Ingenieurfakultät Bau Geo Umwelt
Lehrstuhl für Computergestützte Modellierung und Simulation
Prof. Dr.-Ing. André Borrmann

Entwicklung eines 4D-BIM-Viewers mit graphbezogener Darstellung von Bauabläufen und - alternativen

Alexander Braun

Masterthesis
für den Master of Science Studiengang Bauingenieurwesen

Autor:	Alexander Braun
Matrikelnummer:	[redacted]
Betreuer:	Prof. Dr.-Ing. André Borrmann
	Dipl.-Ing. (FH) Simon Daum
Ausgabedatum:	15. Dezember 2012
Abgabedatum:	15. Juni 2013

Zusammenfassung

Die Baufortschrittsüberwachung auf Großbaustellen läuft heutzutage noch weitestgehend manuell ab. Mit technischer Hilfe in Form von Photogrammetrie kann dieser Vorgang automatisiert werden.
In dieser Thesis wird ein Überblick über die Grundlagen, technischen Hintergründe und den aktuellen Forschungsstand der vierdimensionalen Bauablaufplanung sowie deren Erfassung und Darstellung gegeben.
Die besondere Herausforderung bei der Erfassung von erkannten Bauteilen ist dabei die Abbildung der logischen Abhängigkeiten zu anderen Bauteilen. Dies stellt einen essentiellen Teil der Automatisierung der Baufortschrittskontrolle dar, da bei der Erkennung mit Hilfe der Photogrammetrie nicht immer alle Bauteile sichtbar sind.
Einen Lösungsansatz bietet hier die Graphentheorie, da nach Erstellung eines Abhängigkeits-Graphen sämtliche relevanten Bauteile anhand von Graph-Algorithmen abgebildet werden können.
Zum Nachweis der Tragfähigkeit der erarbeiteten Erkenntnisse wird ein Softwaretool entwickelt, welches ein digitales Modell eines Bauprojektes darstellen kann und den Baufortschritt visualisiert.

Abstract

Today, the construction process measurement on large construction sites still runs mainly manually. This process can be automated with technical assistance in the form of photogrammetry.
This thesis gives an overview of the basics, technical background and current research in the four-dimensional construction process planning and their capture and representation.
The special challenge in the detection of recognized components is the mapping of the logical dependencies to other building components. This is an essential part of the automation of the construction process measurement as not all building components are visible while using photogrammetry methods.
The graph theory provides a solution approach, since all relevant components can be detected by means of graph algorithms after creating a dependency graph.
Based on the gained insights, a software tool is developed which can show a digital model of a construction project and visualize the progress of construction.

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

BIM	Building Information Modeling
GUI	Graphical User Interface
IDM	Information Delivery Manual
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
WPF	Windows Presentation Foundation
IFD	International Framework for Dictionaries
STEP	Standard for the Exchange of Product model data
XML	Extensible Markup Language
DFG	Deutsche Forschungsgesellschaft
BSD	Berkeley Software Distribution
CDDL	Common Development and Distribution License
AGPL	GNU Affero General Public License
OSGPL	OpenSceneGraph Public License
CAD	Computer-aided design
CSG	Constructive Solid Geometry
NURBS	Non-Uniform Rational B-Splines

Kapitel 1

Einführung und Motivation

1.1 Ausgangssituation

Große Bauprojekte auf der gesamten Welt halten nur sehr selten den geplanten Kosten- und Zeitrahmen ein.

Aktuelle Projekte in Deutschland wie der Stuttgarter Tiefbahnhof (S21), die Hamburger Elbphilharmonie oder der geplante Berliner Großflughafen (BER) zeigen, dass trotz jahrelanger und intensiver Planung vielschichtige Arten von Problemen auftreten können. Der Grund hierfür sind zum einen die großen Unwägbarkeiten im Baubereich, zu denen zum Beispiel Lieferengpässe, unerwartete Beschaffenheit des Baugrunds sowie Witterungsbedingungen oder andere unvorhersehbare Randbedingungen gehören.

Ein weiterer Grund für Probleme im Bauablauf ist aber die hochkomplexe Prozessüberwachung. Auf Großbaustellen finden viele Prozesse parallel statt. Ein exakt definierter Bauablauf ist schwer zu bestimmen oder gar vorherzusagen. Dadurch entstehen weitere Unwägbarkeiten im Bereich der Baulogistik und der Gesamtplanung eines Objekts. Dies macht die Prozessüberwachung zu einem essentiellen Teil der Organisation einer Baustelle.

Wenngleich der Rohbau auf den meisten Baustellen bereits recht strukturiert abläuft, lohnt sich hier eine genauere Betrachtung der Bauabläufe.

Aktuell erfolgt die Fortschrittskontrolle meist händisch oder mittlerweile auch über Tablet-Computer direkt auf der Baustelle. In jedem Fall aber wird der Fortschritt durch den Bauleiter manuell in Bautagebüchern festgehalten.

Dies birgt neben dem hohen personellen Aufwand auch diverse Risiken bezüglich Inkonsistenzen und Aktualität. Ein automatisiertes Verfahren wäre hier empfehlenswert.

1.2 Idee und Ziel dieser Arbeit

In dieser Thesis soll ein Beitrag zur Forschung im Bereich der Automatisierung der Bauüberwachung geschaffen werden.

Die Grundlage soll dabei die Erfassung einer Baustelle mit Hilfe der Photogrammetrie sein. Dabei soll aus den digitalen Fotografien des jeweils aktuellen Bauzustandes eine dreidimensionale Punktwolke generiert werden. Diese wird anschließend mit einem vorhandenen digitalen dreidimensionalen Modell des Bauprojektes abgeglichen.

Der Schwerpunkt liegt hierbei auf der Erkennung einzelner Bauteile um dadurch den aktuellen Baufortschritt abschätzen zu können. Durch die Verknüpfung des digitalen Gebäudemodells mit einem zugehörigen Bauablaufplan kann somit eine Aussage über den Baufortschritt und eventuelle Abweichungen vom geplanten Baufortschritt getroffen werden.

Bei fortlaufender und automatisierter Beobachtung einer Baustelle mit Hilfe einer Kamera, welche eine drahtlose Übermittlung ihrer Aufnahmen ermöglicht, kann dieser Prozess weiter automatisiert werden. Zukünftig könnte somit der jeweils aktuelle Bauzustand erfasst und weiterverarbeitet werden. Dadurch kann schnell und unkompliziert auf eventuelle Terminprobleme reagiert werden. Eine stetige Bauprozessüberwachung könnte somit auch dezentral und vor allem auf aktuellen Daten basierend durchgeführt werden.

Zur Umsetzung dieser Idee sollen in dieser Thesis die folgenden Thematiken behandelt werden:

Neben einer umfangreichen Hintergrund- und Literaturrecherche über den aktuellen Stand im Bereich der dreidimensionalen Bauwerksmodellierung unter Berücksichtigung der Zeit (4D) wird ein besonderes Augenmerk auf Möglichkeiten zur Abbildung von Bauablauf-Alternativen und der Visualisierung selbiger gelegt.

Dabei werden aktuelle Dateistandards wie etwa die Industry Foundation Classes (IFC) untersucht sowie ein besonderes Augenmerk auf die Abbildung von Abhängigkeiten im Bauprozess und deren Visualisierung anhand der Graphentheorie behandelt.

Zusätzlich soll ein Software-Tool entwickelt werden, welches ein vorhandenes 3D-Modell importieren sowie darstellen kann. Dabei werden aktuell verfügbare IFC-Bibliotheken verglichen und auf ihre Tauglichkeit für das geplante Projekt untersucht. Des Weiteren sollen Funktionalitäten konzipiert und umgesetzt werden, um die Bauteilabhängigkeiten graphbasiert visualisieren zu können und Ablauf-Alternativen darstellen zu können.

1.3 DFG-Forschungsprojekt

Basierend auf diesen Ergebnissen soll im Rahmen eines Forschungsprojekts vertieft an der Erkennung von Bauteilen und einer automatisierten Ablaufplanung gearbeitet werden.

Die Deutsche Forschungsgesellschaft hat in diesem Zusammenhang einen Antrag der Lehrstühle für Computergestützte Modellierung und Simulation sowie Photogrammetrie der TU München genehmigt, der sich mit der automatisierten Erfassung des aktuellen Bauzustands befasst.

Unter dem Titel „Entwicklung eines automatisierten Verfahrens zur Baufortschrittskontrolle auf Basis der Integration von Punktwolkeninterpretation und 4D-Bauwerksmodellierung“[1] soll ein Verfahren entwickelt werden, auf Basis einer automatisiert erstellten Punktwolke einen Abgleich mit einem vorhandenen 3D-Modell durchzuführen.

