bedeutung. Der verallgemeinernde Begriff Infrastruktur wird im Rahmen dieser Arbeit für das Teilgebiet Verkehrsinfrastruktur verwendet.

Es existieren zahlreiche Anwendung verschiedener Softwareanbieter um BIM dafür zu nutzen Bauwerke, anhand eines digitalen Modells mit semantischen und geometrischen Informationen, über den gesamten Lebenszyklus darzustellen. Für infrastrukturspezifische BIM-Modellierung existieren bereits Anwendungen und integrierte Erweiterungen etablierter BIM-Software, die allerdings nicht dem Entwicklungsstand des Hochbausektors entsprechen.

Bei Pilotprojekten, wie beispielweise der Filstalbrücke in Baden Württemberg, wird die Nutzung von BIM in der Planungs- und Ausführungsphase untersucht. Die gewonnen

Erfahrungen werden benötigt, um ein besseres Verständnis der Strukturen, Prozessabläufe und Interaktion handelnder Personen bei der Verwendung von BIM zu erhalten (Bundesministerium für Verkehr und digitale Infrastruktur, 2017).

Besonders in der Verkehrsinfrastruktur mit starken dynamischen Belastungen durch die Einwirkung verschiedener Verkehrsmittel und unvorhersehbaren Ereignissen ist eine Standardisierung von BIM in diesem Sektor ein wichtiger Schritt. Hierbei soll nicht die fachliche Expertise der jeweiligen Fachdisziplinen ersetzt werden, sondern ein unterstützendes Werkzeug für eine Verbesserung des Ablaufs und Sicherheit geschaffen werden.

1.2 Zielsetzung der Arbeit

Das von der Non-Profit-Organisation buildingSMART entwickelte Datenmodell Industry Foundation Classes (IFC) ist ein offener Standard im Bauwesen und ermöglicht einen Austausch von Informationen zwischen verschiedenen proprietären Softwareanwendungen (buildingSMART e.V., 2019d).

Mit der Einführung der Version IFC4X2 ist das herstellerneutrale Datenaustauschformat IFC um Entitäten zur Beschreibung von Brückenbauwerken erweitert worden. Diese Erweiterung ermöglicht es erste Bauwerke aus dem Infrastrukturbereich, durch die Vorgabe einer festen allgemeinen Datenstruktur, in das Datenformat zu übermitteln.

Zielsetzung der folgenden Arbeit ist die Konzipierung und Entwicklung einer Exportmöglichkeit für Brückenmodelle, die in einer existierenden Software zur Brückenmodellierung erstellt werden. Mittels einer prototypischen Implementierung eines IfcBridge-Exports wird mit Nutzung visueller Programmierung ein Brückenmodell in das IFC4X2 Format übersetzt.

Im Vorfeld der Implementierung der Exportmöglichkeit wird zunächst der Entwicklungsprozess eines linearen Bauwerks in der Vorplanungsphase untersucht. Hierzu wird eine Beispielbrücke mit den bereits existierenden BIM-Softwareanwendungen Autodesk InfraWorks (IW) 2020 und Autodesk Revit 2020 erstellt, um den Ablauf auf Problemstellungen und Datenverluste, mit dem Fokus auf IFC, zu untersuchen. Die benötigten Funktionen des Exportmechanismus, um ein BIM-Modell in das IFC-Schema zu bringen, werden hierdurch ermittelt.

Für die Erstellung des konzeptionellen Modells wird Autodesk IW 2020 verwendet. Durch die proprietäre Schnittstelle bietet sich die BIM-Modellierungsanwendung Autodesk Revit 2020 zur weiteren Bearbeitung an. Anschließend werden in Dynamo, eine in Revit integrierte visuelle Programmiersprache, die bereits vorhanden Informationen des Modells durch semantische Parameter ergänzt und mithilfe der entwickelten Applikation IfcBridge Toolkit in das IFC Schema übersetzt.

1.3 Aufbau der Arbeit

Kapitel 2 umfasst eine detailliertere Ausführung zu BIM und dessen Bedeutung für den Infrastrukturbereich, um zu einem besseren Verständnis der Arbeit beizutragen. Ein besonderer Fokus liegt auf allgemeine Informationen zu BIM, Entwicklungsstand für Infrastrukturprojekte und Konzepte des BIM und IFC.

In Kapitel 3 wird auf das IfcBridge-Projekt als Erweiterung des IFC Schemas eingegangen. Es wird ein Überblick über den Aufbau und die Funktionsweise dargestellt. Besonders die Änderung der räumlichen Struktur für linienförmige Bauwerke wird detaillierter beschrieben, da es sich um eine signifikante Erweiterung des Schemas handelt.

Im darauffolgenden Kapitel 4 werden der Erstellungsprozess und die Aufbereitung einer Beispielbrücke näher beleuchtet und die dabei entstandenen Problematiken werden anschließend kritisch analysiert. Abschließend beinhaltet Kapitel 4 eine Untersuchung des Modells auf wichtige Informationen, die für den Export in das IFC Format essenziell sind.

Einen vertieften Überblick über die Funktionsweise und den Aufbau des Codes und die Codeblöcke, die eine Erstellung von IfcBridge Modellen ermöglichen, wird in Kapitel 5 gegeben.

In Kapitel 6 wird mit der Erstellung eines IfcBridge Modells abgeschlossen und das Ergebnis wird in einem IFC-Viewer betrachtet. Abschließend wird das Ergebnis der Bachelorarbeit in Kapitel 7 evaluiert und gibt einen Ausblick auf weitere Erweiterungsmöglichkeiten

2 Entwicklung von Building Information Modeling im Infrastrukturbereich

In diesem Kapitel werden BIM als Konzept und das neutrale Datenschema IFC definiert. Des Weiteren wird BIM und IFC im Infrastrukturbereich näher beleuchtet.

2.1 Begriffserklärung BIM

Die Bauindustrie mit einer jährlichen Produktionsleistung von 1,3 Billionen Euro ist für Volkswirtschaften ein wichtiger Wirtschaftssektor. Dennoch hat sich die jährliche Produktivitätsrate in den letzten 20 Jahren nur um 1% gesteigert (EUBIM Taskgroup, 2017). Diese vergleichsweise geringe Steigerung resultiert unter anderem aus einer verspäteten Digitalisierung des Bauwesens. Ein Kernproblem, warum die Digitalisierung der Baubranche nur langsam voranschreitet, liegt in der Übermittlung projektspezifischer Daten im Architecture, Engineering and Construction-Sektor (AEC-Sektor), die von einer Informationsweiterleitung durch Pläne und Textdokumentationen geprägt ist. Wichtige Daten, die bereits digital erstellt sind, gehen in diesem Prozess verloren. Auch eine Weiterverwendung der Daten für andere Projektphasen in einem Bauwerkslebenszyklus ist nicht direkt möglich (Borrmann et al., 2015). Der AEC-Sektor bedeutet die computergestützte Zusammenarbeit im Konstruktions-, Ingenieur- und Bauwesen. Es umfasst neben rein konstruktiver Modellerstellung auch bestimmte administrative Aufgaben (Azhar, 2011).

Im Rahmen der fortschreitenden Digitalisierung entwickelt sich der Begriff BIM zu einem Kernthema der Bauwelt. BIM basiert auf der Idee einer durchgängigen Nutzung eines digitalen Zwillings von einem Bauwerk über den gesamten Lebenszyklus hinweg und stellt dadurch eine Lösungsmöglichkeit für gängige Problematiken im Bausektor dar (Borrmann et al., 2015). Eine Auswahl der vielfältigen Einsatzmöglichkeiten von BIM ist in Abbildung 2.1 schematisch dargestellt.

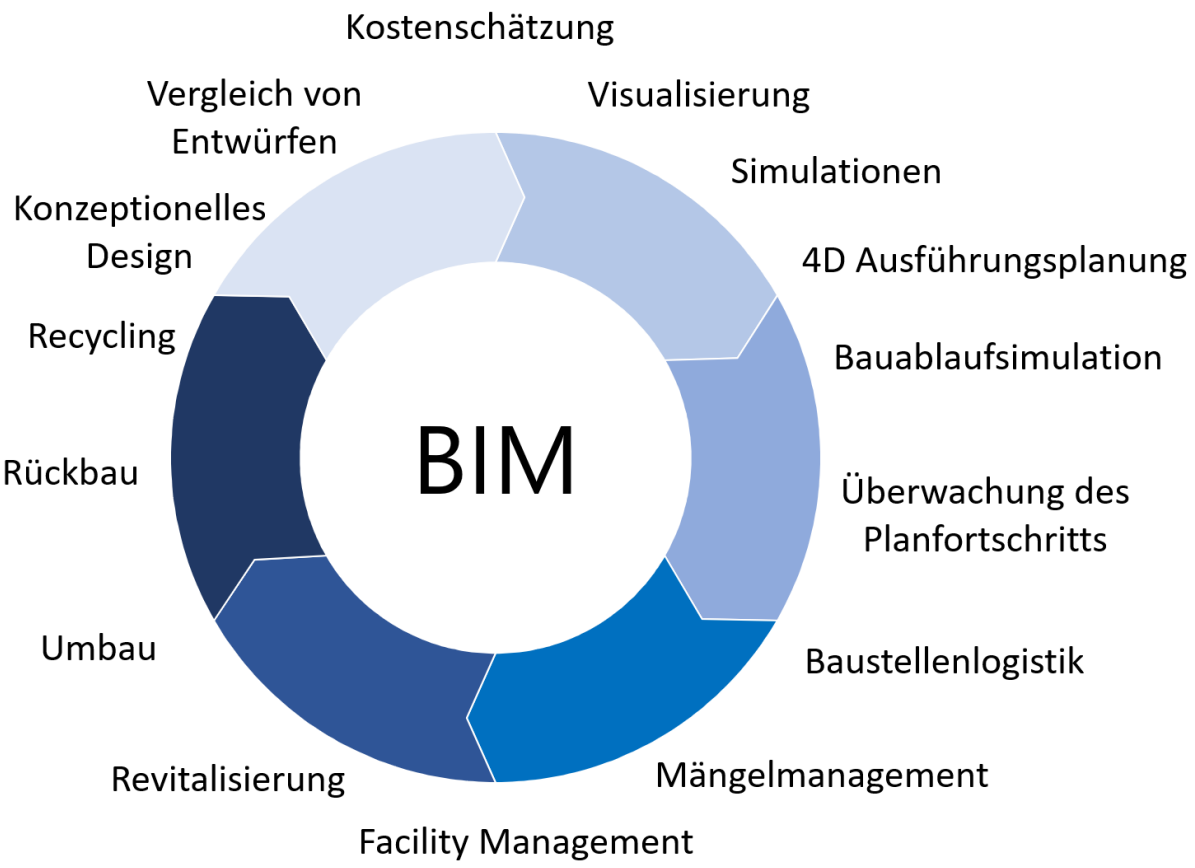


Abbildung 2.1: Auswahl von Anwendungsfälle für BIM-Modelle über den gesamten Lebenszyklus (Borrmann et al., 2015)

2.2 Begriffsdefinition und Bedeutung von BIM

BIM als übergeordneter Begriff besitzt keine allgemein gültige Definition (Carmona et al., 2007). Je nach Auslegung werden unterschiedliche Aspekte verschieden stark gewichtet.

Konsistente Datenerfassung und die Nutzung herstellerneutraler Datenformate für die Datenübermittlung zwischen verschiedenen Fachbereichen ist die Basis der meisten Definitionen. Das BMVI stellt im Stufenplan für Digitales Planen und Bauen folgende Anforderungen an BIM:

„Building Information Modeling bezeichnet eine kooperative Arbeitsmethodik, mit der Grundlage digitaler Modelle eine Bauwerks die für seinen Lebenszyklus relevanten Informationen und Daten konsistent erfasst, verwaltet und in einer transparenten Kommunikation zwischen den Beteiligten ausgetauscht oder für die weitere Bearbeitung übergeben werden“ (Bundesministerium für Verkehr und digitale Infrastruktur, 2015)

Wie der Anforderungsdefinition des BMVI für BIM zu entnehmen ist, wird die Notwendigkeit von dreidimensionalen Geometrien nicht explizit gefordert. In anderen Definitionen von BIM wird diese Anforderung gestellt. Viele BIM-Anwendungsfälle können ohne dreidimensionale geometrische Beschreibung nicht genutzt werden. BIM-Anwendungsfälle sind Prozessschritte, die spätestens im BIM-Abwicklungsplan zum Erreichen der BIM-Ziele definiert werden. Beispielsweise werden Visualisierung, Kollisionserkennung, Massenermittlung diesem Begriff zugeordnet (Bayrische Architektenkammer, 2019).

Daher ist in der Definition nach Borrmann et al. (2015), welches als Beispiel dient, die dreidimensionale Beschreibung vorausgesetzt:

„Unter einem Building Information Model versteht man ein umfassendes digitales Abbild eines Bauwerks mit großer Informationstiefe. Dazu gehören neben der dreidimensionalen Geometrie der Bauteile vor allem auch nicht-geometrische Zusatzinformationen wie Typinformationen, technische Eigenschaften und Kosten. Der Begriff Building Information Modelling beschreibt entsprechend den Vorgang zur Erschaffung, Änderung und Verwaltung eines solchen digitalen Bauwerksmodells mithilfe entsprechender Softwarewerkzeuge." (Borrmann et al., 2015)

Die Nutzung eines digitalen Modells über den gesamten Lebenszyklus bietet ein enormes Potential. Durch den Einsatz von BIM-Methoden können Projektdaten konsequent genutzt werden und das zeitaufwändige Aufbereiten der Daten ab dem Zeitpunkt der Gebäudenutzung vereinfacht werden. Allerdings kommt es durch die mangelnde Verwendung herstellerneutraler Datenformate zu großen Datenverlusten bei Bauprojekten. Wegen der Interoperabilität verschiedener Softwareanwendungen sind im Jahr 2002 bei Planung, Ausführung und Betrieb in Amerika Mehrkosten in Höhe von 15,8 Milliarden US-Dollar angefallen (Gallaher et al., 2004).

Das angestrebte Ziel einer offenen BIM-Integration über alle Phasen eines Bauwerkslebenszyklus wird als big open BIM bezeichnet. Als Kombination aus big BIM und open BIM soll eine Bearbeitung durch verschiedene Softwarelösungen ohne Komplikationen ermöglicht werden. Der Datenaustausch findet in diesem Konzept herstellerneutral statt. Das Prinzip der fächerübergreifenden Arbeit entlang einer interdisziplinären Wertschöpfungskette, welches als big BIM bezeichnet wird, soll durch das Konzept von open BIM, eine Softwarelandschaft mit offenen Schnittstellen zu nutzen, erweitert werden (N+P Redaktion, 2018).

Durch das von buildingSMART entwickelte herstellerunabhängige Datenformat IFC zur Beschreibung von Bauwerksmodellen können, wegen einem vorgegebenen Schema, Datenverluste im Übermittlungsprozess verringert werden. Es beinhaltet umfangreiche Datenstrukturen zur Beschreibung von Objekten.

Mit IFC können Bauwerksmodelle zwischen Applikationen verschiedener Hersteller übertragen werden. Die Repräsentation von Projekt- und Raumstrukturen, Aggregation von Modellelementen und die logischen Beziehungen zwischen Elementen sind darin enthalten (DIN Bauportal GmbH, o.J.). In den vergangenen Jahren wird das IFC-Format zur Realisierung von Open BIM bevorzugt und ist bereits in zahlreichen BIM-Programmen als Austauschformat fest integriert (Borrmann et al., 2015). Mit der Veröffentlichung der Version IFC4 werden geometrische und semantische Informationen, die zur Beschreibung eines digitalen Modells im Bereich des Hochbaus benötigt werden, abgedeckt. Durch die Zertifizierung als ISO-Standard 16739 ist IFC zudem von größerem Interesse für staatliche Bauvorhaben. Beispielweise wird in Norwegen IFC als verbindliches Austauschformat für Vergabe- und Genehmigungsverfahren vorgeschrieben (Borrmann et al., 2015). Bis zu der Erweiterung, IFC4, ist das Datenformat immer noch ausschließlich für den Hochbau konzipiert, wird allerdings derzeit für Infrastruktur erweitert. Mit den Erweiterungen der Version IFC4X1 ist der Grundstein für die Anwendung in der Infrastruktur gelegt worden.

2.3 Building Information Modelling im Infrastrukturbereich

BIM ist ursprünglich für Projekte im Bereich des Hochbaus entwickelt worden. Einige Methoden, die für den Hochbau typische Anwendungsfälle darstellen, sind auch für den Infrastrukturbereich adaptierbar. Beispielsweise können Kollisionsprüfungen von Infrastruktur-Modellen problemlos durchgeführt werden, da diese BIM-Methodik nur ausreichend genaue dreidimensionale Geometrie benötigt (Esser, 2018). Für die Modellierung von Bauwerken im Bereich der Infrastruktur existieren bereits Softwarelösungen, wie beispielsweise Autodesk Civil 3D oder Allplan Bridge. Des Weiteren können etablierte BIM Softwareanwendungen, die für den Hochbau optimiert sind, auch genutzt werden, um infrastrukturelle Tragwerke zu erzeugen (Borrmann et al., 2019a). Vorteile, die für BIM Modelle aus dem Hochbau gelten, wie beispielweise die Erstellung von adaptiven Familien, können übernommen werden (Langwich, 2016).

Die benötigten nicht-geometrischen Informationen der Bauteile werden bei der Modellierung mit etablierten BIM Softwarelösungen nachträglich implementiert. Dabei werden firmenintern eigene Parameteranforderungen definiert, um eine spätere Bearbeitung zu ermöglichen. So kommt es zu einer großen Varianz bei der Frage wie und wann welches Bauteil bestimmte Attribute aufweisen muss. Diese Vorgehensweise ist kritisch zu bewerten, da in der weiteren Bearbeitung die Softwareanwendungen an die selbstdefinierten Parameterstrukturen angepasst werden und somit im Gegensatz zum Ziel einer durchgängigen Planung stehen (Esser, 2018). Eines der großen Probleme, um BIM-Anwendungen im Infrastrukturbereich einzusetzen, ist die räumliche Ausdehnung von linienartigen Bauwerken, die sich über eine große Distanz erstrecken. Bestehende BIM-Methoden, die aus dem Hochbaubereich kommen, sind nur für Bauwerke mit einer räumlichen Begrenzung über wenige 100 Meter ausgelegt. Bei der räumlichen Beschreibung von Modellen aus dem Hochbau und der Infrastruktur ergeben sich zwei unterschiedliche Typologien. Die Objekttypologie „Gebäude“ existiert in beiden Bausektoren und wird mittels Ebenen und Rastern beschrieben. Für linienartige Bauwerke existiert allerdings eine weitere Typologie mit der Bezeichnung Achse (Esser, 2018). Hier erfolgt die Orientierung entlang Achsen. Um die aufgezeigten Problematiken einer flächendeckenden Verwendung von BIM in der Infrastruktur zu lösen, sind zeitaufwändige Standardisierungsprozesse und allgemeine Konzepte für Datenmanagement nötig. Durch das große Interesse im öffentlichen und privaten Sektor sind Erweiterungen für die Infrastruktur bereits in Entwicklung. (Ehle, 2019)

Das BMVI hat einen Stufenplan für digitales Planen und Bauen erstellt und die dadurch initiierten Forschungsprojekte sollen BIM schrittweise für den Infrastruktur-Sektor zugänglich machen. buildingSMART gründete das Projekt Infrastructure Room, um IFC als etabliertes herstellerneutrales und offenes Datenaustauschformat für Bauvorhaben aus dem Infrastrukturbereich zu erweitern (buildingSMART e.V., 2019c). Mit IFC4x1 sind die Projekte IFC Alignment und IFC Overall Architecture in das Datenformat aufgenommen worden und stellen die Grundlage für kommende Erweiterungen dar. IFC Road, IFC Rail und IFC Bridge basieren auf der bereits implementierten Erweiterung (Bormann et al., 2017).

IFC Alignment dient zur Beschreibung von Trassierungsachsen für Gleise und Straßen. Die Trassierungsachse wird im IFC Schema als Alignment bezeichnet und ist folglich essenziell für linienartige Bauwerksmodelle.

IFC Overall Architecture beinhaltet das digitale Geländemodell, Klassifikationen für lineare Bauteile, optimierte Geometrien und die Nutzung geodätischer Koordinatensysteme. Die allgemeinen Anforderungen können somit für Verkehrsinfrastruktur abgedeckt werden. Des Weiteren sind folgende Richtlinien für weitere IFC-INFRA Projekte definiert (Bormann et al., 2017):

- **Minimaler Eingriff**

Eine größtmögliche Kompatibilität zu früheren Versionen wird durch minimale Erzeugung neuer Entitäten ermöglicht.

- **Maximaler Ausbau**

Vorhandene Datenstrukturen und Klassen werden so gut wie möglich wiederverwendet.

- **Internationale Reichweite**

In Datenmodellen sollen nur international anerkannte Elemente vorhanden sein, da IFC als Standard bereits starke Erweiterungsmechanismen besitzt wie beispielsweise *property sets* für nationale und regionale Konzepte.

In Abbildung 2.2 ist die stückweise Entwicklung von IFC für die Infrastruktur schematisch dargestellt.

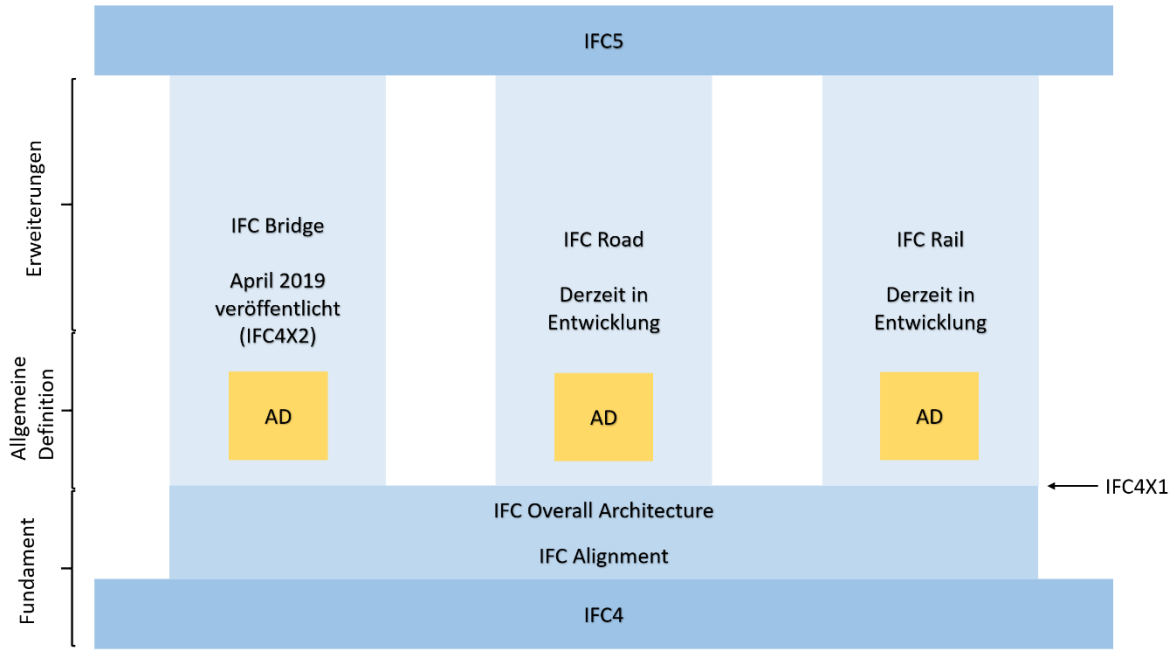


Abbildung 2.2: Projektstruktur für den Erarbeitungsprozess für IFC5. (Liebich, 2017)

Der Entwicklungsprozess von der Version IFC4 zu IFC5 ist veranschaulicht. IFC5 soll einen flächendeckenden Einsatz des Datenschemas für den Infrastrukturbau ermöglichen. Neben den dargestellten Erweiterungen sind die Projekte IfcTunnel und IfcHarbour anzufügen. Die Bachelorarbeit setzt die IFC-Erweiterung IfcBridge in den Fokus. Da Brücken als Unterkonstruktionen für Gleis- und Straßenabschnitte benötigt werden, sind die Projekte IfcRoad und IfcRail zusätzlich abgebildet.

Das IfcBridge Projekt ist die erste auf IFC Overall Architecture und IFC Alignment basierende Erweiterung und wird in Kapitel 3 vorgestellt. Hierbei werden Aufbau, Entwicklungsstand und Neuerungen erörtert und anschließend analysiert.

3 Die IFC-Bridge Erweiterung der Industry Foundation Classes

3.1 Allgemeine Informationen

Als Erweiterung des herstellerneutralen Datenaustauschformates ermöglicht IfcBridge ein Brückenmodell mit angepasster Semantik und Geometrie zu beschreiben (Castaing et al., 2018a). Mit einer Entwicklungszeit von zwei Jahren ist es als „fast track“ Projekt von buildingSMART InfraRoom anzusehen.

