Fachhochschule Nordwestschweiz
Hochschule für Architektur, Bau und Geomatik

Institut Digitales Bauen
MSc FHNW VDC

Master of Science FHNW in Virtual Design and Construction

Automatisierte Baufortschrittsermittlung und -kontrolle

Vorgelegt von: Lucien Zenners

Thesis Begleiter: Prof. Lukas Schildknecht, FHNW
lukas.schildknecht@fhnw.ch

Thesis Experte: Oliver Schneider, FHNW
oliver.schneider@fhnw.ch

Praxispartner:in: Innosuisse Forschungsprojekt BMETRY

Muttenz, 13. Januar 2022

Abstract

Aktuelle Daten über den Stand des Projektes sind für die Arbeit von Projektmanager:innen eine wichtige Grundlage. In dieser Arbeit wurde deshalb untersucht, wie der Baufortschritt automatisiert erfasst, ausgewertet und dargestellt werden kann. Die Interpretation der aktuellen Lage soll damit datengetrieben erfolgen und nicht mehr ausschliesslich auf den Erfahrungen und Interpretationen von Bauleiter:innen beruhen. Basierend auf den Erkenntnissen der Recherche wurde ein Konzept entwickelt, welches anschliessend mit einem Proof of Concept validiert wurde. Dieses besteht aus einem datenbankbasierten 4D digitalen Bauwerksmodell (DBM), einem Soll-Ist-Modellvergleich und einer Visualisierung des Baufortschritts anhand eines farbcodierten DBM. Der Modellvergleich erfolgt dabei geometrisch, basierend auf einem im Forschungsprojekt BMETRY generierten As-Built-DBM und wurde in Rhino Grasshopper umgesetzt. Schlussendlich konnte mit dieser Arbeit gezeigt werden, dass ein datenbankbasierter Ansatz für die automatisierte Baufortschrittsermittlung und -Kontrolle heute schon in der Praxis umsetzbar ist und sich gut für Open-BIM-Prozesse eignet. Bestätigt wurde dies, indem er in einem Zusammenhängenden Workflow vom Modellvergleich bis zur Visualisierung umgesetzt werden konnte.

Selbständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig sowie ohne unerlaubte fremde Hilfe und ausschliesslich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Wil, 12. Januar 2022

Ort, Datum

Unterschrift

Inhalt

1. Abbildungen 4
2. Tabellen 6
3. Abkürzungen 7
4. Einleitung 8
5. Methodik 10
6. Stand der Forschung 13
   6.1. 4D Digitale Bauwerksmodelle 13
      6.1.1. Proprietäre, geschlossene Systeme 13
      6.1.2. Attributbasierte Ansätze 14
      6.1.3. 4D-IFC 16
      6.1.4. Datenbankbasierter Ansatz 19
   6.2. Modellvergleich: As-Planned vs. As-Built 21
      6.2.1. Geometriebasierte Ansätze 21
      6.2.2. Visuelle Ansätze 24
      6.2.3. Den Modellvergleich unterstützende Methoden 25
   6.3. Darstellung & Auswertung 27
7. Konzept 31
8. Proof of Concept 35
   8.1. As-Planned- & As-Built-DBM 36
   8.2. Modellvergleich 38
   8.3. Datenbank 41
   8.4. Visualisierung 43
   8.5. Nicht implementierte Features 44
9. Exkurs Proof of Concept 4D-IFC 47
10. Diskussion 49
11. Literatur 51
12. Anhang 56

1. Abbildungen

Abbildung 1: Suchmethode und Resultate 11

Abbildung 2: Verknüpfung eines Terminplan und DBM mit Hilfe von Autodesk Navisworks 13

Abbildung 3: Erstellung und Export eigener PSet mit Autodesk Revit. Definition Eigenschaften in Autorenwerkzeug (links oben), Zusammenfassen von Eigenschaften in ein eignes PSet (unten), Darstellung in Model Viewer (rechts oben). 15

Abbildung 4: Erstellung eines PSet mit Hilfe eines Python-Skriptes (oben) und Darstellung in einem Model Viewer (unten) 15

Abbildung 5: Ausschnitt IFC-Schema (IFC4). Für die Modellierung von Bauprozessen notwendige Klassen (grün). 17

Abbildung 6: Schematische Darstellung von Verschiedenen Beziehungen die für einen Task modelliert werden können (IFC4). Beziehungen mit Hilfe von Subklassen der Klasse IfcRelationship (rot), Beziehung durch Attribute (lila Linien). 18

Abbildung 7: ERD der Beziehung zwischen Terminplan und BIM-Model (Park et al., 2017). 19

Abbildung 8: Beispiel WBS-Code (Park et al., 2017). 20

Abbildung 9: Formel für die Point Cloud Density (Bognot et al., 2018) 22

Abbildung 10: Verifizierung der Wandgeometrie: (a) Punktwolke, (b) Ausgangspunkt und nächste 102'400 Nachbarn, (c) alle nächsten Nachbarn, (d) der Wand zugeordnete Punkte (Bariczová et al., 2021). 23

Abbildung 11: Konzep Modellvergleich: Voxelraster (gelb), 2D-Raster über Elementoberfläche (grün), Oberfläche eines Elements (schwarz dick), Bounding Box zur Rasterzellen Verifizierung (schwarz dünn) (Tuttas et al., 2015). 24

Abbildung 12: Detektion und Segmentierung von vorfabrizierten Betonelementen auf zwei verschieden Baustellen: (a) und (b) (Wang et al., 2021). 25

Abbildung 13: Bildgenerierung: (a) Kamerasetup in Blender, (b) gerendertes Bild der selektierten Kamera, (c) binäre Maske für zu untersuchendes Element (Pal et al., 2022). 26

Abbildung 14: Konzept Precedence Relationship Graph: Checkpoint-Komponenten (rot) (Braun et al., 2021). 27

Abbildung 15: EVMS-Diagramm mit S-Kurve (Kim et al., 2020). 28

Abbildung 16: Anwendung der RoC-Methode auf ein Bauprojekt (Alzraiee, 2018). 29

Abbildung 17: Ein in einem Farbcode dargestelltes DBM, im von (Park et al., 2017) entwickelten Webviewer. 30

Abbildung 18: Teilprojekte BMETRY (Schneider et al., 2021). 31

Abbildung 19: Konzept Systemarchitektur 32

Abbildung 20: Workflow Datenaktualisierung: (a) & (b) Update IFC-Dateien, (c) Update Terminplan. 33

Abbildung 21: Workflow User-Anfrage 34

Abbildung 22: Schematische Darstellung der einzelnen Bestandteile des PoC und ihre Beziehungen zueinander 35

Abbildung 23: Workflow PoC für das Szenario «Update Ist-DBM». 36

Abbildung 24: As-Planned-DBM für PoC: Zone Nord (blau) und Zone Süd (grün). 37

Abbildung 25: Ausgewählte Beispiele der AB-DBM: (a) Stand 13.01.23 nach Bauablauf, (b) Stand 31.01.23 nach Bauablauf und (c) Stand 23.02.23 Abweichung von Bauablauf. 38

Abbildung 26: Übersicht Grasshopper-Skript für den Modellvergleich: (a) Input, (b) Geometrievergleich für jedes Element pro Elementtyp aufgeteilt nach Elementtyp, (c) Resultate Geometrievergleich als CSV-String, (d) aktuelles Datum und (e) Output. 39

Abbildung 27: PoC Grasshopper-Skript Komponente Geometrievergleich Wände: (a) Filterung der IFC-Datei nach Wänden, (b) Zuordnung sich schneidender Wände des As-Planned- und As-Built-Modells, (c) Check, ob die Elemente wirklich zusammengehören, (d) Extaktion der der Bauteilmengen, (e) Berechnung Fortschritt pro Element und (f) Extraktion Element-GUID. 39

Abbildung 28: Konzept Geometrievergleich: As-Planned-Wand (grün) und AB-Wände (schwarze Linien) 40
Abbildung 29: Beispiel-Output Teilprojekt "4D Exctraction" von BMETRY: Unterschied in der Wanddicke bei extrahierten Wänden auf Grund des Erkennungsalgorithmus (Marcinkeviciute, 2023). 41
Abbildung 30: Formel zur Berechnung des Fortschritts einer Wand. 41
Abbildung 31: UML der Beziehungen zwischen den einzelnen Tabellen der Datenbank. 42
Abbildung 32: Visualisierung Baufortschritt mit einem interaktiven Dashboard in MS PowerBI. 43
Abbildung 33: Visualisierung Baufortschritt mit als Balkendiagramm in Autodesk Navisworks. 44
Abbildung 34: Extraktion der tragenden Schicht (grün) einer Wand mit Grasshopper. 44
Abbildung 35: Ergänzung einer Relation zwischen den verschiedenen Etappen des Bauablaufs (rot). 45
Abbildung 36: Ergänzung der Klasse EtappenTyp um Aussagen über den voraussichtlichen Abschluss einer Etappe machen zu können. 46
Abbildung 37: Resultat Import 4D-IFC in Autodesk Navisworks. 47
Abbildung 38: Ausschnitt 4D-IFC-Datei: IfcScheduleTimeControl (blau), IfcRelAssignsTasks (grün) und IfcRelAssignsToProcess (orange). 48

2. Tabellen

Tabelle 1: Suchbegriffe 10
Tabelle 2: Ausschlusskriterien beim Titel- und Abstract-Screening. 12
Tabelle 3: Auswahl an Attributen der Entität IfcTaskTime (buildingSMART, 2017) 18
Tabelle 4: Terminplan synthetisches As-Planned-DBM 37
Tabelle 5: Matrix zur Klassifizierung von Bauelementen 42
Tabelle 6: Matrix zur Klassifizierung der Etappen (umgesetzt) 43
Tabelle 7: Matrix zur Klassifizierung der Etappen (nicht implementiert) 45

3. Abkürzungen

4D-DBM	=	Digitales Bauwerksmodell (3D + Zeit)
AR	=	Augmented Reality
BIM	=	Building Information Modeling
B-Rep	=	Boundary Representation
bSI	=	buildingSMART International
CDE	=	Common Data Environment
CNN	=	Convolutional Neural Network
CSV	=	Comma-Separated Values
DBM	=	Digitales Bauwerksmodell
ERD	=	Entity Relationship Diagram
EVMS	=	Earned Value Management System
FHNW	=	Fachhochschule Nordwestschweiz
GUID	=	Global Unique Identifier
IFC	=	Industry Foundation Classes
JSON	=	JavaScript Object Notation
MVD	=	Model View Definition
PoC	=	Proof of Concept
PRG	=	Precedence Relationship Graph
PSet	=	IfcPropertySet
RoC	=	Rules of Credit
SQL	=	Structured Query Language
VR	=	Virtual Reality
WBS	=	Work Breakdown Structure (Projektstrukturplan)
XML	=	Extensible Markup Language

4. Einleitung

Aktuelle Daten über den Baufortschritt sind für die Arbeit von Projektmanager:innen eine wichtige Grundlage, da Termine, zusammen mit Kosten und Qualität, die Schlüsselkennzahlen (Key Performance Indicators) für erfolgreiche Bauprojekte bilden (Alizadehsalehi & Yitmen, 2019; Atkinson, 1999). In der Theorie spricht man auch vom Eisernen Dreieck (Iron Triangle), ein Indiz für die starken Abhängigkeiten die zwischen diesen Faktoren bestehen (Fahri et al., 2015). Dennoch erfolgt die Erfassung des Baufortschritts in der heutigen Praxis meistens auf Baustellenrundgängen manuell, vor Ort (Blaser Zürcher, 2022; Riecke, 2022). Die Interpretation der aktuellen Lage beruht dabei auf der Erfahrung des / der jeweiligen Bauleiter:in. Diese Faktoren machen das heutige Vorgehen zeitintensiv, arbeitsaufwändig (Bosche et al., 2009; Navon & Sacks, 2007), zu unregelmässig (Fard & Peña-Mora, 2012) und fehleranfällig (Yang et al., 2015). 4D Digitale Bauwerksmodell (DBM), also dreidimensionale DBM plus den Faktor Zeit, bieten die Möglichkeit, die bestehenden Prozesse durch Automation, Simulation und verständlichere Darstellungen signifikant zu verbessern (Golparvar-Fard et al., 2015; O'Keeffe et al., 2017; Pal et al., 2021).

In dieser Arbeit wird, deshalb versucht eine auf 4D-DBM basierender automatisierte Baufortschrittsermittlung und -kontrolle umzusetzen. Dazu wird in einem ersten Schritt der aktuelle Stand der Forschung, anhand von drei Teilfragestellungen, recherchiert. Mit der Ersten wird untersucht, wie zeitliche Informationen in einem DBM modelliert werden können. Denn Leite et al. (2016) haben diesen Aspekt als eine der grössten Schwierigkeiten zur Umsetzung einer auf DBM basierende Baufortschrittskontrolle identifiziert. Anhand der Zweiten wird die Frage, wie auf der Baustelle erfasste As-Built-Daten automatisiert mit einem As-Planned-DBM verglichen und anschliessend ausgewertet werden können, beantwortet. Die Dritte befasst sich mit der Visualisierung des Baufortschritts. Hierbei wird untersucht, wie diese anhand von DBM umgesetzt werden kann. Golparvar-Fard et al. (2015) und Pal et al. (2021) sehen darin grossen Potential die Zusammenarbeit der Projektbeteiligten zu verbessern, indem der aktuelle Baufortschritt für alle verständlich dargestellt und jederzeit abrufbar sein würde.

Die Umsetzung der in dieser Arbeit entwickelten Lösung erfolgt in engem Austausch mit dem Innosuisse Forschungsprojekt BMETRY (BMETRY) und baut auf dem aktuell darin entwickelten Mess- und Dokumentationssystem auf. Die Grundlage, für die in BMETRY entwickelte Lösung, bilden hochauflösende Kameraaufnahmen inklusive exakter Positionsdaten, die von einem Baustellenkran aus, automatisiert erfasst werden. Mithilfe von Photogrammetrie, datengetriebener Bildverarbeitung und maschinellem Lernen werden die erhobenen Daten anschliessend ausgewertet und weiterverarbeitet. In einem nächsten Schritt werden daraus Vektordaten und semantische Informationen extrahiert,

die zur Kontrolle, Dokumentation und Qualitätssicherung weiterverwendet und in einem Webviewer dargestellt werden. Dabei stellt die Extraktion und Modellierung eines As-Built-DBM ein wichtiges Teilprojekt dar, welches die Schnittstelle, zur in dieser Arbeit entwickelten Lösung, bildet. (Schneider et al., 2021)

Aktuell in der Forschung verwendete Methoden den einen Soll-Ist-Modellvergleich umzusetzen basieren meist auf As-Built-Punktwolken oder -Bilddaten (Yang et al., 2015). Die Erkennungsrate diese Ansätze leidet allerdings aufgrund von verdecken Elementoberflächen durch temporäre Installationen, Baumaschinen oder Personal (Braun et al., 2021). Deshalb scheint ein Soll-Ist-Modellvergleich, basierend auf dem in BMETRY generierten As-Built-DBM, eine vielversprechende Lösung zu sein, da darin keine Störelemente enthalten sind (Schneider et al., 2021).

Die in BMETRY entwickelte Lösung wird in Pilotprojekten bereits in der Praxis angewendet (Schneider et al., 2021). Deshalb wird die in dieser Arbeit entwickelte Lösung so umgesetzt, dass sie mit BMETRY kompatibel ist und auf den darin entwickelten Ansätzen aufbaut. Die Integration der entwickelten Lösung in ein praxistaugliches System wird dadurch stark vereinfacht. Zudem dienen die bei dieser Arbeit gewonnenen Erkenntnisse allgemein als Grundlage für die Weiterentwicklung von BMETRY.

Der Aufbau dieser Arbeit umfasst einen theoretischen und einen praktischen Teil. Im theoretischen Teil wird der aktuelle Stand der Forschung zu den Themen 4D-Modellierung, Modellvergleich und Darstellung untersucht. Dazu dienen eine systematische Literaturrecherche und Fachgespräche mit verschiedenen Personen aus der Praxis und Forschung. Aufbauend auf den im theoretischen Teil gewonnenen Erkenntnissen wird der praktische Teil entwickelt und umgesetzt. Dieser basiert auf einem Modellvergleich zwischen einem von BMETRY generierten As-Built-DBM und einem AS-Planned-4D-DBM. Der daraus abgeleitete Baufortschritt wird anschliessend anhand eines DBM visualisiert.

5. Methodik

Der aktuelle Stand der Forschung im Bereich der automatisierten Baufortschrittsermittlung- und -kontrolle» wurde mit Hilfe einer systematischen Literaturrecherche ermittelt. Um ein Bias zu verhindern, wurden die beiden Datenbanken von Scopus und Swisscovery verwendet. Tabelle 1 zeigt die relevanten Suchbegriffe, die verwendet wurden. Gesucht wurde sowohl im Titel, dem Abstract und in den Key-Words.

	Schlagwörter	Synonyme / Verwandte Begriffe
inklusive	Construction	
	Progress Monitoring	Progress Measuring, Progress Tracking, Progress Updating
	BIM	Building Information Model, Building Information Modeling
exklusive	Infrastructure	

Tabelle 1: Suchbegriffe

Ausgeschlossen wurden alle Resultate, die sich mit Infrastrukturbauten auseinandersetzen, da es sich bei diesen meist um lineare Baustellen, mit grosser geografischer Ausdehnung handelt. Im Gegensatz zu klassischen Hochbauprojekten haben diese deshalb andere Anforderungen an eine Baufortschrittskontrolle (Riecke, 2022). Das Innosuisse Forschungsprojekt BMETRY fokussiert aktuell jedoch ausschliesslich auf Hoch- und Tiefbauprojekte, die mit Hilfe von Baukränen, ganzheitlich erfasst werden (Schneider et al., 2021).

In Abbildung 1 ist die systematische Literaturrecherche schematisch dargestellt. Auf Grund der zeitlichen Begrenzung dieser Arbeit wurde nur Literatur von 2016 und neuer, begutachtete Literatur (Peer-Review) und Literatur in englischer oder deutscher Sprache berücksichtigt. Diese Filter haben die Menge der Resultate signifikant begrenzt und die Suche nach wissenschaftlich relevanter und aktueller Literatur unterstützt. Gefiltert wurde direkt mit Hilfe der Suchmaschinen. Anschliessend wurden alle aus der Suche resultierenden Literaturen, in das Literaturverwaltungsprogramm Zotero importiert und Duplikate gelöscht. In den nächsten Schritten wurde die Literatur anhand der Titel und anschliessen anhand des Abstracts weiter gefiltert, damit schlussendlich nur noch relevante Literatur berücksichtigt wurde.

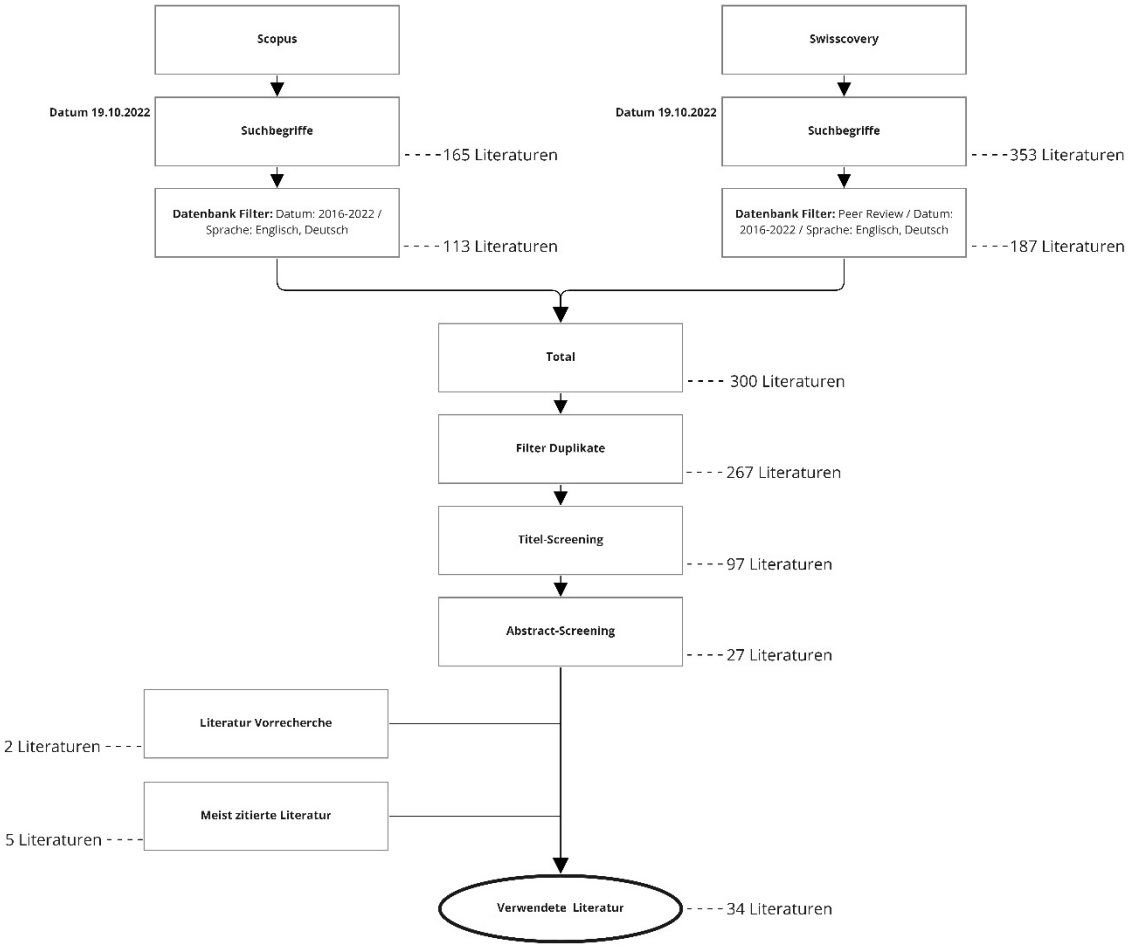


Abbildung 1: Suchmethode und Resultate

Die verwendeten Ausschlusskriterien sind in Tabelle 2 aufgeführt. Das Resultat dieses Screening-Prozesses waren 27 Literaturen. Dazu kamen noch zwei weiteren Literaturen aus der Vorrecherche und die fünf meistzitierten Literaturen der eingeschlossenen 27 Literaturen. Schlussendlich wurden somit 34 Literaturen verwendet, um den Stand der Forschung zu analysieren. Da das IFC-Schema De-Facto-Standard für den Austausch von Bauwerksinformationen im Hochbau ist (Borrmann et al., 2021a), war neben den Literaturen die Dokumentation der Industry Foundation Classes (IFC), die von buildingSMART International (bSI) gepflegt wird, wichtiger Bestandteil dieser Arbeit.