Dabei sollen mit einer kalibrierten Digitalkamera mithilfe von Methoden der photogrammetrischen Aufnahme und Verarbeitung Fotos von der Baustelle gemacht werden. Um eine automatisierte Umsetzung im Sinne der Arbeitsersparnis zu realisieren, wäre ein mögliches Einsatzszenario, die Kameras an einem oder mehreren Kränen anzubringen. Die aufgenommenen Fotos sollen im Anschluss analysiert und in 3D-Punktwolken transformiert werden, welche dann mit einem vorhandenen digitalen dreidimensionalen Modell des geplanten Gebäudes abgeglichen werden. Eine besondere Herausforderung ist es hierbei, geeignete Verfahren zu entwickeln, um eventuelle Bauungenauigkeiten sowie Behelfskonstruktionen softwaregestützt zu erkennen.

Die Idee dieses Forschungsprojektes lässt sich grundsätzlich für jede Baustelle anwenden. Zielgruppe dieser Forschungsarbeit sind jedoch besonders große und komplexe Bauvorhaben, da der Aufwand der Automatisierung bei kleinen Projekten sehr kritisch zu hinterfragen ist. Der Kosten/Nutzen-Faktor ist hier erwartungsgemäß sehr gering, da sich bei kleineren Baustellen in der Regel auch manuell ohne größere Probleme ein Überblick über den aktuellen Bauverlauf im Rohbau schaffen lässt.

1.4 Aufbau der Arbeit

In Kapitel 2 wird das grundlegende Konzept der Baufortschrittserkennung beschrieben. Daraufhin werden in Kapitel 3 die für diese Aufgabenstellung wichtigen Grundbegriffe der Bauinformatik sowie deren zugehörige Technologien vorgestellt. Auch werden die für dieses Projekt wichtigen Grundlagen im Bereich der Photogrammetrie erläutert.

[1] DFG GEPRIS, http://gepris.dfg.de/gepris/OCTOPUS/;jsessionid=0FD3AF0D071D4531366821C7E40F4023?context=projekt&id=220340609&module=gepris&task=showDetail

Anschließend wird in Kapitel 4 vertieft auf die 4D-Modellierung im Bauwesen eingegangen und bisherige Forschungsergebnisse vorgestellt.

Darauf aufbauend wird in Kapitel 5 auf den Bauablauf sowie Ablauf-Alternativen, die sich während der Bauphase ergeben, eingegangen.

In Kapitel 7 werden die für das geplante Software-Tool benötigten Technologien erläutert. Anschließend wird in Kapitel 8 die Implementierung des Software-Tools eingehend beschrieben und erläutert, wie gewonnene Erkenntnisse angewendet werden.

Im darauf folgenden Kapitel 9 wird eine Baustelle vorgestellt, anhand derer der Bauablauf überwacht und basierend auf dem aktuellen Forschungsstand untersucht werden soll.

Abschließend (Kapitel 10) wird eine Zusammenfassung und ein Ausblick auf die geplanten Arbeiten während des Deutsche Forschungsgesellschaft (DFG)-Projekts gegeben.

Kapitel 2

Konzept

Wie bereits in Kapitel 1.2 erläutert, ist das Ziel dieser Arbeit die Unterstützung der automatisierten Baufortschrittskontrolle.

2.1 Herausforderungen

Besonders wichtig ist die Baufortschrittskontrolle auf großen Baustellen, bei denen viele Werkstoffe und Baumaterialien „just in time“ geliefert werden. Hierbei werden die Materialien direkt und ohne Lagerung verbaut, um so Kosten sparen zu können. Treten auf einer Baustelle Verzögerungen ein, ist es wichtig, diese so schnell wie möglich zu erkennen, um gegensteuern zu können und gegebenenfalls Lieferungen verschieben zu lassen. Diese Problematik trifft selbstverständlich in geringerem Umfang auch auf kleine Baustellen zu. Verzögerungen sind auch in der Personalverwaltung und bei anderen Ressourcen von Bedeutung, da Fehlplanungen stets mit Zusatzkosten verbunden sind.

Die Prozessüberwachung ist jedoch gerade bei großen Bauprojekten mit hohem Aufwand verbunden. Aufgrund der Komplexität von Bauprojekten im Allgemeinen, und der Vielzahl an einzelnen Bauteilen im Besonderen, ist die Fortschrittskontrolle schwer umsetzbar und bindet viele Ressourcen. Deshalb wäre hier eine Automatisierung sinnvoll, um schnell und exakt auf Änderungen reagieren zu können.

Besondere Herausforderungen bei der Erfassung des aktuellen Bauzustandes sind durch die Methoden der Photogrammetrie gegeben. Nachdem die Aufnahmen mit handelsüblichen Kameras aufgenommen werden, ist auf den Aufnahmen nur der jeweilige Sichtbereich zu sehen. Durch Verschattungen von Bauteilen aufgrund anderer Bauteile oder Baubehelfe sind die relevanten Elemente später nicht sichtbar. Dieses Problem kann teilweise durch Aufnahmen aus verschiedenen Perspektiven gelöst werden. Bauteile, die sich innerhalb des Gebäudes befinden, können von außen jedoch nicht aufgezeichnet werden.

Diese Herausforderung muss mit anderen Hilfsmitteln angegangen werden. Hierzu bieten sich unter anderem Abhängigkeitsgraphen an (Abschnitt 5.8).

Eine weitere Herausforderung bei der Erfassung des Bauzustandes ist die Ungenauigkeit bei der Punktwolkenaufnahme. Diese entsteht zum Beispiel aufgrund großer Entfernung zwischen Kamera und aufzunehmendem Objekt. Nachdem die Punktwolken aus mehreren zusammengesetzten Bildern generiert werden, ergeben sich zusätzliche Ungenauigkeiten durch zu geringe örtliche Abstände zwischen den Aufnahmepunkten. Diese Probleme werden aktuell in der Photogrammetrie behandelt.

2.2 Konzept

Der geplante Ablauf der Bauzustandserfassung wird im Folgenden überblicksmäßig erläutert.

2.2.1 Erfassung des aktuellen Bauzustands

Die Erfassung des Bauistzustands soll mit handelsüblichen Digitalkameras erfolgen, um eine kostengünstige und universell einsetzbare Einsetzbarkeit zu gewährleisten. Denkbar wäre die Montage einer Kamera an einem Kran oder ähnlichen Fixpunkten, von denen aus permanent Fotos gemacht werden.

Um die Bilder digital auswerten zu können und in eine dreidimensionale Punktwolke zu überführen, werden Methoden der Photogrammetrie benötigt (Abschnitt 3.1).

Beispielhaft ist in Abbildung 2.1 die Punktwolke der Projektbaustelle (Kapitel 9) dargestellt. Der Bau befindet sich in einer sehr frühen Phase, in der gerade das Kellergeschoss verschalt und betoniert wird.

2.2.2 Detektion von Bauteilen

Sobald die Aufnahmen in Punktwolken umgewandelt worden sind, können diese mit Hilfe von Algorithmen weiterverarbeitet werden. Um die Baufortschrittskontrolle zu automatisieren, ist es nötig, die einzelnen Bauteile im Modell zu erkennen. Ein möglicher Ansatz hierfür ist, die Punktwolke in Flächen aufzuteilen, welche in einem ersten Schritt eine Hüllfläche ergeben und als mögliches Bauteil erkannt werden. In einer fortgeschrittenen Variante wäre auch ein direkter Abgleich mit einem vorhandenen Building Information Model denkbar.

Eine besondere Herausforderung dabei ist es, Behelfskonstruktionen wie zum Beispiel Schalungen herauszufiltern (siehe Abschnitt 2.1).

Abbildung 2.1: Punktwolke der Projektbaustelle in der Bauphase. Quelle: Sebastian Tuttas

2.2.3 Abgleich mit dem digitalen Modell

In Kapitel 3.2 wird das Konzept der intelligenten digitalen Modelle eingeführt. Für das geplante Vorhaben ist es unumgänglich, dass ein solches dreidimensionales, digitales Modell vorhanden ist.

Diese Modelle enthalten Informationen über die einzelnen Bauteile, darunter neben den Materialeigenschaften auch die jeweilige Geometrie. Mit Hilfe dieser Daten soll ein Abgleich mit dem Ist-Zustand der Baustelle erfolgen, um so beurteilen zu können, welche Bauteile vorhanden sind.

2.2.4 Bestimmung von Abweichungen im Zeitplan

Am Ende des geplanten Verfahrens steht der Abgleich des erfassten Zustands mit dem geplanten Bauzeitplan. Dazu muss das digitale Modell mit Zeitdaten angereichert werden. Anhand derer kann so eine Aussage über die Einhaltung oder eventuelle Abweichungen im Bauprozess gemacht werden. Durch diese Methode soll der gesamte Vorgang der Prozessüberwachung unterstützt und vereinfacht werden, da die mühsame manuelle Erfassung des Bauzustandes und dessen Abgleich mit dem Ablaufplan entfallen können.

2.2.5 Nutzung von Prozessgraphen

In Kapitel 5 wird vertieft auf die Abbildung von Bauteilabhängigkeiten mit Hilfe von Graphen eingegangen.

Diese beschreiben die technologischen Abhängigkeiten (siehe Kapitel 5.4), welche den Bauablauf vorschreiben.