IfcBridge basiert auf der IFC Alignment Erweiterung und folgt den Richtlinien des IFC Infra Overall Architecture Projekts. Wegen der kurzen Projektdauer wurde die Anzahl der Darstellungsmöglichkeiten von Brückentragwerke nach ausgiebiger Analyse des Projektteams für die IFC4X2 Version auf folgende fünf Darstellungsarten beschränkt (Castaing et al., 2018a):

- Plattenbrücke
- Balkenbrücke
- Rahmenbrücke und Starrrahmenbrücke
- Düker

Brückenarten die für die nächsten IFC Bridge Erweiterungen angekündigt sind:

- Fachwerkbrücke
- Langerscher Balken
- Bogenbrücke
- Auslegerbrücke
- Hängebrücke und Schrägseilbrücke

Bei Betrachtung der Materialebene sind derzeit nur Kombinationen von Stahl, Beton und Stahlbeton berücksichtigt, damit werden allerdings die Anforderungen der meisten Brückenbauwerke abgedeckt (Borrmann et al., 2019b).

In Abbildung 3.1 werden die Verschiedenen BIM-Anwendungen, die von IFC Bridge unterstützt werden, dargestellt. Hierdurch können die essenziellsten BIM-Methodiken besonders in der Planungs- und Ausführungsphase eines Projektes genutzt werden.

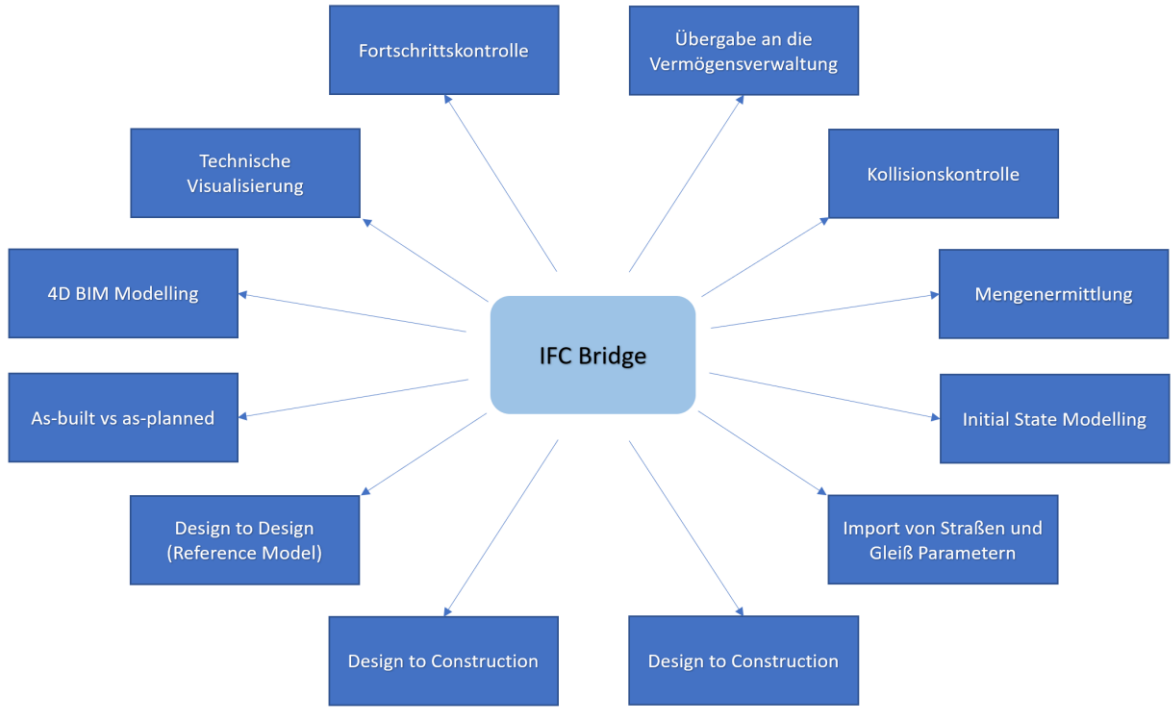


Abbildung 3.1: Von IfcBridge Erweiterung unterstützte BIM-Anwendungen (Borrmann et al., 2019b)

Wegen ihrer hohen Komplexität wurden die Anwendungsfälle *Design to Design (Full model Logic)*, *Structural analysis*, *Code Compliance Checking*, *Drawing generation and exchange* und *Prefabrication and manufacturing* für die Erweiterung IFC4X2 ausgeschlossen. Allerdings befindet sich IfcBridge weiterhin in Entwicklung und wird in den nächsten Versionen die BIM-Anwendungsfälle ergänzen (Castaing et al., 2018a).

3.2 Räumliche Hierarchie

Neben den grundlegenden Beschreibungsmöglichkeiten und Anwendungsmethoden, die in Sektion 3.1 beschrieben werden, ist die Strukturierung von besonderer Bedeutung. Die räumliche Hierarchie spielt hierbei eine wichtige Rolle, da lineare Bauwerke nicht durch die herkömmliche Struktur aus dem Hochbau beschrieben werden kann. Grundlegend sind Brücken auch nach den Grundprinzipien der IFC Modellierung in drei Projektstrukturpläne zu unterteilen, die in Abbildung 3.2 vorgestellt werden.

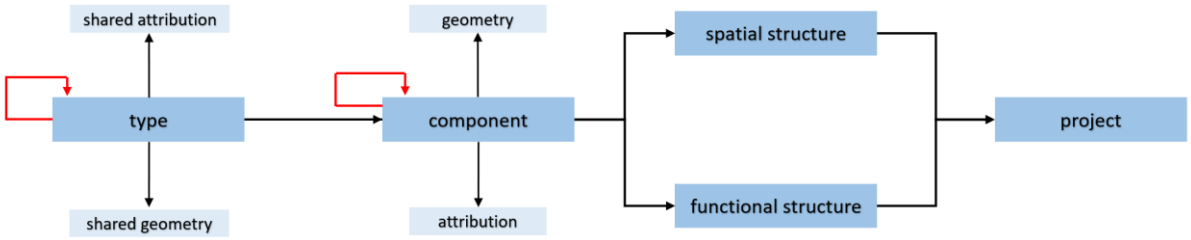


Abbildung 3.2: Aufbau der Hauptstrukturpläne spatial breakdown structure, component breakdown structure und functional breakdown structure (Castaing et al., 2018b)

Bei einem dieser Projektstrukturpläne handelt es sich um die räumliche Hierarchie (*spatial heirarchy*), die als minimale Voraussetzung eines Projekts drei Ebenen fordert. An oberster Stelle jeder IfcBridge Datei ist das *IfcProject* angeordnet. Diese Hierarchie-Ebene enthält allgemeine Informationen, wie beispielsweise Längeneinheiten und Projektkoordinaten, die für das gesamte IFC Modell gelten. Dem *IfcProject* untergeordnet ist die *IfcSite*, die die Baufläche definiert. Hier können der Bodenaufbau und mögliche Bestandsgebäude enthalten sein. An unterster Stelle werden die allgemeinen Informationen der Brücke in der Ebene *IfcBridge* festgehalten (Castaing et al., 2018b).

Bei einer Brücke handelt es sich um eine komplexe Tragstruktur, deren Bauteile unterschiedliche Eigenschaften aufweisen müssen. Durch die Ebene *IfcBrigepart* kann die räumliche Struktur eines Brückentragwerks detaillierter unterteilt werden. Hierbei wird zwischen drei Typisierungen unterschieden.

Bridgeparts des Typs COMPLEX definiert einen Teilbereich einer Brücke, der aus einer Vielzahl von Bauteilen bestehen kann. Der Typ ELEMENT definiert eine Zuordnung einer monolithischer Brückenkomponente zu einer Entität. Falls diese jedoch aus mehreren Bauteilelementen besteht wird es als PARTIAL typisiert (buildingSMART e.V., 2019b). Im Folgenden werden die für *IfcBridgepart* existierenden Typen aufgelistet und zugeordnet.

- Abutment (ELEMENT oder PARTIAL)
- DECK / DECK_SEGMENT (ELEMENT oder PARTIAL)
- FOUNDATION (ELEMENT oder PARTIAL)
- PIER / PIER_SEGMENT (ELEMENT oder PARTIAL)
- PYLON (ELEMENT oder PARTIAL)
- SUBSTRUCTURE (COMPLEX)
- SUPERSTRUCTURE (COMPLEX)

- SURFACESTRUCTURE (COMPLEX)
- USERDEFINDED / NOTDEFINED

Substructure, *Superstructure* und *Surfacestructure* werden genauer beschrieben, da alle Brückenkomponenten in einer dieser *IfcBridgeparts* definiert sind. Dies ermöglich eine schnelle Gliederung der Aufgabenbereiche verschiedener Elemente.

Die *Substructure* erfasst alle Komponenten, die maßgeblich für Tragfähigkeit und Weiterleitung der Krafteinflüsse in den Boden verantwortlich sind. Für die Beschreibung des oberen Anteils eines Brückentragwerks, inklusive Straßenaufbau sowie Sicherheits- und Leitsysteme, wird die Kategorie *Superstructure* verwendet. Die *Surfacestructure* als dritte Klasse, hat die Besonderheit ausschließlich Elemente aus der *Superstructure* zusätzlich zu klassifiziert. Dies ist maßgeblich für eine klare Beschreibung der verschiedenen Schichten der Straßenstruktur. Bauteilelemente, die durch Entitäten beschrieben sind, können auch durch die Ebene *IfcBridgepart* in die räumliche Hierarchie mit aufgenommen werden. Hierbei ordnet sich die Entitäten zur Beschreibung der Bauteile der Hauptstrukturebene unter (Castaing et al., 2018a).

IfcRelAggregates als Aggregationsbeziehung übergibt im IFC-Schema untergeordnete Ebenen an die nächst höhere weiter. Einzelne Bestandteile der räumlichen Hierarchie werden dadurch untereinander in Beziehung gesetzt (Castaing et al., 2018b).

In Abbildung 3.3 wird anhand eines Brückenmodells und eines Baumdiagrammes die räumliche Hierarchie für eine bessere Veranschaulichung dargestellt.

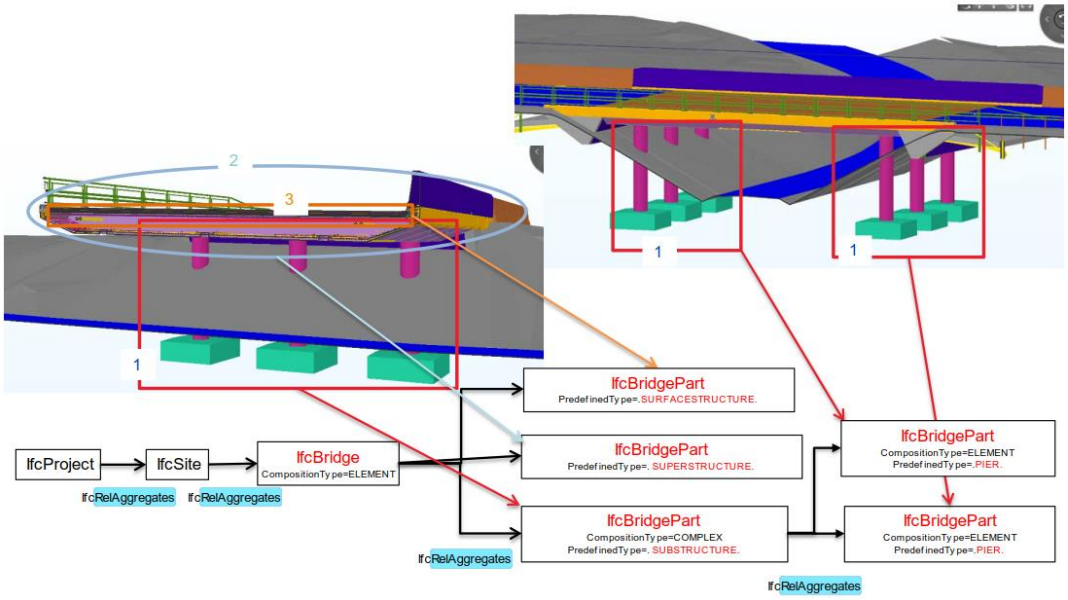


Abbildung 3.3: Schematischer Aufbau der räumlichen Hierarchie eines IfcBridge-Modells (Castaing et al., 2018b)

Die rot markierten Ebenen der räumlichen Hierarchie einer Brücke im IFC-Schema sind durch die IfcBridge Erweiterung eingeführt worden. *IfcBridge* dient zur Beschreibung der gesamten Brückenstruktur und ist das Äquivalent zu *IfcBuilding*, das zur Beschreibung von Gebäuden genutzt wird. Die Ebene *IfcBridge* lässt sich in den Unterbau, Oberbau und Straßenaufbau unterteilen. Diese bestehen aus mehreren Brückenkomponenten, die durch verschiedene *IfcBridgeparts* in die Hierarchie eingegliedert sind.

3.3 Hinzugefügte Entitäten und Predefined Classes

Durch die Implementierung der IFC Overall Architecture Guideline sind möglichst wenig neue Entitäten geschaffen worden. Ein gutes Beispiel hierfür ist ein Gehweg, der als *IfcSidewalk* anstelle von einer neuen Klasse, als neuer vordefinierter Typ zu *IfcSlab* zugeordnet wird. Neue Entitäten zur Beschreibung brückenspezifischer Elemente werden ergänzt, um eine vollständige Beschreibung aller Brückenkomponenten zu ermöglichen. Abbildung 3.4 bildet eine Gegenüberstellung der neuen Entitäten und einer Auswahl bereits bestehender Klassen ab.

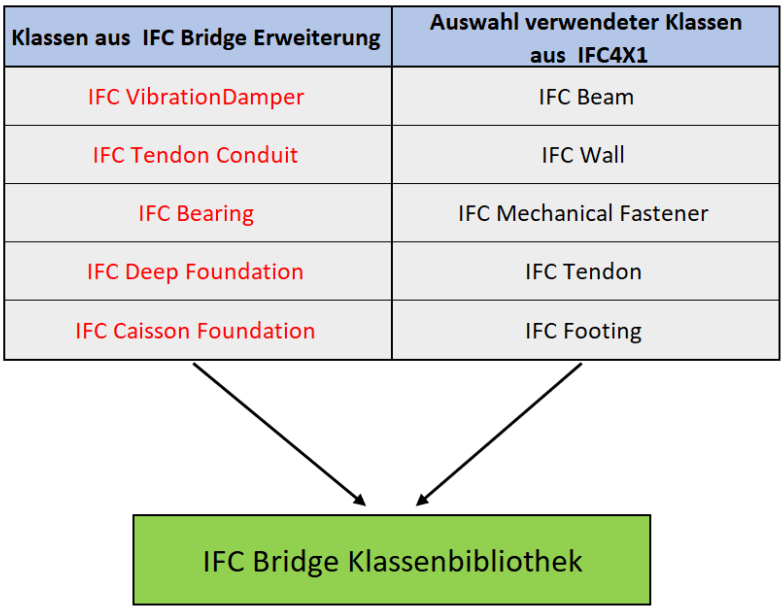


Abbildung 3.4: Gegenüberstellung der neu erstellten Entitäten aus IFC4X2 und vorhandener aus IFC4X1 (Borrmann et al., 2019b)

Die eingeführten Entitäten ermöglichen eine Beschreibung von Elementen einer Brücke, die nicht durch Ergänzung bestehender Klassen definiert werden können.

Die meisten Brückenelemente eines Modells können mit bereits bestehenden Klassen definiert werden. Dafür müssen *predefinded types* zur Unterkategorisierung der vorhandenen Entitäten hinzugefügt werden, damit ein Bauteil genauer beschrieben werden kann. In Tabelle 3.1 werden die neu definierten Unterklassen dargestellt.

Tabelle 3.1 Übersicht der predefined types für eine Beschreibung von Brückenelementen mit bestehenden Entitäten (Borrmann et al., 2019b)

IFC Member	IFC Plate	IFC Beam	IFC Element Assembly
- Stiffening Rib - Arch Segment - Suspension Cable - Suspender - Stay Cable	- Flange Plate - Web Plate - Stiffener Plate - Gusset Plate - Splice Plate - Cover Plate - Base Plate	-Girder Segement -Diaphragm -Piercap -Hatstone -Cornice - Edgebeam	- Abutment - Pier - Pylon - Cross Bracing - Deck
IFC Slab	**IFC Column**	**IFC ProjectElement**	**IFC Building Elementpart**
- Approach Slab - Paving - Sidewalk - Wearing	- Pierstem - Pierstem Segment -Standcolumn	- Blister - Deviator	- Apron
IFC Covering	**IFC Wall**	**IFC Georaphic Element**	**IFC Mechanical Fastener**
- Coping	- Retainingwall	- Soil Boring Point	- Coupler
IFC VibrationIsolator	**IFC SurfaceFeature**	**IFC Discrete Accessory**	
- Base	- Defect	- Expansion Joint Device	

Durch die Einführung vordefinierter Klassen von Brückenelementen wird die Voraussetzung aus der Erweiterung IFC Overall Architecture, eine größtmögliche Kompatibilität zu Vorgängerversionen zu schaffen, eingehalten. Viele Elemente eines IfcBridge-Modells können mit diesen beschrieben werden.

3.4 Anpassung geometrischer Repräsentationen an lineare Bauteile

Um die BIM-Anwendungsfälle aus Tabelle 3.1 für Brückentragwerke anzuwenden, werden bestimmte geometrische Repräsentationen vorausgesetzt. Zum einem können hierfür explizite BRep Geometrien (*Boundary Representation Geometrien*) verwendet werden. Zum anderen ist eine Repräsentation über implizite Geometrien, wie beispielweise *Sweeps*, *Rotations* und *Lofts*, möglich (Borrmann et al., 2019b).

In den in Kapitel 3.1 beschriebenen BIM-Anwendungen sollen die geometrischen Repräsentationen von den Komponenten der *Surfacestructure* über *Sweeps* erzeugt werden. Die Nutzung von *TriangulatedFaceSets* für die genannten Elemente würde zu einer Erhöhung der Datenmenge und einer ungenaueren Repräsentation führen, wodurch manche unterstützen Anwendungen nicht verwendet werden können.

IfcTriangulatedFaceSets stellt die Geometrie eines Bauteils durch ein mosaikartiges Gefüge von aneinandergereihten Dreieckflächen dar. Da sich die Komponenten der *Surfacestructure* an den Verlauf des Alignments anpassen, kann dadurch die Geometrie nicht ideal beschrieben werden. Für eine rein repräsentative Beschreibung dieser Komponenten oder für eine Darstellung von rechteckigen Körpern bietet sich *IfcTriangulatedFaceSet* an (Borrmann et al., 2019b).

IfcSectionedSolidHorizontal, eine speziell entwickelte geometrische Beschreibung für den Infrastrukturbereich, ist im Rahmen der Erweiterung IFC Alignment in der IFC4X1 Version dem IFC Schema hinzugefügt worden und spielt in diesem Kontext eine wichtige Rolle. Im Vergleich zu gewöhnlichen *IfcSweptSolidAreas* kann ein *Sweep* entlang einer *IfcCurve* durchgeführt werden, wobei die *Cross Sections* variiert werden können. *Cross Section*s des y-Vektors weisen stehts in Richtung der globalen z-Achse. Dies bedeutet, dass die Körper, die durch *IfcSectionedSolidHorizontal* dargestellt werden, stets in die globale z-Richtung ausgerichtet sind (Borrmann et al., 2019b). Dennoch werden beide Repräsentationsmethoden benötigt, um alignmentbasierte Geometrien darstellen zu können (Castaing et al., 2018a). In Abbildung 3.5 wird die geometrische Repräsentationsmethode *IfcSectionedSolidHorizontal* veranschaulicht.

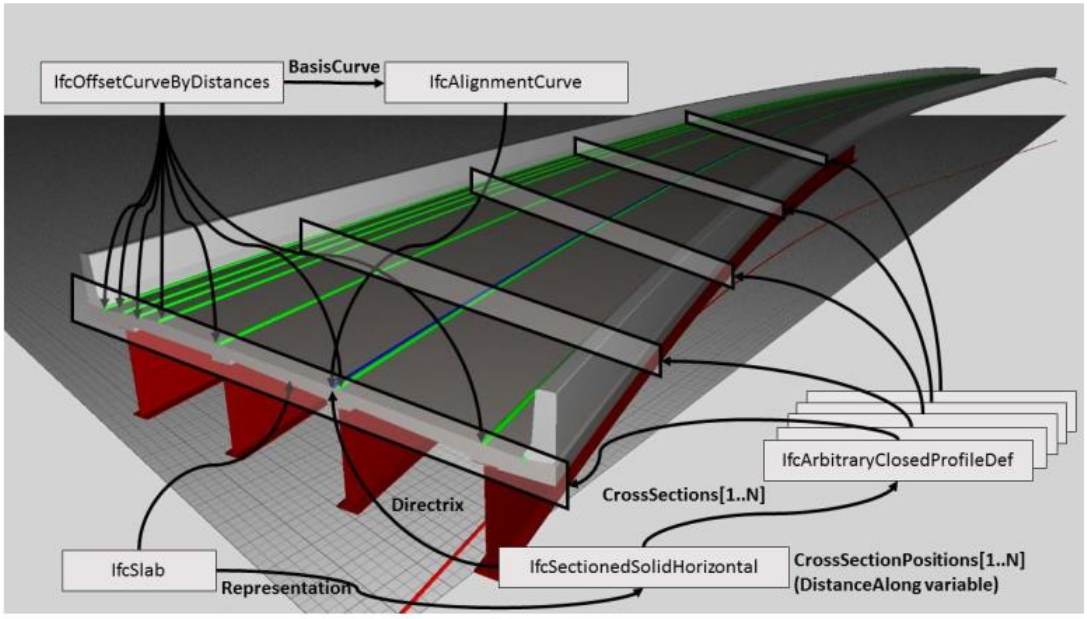


Abbildung 3.5: Visualisierung der geometrischen Repräsentation einer Brücke durch IfcOffsetCurveByDistances, IfcArbitraryClosedprofileDef, IfcSectionedSolidHorizontal und IfcAlignmentCurve (Markič, 2017)

Die Brückenträger sind mit der geometrischen Repräsentation *IfcSectionedSolidHorizontal* im IFC-Format erstellt. Die Orientierung der Komponenten entlang der Basiskurve Alignment erfolgt durch parallel verlaufende Kurven. Entlang des Alignments sind *CrossSections* definiert, die den Querschnitt des Trägers vorgeben.

3.5 Model View Definitions

Eine Model View Definition (MVD) im IFC-Datenformat verringert die Komplexität eines Modells, indem nur eine Teilmenge des IFC-Schemas definiert wird. Mittels enthaltener Leitlinien und Vereinbarungen für die Implementierung der genutzten IFC Konzepte können mehrere fachspezifische Austauschanforderungen erfüllt werden (buildingSMART e.V., 2019a).

Für das IfcBridge Dateiformat sind Bridge Reference View (RV), Alignment-based Bridge Reference View (ARV), Bridge Design Transfer View (DTV) und Bridge Asset Management Handover View als brückenspezifische MVDs definiert. RV und DTV sind aus der Vorgängerversion IFC4 übernommen und werden für Brückentragwerke ergänzt (Borrmann et al., 2019b). Der Hauptunterschied zwischen den beiden Teilmengendefinitionen liegt in der geometrischen Repräsentation von Elementen.

IfcCSGSolid Geometrien (*CSG = Constructive Solid Geometry*) werden nicht in der MVD Reference View unterstützt. Um diese Repräsentation verwenden zu können muss die Bridge Design Transfer View verwendet werden (Borrmann et al., 2019b) .

Ein weiterer Unterschied liegt in der Beschreibung von BRep-Geometrien. DTVs sind in der Lage diese durch *IfcFacetedBrep* und *IfcAdvancedBrep* zu beschreiben. RVs müssen auf *IfcPolygonalFaceSet* zugreifen, um BRep repräsentieren zu können (Castaing et al., 2018a). Die MVD ARV ist um die Verwendung von *IfcAlignments* und *IfcSectionedSolidHorizonta*l erweitert worden und hebt dadurch die Bedeutung eines Alignments für lineare Infrastruktur hervor (Borrmann et al., 2019b).

Generell sollen komplexe Brückenkonzepte durch ein Alignment beschrieben werden. Alignment-basierte IfcBridge-Modelle werden von allen MVDs, außer der RV, unterstützt. Durch das RV können auch Brückenmodelle, die ohne Alignment digital konstruiert sind, in das IFC4X2-Schema übergeben werden. Die Alternative zu einer align-

mentbasierten Repräsentation ist in der Einführungsphase von BIM im Infrastrukturbereich eine sinnvolle Maßnahme, da viele existierende BIM-Softwareanwendungen noch nicht für infrastrukturelle Bauwerke ausgelegt sind.

4 Thematisch verwandte wissenschaftliche Arbeiten

Bereits in vorherigen Arbeiten und Publikationen sind Modellierungsprozesse, Bearbeitungen und Datentransferierungen für BIM-Softwareanwendungen untersucht worden. Die Arbeit von (Nitschke, 2015) setzt sich mit der Umsetzung von BIM im Bereich des Ingenieurbaus anhand von Brückenmodellen auseinander. Dabei werden ausschließlich Produkte von Autodesk verwendet, um ein Modell zu erzeugen und dessen Nutzungsmöglichkeiten aufzuzeigen. In der Publikation von (Trzeciak et al., 2018) werden Brückenmodelle zwischen den kommerziell genutzten BIM-Softwareanwendungen Autodesk Revit und Allplan mithilfe von IFC übermittelt. Um das Design-to-design Szenario zu testen wird die Version IFC2X3 für den Datenaustausch verwendet. Anhand eines BIM-Modells der Donnersberger Brücke in München wird eine Machbarkeitsstudie einer Nutzung von IFC4X1 für einen Datenaustausch von Brückenmodellen durchgeführt (Raidl, 2018). Eine Übersicht früherer Entwürfe für eine IfcBridge Erweiterung wird in einer Publikation von (Markič, 2017) präsentiert. (Borrmann et al., 2019b) stellt den Aufbau der IfcBridge Erweiterung dar und geht auf das Projektmanagement während eines IfcBridge Projekts ein.