Ausschlusskriterium	Allfällige Begründung
Baufremdes Fachgebiet	
Befasst sich ausschliesslich mit der Erfassung von Daten (z. B.: automatisierte Drohnenflüge)	Die Erfassung der Daten wurde in BMETRY schon vertieft analysiert.
Befasst sich ausschliesslich mit Scan2BIM-Methoden	Das in BMETRY entwickelte Scan2BIM-Verfahren bildet eine wichtige Grundlage für diese Arbeit. Diese zu hinterfragen, würde diese Arbeit zu weit fassen.
Nicht in Deutsch / Englisch verfasst	
Rein visuelle Prüfungen von Baufortschritt (z. B.: Augmented Reality (AR))	Fällt nicht in die Kategorie einer automatisierten Baufortschrittskontrolle.

Fokussiert auf den Baufortschritt des Innenausbaus	BMETRY basiert auf von einem Baukran aus aufgenommener Bilddaten. Der Innenausbau eines Gebäudes kann damit nicht erfasst werden.
Nicht Bauteilorientiert (z. B.: Baufortschritt anhand von bestelltem Material)	Bietet keinen Mehrwert für BMETRY.
Kein Zugriff auf den Volltext	

Tabelle 2: Ausschlusskriterien beim Titel- und Abstract-Screening.

Um festzustellen, was eine atomisierte Baufortschrittskontrolle leisten muss, damit ein echter Mehrwert für Anwender:innen in der Praxis entsteht, wurden Fachgespräche mit verschiedenen Baufachleuten geführt. Dabei wurde darauf geachtet ein möglichst ganzheitliches Bild zu schaffen, indem das Gespräche zu aktuell in Praxis und / oder Lehre tätigen Bauleiter:innen gesucht wurde, egal ob diese auf der Baustelle noch mit 2D-Plänen oder schon mit DBM arbeiten.

6. Stand der Forschung

6.1. 4D Digitale Bauwerksmodelle

Es wurden vier verschiedenen Ansätze gefunden, Zeitinformationen mit einem DBM zu verbinden.

6.1.1. Proprietäre, geschlossene Systeme

Software wie Autodesk Navisworks, Synchro 4D, oder Desite BIM bieten die Möglichkeit Terminpläne mit einem DBM zu verbinden. Die Terminpläne lassen sich dabei manuell oder halbautomatisiert in der jeweiligen Software erstellen. Zudem bietet Desite BIM die Möglichkeit Terminpläne basierend auf festgelegten Regeln und Parametern direkt von einem DBM abzuleiten und zu generieren. Weiter gibt es in all diesen Softwarelösungen die Möglichkeit Terminpläne zu importieren. Zum Beispiel mit Hilfe einer einfachen Textdatei wie CSV oder als direkte Schnittstelle mit Projektmanagementanwendungen wie Microsoft Project. (Autodesk, 2022; bentley, 2022; thinkproject, 2022)

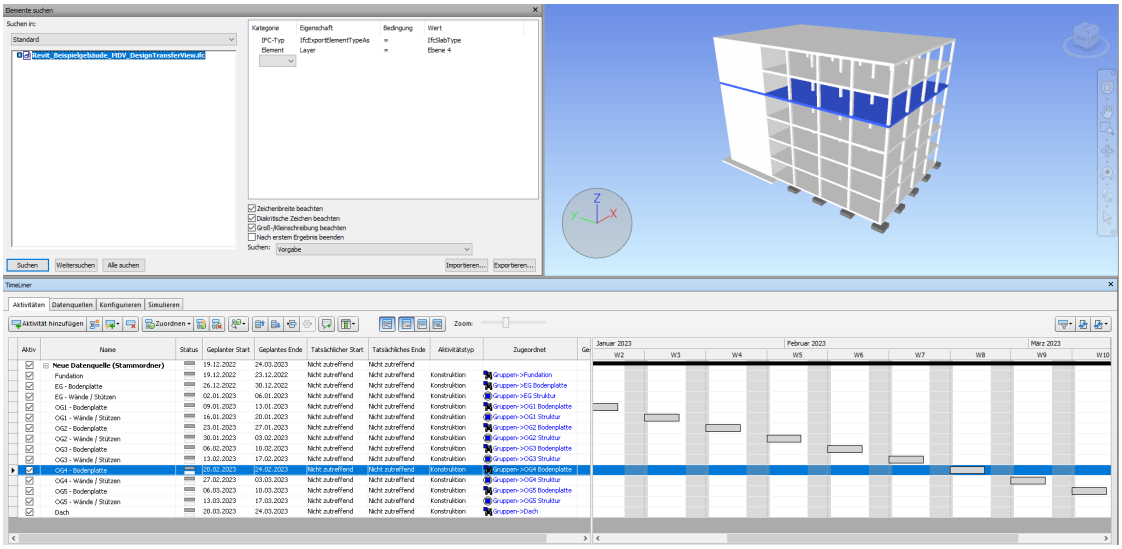


Abbildung 2: Verknüpfung eines Terminplan und DBM mit Hilfe von Autodesk Navisworks

Die Verknüpfung von Elementen des DBM mit spezifischen Teilprozessen des Terminplans kann manuell erfolgen oder basierend auf festgelegten Regeln. Werden die Elemente regelbasiert mit dem Terminplan verknüpft, wird die Verknüpfung bei Änderungen des Terminplans oder des Modells automatisiert angepasst. Wie eine solche Zuweisung funktioniert, ist softwarespezifisch gelöst. Abbildung 2 zeigt ein Beispiel aus Autodesk Navisworks. Mit Hilfe von Suchgruppen können spezifisch Elemente des DBM anhand beliebiger Elementeigenschaften isoliert und anschliessend mit einem importierten oder lokal erstellten Terminplan verknüpft werden. Sind die Suchgruppen sauber mit dem Terminplan verknüpft und so

ausformuliert, dass alle Elemente einem Prozess zugeordnet werden können, ist dies ein einmaliger Aufwand.

Wird dieser Ansatz für die Erstellung eines 4D Digitalen Bauwerksmodells (4D-DBM) verwendet, bringt das den Vorteil, dass man von den integrierten Funktionen der jeweiligen Software profitieren kann. Zudem bieten einige Softwareprodukte, wie Desite BIM, Möglichkeiten die Funktionen mit eigenen Skripten individuell zu erweitern (thinkproject, 2022). Park et al. (2017) weist allerdings darauf hin, dass mit diesem Ansatz 4D-DBM-Daten nur mit Hilfe der jeweiligen Software erstellt, angepasst und ausgetauscht werden können. Heutige Softwareprodukte bilden damit grösstenteils Automatisierungsinseln (Borrmann & Koch, 2021). Das hat zur Folge, dass ein solcher datei- und softwarebasierter Ansatz zu Informationsverzögerung und Wiedersprüchen in den Daten führen kann (Froese et al., 2000) und kein verlustfreier, digitaler Datenfluss im Sinne eines BIG-BIM-Ansatzes gewährleistet werden kann (Borrmann & Koch, 2021).

6.1.2. Attributbasierte Ansätze

Ein DBM enthält typischerweise dreidimensionale Geometrien der einzelnen Elemente eines Bauwerks in einem vordefinierten Detaillierungsgrad. Zudem umfasst es nicht-physische Elemente wie Räume und Zonen sowie eine hierarchische Projektstruktur (Borrmann et al., 2021b). Diese Elemente werden in der Datenmodellierung als Entitäten bezeichnet (Artus et al., 2021). Jede dieser Entitäten verfügt über einen Satz semantischer Informationen, wie z. B. Elementtyp, bauphysikalische Eigenschaften oder Beziehungen zu anderen Elementen (Borrmann et al., 2021b). Dieser Satz an Informationen bildet die Attribute also die Elementeigenschaften einer Entität. Dabei besitzen alle Entitäten eines Typs dieselben Attribute, allerdings mit individuellen Attributwerten (Artus et al., 2021).

Diese Attribuierung kann nun genutzt werden, um 4D-Informationen in ein DBM zu integrieren. Dies kann direkt in einem Autorenwerkzeug, wie Graphisoft ArchiCAD, Autodesk Revit, oder Vectorworks, entweder manuell oder automatisiert mit Hilfe eines Zusatzmodules, wie z. B. Dynamo für Autodeskt Revit, umgesetzt werden. Die Attribute können daraufhin in der entsprechenden Software bearbeitet oder mit Hilfe eines beliebigen Austauschformats weitergegeben werden.

In einem Open-BIM-Prozess ist der defacto Standard IFC als Austauschformat besonders geeignet. Dabei ist zu beachten, dass im IFC-Schema zwischen statisch integrierten Attributen und dynamisch erzeugbaren Eigenschaften (engl. properties) unterschieden wird. Diese Unterscheidung dient dazu, dass ohnehin umfangreiche Schema nicht unnötig aufzublasen. Die Umsetzung erfolgt mit Hilfe von Subklassen von IfcProperty. Diese können einzeln oder als IfcPropertySets (PSet) gruppiert einem Element zugewiesen werden (Borrmann et al., 2021a). In der IFC-Dokumentation sind

einige standardisierte PSet aufgeführt, allerdings keines, dass explizit für Baufortschritt verwendet werden könnte (buildingSMART, 2022). Abbildung 3 zeigt deshalb wie eigene Eigenschaften in einem Autorenwerkzeug erstellt, als PSet exportiert und einem Model Viewer dargestellt werden können.

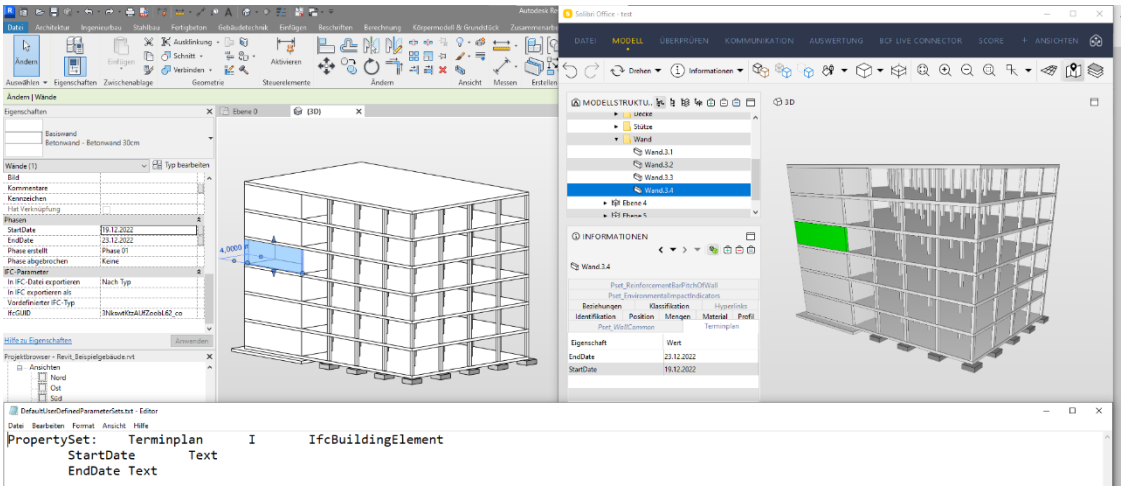


Abbildung 3: Erstellung und Export eigener PSet mit Autodesk Revit. Definition Eigenschaften in Autorenwerkzeug (links oben), Zusammenfassen von Eigenschaften in ein eignes PSet (unten), Darstellung in Model Viewer (rechts oben).

Werden die Attribute nicht in einem Autorenwerkzeug erstellt, gibt es Möglichkeiten 4D-Informationen direkt, z. B. mit Hilfe von Python- oder Grasshopper-Skripten, in eine IFC-Datei zu integrieren. Abbildung 4 zeigt an einem einfachen Python-Skript, wie dieses Vorgehen umgesetzt werden kann. Das Python-Skript ist in Anhang 11.1 dokumentiert.

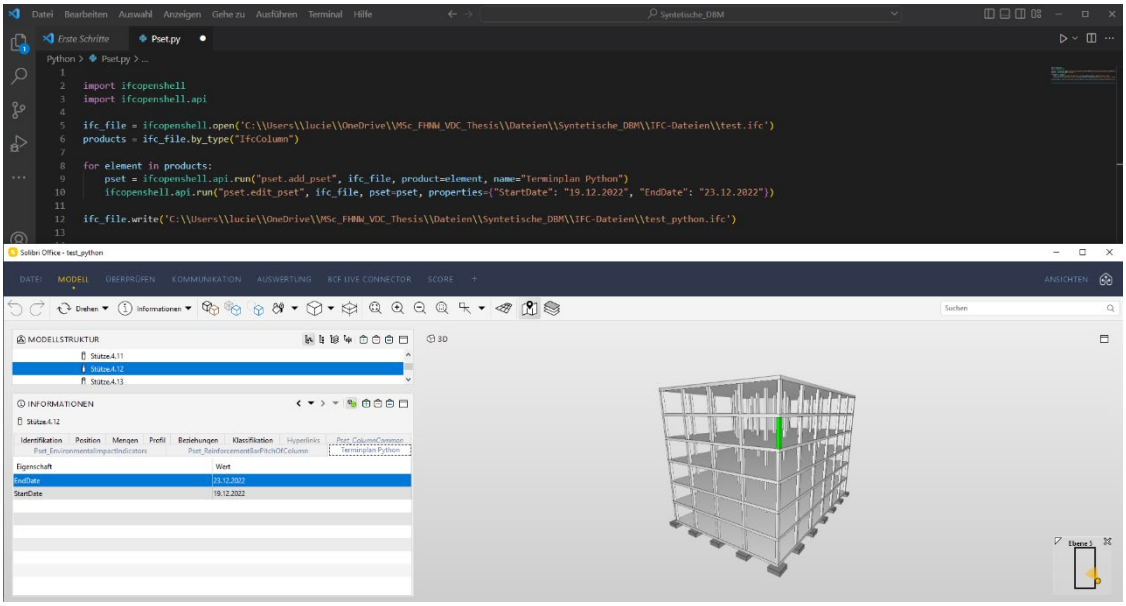


Abbildung 4: Erstellung eines PSet mit Hilfe eines Python-Skriptes (oben) und Darstellung in einem Model Viewer (unten)

Der Vorteil dieses auf Attributen basierenden Ansatzes besteht darin, dass er heute in der Praxis umgesetzt werden kann, da alle dem Autor bekannten Autorenwerkzeuge und Model Viewer PSet verarbeiten können. Grund dafür ist, dass die Hürde für Softwarehersteller entsprechende Funktionen zu implementieren, gering ist (Borrmann et al., 2021a). Der Nachteil hingegen ist, dass zwischen den einzelnen PSet keine Abhängigkeiten bestehen. Wird ein Terminplan angepasst muss für jedes Element einzeln geprüft werden, ob die im PSet hinterlegten Daten noch aktuell sind. Bei der Komplexität von Bauterminplänen und Menge an Elementen eines Bauwerks ist dieser Ansatz deshalb aufwändig und fehleranfällig.

Eine weitere Möglichkeit den auf Attributen basierenden Ansatz umzusetzen ist mit Hilfe von Common Data Environment (CDE) Lösungen, wie z. B. Revizto oder Trimble Connect. Diese erlauben es Nutzern eigene Attribute für Elemente zu erstellen. Die erstellten Eigenschaften können daraufhin als Listen oder Tabellen, allerdings nicht als IFC-Datei, exportiert werden und sind deshalb grundsätzlich an das jeweilige System gebunden. Die Verknüpfung der Elemente mit den zugehörigen eigenen Attributen erfolgt über den Global Unique Identifier (GUID). Der Nachteil dieses Vorgehens besteht darin, dass die erstellten Informationen verloren gehen, sollte sich die GUID eines Elements ändern. Der Vorteil ist, dass Projektbeteiligte ortsunabhängig, sofern sie über ein entsprechendes technisches Gerät verfügen, Attribute auf der CDE-Plattform anpassen können. (Ferraro, 2022; Schweizer, 2022).

6.1.3. 4D-IFC

In Kapitel 6.1.2 wurde die Möglichkeit beschrieben 4D-Informationen als PSet einem Bauelementen anzuhängen. Das IFC-Schema bietet allerdings spezifische Klassen, um Vorgänge, die eine zeitlichen Ausdehnung haben, abzubilden (Borrmann et al., 2021a). Abbildung 5 zeigt die für die Modellierung von Bauprozessen notwendigen Klassen. Eine der wichtigsten Klasen ist dabei der IfcTask, eine Subklasse des IfcProcess (Alex Braun et al., 2015). Dieser beschreibt ein einzelnes, abgegrenztes, und identifizierbares Arbeitspaket (Task) (buildingSMART, 2017). Um die in einem Bauprozess bestehenden Abhängigkeiten zwischen verschiedenen Tasks zu modellieren, werden Subklassen der Klasse IfcRelationship verwendet. Hiermit lassen sich Task-Task, Task-Bauelement sowie Task-Kontrollelement-Beziehungen erstellen (Hamledari et al., 2017).

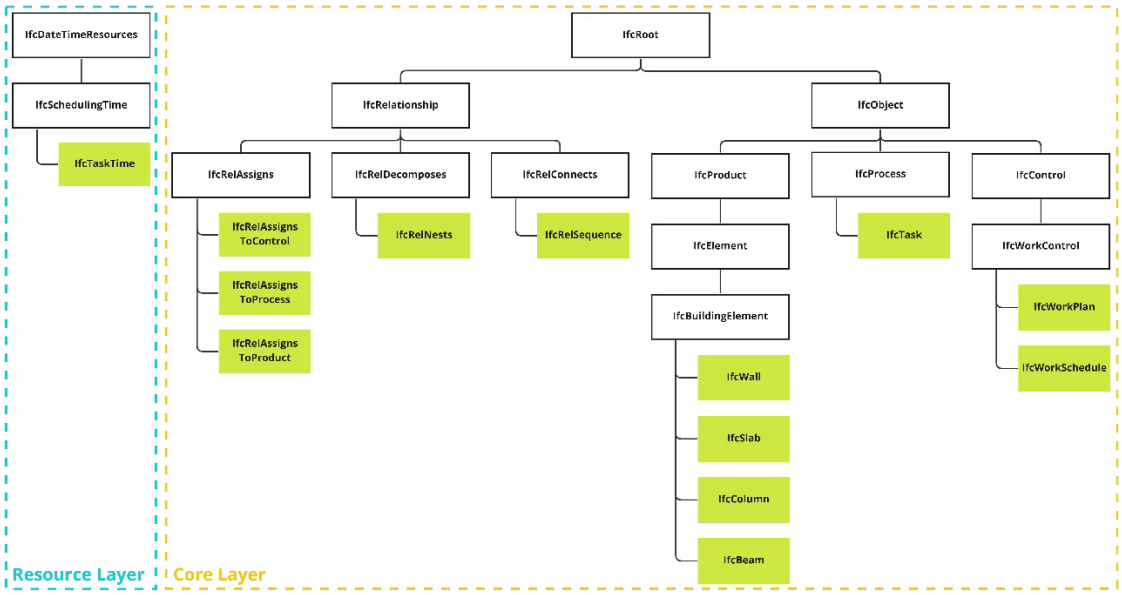


Abbildung 5: Ausschnitt IFC-Schema (IFC4). Für die Modellierung von Bauprozessen notwendige Klassen (grün).