Kapitel 3

Technische Grundlagen

Das Fachgebiet der Bauinformatik hat sich in den letzten Jahren rasant weiterentwickelt. Gerade im Bereich der drei- und vierdimensionalen Baumodelle machte sich dieser Innovationsschub bemerkbar. In diesem Kapitel werden nun die technischen Grundlagen bezogen auf den Bereich der digitalen Bauablaufplanung erörtert und in Zusammenhang mit den Zielen dieser Thesis gebracht.

3.1 Photogrammetrie

Die Photogrammetrie soll in diesem Projekt zur Erfassung des jeweils aktuellen Bauzustands verwendet werden.

Dabei umfasst die Photogrammetrie Verfahren, die anhand von Fotografien von Objekten deren Form, Lage und Abmessung bestimmen können. Aufgrund der Reichweite von bis zu 100 Metern eignet sich die Photogrammetrie sehr gut für die Erfassung von größeren Objekten wie zum Beispiel Baustellen (Tuttas, 2013).

Der große Vorteil dieses Verfahrens ist, dass man im Vergleich zur dreidimensionalen Lasermessung keine teuren Gerätschaften benötigt. Des Weiteren entfällt eine aufwendige Justierung des Gerätes vor Ort. Es genügt eine handelsübliche (digitale) Spiegelreflexkamera, welche vor dem ersten Einsatz einmalig kalibriert werden muss. Dabei wird in der Regel der Zoom des Objektives fixiert und die Brennweite auf „*unendlich*“ gestellt.

Dadurch ist gewährleistet, dass sämtliche Bilder, die für eine Messung geschossen werden, miteinander kombiniert werden können.

Um ein Gebäude dreidimensional erfassen zu können, ist es unabdingbar, viele Bilder aus verschiedenen Perspektiven aufzunehmen, um später eine 3D-Rekonstruktion am Computer

umsetzen zu können. Wichtig ist dies auch, um zum Beispiel Objekte erfassen zu können, die aus der Sicht von einem einzigen Standort aus verdeckt wären.

Die Fotos können im Nachhinein mit Hilfe von Spezial-Software kombiniert werden. Diese kann darauf basierend Punktwolken erstellen, welche dann weiterverarbeitet oder betrachtet werden können. Eine bekannte Open Source Lösung zum Anzeigen und Bearbeiten von Punktwolken ist zum Beispiel MeshLab (http://meshlab.sourceforge.net/).

Um mehrere Aufnahmen kombinieren zu können, ist es empfehlenswert, mit sogenannten Zielmarken (siehe Abbildung 3.1) zu arbeiten. Diese sollten auf allen Fotos sichtbar sein und können dann als Fixpunkte markiert werden. Dadurch wird die Genauigkeit der Schnittflächen zwischen den einzelnen Aufnahmen deutlich verbessert. Des Weiteren kann bei größeren Distanzen so eine gewisse Messungenauigkeit ausgeglichen werden.

Abbildung 3.1: Zielmarken auf der Demonstrationsbaustelle. *Quelle: eigene Aufnahme*

Ein weiterer Vorteil der Zielmarken ist die Möglichkeit, diese absolut zu verorten. D.h. man vermisst diese nach geodätischen Vorgaben und kann die erzeugte Punktwolke dadurch verorten und in bestehende Karten und Modelle integrieren.

3.2 Building Information Modeling

Building Information Modeling (BIM) ist die wohl bedeutendste Entwicklung im digitalen Baubereich der letzten Jahre.

Mit diesem Begriff wird die Methode beschrieben, den gesamten Lebenszyklus eines Gebäudes digital und zentral zu beschreiben (siehe Abbildung 3.2). Das zu diesem Prozess gehörige digitale Modell nennt man *Building Information Model* (Eastman *et al.*, 2011).

Building Information Model

In dem *Building Information Model* werden sämtliche Informationen abgespeichert, die für das Gebäude von Bedeutung sind. Zum einen sind dies Daten, welche die Geometrie der einzelnen Bauteile eines Modells beschreiben. Zusätzlich können die Eigenschaften dieser Bauteile und deren Lage und Abhängigkeiten zueinander hinterlegt werden.

Außerdem ist es möglich, Bauprozessdaten sowie Informationen über den weiteren Lebenszyklus des Gebäudes abzuspeichern. Hierzu gehören sämtliche Daten, die für die in Abbildung 3.2 genannten Prozesse relevant sind.

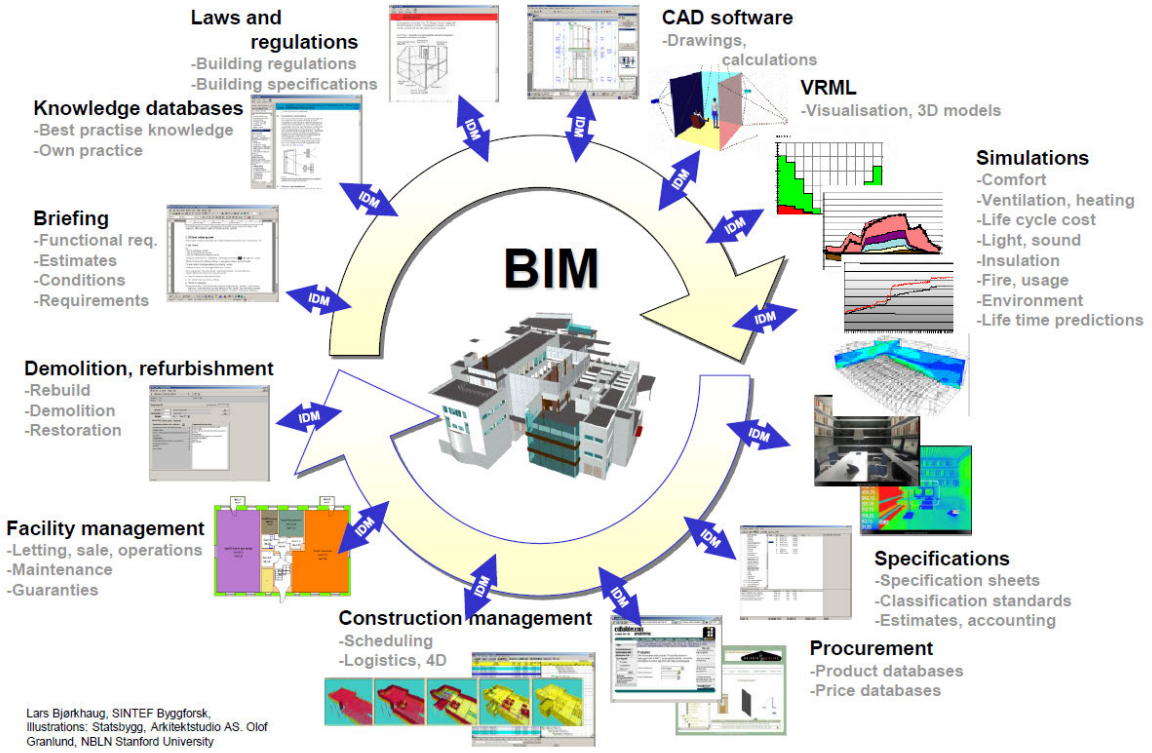

Abbildung 3.2: BIM Schema. Abbildung von: Byggforsk, Olof Granlund, NBLN University of California, Stanford University

Das Spektrum der erfassten Prozesse umfasst die gesamte Lebensdauer eines Bauwerks. Hierzu gehören zu Beginn die Planung mit der Erfassung sämtlicher Anforderungen und rechtlicher Vorgaben. Anschließend das Erstellen erster Modelle und Berechnungen, welche im Laufe des Planungsprozesses detaillierter ausgeführt werden. In den Kreislauf spielen auch sämtliche Visualisierungen und Simulationen mit hinein. Mithilfe des BIM-Kreislaufes werden auch die Informationen zur Beschaffung und Fertigung von Bauteilen abgebildet und gespeichert. Neben dem exakten Bauablauf werden nach Fertigstellung des Gebäudes auch die Daten der Gebäudetechnik in dem *Building Information Model* abgespeichert, welche zum Betrieb und für die Wartung und Instandhaltung benötigt werden. Der Kreis schließt sich mit dem Abriss des Gebäudes. Nachdem sämtliche Informationen über die einzelnen Bauteile und technischen Anlagen vorhanden sind, kann der Rückbau effizient und strukturiert ablaufen.

3.2.1 Vision

Übergreifendes Ziel des BIM ist es, das Gebäude bereits in der frühen Planungsphase des Architekten zu erfassen und entsprechend zu dokumentieren. Es erfolgt ein ständiger Austausch mit sämtlichen Projektbeteiligten, wie zum Beispiel Bauingenieuren oder Gebäudetechnikern. Diese arbeiten gemeinsam an dem dreidimensionalen Modell.

Noch während der Planungsphase fließen alle erforderlichen Änderungen direkt in das Projekt mit ein. Die Terminplanung und die Mengenermittlung für sämtliche Baumaterialien wird direkt aus dem Building Information Model abgeleitet.

Durch die frühzeitige Kooperation können mögliche Konfliktpotenziale, schon lange bevor sie auftreten, proaktiv erkannt und behoben werden.