Durch die kürzliche eingeführt Version IFC4X2 ist die Erweiterung IfcBridge in das IFC Schema integriert. Allerdings findet diese Erweiterung derzeit noch wenig Anwendung in BIM-Methoden. Eine Exportmöglichkeit eines IfcBridge-Modells aus einer kommerziellen Softwareanwendung ist noch nicht vorhanden. Durch die im folgenden Kapitel erklärten Schritte einer simulierten Vorplanungsphase für ein Brückenmodell, sollen Erkenntnisse gewonnen werden, wie geometrische und semantische Informationen von Brücken in Autodesk Revit 2020 nach Übermittlung aus IW 2020 dargestellt werden. Durch das gesammelte Wissen ist eine prototypische Applikation entwickelt worden, die unter Zuhilfenahme visueller Programmierung Brückenmodelle als IFC4X2-Datei abspeichert.

5 Konzeptionelle Modellierung, Modelverfeinerung und Datensortierung

Dieses Kapitel stellt den Erstellungsprozess eines Beispielmodells vor, um Zusatzinformationen und Erkenntnis über die Geometrie eines infrastrukturellen Bauwerkes in existierenden BIM-Softwareanwendungen zu erlangen. Für die konzeptionelle Modellierung wird Autodesk IW 2020 verwendet. Wegen einer proprietären Exportmöglichkeit in IW erfolgt die anschließende Modellverfeinerung in Autodesk Revit 2020. Hier ist als Plugin die visuelle Programmiersprache Dynamo integriert, die eine zusätzliche Datenaufbereitung durchführt.

Obwohl es für manche BIM-Modellierungssoftware Plugin-Lösungen für das konzipieren einer Brücke gibt, existiert noch keine Anwendung, in der die IFC Erweiterung IfcBridge für eine herstellerneutrale Datenübermittlung verwendet werden kann.

Dieses Kapitel beschreibt den Entwicklungsprozess eines konzeptionellen Modells einer Brücke, wobei Einschränkungen und fehlerhafte Ergebnisse analysiert werden. Es sollen Erkenntnisse für den Aufbau des geplanten IfcBridgeExport Toolkits gewonnen werden.

5.1 Konzeptionelle Modellierung, Modellverfeinerung und Datensortierung eines Beispielmodells

5.1.1 Allgemeines

Bei Autodesk IW 2020 handelt es sich um eine Infrastruktur-Planungssoftware im Bereich des BIM. Ingenieure und Konstrukteure können durch die Bereitstellung verschiedener Werkzeuge in kurzer Zeit konzeptionelle Ideen für Straßen und Brücken in einer virtuellen Umgebung erstellen (Autodesk, 2019).

Variantenanalysen eines Projekts können durch schnelle Abänderung von Modellelementen in dieser Software mit geringem Aufwand angewendet werden. Wegen einer direkten Schnittstelle zu der etablierten BIM-Anwendung Autodesk Revit 2020 ist ein fließender Übergang von der Erstellung des konzeptionellen Modells zu einer Modelverfeinerung möglich (Esser et al., 2019).

5.1.2 Erstellungsprozess der Beispielbrücke

Konstruktion des Straßenabschnitts

Bevor eine Brücke in IW geschaffen werden kann, muss eine Komponentenstraße erstellt werden. In diesem Schritt kann zwischen einer vielfältigen Auswahl an Straßenausführungen gewählt werden. Benötigte Radien und Längen der Straßenteilabschnitte werden durch, dafür optimierte, Werkzeuge erzeugt. Das erstellte Straßenmodell besitzt bereits eine normgerechte Querneigung und die Längsneigung kann in einer spezialisierten Ansicht der Straßengradiente korrigiert werden. In Abbildung 5.1 wird der fertige Straßenabschnitt mit einem Straßenquerschnitt und Höhenplan dargestellt.

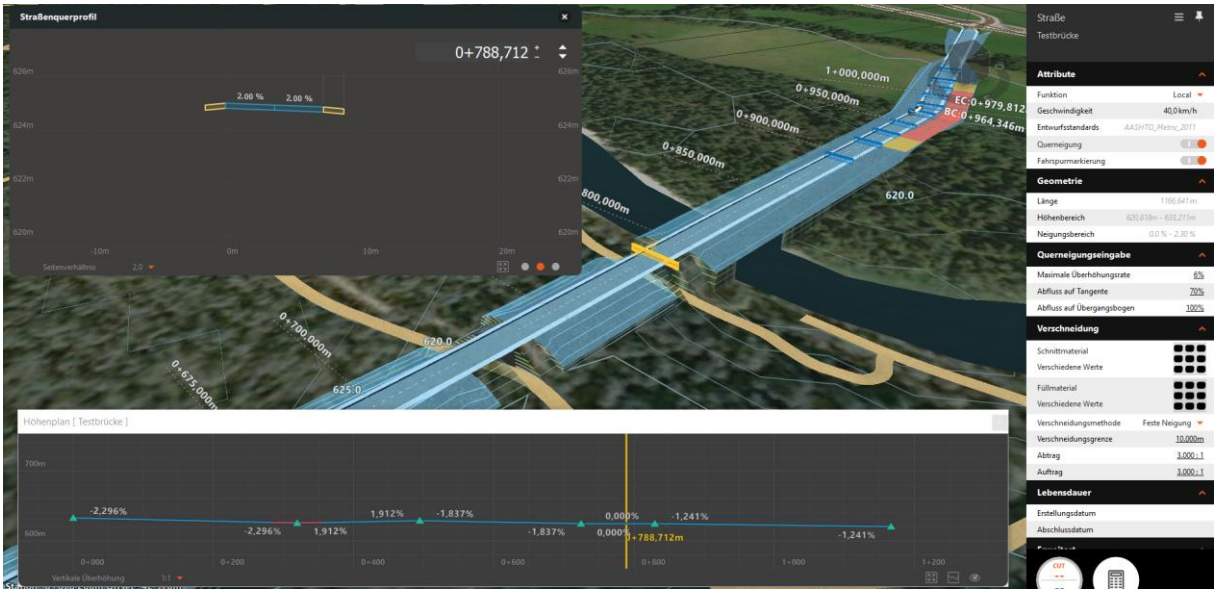


Abbildung 5.1: Straßenteilabschnitt mit Höhenplan und Straßenquerschnitt

Eine Komponentenstraße führt über Landwirtschaftswege und einen Fluss, wobei die Längsneigung der Straße anhand des Höhenplans angezeigt wird. Das Straßenquerprofil veranschaulicht die Querneigung, die automatisch bei der Erstellung einer Komponentenstraße berücksichtigt wird.

Einfügen der Brücke in das Straßenmodell und anschließende Modifikation

Brückentragwerke können als Strukturtyp der Straße in das Modell eingefügt werden. Ein standardisiertes Brückenmodell wird nach Festlegung der Lage generiert. Typische Komponenten einer Brücke, deren Anzahl und Länge an die Dimensionierung

des Tragwerks angepasst sind, werden in diesem Prozessschritt dem Modell hinzugefügt. Eine Änderung der Elementtypen und Anpassung der entsprechenden Parameter ist möglich, wodurch unterschiedliche Darstellungsmethoden des konzeptionellen Modells getestet werden können. Sicherheits- und Leitsysteme werden als Gestaltungselemente in dem Modell platziert. Die Geometrie ist, im Gegensatz zu den Bauteilen, nicht veränderbar. Die Platzierung erfolgt an, parallel zu der Trassierungsachse verlaufenden, Nebenachsen. Der Abstand dieser Elemente zur Kurve und deren Ausrichtung kann manuell abgeändert werden. Die Objektbibliothek von IW kann in der Version 2020.0 durch länderspezifische Erweiterungen Country-Kits ergänzt werden. In diesen Erweiterungen sind, entsprechend dem nationalen Standard des ausgewählten Landes, vorgeschriebene Bauteilelemente, Straßenklassen und Gestaltungselemente enthalten. Gestaltungselemente können auch durch das Hinzufügen eigener parametrischer Modelldateien des Typs Inventor Part File ergänzt werden. Im Fallbeispiel dieser Arbeit wird allerdings darauf verzichtet, da IW ausschließlich zur Erstellung eines konzeptionellen Modells verwendet wird.

Das generierte Brückenmodell mit standardisierter Tragstruktur, dargestellt in Abbildung 5.2, wird mit dem vollendeten Brückenmodell, dargestellt in Abbildung 5.3, verglichen. Den Gestaltungsspielraum den IW mit den Onboardmitteln einem Ingenieur bietet, wird hier deutlich.

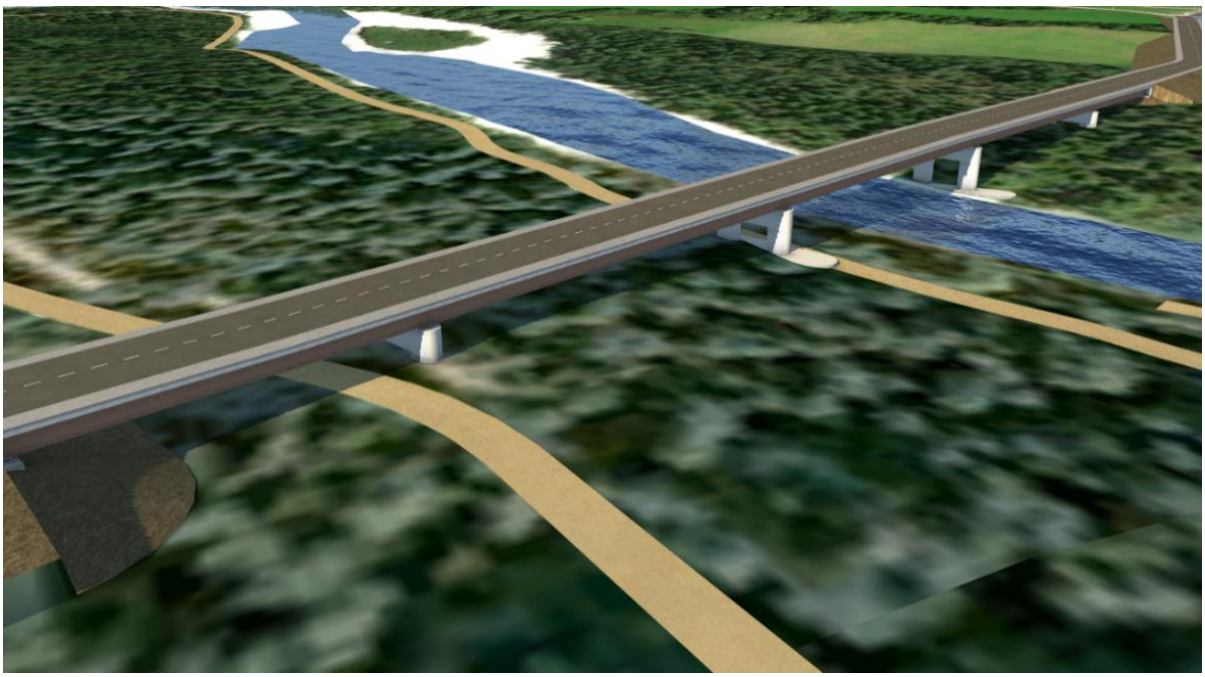


Abbildung 5.2: Hinzufügen der standardisierten IW-Brücke in die Straßenstruktur

Eine Brücke ist als Teilkomponente der erstellten Straße hinzugefügt worden, wobei alle benötigten Tragelemente enthalten sind. Der Aufbau der Straße hat sich nicht geändert.

Abbildung 5.3: Modifikation der Brückenelemente und Platzierung von Gestaltungselementen

Leit- und Sicherheitssystem sind dem Straßenabschnitt hinzugefügt worden. Die Brückenträger und Pfähle sind manuell geändert und angepasst.

5.1.3 Fehleranalyse der InfraWorks-Werkzeuge

Im Rahmen der Erstellung des Brückenmodells wurden die implementierten Modellierungswerkzeuge genau getestet. Hierbei wurden Probleme, die die Verwendung von Modellierungsprozessen in IW einschränken, festgestellt. Im Folgenden ist ein kurzer Überblick der drei Hauptproblemstellungen und deren Auswirkungen zu finden.

Geometrische Anpassung der Gestaltungselemente bei Längsneigungen

Brücken, als Bestandteil eines linienförmigen Bauwerks, müssen Querneigungen und Längsneigung im Rahmen von Minimal- und Maximalneigungen besitzen. Somit wird ein kontrollierter Abfluss von Niederschlagswasser gewährleistet. Gestaltungselemente zählen in IW allerdings nicht als Straßenkomponenten und werden als unver-

änderbare geometrische Körper entlang einer Hilfsachse positioniert. Diese Gestaltungselemente werden zur genaueren Visualisierung von linienförmigen Bauwerken platziert und können nicht an den geringen Winkel der Längsneigung angepasst werden. Am Beispiel des Elements Leitsystem wird deutlich, dass eine Weiterverarbeitung der daraus gewonnen Daten sich als zeitaufwändig erweist, da eine manuelle Abänderung der Winkel von jedem in IW eingefügten Gestaltungselement nötig ist. In Abbildung 5.4 wird die Problematik anhand falsch positionierter Lärmschutzbarrieren verdeutlicht.

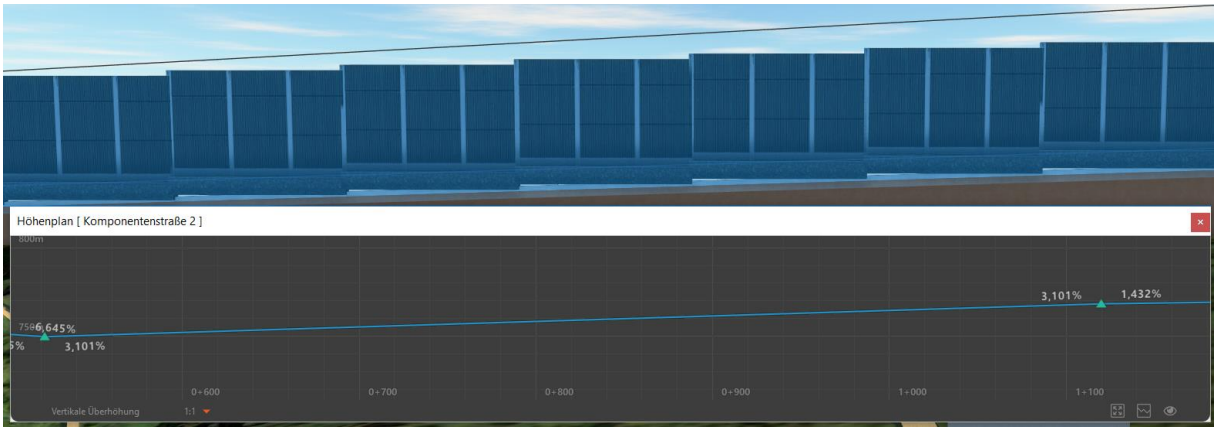


Abbildung 5.4: Vergleich der falsch ausgerichteten Lärmschutzbarrieren zu der benötigten Längsneigung

Lärmschutzbarrieren, die als Gestaltungselement einer IW-Brücke hinzugefügt sind, passen sich nicht an der vorgegeben Längsneigung von 3,101% an. Eine inakkurate Repräsentation des hinzugefügten Elements ist als Resultat festzustellen.

German Country-Kit

Die länderspezifische Erweiterung der Objektbibliothek ermöglicht eine bessere Darstellung von Brücken durch standardisierte Elemente eines entsprechenden Landes. Die deutsche Erweiterung weist im Vergleich zu anderen Ländern, wie Norwegen und Schweden, einige Defizite auf, die im Folgenden erläutert werden.

Durch die Verwendung der länderspezifischen Bibliothek kann eine Vielzahl an DIN-normierten Straßentypen zur Beschreibung des Straßenaufbaus verwendet werden. Bei Erstellung von Komponentenstraßen, die man zur Generierung von Brücken benötigt, können jedoch diese länderspezifischen Klassen nicht verwendet werden. Aus-

schließlich bei gewöhnlichen Straßen werden die DIN Straßentypen als Option angegeben. Ein potenzieller Lösungsweg besteht darin, Straßen zu Komponentenstraßen umzuwandeln, allerdings entstehen dadurch Einschränkungen im Modellierungsprozess. Straßenteilabschnitte des Typs Straße werden im Gegensatz zu Komponentenstraßen mittels Mausklick generiert und besitzen folglich nicht die exakte Länge, die vom Anwender benötigt wird. Desweitern besteht keine Option den Radius von Übergangsbögen manuell zu bestimmen. Die normgerechte Repräsentation einer Straße geht daher mit dem Verlust einer genaueren Bestimmung der geographischen Lage einer Trassierungsachse einher.

Gestaltungselemente, die für eine korrekte Darstellung nationaler Sicherungs- und Leitsysteme verantwortlich sind, existieren in der derzeitigen länderspezifischen Erweiterung für Deutschland nicht. Der Rückgriff auf eine allgemeine Darstellung ist nötig und muss im zweiten Schritt der Vorplanung während der Modellverfeinerung ersetzt werden.

Fehlerhafte Trassierungsachse von Straßentypen

Das Alignment eines linienförmigen Bauwerks in der Infrastruktur ist in Bezug auf den neuen IfcBridge Standard ein Kernelement und repräsentiert die Trassierungsachse eines Projekts. Bei der Erstellung einer Komponentenstraße unter Verwendung bestimmter Straßentypen wird die Schwerpunktachse an der Außenseite des Bauwerks platziert. Da eine Anpassung dieser Achse nicht möglich ist, kann die fehlerhafte Platzierung des Alignments nicht nachträglich korrigiert werden. Bei der Generierung von Brückentragwerken für eine dieser Straßen, entsteht demnach eine fehlerhafte Positionierung der Tragstruktur. Die hinzugefügten Elemente werden in Abhängigkeit des Alignments platziert und die daraus resultierende Verschiebung ist in der IW Darstellung nicht direkt erkennbar. Wenn das Projekt nach dem Export in Revit betrachtet wird, ist der Unterschied unmittelbar erkennbar. Die betroffenen Bauteile besitzen neben inkorrekter Position auch eine fehlerhafte Geometrie. Das fehlerhafte Modell muss bei falscher Position des Alignments aufwändig durch einen Fachplaner nachgebessert werden. Deshalb bieten sich die Straßentypen mit fehlerhafter Alignment-Platzierung nur für Visualisierung eines Modells in IW an. In Abbildung 5.5 ist durch ein falsch platziertes Alignment die Auswirkung auf die Darstellung in Revit veranschaulicht.

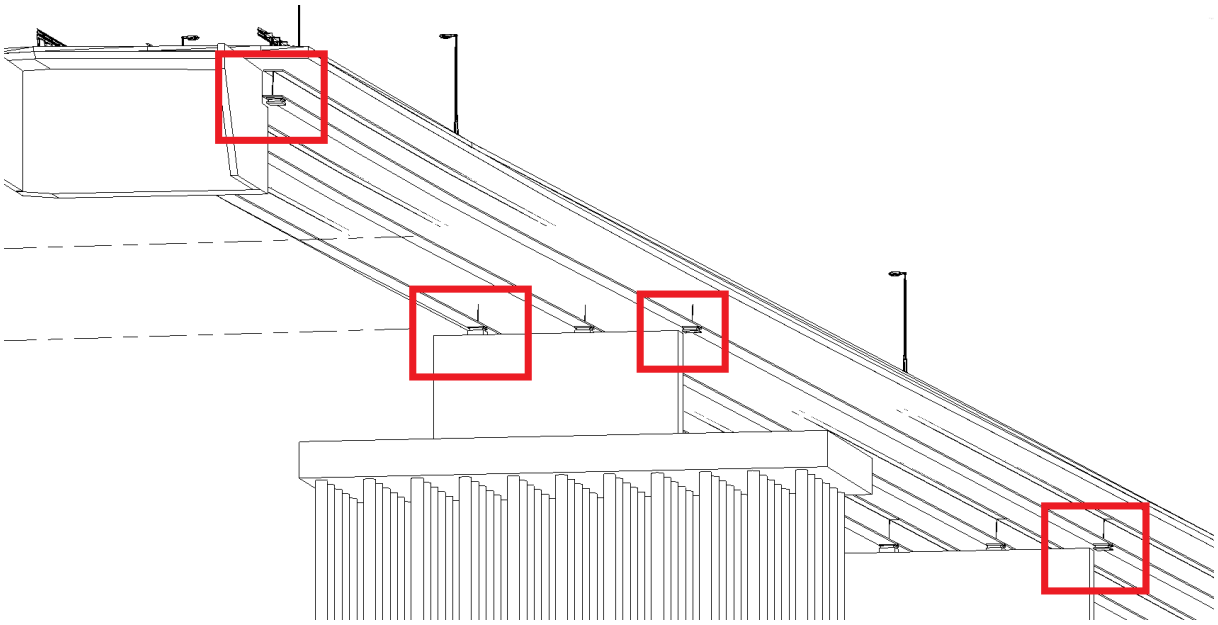


Abbildung 5.5: Geometrische Repräsentation der Beispielbrücke in Revit. Fehlerhafte Bauteile Rot umrandet.

Die rotmarkierten Bereiche zeigen die fehlerhafte Positionierung des Unterbaus an. Der Brückenpfahl besitzt eine Querverschiebung, sodass die äußeren Brückenträger nicht die Kräfte auf diesen weiterleiten können.

5.2 Analyse der Modellinhalte einer Beispielbrücke zwischen InfraWorks und Revit

Die Übergabe der erstellten Daten in die tatsächliche Umgebung ist eine notwendige Aufgabe im Erstellungsprozess eines Modells. Das geodätische Koordinaten Referenz System (gKRS) muss unter Berücksichtigung von Verzerrungen dafür festgelegt werden. Das verwendete gKRS repräsentiert die Metadaten des Projizierten Koordinaten System (PKS), ein rechtsgerichtetes dreidimensionales kartesisches Koordinatensystem (Markič et al., 2019). Dies ist beispielweise für Visualisierung und Kalkulationen eines Modells von Vorteil. Um das in IW erstellte Beispielmodell über die Schnittstelle in Revit zu exportieren, wird ein PKS benötigt. Das Gauß-Krüger-Koordinatensystem wird für diesen Vorgang verwendet.

Trotz einer proprietären Schnittstelle zwischen IW 2020 und Revit 2020 ist ein Verlust semantischer Informationen des Beispielmodells zu erkennen und wird im Folgenden beschrieben.

Alignment

Revit ist eine häufig genutzte BIM-Software für den Hochbaubereich, welche keine Erweiterungen für digitale Modelle im Infrastrukturbereich beinhaltet. Daher ist der Verlust des Alignments, als Grundelement einer Brücke, während der Datenübertragung nachvollziehbar. Folglich kann beispielsweise die Positionierung und geometrische Beschreibung eines Brückenträgers nicht mehr in Abhängigkeit von einem Alignment beschrieben werden.

Kategorisierung der Elemente

In der BIM basierten Software Revit werden Komponenten in verschiedenen Kategorien zugeordnet. Ähnliche Objektfamilien werden wiederum in Kategorien zusammengefasst, wobei eine Objektfamilie in weitere Familientypen unterteilt wird (Esser et al., 2019).

Objekte einer Revit-Datei enthalten nicht nur geometrische, sondern auch semantische Informationen, wie zum Beispiel Material, Kosten oder Erstellungsdatum. Alle exportierten Brückenkomponenten der *Substructure* werden in Revit mit einer *Direct Shape Geometrie* dargestellt und sind über generische Familien klassifiziert. Die Namen der Kategorien werden aus IW übernommen, sodass beispielsweise ein Fundament die Typbezeichnung Brückenfundament trägt. Die geometrischen Parameter der Bauteile sind in dem Revit-Modell enthalten. Im Vergleich zu IW können diese nicht nachträglich abgeändert werden. Semantische Informationen, wie bereits vordefinierte Materialien oder in IW selbstständig definierte Kennungen, sind nicht mehr existent. Ausschließlich eine Element-ID der generischen Familien ist vorhanden. IW-Komponenten der Straße und der Gestaltungselemente sind während des Exports nicht identifiziert worden und werden als allgemeines Modell repräsentiert. Bei der Semantik dieser Elemente verhält es sich wie bei den oben beschriebenen Brückenkomponenten.

Resultierend wird eine erhebliche Einschränkung der BIM-Fähigkeit des Modells festgestellt. Eine weitere Bearbeitung der Brücke ist nur unter großen Zeitaufwand möglich.

5.3 Datenstrukturierung des Modells durch UniClass Klassifizierungen

Um eine Weiterverarbeitung des Modells mithilfe visueller Programmierung zu ermöglichen, ist eine Kategorisierung der Elemente des Typs „Allgemeinen Modelle“ durch ein Klassifizierungssystem nötig. UniClass 2015, als ein allgemeines Klassifizierungssystem, bietet sich durch seine Herstellerneutralität an und wird für den Einteilungsprozess verwendet. UniClass 2015 wird wie folgt definiert:

“Uniclass 2015 is a unified classification for the UK industry covering all construction sectors. It contains consistent tables classifying items of all scale from a facility such as a railway down through to products such as a CCTV camera in a railway station.” (Delany, 2017)

Durch das von National Building Specification entwickelte System können Elemente verschiedener Konstruktionsbereiche verschiedenen Objektklassen zugeordnet werden. Hierfür werden bereichsspezifische Bibliotheken mit Elementen, die durch ein Codierungssystem eindeutig zugeordnet werden, zur Verfügung gestellt. Besonders bei BIM-Modellen mit selbstdefinierten Familien kann durch die Nutzung eines solchen Systems ein verbesserter Datenaustausch erfolgen.