IfcRelAssignsToProduct definiert eine Beziehung zwischen einem Task und einem Bauelement. IfcRelNests erlaubt es Tasks ineinander zu verschachteln, Indem ein übergeordneter Task (Root-Task) definiert wird, dem untergeordnete Tasks (Subtasks) zugeordnet werden. Der Root-Task wird ab IFC-Version 4 dazu verwendet eine Verbindung zur übergeordneten Klasse IfcWorkSchedule oder IfcWorkPlan herzustellen. IfcRelSequence erlaubt es verschiedenen Tasks in eine Reihenfolge zu setzen. Abbildung 6 zeigt schematisch, wie die Beziehungen Task-Bauelement und Task-Task umgesetzt werden können. (buildingSMART, 2017)

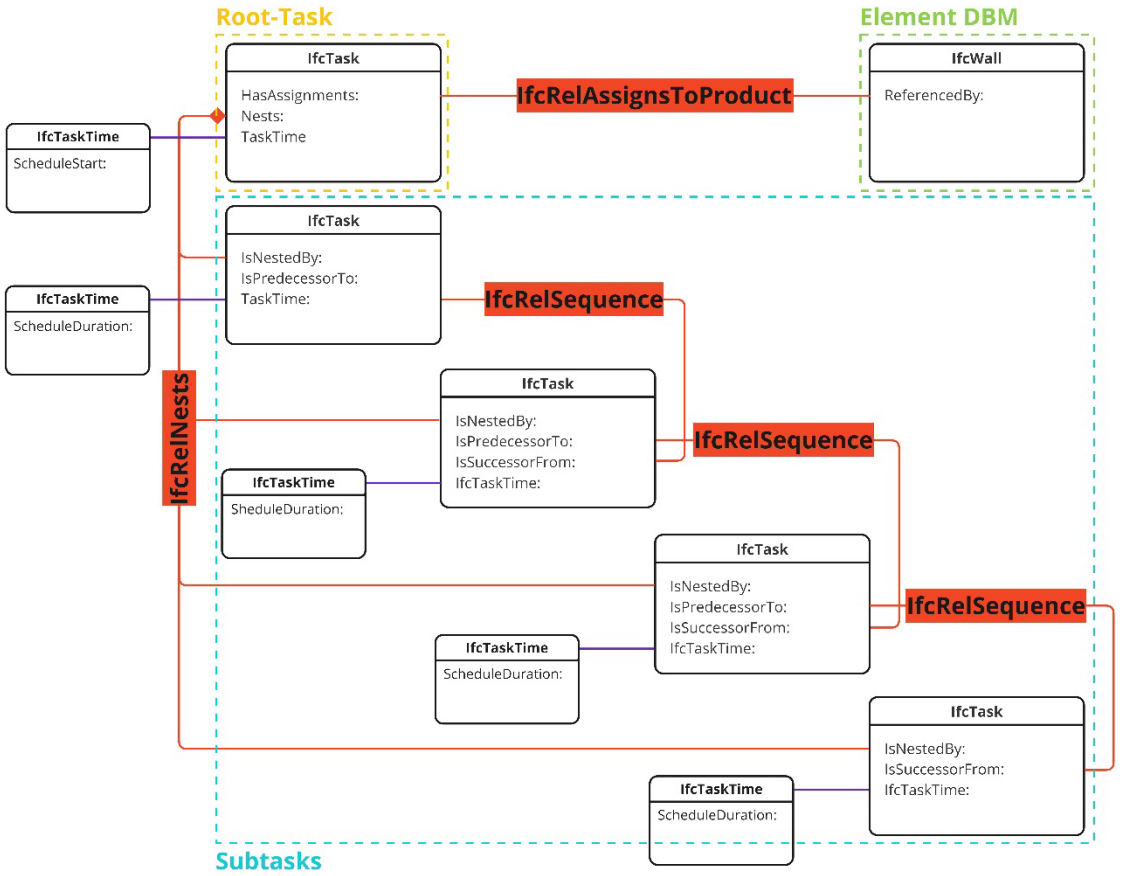


Abbildung 6: Schematische Darstellung von Verschiedenen Beziehungen die für einen Task modelliert werden können (IFC4). Beziehungen mit Hilfe von Subklassen der Klasse IfcRelationship (rot), Beziehung durch Attribute (lila Linien).

Um die zeitliche Ausdehnung eines IfcTasks zu modellieren, kommt die Klasse IfcTaskTime zum Einsatz. In Tabelle 3 sind die wichtigsten Attribute dieser Klasse aufgeführt. Die Beziehung zwischen der Klasse IfcTaskTime und IfcTask wird direkt über das Attribut TaskTime der Klasse IfcTask definiert (vgl. Abbildung 6) (buildingSMART, 2017). Es ist zu beachten, dass diese Klasse erst seit der IFC Version 4 existiert. In älteren Versionen wird die zeitliche Ausdehnung eines Tasks mit Hilfe der Klasse IfcScheduleControlTime, einer Subklasse der Klasse IfcControl, definiert (buildingSMART, 2005).

#	Attribut	Typ	Beschreibung
5	ScheduleDuration	IfcDuration	Geplante Dauer
6	ScheduleStart	IfcDateTime	Geplanter Start
7	ScheduleFinish	IfcDateTime	Geplantes Ende
17	ActualStart	IfcDateTime	Echter Start
18	ActualFinish	IfcDateTime	Echtes Ende
20	Completion	IfcPositiveRatioMeasure	Prozent fertiggestellt

Tabelle 3: Auswahl an Attributen der Entität IfcTaskTime (buildingSMART, 2017)

Bis jetzt ist die Möglichkeit 4D-Informationen mit Hilfe des IFC-Schemas zu modellieren weitgehend ungenutzt. In der berücksichtigten Literatur haben sich

lediglich Alex Braun et al. (2015) und Hamledari et al. (2017) für diesen Weg entschieden. Ein Grund dafür könnte sein, dass es nur wenig Software gibt, die IfcTasks, IfcProcesses etc. lesen, interpretieren und darstellen kann. Denn Softwarehersteller müssen sich nur gegen Teilbereiche des IFC-Schemas, sogenannte Model View Definitions (MVD) zertifizieren lassen und nicht gegen das gesamte IFC-Schema (Borrmann et al., 2021a). Wieso dieser Teil des IFC-Schemas oftmals ausgeklammert wird, kann im Zuge dieser Arbeit nicht beantwortet werden, da in der berücksichtigten Literatur keine Hinweise darauf gefunden wurden. Anzumerken ist jedoch, dass während der Recherche für diese Arbeit wurde, lediglich ein von Geometry Gym entwickeltes Plugin für Navisworks gefunden, dass es erlaubt, 4D-IFC-Dateien zu importieren (Geometry Gym, 2022). Bei Versuchen dieses Plugin anzuwenden konnten allerdings keine positiven Resultate erzielt werden. Der Entwickler gibt als Grund an, dass sich die Programmierschnittstelle (API) für Navisworks 2023 geändert habe (jonm, 2022). Auf Tests mit älteren Navisworks-Versionen wurde im Rahmen dieser Arbeit verzichtet.

Dementsprechend ist der grösste Nachteil dieser Herangehensweise, dass sie heute noch nicht in der Praxis angewendet werden kann. Der grosse Vorteil besteht darin, dass die Abhängigkeiten zwischen verschiedenen Arbeitspaketen modelliert werden können. Anpassungen im Terminplan, können damit leichter umgesetzt und allfällige Fehler oder Überschneidungen leichtert erkannt werden. Zudem setzt dieser Ansatz ganz auf das offene Austauschformat IFC, weswegen er gut für Open-BIM-Prozesse geeignet ist (Borrmann et al., 2021a).

6.1.4. Datenbankbasierter Ansatz

Park et al. (2017) haben einen web- und datenbankbasierten Ansatz zur Visualisierung und Auswertung von 4D-DBM entwickelt. Umgesetzt wurde dies mit Hilfe eines Webviewers der auf eine zentrale Datenbank (MS-SQL Server 2014) zugreift. Diese besteht aus drei Klassen: Terminplan, BIM-Model und WBS-BIM. Abbildung 7 zeigt die Beziehungen dieser Klassen als Entity Relationship Diagram (ERD).

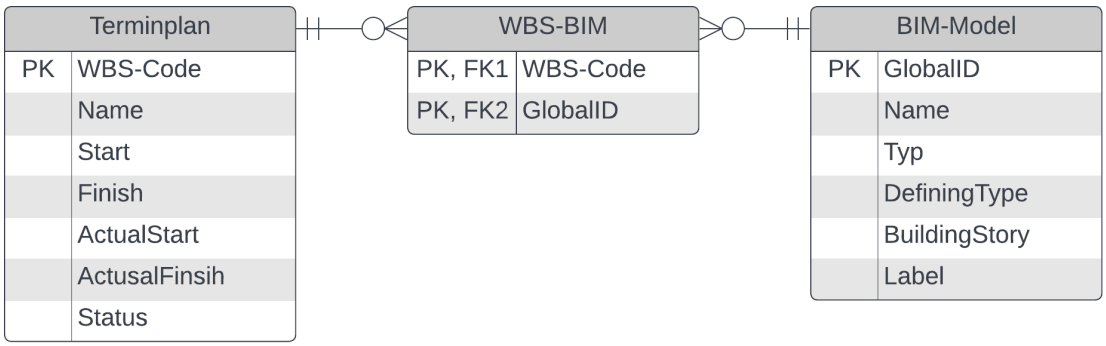


Abbildung 7: ERD der Beziehung zwischen Terminplan und BIM-Model (Park et al., 2017).

Die Klasse Terminplan besitzt 7 Attribute, wobei der WBS-Code den Primärschlüssel bildet. WBS steht für Work Breakdown Structure. Diese wird vom (Project Management Institute, 2021) als ergebnisorientierte hierarchische Zerlegung der vom Projektteam auszuführenden Arbeiten definiert. Die WBS oder zu Deutsch der Projektstrukturplan beschreibt also ein Vorgehen, bei dem Projekte in kleinere, plan- und kontrollierbare Teilaufgaben und Arbeitspakete unterteilt wird. Die WBS-Codes, in der von Park et al. (2017) entwickelten Datenbank, basieren auf der MasterFormat und UniFormat Klassifizierung. Abbildung 8 zeigt beispielhaft, wie sich der WBS-Code für Maurerarbeiten im ersten Obergeschoss des ersten Gebäudes zusammensetzt (ein Projekt kann aus mehreren Gebäuden bestehen). Die Elemente eines DBM werden mit Hilfe ihrer Attribute einem WBS-Code und damit einem Arbeitspaket innerhalb des Terminplans zugewiesen. Enthält der Name eines Elements z. B. das Wort Mauerwerk oder Ziegel kann dieses Element den Mauererarbeiten zugewiesen werden. Die Elemente mit ihren Attributen können direkt aus einer IFC-Datei extrahiert und in die Datenbank übernommen werden.

Abbildung 8: Beispiel WBS-Code (Park et al., 2017).

Wird die Datenbank aktualisiert oder das Systemdatum geändert, löst dies eine definierte Folge von Aktionen aus. Zum einen werden sechs verschiedenen Auswertungstabellen erstellt, die die Elemente des DBM nach ihrem Status ordnen: geplant, laufend, verzögert (später Start), verzögert (laufend), verzögert (nicht gestartet), fertig. Der Status ist ein Attribut der Klasse Terminplan und wird von den Attributen Start, Finish, LateStart und LateFinish abgeleitet. Zum anderen wird die Tabelle Terminplan dupliziert und unter einem neuen Namen gespeichert. Damit wird eine Momentaufnahme des aktuellen Standes archiviert. Dies dient der Dokumentation des Bauprozesses und bietet Nutzern die Möglichkeit verschiedene Zeitpunkte des Bauprozesses nachzuverfolgen und auszuwerten. (Park et al., 2017)

Ein Vorteil des von Park et al. (2017) entworfene Ansatzes ist, dass dieser heute schon in der Praxis anwendbar ist, wie sie mit ihren Fallstudien aufzeigen. Weiter sind die relevanten Informationen von allen Projektbeteiligten einsehbar und der Ansatz funktioniert im Zusammenspiel mit dem herstellerunabhängigen Austauschformat IFC. Damit ist er für Open-BIM-Prozesse geeignet. Zudem können Informationen, die in einer rationalen Datenbank gespeichert sind, ohne Informationsverluste weitergegeben und archiviert werden. Ein Nachteil ist, dass die Datenbank mit Hilfe der GUID mit den Elementen eines DBM verbunden sind. Verändert sich die GUID geht auch diese Verbindung verloren. Zudem ist nicht genau beschrieben, wie der Terminplan in die Datenbank integriert wird und wie schnell und mit wie viel Aufwand diese bei Änderungen angepasst werden kann.

6.2. Modellvergleich: As-Planned vs. As-Built

Yang et al. (2015) haben die verschiedenen Ansätze den Ist-Zustand (As-Built) eines Gebäudes mit dem Soll-Zustand (As-Planned) zu vergleichen, in zwei Kategorien geteilt: geometriebasierte und visuelle Ansätze. Geometriebasierte Ansätze umfassen diejenigen, die den Baufortschritt mit Hilfe von As-Built-Geometrien in einem dreidimensionalen Raum bestimmen. Visuelle Ansätze hingegen basieren auf Rückschlüssen, die aus 2D-Bilddaten gezogen werden. Auf die meiste berücksichtigte Literatur lässt sich diese Einteilung ohne Weiteres anwenden. Beide Ansätze büssen allerdings aufgrund der vielen temporären Strukturen, Baumaschinen und gelagertem Material, die die Sicht auf die Elementoberflächen verdecken, Genauigkeit bei der richtigen Erkennung von Elementen ein (Braun et al., 2021). Zudem liefern die meisten Methoden in der berücksichtigten Literatur nur boolesche Werte über den Fertigstellungsgrad eines Elements (gebaut / nicht gebaut). Diese Probleme gehen Braun et al. (2020 und Pal et al. (2022, 2021) mit verschiedenen Ansätzen, welche in Kapitel 6.2.3 genauer behandelt werden, ein.

Folgend wird vertieft auf die relevanten Ansätze der berücksichtigten Literatur eingegangen. Ansätze die auf Grundlage von digitalen Bautagebüchern oder automatisierter Liefererfassung (z. B. QR-Codes) basieren, wurden nicht weiter berücksichtigt, da diese keinen Mehrwert für das auf Bilddaten und Punktwolken basierende, Innosuisse Forschungsprojekt BMETRY schaffen.

6.2.1. Geometriebasierte Ansätze

Kielhauser et al. (2020) Kim et al. (2020) haben Ansätze entwickelt, bei denen aus Punktwolken ein Ist-Volumen generiert wird, welches anschliessend mit dem aus einem DBM abgeleiteten Soll-Volumen zu einem Zeitpunkt X verglichen wird. Aus allfälligen Differenzen lassen sich Rückschlüsse auf den Baufortschritt ziehen. Mit diesem Ansatz wird der Baufortschritt für das gesamte Bauwerk oder wie bei Kielhauser et al. (2020) für verschiedenen Sektionen gemessen. Dementsprechend können keine Rückschlüsse auf einzelne Elemente gemacht werden. Kielhauser et al. (2020) gehen bei ihrem Ansatz noch einen Schritt weiter, indem sie sowohl Aussagen zum Baufortschritt als auch zur Qualität des gebauten Zustandes machen. Hierzu werden das generierte Ist-Volumen und das Soll-Volumen übereinandergelegt. Diejenigen Stellen, an denen sich die Volumen unterscheiden, werden anschliessend markiert und klassifiziert. Dabei ist zu beachten, dass lediglich automatisiert erkannt werden kann, wenn sich ein Teil des Volumens an einer falschen Stelle befindet oder sich in Höhe, Länge oder Breite unterscheidet. Wieso diese Unterschiede bestehen, muss anschliessend manuell identifiziert werden.

Pučko et al. (2018) haben einen auf kontinuierlich mit Helmkameras generierten Punktwolken basierten Ansatz entwickelt. Der Vorteil dieses Ansatzes liegt vor allem

in der Art und Weise wie die Daten erfasst werden. Da die Helmkameras immer nur den Teil des Gebäudes, an welchem gearbeitet wird, aufnehmen, muss nicht bei jeder Auswertung das gesamte Gebäude berücksichtigt werden. Dies wirkt sich positiv auf die Rechengeschwindigkeit aus und erlaubt es den Baufortschritt in kürzeren Intervallen zu prüfen. Für den Modellvergleich wird in einem ersten Schritt die generierte Punktwolke aufgeräumt, indem im As-Planned-Modell um jedes Element eine leicht vergrösserte Bounding Box erstellt wird. Nur diejenigen Punkte der Punktwolke, die innerhalb der jeweiligen Bounding Box liegen, werden anschliessend für den Modellvergleich verwendet. Mit Hilfe einer an der Universität von Maribor entwickelten Software, werden anschliessend Elemente als gebaut verifiziert, wenn ein Drittel ihrer Oberfläche mit Punkten der As-Built-Punktwolke bedeckt ist (Rebolj et al., 2017).

Bognot et al. (2018) Reja et al. (2021) vergleichen eine As-Built-Punktwolke mit einer synthetisch generierten As-Planned-Punktwolke, um davon Aussagen über den Baufortschritt abzuleiten. Dazu wird eine leicht vergrösserte Bounding Box um jedes geplante Element des As-Planned-DBM erzeugt. Nur Punkte innerhalb der jeweiligen Bounding Box werden für das entsprechende Element berücksichtigt. Reja et al. (2021) vergleichen nun die Anzahl As-Built-Punkte mit der Anzahl der As-Planned-Punkte innerhalb einer Bounding Box. Das Verhältnis zwischen vorhandenen Punkten zu erwarteten Punkten ergibt anschliessend den prozentualen Fortschritt eines Elements. Bognot et al. (2018) hingegen berechnen für jedes Element die erwartete Point Cloud Density (vgl. Abbildung 9) sowie einen Objektschwerpunkt. Anhand dieser beiden Werte wurden statistische Schwellenwerte für jeden Elementtyp festgelegt. Liegen die festgestellten Werte eines Elements innerhalb dieser Schwellenwerte wird es als gebaut identifiziert. Das Resultat dieser Auswertung ist somit ein boolescher Wert.

$$Point\ Cloud\ Density = \frac{Anzahl\ Punkte}{Totale\ Elementfläche}$$

Abbildung 9: Formel für die Point Cloud Density (Bognot et al., 2018)

Auch Chen & Cho (2018) nutzen für ihren Ansatz eine synthetische As-Planned sowie eine generierte As-Built-Punktwolke. Zunächst werden die beiden Punktwolken mit einem Voxel-Raster überlagert. Es ist zu beachten, dass es sich um bereits registrierte Punktwolken handelt. Die beiden Voxel-Raster liegen deshalb exakt übereinander. Zudem kann jedes As-Built-Voxel mit Hilfe des entsprechenden As-Planned-Voxels einem bestimmten Element zugeordnet werden. Für jedes As-Built sowie As-Planned-Voxel wird nun mit Hilfe der jeweiligen Punktwolke ein Zustand definiert (besetzt / unbesetzt). Anschliessend wird aus dem Vergleich der zugehörigen As-Built und As-Planned-Voxel für jedes Element ein Fertigstellungsgrad berechnet. Dieser Fertigungsgrad ist höher für präzise konstruierte Elemente und niedriger für mangelhafte oder ausstehende Elemente.

Bariczová et al. (2021) haben eine Methode zur Verifizierung der Geometrie einer gebauten Wand im Vergleich zum geplanten Gegenstück entwickelt. Dazu werden aus einer IFC-Datei der Ursprung, die Höhe, die Dicke, die Länge und der Normalvektor einer Wand extrahiert. Aus diesen Informationen wird eine Wandfläche und ihr Mittelpunkt generiert. Über diese generierte Fläche der As-Planned-Wand wird nun die As-Built-Punktwolke gelegt. Ausgehend vom Punkt, der dem Mittelpunkt der generierten Fläche am nächsten ist, werden in verschiedenen Iterationen die naheliegendsten Nachbarn gesucht. Mit Hilfe der Ausrichtung der generierten Wandfläche kann bei jeder Iteration bestimmt werden, ob ein Punkt einen Teil der untersuchten Wand darstellt (vgl. Abbildung 10). Anschliessend wird die Geometrie der gebauten Wand verifiziert, indem die der Wand zugeordneten Punkte statistisch ausgewertet werden. Zum einen indem die Ausrichtung der Regressionsfläche mit derjenigen der generierten Fläche verglichen wird und zum anderen indem die Standardabweichung von Punkten zur generierten Fläche berechnet wird.

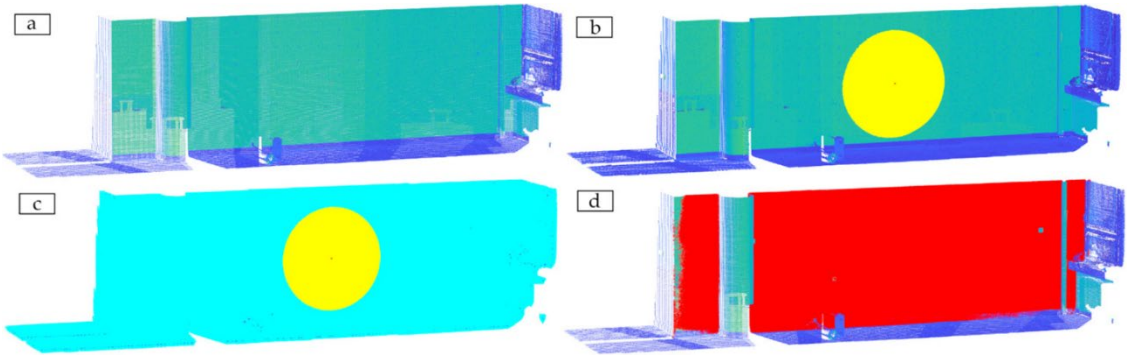


Abbildung 10: Verifizierung der Wandgeometrie: (a) Punktwolke, (b) Ausgangspunkt und nächste 102'400 Nachbarn, (c) alle nächsten Nachbarn, (d) der Wand zugeordnete Punkte (Bariczová et al., 2021).

Tuttas et al. (2015) haben eine Modellvergleichsmethode entwickelt, welche Grundlage für verschiedenen weitere Publikationen, wie Alizadehsalehi & Yitmen (2019), Braun et al. (2020, 2016, 2016) und Qu et al. (2017), war. Mit einem FWCI-Score von 4.2 ist dieser Ansatz damit einer der einflussreichsten (Scopus, 2015). Die von Tuttas et al. (2015) entwickelte Methode basiert auf einem Vergleich von mittels Bilddaten generierter Punktwolken mit einem 3D Meshmodell eines As-Planned-DBM. Dazu kommen zwei verschiedenen Raster zum Einsatz (vgl. Abbildung 11). Das Erste ist ein Voxelraster, welches über das gesamte betrachtete Areal gelegt wird. Es dient dazu festzustellen, welche Teile des Gebäudes durch die verwendetet Bilddaten verifiziert werden können. Umgesetzt wird dies mit Hilfe von Raycasting, da die Positionen der Kameras, die zur Bilddatenerfassung verwendet werden, bekannt sind. Das Ergebnis ist ein Voxelraster, bei welchem jedes Voxel entweder als frei, besetzt oder unbekannt klassifiziert ist. Das Zweite ist ein zweidimensionales Raster, dass über jedes Dreieck des 3D Meshmodells gelegt wird. Jede Zelle dieses Rasters wird nun beidseitig um die Länge *b* extrahiert. Dabei entsteht die in Abbildung 11 dargestellte Bounding Box. Um eine der Rasterzellen nun als existierend zu

verifizieren, werden die Punkte der As-Built-Punktwolke, welche innerhalb der entsprechenden Bounding Box liegen, anhand von drei Kriterien ausgewertet: Mittelwert, Standardabweichung zwischen dem orthogonalen Abstand der berücksichtigten As-Built-Punkte zur Elementoberfläche und Winkel zwischen der Normalen der Elementoberfläche und derjenigen der Regressionsfläche der As-Built-Punkte. Das Resultat der von Tuttas et al. (2015) durchgeführte Fallstudie zeigt, dass die sichtbaren Dreiecke der Elementoberfläche mit dieser Methode mit hoher Genauigkeit als gebaut oder nicht gebaut verifiziert werden können. Allerdings besteht ein grosser Nachteil darin, dass in 75% - 90% der Fälle keine Aussage gemacht werden konnte, weil die Oberfläche der Elemente nicht oder nicht genügend sichtbar war.