Während der Bauphase kann durch kontinuierliches Monitoring jederzeit der aktuelle Bauzustand ermittelt und genaue Aussagen über die verbleibende Bauzeit und entstehende Kosten gemacht werden. Die Betreuung des Gebäudes nach Fertigstellung wird durch detaillierte Technikpläne enorm erleichtert.

Am Ende des Kreislaufes steht der Rückbau. Aufgrund der genauen Kenntnis der Baumaterialien wird eine korrekte Entsorgung gewährleistet.

3.2.2 Stand der Technik

Aktuell entspricht die Vision von BIM noch nicht der Realität. So führen mittlerweile sämtliche großen Softwarehäuser wie zum Beispiel Autodesk, Graphisoft oder Nemetschek BIM-Software, mit deren Hilfe bereits wichtige Schritte von der Modellierung bis hin zur Berechnung von Gebäuden bewältigt werden können.

Jedoch funktioniert das Zusammenspiel zwischen den einzelnen Sparten noch nicht zufriedenstellend. Das liegt vorwiegend an unzureichenden Datenaustauschmöglichkeiten.

In Deutschland ist ein weiterer wichtiger Punkt die Auslegung der meisten Verträge, welche eine Weitergabe von Projektdaten ausdrücklich untersagen. Viele Architekturbüros geben ihre digitalen Entwürfe nur ungern weiter, da sie damit die Kontrolle über deren weitere Verbreitung verlieren.

Um diese Problematik zu beheben, müssten Gesetzesänderungen umgesetzt werden. In England wird zum Beispiel ab dem Jahr 2016 gefordert, dass Bauwerke für die öffentliche Hand mit BIM-Technologie geplant und gebaut werden müssen (UK-HM-Government, 2012). Dadurch wird der digitale Austausch gefördert, um den BIM-Standard zu etablieren. Deshalb setzen dort bereits jetzt viele Ingenieurbüros auf diese neue Technologie (Brandt, 2012).

Auch die Baufortschrittskontrolle direkt auf der Baustelle funktioniert wie bereits eingangs erwähnt weitestgehend noch ohne digitale Unterstützung. Aktuell werden deshalb gerade in diesem Bereich große Anstrengungen unternommen, um auch hier die Vorteile der BIM-Technologie zu realisieren.

Eine gängige Lösung ist die Nutzung von Tablet-PCs, mit deren Hilfe man die digitalen Modelle auf die Baustelle bringt. Am *Lehrstuhl für Computergestützte Modellierung und Simulation* der TU München wurden hier zum Beispiel von Matthias Andrae im Rahmen seiner Masterarbeit mit dem Titel „Entwicklung eines Mangelaufnahme Systems auf Mobilen Geräten für den Einsatz bei der Objektüberwachung zur weiteren zentralen Verarbeitung“ Untersuchungen vorgenommen (Andrae, 2012).

Autodesk hat vor wenigen Wochen mit BIM 360 Field ebenfalls eine „On site“-Lösung vorgestellt, die den BIM-Gedanken auf der Baustelle etablieren soll (Autodesk, 2013a).

3.2.3 Relevanz und Einordnung in diese Thesis

BIM stellt mittlerweile einen wichtigen Teilbereich der angewandten Bauinformatik dar. Auch diese Thesis und das anschließende Forschungsprojekt gliedern sich in diese Kategorie ein und weisen typische BIM-Merkmale auf.

Die Prozessüberwachung einer Baustelle findet während der Ausführungsphase statt. Im Rahmen dieser Arbeit wird davon ausgegangen, dass vorhandene dreidimensionale Modelle genutzt werden, um diese mit dem Ist-Zustand auf der Baustelle abzugleichen. Um daraus sinnvolle Rückschlüsse ziehen zu können, ist es von großer Bedeutung, dass es sich hierbei um ein „intelligentes“ Modell handelt - also ein Building Information Model, in welchem zusätzliche Informationen wie die geplante Konstruktionsdauer oder Abhängigkeiten zu anderen Bauteilen hinterlegt sind.

3.3 Industry Foundation Classes

Durch die Digitalisierung der Bauwelt und die Entwicklung des BIM-Grundgedankens wurde ein einheitlicher und offener Standard zum Datenaustausch nötig. Aus diesem Grund hat sich im Jahr 1994 auf Initiative von Autodesk und anderen Firmen die „Industry Alliance for Interoperability“ gegründet, welche heute unter dem Namen „buildingSMART“ besteht (buildingSMART, 2013b).

Das Ziel dieser Organisation ist die Entwicklung offener Standards zum digitalen Austausch von Gebäudemodellen. Zu diesem Zweck wurden mehrere Richtlinien entwickelt, welche in Abbildung 3.3 dargestellt werden.

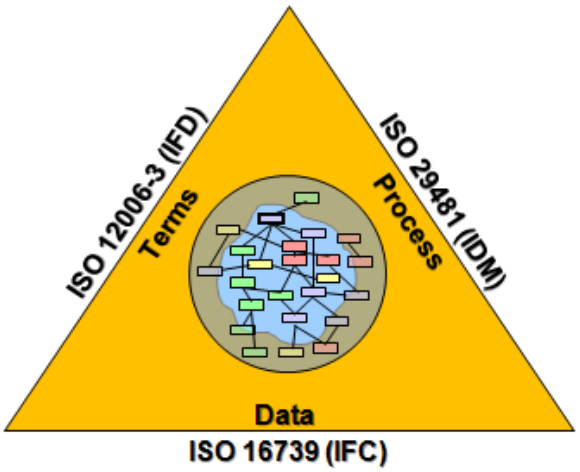

Abbildung 3.3: IFC Triangle. Abbildung von: www.buildingsmart-tech.org

Zum einen besteht der Standard *Information Delivery Manual (IDM)*, welcher die Prozesse der Bauplanung, Ausführung und den Betrieb von Bauwerken beschreibt. Daneben existiert das *International Framework for Dictionaries (IFD)*, welches die Schnittstelle zwischen den Produktinformationen und dem Building Information Model darstellt. Dort wird definiert, wie die beiden Bereiche miteinander verknüpft sind.

Der bedeutendste und wichtigste Standard von buildingSMART sind jedoch die IFC. Das offene Dateiformat wird mittlerweile von allen großen Softwarelösungen im Baubereich unterstützt. Es dient dem standardisierten Dateiaustausch von Building Information Models mit sämtlichen ihnen zugeordneten Eigenschaften. Dazu zählen neben den Bauteilinformationen auch die 4D-Daten, also Zeitdaten, welche den Bauablauf definieren.

3.3.1 Entwicklung der IFC

Am 12. März 2013 wurde der Entwurf der IFC4, der vierten Version der IFC veröffentlicht. Parallel dazu ist die IFC als ISO-Standard (ISO 16739) registriert worden.

Mit diesem Entwurf wurden einige weitreichende Neuerungen eingeführt, die unter anderem die Bauablaufplanung betreffen. Auf diese Änderungen wird in Kapitel 3.3.3 eingegangen.

Die aktuell in kommerziell verfügbarer Software genutzte Version ist jedoch noch IFC2x3. Es wird erwartet, dass es noch mindestens ein Jahr dauert, bis die neue Version flächendeckend unterstützt wird (buildingSMART, 2013a).

3.3.2 Datenformate

Die IFC sind ein Datenmodell, das mithilfe der Datenmodellierungssprache EXPRESS definiert wird. Diese Datenmodellierungssprache ist Teil des Standard for the Exchange of Product model data (STEP) und als ISO-Standard 10303-11 standardisiert.

IFC

Das Standardformat der IFC ist das IFC-Format. Es basiert auf dem *STEP*-Datenformat. Als Datenmodellierungssprache kommt EXPRESS zum Einsatz (ISO 10303-11, 1998).

```
#43 = IFCRELAGGREGATES('0khirZNU18ERoHZb9vmQvA', #2, 'ProjectContainer', 'ProjectContainer for Sites', #1, (#23));
#44 = IFCRELCONTAINEDINSPATIALSTRUCTURE('0U$K90gfb0dh9CajiB18yp', #2, 'Default Building', 'Contents of Building Storey', (#45, #149), #35);
#45 = IFCWALL('2Ny_EObwL1_fndzvzqPHqY', #2, 'Wall xyz', 'Description of Wall', $, #46, #51, $);
#46 = IFCLOCALPLACEMENT(#36, #47);
#47 = IFCAXIS2PLACEMENT3D(#48, #49, #50);
#48 = IFCCARTESIANPOINT((0., 0., 0.));
#49 = IFCDIRECTION((0., 0., 1.));
#50 = IFCDIRECTION((1., 0., 0.));
#51 = IFCPRODUCTDEFINITIONSHAPE($, $, (#72, #101));
#52 = IFCPROPERTYSET('3fW0k$zSz488FIpjxPbOBS', #2, 'Pset_WallCommon', $, (#53, #54, #55, #56, #57, #58, #59, #60, #61, #62));
#53 = IFCPROPERTYSINGLEVALUE('Reference', 'Reference', IFCTEXT(''), $);
#54 = IFCPROPERTYSINGLEVALUE('AcousticRating', 'AcousticRating', IFCTEXT(''), $);
#55 = IFCPROPERTYSINGLEVALUE('FireRating', 'FireRating', IFCTEXT(''), $);
#56 = IFCPROPERTYSINGLEVALUE('Combustible', 'Combustible', IFCBOOLEAN(.F.), $);
#57 = IFCPROPERTYSINGLEVALUE('SurfaceSpreadOfFlame', 'SurfaceSpreadOfFlame', IFCTEXT(''), $);
#58 = IFCPROPERTYSINGLEVALUE('ThermalTransmittance', 'ThermalTransmittance', IFCREAL(2.400E-1), $);
#59 = IFCPROPERTYSINGLEVALUE('IsExternal', 'IsExternal', IFCBOOLEAN(.T.), $);
#60 = IFCPROPERTYSINGLEVALUE('ExtendToStructure', 'ExtendToStructure', IFCBOOLEAN(.F.), $);
#61 = IFCPROPERTYSINGLEVALUE('LoadBearing', 'LoadBearing', IFCBOOLEAN(.F.), $);
```