Codes für eine eindeutige Beschreibung von Objekten werden in Bibliotheken für unterschiedliche Bereiche untergliedert. In Abbildung 5.6 ist eine Übersicht der existierenden Bibliotheken dargestellt.

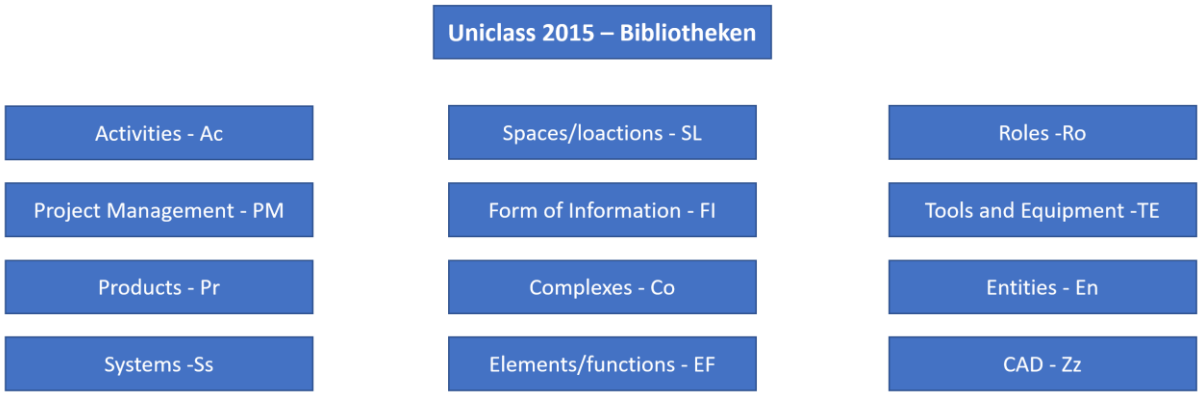


Abbildung 5.6:Übersicht der Uniclass Bibliotheken mit den in Code verwendeten Abkürzungen (NBS, 2019)

Uniclass 2015 wird in mehreren Teilbibliotheken für unterschiedliche Aufgabenbereiche unterteilt. Alle Codes beginnen mit den Abkürzungen der Bibliotheken und lassen sich dadurch leicht zuordnen.

Für die Implementierung wird Fachwissen benötigt, da der passende Klassifikationscode manuell dem Element hinzugefügt werden muss. Projekte mit allgemeinen Elementtypisierungen weisen eine höhere Flexibilität auf und andere Programme können leichter die Modellinformationen adaptieren. In Bezug auf die IfcBridge Erweiterung sind Brückenmodelle anderer Softwareanwendungen dadurch besser in genutzten BIM-Anwendungen zu verarbeiten (Esser et al., 2019).

Komponenten einer Beispielbrücke sind teilweise, wie in Kapitel 5.3 beschrieben, über generische Familien kategorisiert. Elemente der Kategorie „Allgemeine Modelle“ sind mithilfe von UniClasses durch zutreffende Elementbezeichnung beschrieben. Hierfür werden die ID-Daten der Revit-Bauteile verwendet, da ansonsten keine Möglichkeit besteht die Codes auf direkten Weg zu integrieren. In dem Feld Kennzeichnen wird der benötigte Klassifizierungscode manuell eingeben. Damit in der Revit-Datei die Bedeutung des Codes ohne eine Suche in den UniClass Bibliotheken erkenntlich wird, ist die Objektbezeichnung des zugehörigen Codes im Feld Kommentare hinterlegt. Missverständnisse bei falscher Codierung können dadurch aufgelöst werden, dass im Zweifel für das Objekt der richtige Code im Nachtrag ermittelt werden kann. In Abbildung 5.7 werden die fertig definierten Komponenten der Beispielbrücke dargestellt.

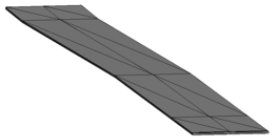
Manuell hinzugefügte Daten	Revit Element	Manuell hinzugefügte Daten	Revit Element
Kommentare: Guardrail Kennzeichen: TE_20_30_30		Kommentare: Street Signs Kennzeichen: Pr_40_10_77_85	
Manuell hinzugefügte Daten	Revit Element	Manuell hinzugefügte Daten	Revit Element
Kommentare: Lamps Kennzeichen: Pr_70_70_46		Kommentare: Asphalt concrete (AC) surface courses Kennzeichen: Pr_35_31_05_05	
Manuell hinzugefügte Daten	Revit Element	Manuell hinzugefügte Daten	Revit Element
Kommentare: Pavement Kennzeichen: EF_30_60		Kommentare: Hatched Road - Marking System Kennzeichen: Ss_40_10_90_36	

Abbildung 5.7: Verwendete Uniclass 2015 Codes zur Beschreibung nicht klassifizierter Brückenkomponenten

Die Elemente der Beispielbrücke sind durch verwendete UniClass 2015 Codes eindeutig definiert. Die Namensbezeichnung der Komponenten sind zusätzlich mitaufgenommen worden.

5.4 Datensortierung mittels visueller Programmierung

5.4.1 Begriffsdefinition und Potential visueller Programmierung für den Sortierungsprozess von BIM-Modellen.

Die für Nutzer angepasste Selektion und Filterung von Daten ist in der Informatik eine bekannte Fragestellung. Durch die Etablierung von BIM im AEC-Sektor entsteht ein digitales Modell des Projekts, welches für verschiedene Fachbereich in Teilmodelle untergliedert wird.

Diese Teilmodelle werden ausschließlich benötigt, um Informationen, zugeschnitten für die Anforderungen verschiedener Bereiche, zu übermitteln. Die Vorgehensweise bewirkt eine erleichterte Bearbeitung von komplexen Projekten. Die Nutzung objekteorientierter Programmiersprachen für die Eingrenzung der benötigten Informationen

überschreitet das Wissen der meisten Planer im Bausektor (Preidel et al., 2017). Visuelles Programmieren ist durch den anwendungsfreundlichen Aufbau eine vielversprechende Lösungsmöglichkeit und wird wie folgt definiert:

„Allgemein wird eine visuelle Sprache als eine formale Sprache mit einer visuellen Syntax und Semantik definiert. Das heißt, es wird ein modulares System aus Zeichen und Regeln durch visuelle, anstatt textuelle Elemente repräsentiert, im semantischen wie syntaktischen Bereich. Durch diese Art der Darstellung kann die visuelle Sprache sehr viel schneller und einfacher durch den Menschen interpretiert werden. Oft werden die visuellen Sprachen auch als flow-based bezeichnet, da sie komplexe Strukturen als Informationsfluss darstellen“ (Ritter et al., 2015)

Revit 2020 besitzt ein Zusatzmodul, welches die Programmiersprache Dynamo in die Software integriert. Mittels Dynamo ist eine Weiterbearbeitung von Revit-Projekten mit visueller Programmierung möglich. Hierfür können vordefinierte Blöcke verwendet werden, die die meisten Parameter und Funktionen der BIM-Anwendung abrufen.

Modellinformationen können ohne zusätzlichen Arbeitsaufwand direkt in eine grafische Programmieroberfläche übernommen werden, wodurch eine Sortierung von Informationen und die Bearbeitung eines Modells möglich wird. Im vorliegenden Fallbeispiel wird Dynamo 2.1 genutzt, um eine Sortierung der Brückenelemente vorzunehmen. Währenddessen wird geprüft, ob alle in den Komponenten enthaltenen Elemente abgerufen werden können. Des Weiteren werden die geometrischen Informationen der Bauteile analysiert, um Erkenntnisse für Darstellungsmöglichkeiten im IFC4X2 Format zu erlangen.

5.4.2 Informationsgewinn anhand des Sortierungsprozesses für das Beispielmodell

Komponenten der *Substructure* des Beispielsmodells sind in Revit in Familienkategorien gegliedert. Somit ist eine direkte Filterung einzelner Komponenten in Dynamo mittels zweier Codeblöcken möglich. Durch weitere Codeblöcke wird im Anschluss die geometrischen Informationen der Elemente untersucht. Die betrachteten Körper werden durch die geometrische Darstellung des Typs Solid repräsentiert. Bei einem Körper des Typs Solid in Revit handelt es sich um eine komplexere Geometrie mit einem

großen Datensatz an Informationen für eine korrekte Darstellung. In Abbildung 5.8 werden die Informationen, die aus einem Solid gewonnen werden, veranschaulicht.

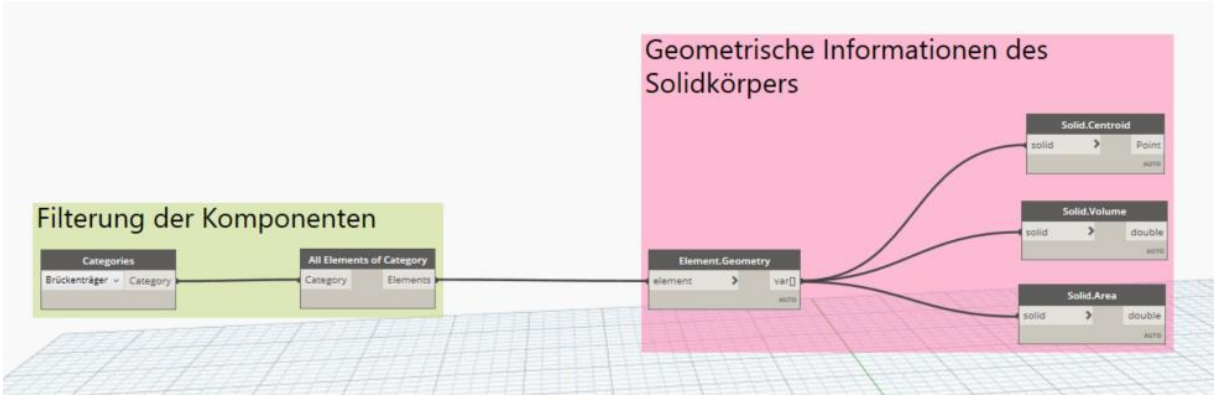


Abbildung 5.8: Filterung von Brückenkomponenten anhand bestehender Kategorien (Links). Grundinformationen der geometrischen Repräsentation (Rechts).

In der Sektion *Filterung der Komponenten* werden alle Bauteilkomponenten der generische Familien Brückenträger abgerufen. Im Anschluss wird die geometrische Repräsentation der Revit-Elemente in Sektion *Geometrische Informationen des Solid-Körpers* geometrische Informationen ermittelt. Durch weitere Codeblöcke werden Mittelpunkt, Volumen und Flächeninhalt berechnet.

Durch den Zugriff auf die Oberflächen der Körper können in Dynamo die Oberflächen der Komponenten als *Polysurfaces* und *Nurbssurfaces* betrachtet werden. Dadurch werden weitere Daten für eine detaillierte Beschreibung der Körper gefunden. In Abbildung 5.9 werden die gewonnen Informationen durch das verwendete Dynamoskript dargestellt. Anhand der gefilterten Informationen ist die Grundlage für die richtige Darstellung und Positionierung der Körper im IFC-Format möglich.

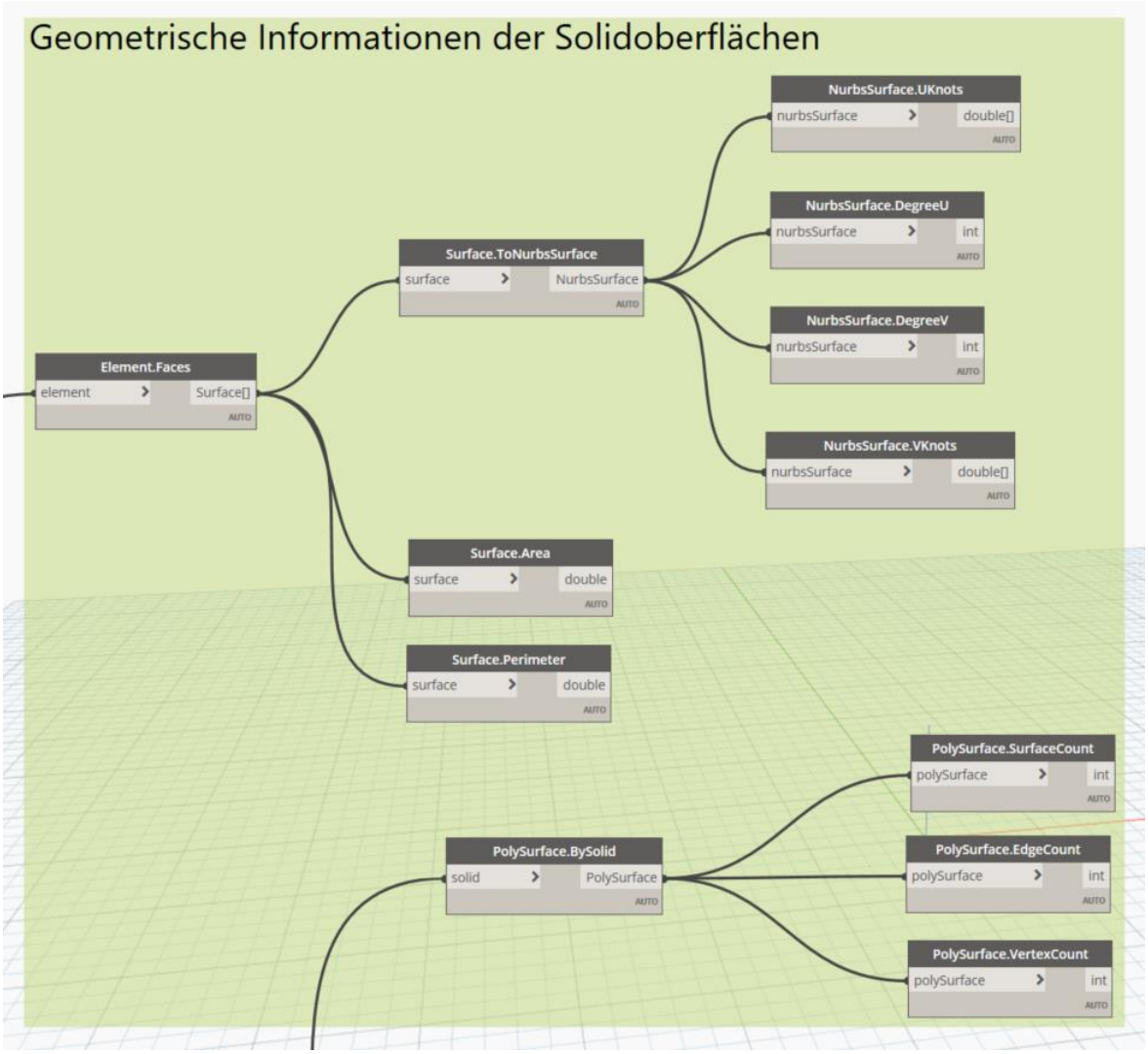


Abbildung 5.9: Detaillierte geometrische Informationen eines Solid-Körpers durch Verwendung von PolySurface und NurbsSurface Codeblöcke

Da es sich bei einem *Polysurface* um einen Solid-Typen handelt, kann dieser direkt übergeben werden. Dadurch werden weitere Informationen über das Bauteilelement gesammelt. Auch die Oberflächengeometrie des Körpers wird betrachtet, dabei wird die geometrische Repräsentation *Nurbssurfaces* genutzt.

Einzelne Elementtypen die in der Kategorie „Allgemeine Geometrie“ zusammengefasst sind, müssen durch ein aufwändigeres Dynamoskript sortiert werden. Dafür werden die nachträglich implementierten Uniclasses, die in den ID-Eigenschaften hinterlegt sind, verwendet. Ein benutzerdefinierter Block wird für diesen Prozess entwickelt, um das Dynamoskript der vollständigen Sortierung der Brückenelemente übersichtlich zu halten. Diese werden in Dynamo verwendet, um einen Teilbereich des Skripts hinter

einem Block zu verbergen. In Abbildung 5.10 wird der benutzerdefinierte Block mit den verwendeten Codeblöcken mit der Darstellung in einem Dynamoskript verglichen.

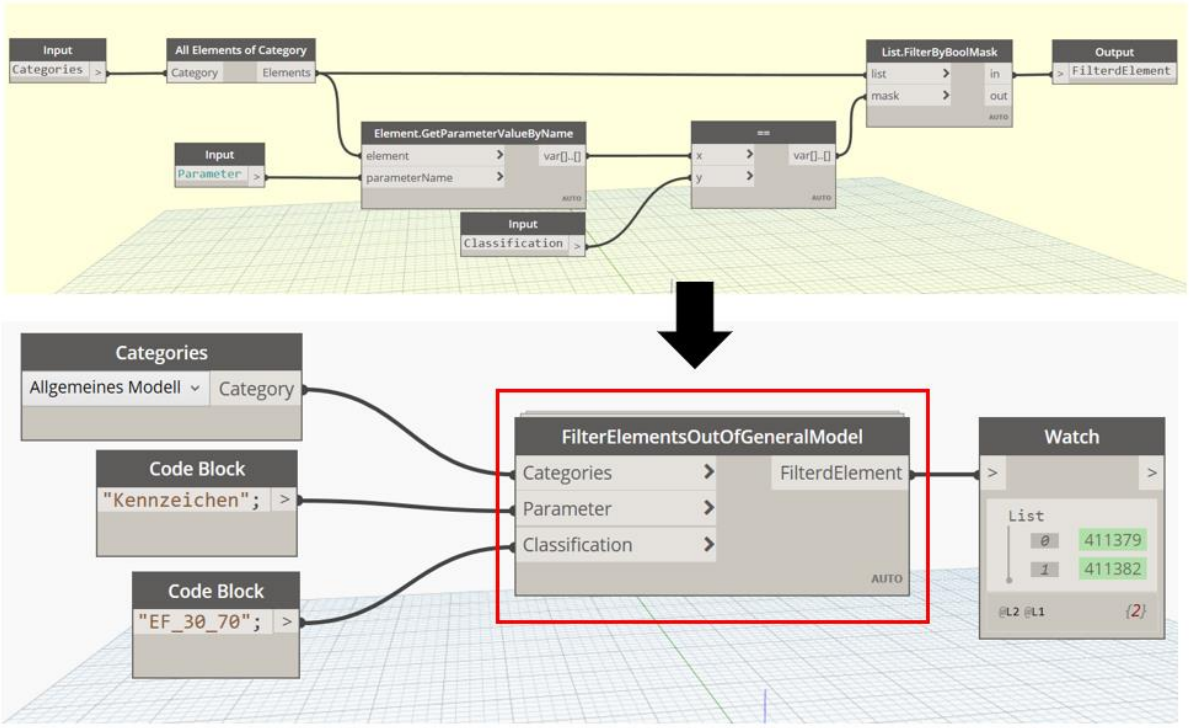


Abbildung 5.10: Benutzerdefinierter Codeblock für Elementfilterung der Allgemeinen Modelle (Oben). Darstellung in einem Dynamoskript (Unten).

Bestimmte Elemente des Allgemeinen Modells werden durch die Klassifizierung mit Uniclass 2015 Codes gefiltert. Der benutzerdefinierte Codeblock besitzt eine Beschreibung der Input- und Outputparameter für ein besseres Verständnis.

Im Unterschied zu den Solid-Körpern handelt es sich bei diesen Elementen um den geometrischen Repräsentationstyp Mesh, der einen Körper durch ein Polygonnetz darstellt. Durch die Anwendung der von Dynamo bereitgestellten Blöcke, können die kartesischen Koordinaten und die Abhängigkeiten untereinander als Informationen über die geometrische Repräsentation gewonnen werden. Dennoch stellt sich die Frage, wie auf diese Informationen in der Revit API zugegriffen werden kann.

6 IfcBridge Toolkit und die Interaktion mit Dynamo

Die aus .Net-basierte Klassenbibliotheken bestehende Applikation IfcBridge Toolkit soll die geometrischen und semantischen Informationen eines BIM-Modells aus einer kommerziell genutzten Software in eine IFC Datei, der Version IFC4X2, übermitteln. Für den Datentransfer wird als Benutzeroberfläche die visuelle Programmiersprache Dynamo verwendet. Die Applikation selbst ist in Microsoft Visual Studio erstellt, unter Nutzung der Hochsprache C-Sharp (C#). Zusätzlich existiert eine .Exe Datei, die durch manuelle Eingabe und Nutzung des Toolkits, ein IfcBridge-Modell erstellen kann.

6.1 Erkenntnisse aus dem Vorplanungsprozess

Anhand der gewonnenen Erkenntnisse aus der Erstellung und Bearbeitung des Beispielmodells, wie in Kapitel 5 beschrieben, können folgende Rückschlüsse für die Implementation der Applikation getroffen werden.

Da das Alignment aus IW nicht in Revit übertragen wird, kann kein *IfcLinearPlacement* für die erfolgreiche Platzierung der Elemente genutzt werden. Ersatzweise muss auf die Platzierungsmethode *IfcLocalPlacement* zurückgegriffen werden, um die Lage der Brückenkomponenten zu definieren. Die dafür benötigten Koordinaten des Körperschwerpunktes eines Elements sind in der Revit Application Programming Interface (API) vorhanden. Dadurch ist die Nutzung der MVD Alignment-based Bridge View nicht für die genierten Beispielbrücken anwendbar. Die Applikation soll dennoch eine Repräsentation der IfcBridge-Datei als RV in einem IFC-Viewer ermöglichen.

Durch die Filterung der geometrischen Informationen von Mesh-Körpern in Dynamo erscheint eine geometrische Beschreibung als *IfcTriangulatedFaceSet* für die Repräsentation im IFC-Schema möglich. Mit dem Revit-Plugin RevitLookup können Eigenschaften und Beziehungen von Elementen in der Revit API betrachtet werden. Die Bauteilelemente mit der geometrischen Repräsentation des Typs Mesh werden mit dem Plugin untersucht, um herauszufinden wie die benötigten Informationen, über die Koordinaten und deren Beziehung untereinander, bei der Benutzung der Applikation abgerufen werden können. Die gewonnen Informationen werden durch die Nutzung von RevitLookup in Abbildung 7.1 dargestellt.

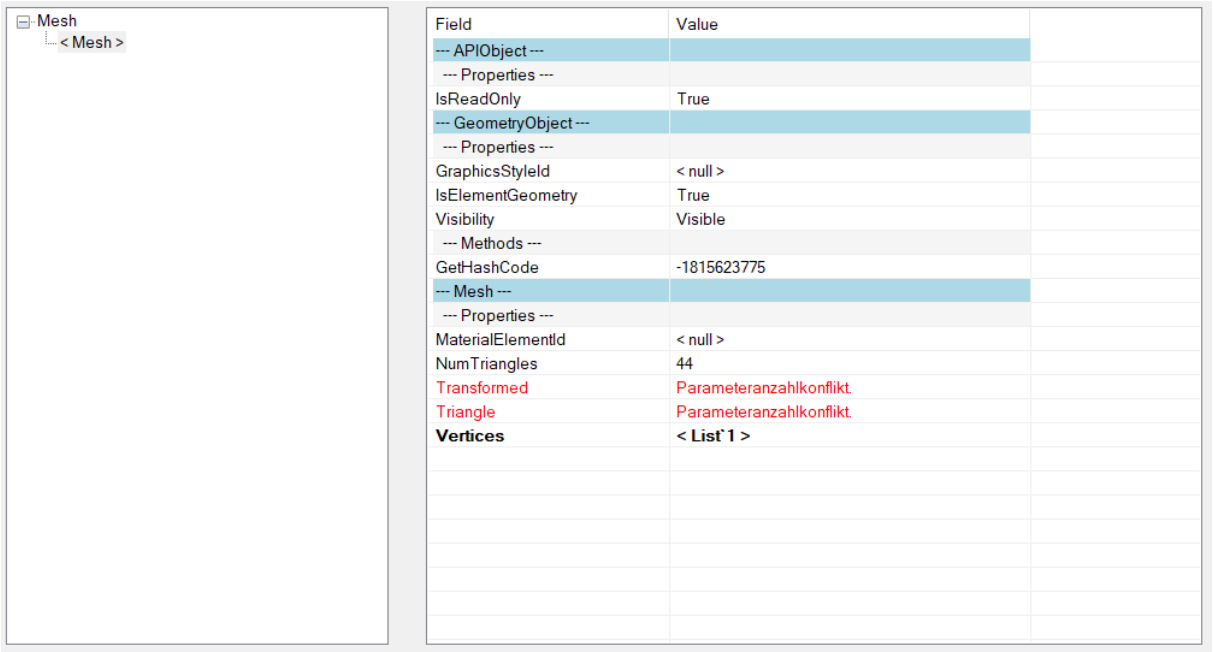


Abbildung 6.1: Geometrische Informationen von Mesh-Geometrien aus der Revit API

Die Anzahl der Dreiecke, die zur Darstellung der Mesh-Geometrie benötigt werden, sind vorhanden. Auch eine Liste der Vektoren der Dreieckspunkte ist abrufbar und enthält die benötigten Koordinaten. Allerdings kann keine Liste an Dreiecksbeziehungen gefunden werden, die für einen Export als *IfcTriangulatedFaceSet* benötigt werden. Wegen dem nicht eindeutigen Ergebnis werden die Brückenkomponenten, die durch Mesh-Geometrien in Revit dargestellt sind, im Rahmen dieser Arbeit für die Erstellung der Applikation ausgeschlossen.