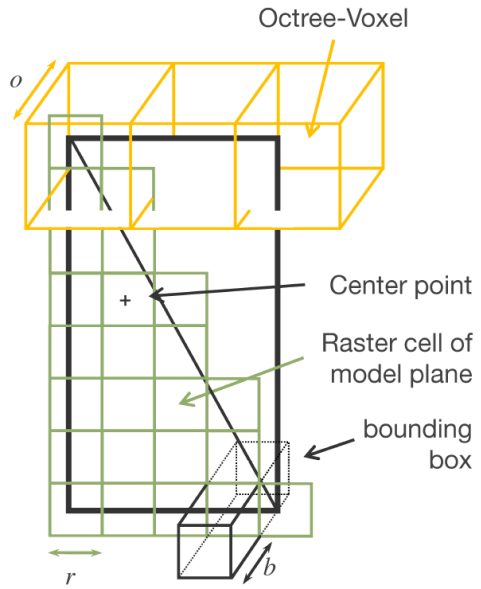


Abbildung 11: Konzep Modellvergleich: Voxelraster (gelb), 2D-Raster über Elementoberfläche (grün), Oberfläche eines Elements (schwarz dick), Bounding Box zur Rasterzellen Verifizierung (schwarz dünn) (Tuttas et al., 2015).

6.2.2. Visuelle Ansätze

Wang et al. (2021) haben eine Methode entwickelt, den Fortschritt von vorfabrizierte Betonelementen zu detektieren. Diese Methode basiert ausschliesslich auf Bilddaten von Überwachungskameras. In einem ersten Schritt werden mit Hilfe des Convolutional Neural Network (CNN) für Bildsegmentierung, Mask R-CNN, die vorfabrizierten Betonelemente in jedem Frame detektiert und segmentiert. Das Resultat ist ein Set von Bounding Boxes (vgl. Abbildung 12). Die Bounding Boxes werden anschliessend mit der Multy-Object-Tracking-Methode DeepSORT über aufeinanderfolgende Frames getrackt. Das Resultat dieses Schrittes umfasst die Pixel eines Elementes und deren Position in verschiedenen Frames. Die Verschiebung der einzelnen Pixel wird anschliessend analysiert. Ist der resultierende Wert unterhalb eines gewissen Schwellenwertes wird das Element als installiert klassifiziert. Daraufhin wird ein vierter Schritt ausgelöst. In diesem wird das installierte Element analysiert

und die Wandachse mit Hilfe eines Lokalisationsalgorithmus aus den 2D-Bilddaten extrahiert und in einen dreidimensionalen Raum überführt. Anschliessend wird die soeben generierte Elementposition sowie ein Zeitstempel als JavaScript Object Notation (JSON) Datei exportiert. Diese kann daraufhin mit einem selbst designten Plugin in Autodeskt Revit eingelesen werden. Mit Hilfe der Elementposition kann der Zeitstempel nun dem richtigen Wandobjekt in Autodesk Revit zugeordnet und als Attribut angehängt werden (vgl. Kapitel 6.1.2).

Abbildung 12: Detektion und Segmentierung von vorfabrizierten Betonelementen auf zwei verschieden Baustellen: (a) und (b) (Wang et al., 2021).

Vincke & Vergauwen (2022) haben ein Framework zur automatisierten Qualitätskontrolle des gebauten Zustandes entwickelt. Als Input für dieses Framework dienen auf der Baustelle aufgenommene Bilder sowie Passpunkte für die photogrammetrische Weiterverarbeitung. Dieser Input wird mit einer der ersten Phasen einer photogrammetrischen Pipeline, die Ausrichtung und Registrierung der Bilder, verarbeitet. Anschliessend werden für jedes Element diejenigen Bilder herausgesucht, auf denen es optimal zu erkennen ist. Anhand der im vorherigen Schritt bestimmten Kamerapositionen werden nun im geplanten DBM mehrere Bilder generiert. Aus diesen werden dann, mittels einer Brute-Force-Methode, diejenigen ausgewählt die am ehesten der Realität entsprechen. In einem letzten Schritt wird sowohl auf den As-Built als auch auf den As-Planned Bilddaten das zu untersuchende Element maskiert. Die beiden Masken werden anschliessend übereinandergelegt. Aus allfälligen Abweichungen können schlussendlich Rückschlüsse auf die Qualität des gebauten Elements geschlossen werden.

6.2.3. Den Modellvergleich unterstützende Methoden

Pal et al. (2022, 2021) haben einen auf Materialerkennung basierenden Ansatz entwickelt, Fertigungsschritte eines Elementes zu erkennen. Hierzu wird in einem ersten Schritt ermittelt, ob der Bau eines Elementes gestartet hat. Umgesetzt wird dies mit Hilfe von leicht vergrösserten Bounding Boxes, die um die As-Planned-Elemente eines DBM gelegt werden. Aus den darin liegenden Punkten einer As-Built-Punktwolke kann nun der Zustand eines Elementes abgeleitet werden. Wird ein

Element als gestartet klassifiziert, wird in Blender eine Kamera generiert, auf der die grösste Fläche des zu untersuchenden Elementes sichtbar ist. Von der generierten Kameraposition aus wird ein Bild aus der As-Built-Punktwolke gerendert und eine binäre Maske des Elements erstellt (vgl. Abbildung 13). Mit Hilfe der binären Maske kann das gerenderte Bild so zugeschnitten werden, dass nur noch das zu untersuchende Element sichtbar ist. Um die Bauphase des Elements zu bestimmen wurde ein CNN trainiert, dass jeden Pixel des Bildausschnittes klassifiziert. Die Klassifizierung entspricht dabei den Aktivitäten, die zum Bau eines Elements durchgeführt werden: Schalung stellen, Armierung legen usw. Um prozentual anzugeben, wie weit eine Aktivität fortgeschritten ist, wird die aufsummierte Fläche der einer Aktivität zugeordneten Pixel durch die totale untersuchte Fläche geteilt.

Abbildung 13: Bildgenerierung: (a) Kamerasetup in Blender, (b) gerendertes Bild der selektierten Kamera, (c) binäre Maske für zu untersuchendes Element (Pal et al., 2022).

Braun et al. (2020) haben ein mehrschichtiges Framework entwickelt, um die Erkennungsrate von Elementen bei geometriebasierten Ansätzen zu erhöhen. Damit das Framework funktioniert ist es wichtig, dass die As-Built-Punktwolke mit Hilfe eines Structure-from-Motion-Prozesses generiert wird, weil Bilddaten und Kamerapositionen wichtiger Bestandteil des Frameworks sind. Weiter wird ein As-Planned-DBM mit verschiedenen semantischen Informationen zu jedem Element vorausgesetzt. In einem ersten Schritt wird versucht ein Element mit Hilfe der As-Built-Punkte innerhalb einer As-Planned Bounding Box, die um jedes Element generiert wird, zu detektieren. Gelingt dies nicht, das Element wird allerdings aufgrund des Terminplans erwartet, wird eine von Braun & Borrmann (2019) entwickelte Methode angewendet, mit der anhand von Bilddaten und korrespondierenden Kamerapositionen identifiziert werden kann, welche Elemente auf den verwendeten Bilddaten sichtbar sind. Dieser Schritt hilft dabei, die Fehlerquote bei der Elementerkennung zu reduzieren. Sollte das Element als detektierbar klassifiziert worden sein, vergrössern Braun et al. (2020) in einem nächsten Schritt die Bounding Box. Damit wird geprüft, ob das Element eventuell eingeschalt ist. Ergibt diese Prüfung ein positives Resultat, wird das Element als «im Bau» markiert. Kommt es auch hierbei zu einem negativen Resultat, kommt ein CNN zum Einsatz. Hierfür wird das zu untersuchende Element auf einem Bild, auf welchem es gut zu sehen ist, mit Hilfe des daruntergelegten DBM, maskiert. Das CNN versucht nun verschiedene Materialen anhand der Farbe zu detektieren, um Hinweis auf die Existenz des Elements zu sammeln. Kommt auch dieser Schritt zu einem negativen Resultat, wird das Element

als «nicht identifiziert» markiert. Eine weitere Methode die Erkennungsrate mit dem entwickelten Framework zu erhöhen, ist die Verwendung eines Precedence Relationship Graphs (PRG). Dieser wurde von Alex Braun et al. (2015) erstmals eingeführt und bildet die Abhängigkeiten, die zwischen Elementen eines Gebäudes bestehen, ab. Der PRG hilft die Erkennungsrate zu erhöhen indem einige Komponenten, die sogenannten Checkpoint-Komponenten, identifiziert werden (Braun et al., 2021). Auf diese Komponenten folgende Elemente können nur gebaut werden, wenn das vorhergehende Element fertiggestellt ist (vgl. Abbildung 14). Beispielsweise kann eine Stütze nur dann gebaut werden, wenn die darunterliegende Stütze oder Bodenplatte fertiggestellt ist. Anhand dieser Abhängigkeiten lässt sich die Existenz nicht detektierter Elemente ableiten, wenn ein darauffolgendes Element als «im Bau» oder «fertiggestellt» identifiziert wird.

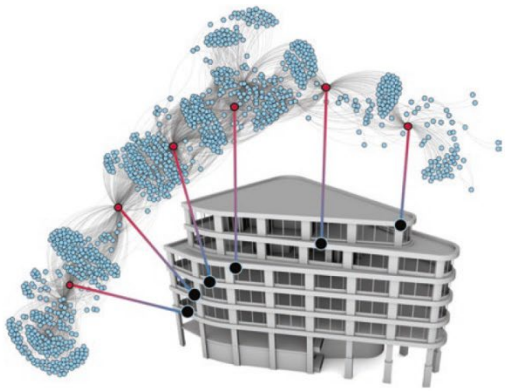


Abbildung 14: Konzept Precedence Relationship Graph: Checkpoint-Komponenten (rot) (Braun et al., 2021).

6.3. Darstellung & Auswertung

Der Baufortschritt im Hochbau wird aktuell in der Praxis als Balkendiagrammen und in seltenen Fällen als Liniendiagramm dargestellt. Bei Balkendiagrammen wird der Soll-Balken einer geplanten Tätigkeit digital in einer Projektmanagementsoftware oder analog auf einem ausgedruckten Terminplan gespalten, so dass der Ist-Zustand darunter eingezeichnet werden kann. Der Baufortschritt kann anschliessend aus den unterschiedlichen Balkenlängen abgeleitet werden. Für Liniendiagramme wird hingegen die Soll- und Ist-Steigung einer Linie verglichen, um Aussagen über den Baufortschritt zu treffen. Beide Methoden können bei grösseren Projekten schnell unübersichtlich werden und stellen nur selten die unterschiedlichen Leistungen zur Erstellung eines Elementes dar. (Blaser Zürcher, 2022; Martin, 2022; Riecke, 2022; Schweizer, 2022)

Im Gegensatz zu den aktuell angewendeten Methoden bietet die Darstellung und Auswertung des Baufortschritts anhand von DBM einige Vorteile. So können die vorliegenden Informationen damit einfach verständlich und dennoch detailliert, den verschiedenen Anspruchsgruppen präsentiert werden (Eastman et al., 2011). Zudem

lässt sich der Ist- und Soll-Zustand eines Bauwerks mit Hilfe verschiedener Software simulieren, was helfen kann Rückschlüsse auf den weiteren Projektverlauf zu ziehen (Crowther & Ajayi, 2021).

Eine weitere Methode mit welcher der weitere Projektverlauf prognostiziert werden kann wurde von Kim et al. (2020) angewendet. Bei ihrer Methode zum Modellvergleich werden der Ist- und Soll-Zustand anhand von gemessenen Gebäudevolumen festgestellt. Ausgewertet wird der daraus abgeleitete Baufortschritt mit Hilfe des Earned Value Management Systems (EVMS) (vgl. Abbildung 15). Hiermit lässt sich der aktuelle Baufortschritt darstellen und das Projektrisiko in Bezug auf Zeit und Projektkosten bewerten (Vandevoorde & Vanhoucke, 2006).

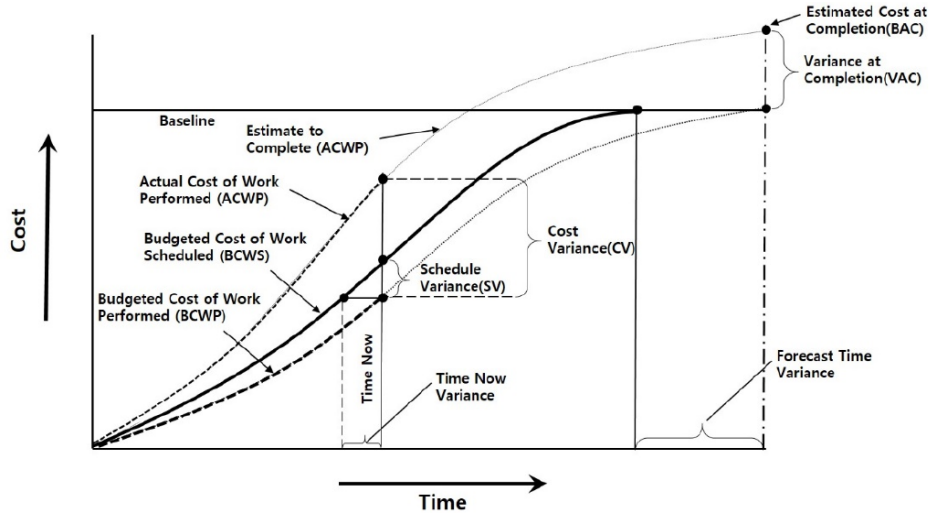


Abbildung 15: EVMS-Diagramm mit S-Kurve (Kim et al., 2020).

Eine weitere Anwendung des EVMS lässt sich bei Alzraiee (2018) finden. Hierbei wird das EVMS gemeinsam mit der Rules of Credit (RoC) Methode verwendet. Die RoC-Methode basiert auf definierten Meilensteinen und deren prozentualem Fortschritt (Oracle Primavera, 2022). Abbildung 16 zeigt einen Ausschnitt der von Alzraiee (2018) entwickelten RoC-Methode. Hierfür werden alle für die Methode wichtigen Informationen aus einem DBM extrahiert. Anschliessend schätzen die Kalkulatoren die Material- und Arbeitskosten und übertragen diese in die entsprechenden Felder. Jedes Arbeitspaket ha eine eindeutige Identifikation, die für die Fortschrittsverfolgung sowie Abrechnungen genutzt wird.

Cost Code	Unique Identifier	Item	Discipline	Rules of Credit	Weight	Budget Cost ($)	Estimated Hr	Actual Cost ($)	Earned Value	Progress %
01-xxx-xx-xxx			**Mobilization**	Start	50%					
				Complete	50%					
02-xxx-xx-xxx			**Procurement**	Issue purchase order	25%					
				Receive Vendor Drawings	25%					
				Fabrication/ Delivery	40%					
				Received by Contractor	10%					
03-xxx-xx-xxx			**Fabrication**	Material Received	10%					
				Fabrication	80%					
				Testing	5%					
				Transportation	5%					

Abbildung 16: Anwendung der RoC-Methode auf ein Bauprojekt (Alzraiee, 2018).

Die naheliegendste und in der berücksichtigten Literatur meist verwendete Darstellungsform des Baufortschritt basiert auf Farbcodes, mit welchen die Elemente eines DBM eingefärbt werden können. Dabei unterscheiden sich Ansätze lediglich in der Anzahl der Abstufungen, die gewählt wurden. Am unteren Ende des Spektrums befinden sich Bognot et al. (2018) mit einem binären Farbcode (gebaut / nicht gebaut) und Qu et al. (2017) mit einer dreiteiligen Klassifizierung (gebaut / nicht gebaut / unbekannt). Detailliertere Farbcodes wurden von Park et al. (2017), die eine fünfstufige Einteilung verwendet haben (fertiggestellt / verzögert (nicht gestartet) / verzögert (überfällig) / laufend / geplant) und Hamledari et al. (2017), die sich für einen Farbverlauf basierend auf einem berechneten Leistungsindex entschieden haben (hinter dem Terminplan: Grün bis Rot, vor dem Terminplan: Grün bis Blau). Auch die Gesprächspartner für die Fachgespräche haben positiv auf diese Art der Darstellung reagiert und sehen darin grosses Potential.

Umgesetzt wurde der Farbcode auf verschiedenen Weisen. Alexander Braun et al. (2015), Golparvar-Fard et al. (2015) und Park et al. (2017) haben eigene Webviewer entwickelt, um die DBM inklusive darzustellen (vgl. Abbildung 17). Hamledari et al. (2017) haben die Elemente in einer IFC-Datei mit Hilfe des Attributs Styles der Klasse IfcSurfaceStyle übersteuert. Der Farbcode ist damit in jeder IFC-kompatiblen Software sichtbar.

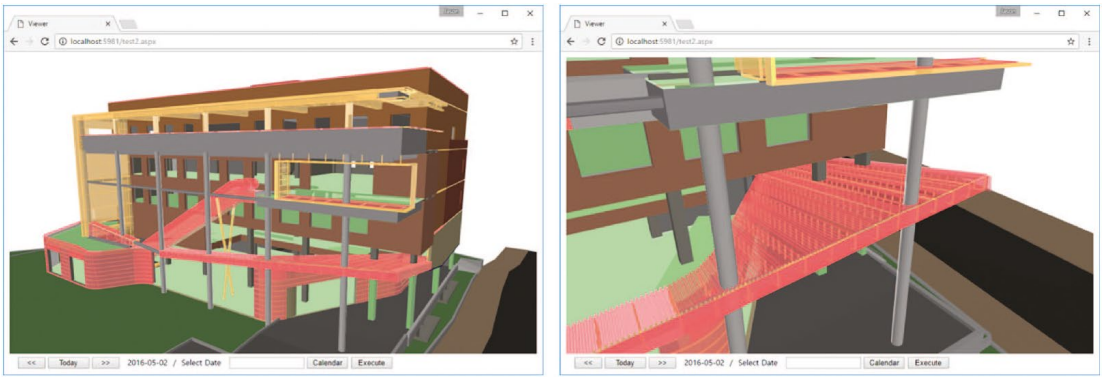


Abbildung 17: Ein in einem Farbcode dargestelltes DBM, im von (Park et al., 2017) entwickelten Webviewer.

7. Konzept

Das in dieser Arbeit entwickelte Konzept für eine automatisierte Baufortschrittsermittlung und -kontrolle baut auf der aktuell in BMETRY entwickelten Lösung auf. Diese gliedert sich in vier Teilprojekte (vgl. Abbildung 18). Dabei umfasst das Teilprojekt «4D Imagery» die automatisierte Datenerfassung und Generierung georeferenzierter photogrammetrischer Produkte. Darauf aufbauend werden in «4D Extraction» relevante Bauelemente und Einlagen als semantische Objekte extrahiert. Um einen Soll-Zustand, als Grundlage für verschiedene Auswertungen, wie den aktuellen Baufortschritt, abzubilden, wird in «4D Building Model» untersucht, wie zeitliche Informationen in DBM modelliert werden können. Die Resultate dieser drei Teilprojekte fliessen «4D Viewer» zusammen. Hierbei wird die zentrale Plattform für die Visualisierung der erfassten, extrahierten und generierten Daten entwickelt. (Schneider et al., 2021)

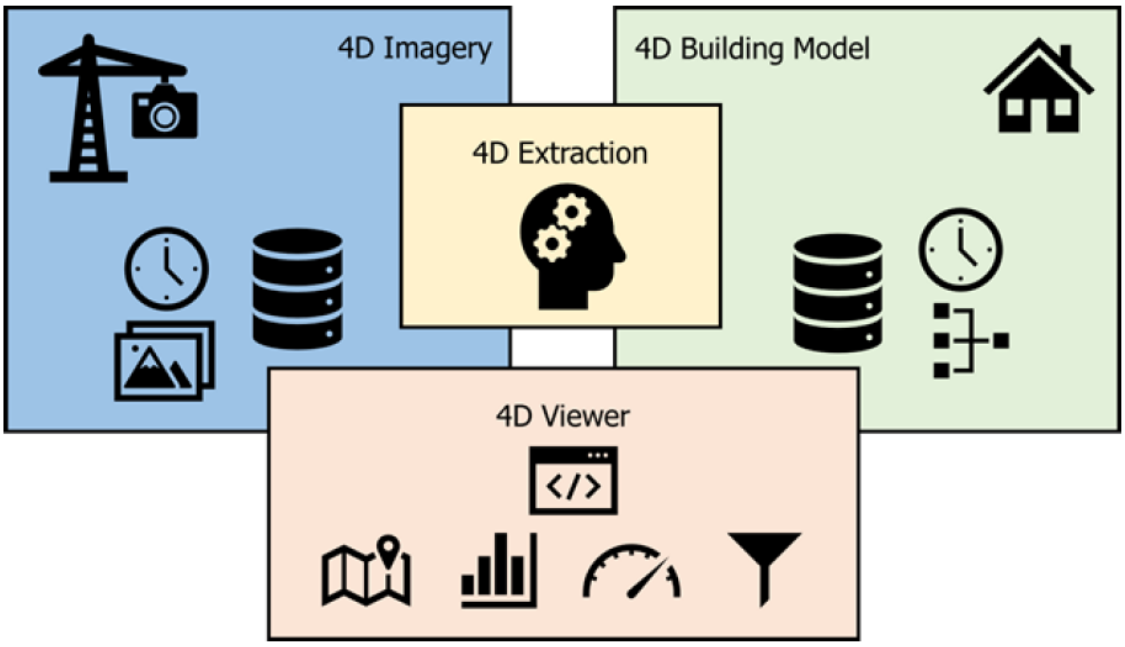


Abbildung 18: Teilprojekte BMETRY (Schneider et al., 2021).