Abbildung 3.4: IFC Beispielcode, basiert auf einem Revit-Export.

ifcXML

Das ifcXML-Format (Abbildung 3.5) basiert auf dem Extensible Markup Language (XML)-Standard. Hierzu wird das in EXPRESS formulierte Datenmodell auf das XML-Schema abgebildet. Es ist im ISO-Standard 10303-28 als *STEP-XML* definiert und soll vor allem den Im- und Export sowie die Konvertierung von IFC-Daten erleichtern. Die Dateigröße ist durch das XML-Format allerdings bis zu 400% größer als das IFC-Format.

Da die Nachfrage nach dem XML-Format aufgrund dessen einfacher Implementierung sehr hoch ist, wurde mit IFC4 eine komprimierte IfcXML-Version namens simple ifcXML eingeführt. Diese ermöglicht eine Reduzierung der XML-Elemente auf 14% im Vergleich zur alten Version.

```
<IfcWall id="i45">
  <GlobalId>2fvP_OV8z39gMD3YIHVyWr</GlobalId>
  <OwnerHistory>
    <IfcOwnerHistory xsi:nil="true" ref="i2"/>
  </OwnerHistory>
  <Name>Wall xyz</Name>
  <Description>Description of Wall</Description>
  <ObjectPlacement>
    <IfcLocalPlacement xsi:nil="true" ref="i46"/>
  </ObjectPlacement>
  <Representation>
    <IfcProductDefinitionShape xsi:nil="true" ref="i51"/>
  </Representation>
</IfcWall>
```

Abbildung 3.5: ifcXML Beispiel, basiert auf dem HelloWall-Beispiel.

ifcZIP

Das ifcZIP-Format ist eine komprimierte Version des IFC-Formats. Es komprimiert die Datei nach dem Pkzip 2.04g Algorithmus. Dadurch wird eine Reduzierung der Dateigröße um 20 - 40 % erreicht.

3.3.3 Objekte

Das Datenmodell der IFC ist objektorientiert. Sämtliche Informationen über Bauteile und deren zusätzliche Informationen werden in Objekten (engl.: *Entities*) gespeichert. Insgesamt stehen in der Version IFC2x3 653 verschiedene *Entities* zur Verfügung (buildingSMART, 2007a).

Als gängige Beispiele seien hier *IfcWall* für Wände, *IfcFooting* für Fundamente oder *IfcSlab* für Decken genannt.

Vererbungshierarchien

In EXPRESS können Objekte in Klassen beziehungsweise Unterklassen definiert werden. Dieses Konzept wird in gängigen objektorientierten Programmiersprachen wie C# oder Java umgesetzt (Fliegner, 2003).

Die Vererbung wird hier über *SUBTYPE OF* (Verweis auf Vaterklasse) sowie *SUPERTYPE OF* (Verweis auf Kindklasse) umgesetzt.

Referenzierung von Eigenschaften

Die Objekte können über ihre IDs miteinander verknüpft werden. In Abbildung 3.4 sieht man bei ID #44 sowie bei ID #52, dass diese über Eigenschaften aus anderen Objekten definiert werden und diese entsprechend referenzieren.

Abbildung 3.6 zeigt die durch die Referenzierungen entstehenden Abhängigkeiten, welche in *IfcRoot* ihren Ursprung finden. Diese werden in der grafischen Darstellungssprache EXPRESS-G dargestellt.

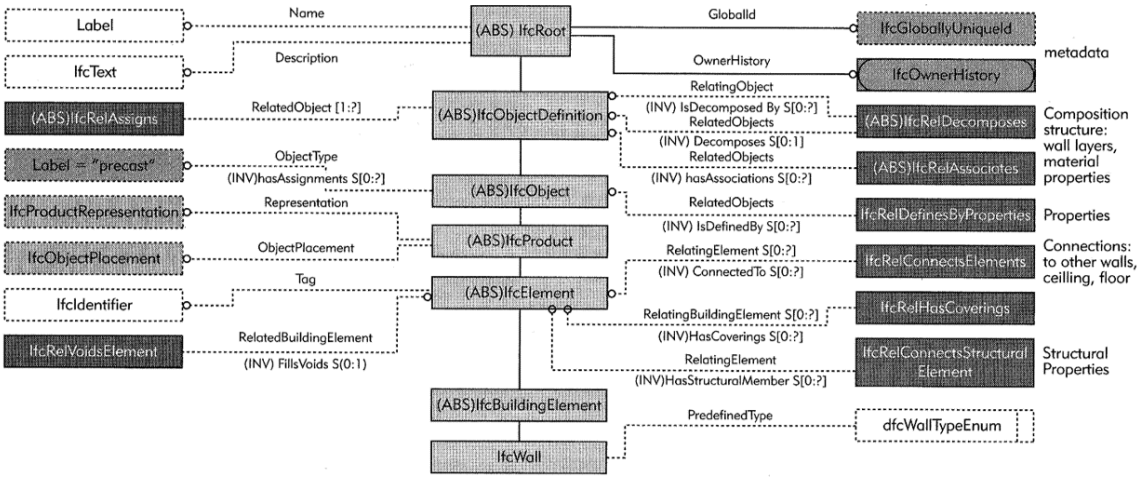

Abbildung 3.6: IFC Hierarchien in einem EXPRESS-G Diagramm. Quelle: (Eastman *et al.*, 2011)

Für die Bauablaufplanung ist das Objekt *IfcTask* von besonderer Bedeutung.

IfcTask

Das Objekt kann jedem beliebigen anderen Ifc-Objekt zugewiesen werden und dadurch diesem einen Arbeitsschritt zuordnen. Über das übergeordnete Objekt *IfcProcess* können dem Arbeitsschritt auch ein vorhergehender (*IsSuccessorFrom*) und ein nachfolgender (*IsPredecessorTo*) Prozess zugeordnet werden. Dadurch können logische Abhängigkeiten (Beschreibung siehe Abschnitt 5.4) zwischen einzelnen Bauteilen beschrieben werden.

Eine mit der IFC4 eingeführte Neuerung betrifft eine grundlegende und für das Prozessmanagement essentiell wichtige Planungseigenschaft. Bis zur Version IFC2x3 existierte keine geeignete Möglichkeit, einem Prozess einen Start- sowie Endzeitpunkt zuzuweisen. Es existieren zwar die *IfcTimeSeries*, diese sind jedoch laut buildingSMART eher für Wetterdaten und Messungen gedacht als für die Prozesssteuerung.

Dies ist der Grund, warum es bis zum jetzigen Zeitpunkt keine Software gibt, die 4D-Daten im IFC-Format abspeichern kann. So können gängige Programme wie zum Beispiel Auto-

desk Navisworks zwar IFC-Modelle einlesen und verwenden, das Speichern ist aber nur in proprietären Formaten möglich. Eine Weiterverwendung von mit diesen Produkten erstellten Terminplänen ist also zum jetzigen Zeitpunkt nicht möglich.