Des Weiteren sollen die geometrische Repräsentation von *IfcBridgeparts*, falls keine alignmentbasierte Beschreibung möglich ist, durch *IfcFacetedBreps* oder *IfcFacetedBrepsWithVoids* dargestellt werden (buildingSMART e.V., 2019b). Da die benötigten Informationen nicht in der vorhandenen Mesh-Geometrie vorhanden sind, kann diese Vorgabe nicht erfüllt werden und verstärkt den getroffenen Entschluss Mesh-Geometrien in der Applikation nicht zu unterstützen.

Brückenkomponenten, deren Geometrie in Revit durch ein Solid beschrieben sind, besitzen genügen Informationen für eine geometrische Beschreibung als *IfcFacetedBrep* im IFC Schema. Dafür muss auf die Oberflächen der Komponenten zugegriffen werden, die in der Revit API als *Faces* für Solid-Körper hinterlegt sind. In Abbildung 7.2 werden die Informationen eines Solid-Körpers des Beispielmodells dargestellt.

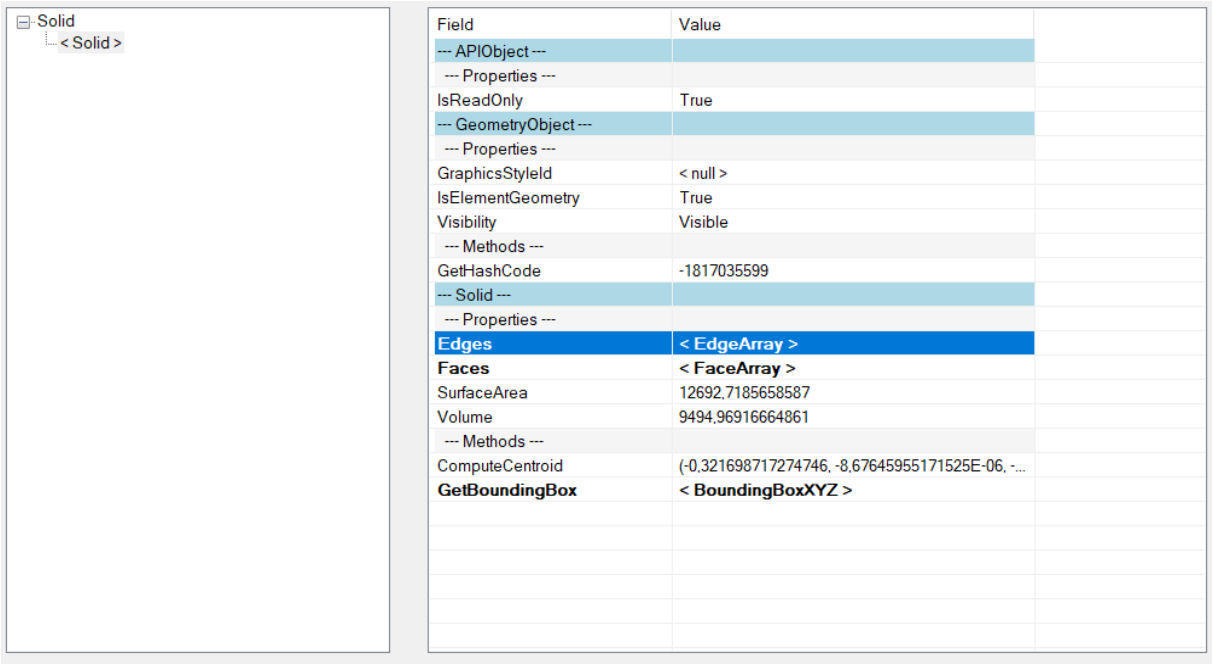


Abbildung 6.2: Geometrische Informationen eines Solid-Körpers in der Revit API

Die Solid-Geometrien besitzen Informationen über *Edges* und *Faces*. Durch eine Lister der Solid-Faces wird die Oberflächengeometrie der Komponenten beschrieben. Diese geometrische Information kann für eine Nutzung in der Applikation aufbereitet werden, da die Eckpunkte der vorhanden Faces für die geometrische Repräsentation *IfcFacetedBrep* im IFC Schema genutzt werden kann.

Da wichtige Komponenten der Brücke, wie beispielsweise Brückenträger, Brückenpfeiler, Fundamente und Auflager, in Revit als Solid-Körper dargestellt werden, kann trotz des Ausschlusses der Mesh-Körper ein aussagekräftiges IfcBridge-Modell erstellt werden.

Wie in Kapitel 5 beschrieben, können die semantischen Informationen aus IW 2020 während des Übermittlungsprozesses nicht in das Revit Modell übertragen werden. Um die Darstellbarkeit semantischer Informationen in einem IfcBridge-Modell aufzuzeigen, wird im Toolkit als Beispiel eine Möglichkeit geschaffen Materialien verschiedener Komponenten zu definieren.

Die durch das verwendete Dynamoskript gewonnenen Informationen, beschränken sich ausschließlich auf die geometrische Beschreibung der Brücke. Um die Minimalanforderungen eines IfcBridge-Modells zu erfüllen, muss zusätzlich die räumliche Hierarchie implementiert sein. Die korrekte Darstellung der geometrischen Daten und die

räumliche Struktur einer IfcBridge sind zwei essenzielle Rahmenbedingungen der entwickelten Applikation.

6.2 Verwendete Softwareanwendungen zur Erstellung der Applikation

Im Folgenden wird auf genutzte Hilfsmittel und Softwareanwendungen der entwickelten Applikation und deren Verwendungszweck eingegangen.

Microsoft Visual Studio 2017

Microsoft Visual Studio (MVS) bietet eine integrierte Entwicklungsumgebung für verschiedene Hochsprachen an. Nutzer können in dieser Umgebung neben nativen Win32/Win64-Programmen und dem .NET Framework eine Vielzahl weiterer Anwendungen entwickeln. Da MVS C# als Programmiersprache zulässt und über NuGet eine Vielzahl an .NET Framework Bibliotheken integrierbar sind, wird das IfcBridge Toolkit in dieser Umgebung programmiert. Die programmierte Applikation beinhaltet mehrere Klassenbibliotheken, die eine Datenübermittlung in das IFC4X2-Schema ermöglicht.

C# als verwendete Hochsprache

Bei der Wahl der Programmiersprachen und externer Bibliotheken ist auf die Kompatibilität mit dem geplanten Projekt zu achten. Die Zwecke der Applikation müssen im Vorfeld genau untersucht werden.

Zur Entwicklung des IfcBridge Toolkits, mit dem Ziel einer Datenumwandlung eines BIM-Modells in das IFC Schema unter Verwendung der visuellen Programmiersprache Dynamo, stehen die beiden Hochsprachen C# und Python zur Auswahl. Diese Programmiersprachen werden von Dynamo zur Erstellung von selbstentwickelten Codeblöcken unterstützt. Obwohl Python-Codes direkt in der Dynamo-Oberfläche geschrieben werden können, wird C# als Programmiersprache bevorzugt, da Bibliotheken des xBIM-Toolkits, die das IFC Schema vollständig abdecken, existieren. Benötigte Funktionen lassen sich besser erstellen, da durch xBIM Funktionalitäten für die Interaktion mit IFC-Modellen bereitgestellt werden. Weitere Vorteile sind die Nutzbarkeit von Debugging Prozessen und Testumgebungen. Des Weiteren wird bei Verwendung der Revit API für benutzerdefinierte Applikationen C# als Programmsprache von Autodesk empfohlen.

xBim-Toolkit

Bei dem xBIM-Toolkit (eXtensible Building Information Modelling Toolkit) handelt es sich um eine .NET Open Source Software, die das IFC Schema unterstützt. Durch die volle Implementierung von geometrischen, topologischen, operativen und visualisierenden IFC Eigenschaften, wird dem Nutzer das Erstellen und Bearbeiten von BIM-Modellen im IFC Schema ermöglicht. Die Funktionsbibliotheken für die Datenmanipulation sind in C# formuliert (xBim Toolkit, 2019).

Für die Implementierung des IfcBridge Toolkits wird die die xBim Toolkit Bibliothek *xBim.ifcRail.dll* genutzt. In dieser Version sind alle nötigten Bestandteile der IfcBridge Erweiterung enthalten. Die genutzte Programmbibliothek kann nicht über NuGet in MVS installiert werden und wird deshalb der Projektmappe als externe Bibliothek hinzugefügt. Über NuGet können durch die Installation von xBim Essentials mehrere Programmbibliotheken in eine Applikation geladen werden, die für das IfcBridge Toolkit benötigt werden.

Dynamo

In Kapitel 5.4.1 wird die der Nutzen visueller Programmiersprachen angeschnitten. Visuelle Programmierung kann für verschiedene Anwendungsbereiche in der Bauindustrie genutzt werden, um die Datenverarbeitung von Personen aus dem AEC-Sektor zu vereinfachen. Durch die nutzerfreundliche Handhabung sind visuell programmierte Skripte im Vergleich zu textbasierten Computersprachen schnell zu erstellen und werden leichter verstanden. Die vorhanden Funktionsbibliotheken können durch benutzerdefinierte Funktionen ergänzt werden (Preidel et al., 2017).

Die erstellten Funktionen der Applikation sollen über Dynamo 2.1 angewendet werden, da diese Sprache direkt als Plugin in Revit geöffnet werden kann. Dynamo ist eine visuelle Programmiersprache, die einen direkten Zugriff auf die Revit API ermöglicht. Dadurch können die erstellten Dynamo-Funktionen der Applikation direkt mit den benötigten Informationen der betrachteten Bauteilkomponenten befüllt werden.

Zero Touch Nodes

Um die entwickelten Funktionen für eine korrekte Übermittlung der Daten in das IFC Format durch Nutzung von Dynamo zu gewährleisten, müssen diese als selbstdefinierte Blöcke abrufbar sein. Ein in C# geschriebener Block wird als Zero Touch Node (ZTN) bezeichnet. Öffentliche Funktionen eines MVS-Projekts können über die in NuGet erhältliche Bibliothek *DynamoVisualProgramming.ZeroTouchLibary* als Codeblöcke in Dynamo dargestellt werden.

6.3 Übersicht und Funktionsweise der Toolkitarchitektur

Das Toolkit, welches für eine erfolgreiche Übertragung des Revit-Modells in das IFC-Schema benötigt wird, folgt objektorientierten Prinzipien und die Struktur ist dabei modular für eine bessere Nutzerinteraktion gestaltet. Abbildung 6.3 zeigt die Applikationsstruktur, die zur Übersetzung des BIM-Modells in das IfcBridge Schema implementiert ist.

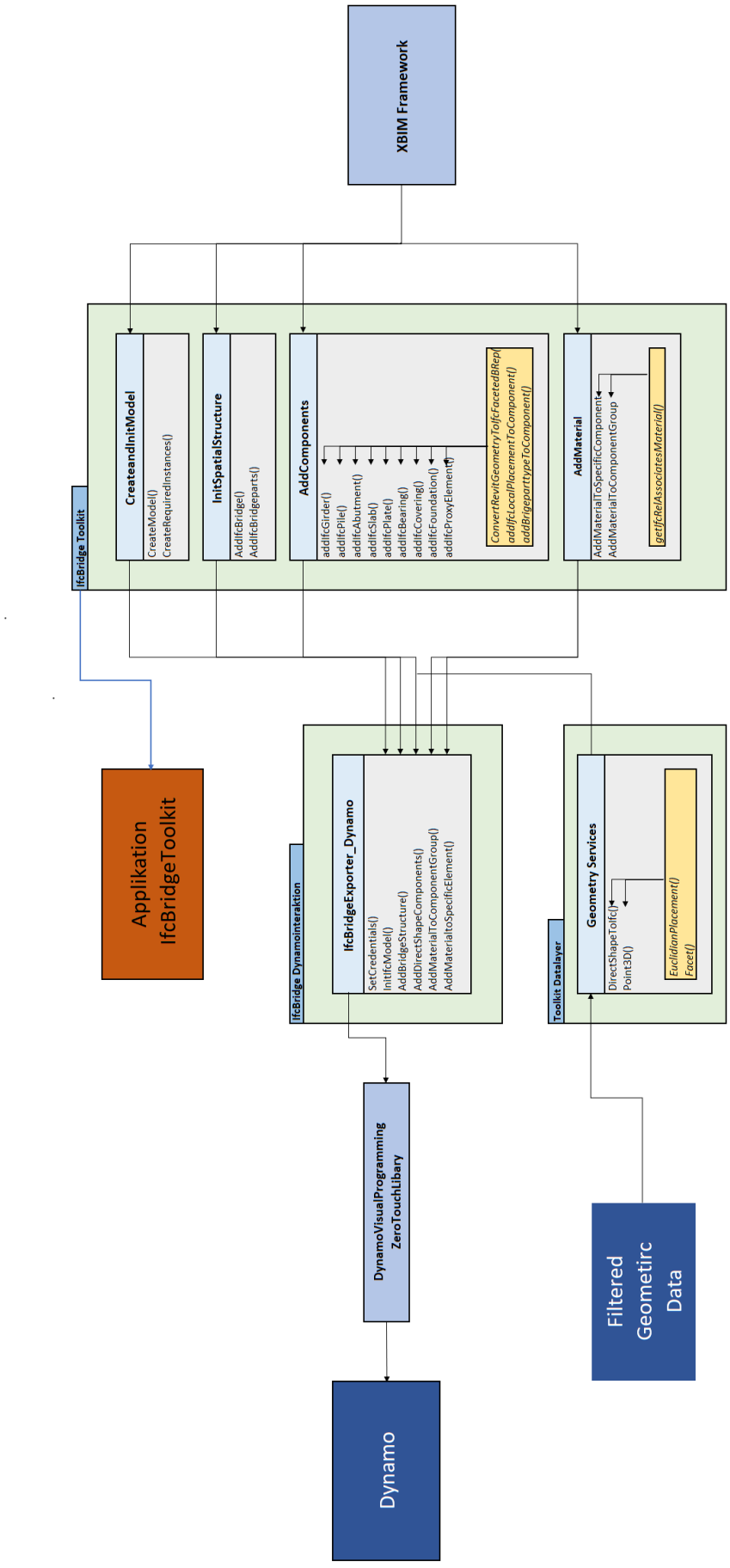


Abbildung 6.3 Aufbau der IfcBridge Toolkitarchitektur

Wie in Abbildung 6.3 dargestellt, wurden auf Grundlage des xBIM Toolkits die Klassenbibliothek IfcBridge Toolkit in der Applikation erstellt. Durch die volle Unterstützung der IFC Datenstruktur können ohne zusätzliche Implementierung alle IFC Befehle als Funktion in einem C# Projekt abgerufen werden.

Des Weiteren ist zu erkennen, dass die Applikation aus zwei Hauptbestandteilen besteht. Zum einem aus dem IfcBridge Toolkit, welches Funktionen für die Generierung IFC typischer Daten zu Verfügung stellt. Zum anderen aus einer Klassenbibliothek, die zur Erstellung von Zero Touch Nodes benötigt wird.

Diese Codeblöcke rufen Funktionen aus dem IfcBridge Toolkit auf, um ein Brückenmodell über Dynamo als IFC Datei zu speichern. Durch die strikte Trennung der allgemeinen und der softwarespezifischen, implementierenden Funktionen, kann eine IfcBridge Datei auch außerhalb der Softwareumgebung von Dynamo implementiert werden.

Die Klassenbibliothek IfcBridge Toolkit ist das Kernelement der Applikation und besteht aus mehreren Klassen. Diese decken unterschiedliche Aufgaben während des Erstellungsprozesses eines IfcBridge-Modells ab. Die Unterteilung ermöglicht einen besseren Überblick über die Schritte, die für eine erfolgreiche Generierung von bestimmten Informationen eines IFC-Modells nötig sind.

Die Funktionen der Klasse *CreateandInitModel()* fungieren als Initiator einer IFC-Datei und die Grundstruktur eines Modells kann hiermit generiert werden. Integriert sind allgemeine Informationen über das Projekt, die in der räumlichen Hierarchie dem *IfcProject* zugeordnet sind. Außerdem wird die *IfcSite* erstellt und mit den benötigten Baustelleninformationen befüllt. *InitSpatialStructure()* wird für eine genauere Unterteilung der räumlichen Hierarchie einer Brücke genutzt.

Bauwerkskomponenten werden durch *AddComponents()* einer IfcBridge Datei hinzugefügt, wobei die Geometrie durch *IfcFacetedBreps* beschrieben wird. Die Positionierung erfolgt über *IfcLocalPlacement*. Durch *IfcRealAggregates* werden die erstellten Elemente in die räumliche Hierarchie eingeordnet (buildingSMART e.V., 2019b).

Mit *AddMaterial()* werden Komponenten auf Materialebene beschrieben, wobei zwischen einzelnen Elementen oder allen Komponenten einer Entität gewählt werden kann.

Während des Implementierungsprozesses der Klassenbibliothek IfcBridge Toolkit wurde darauf geachtet, dass alle Funktionen universell genutzt werden können. Dies ermöglicht Datensätze von anderen Brückenmodellen unterschiedlichster Softwareanbieter zu verarbeiten. Dazu wird eine .EXE Datei in der Microsoft Visual Studio Projektmappe ergänzt, in der, unter Nutzung des IfcBridge Toolkits, IfcBridge Dateien hart codiert werden können.

Ein Beispielmodell einer Brücke soll durch Nutzung der Daten aus Revit direkt als IfcBridge Datei abgespeichert werden, ohne nachträgliche Manipulation des Quellcodes. Dies wird durch die visuelle Programmiersprache Dynamo, mithilfe selbstdefinierter Codeblöcke, ermöglicht. Die Klassenbibliothek *IfcBridgeExporter_Dynamo* wurde für diesen Zweck entwickelt. Die darin definierten Zero Touch Nodes greifen auf die Funktionen des Toolkits zu und durch vordefinierte Inputparameter werden die benötigten Informationen in Dynamo eingegeben.

Mit einem Debugging-Prozess kann in MVS der Generierungsprozess der Daten durch die implementierten Codeblöcke überwacht werden. Fehler in dem erstellten Code führen zu einem Abbruch des Dynamoskriptes und werden in MVS kenntlich gemacht.

Um die geometrischen Daten des Revit-Modells verarbeiten zu können, wird die Klassenbibliothek *Filtered Geometric Data* genutzt. Diese wurde im Rahmen einer Publikation von (Esser et al., 2019) entwickelt und ist in der in der Applikation integriert. Durch die darin implementierten Funktionen können die Informationen des Revit-Modells über die Beschreibung eines Elements für die Erstellung eines *IfcFacetedBreps* genutzt werden.

6.4 Beschreibung und Funktion der Zero Touch Nodes Inhalte

Der folgende Teil der vorliegenden Bachelorarbeit geht näher auf die einzelnen ZTNs ein. Es werden ausschließlich die Aufgaben der Codeblöcke und die Nutzung der verschiedenen Funktionen des IfcBridge Toolkits betrachtet. Die dafür geschriebenen Codes sind in der digitalen Anlage des Anhangs B zu finden.

6.4.1 Credentials()

Der ZTN *Credentials()* enthält Informationen über die Applikation, Toolkitnutzer und der Organisation des Nutzers, welche für die Hierarchie-Ebene *IfcProject* wichtig sind. An einem BIM-Modell arbeiten meistens mehrere Fachplaner verschiedener Gewerke, die Teilmodelle nutzen und ihre erstellten Daten in das Gesamtmodell übergeben.

Alle ZTNs der Applikation besitzen einen Inputparameter *Credentials*, der nur durch *Credentials()* befüllt werden kann. Dadurch ist der Nutzer bei einer Datenübergabe verpflichtet seine persönlichen Daten anzugeben. Hierdurch wird gewährleistet, dass der Planer, der einen bestimmen IFC-Teilabschnitt durch das Toolkit erstellt, ermittelt werden kann.

Des Weiteren kann bei Betrachtung einer IfcBridge Datei schnell herausgefunden werden, ob benötigte geometrische und semantische Informationen eines BIM Modells von den richtigen Gewerken implementiert wurden. Abbildung 6.4 zeigt den Ablaufprozess bei Nutzung des Codeblocks.

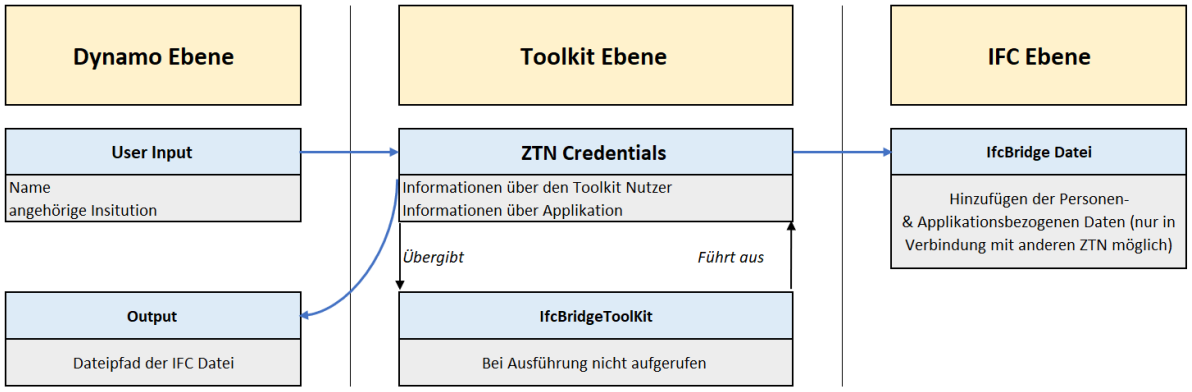


Abbildung 6.4: Ablaufschema der Funktion Credentials()

Im Gegensatz zu den anderen in der Bibliothek enthaltenen Funktionen, ruft *Credentials()* direkt Funktionen des xBIM-Toolkits auf . Diese Vorgehensweise wird als bessere Variante angesehen, da nur die xBIM-Funktion *xbimEditorCredentials*, deren Eigenschaften durch Strings definiert sind, abgerufen wird.

6.4.2 InitIfcModel()

InitIfcModel() wird verwendet, um die Grundstruktur der IFC Datei herzustellen, welche an einem frei wählbaren Ort abgespeichert wird. Zunächst wird die Klasse *CreateandInitModel()* aus dem IfcBridge Toolkit abgerufen. Das Projekt wird dadurch initialisiert und grundlegende Daten werden dem IFC Schema hinzugefügt. Zwei Funktionen finden Anwendung: zum einem *CreateModel()*, das die übergeordneten Informationen, wie beispielsweise Längeneinheiten und das lokale Koordinatensystem, übermittelt. Zum anderen *CreateRequiredInstances()*, welche *IfcSite* in die räumliche Hierarchie einfügt und mit den benötigten Parametern befüllt. Durch die Inputparameter *Filename* und *Directory* kann die Grundstruktur des IFC-Modells an einem beliebigen Ort abgespeichert werden. Der Nutzer besitzt die Option, das Brückenmodell durch weitere Inputparameter zu beschreiben und diese in der IFC Datei abzulegen.

Es muss mit diesem Codeblock in Dynamo begonnen werden, um ein erfolgreiches IfcBridge Modell mit dem Toolkit zu generieren, da ohne allgemeine, projektunabhängige Informationen ein Export in das neutrale Datenschema IFC nicht möglich ist. Abbildung 6.5 stellt schematisch die Schritte zur Erstellung einer IFC4X2-Datei unter Nutzung des IfcBridge Toolkits dar.

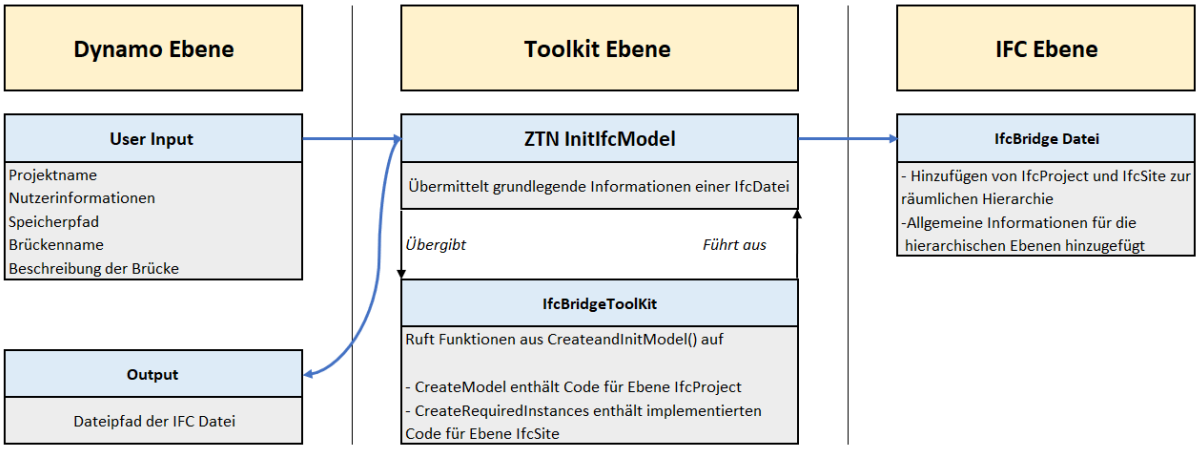


Abbildung 6.5: Ablaufschema bei Nutzung der Funktion InitIfcModel()

6.4.3 AddBridgeStructure()

Um die brückenspezifische räumliche Struktur der IFC Datei hinzuzufügen, ist der ZTN *AddBridgeStructure()* implementiert. Die bereits beschriebenen Codeblöcke *Credentials()* und *InitIfcModel()* beinhalten allgemeine Informationen, die jedes digitale Infrastrukturmodell benötigt. Die räumliche Struktur kann sich im Hinblick auf die, derzeit in Entwicklung befindenden, IFC-Erweiterungen ändern.