Der in dieser Arbeit entwickelte Workflow besteht seinerseits aus zwei Teilprojekte: 4D-DBM und Modellvergleich. Die Generierung eines 4D basiert auf dem von Park et al. (2017) entwickelten datenbankbasierten Ansatz. Dieser Ansatz wurde gewählt, weil relationale Datenbankmodelle ein etablierter Standard in der Praxis sind. Daher können die darin enthaltenen Informationen von verschiedenen Softwarelösungen interpretiert und anschliessend weiterverarbeitet oder dargestellt werden. Damit ist dieser Ansatz gut für Open-BIM-Prozesse geeignet, da alle Disziplinen Zugriff auf die in der Datenbank enthaltenen Informationen haben (Nigel & Eduard, 2021). Weiter kommen Nigel & Eduard (2021) zum Schluss, dass datenbankbasierten Ansätze für die Abwicklung von BIM-Prozessen die Datenredundanz in einem Projekt verringert

und die Datensicherheit erhöhen. Der Modellvergleich erfolgt geometriebasiert auf den in «4D Extraction» generierten Modellen und wird in Kapitel 8.2 genauer beschrieben.

In Abbildung 19 ist dargestellt, wie die Systemarchitektur zur Umsetzung des erarbeiteten Konzepts aussehen könnte. Der Webviewer entspricht hierbei der im Teilprojekt «4D Viewer» von BMETRY entwickelten zentralen Plattform. Ihr wird ein CDE zu Grunde gelegt, in welchem alle relevanten Daten abgelegt und gespeichert sind. Dieses wiederrum besteht aus einer rationalen Datenbank, um Information in Relation zueinander zu speichern und einem Dateisystem als zentrale Dateiablage. Für eine automatisierte Baufortschrittsermittlung und -kontrolle ist in der Datenbank ein Terminplan hinterlegt, während im Dateisystem Soll- / Ist-DBM abgelegt. Aktuelle Elementeigenschaften, wie Mengen und Spezifikationen, können mit Hilfe von Skripten aus den IFC-Dateien extrahiert werden und anschliessend in die Datenbank inklusive aller relevanten Relationen eingepflegt werden. Zusätzlich gibt es eine Komponente für den Modellvergleich. Diese hat die As-Plannd-IFC Datei sowie die As-Built-IFC-Datei, welche im BMETRY Teilprojekt «4D Extraction» generiert wird, als Input und schreibt ihren Output, den Baufortschritt pro Element, direkt in die Datenbank. Dort werden die Informationen verarbeitet und ausgewertet, um sie schliesslich den Nutzern, im Webviewer, zu visualisieren.

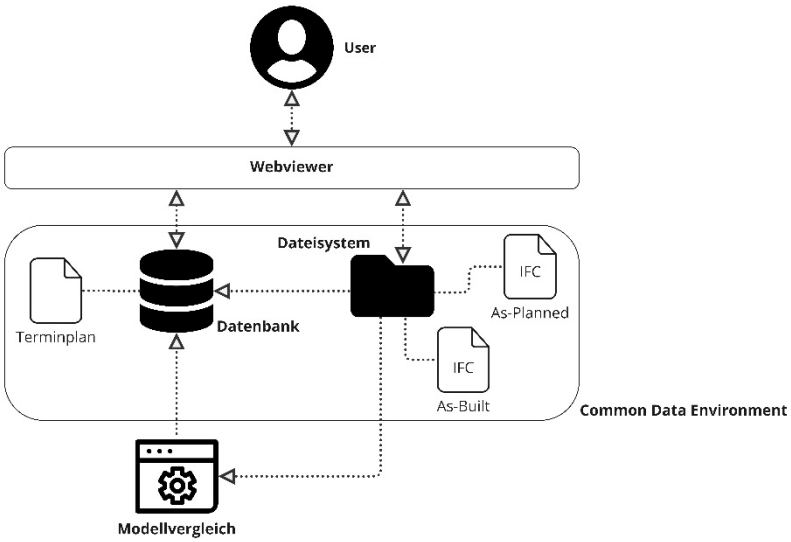


Abbildung 19: Konzept Systemarchitektur

In Abbildung 20 wird aufgezeigt, wie die einzelnen Komponenten dieses Konzepts interagieren, wenn für die Baufortschrittsermittlung relevante Daten aktualisiert werden. Dafür wird zwischen drei verschiedenen Szenarien unterschieden: (a) Update Ist-DBM, (b) Update Soll-DBM und (c) Update Terminplan. (a) und (b) sind im Grundsatz sehr ähnlich. Dieser Workflow wird ausgelöst, wenn eine neue Ist- oder Soll-IFC-Datei hochgeladen wird. Um den Projektverlauf zu dokumentieren und Änderungen im Nachhinein nachvollziehen zu können, wird zuerst die aktuelle Datei

archiviert, bevor sie durch die neue Datei ersetzt wird. Daraufhin wird die Modellvergleichskomponente ausgelöst, um den aktuellen Baufortschritt pro Element zu bestimmen. Das Ergebnis wird anschliessend an die Datenbank weitergegeben, welche ihrerseits die Tabelle, in welcher der Fortschritt pro Element gespeichert ist, archiviert, bevor sie mit den neuen Informationen aktualisiert wird. Szenario (c) beschreibt den Workflow, wenn der Terminplan angepasst wird. Dies hat zur Folge, dass der aktuelle Terminplan archiviert wird, bevor er durch den neuen ersetzt wird. Wird der Terminplan aktualisiert ist kein neuer Modellvergleich nötig, da bei diesem der Fortschritt pro Element, an einem bestimmten Stichtag, festgestellt wird, wobei der Terminplan dazu nicht relevant ist. Deshalb können bei einer Änderung des Terminplans alle resultierenden Veränderungen innerhalb der Datenbank erfolgen.

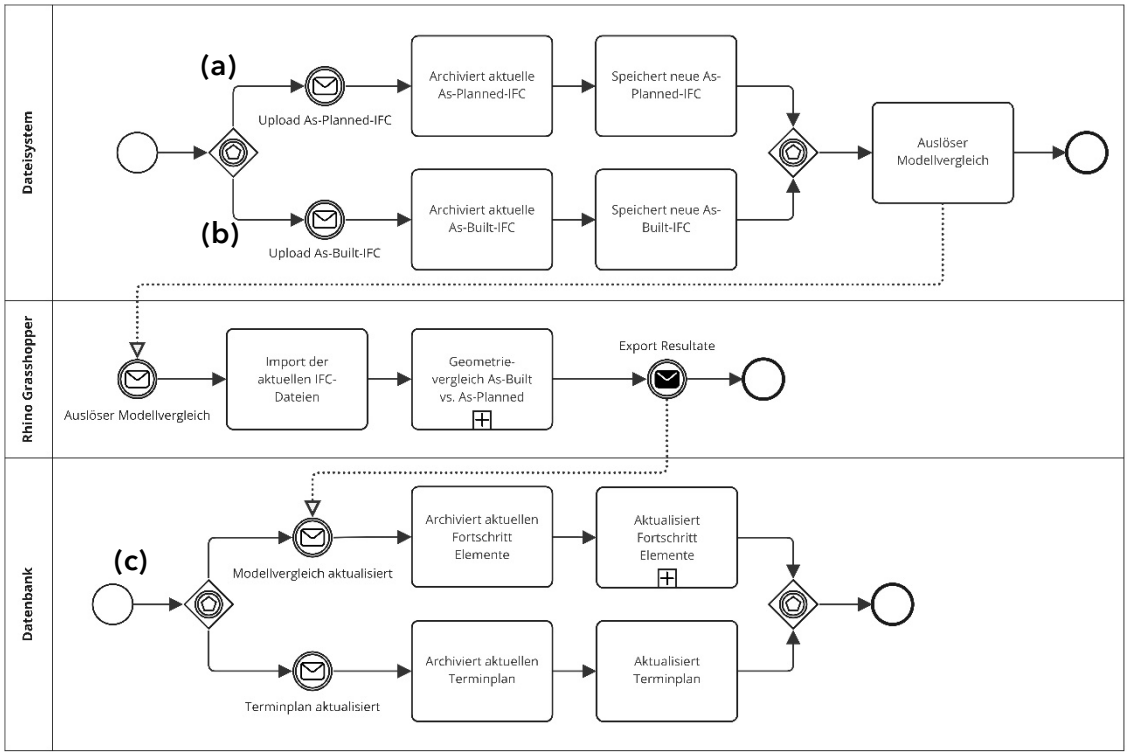


Abbildung 20: Workflow Datenaktualisierung: (a) & (b) Update IFC-Dateien, (c) Update Terminplan.

Abbildung 21 zeigt den Workflow, wenn ein Nutzer auf den Webviewer zugreift. Hierbei werden in einem ersten Schritt alle Auswertungen in der Datenbank archiviert und anschliessend aktualisiert. Basierend auf diesen Auswertungen und dem im Dateisystem abgelegten Soll-DBM generiert der Webviewer eine farbcodierte Ansicht des Ist-Zustandes sowie Statistiken, die helfen, den Baufortschritt besser interpretieren zu können. Anschliessend werden diese Informationen dem Nutzer in dynamischen Ansichten präsentiert.

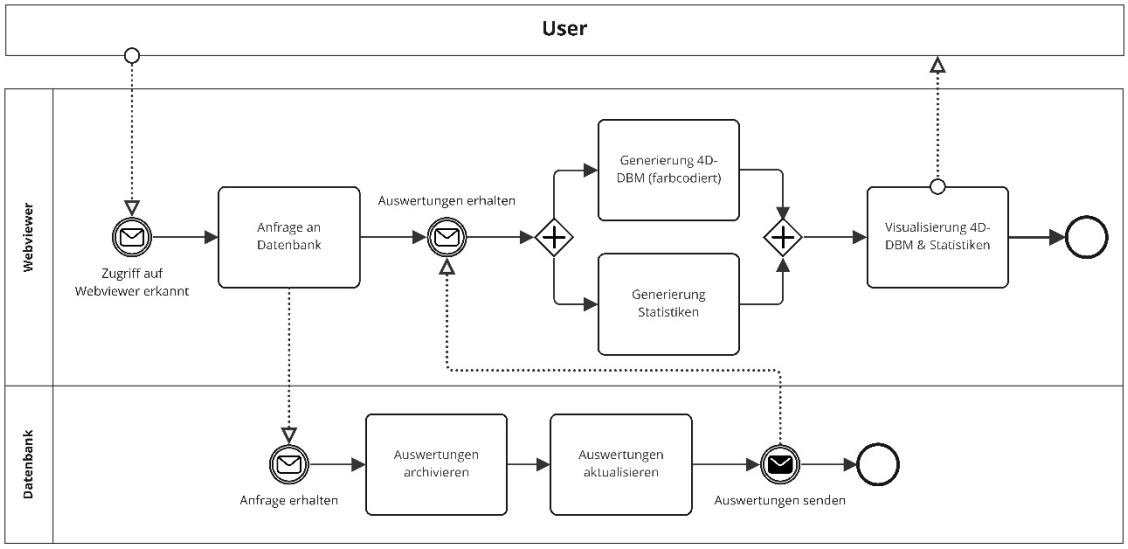


Abbildung 21: Workflow User-Anfrage

8. Proof of Concept

Die Funktionalität des in Kapitel 7 beschriebenen Konzepts wurde als Proof of Concept (PoC) in vereinfachter Form umgesetzt. In Abbildung 22 sind die einzelnen Bestandteile und ihre Beziehung zueinander dargestellt. Für die Umsetzung des PoC wurde teilweise auf bestehende Softwarelösungen, wie MS Access als Datenbank sowie MS PowerBI und Autodesk Navisworks zur Visualisierung, gesetzt. Dadurch wurde der eigene Entwicklungsaufwand minimiert, indem softwarespezifische Funktionen zur Umsetzung genutzt werden konnten. Dennoch mussten zusätzlich eigene Skripte in Python und Grasshopper geschrieben werden, um spezifische Bestandteile des PoC umzusetzen.

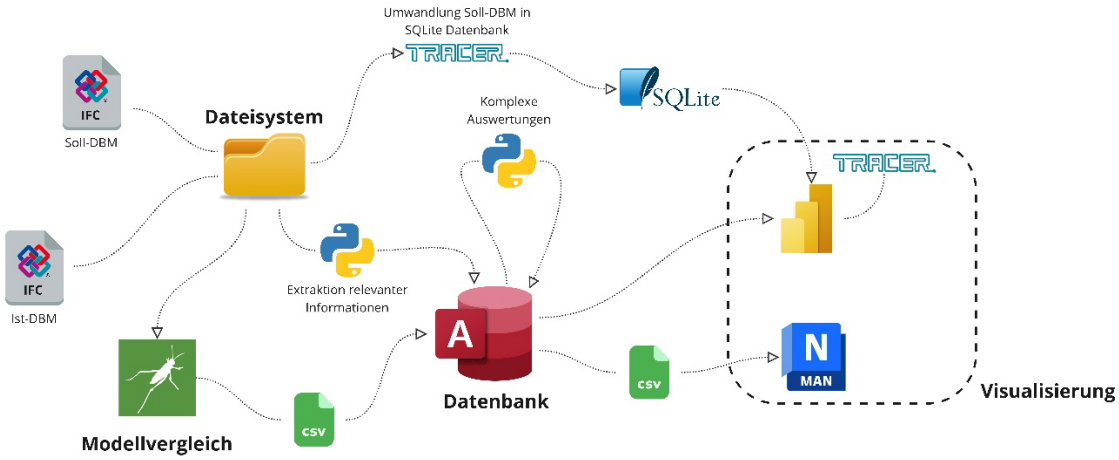


Abbildung 22: Schematische Darstellung der einzelnen Bestandteile des PoC und ihre Beziehungen zueinander.

Für das PoC wurde das Szenario «Update Ist-DBM» umgesetzt. Der entsprechende Workflow ist in Abbildung 23 dargestellt. In einem ersten Schritt werden das Soll- und Ist-DBM als IFC-Datei in Grasshopper importiert, um die beiden Modell geometrisch zu vergleichen. Die daraus resultierenden Resultate werden als CSV-Datei exportiert. Die entsprechende Datei ist direkt mit der MS-Access Datenbank verknüpft, weshalb sie automatisiert aktualisiert wird. Mit den aktualisierten Informationen werden anschliessend die Auswertungen aktualisiert. Diese sind direkt mit MS PowerBI verknüpft und können dort zusammen mit dem DBM in einem interaktiven Dashboard visualisiert werden (vgl. Kapitel 8.6). Zusätzlich wird eine CSV-Datei exportiert, welche in Autodesk Navisworks eingelesen werden kann, um den Soll- und Ist- Zustand des Baufortschritts zu simulieren (vgl. Kapitel 8.6).

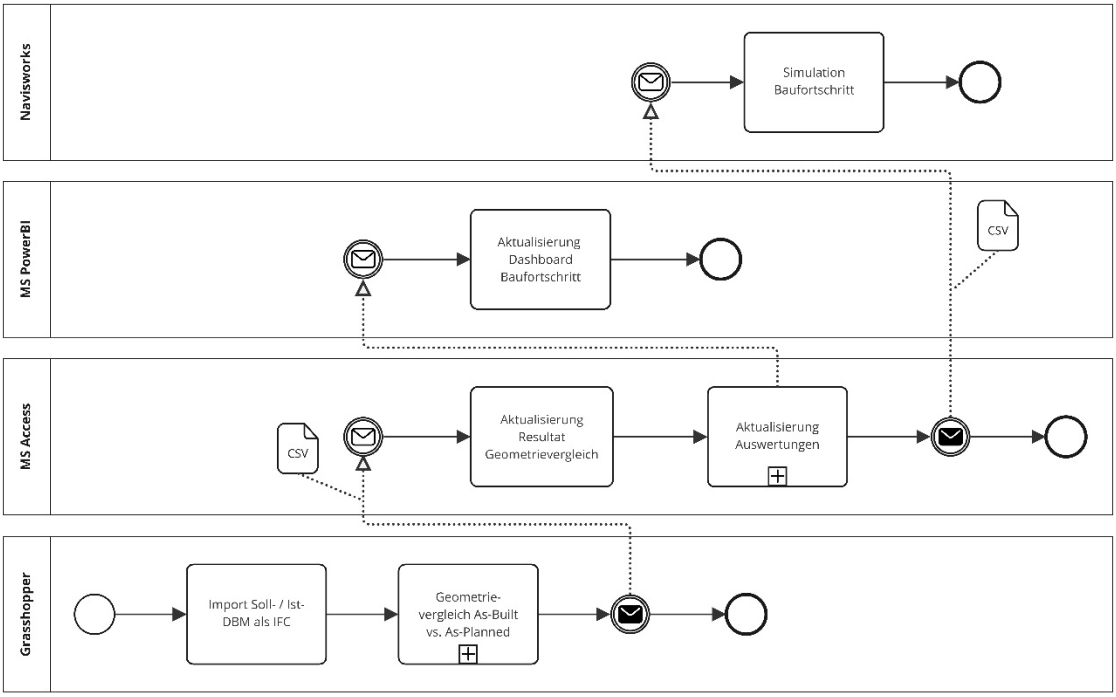


Abbildung 23: Workflow PoC für das Szenario «Update Ist-DBM».

8.1. As-Planned- & As-Built-DBM

Um den Soll-Zustand als Modell abzubilden, wurde das in Abbildung 24 dargestellte synthetische As-Planned-DBM erstellt. Für den PoC wurden synthetische Modelle verwendet, um mehr Kontrolle über die verwendeten Daten zu haben. Bei dem erstellten Gebäude handelt es sich einen dreistöckigen Betonbau, der sich aus verschiedenen Elementtypen zusammensetzt: Fundationen, Bodenplatten, Wände und Stützen. Die Elemente sind alle einschichtig modelliert, um den umgesetzten Modellvergleich zu vereinfachen. Es ist jedoch denkbar den Ansatz auch für mehrschichtige Bauteile zu verwenden, wenn die entsprechenden Features in den Modellvergleich integriert werden (vgl. Kapitel 8.7).

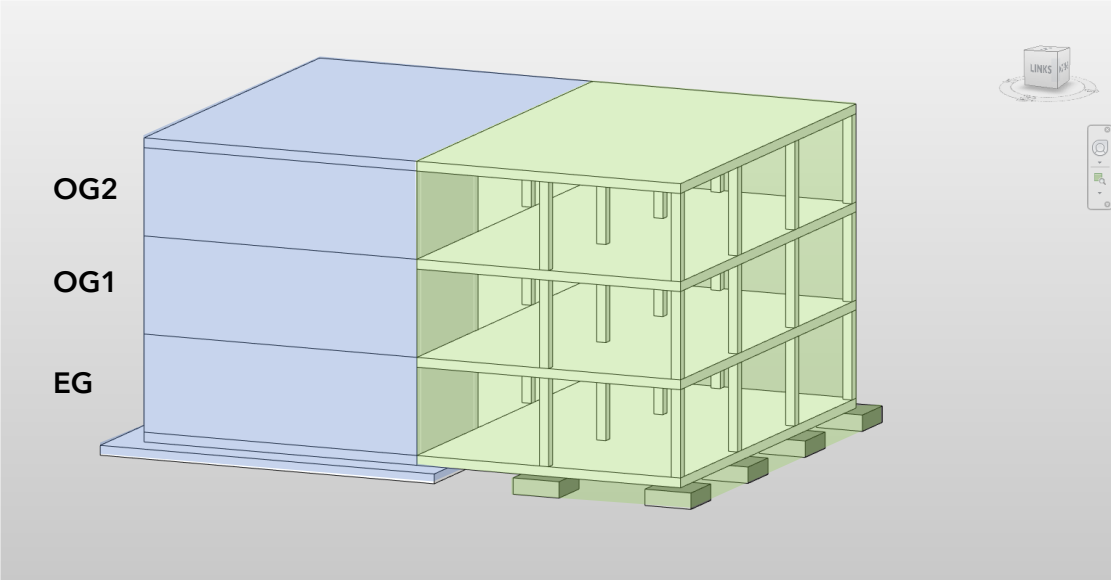


Abbildung 24: As-Planned-DBM für PoC: Zone Nord (blau) und Zone Süd (grün).

Für das synthetische As-Planned-DBM wurde ein vereinfachter Bauablauf definiert. In diesem werden Elemente anhand des Geschosses, Typs und der Zone, in der sie liegen zu Etappen zusammengefasst. Die Etappen werden daraufhin nacheinander errichtet, was schlussendlich zum in Tabelle 4 abgebildeten Terminplan führt. Die Zone, zu welcher ein Bauteil gehört, muss in der Autorensoftware, Autodesk Revit, jedem Bauteil manuell zugeordnet und anschliessend als eigenes PSet exportiert werden (vgl. Kapitel 6.1.2). Elemente, die in die in zwei verschiedenen Zonen liegen würden, wie die Bodenplatten, müssen so modelliert werden, dass sie eindeutig einer Zone zugeordnet werden können. Schlussendlich wird das As-Planned-DBM als IFC-Datei exportiert, damit es für den Modellvergleich verwendet werden kann. Dabei ist es wichtig, dass die Bauteilmengen und eigenen PSet mitexportiert werden.