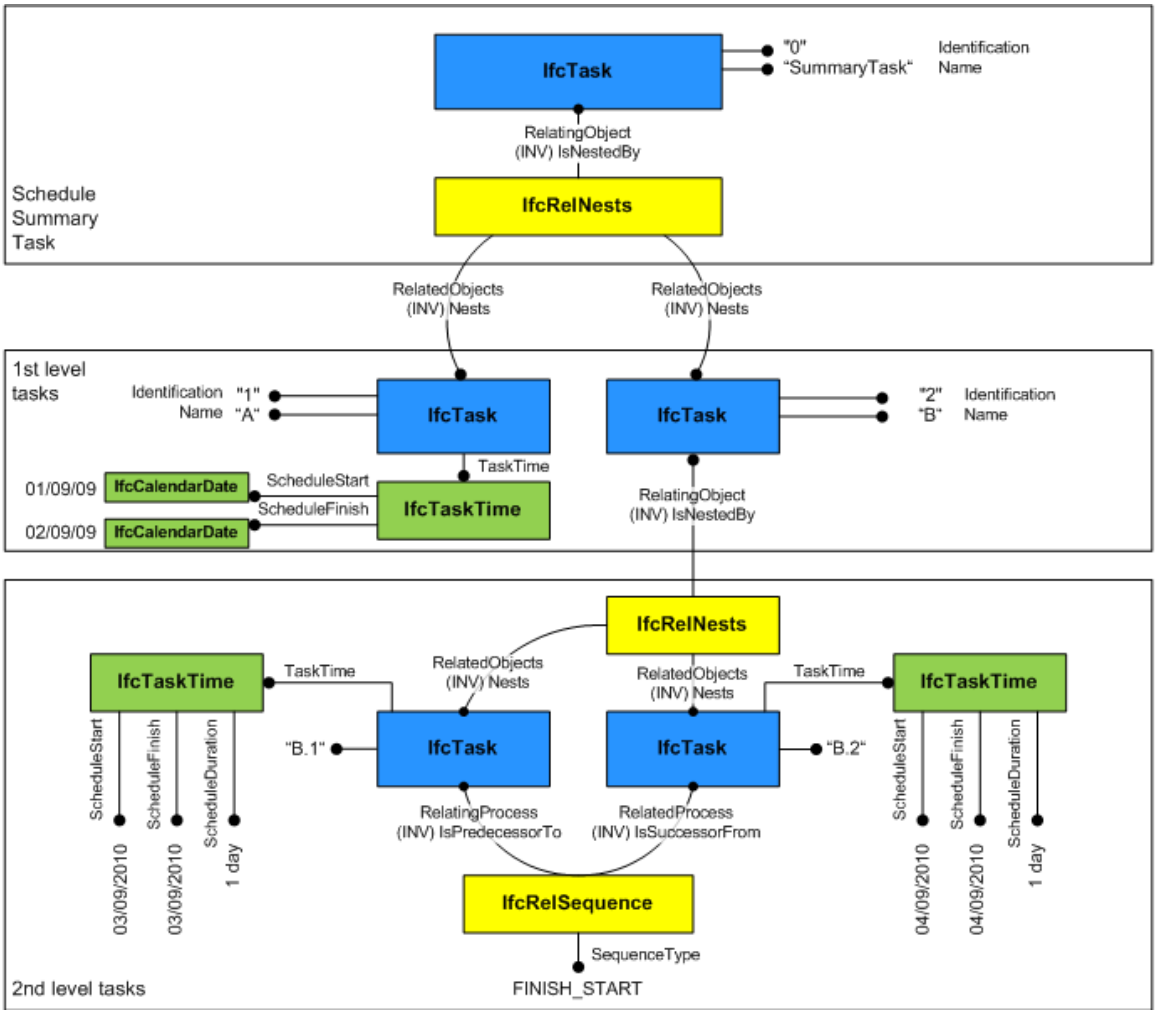

Abbildung 3.7: ifcTask Instantiation Diagram - www.buildingsmart-tech.org

Das dürfte sich aber mit der Einführung von IFC4 ändern. In Abbildung 3.7 wird (unter anderem) das neu eingeführte Objekt *IfcTaskTime* gezeigt, welches direkt *IfcTask* untergeordnet ist. Damit steht ein direktes Kindobjekt zur Verfügung, welches genau einem Prozess (*Task*) zugewiesen wird. Gut zu sehen ist in dem Diagramm auch die Nutzung der Vererbungshierarchie, welche es ermöglicht, einen höheren Detaillierungsgrad bei der Ablaufplanung zu erzielen. Hier wird auch gezeigt, welche Eigenschaften den einzelnen prozessrelevanten IFC-Objekten zugeordnet sind.

3.3.4 Geometrie

In der Informatik gibt es verschiedene Möglichkeiten, Geometrie zu beschreiben und darzustellen. Für das Building Information Modeling ist besonders die dreidimensionale Darstellung der Modelle von Bedeutung. Die einzelnen Bauteile können dabei durch mehrere Techniken repräsentiert werden.

Im Computer-aided design (CAD)-Bereich besonders verbreitet ist das Constructive Solid Geometry (CSG)-Schema (Foley *et al.*, 1994). Hier werden die einzelnen Elemente durch Grundkörper wie Quader, Würfel oder Zylinder erstellt und mit Hilfe von Operationen wie Vereinigung, Schnitt und Differenz kombiniert.

Ein weiteres, erwähnenswertes Schemata ist die Oberflächendarstellung (engl: *Boundary Representation*), bei der einzelne Objekte durch Freiformflächen dargestellt werden. Bekannte Freiformflächen sind Non-Uniform Rational B-Splines (NURBS). Diese, mathematisch definierten Kurven, können nahezu jede beliebige Geometrie darstellen.

In der ISO 10303-42:1994, *Industrial Automation Systems and Integration: Product Data Representation and Exchange*, werden die verschiedenen Möglichkeiten beschrieben, mit Hilfe derer Geometrie in den IFC beschrieben werden kann (buildingSMART, 2007b). Es besteht beispielsweise die Möglichkeit, Punkte direkt über deren Koordinaten, über parametrische Kurven oder auch über Offsetkurven zu bestimmen. Seit der IFC4 ist auch eine Darstellung mittels NURBS möglich. Die jeweilige Repräsentation eines Bauteils wird in dem IFC-Objekt *IfcShapeRepresentation* definiert und gespeichert.

All diese Möglichkeiten können von vielen Visualisierungstools nicht direkt dargestellt werden und müssen zur grafischen Ausgabe in Dreiecke umgewandelt werden. Diese Methode, die auch Parkettierung genannt wird, ist in der IFC4 als neue Beschreibungsmöglichkeit integriert, wird jedoch aktuell aufgrund der noch sehr geringen Verbreitung kaum verwendet.

3.3.5 Anwendung

Im Rahmen dieser Thesis ist geplant, das IFC-Format zu verwenden, um 3D-Modelle von Gebäuden in ein Softwaretool zu laden und dort weiterzuverarbeiten. Besonders wichtig ist dabei die Übermittlung von Bauteilinformationen, sowie der zugehörigen Geometrie, um diese darstellen zu können.

Die in IFC4 eingeführten zeitlichen Datenstrukturen erleichtern das geplante Vorhaben weiter. Zum einen ist der direkte Import der Prozessdaten möglich - und, falls diese noch nicht vorhanden sind, besteht hiermit nun die Möglichkeit, diese Daten im Programm zu aggregieren und dann gegebenenfalls in der IFC-Datei abzuspeichern.

3.4 Zusammenfassung

Die in diesem Kapitel erläuterten Technologien und Prozesse bilden die Grundlage für das beschriebene Konzept der automatisierten Baufortschrittsüberwachung.

Im Zusammenspiel von Building Information Modeling, der Photogrammetrie sowie den Industry Foundation Classes besteht die Möglichkeit die Automatisierung weiter voran zu treiben. Dabei bieten die Methoden der Photogrammetrie sehr gute Möglichkeiten, um den Ist-Zustand auf der Baustelle aufzunehmen. Der Grundgedanke des Building Information Modeling wird mit diesem Vorhaben fortgeschrieben. Die Prozessüberwachung, welche ein Bestandteil des BIM ist, wird durch die Automatisierung vereinfacht und schließt sich nahtlos in den bestehenden Kreislauf.

Für diese Thesis eignen sich die IFC hervorragend für den Datenaustausch. Als OpenSource-Standard mit wachsender Verbreitung, ist es mittlerweile mit nahezu allen Modellierungsprogrammen möglich, eine IFC-Datei des erstellten Modells zu erzeugen. Durch die Verwendung dieses Formats kann so das Einlesen von nahezu jedem Modell ermöglicht werden. Durch die geplante bessere Unterstützung von Bauprozessdaten, besteht für die Zukunft die Möglichkeit auch diese Daten gemeinsam mit dem 3D-Modell und den zugehörigen Bauteilinformationen zu importieren.

Im folgenden Kapitel soll nun der aktuelle Stand der Wissenschaft in der Baufortschrittskontrolle und -überwachung aufgezeigt werden.

Kapitel 4

4D-Modellierung im Bauwesen

Im Bereich der 4D-Modellierung im Bauwesen wurden in Deutschland bereits eine ganze Reihe von Forschungsvorhaben durchgeführt. In diesem Abschnitt wird der aktuelle Stand vorgestellt, der vor allem auf der Arbeit der entsprechenden Bauinformatik-Institute an den Universitäten Weimar, Berlin, Bochum und München beruht.

4.1 Definition

Im Bauwesen definiert die vierte Dimension die Zeit. Konkret wird ein dreidimensionales Modell also mit zeit- und prozess-relevanten Informationen angereichert. Dadurch wird erreicht, dass ein *Building Information Model* auch im Bereich der Prozessplanung verwendet werden kann.

Dies bietet eine Vielzahl an neuen Möglichkeiten zur Nutzung in der Bauinformatik.

In der BIM-Branche herrscht Uneinigkeit darüber, ob nicht nur vier sondern aktuell sechs Dimensionen existieren. Mit der fünften Dimension sollen die Kosten eines Bauwerks beziehungsweise der einzelnen Bauteile beschrieben werden. Die sechste Dimension stellt Informationen über die Gebäudetechnik (Facility Management) und deren Lebenszyklus zur Verfügung.