In Bezug auf die ifcBridge Erweiterung existiert eine klar definierte Zuordnung der Komponenten zu den unterschiedlichen *IfcBridgeparts* (buildingSMART e.V., 2019b). Bei anderen Erweiterungen, wie beispielweise IfcRail, gibt es im Moment noch keine Gewissheit, ob die räumliche Struktur, wie sie in IfcBridge beschrieben wird, unverändert bleibt.

Der zusätzliche Codeblock *AddBridgeStructure()* ermöglicht eine Trennung zwischen allgemeiner räumlicher Hierarchie und erweiterungsspezifischer hierarchischer Ebenen. Die Applikation kann für künftige Projekt durch weitere Codeblöcke, die die räumliche Struktur der gewünschten Erweiterung enthält, schnell ergänzt werden. Diese Flexibilität hilft, weitere Ergänzungen des IFC Schemas mit Fokus auf Infrastruktur unmittelbar in das IfcBridge Toolkit zu integrieren.

Für die Minimalanforderung der *IfcSpatialStructure* eines IfcBridge Modells wird die unterste Hierarchie Ebene *IfcBridge* hinzugefügt. Des Weiteren werden drei wichtige Haupttypen der Ebene *IfcBridgepart* zur genaueren Unterteilung von Komponenten verwendet. Bei diesen handelt es sich um *Substructure*, *Superstructure* und *Surfacestructure*, da damit eine allgemeine Zuordnung der Elemente in bestimmte Brückenbereiche erreicht werden kann. Andere *IfcBridgePartTypeEnums* sind nicht implementiert. Der Ablaufprozess bei der Verwendung des Toolkits, um die räumliche Hierarchie einer Brücke in die IFC-Datei hinzuzufügen, wird in Abbildung 6.6 gezeigt.

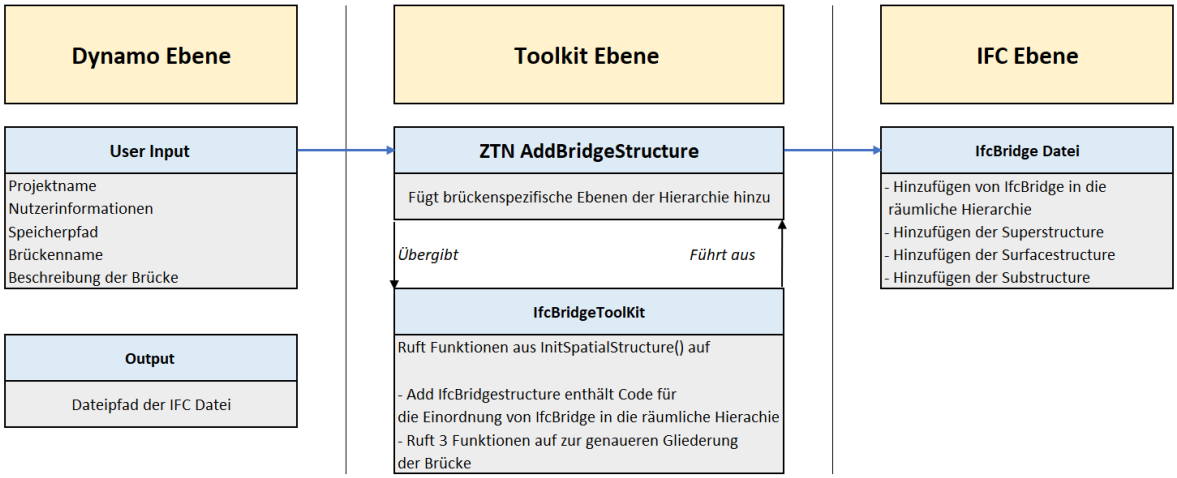


Abbildung 6.6: Ablaufschema der Funktion AddBridgeStructure() für die Erstellung von IFC-Daten

6.4.4 AddDirectShapeComponents()

Nachdem durch die vorherigen Codeblöcke die IFC Struktur eines IfcBridge-Modells erstellt wurden, können mittels *AddDirectShapeComponents()* Komponenten in die Struktur eingefügt werden. Über die separate Klassenbibliothek *Geometry Services* wird aus der Revit-API die Geometrie aufbereitet, sodass auf die Oberflächenbeschreibung und die lokale Platzierung der Revit Bauteile zugegriffen werden kann. Durch Inputparameter wählt der Nutzer Revit-Komponenten aus und gibt die gewünschte Entität manuell ein. Über ein *SwitchCase* in der ZTN-Funktion ruft man die Funktionen zur Erstellung verschiedener Entitäten aus dem Toolkit ab. Die Elemente werden stückweise in die IfcBridge Datei übertragen und enthalten Informationen über die geometrische Beschreibung, Platzierung und Zuordnung in die erstellten *IfcBridgeparts*. Der schematische Ablauf für die Übergabe von Brückenkomponenten in das IFC-Schema ist in Abbildung 6.7 dargestellt.

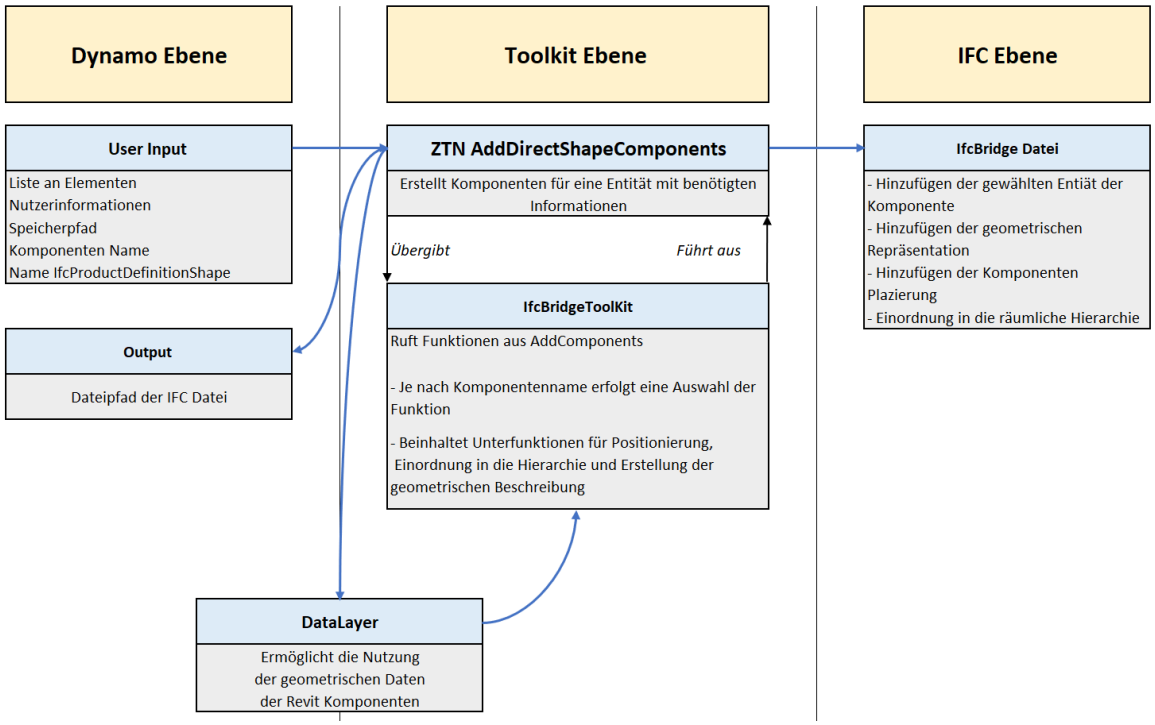


Abbildung 6.7: Ablaufschema der Funktion AddDirectShapeComponents() für die Erstellung von IFC-Daten

6.4.5 Materialzuordnung

Da das Revit-Modell der Brücke nicht die semantischen Informationen aus IW übernimmt, besteht keine Möglichkeit diese an das IfcBridge-Modell zu übergeben. Um eine nachträgliche manuelle Definition der Komponenten auf Materialebene zu ermöglichen, sind zwei Codeblöcke, *AddMaterialToComponentGroup()* und *AddMaterialToSpecificComponent()*, in die Applikation implementiert. Zur Ausführung des Vorgangs nutzen beide Funktionen Informationen aus einer bereits erstellten IfcBridge Datei.

AddMaterialToComponentGroup() ermöglicht die Zuordnung eines Materials für alle Komponenten einer Entität. Der Nutzer kann durch die Inputparameter *Materialname* und *Materialdefinition* das benötigte Material beschreiben. Durch das Abrufen einer dafür implementierten Funktion, hilft ein *SwitchCase* die richtige Entität innerhalb der IFC Datei anzusteuern. Der Nutzer wählt mit Hilfe des Inputparameter *ElementType* die benötigte Klasse selbst aus.

Der Codeblock *AddMaterialToSpecificComponent()* erkennt einzelne Komponenten anhand der ID-Nummer in einer IFC Datei. Die Möglichkeit einzelne Komponenten auf Materialebene zu definieren, wird beispielsweise für Sonderkonstruktionen mit anderen Materialansprüchen verwendet. Die Materialdefinition in der IfcBridge Datei durch

Nutzung der Funktionen aus dem Toolkit wird in Abbildung 6.8 schematisch dargestellt.

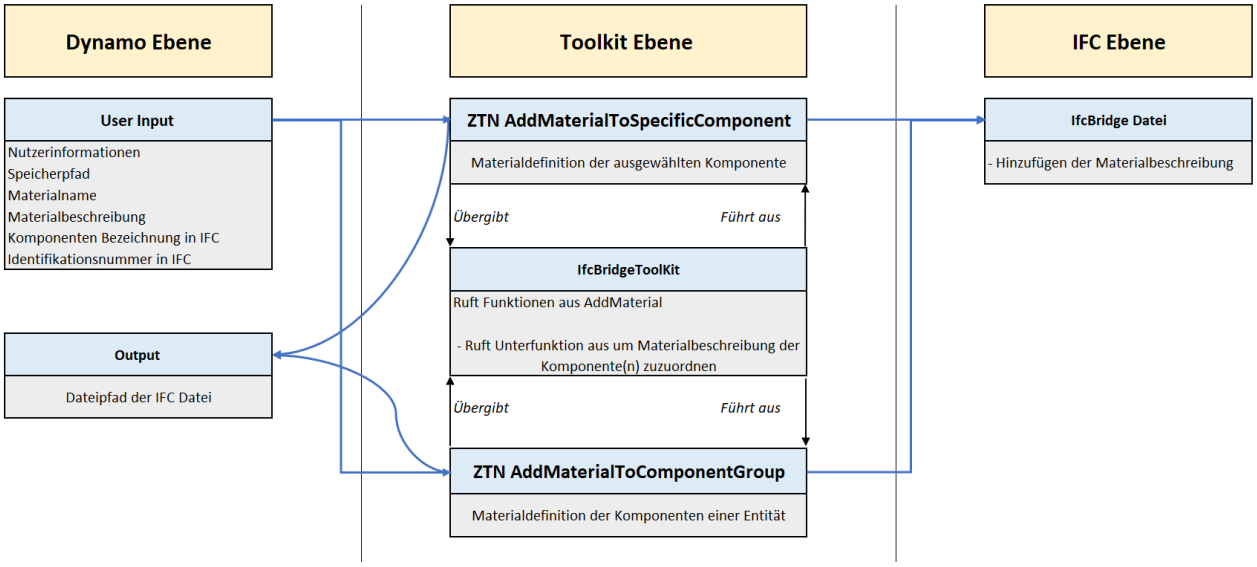


Abbildung 6.8: Ablaufschema der Materialdefinition von Komponenten unter Nutzung des IfcBridge Toolkits

7 IfcBridge Export unter Verwendung des IfcBridge Toolkits

In diesem Kapitel wird die erstellte Applikation IfcBridge Toolkit dahingehend getestet, ob eine fehlerfreie Erstellung einer IfcBridge Datei aus einem Revit-Modell möglich ist. Bei dem verwendeten Modell handelt es sich um ein Beispielmodell, das anhand der in Kapitel 5 beschriebenen Prinzipien erstellt wurde. Abbildung 7.1 zeigt eine Gegenüberstellung der verwendeten Beispielbrücke aus IW 2020 und Revit 2020.

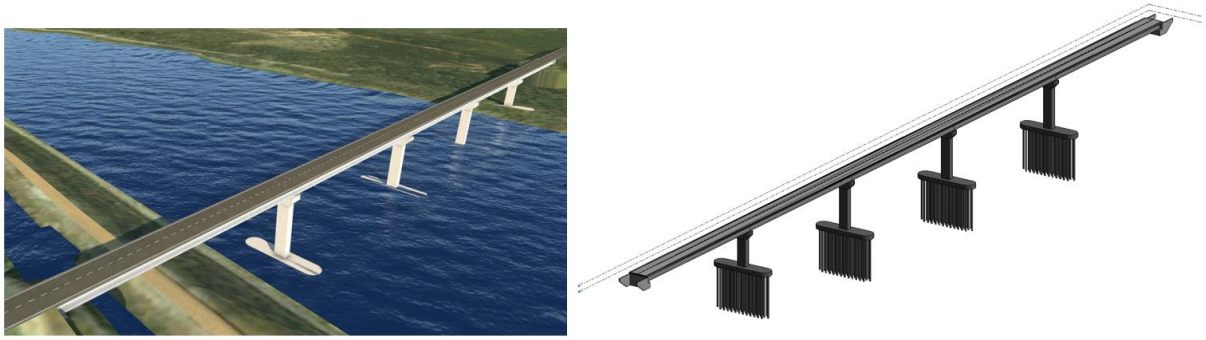


Abbildung 7.1: Darstellung der Beispielbrücke in IW 2020 (Links). Darstellung der Brücke in Revit 2020 (Rechts)

Es werden die Brückendarstellung vor und nach der Datenübergabe zu Revit gezeigt. Die geometrische Darstellung der Bauteile ist unverändert. Für eine bessere Veranschaulichung werden die Repräsentationen der gezeigten Brücken in Anhang A angefügt.

Zunächst wird der Generierungsprozess der IFC-Datei, der durch die in Dynamo hinzugefügten Zero Touch Nodes möglich wird, genauer betrachtet. Im Anschluss wird die erstellte IFC Datei in den IFC Viewern FZK-Viewer (Karlsruher Institut für Technologie, 2019) und BimVision (BimVision, 2019) auf Richtigkeit geprüft.

7.1 Erläuterung der Erstellungsschritte

Im digitalen Anhang kann das gesamte Dynamoskript betrachtet werden, welches für eine vollständige Datenübertragung der Brückenkomponenten benötigt wird. Im folgenden Abschnitt wird auf die verwendeten Codeblöcke des Skriptes eingegangen und die daraus generierten IFC Teilelemente näher beleuchtet. Um das Toolkit nutzen zu

können, muss zuerst der Codeblock *InitIfcModel()*verwendet werden. In Abbildung 7.2 wird dieser Codeblock mit den benötigten Inputparametern dargestellt.

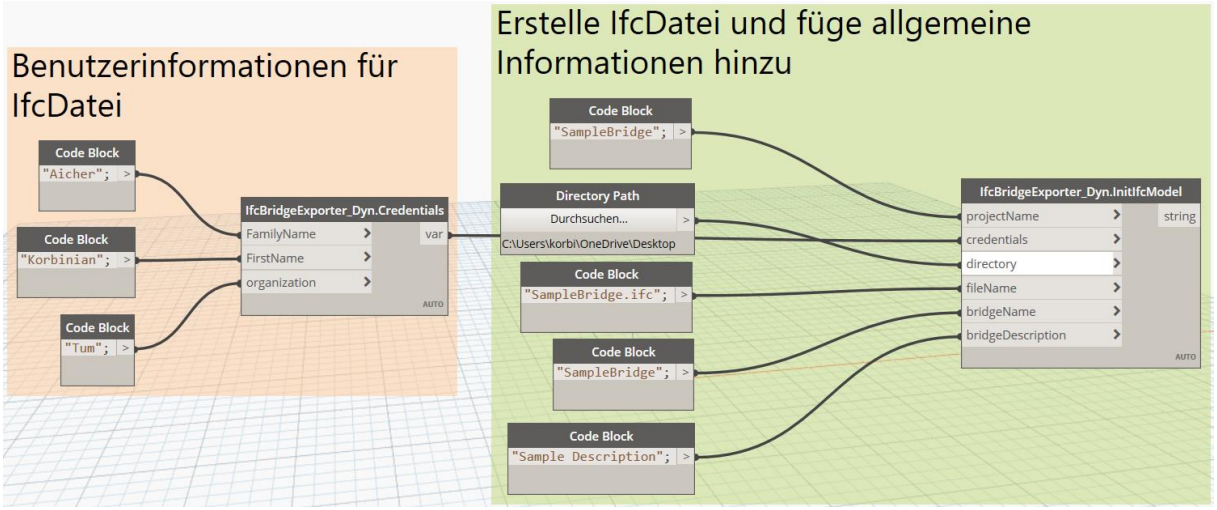


Abbildung 7.2: Credentials() befüllt mir Inputparametern in Dynamooberfläche (Links). InitIfcModel() befüllt mit benötigten Inputparametern (Rechts).

Der Codeblock wird, wie in Kapitel 6 beschrieben, für jeden weiteren Codeblock des Toolkits benötigt. In diesem Schritt wird eine IFC Datei in dem angegeben Pfad generiert und es werden grundlegende Informationen in dieser Datei gespeichtert. In Abbildung 7.3 sind, die vom Codeblock generierten, Informationen abbgebildet. Das IFC Schema wurde in der Datei bereits zu IFC4X2 abgeändert. Durch die Verwendung *xBIM.ifcRail.dll* des xBim Toolkits wird ein anderes Schema, das nicht in einem Viewer gelesen werden kann, übermittelt.

```
ISO-10303-21;
HEADER;
FILE_DESCRIPTION (('''), '2;1');
FILE_NAME ('C:\\Users\\korbi\\OneDrive\\Desktop\\test03.ifc', '2019-07-29T08:22:59', (''), (''), 'Pro
FILE_SCHEMA (('IFC4X2'));
ENDSEC;
DATA;
#1=IFCPROJECT('18NWuIc9H0ZwhWRZnsBWv0',#2,'SampleBridge',$,$,$,$,$,#7);
#2=IFCOWNERHISTORY(#5,#6,$,.ADDED.,1564388581,$,$,0);
#3=IFCPERSON($,'Aicher','Korbinian',$,$,$,$,$);
#4=IFCORGANIZATION($,'Tum',$,$,$);
#5=IFCPERSONANDORGANIZATION(#3,#4,$);
#6=IFCAPPLICATION(#4,'1.0','TUM_CMS_IfcBridgeToolkit',$);
#7=IFCUNITASSIGNMENT((#8,#9,#10,#11));
#8=IFCSIUNIT(*,.LENGTHUNIT.,$,.METRE.);
#9=IFCSIUNIT(*,.PLANEANGLEUNIT.,$,.RADIAN.);
#10=IFCSIUNIT(*,.AREAUNIT.,$,.SQUARE_METRE.);
#11=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
#12=IFCLOCALPLACEMENT($,#13);
#13=IFCAXIS2PLACEMENT3D(#14,$,$);
#14=IFCCARTESIANPOINT((0.,0.,0.));
#15=IFCGEOMETRICREPRESENTATIONCONTEXT($,'Model',3,$,#13,$);
#16=IFCSITE('3vhoOdtdv9ZOfBPNx480ay',#2,'BridgeSite','SiteDescription',$,#12,$,$,$,$,$,0.,$,$);
#17=IFCRELAGGREGATES('3nqSGaDuv9Z8NgFOH2OfG9',#2,$,$,#1,(#16));
```

Abbildung 7.3: Quellcode des IFC-Headers und grundlegenden Projektinformationen

Im nächsten Schritt wird die räumliche Hierarchie einer Brücke hinzugefügt. Im Gegensatz zur Version IFC4X1, ist eine brückenspezifische *IfcSpatialStructure* in der IfcBridge Erweiterung IFC4X2 definiert. In Abbildung 7.4 ist der genutzte Codeblock dargestellt. Der Inputparameter *Credentials* wird bei der Erstellung der Beispieldatei durch denselben Codeblock dargestellt, welcher in Abbildung 7.2 beschrieben ist. Da der Dateipfad der IFC Datei bei jedem Block als Output Parameter festgelegt ist, kann der Inputparameter *StoreFilePath* bei jedem Codeblock durch den vorherigen Block beschrieben werden.

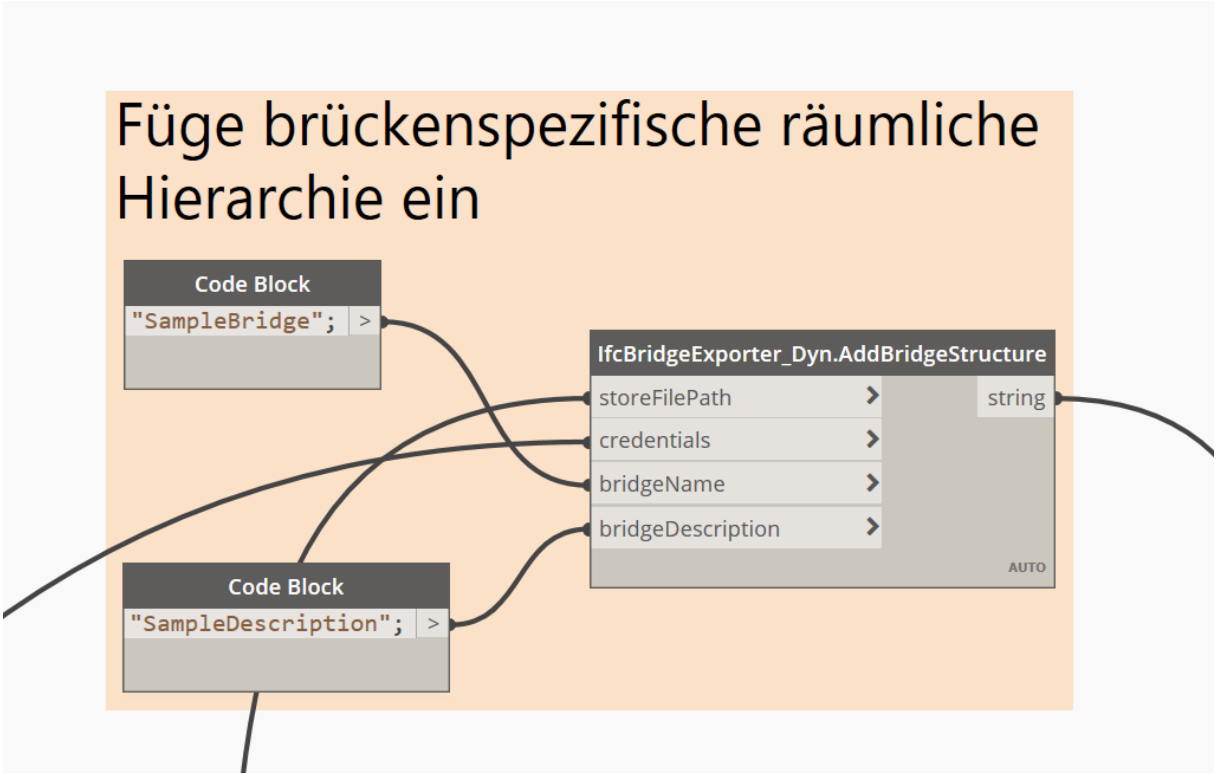


Abbildung 7.4: Mit Inputparametern befüllter Codeblock AddBridgeStructure() in Dynamo Oberfläche

In Abbildung 7.5 ist sind die hinzugefügten Zeilen der IFC Datei zu sehen. Durch die erneute Personen- und Applikationsbeschreibung wird deutlich, dass ein anderer Codeblock verwendet wird, um den Abschnitt zu erstellen.

```
#18=IFCBRIDGE('2WYf5zRLH9w98BJoe3MSlx',#19,'SampleBridge','SampleDescription',$,#23,$,$,$,$);
#19=IFCOWNERHISTORY(#21,#22,$,.ADDED.,1564388581,$,$,0);
#20=IFCPERSON($,'Aicher','Korbinian',$,$,$,$,$);
#21=IFCPERSONANDORGANIZATION(#20,#4,$);
#22=IFCAPPLICATION(#4,'1.0','TUM_CMS_IfcBridgeToolkit',$);
#23=IFCLOCALPLACEMENT($,#24);
#24=IFCAXIS2PLACEMENT3D(#25,#26,#27);
#25=IFCCARTESIANPOINT((0.,0.,0.));
#26=IFCDIRECTION((0.,0.,1.));
#27=IFCDIRECTION((1.,0.,0.));
#28=IFCRELAGGREGATES('2Uzt3gvYz8euNJHa8vxwst',#19,$,$,#16,(#18));
#29=IFCBRIDGEPART('1A6mGl3On21xM0iamJH9Ch',#19,'Superstructure',$,$,#30,$,$,.ELEMENT.,.SUPERSTRUCTURE.);
#30=IFCLOCALPLACEMENT($,#31);
#31=IFCAXIS2PLACEMENT3D(#32,#33,#34);
#32=IFCCARTESIANPOINT((0.,0.,0.));
#33=IFCDIRECTION((0.,0.,1.));
#34=IFCDIRECTION((1.,0.,0.));
#35=IFCBRIDGEPART('3BUrb6JTL9MfWSf7KmhKMK',#19,'Substructure',$,$,#36,$,$,.ELEMENT.,.SUBSTRUCTURE.);
#36=IFCLOCALPLACEMENT($,#37);
#37=IFCAXIS2PLACEMENT3D(#38,#39,#40);
#38=IFCCARTESIANPOINT((0.,0.,0.));
#39=IFCDIRECTION((0.,0.,1.));
#40=IFCDIRECTION((1.,0.,0.));
#41=IFCBRIDGEPART('3b0h5foqDBHfZy4E5rd6yK',#19,'Surfacestructure',$,$,#42,$,$,.ELEMENT.,.SURFACESTRUCTURE.);
#42=IFCLOCALPLACEMENT($,#43);
#43=IFCAXIS2PLACEMENT3D(#44,#45,#46);
#44=IFCCARTESIANPOINT((0.,0.,0.));
#45=IFCDIRECTION((0.,0.,1.));
#46=IFCDIRECTION((1.,0.,0.));
#47=IFCRELAGGREGATES('0qAFDz5VXFHRs3GkE9xpTw',#19,$,$,#18,(#29,#35,#41));
```

Abbildung 7.5: IFC-Quellcode der hinzugefügten räumlichen Hierarchie für Brücken

Nachdem die räumliche Hierarchie feststeht, werden die Revit Komponenten in das IFC Format übergeben. In Abbildung 7.6 ist der dafür benötigte Codeblock dargestellt.