Vorgang	Dauer	Anfang	Ende
EG	**34**	**02.01.2023**	**16.02.2023**
Fundation Zone Nord	3	02.01.2023	04.01.2023
Fundation Zone Süd	3	05.01.2023	09.01.2023
EG Bodenplatte Zone Nord	4	10.01.2023	13.01.2023
EG Bodenplatte Zone Süd	4	16.01.2023	19.01.2023
EG Wände Kern	5	20.01.2023	26.01.2023
EG Stützen Zone Nord	3	27.01.2023	31.01.2023
EG Stützen Zone Süd	4	01.02.2023	06.02.2023
Geschossdecke über EG Zone Nord	4	07.02.2023	10.02.2023
Geschossdecke über EG Zone Süd	4	13.02.2023	16.02.2023
OG1	**20**	**17.02.2023**	**16.03.2023**
OG1 Wände Kern	5	17.02.2023	23.02.2023
OG1 Stützen Zone Nord	3	24.02.2023	28.02.2023
OG1 Stützen Zone Süd	4	01.03.2023	06.03.2023
Geschossdecke über OG1 Zone Nord	4	07.03.2023	10.03.2023
Geschossdecke über OG1 Zone Süd	4	13.03.2023	16.03.2023
OG2	**20**	**17.03.2023**	**13.04.2023**
OG2 Wände Kern	5	17.03.2023	23.03.2023
OG2 Stützen Zone Nord	3	24.03.2023	28.03.2023
OG2 Stützen Zone Süd	4	29.03.2023	03.04.2023
Dach über OG2 Zone Nord	4	04.04.2023	07.04.2023
Dach über OG2 Zone Süd	4	10.04.2023	13.04.2023

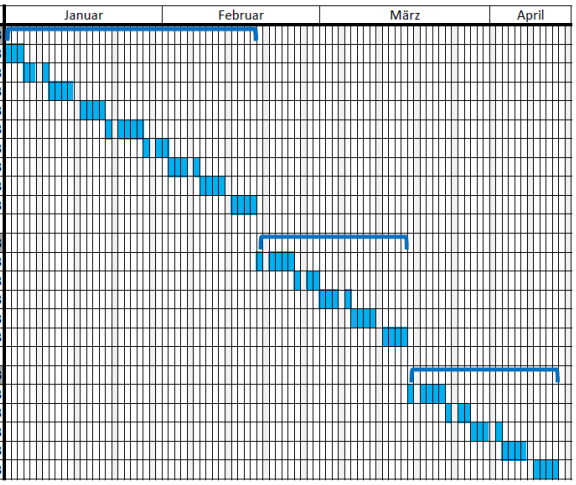


Tabelle 4: Terminplan synthetisches As-Planned-DBM

Für die As-Built-DBM wurden manuell Teile des As-Planned-DBM exportiert, um den Bauablauf zu simulieren. Ausgewählte Beispiel der As-Built-DBM sind in Abbildung

25 dargestellt. Dabei wurde darauf geachtet, dass die As-Planned-Modelle zu Beginn dem geplanten Bauablauf entsprechen, wie in (a) und (b) dargestellt. Gegen Ende wurde bewusst vom geplanten Bauablauf abgewichen, siehe (c), um zu prüfen, ob die Elemente und Etappen richtig klassifiziert werden können.

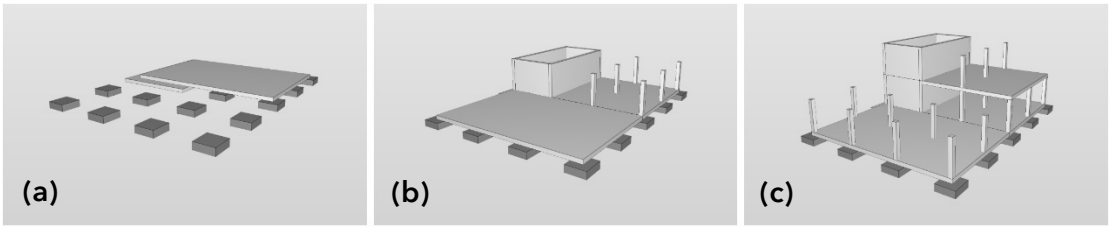


Abbildung 25: Ausgewählte Beispiele der AB-DBM: (a) Stand 13.01.23 nach Bauablauf, (b) Stand 31.01.23 nach Bauablauf und (c) Stand 23.02.23 Abweichung von Bauablauf.

8.2. Modellvergleich

Um den Baufortschritt für jedes Element festzustellen, wurde ein geometrischer Modellvergleich in Grasshopper entwickelt Für die Umsetzung wurde das Zusatzmodul GeometryGym verwendet, mit dessen Hilfe IFC-Dateien eingelesen und weiterverarbeiten werden können. Abbildung 26 zeigt eine Übersicht des gesamten Skripts. Das Skript ist von links nach rechts zu lesen, da Grasshopper das Skript in dieser Weise ausführt. Zuerst werden das Soll- und Ist-DBM als IFC-Dateien eingelesen (a). Anschliessend kommt es zu einem parallelen Prozess, bei dem für jedes Element separat der aktuelle Fortschritt bestimmt wird (b). Die Parallelität entsteht dabei, weil die Elemente nach Elementtyp gruppiert werden müssen. Der Grund dafür ist, dass die Bauteilmengen der einzelnen Elemente in bauteilspezifischen PSet gespeichert werden, wie zum Beispiel «Qto_FootingBaseQuantities» (buildingSMART, 2017). Deshalb muss auf die Mengen für jedes Bauteil separat zugegriffen werden (blau: Fundation, grün: Stützen, gelb: Bodenplatten, rot: Wände). Die Resultate des jeweiligen Geometrievergleichs werden anschliessend in einen CSV-String mit der zugehörigen Element-GUID und dem Datum (d) des aktuellen Stichtags verknüpft (c). Schlussendlich werden die generierten CSV-Strings zusammengefasst und auf Knopfdruck als eine CSV-Datei exportiert (e). Folgend wird genauer auf den eigentlichen Geometrievergleich, auf dessen Grundlage der Elementfortschritt berechnet wird, eingegangen.

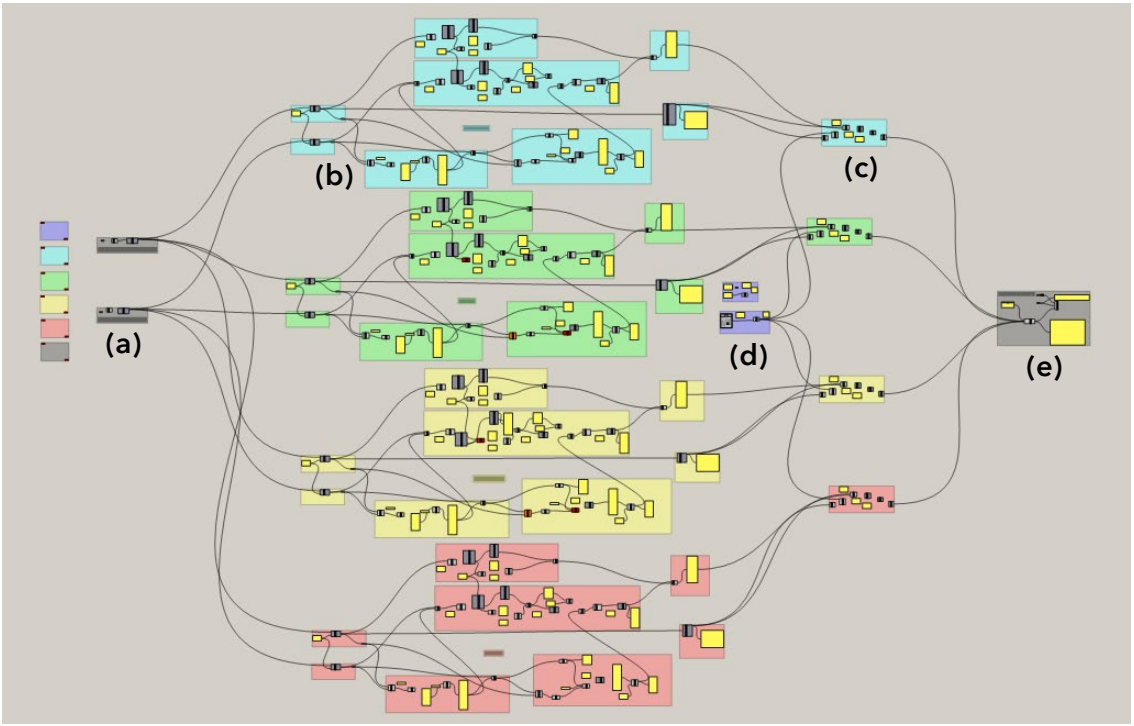


Abbildung 26: Übersicht Grasshopper-Skript für den Modellvergleich: (a) Input, (b) Geometrievergleich für jedes Element pro Elementtyp aufgeteilt nach Elementtyp, (c) Resultate Geometrievergleich als CSV-String, (d) aktuelles Datum und (e) Output.

Abbildung 27 zeigt wie der Geometrievergleich der verschiedenen Elemente aufgebaut ist. Dargestellt ist der Teil, in welchem die Wände ausgewertet werden. Der Geometrievergleich für die anderen Elementtypen ist allerdings analog aufgebaut. In einem ersten Schritt wird die IFC-Datei nach der entsprechenden IFC-Klasse (hier: IfcWall) gefiltert, um die entsprechenden Elementtypen zu isolieren (a). Für jedes Element wird anschliessend ein B-Rep (Boundary Representation) generiert, welches für den weiteren Geometrievergleich verwendet wird.

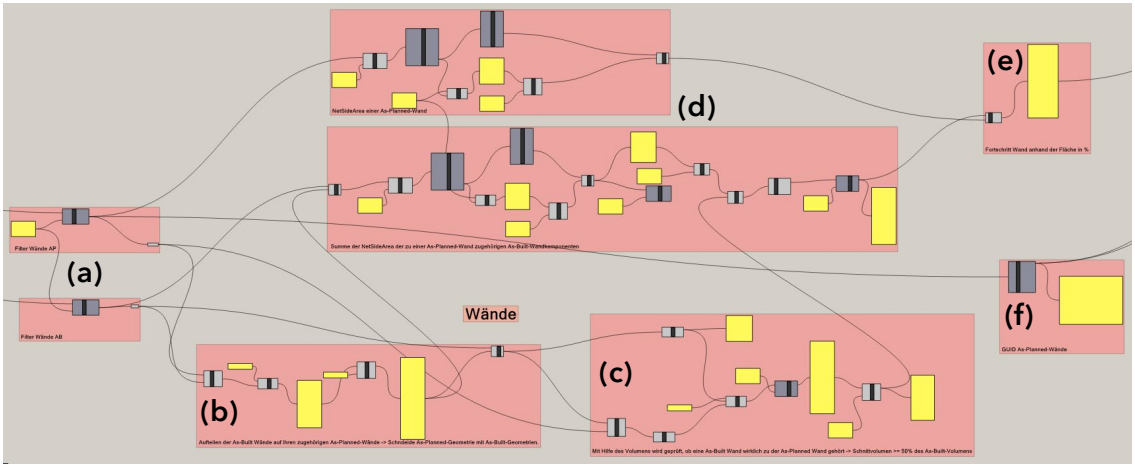


Abbildung 27: PoC Grasshopper-Skript Komponente Geometrievergleich Wände: (a) Filterung der IFC-Datei nach Wänden, (b) Zuordnung sich schneidender Wände des As-Planned- und As-Built-Modells, (c) Check, ob die Elemente wirklich zusammengehören, (d) Extaktion der der Bauteilmengen, (e) Berechnung Fortschritt pro Element und (f) Extraktion Element-GUID.

In einem nächsten Schritt werden alle As-Planned-Elemente mit allen As-Built-Elementen geschnitten, um festzustellen welche Elemente miteinander verglichen werden müssen Dieser Schritt ist wichtig, weil es sich um dasselbe Element oder ein Teil davon handeln könnte (b). Abbildung 28 zeigt dies am Beispiel einer einzelnen Wand. Das Grüne Volumen stellt die As-Planned-Wand dar und die schwarzen Drahtmodelle die As-Built- Wände. Alle vier As-Built-Wände schneiden die As-Planned-Wand. Jedoch sind nur drei dieser As-Built-Wände Bestandteil der As-Planned-Wand. Deshalb wird anschliessend anhand eines Schnittvolumens aus As-Planned- und As-Built-Wand gebildet (c). Dieses wird mit dem Volumen der As-Built-Wand verglichen. Liegt das Verhältnis unter einem gewissen Schwellenwert (hier 50%) wird die As-Built-Wand nicht weiter für die Fortschrittsermittlung der As-Planned-Wand berücksichtigt.

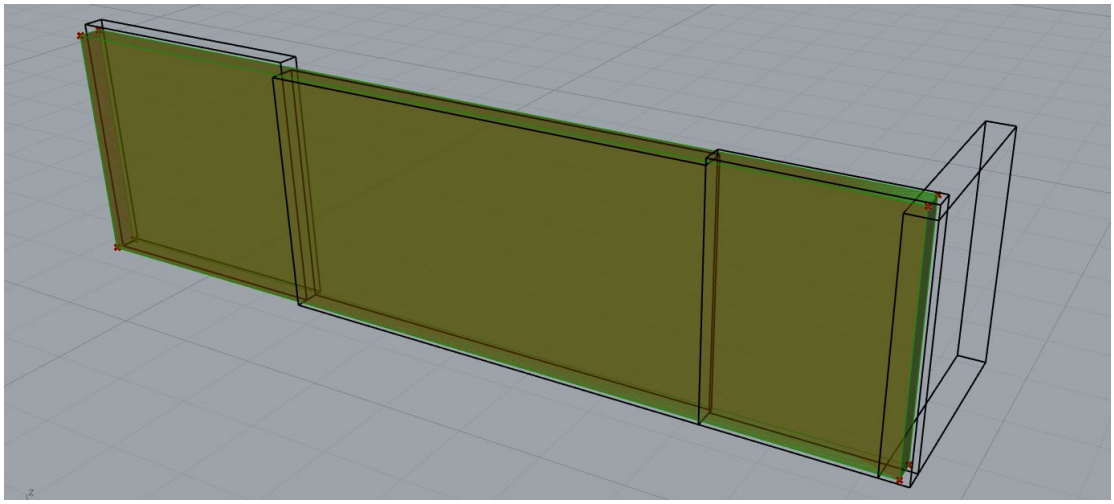


Abbildung 28: Konzept Geometrievergleich: As-Planned-Wand (grün) und AB-Wände (schwarze Linien)

Ein Vorteil dieses Vorgehens ist, dass bei der Modellierung des As-Planned-DBM nicht darauf geachtet werden muss Elemente in einzelnen Etappen (z. B.: Betonieretappen) zu modellieren. Zudem ist dieses Vorgehen wichtig, da es bei der Generierung des As-Built-DBM im Teilprojekt «4D Extraction» von BMETRY passieren kann, dass Elemente aufgrund des Erkennungsalgorithmus nicht als Ganzes, sondern zerstückelt extrahiert werden (vgl. Abbildung 29).

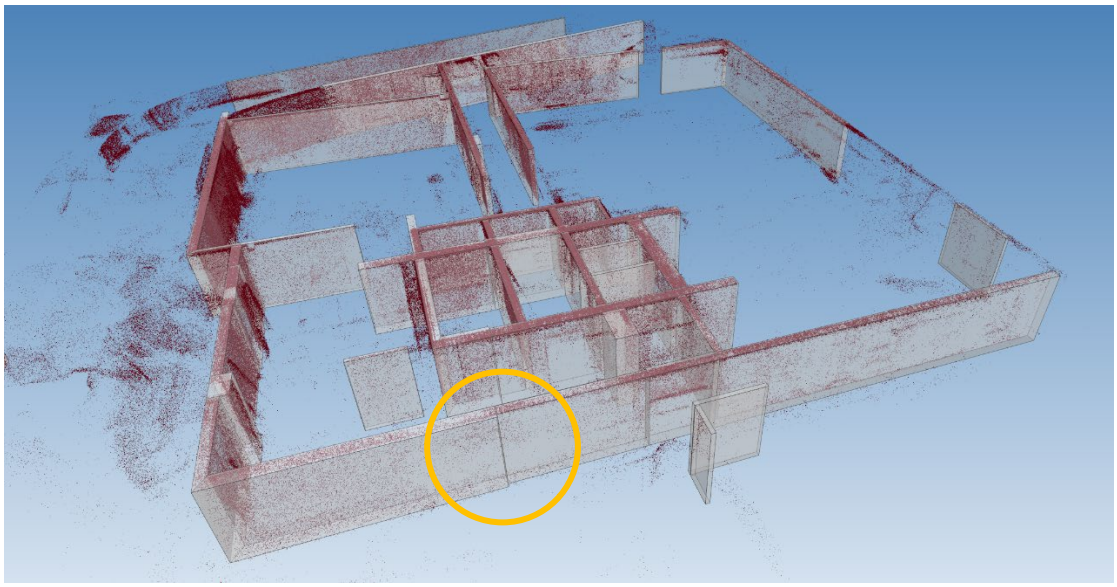


Abbildung 29: Beispiel-Output Teilprojekt "4D Exctraction" von BMETRY: Unterschied in der Wanddicke bei extrahierten Wänden auf Grund des Erkennungsalgorithmus (Marcinkeviciute, 2023).

Ist dieser Schritt abgeschlossen können nun die Basismengen jedes Elements aus der jeweiligen IFC-Datei extrahiert werden. Für das Element Wand ist die Nettofläche ausschlaggebend. Anschliessend wird der Fortschritt des Elements berechnet(e). Die entsprechende Formel für Wände ist in Abbildung 29 dargestellt. Für andere Elementtypen lauten die Formel analog, allerdings indem die Nettofläche durch die elementspezifische Metrik ersetzt wird.

$$\frac{\sum Nettofläche_AsBuiltWände}{Nettofläche_AsPlannedWand} = FortschrittWand$$

Abbildung 30: Formel zur Berechnung des Fortschritts einer Wand.

Um schlussendlich den Fortschritt einem bestimmten Element zuweisen zu können, wird aus der As-Planned-IFC-Datei noch die Element-GUID extrahiert (f).

8.3. Datenbank

Die Datenbank als zentrales Element dieses Konzeptes wurde mit MS Access umgesetzt. Es wurde diese bestehende Softwarelösung gewählt, da diese umfangreichen Funktionalitäten zur Bearbeitung, Verarbeitung und Auswertung von Daten bietet, ohne dazu Befehle in SQL (Structured Query Language) schreiben zu müssen. In Abbildung 31 ist die Struktur der Datenbank in der grafischen Modellierungssprache Unified Modeling Language (UML) dargestellt.

In der Klasse Etappe wird der Terminplan abgebildet. Die Klasse Element enthält alle Informationen zu den Elementen des Bauprojektes. Die entsprechenden Attribute können mit Hilfe eines Python-Skripts aus der As-Planned-IFC-Datei extrahiert. Die

Klasse Geschoss dient dazu Elemente der richtigen Etappe zuordnen zu können. In der Klasse FortschrittElement sind die Resultate des Geometrievergleichs enthalten.

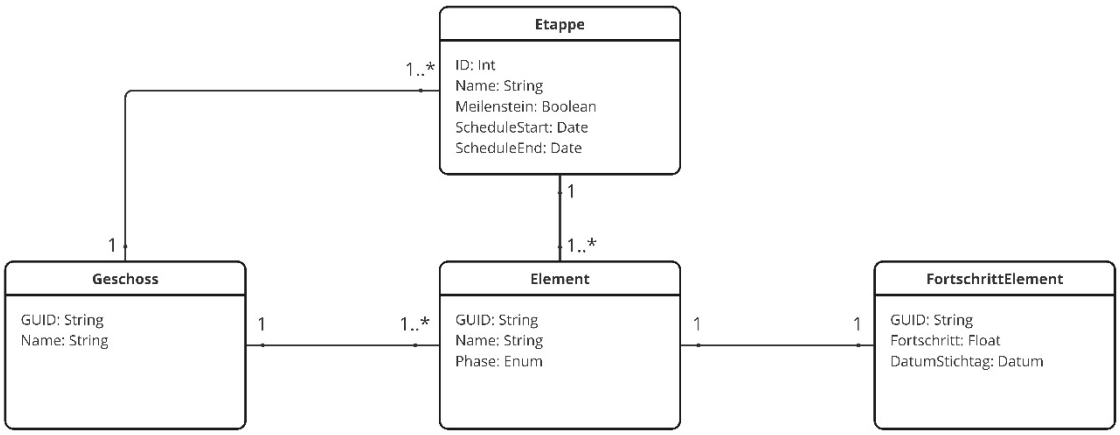


Abbildung 31: UML der Beziehungen zwischen den einzelnen Tabellen der Datenbank.

Auf Basis dieser vier Klassen kann nun der Baufortschritt durch verschiedene Auswertungen bestimmt und analysiert werden. In einem ersten Schritt wird hierzu für jedes Element ein Status anhand des Fortschritts, Stichtags und Terminplans definiert. Die entsprechende Matrix, inklusive des daraus abgeleiteten Farbcodes, ist in Tabelle 5 dargestellt. Wird der Status eines Elements als «fertig frühzeitig», «fertig» oder «fertig spät» festgelegt, wird das Messdatum als Datum der Fertigstellung definiert. Umgesetzt wurde diese Klassifizierung mit Hilfe eines Python-Skriptes, weil die Umsetzung dieser Auswertung für den Autor aus zwei Gründen nicht in MS Access möglich war. Zum einen ist im Auswertungsassistent die Anzahl Zeichen für Formeln begrenzt, weswegen kein ausführliches If-Else-Statements geschrieben werden konnte. Der zweite Grund hängt mit der Aktualisierung zusammen. Das Datum in der Klasse FortschrittElement verändert sich bei jedem neuen Geometrievergleich, weswegen der abgeleitete Status jedes Mal aufs Neue berechnet wurde. Dadurch wurden Elemente mit dem Status «fertig frühzeitig» oder «fertig» mit der Zeit als «fertig spät» klassifiziert und alle Elemente hatten das letzte Messdatum als Fertigstellungsdatum.