Diese Darstellung ist zwar umstritten, gibt aber einen Ausblick auf weitere Entwicklungsschwerpunkte im Bauwesen und dem Building Information Modeling.

4.2 Ablaufplanung

Zu Beginn der BIM-Ära wurde die Möglichkeit, Zeitdaten mit dem Building Information Model zu verknüpfen, primär für die Prozessplanung an sich genutzt.

Bekannte Planungstools wie BIMsight von Tekla Systems, RIB iTWO von RIB Software oder Navisworks von Autodesk (Abbildung 4.1) sind gute Beispiele für diese Phase der Projektplanung.

Deren Einsatzgebiet ist aber klar auf die Prozessverwaltung vor Beginn der Bauphase beschränkt und es ist bereits in diesem Stadium der Planung möglich, Kollisionskontrollen durchzuführen. Dadurch können Planungsfehler noch vor Baubeginn behoben werden.

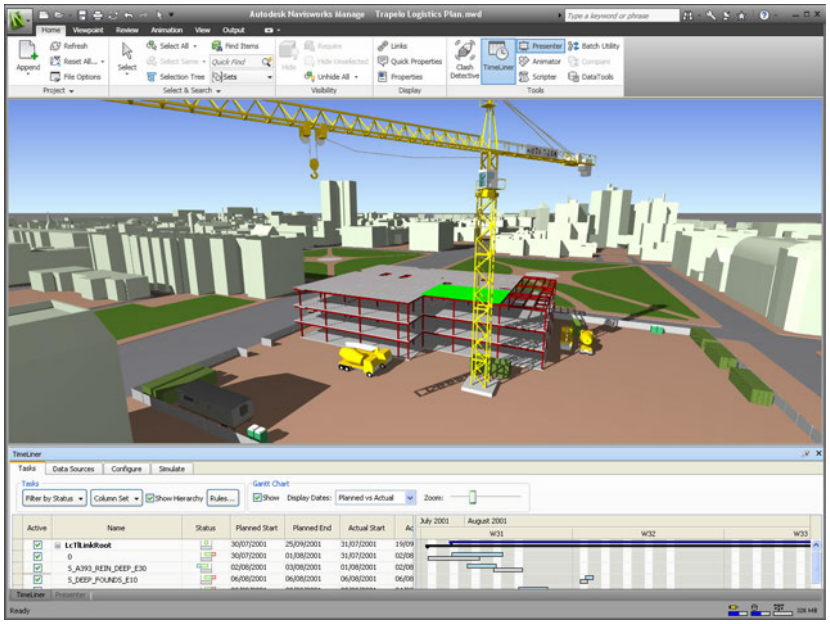

Abbildung 4.1: Bauprozessplanung mit Navisworks. Abbildung von: Autodesk.com

4.3 Abbildung von Bauteilen

Ablaufpläne können nur teilweise und unvollständig mit Hilfe von automatisierten Verfahren erzeugt werden, da logische und zeitliche Abhängigkeiten von vielen äußeren Einflüssen abhängig sind. Diese werden zum Teil von Projektplanern vorgegeben und müssen somit immer manuell eingegeben werden.

Anders sieht es bei den technologisch bedingten Abhängigkeiten und Prozessen aus. Auch hier ist eine endgültige und eindeutige Automatisierung nicht möglich.

Martin Fischer hat hier einen Ansatz definiert, der die einzelnen Prozesse in verschiedene Kategorien einteilt. Diese sind nach ihrem Detaillierungsgrad sortiert und können so je nach Bedarf gefiltert werden und bieten auch dem Prozessplaner Möglichkeiten, seine Arbeitsweise zu optimieren (Fischer & Aalami, 1996).

Auch Tauscher beschreibt in seiner Dissertation ein Verfahren, mit dessen Hilfe die Automatisierung zumindest teilweise ermöglicht wird (Tauscher, 2011). Er wählt einen objektorientierten Ansatz, um jedes Bauteil nach seinen Eigenschaften zu kategorisieren.

Dabei wird jedem Bauteil ein Prozess zugeordnet (siehe Kapitel 5.1). Anschließend werden wichtige Eigenschaften der Bauteile mit einer Prozessdatenbank verglichen, um diese Gruppen zuzuordnen und entsprechend einzuteilen. Gut verwendbare Eigenschaften sind etwa dessen Bauart oder auch das Baumaterial (Abbildung 4.2).

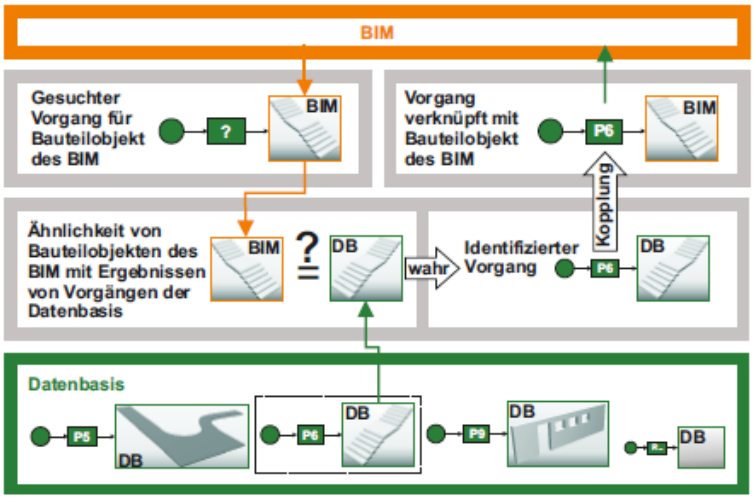

Abbildung 4.2: Prinzip des Vergleichs und der Zuordnung von Bauteilobjekten als Ergebnis von Vorgängen. Abbildung von: Eike Tauscher

Dadurch kann eine „halb-intelligente“ Softwareunterstützung für die Bestimmung von Prozessen umgesetzt werden.

Auch Wolfgang Huhnt widmet sich dieser Thematik und hat einen mathematischen Formalismus eingeführt, der auf Basis der Mengentheorie die Analyse der Abhängigkeiten von Prozessen ermöglicht (Huhnt, 2005).

Die hier genutzte Möglichkeit der Prozessbestimmung lässt sich so auf die Definition von Abhängigkeiten erweitern. Somit kann auch dieser Arbeitsschritt zumindest teilautomatisiert werden.

4.4 Bildgestützte Baufortschrittskontrolle

Die Fortschrittskontrolle auf der Baustelle wurde bereits mehrfach untersucht. Bisher wurden dabei meist 2D- jedoch zum Teil auch schon 3D-Abgleiche mit einem rekonstruierten Modell erstellt. Dabei wurde jedoch nur eingeschränkt auf ein *Building Information Model* als Referenz zurückgegriffen.

Dies soll im Rahmen des eingangs erwähnten Forschungsprojekts aufgegriffen werden. Dadurch soll direkt auf der Baustelle eine einfache, digitale Nutzungsmöglichkeit für die Prozessüberwachung mit Hilfe von BIM ermöglicht werden.

Im Bereich der Baufortschrittsüberwachung hat unter anderem Changmin Kim die Baufortschrittskontrolle anhand von Punktwolken aus Laserscannern untersucht (Kim *et al.*, 2013). Er stellt dabei fest, dass es mit Hilfe von Laserscannern eine Erkennung von Bauteilen und deren Abgleich mit einem 3D-Modell machbar ist, geht jedoch von einer strikten Befolgung des vorgegebenen Bauablaufplanes aus. Da dies in der Regel jedoch nicht der Fall ist, wird auf diese Problematik ein besonderes Augenmerk geworfen.

Kapitel 5

Bauprozessmodellierung

Mit der Bauablaufplanung wird im Allgemeinen die Organisation und zeitliche Planung der Bauphase eines Bauwerkes beschrieben.

Dabei sind die gesetzten Ziele unter Einhaltung von Termin-, Kosten- und Qualitätsvorgaben zu realisieren (Enge, 2009). In diesem Kapitel werden die verschiedenen Parameter und Eigenschaften beleuchtet, die während der Prozessplanung zu beachten sind.

Besonderes Augenmerk gilt hier den Abhängigkeiten und Alternativen im Bauablauf und wie diese in der Bauinformatik interpretiert und berücksichtigt werden.

5.1 Prozess

In der Baubranche hat der Begriff Prozess mehrere Bedeutungen. Allgemein beschreibt ein Prozess einen oder mehrere Arbeitsschritte in einem Bauprojekt. Der Prozess ist durch einen Start- und einen Endzeitpunkt definiert. Innerhalb eines Prozesses können wiederum Teilprozesse ablaufen (Huhnt, 2005). Beispielsweise ist der Vorgang „Betonwand erstellen“ ein Prozess. Er besteht aus mehreren Teilprozessen wie „Schalung errichten“ und „Betonieren“.

In dieser Thesis wird der Begriff Prozess dem Erstellen eines Bauteils gleichgesetzt. Eventuell nötige Teilprozesse oder gar nötige Vorprozesse werden nicht gesondert erwähnt, gleichwohl sie natürlich vorhanden sein können. Auf diesen Aspekt wird in Kapitel 4 intensiver eingegangen.