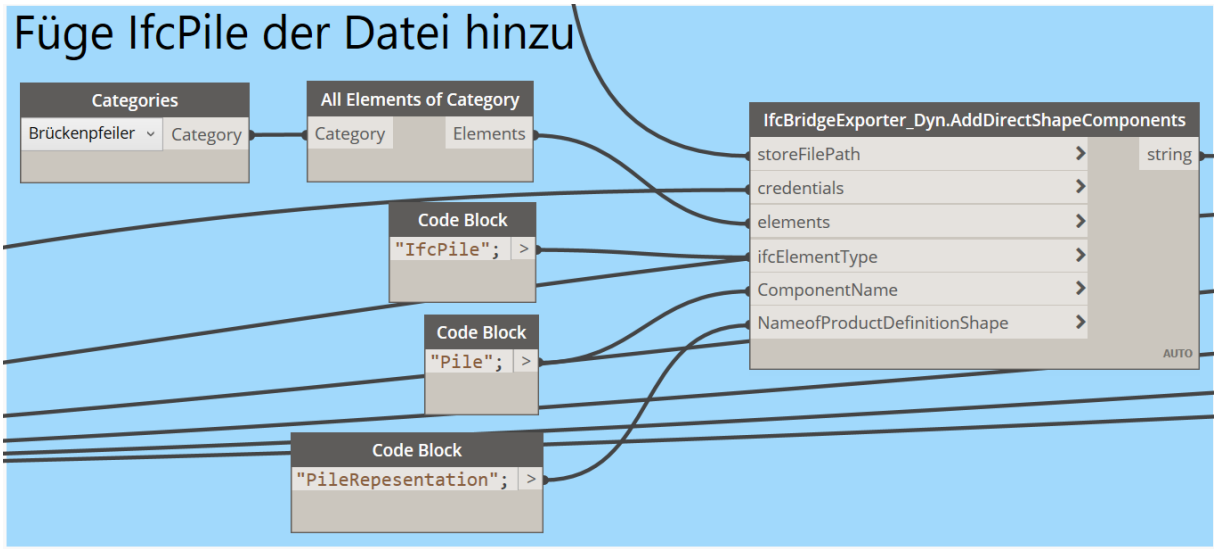


Abbildung 7.6: Anwendung von AddDirectShapeComponents zur Erstellung von IFC-Daten

Alle betrachteten Komponenten können durch diesen Codeblock in die IfcBridge Datei übertragen werden. Der Inputparameter *ifcElementType* ordnet die Bauteile in die zugehörige Entität ein. Die Beispieldatei benutzt die Entitäten *IfcBeam*, *IfcPile*, *IfcSlab*,

IfcBearing, *IfcFoundation* und *IfcBearing* zur Klassifizierung der Elemente. In Abbildung 7.7 sind Abschnitte der übergebenen, in der Datei gespeicherten, Informationen eines *IfcBeams* dargestellt.

```
#48=IFCBEAM('0o6vvPTp9DBO_7p2lz5QU$',#49,'Girder0',$,$,#53,#175,$,$);
#49=IFCOWNERHISTORY(#51,#52,$,.ADDED.,1564388842,$,$,0);
#50=IFCPERSON($,'Aicher','Korbinian',$,$,$,$,$);
#51=IFCPERSONANDORGANIZATION(#50,#4,$);
#52=IFCAPPLICATION(#4,'1.0','TUM_CMS_IfcBridgeToolkit',$);
#53=IFCLOCALPLACEMENT($,#54);
#54=IFCAXIS2PLACEMENT3D(#55,#56,#57);
#55=IFCCARTESIANPOINT((0.,0.,0.));
#56=IFCDIRECTION((0.,0.,1.));
#57=IFCDIRECTION((1.,0.,0.));
#58=IFCFACETEDBREP(#59);
#59=IFCCLOSEDSHELL((#66,#73,#80,#87,#94,#101,#108,#115,#122,#129,#136,#143,#158,#173));

#174=IFCSHAPEREPRESENTATION(#15,'Body','Brep',(#58));
#175=IFCPRODUCTDEFINITIONSHAPE('GirderRepresentation',$,(#174));
#176=IFCRELAGGREGATES('3BYpiZOPj7OwFFvvNQ5y0s',#49,$,$,#29,(#48));
```

Abbildung 7.7: IFC-Quellcode der wichtigsten Bauteilinformationen

Die benötigten Parameter für I*fcClosedShell* sind nicht in der Abbildung zu sehen, da dazu eine Vielzahl an *IfcFaces* erforderlich sind.

Nach dem Hinzufügen der Komponenten kann das erstellte IfcBridge-Modell in einem Viewer betrachtet werden. Zusätzlich kann durch die implementierten Codeblöcke *AddMaterialToSpecificElements()* und *AddMaterialToComponentGroup()* die Semantik des Modells um Materialparameter ergänzt werden. Wie in Kapitel 6.4.5 beschrieben, besteht die Möglichkeit alle Elemente einer Entität gleichzeitig auf der Materialebene zu definiert oder nur einzelne, indem die Datei interne Element-ID abruft. Abbildung 7.8 zeigt die beiden Möglichkeiten, um Materialien zu übergeben.

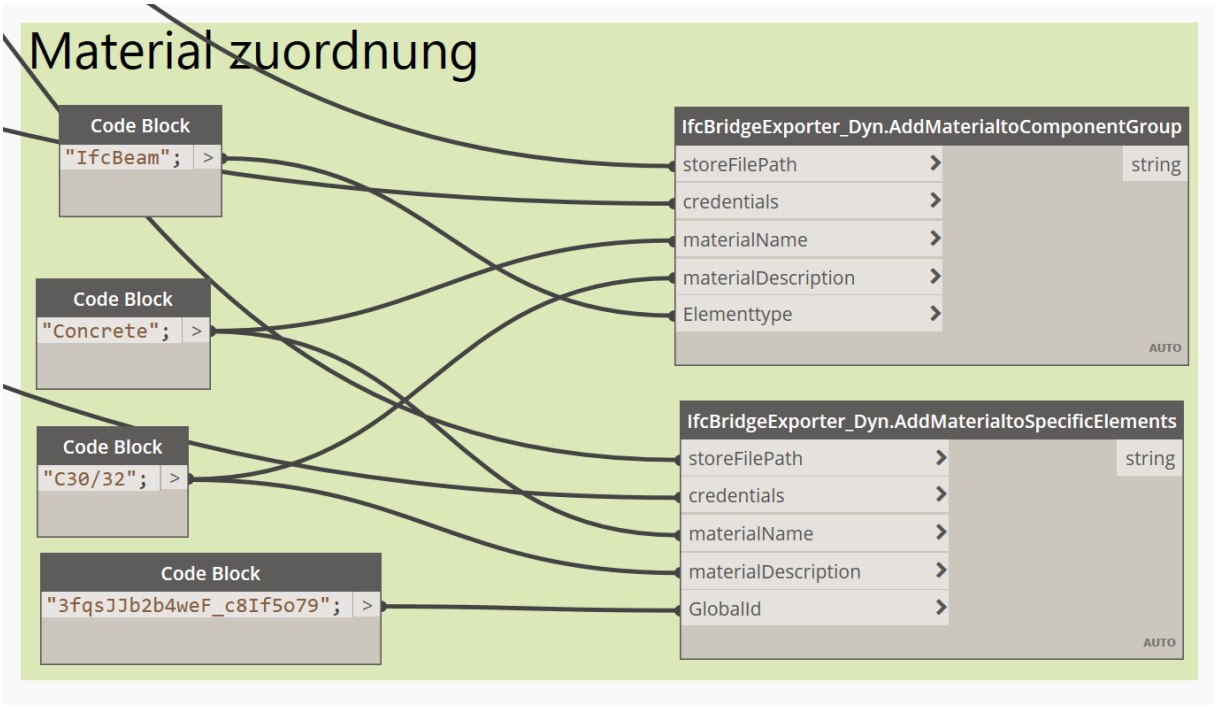


Abbildung 7.8: Materialzuordnung einer ganzen Komponentenfamilie (Oben). Materialzuordnung eines ausgewählten Bauteils (Unten).

7.2 Analyse der IfcBridge Datei unter Verwendung von FZK-Viewer und BimVision

Um die erstellte IFC Datei zu prüfen, wird das Ergebnis in zwei verschiedenen IFC Viewer betrachtet. Zielsetzung ist eine aussagekräftige Repräsentation der übergeben Bauteilelemente und eine korrekte Darstellung der räumlichen Hierarchie. Genutzt werden die Viewer FZK-Viewer 5.1 und BimVision 2.21, da diese IFC Daten der Version IFC 4X2 verarbeiten können.

7.2.1 Begutachtung des Modells durch BimVision

Die Darstellung der exportierten Brücke in dem Viewer BimVision wird in Abbildung 7.8 präsentiert.

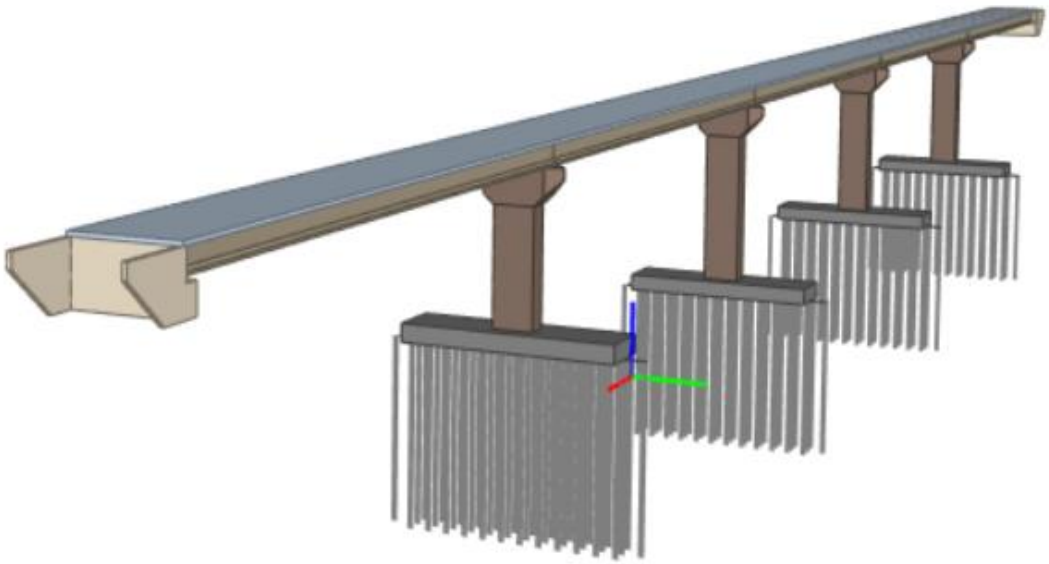


Abbildung 7.9: Geometrische Repräsentation des erstellten IfcBridge-Modells in BimVisio

Wie in Abbildung 7.8 erkennbar ist, wird die Datei ohne Fehlerreport im Viewer dargestellt, sodass man von einer erfolgreichen Anwendung des IfcBridge Toolkits für den Export eines Revit-Modells sprechen kann. Bei genauerer Betrachtung der geometrischen Darstellung werden Unterschiede bei Bohrpfählen und Stützen im Vergleich zum ursprünglichen Modell sichtbar. In Abbildung 7.9 stehen die Darstellungsmethoden dieser Komponenten aus Revit, dem des Viewers gegenüber.

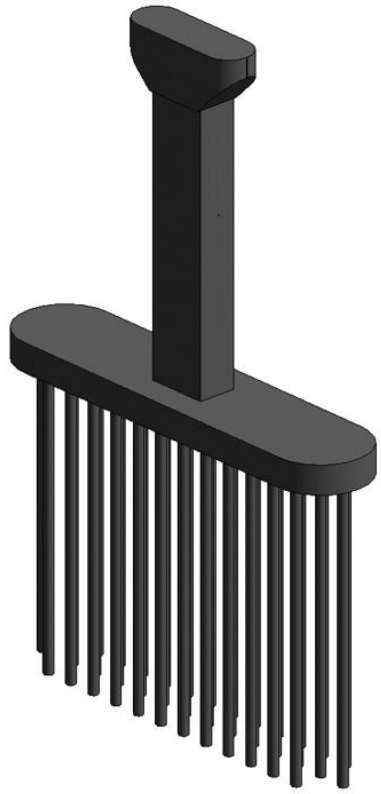
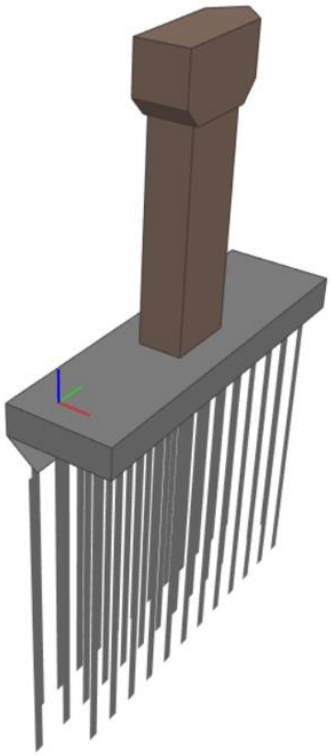
Darstellung in Autodesk Revit	Darstellung in BimVisio
	

Abbildung 7.10: Fehlerhafte geometrische Darstellung des Brückenpfeilers und Fundament Bohrpfähle

Der Verlust der genauen geometrischen Beschreibung ergibt sich aus der Nutzung der Darstellungsmethode *IfcFacetedBrep*. Diese Methode beschreibt Volumenkörper über Teilflächen, die durch einen Polygonzug definiert sind. Es wird vermutet, dass die Geometrie in Revit anders beschrieben wird. Wahrscheinlich existieren nicht genügen kartesische Koordinaten, um die Geometrie mit *IfcFacetedBrep* besser zu beschreiben.

7.2.2 Begutachtung des Modells durch FZK-Viewer

In Abbildung 7.10 ist die Darstellung der identischen Brücke aus dem FZK-Viewer abgebildet. Die räumliche Hierarchie der Datei wird, wie in dem Bild zu sehen, korrekt angezeigt.

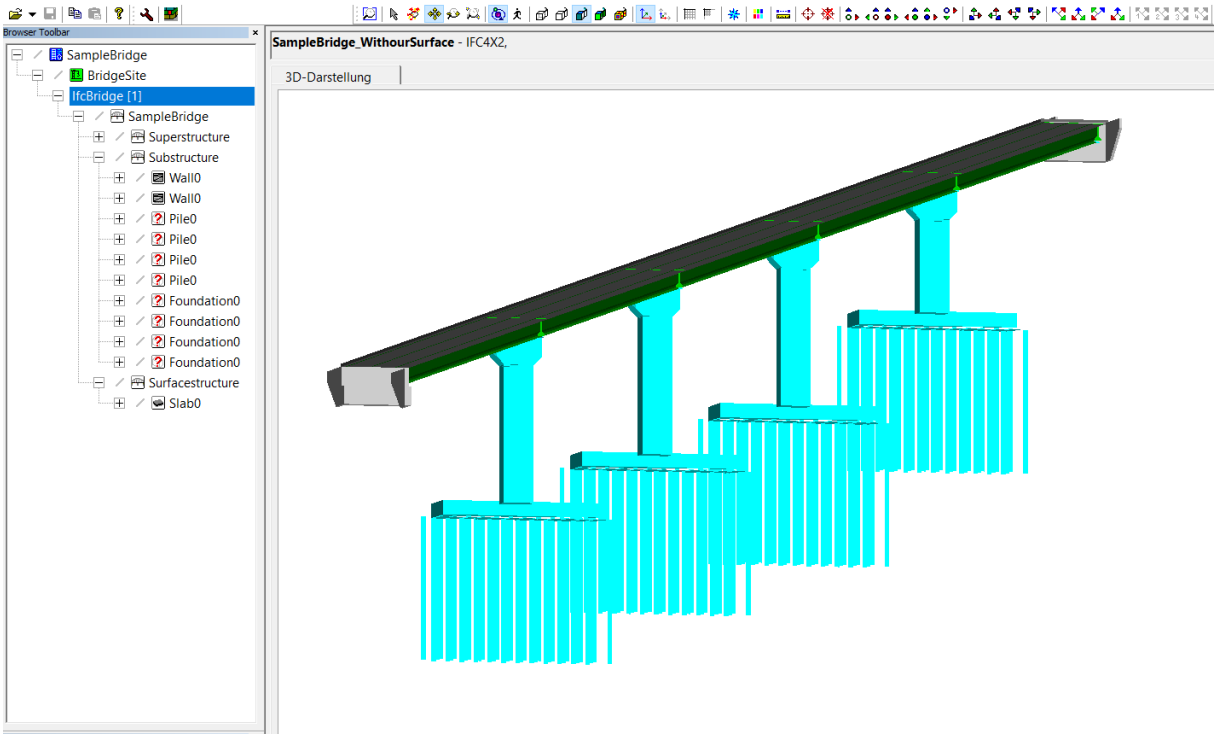


Abbildung 7.11: Räumliche Hierarchie und geometrische Repräsentation der Beispielbrücke im FZK-Viewer

Augenscheinlich ist die Repräsentation des Viewers nicht mit der Qualität von BimVisio zu vergleichen. Allerdings sind die integrierten Werkzeuge zur Überprüfung eines Modells gut geeignet. Auch fehlerhafte IFC-Modelle sind in dieser Software abbildbar. Ein integrierter Message Log kann Problemmeldungen fehlerhafter Implementierungen abrufen und wurde während des Entwicklungsprozesses verwendet, um falsche IFC Zuordnungen zu finden.

8 Zusammenfassung, Diskussion und Ausblick der Arbeit

In diesem Kapitel werden die Erkenntnisse und erarbeitete Applikation nochmals aufgegriffen. Dabei wird das Ergebnis kritisch beleuchtet und ein Ausblick für die zukünftige Nutzung von BIM für Infrastrukturbauwerksplanung gegeben.

8.1 Zusammenfassung

Durch das steigende Interesse einer breiten Nutzung von BIM für Infrastrukturellebauwerksplanung ist ein standardisiertes neutrales Datenaustauschformat von großer Bedeutung, damit ein geregelter Datentransfer zwischen allen Projektbeteiligten möglich ist. Das von vielen Softwareanwendungen integrierte Format IFC hat mit der Version IFC4X1 die Grundlage für den Infrastrukturbereich gesetzt. Mit der Veröffentlichung der Version IFC4X2 können Brückenbauwerke durch die IfcBridge Erweiterung in dem Datenschema IFC abgebildet werden. Diese Erweiterung dient als Ausgangspunkt zur Erstellung des IfcBridge Toolkits, wodurch IfcBridge-Modelle unter Zuhilfenahme visueller Programmierung generiert werden.

Um Erkenntnisse zu erlangen, welche geometrische und semantische Informationen eines digitalen Brückenmodells in bereits existierenden BIM-Softwareanwendungen vorhanden sind, wurden Brückenmodelle mit der Infrastruktur-Planungssoftware IW 2020 erstellt und über eine proprietäre Schnittstelle als Revit-Modell generiert. Die Prozessschritte sind währenddessen analysiert worden, wobei Problematiken und Erkenntnisse, die für die Erstellung der Applikation relevant sind, aufgezeigt werden.

Nach der Datenaufbereitung, mittels dem neutralen Klassifikationssystem Uniclass 2015 und der anschließenden Informationsfilterung in Dynamo, wurde die .NET Applikation IfcBridge Toolkit implementiert.

Das erstellte Toolkit verwendet ZTN, damit in Dynamo die implementierten Funktionen genutzt werden können, um ein IfcBridge-Modell zu generieren. Die Klassenbibliothek *IfcBridgeToolkit* ist unabhängig von der Bibliothek *IfcBridgeExporter_Dynamo.* Dadurch ist eine Nutzung der Funktionen der Klassenbibliothek IfcBridge Toolkit softwareunabhängig möglich.

Die Prüfung des IfcBridge-Modells zeigt, dass ein digitales Brückenmodell als IFC-Schema in der Version IFC4X2 abgebildet werden kann. Allerdings können durch den Prototypen nur die Minimalanforderungen eines IfcBridge-Modells abgedeckt werden.

8.2 Diskussion

Die vorliegende Bachelorarbeit zeigt anhand eines Entwicklungsprozesses in proprietären Softwareanwendungen die erfolgreiche Generierung eines Brückenmodells in dem herstellerneutralen Datenschema IFC. Durch eine visuelle Programmiersprache ist eine leicht verständliche Handhabung des Toolkits für Nutzer möglich und vermittelt zudem einen Überblick über die Struktur eine IfcBridge-Datei. Damit die Applikation die komplette IfcBridge-Struktur abbilden kann, sind jedoch weitere Entwicklungsschritte nötig.

8.2.1 Alignment

In der IFC-Version IFC4X2 können die geometrischen Repräsentationen und Platzierungen von Brückenkomponenten entlang eines Alignments dargestellt werden. Besonders für Elemente, die der Alignment-Achse folgen, ist diese Art von Beschreibung von großem Vorteil. Da in den erzeugten Beispielmodellen kein Alignment vorhanden ist, wurde alternativ eine andere geometrische Repräsentation und Platzierungsmethode gewählt. Eine Möglichkeit einer Erstellung von alignmentbasierten Brücken für das IFC-Schema wurde in den entwickelten Prototypen nicht implementiert. Durch den Aufbau der Applikationsarchitektur können die benötigten Funktionen ergänzt werden. Wegen des Verlustes der Semantik während der Übertragung der Beispielbrücken werden Informationen nicht geometrischer Art nicht in die IFC Datei übertragen. Das IFC Format stellt allerdings Möglichkeiten für semantische Beschreibungen zur Verfügung. Dies wird in den Prototypen demonstriert, indem durch implementierte Funktionen Material zugeordnet werden kann.

8.2.2 Geometrische Repräsentation

Durch die Applikation können derzeit nur Brückenkomponenten der geometrischen Repräsentation des Revit-Typs Solid in das IFC-Schema übermittelt werden. Bei der Filterung in Dynamo sind die geometrischen Informationen der Mesh-Körper abrufbar.

Bei Betrachtung der Mesh-Körper durch RevitLookup wird nicht erkenntlich, wie genau die Elemente aufbereitet werden können, um die geometrischen Daten für die Applikation zugänglich zu machen. Des Weiteren ist eine geometrische Repräsentation als *IfcTriangulatedFaceSet* eine nicht ideale Lösung zur Beschreibung von Brückenkomponenten. Im Rahmen der Arbeit werden ausschließlich Bauteilkomponenten mit der geometrischen Repräsentation eines Solid zur Erstellung eines IfcBridge-Modells mithilfe des entwickelten Prototyps genutzt. Diese Entscheidung wurde getroffen, da bei der Betrachtung der Elementinformationen sowohl in Dynamo als auch in RevitLookup auf die Faces der Körper zugegriffen werden kann. Faces beschreiben die Oberflächen der Komponenten, wodurch eine Repräsentation in dem IFC-Schema mittels *IfcFacetedBrep* als gut machbar erscheint. Durch die Nutzung der Klassenbibliothek *Geometry_Services von* (Esser et al., 2019) werden bereits Solid-Körper für die Applikation vorbereitet. Die geometrischen Beschreibungen *IfcSectionedSolidHorizontal* und *IfcSweptAreSolid* können als weiterer Entwicklungsschritt für die Applikation zugänglich gemacht werden, da die flexible Struktur des Toolkits eine schnelle Ergänzung der geometrischen Beschreibung des IFC Elements ermöglicht.