	Fortschritt > 0.9	Fortschritt < 0.9 & Fortschritt > 0.1	Fortschritt < 0.1
MessMessDatum < ScheduleScheduleStart	fertig frühzeitig	laufend frühgzritig	geplant
MessDatum = ScheduleStart	fertig frühzeitig	laufend	geplant
MessDatum zwischen ScheduleStart und ScheduleEnd	fertig	laufend	geplant
MessDatum = ScheduleEnd	fertig	laufend überfällig	geplant überfällig
MessDatum = ScheduleEnd Und Etappe = Meilenstein	fertig	laufend überfällig kritisch	geplant überfällig kritisch
MessDatum > ScheduleEnd	fertig spät	laufend überfällig	geplant überfällig
MessDatum > ScheduleEnd Und Etappe = Meilenstein	fertig spät	laufend überfällig kritisch	geplant überfällig kritisch

Tabelle 5: Matrix zur Klassifizierung von Bauelementen.

Nachdem alle Elemente klassifiziert sind, kann davon der Status der jeweiligen Etappe, zu welcher sie gehören, abgeleitet werden. Dazu wurde die in Tabelle 6 dargestellte Matrix verwendet. Eine umfangreichere Klassifizierung ist in Kapitel 8.5 zu finden. Diese wurde allerdings nicht implementiert, da dafür ebenfalls ein Python-

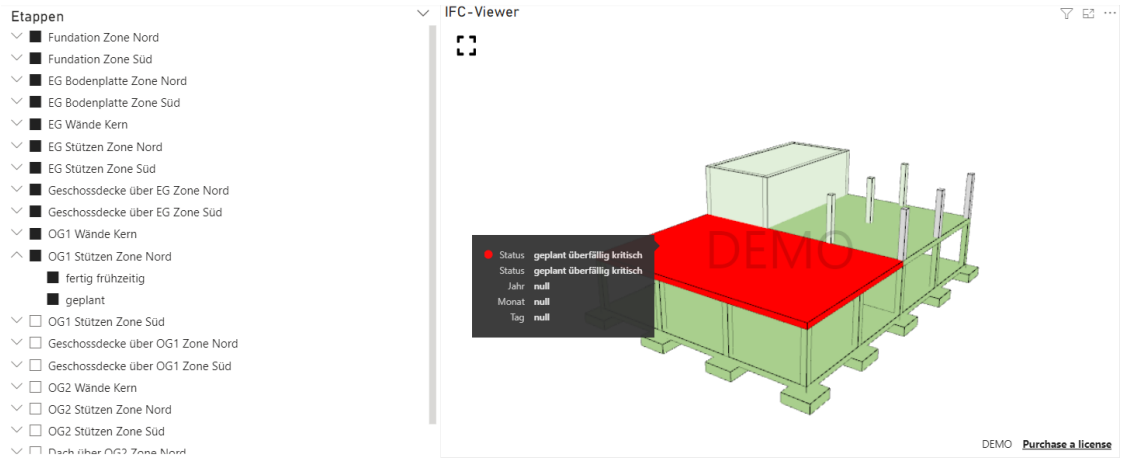
Skript notwendig gewesen wäre, auf welches im Umfang dieser Arbeit verzichtet wurde.

	alle Elemente fertig	1 oder mehr Elemente laufend, fertig oder geplant	alle Elelemente geplant
Etappe	fertig	laufend	geplant

Tabelle 6: Matrix zur Klassifizierung der Etappen (umgesetzt)

8.4. Visualisierung

Für die Visualisierung des Baufortschritts wurden zwei verschiedenen Softwarelösungen verwendet. Abbildung 32 zeigt den Baufortschritt mit Hilfe eines interaktiven Dashboards in PowerBI. Die MS Access Datenbank kann hierzu über eine direkte Schnittstelle in MS PowerBI verbunden werden. Das Visual mit welchem das As-Planned-DBM dargestellt wird wurde von Proving Ground entwickelt und heisst Tracer (Proving Ground, n.d.). Diesem liegt eine SQLite-Datenbank zu Grunde, welche von Tracer aus der entsprechenden IFC-Datei-generiert wird. Auch diese Datenbank lässt sich mit dem richtigen Treiber direkt in MS PowerBI einbinden.



GUID	Fortschritt Element	Etappe	ActualFinish	Status	ScheduleStart	ScheduleEnd	Meilenstein
26VRoJONX5KxmldQTjJOb$	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJOj6	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJOX8	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJOY$	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJVOB	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJVPS	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJVRU	1.00	Fundation Zone Nord	04.01.2023 00:00:00	fertig	02.01.2023 00:00:00	04.01.2023 00:00:00	False
26VRoJONX5KxmldQTjJVHC	1.00	Fundation Zone Süd	09.01.2023 00:00:00	fertig	05.01.2023 00:00:00	09.01.2023 00:00:00	False

Abbildung 32: Visualisierung Baufortschritt mit einem interaktiven Dashboard in MS PowerBI.

Als zweite Software zu Darstellung des Baufortschritts wurde Autodesk Navisworks gewählt. Da hierfür keine direkte Schnittstelle zu MS Access existiert, müssen die entsprechenden Daten als CSV-Datei exportiert werden. Diese kann anschliessend mit Autodesk Navisworks verknüpft werden. In Autodesk Navisworks wird der Baufortschritt als Balkendiagramm ersichtlich. Zudem lassen sich der Soll- sowie der Ist- Bauablauf anhand der angegebenen Datumsinformationen simulieren.

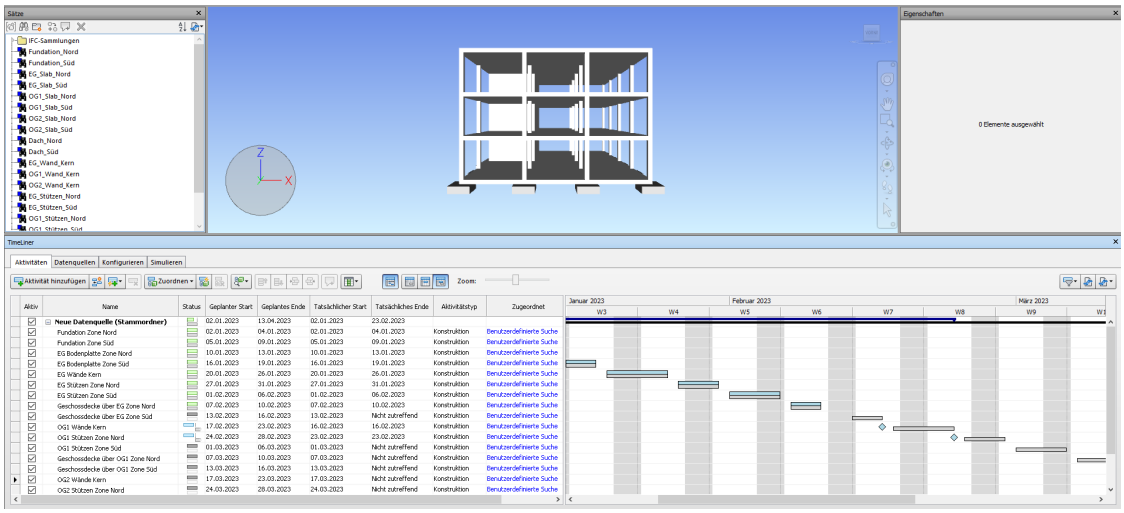


Abbildung 33: Visualisierung Baufortschritt mit als Balkendiagramm in Autodesk Navisworks.

8.5. Nicht implementierte Features

Bei der Umsetzung des PoC musste, im Umfang dieser Arbeit auf die Implementierung verschiedener Features verzichtet werden. Da darin trotzdem spannende Erkenntnisse stecken werden einige ausgewählte folgend erläutert.

Das erste nicht implementierte Feature betrifft den Modellvergleich. Für diesen müssen die Elemente des As-Planned-DBM einschichtig modelliert werden. Mit dem Zusatzmodul GeometryGym für Grasshopper kann jedoch auf die IFC-Klasse IfcMaterialLayerSet und damit auf die verschiedenen Schichten eines Bauteils zugegriffen werden (vgl Abbildung 34). Damit wäre es möglich auch DBM mit einem niedrigen Detaillierungsgrad für den für den Modellvergleich zu verwenden, indem zur die zu prüfende Schicht für den Modellvergleich verwendet wird.

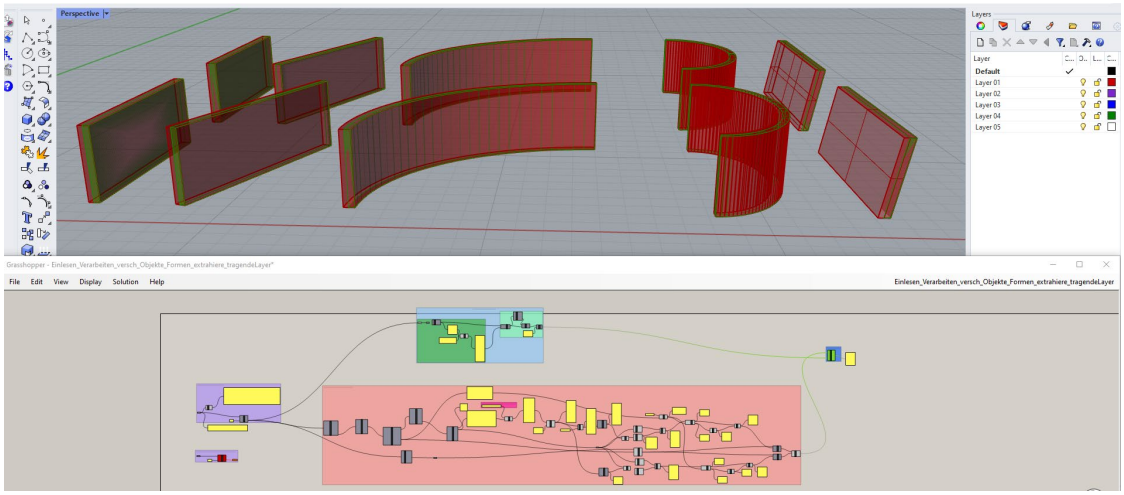


Abbildung 34: Extraktion der tragenden Schicht (grün) einer Wand mit Grasshopper.

Ein weiteres Feature, das nicht implementiert wurde, ist die ausführliche Klassifizierung der Etappen. In Tabelle 7 ist die entsprechende Matrix dargestellt.

	alle Elemente fertig	1 oder mehr Elemente laufend, fertig oder geplant	alle Elelemente geplant
MessDatum < ScheduleStart	fertig frühzeitig	laufend frühgzritig	geplant
MessDatum = ScheduleStart	fertig frühzeitig	laufend	geplant
MessDatum zwischen ScheduleStart und ScheduleEnd	fertig	laufend	geplant
MessDatum = ScheduleEnd	fertig	laufend überfällig	geplant überfällig
MessDatum = ScheduleEnd Und Etappe = Meilenstein	fertig	laufend überfällig kritisch	geplant überfällig kritisch
MessDatum > ScheduleEnd	fertig spät	laufend überfällig	geplant überfällig
MessDatum > ScheduleEnd Und Etappe = Meilenstein	fertig spät	laufend überfällig kritisch	geplant überfällig kritisch

Tabelle 7: Matrix zur Klassifizierung der Etappen (nicht implementiert).

Ein drittes nicht implementiertes Feature sind die Relationen die zwischen den einzelnen Etappen bestehen. In der umgesetzten Datenbank fehlen diese komplett (vgl. Abbildung 35). Solche würden jedoch Helfen zusätzliche Aussagen bezüglich des aktuellen Baufortschritts treffen zu können, da gewisse Etappen nur gebaut werden können, wenn die vorhergehende fertiggestellt ist. Zudem könnten so die Start- und Enddaten einer Etappe automatisiert berechnet werden, wenn sich bei einer vorhergehenden Etappe etwas ändert. Dieses Feature kommt aus dem Konzept des PRG, welcher von (Alex Braun et al., 2015) entwickelt wurde. In diesem Fall ist es jedoch auf Bauetappen bezogen und nicht auf einzelne Elemente.

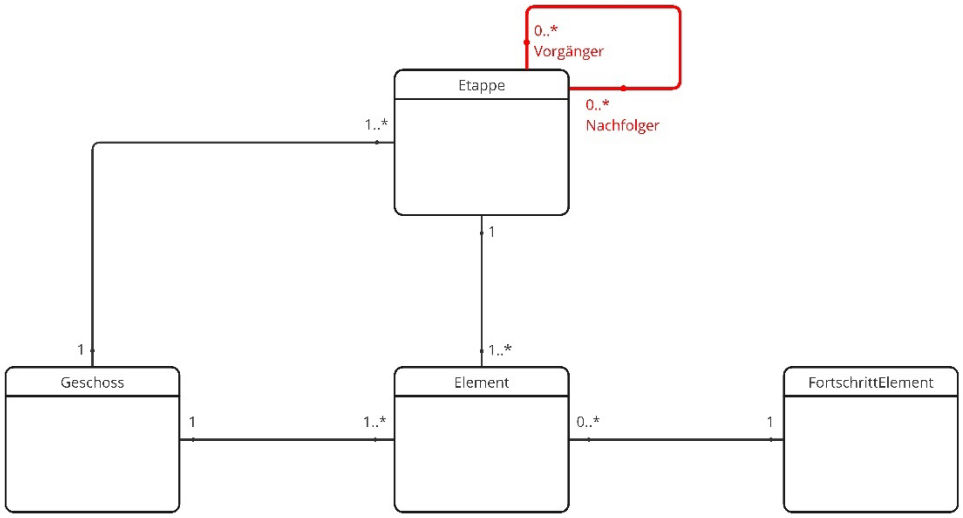


Abbildung 35: Ergänzung einer Relation zwischen den verschiedenen Etappen des Bauablaufs (rot).

Ein weiters nicht umgesetztes Feature in der Datenbank ist die automatisierte Berechnung des voraussichtlichen Abschlusses einer Etappe. Mit der in Abbildung 36 ergänzten Klasse EtappenTyp könnten Kennwerte zur Dauer einer Etappe mit der jeweiligen Etappe verknüpft werden. Diese Kennwerte können entweder im Laufe des Projektes gebildet werden, indem die wirkliche Dauer ähnlicher Etappen gemessen und ausgewertet wird oder auf Kennwerten der Projektbeteiligten basieren. Damit wäre es möglich zu berechnen, wann eine Etappe oder ein ganzes Projekt fertig würde, wenn im selben Tempo, mit derselben Leistung weitergearbeitet wird. Je mehr Parameter bekannt sind, desto genauer würde diese voraussage werden. Auch eine auf maschinellem Lernen basierende Lösung für dieses feature wäre denkbar.

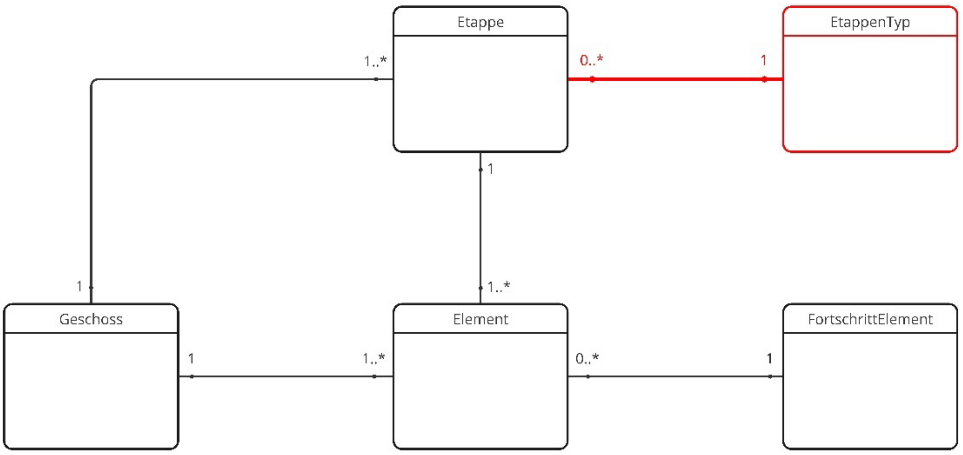


Abbildung 36: Ergänzung der Klasse EtappenTyp um Aussagen über den voraussichtlichen Abschluss einer Etappe machen zu können.

9. Exkurs Proof of Concept 4D-IFC

Im Zuge dieser Arbeit wurde die Möglichkeit untersucht zeitliche Informationen direkt in IFC-Dateien zu integrieren. Das von GeometryGym entwickelte Plugin für Autodesk Navisworks und Zusatzmodul für Grasshopper bieten in der Theorie die Möglichkeit einen solchen Workflow umzusetzen (Geometry Gym, 2022). Leider stellt sich heraus, dass die API für Autodesk Navisworks 2023 geändert wurde (jonm, 2022), weshalb die zeitlichen Informationen generierten 4D-IFC-Dateien nicht vollständig in den Timliner übernommen werden konnten. In Abbildung 37 ist das Resultat des Imports einer 4D-IFC-Datei in Autodesk Navisworks dargestellt. Es ist zu sehen, dass die Hierarchie des erstellten Terminplanes übernommen wurde: IfcWorkPlan als «Rohbau1», IfcTask (Root-Task) als «Walls» und die beiden IfcTasks «Wall_1» und «Wall_2» als Subtasks. Jedoch wurde das Start- und Enddatum von «Wall_1» bei «Wall_2» eingetragen wohingegen «Wall_1» kein Datum zugewiesen bekommen hat. Zudem ist in der Spalte «Zugeordnet» des Timeliners zu sehen, dass erkannt wurde, dass den beiden Tasks Elemente zugewiesen sind. Klickt man allerdings auf die Auswahl erscheinen keine Elemente im Eigenschaftsfenster.

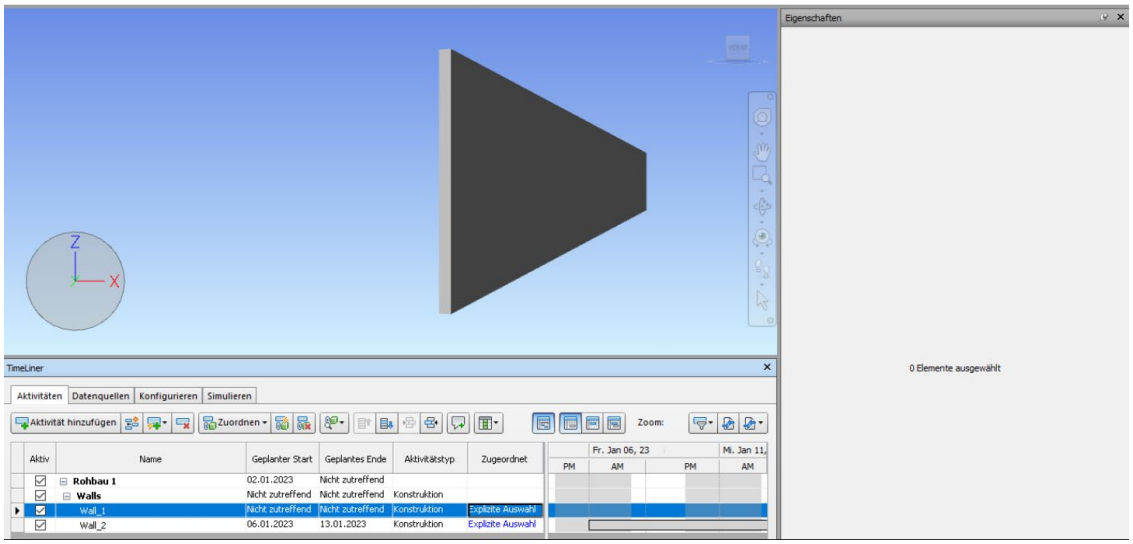


Abbildung 37: Resultat Import 4D-IFC in Autodesk Navisworks.