8.2.3 Materialdefinition

Durch die beiden ZTN *AddMaterialToComponentGroup()* und *AddMaterialToSpecificComponent()* soll dargestellt werden, dass Komponenten eines IfcBridge-Modells auch nachträglich mit semantischen Informationen ergänzt werden können. Um einem spezifischen Element Material zuzuweisen, soll in der Applikation die Identifikationsnummer der Komponente in der IFC Datei verwendet werden. Auch bei korrekter Eingabe der ID, wird die erstellte Materialbeschreibung nicht dem Element zugeordnet.

Im Idealfall wird die Materialbeschreibung für alle Komponenten einer Entität durch *AddMaterialToComponentGroup()* definiert, indem während der Ausführung der Funktion alle geforderten IFC-Elemente abgerufen werden. Dennoch wird nur der ersten Komponente Material zugewiesen. Die Behebung dieser Fehler ist in der weiteren Entwicklung des Prototypen essentiell. Trotz fehlerhaften Funktionen, konnte einer Komponente Material zugewiesen werden und dadurch wird aufgezeigt, dass IfcBridge-Modelle auch im Nachtrag schnell durch die Nutzung einer visuellen Programmiersprache um Semantik ergänzt werden kann.

8.2.4 Instabilitäten bei Ausführung

Bei der Nutzung der ZTNs des IfcBridge Toolkits in Dynamo kommt es bei der Ausführung eines Dynamoskriptes zu Komplikationen. Wenn das gesamte Brückenmodell in Dynamo in das IFC4X2-Schema überführt werden soll, ist ein Absturz der Software nicht auszuschließen. Um diese Problematik zu umgehen empfiehlt sich eine Schrittweise Erstellung eines IfcBridge-Modells, indem jeder ZTN einzeln ausgeführt wird. Diese Vorgehensweise wird ermöglicht, da der Inputparameter *StoreFilePath* auch durch den Dynamo internen Codeblock *Filepath* befüllt werden kann. Die allgemeine Struktur einer IfcBridge Datei kann problemlos hinzugefügt werden. Bei der Nutzung des ZTN *AddDirectShapeComponents()* kommt es auch bei der empfohlenen Vorgehensweise zu einem Absturz der Software. Elemente der Kategorie Brückentafel, Brückenpfeiler und Brückenwiederlager können problemlos übergeben werden. Wenn Elemente der Kategorie Brückenfundamente, Brückenträger und Brückenlager in die IFC Datei eingefügt werden sollen, kann es zu einem Absturz von Dynamo und Revit kommen. Wird *AddDirectShapeComponents()* für diese Elemente ausgeführt, ist eine Überprüfung mittels einem Debugging-Prozess in MVS nötig. Dadurch kann jede Zeile einzeln ausgeführt werden und eine komplikationslose Ausführung ist möglich. Diese Vorgehensweise erweist sich allerdings als sehr zeitintensiv. Um den Prototypen nutzerfreundlicher zu gestalten, muss die Applikation für Dynamo optimiert werden. Der Optimierungsprozess des IfcBridge Toolkits ist kein Teil der vorliegenden Bachelorarbeit. Trotz der beschriebenen Komplikationen ist das Erstellen eines IfcBridge-Modells, wie in Kapitel 7 beschrieben, ohne IFC Quellcode Manipulation möglich.

8.3 Ausblick

Die Nutzung von BIM für Infrastruktur - Bauprojekte spielt eine immer wichtigere Rolle. In manchen Ländern wird der Einsatz von BIM Methoden für öffentliche Projekte bereits gesetzlich vorgeschrieben. Ein Beispiel dafür ist Singapur. Durch staatliche Initiativen soll in Deutschland im Jahr 2020 eine flächendeckende Nutzung von BIM ermöglicht werden. Hierfür ist die Ausweitung des Konzepts für den Infrastruktursektor besonders wichtig, da sich die meisten öffentlichen Bauvorhaben in diesem Sektor befinden.

Die vorliegende Arbeit hat gezeigt, dass die infrastrukturelle Erweiterung des IFC Standards bereits jetzt genutzt werden kann. Auch eine Datenübermittlung von BIM-Modellen etablierterer Softwareanwendungen für den Hochbau in das IfcBridge Schema kann durch Verwendung einer visuellen Programmiersprache ermöglicht werden. Wobei der erstelle Prototyp für eine Exportmöglichkeit in das IFC-Schema eine Grundlage für Zukünftige Import- und Exportfunktionen darstellt.

Infrastruktur - Bauvorhaben werden traditionell durch zweidimensionale Baupläne beschrieben. Durch die Nutzung von BIM können BIM-Methodiken genutzt werden, um gängige immer wieder auftauchende Problematiken, wie beispielweise Kollisionskontrollen, zu bewältigen.

BIM, als ergänzendes Werkzeug, ermöglicht es dem Fachplaner, einen wirtschaftlichen, sicheren und ressourceneffizienten Bauablauf zu garantieren

Literaturverzeichnis

Autodesk (2019): InfraWorks. Planung und Konzeption von Infrastrukturprojekten im realen Kontext. URL: https://www.autodesk.de/products/infraworks/overview, zuletzt geprüft am 24.09.2019.

Azhar, S. (2011): Building Information Modeling (BIM). Trends, Risks, and Challenges for the AEC Industry. URL: https://ascelibrary.org/doi/pdf/10.1061/(ASCE)LM.1943-5630.0000127.

Bayrische Architektenkammer (2019): BIM Lexikon. URL: https://www.byak.de/planen-und-bauen/architektur-technik/building-information-modelling-bim/bim-lexikon.html, zuletzt geprüft am 24.08.19.

BimVision (2019). Version 2.21: Datacomp. URL: https://bimvision.eu/de/strona-glowna-de/.

Bormann, A., Amann, J., Chipman, T., Hyvärinen J., Liebich T., Muhič, S. et al. (2017): IFC Infra Overall Architecture Project. Documentation and Guidelines. URL: https://www.buildingsmart.org/wp-content/uploads/2017/07/08_bSI_OverallArchitecure_Guidelines_final.pdf, zuletzt geprüft am 24.08.19.

Borrmann, A., Elixmann, R., Eschenbruch, K., Hecker, D., Hochmuth, M., Klempin, C. et al. (2019a): Technologien im BIM-Umfeld. Handreichungen und Leitfäden - Teil 10. URL: https://bim4infra.de/wp-content/uploads/2019/07/BIM4INFRA2020_AP4_Teil10.pdf.

Borrmann, A., König, M., Koch, C. und Beetz, J. (Hg.) (2015): Building Information Modeling. Technologische Grundlagen und industrielle Praxis: VDI Buch. Springer Viweg.

Borrmann, A., Muhič, S., Hyvärinen, J., Chipman, T., Jaud, S., Castaing, C. et al. (2019b): The IFC-Bridge Project. Extending the IFC standard to enable High-Quality exchange of bridge information models. In: *in '2019 European Conference on Computing in Construction'*.

buildingSMART e.V. (2019a): IFC Model Views. URL: https://www.buildingsmart.de/kos/WNetz?art=Project.show&id=140, zuletzt geprüft am 25.08.2019.

buildingSMART e.V. (2019b): Industry Foundation Classes Version 4.2 bSI Draft Standard. URL: https://standards.buildingsmart.org/IFC/DEV/IFC4_2/FINAL/HTML/, zuletzt geprüft am 24.08.2019.

buildingSMART e.V. (2019c): Infrastructure Room. URL: https://www.buildingsmart.org/standards/rooms/infrastructure/, zuletzt aktualisiert am 24.08.2019.

buildingSMART e.V. (2019d): Standards. URL: https://www.buildingsmart.de/bim-knowhow/standards, zuletzt geprüft am 24.08.2019.

Bundesministerium für Verkehr und digitale Infrastruktur (2015): Stufenplan Digitales Planen und Bauen. URL: https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/stufenplan-digitales-bauen.pdf?__blob=publicationFile.

Bundesministerium für Verkehr und digitale Infrastruktur (2017): Umsetzung des Stufenplans Digitales Planen und Bauen. URL: https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/bim-umsetzung-stufenplan-erster-fortschrittsbe.pdf?__blob=publicationFile.

Carmona, J. und Irwin, K. (2007): BIM: Who, How and Why. URL: https://www.facilitiesnet.com/software/article.aspx?id=7546, zuletzt geprüft am 12.08.19.

Castaing, C., Borrmann, A., Benning, P., Dumoulin, C., Chipman, T., Hyvärinen, J. et al. (2018a): IFC-Bridge Fast Track Project. Report WP1: Requirements analysis.

Castaing, C., Borrmann, A., Chipman, T., Dumoulin, C., Hyvärinen, J., Liebich, T. et al. (2018b): IFC Bridge Fast Track Project. Report WP2: Conceptual Model. URL: https://buildingsmart.sharefile.com/share/view/s619ceb7623d4b099/fod4202e-f039-49d7-8e4e-cddabe6efd3c.

Delany, S. (2017): What is Uniclass 2015? URL: https://www.thenbs.com/knowledge/what-is-uniclass-2015, zuletzt aktualisiert am 26.04.2019, zuletzt geprüft am 24.08.2019.

DIN Bauportal GmbH (o.J.): DIN EN ISO 16739 IFC. URL: https://www.din-bauportal.de/Public/BIM/DIN-EN-ISO-16739.aspx, zuletzt geprüft am 25.08.2019.

Ehle, P. (2019): BIM im Infrastrukturbau. URL: https://blog.nupis.de/bim-im-infrastrukturbau/, zuletzt geprüft am 26.08.2019.

Esser, S. (2018): Implementierung einer Datenschnittstelle zur Unterstützung der modelbasierten Planung von Bahnausrüstungstechnik. Masterarbeit. Technische Universität München,. URL: https://publications.cms.bgu.tum.de/theses/2018_Esser_Vilgertshofer.pdf.

Esser, S. und Aicher, K. (2019): IfcBridge Model Generation using Visual Programming.

EUBIM Taskgroup (2017): Handbuch für die Einführung von Building Information Modelling (BIM) durch den europäischen öffentlichen Sektor. URL: http://www.eubim.eu/wp-content/uploads/2018/02/GROW-2017-01356-00-00-DE-TRA-00-1.pdf.

Gallaher, M., O`Conner, A., Dettbarn, J. und Gilday, L. (2004): Cost Analysis of Inadequate Interoperability in the U.S Capital Facilities Industry. URL: https://nvlpubs.nist.gov/nistpubs/gcr/2004/NIST.GCR.04-867.pdf.

Karlsruher Institut für Technologie (2019): KIT - FZKViewer. Version 5.1. URL: https://www.iai.kit.edu/1648.php.

Langwich, O. (2016): 3D wird verbindlich: Brückenkonstruktionen mit Revit. URL: https://www.autocad-magazin.de/3d-wird-verbindlich-brueckenkonstruktion-mit-revit/, zuletzt geprüft am 24.08.19.

Liebich, T. (2017): IFC für Infrastruktur. Neues von IFC-Alignment, IFC-Rail, IFC-Road, IFC-Bridge und IFC Tunnel. URL: https://www.buildingsmart.de/kos/WNetz?art=File.download&id=6316&name=1C-1_Dr.+Thomas+Liebich+%28AEC3%29_IFC+f%C3%BCr+Infrastrukturplanung.pdf, zuletzt geprüft am 24.08.19.

Markič, S. (2017): IFC-Bridge: Previous Initiatives and Their Proposals. URL: https://publications.cms.bgu.tum.de/2017_Markic_FBI.pdf.

Markič, S., Borrman, A., Windischer, G., Glatzl, W., Hofmann, M. und Bergmeister, K. (2019): Requirements for geo-locating transnational infrastructure BIM models. URL: https://publications.cms.bgu.tum.de/2019_Markic_WTC.pdf.

N+P Redaktion (2018): Vergleich von open BIM, closed BIM, little BIM, big BIM und connected BIM. URL: https://blog.nupis.de/vergleich-open-closed-little-big-bim-connected-bim/, zuletzt geprüft am 24.08.2019.

NBS (2019): Uniclass 2015. URL: https://www.thenbs.com/our-tools/uniclass-2015, zuletzt aktualisiert am Juli 2019, zuletzt geprüft am 25.08.2019.

Nitschke, C. (2015): Die Umsetzung von BIM im Ingenieurbau anhand von Brückenmodellen. Bachelorarbeit. Technische Universität München,. URL: https://publications.cms.bgu.tum.de/theses/2015_Nitschke_BIMBridge.pdf.

Preidel, C., Daum, S. und Borrmann, A. (2017): Data retrieval from building information models based on visual programming.

Raidl, C. (2018): 3D-Modellierung von Brücken mit Autodesk Revit und Erweiterung um IFC4x1 Entitäten. Bachelorarbeit. Technische Universität München,. URL: https://publications.cms.bgu.tum.de/theses/2018_Raidl_BA.pdf.

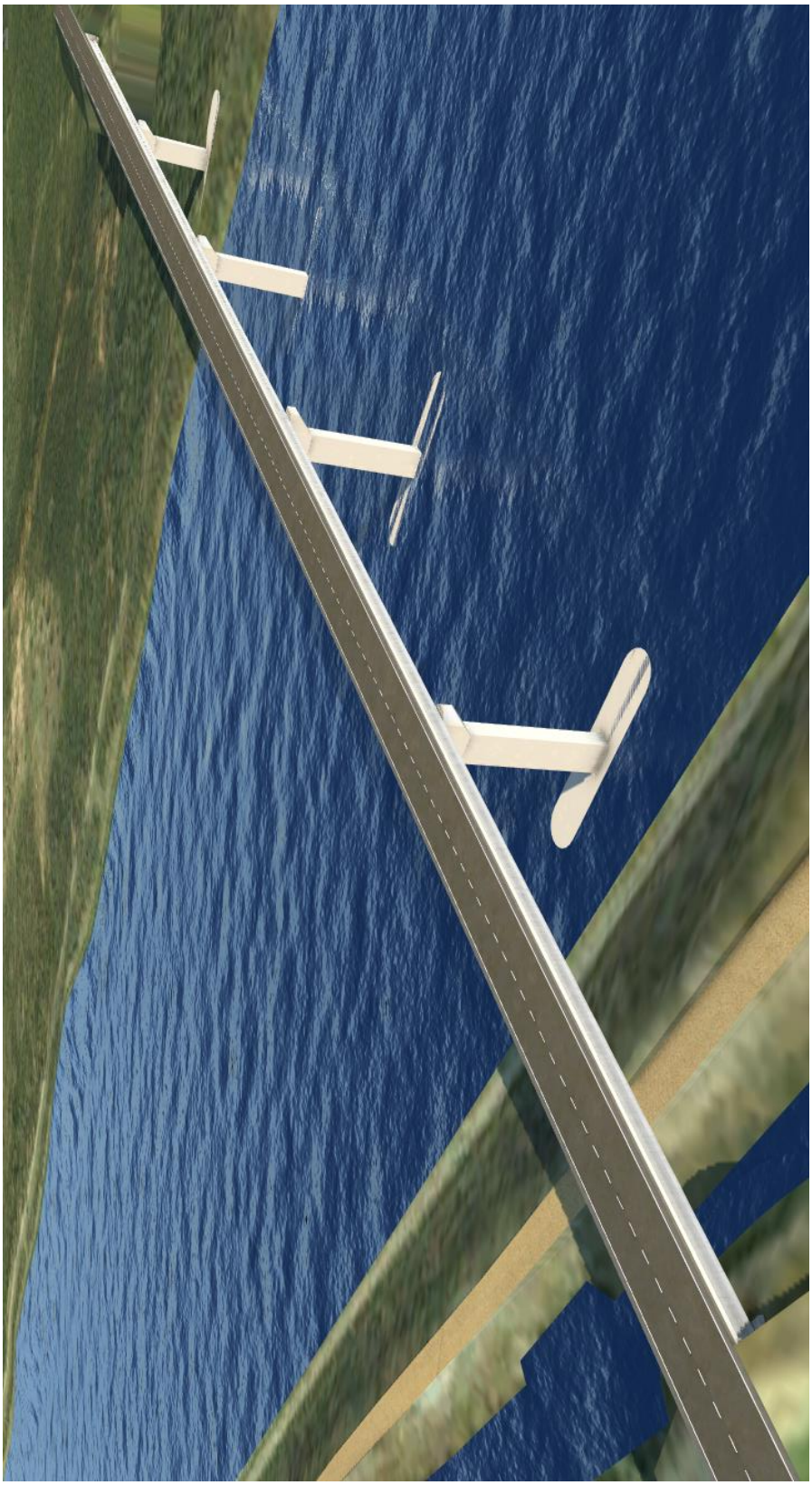
Ritter, F., Preidel, C., Singer, D. und Kaufmann, S. (2015): Visuelle Programmiersprachen im Bauwesen. Stand der Technik und aktuelle Entwicklung. URL: https://publications.cms.bgu.tum.de/2015_Ritter_FBI.pdf.

Trzeciak, M. und Borrmann, A. (2018): Design-to-design exchange of bridge models using IFC. A case study with Revit and Allplan. URL: https://publications.cms.bgu.tum.de/2018_TRZECIAK_ECPPM.pdf.

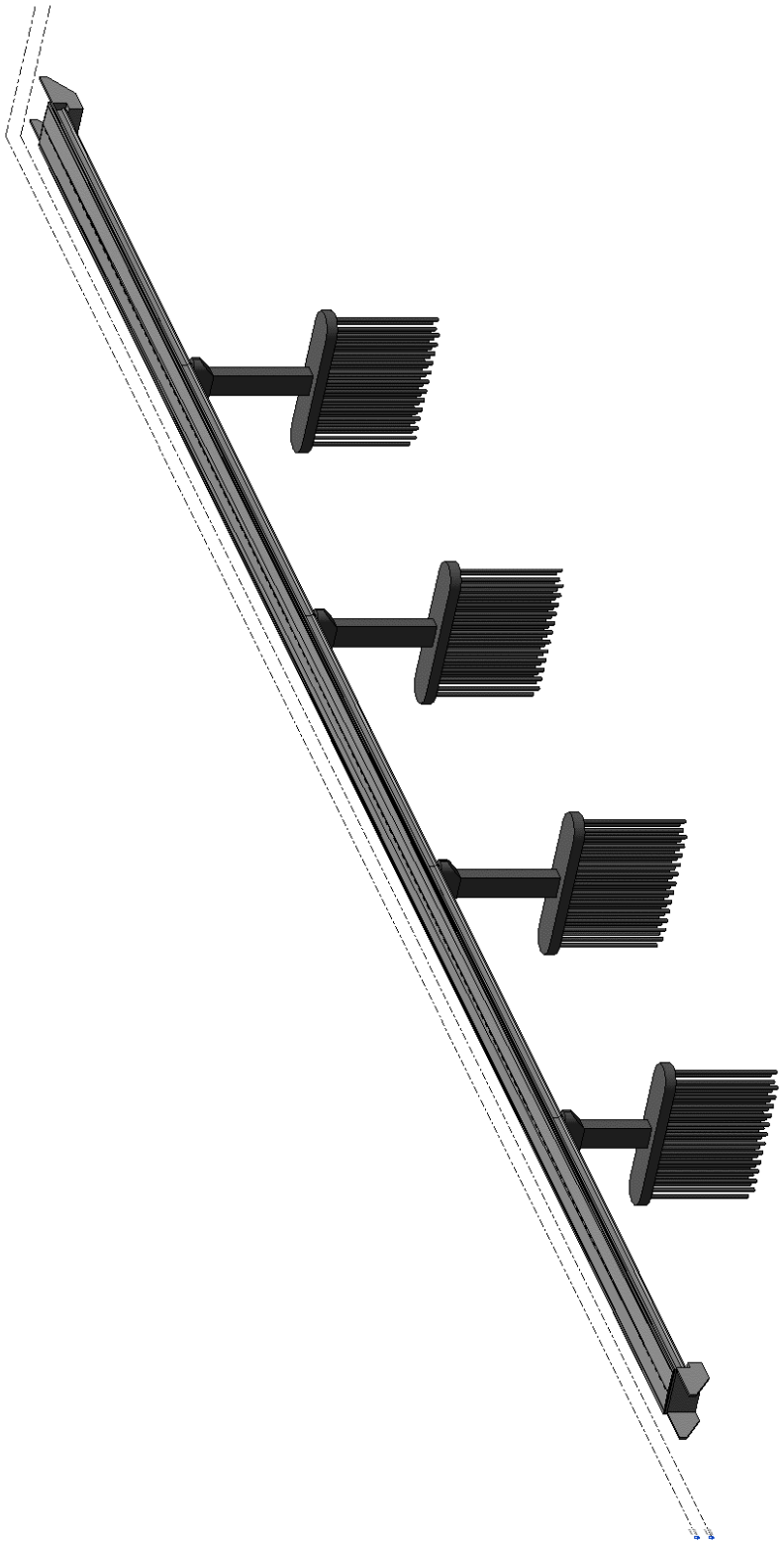
xBim Toolkit (2019). URL: http://docs.xbim.net/, zuletzt geprüft am 25.08.2019.

Anhang A: Bilder

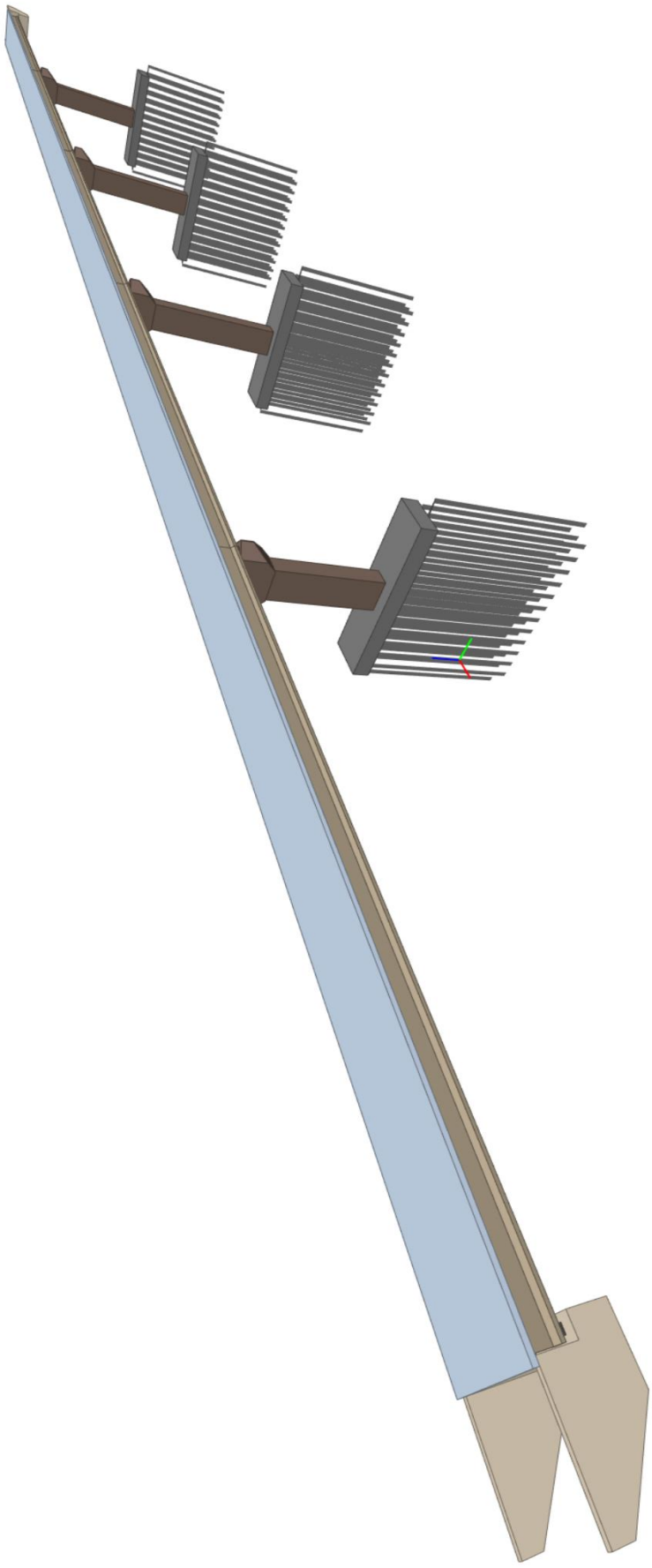
Beispielbrücke Kapitel 7 – Darstellung in InfraWorks 2020



Beispielbrücke Kapitel 7 – Darstellung in Revit 2020



Beispielbrücke Kapitel 8 – Darstellung in BimVisio



Anhang B: Digitaler Anhang

Bei der Abgabe wird dieser Arbeit in digitaler Form beigefügt:

- Die erstellten Beispielbrücken in InfraWorks 2020 sowie die dazugehörigen Revit-Projekte
- Der Quellcode der entwickelten Anwendung IfcBridge Toolkit
- Der benutzerdefinierte Dynamo-Codeblock
- Das Dynamoskript der Datenfilterung für Mesh- und Solid-Körper
- Das Dynamoskript der vollständigen Komponentensortierung
- Das Dynamoskript für den Erstellungsvorgang eines IfcBridge-Modells
- Ein IfcBridge-Beispielmodell im IFC-Format
- Die vorliegende schriftliche Ausarbeitung im PDF-Format
- Abbildungen des IfcBridge Toolkits Schema und Darstellung der Brücke in BimVisio