Öffnet man die entsprechende 4D-IFC-Datei in einem Texteditor sin alle relevanten, Klassen für die Erstellung einer 4D-IFC-Datei zu finden (vgl. Abbildung 38). Wenn der Entwickler des Plugins für Autodesk Navisworks dieses auf die neue API anpasst, sollte es also möglich sein, 4D-IFC-dateien in einer Software darstellen zu können. Für die Umsetzung dieses PoC wurde mit IFC-Version 2x3 gearbeitet, da damit einige Resultate erzielt werden konnten. Mit neueren IFC-Versionen blieb der Timliner leer. Es ist noch darauf hinzuweisen, dass die einige Klassen manuell im Texteditor, der 4D-IFC-Datei angefügt wurden, da das Zusatzmodul für Grasshopper einige IFC-Klassen, wie IfcScheduleTimeControl, nicht unterstützt und bei anderen, wie IfcWorkCalendar, Syntaxfehler beim Export macht.

```
FILE_SCHEMA (('IFC2X3'));
ENDSEC;

DATA;
.
.
#17= IFCOWNERHISTORY(#14,#16,$,.ADDED.,1672682720,$,$,1672682720);
.
.
#47= IFCWALLTYPE('3N7WJBg8PEyPjskiLUSsYJ',#17,'Betonwand',$,$,$,$,$,$,.NOTDEFINED.);
.
.
#56= IFCWALLSTANDARDCASE('2y5drbBQj8kwCbZrVrXvbq',#17,$,$,$,#63,#86,$);
.
.
#91= IFCWALLSTANDARDCASE('1E3nUE1xLDPAUEJlnCP1x6',#17,$,$,$,#100,#117,$);
.
.
#139= IFCWORKPLAN('1TRMp4aivBSvfDNfVdPozA',#17,'Rohbau 1',$,$,'',#143,$,'',0.0,0.0,#144,$,.NOTDEFINED.,'');
#140= IFCCALENDARDATE(2,1,2023);
.
.
#144= IFCCALENDARDATE(2,1,2023);
#148= IFCSCHEDULETIMECONTROL('1YWZZ9B0zAV9FT4wzPtLfz',#17,$,$,$,$,$,$,#149,$,$,$,$,$,$,$,$,$,.F.,$,0.0,0.0,$);
#149= IFCCALENDARDATE(2,1,2023);
#152= IFCSCHEDULETIMECONTROL('3_q10C_yX61uW34hjl1Xea',#17,$,$,$,$,$,$,#153,$,$,$,#170,$,$,$,$,$,.F.,$,0.0,0.0,$);
#153= IFCCALENDARDATE(2,1,2023);
#158= IFCSCHEDULETIMECONTROL('2Yd9GyprrAuhxGtf5YNPbs',#17,$,$,$,$,$,$,#159,$,$,$,#160,$,$,$,$,$,.F.,$,0.0,0.0,$);
#159= IFCCALENDARDATE(9,1,2023);
#160= IFCCALENDARDATE(13,1,2023);
#166= IFCTASK('2KozQozmP9Y86JBS3iGE9Q',#17,'Walls',$,'Konstruktion','da7e89c9-af5b-40a0-9db0-3be009957243',$,$,.F.,0);
#169= IFCTASK('3$gfffEyH1eQVaaMuoI_4j',#17,'Wall_1',$,'Konstruktion','17f5fb28-01a5-4b0d-81df-a09df5cb699a',$,$,.F.,0);
#170= IFCCALENDARDATE(6,1,2023);
#172= IFCRELAGGREGATES('0B6zKkMNCvjc2uCxHEPJ_j',#17,$,$,#166,(#169,#173));
#173= IFCTASK('0_oJhfiHH1Dxp_s9nBVzcJ',#17,'Wall_2',$,'Konstruktion','b22fc796-724a-4dcf-9115-a2e414277a18',$,$,.F.,0);
#175= IFCRELSEQUENCE('0_1pLz4tP3afU0m07D0_VL',#17,$,$,#169,#173,0.0,.FINISH_START.);
#176= IFCRELASSIGNSTASKS('0_1pjj4tP3afU0m07D0_VL',#17,$,$,(#166),$,#139,$);
#177= IFCRELASSIGNSTASKS('0_1pff4tP3afU0m07D0_VL',#17,$,$,(#169),$,#152,$);
#178= IFCRELASSIGNSTASKS('0_1pgg4tP3afU0m07D0_VL',#17,$,$,(#173),$,#158,$);
#179= IFCRELASSIGNSTOPROCESS('0_1pfgfgP3afU0m07D0_VL',#17,$,$,(#56),#47,#169,$);
#180= IFCRELASSIGNSTOPROCESS('0_1phhfgP3afU0m07D0_VL',#17,$,$,(#91),#47,#173,$);
ENDSEC;
```

Abbildung 38: Ausschnitt 4D-IFC-Datei: IfcScheduleTimeControl (blau), IfcRelAssignsTasks (grün) und IfcRelAssignsToProcess (orange).

10. Diskussion

In dieser Arbeit wurde gezeigt, dass es verschiedene Möglichkeiten gibt zeitliche Informationen mit DBM zu verknüpfen. Einige davon sind heute schon in der Praxis umsetzbar, wohingegen andere, wie die 4D-Modellierung in IFC nur theoretisch anwendbar sind. Der in dieser Arbeit entwickelte datenbankbasierte Ansatz, konnte mit dem umgesetzten PoC als legitime, umsetzbare und anwendbare Methode bestätigt werden. Ein grosser Vorteil dieses Ansatzes liegt darin, dass verschiedene Softwareprodukte die darin enthaltenen Daten interpretieren und weiterverwenden können. Dies wurde mit der Visualisierung in MS PowerBI und Autodesk Navisworks gezeigt. Es ist also vorstellbar, dass dieser Ansatz auch in BEMTRY oder anderen ähnlichen Projekten umgesetzt werden kann. Folglich eignet sich dieser Ansatz für Open-BIM-Prozesse

Der geometrische Modellvergleich konnte den Fortschritt aller Bauteile ohne Fehler bestimmen. Das liegt vor allem daran, dass als As-Built-Input ein generiertes DBM verwendet wird und in dieser Arbeit mit synthetischen Daten gearbeitet wurde. Verdeckte Elementoberflächen durch temporäre Strukturen auf der Baustelle, sind deshalb für diesen Modellvergleich keine Einschränkung. Es ist allerdings zu beachten, dass das Problem damit auf das Teilprojekt «4D Extraction» von BMETRY verlagert wird. Wie die Resultate eines Modellvergleichs mit realen Datensätzen ausfallen würde, hängt deshalb stark davon ab, wie präzise das generierte As-Built-DBM ist. Eine weitere Einschränkung des Entwickelten Vorgehens ist, dass der Fortschritt einzelner Elemente nur festgestellt werden kann, wenn ein Teil davon oder das Ganze fertiggestellt sind. Zwischenschritte wie das Stellen von Schalung oder legen von Armierung oder Einlagen können mit diesem Ansatz nicht erkannt werden. Damit bleibt bei jedem Element unklar, wann die Arbeit an daran begonnen hat. Zudem haben sich bei der Umsetzung des PoC einige Modellierungsrichtlinien herauskristallisiert, die zu beachten sind. So müssen Elemente eindeutig einer Etappe zugeordnet werden können. Grosse Elemente, wie lange Wände oder Bodenplatten, müssen um dieses Kriterium zu erfüllendeshalb an entsprechender Stelle geteilt werden. Zudem setzt der Geometrievergleich im aktuellen Stadium noch einen hohen Detaillierungsgrad des As-Planned-DBM voraus, indem Elemente nicht mehrschichtig modelliert werden dürfen. Wenn jedoch das vorgeschlagene Feature zu Extraktion der zu vergleichenden Schicht implementiert wird, könnten auch weniger detaillierte DBM für den Modellvergleich genutzt werden. Etwas, was nicht beachtet werden muss, ist die Modellierung von Zwischenetappen, wie die Betonieretappen einer Wand, solange diese zur selben Etappe gehören.

In der berücksichtigten Literatur und in den Fachgesprächen hat sich gezeigt, dass eine farbliche Kodierung der Bauelemente eine sinnvolle Methode ist, den Baufortschritt verständlicher darzustellen. In Kombination mit Balkendiagrammen

und statistischen Auswertungen, können so fundierte Entscheide für das weitere Vorgehen in einem Bauprojekt ermöglicht werden. Zudem werden diese für den Bau essenziellen Informationen für alle Projektbeteiligten einsehbar und interpretierbar. Dies ist vor allem dann wichtig, wenn es darum geht Metriken aus dem aktuellen Baufortschritt abzuleiten. In den Fachgesprächen und in der berücksichtigten Literatur, wurde der Baufortschritt immer in Kombination mit der Kostenplanung und der Qualität gesehen. Im Rahmen dieser Arbeit konnte allerdings nicht vertieft auf die Relation dieser KPI eingegangen werden. Deshalb wäre es spannend, diese in weiteren Arbeiten genauer zu untersuchen.

Insgesamt wurde mit dem umgesetzten PoC eine funktionierende Basis für weitere Forschung geschaffen. Spannend wäre zum Beispiel zu untersuchen, wie aus den erfassten Daten der weitere Verlauf eines Bauprojektes datengetrieben vorausgesagt werden kann, ohne auf die Erfahrungen und Interpretationen einzelner Bauleiter:Innen vertrauen zu müssen. Zudem müsste das entwickelte PoC in einem nächsten Schritt mit realen Daten getestet werden, um die Praxistauglichkeit zu testen.

11. Literatur

Alizadehsalehi, S., Yitmen, I., 2019. *A Concept for Automated Construction Progress Monitoring: Technologies Adoption for Benchmarking Project Performance Control*, Springer Berlin Heidelberg. Arab. J. Sci. Eng. 2011 44, 4993–5008. https://doi.org/10.1007/s13369-018-3669-1

Alzraiee, H., 2018. *Integrating BIM and earned value management system to measure construction progress*, in: Construction Research Congress 2018: Construction Information Technology - Selected Papers from the Construction Research Congress 2018. pp. 684–693. https://doi.org/10.1061/9780784481264.067

Artus, M., Koch, C., König, M., 2021. *Datenmodellierung*, Springer Vieweg, in: Building Information Modeling: Technology Foundations and Industry Practice, VDI-Buch. Springer Vieweg, Wiesbaden, pp. 147–163. https://doi.org/10.1007/978-3-319-92862-3_28

Atkinson, R., 1999. *Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria*. Int. J. Proj. Manag. 17, 337–342. https://doi.org/10.1016/S0263-7863(98)00069-6

Autodesk, 2022. Navisworks [WWW Document]. URL https://www.autodesk.ch/de/products/navisworks/overview?term=1-YEAR&tab=subscription&plc=NAVSIM (accessed 12.15.22).

Bariczová, G., Erdélyi, J., Honti, R., Tomek, L., 2021. *Wall structure geometry verification using tls data and bim model*, Mdpi. Appl. Sci. 11, 11804. https://doi.org/10.3390/app112411804

bentley, 2022. SYNCHRO [WWW Document]. URL https://www.bentley.com/software/synchro/ (accessed 12.15.22).

Blaser Zürcher, R., 2022. *Fachgespräch mit Dozent MAS FHNW Bauleitung / Dozent DAS FHNW Bauphysik / Mitinhaber ingBP*.

Bognot, J.R., Candido, C.G., Blanco, A.C., Montelibano, J.R.Y., 2018. *BUILDING CONSTRUCTION PROGRESS MONITORING USING UNMANNED AERIAL SYSTEM (UAS), LOW-COST PHOTOGRAMMETRY, and GEOGRAPHIC INFORMATION SYSTEM (GIS)*, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. pp. 41–47. https://doi.org/10.5194/isprs-annals-IV-2-41-2018

Borrmann, A., Beetz, J., Koch, C., Liebich, T., Muhič, S., 2021a. *Industry Foundation Classes – Ein herstellerunabhängiges Datenmodell für den gesamten Lebenszyklus eines Bauwerks*, Springer Vieweg, in: Building Information Modeling: Technology Foundations and Industry Practice, VDI-Buch. Springer Vieweg, Wiesbaden, pp. 147–163. https://doi.org/10.1007/978-3-319-92862-3_28

Borrmann, A., Koch, C., 2021. *Software-Interoperabilität im Bauwesen – Hintergrund und Motivation*, Springer Vieweg, in: Building Information Modeling: Technology Foundations and Industry Practice, VDI-Buch. Springer Vieweg, Wiesbaden, pp. 147–163. https://doi.org/10.1007/978-3-319-92862-3_28

Borrmann, A., König, M., Koch, C., Beetz, J., 2021b. *Die BIM-Methode im Überblick*, Springer Vieweg, in: Building Information Modeling: Technology Foundations and Industry

Practice, VDI-Buch. Springer Vieweg, Wiesbaden, pp. 147–163. https://doi.org/10.1007/978-3-319-92862-3_28

Bosche, F., Haas, C.T., Akinci, B., 2009. *Automated Recognition of 3D CAD Objects in Site Laser Scans for Project 3D Status Visualization and Performance Control*, American Society of Civil Engineers. J. Comput. Civ. Eng. 23, 311–318. https://doi.org/10.1061/(ASCE)0887-3801(2009)23:6(311)

Braun, A., Borrmann, A., 2019. *Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning*. Autom. Constr. 106, 102879. https://doi.org/10.1016/j.autcon.2019.102879

Braun, A., Borrmann, A., Koch, C., 2021. *Automatisierte Baufortschrittserkennung mit BIM*, Springer Vieweg, in: Building Information Modeling: Technology Foundations and Industry Practice, VDI-Buch. Springer Vieweg, Wiesbaden, pp. 563–576. https://doi.org/10.1007/978-3-319-92862-3_28

Braun, A., Borrmann, A., Tuttas, S., Stilla, U., 2016. *Classification of detection states in construction progress monitoring*, in: EWork and EBusiness in Architecture, Engineering and Construction - Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016. pp. 473–480.

Braun, A., Tuttas, S., Borrmann, A., Stilla, U., 2020. *Improving progress monitoring by fusing point clouds, semantic data and computer vision*. Autom. Constr. 116, 103210. https://doi.org/10.1016/j.autcon.2020.103210

Braun, Alex, Tuttas, S., Borrmann, A., Stilla, U., 2015. *A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds*. J. Inf. Technol. Constr. 20, 68–79.

Braun, Alexander, Tuttas, S., Borrmann, A., Stilla, U., 2015. *Automated Progress Monitoring Based on Photogrammetric Point Clouds and Precedence Relationship Graphs*, IAARC. ISARC Proc. 1–7.

buildingSMART, 2022. IFC4.3.1.x dev [WWW Document]. IFC Specif. Database. URL http://ifc43-docs.standards.buildingsmart.org/ (accessed 12.19.22).

buildingSMART, 2017. IFC4 ADD2 TC1 [WWW Document]. IFC Specif. Database. URL https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ (accessed 12.21.22).

buildingSMART, 2005. IFC2x3 TC1 [WWW Document]. IFC Specif. Database. URL https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/TC1/HTML/ (accessed 12.21.22).

Chen, J., Cho, Y.K., 2018. *Point-to-point comparison method for automated scan-vs-bim deviation detection*, in: Proceedings of the 2018 17th International Conference on Computing in Civil and Building Engineering, Tampere, Finland. pp. 5–7.

Crowther, J., Ajayi, S.O., 2021. *Impacts of 4D BIM on construction project performance*, Taylor & Francis. Int. J. Constr. Manag. 21, 724–737. https://doi.org/10.1080/15623599.2019.1580832

Eastman, C., Teichholz, P., Sacks, R., Liston, K., 2011. *BIM Handbook - A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors*, 2nd ed. John Wiley & Sons, New Jersey.

Fahri, J., Biesenthal, C., Pollack, J., Sankaran, S., 2015. *Understanding Megaproject Success beyond the Project Close-Out Stage*. Constr. Econ. Build. 15, 48. https://doi.org/10.5130/AJCEB.v15i3.4611

Fard, M.G., Peña-Mora, F., 2012. *Application of Visualization Techniques for Construction Progress Monitoring*, American Society of Civil Engineers 216–223. https://doi.org/10.1061/40937(261)27

Ferraro, E., 2022. *Softwareeinführung*.

Froese, T., Yu, K., Liston, K., Fischer, M., 2000. *System architectures for AEC interoperability*.

Geometry Gym, 2022. Navisworks | ggNavisIFC [WWW Document]. Geom. Gym Tech. URL (accessed 12.22.22).

Golparvar-Fard, M., Peña-Mora, F., Savarese, S., 2015. *Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models*, American Society of Civil Engineers. J. Comput. Civ. Eng. 29, 04014025. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000205

Hamledari, H., McCabe, B., Davari, S., Shahi, A., 2017. *Automated Schedule and Progress Updating of IFC-Based 4D BIMs*. J. Comput. Civ. Eng. 31. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000660

jonm, 2022. *Creating 4D-Information in GH and importing them into Navisworks*. Rhinoceros Forums.

Kielhauser, C., Manzano, R.R., Hoffman, J.J., Adey, B.T., 2020. *Automated construction progress and quality monitoring for commercial buildings with unmanned aerial systems: An application study from Switzerland*. Infrastructures 5, 1–18. https://doi.org/10.3390/infrastructures5110098

Kim, Seungho, Kim, Sangyong, Lee, D.-E., 2020. *3D Point cloud and BIM-based reconstruction for evaluation of project by as-planned and as-built*, Mdpi. Remote Sens. Basel Switz. 12, 1457. https://doi.org/10.3390/RS12091457

Leite, F., Cho, Y., Behzadan, A., Lee, S., Choe, S., Fang, Y., Akhavian, R., Hwang, S., 2016. *Visualization, Information Modeling, and Simulation: Grand Challenges in the Construction Industry*. J. Comput. Civ. Eng. 30, 04016035. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000604

Marcinkeviciute, D., 2023. *Screenshot Wandextraktion BMETRY*.

Martin, D., 2022. *Fachgespräch mit Projektleiter Bauunterhalt und Geomatik Aegerter & Bosshardt AG*.

Navon, R., Sacks, R., 2007. *Assessing research issues in Automated Project Performance Control (APPC)*. Autom. Constr. 16, 474–484. https://doi.org/10.1016/j.autcon.2006.08.001

Nigel, R., Eduard, M., 2021. *Die Bedeutung der Datenverwaltung in einem OpenBIM-Framework*. Wikipedia. URL https://blog.allplan.com/de/die-bedeutung-der-datenverwaltung-in-einem-openbim-framework (accessed 1.12.23).

O'Keeffe, S., Hyland, N., Dore, C., Brodie, S., 2017. *Automatic Validation of As-Is and As-Generated IFC BIMs for Advanced Scan-to-BIM Methods*.

Oracle Primavera, 2022. Rules of Credit Overview [WWW Document]. Oracle Primav. Cloud Help. URL https://docs.oracle.com/cd/E80480_01/help/en/user/89290.htm (accessed 12.30.22).

Pal, A., Lin, J.J., Hsieh, S.-H., 2022. *Automated Construction Progress Monitoring of Partially Completed Building Elements Leveraging Geometry Modeling and Appearance Detection with Deep Learning*, in: Construction Research Congress 2022: Computer Applications, Automation, and Data Analytics - Selected Papers from Construction Research Congress 2022. pp. 708–718. https://doi.org/10.1061/9780784483961.074

Pal, A., Lin, J.J., Hsieh, S.-H., 2021. *A Framework for Automated Daily Construction Progress Monitoring Leveraging Unordered Site Photographs*, in: Computing in Civil Engineering 2021 - Selected Papers from the ASCE International Conference on Computing in Civil Engineering 2021. pp. 538–545. https://doi.org/10.1061/9780784483893.067

Park, J., Cai, H., Dunston, P.S., Ghasemkhani, H., 2017. *Database-Supported and Web-Based Visualization for Daily 4D BIM*. J. Constr. Eng. Manag. 143. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001392

Project Management Institute, 2021. *The standard for project management and a guide to the project management body of knowledge (PMBOK guide)*, 7th ed, PMI global standard. Project Management Institute, Inc, Newtown Square, Pennsylvania, USA.

Proving Ground, n.d. Tracer [WWW Document]. Proving Ground. URL (accessed 1.13.23).

Pučko, Z., Šuman, N., Rebolj, D., 2018. *Automated continuous construction progress monitoring using multiple workplace real time 3D scans*. Adv. Eng. Inform. 38, 27–40. https://doi.org/10.1016/j.aei.2018.06.001

Qu, T., Zang, W., Peng, Z., Liu, J., Li, W., Zhu, Y., Zhang, B., Wang, Y., 2017. *Construction site monitoring using UAV oblique photogrammetry and BIM technologies*, in: CAADRIA 2017 - 22nd International Conference on Computer-Aided Architectural Design Research in Asia: Protocols, Flows and Glitches. pp. 655–662.

Rebolj, D., Pučko, Z., Babič, N.Č., Bizjak, M., Mongus, D., 2017. *Point cloud quality requirements for Scan-vs-BIM based automated construction progress monitoring*. Autom. Constr. 84, 323–334. https://doi.org/10.1016/j.autcon.2017.09.021

Reja, V.K., Bhadaniya, P., Varghese, K., Ha, Q., 2021. *Vision-Based Progress Monitoring of Building Structures Using Point-Intensity Approach*, in: Proceedings of the International Symposium on Automation and Robotics in Construction. pp. 349–356.

Riecke, H., 2022. *Fachgespräch mit Chefbautleiter Verkehr und Umwelt Aegerter & Bosshardt AG*.

Schneider, O., Nebiker, S., Rub, M., Bärtschi, D., 2021. *BMETRY*, Innosuisse.

Schweizer, R., 2022. *Fachgespräch mit VDC Manager / Teamleiter Erne AG Bauunternehmung*.

Scopus, 2015. Validation of BIM components by photogrammetric point clouds for construction site monitoring [WWW Document]. Scopus. URL https://www.scopus.com/record/display.uri?eid=2-s2.0-84979301709&origin=resultslist&sort=plf-f&src=s&st1=Validation+of+BIM+components+by+photogrammetric+point+clouds+for+construction+site+monitoring&sid=ba380ec437b6db9662dc36c644de932d&sot=b&sdt=b&sl=108&s=TITLE-ABS-KEY%28Validation+of+BIM+components+by+photogrammetric+point+clouds+for+construction+site+monitoring%29&relpos=0&citeCnt=23&searchTerm= (accessed 12.27.22).

thinkproject, 2022. Desite BIM [WWW Document]. URL https://thinkproject.com/de/produkte/desite-bim/ (accessed 12.15.22).

Tuttas, S., Braun, A., Borrmann, A., Stilla, U., 2015. *Validation of Bim Components by Photogrammetric Point Clouds for Construction Site Monitoring*. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. II3, 231–237. https://doi.org/10.5194/isprsannals-II-3-W4-231-2015

Vandevoorde, S., Vanhoucke, M., 2006. *A comparison of different project duration forecasting methods using earned value metrics*. Int. J. Proj. Manag. 24, 289–302. https://doi.org/10.1016/j.ijproman.2005.10.004

Vincke, S., Vergauwen, M., 2022. *Vison based metric for quality control by comparing built reality to BIM*. Autom. Constr. 144. https://doi.org/10.1016/j.autcon.2022.104581

Wang, Z., Zhang, Q., Yang, B., Wu, T., Lei, K., Zhang, B., Fang, T., 2021. *Vision-Based Framework for Automatic Progress Monitoring of Precast Walls by Using Surveillance Videos during the Construction Phase*. J. Comput. Civ. Eng. 35. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000933

Yang, J., Park, M.-W., Vela, P., Golparvar-Fard, M., 2015. *Construction performance monitoring via still images, time-lapse photos, and video streams: Now, tomorrow, and the future*. Adv. Eng. Inform. 29. https://doi.org/10.1016/j.aei.2015.01.011

12. Anhang

Anhang 01: Python-Skript PSet

Anhang 02: Projektordner Proof of Concept

Anhang 03: Grasshopper-Skript zur Extraktion spezifischer Bauteilschichten

Anhang 04: Tests zur Generierung einer 4D-IFC-Datei