5.2 Projekt

Ein Projekt oder auch Vorhaben ist die Summe aller zusammengehörigen Prozesse, die ein Bauobjekt als Ziel haben. Vereinfacht gesagt kann jedes Bauwerk in seiner Gesamtheit als ein

Projekt gesehen werden. Diesem ist genau wie jedem Prozess ein Start- und ein Endzeitpunkt zugewiesen.

Bauprojekte sind durch ihre uneinheitlichen Randbedingungen stets ein Unikat. Dies unterscheidet sie zum Beispiel von Projekten aus der Automobilbranche. Jedes Projekt ist eine einmalige Abfolge von Prozessen. Dies macht die Planung von Prozessen in der Baubranche deutlich komplexer als in anderen Branchen, in denen die Fertigung immer identischer Teile am Fließband beziehungsweise in gleicher Art und Weise erfolgen kann.

5.3 Ressourcen

Der Begriff Ressource ist im Baubereich ein Überbegriff für verschiedene Vorgänge beziehungsweise Handlungen.

Ressourcen können sowohl Materialien, wie zum Beispiel die Menge an verfügbarem Beton, jedoch auch die Menge an verfügbaren Arbeitskräften sein.

Die Verfügbarkeit von Ressourcen ist eine der großen Randbedingungen in der Prozessplanung. Für die Planer gilt es hier, ein gutes Verhältnis aus verwendeten Ressourcen und dem gewünschten Ergebnis zu erzielen. Beispielsweise kann für zeitkritische Aufgaben die Ressource „Arbeitskraft“ erhöht werden, indem man mehr Personal einstellt. Dies verursacht jedoch erhöhten Kostenaufwand. Selbiges gilt für die Verfügbarkeit von Schalungswänden auf der Baustelle. Auch hier könnte man für eine Maximalauslastung eine hohe Anzahl an Wänden bereit stellen. Diese kosten jedoch in hoher Stückzahl mehr und werden dann oft nur zu Spitzenauslastungszeiten benötigt.

5.4 Abhängigkeiten

Prozesse auf einer Baustelle sind meist von vielen verschiedenen Einflüssen abhängig.

Folgende vier Kategorien sind dabei von (Würfele & Bielefeld, 2007) definiert worden:

- **technologisch bedingte, zwingende Abhängigkeiten:** Die wichtigsten und in keinem Fall umgehbaren Abhängigkeiten. Das Dach eines Hauses kann zum Beispiel nicht gebaut werden, bevor nicht die Wände darunter fertiggestellt worden sind.
- **vorgegebene, externe Randbedingungen:** Hierzu gehört zum Beispiel ein fest vorgeschriebener Fertigstellungstermin oder einzelne Zwischenabnahmen.
- **kapazitätsbedingte Abhängigkeiten:** Zu den kapazitätsbedingten Abhängigkeiten zählen vor allem Ressourceneinschränkungen wie das nicht ausreichende Vorhanden-

sein von Baumaterial, Personalengpässe oder Baugerätauslastungen. Diese werden in Abschnitt 6.2 beispielhaft beschrieben.

- **terminplantechnische Abhängigkeiten:** In diese Kategorie gehören Pufferzeiten zur Entzerrung des Terminplans und sonstige Ressourcenanpassungen.

Zusammenfassend ist festzustellen, dass die technologisch bedingten Abhängigkeiten oder auch kausalen Anordnungsbeziehungen die einzigen Randbedingungen sind, welche für ein aus konstruktionstechnischer Sicht fertig geplantes Gebäude in keinem Fall modifiziert werden können. Alle anderen Abhängigkeiten sind gegebenenfalls durch zusätzliche Investitionen in Ressourcen, entsprechende Neuplanungen oder Verzugszeiten anpassbar.

Des Weiteren ist hier auch klar der Unterschied zwischen zeitlich/ finanziellen und technologischen beziehungsweise strukturellen Abhängigkeiten zu sehen. Während erstere für das Bauprojekt an sich, dessen Entstehung und den Auftraggeber von essentieller Bedeutung sind, spielen diese bei der automatisierten Baufortschrittskontrolle eine eher untergeordnete Rolle. Diese Abhängigkeiten werden auch kapazitive Anordnungsbeziehungen genannt, sie werden in dieser Thesis jedoch nicht vertieft behandelt.

5.5 Anordnungsbeziehungen

In der DIN 69 900 sind zusätzlich zu den Abhängigkeiten sogenannte Anordnungsbeziehungen (engl.: *precedence relationships*) definiert (Zimmermann, 2009). Dabei entsprechen die technologischen Randbedingungen einer *Normalfolge*, welche so definiert ist, dass ein Prozess erst nach Fertigstellung eines anderen Prozesses beginnen darf.

Eine *Anfangsfolge* beschreibt eine Anordnungsbeziehung, bei der ein Prozess erst nach Beginn, jedoch ausdrücklich bereits vor Abschluss, eines anderen Prozesses starten darf. Ein Beispiel wären hier die Erdarbeiten auf einer Baustelle, die bereist vor Abschluss der Baustelleneinrichtung, aber erst nach deren Beginn, starten können.

Als *Endfolge* wird eine Anordnungsbeziehung bezeichnet, wenn ein Vorgang vor einem anderen abgeschlossen sein muss. Dies trifft zum Beispiel auf das Betonieren zu, das vor dem Glätten abgeschlossen sein muss. Wegen der schnellen Aushärtung des Betons kann jedoch bereits direkt nach dem Start des Betonier-Prozesses mit dem Glätten begonnen werden.

Als letzte Folge ist die *Sprungfolge* definiert. Diese ist eine Anordnungsbeziehung zwischen dem Anfang eines Vorgangs und dem Ende seines Nachfolgers. Damit kann beispielsweise der Beginn der Baustelleneinrichtung mit dem Ende des Räumens der Baustelleneinrichtung verbunden werden. Zeitgleich wird damit auch die Gesamtprojektdauer bestimmt.

Diese Beziehungen machen die Komplexität des Prozessmanagements deutlich. In dieser Thesis werden Prozesse jedoch ausschließlich vereinfacht über Normalfolgen dargestellt.

5.6 Gantt-Diagramme

Eine gängige Lösung, den Bauprozess zu visualisieren, sind Gantt-Diagramme oder auch Balkenplan genannt (Wilson, 2003). Dabei werden sämtliche Aktivitäten des Gesamtprozesses an einer Zeitachse angetragen (siehe Abbildung 5.1). Der Projektstrukturplan (engl. *work breakdown structure* (*WBS*)) zeigt hier, wie ein Projekt in einzelne Teilprozesse gegliedert wird.

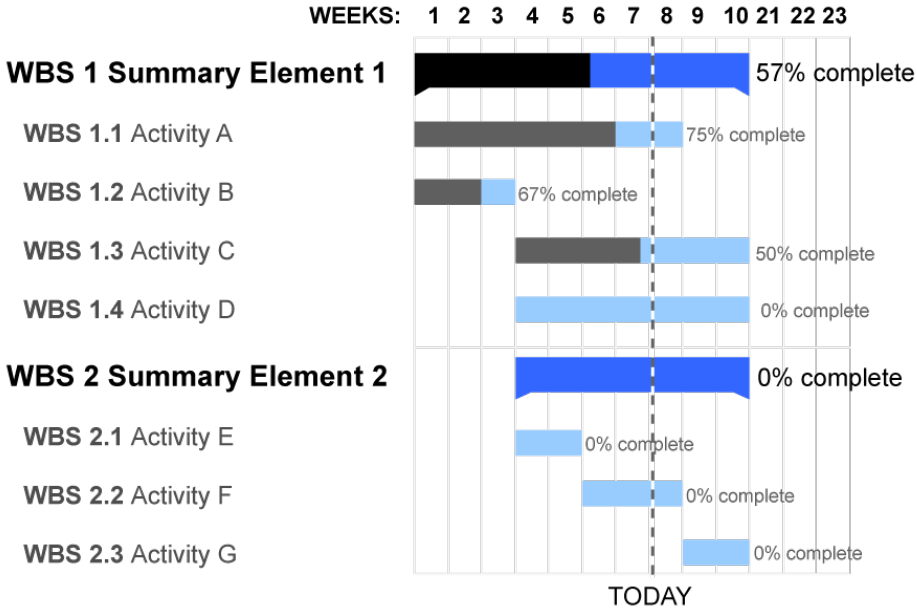

Abbildung 5.1: Gantt-Diagramm - (http://en.wikipedia.org/wiki/Gantt_chart)

Gantt-Diagramme haben den großen Vorteil, dass man auf einen Blick genau erkennen kann, welche Prozesse zu welchem Zeitpunkt ablaufen sollen. Man bekommt so schnell einen Überblick über eventuelle Verzögerungen im Projekt.

Des Weiteren können Abhängigkeiten gut sichtbar gemacht werden. Prozesse, die unabhängig voneinander ablaufen, können zur selben Zeit, also parallel, ausgeführt werden. Diese werden in einem Gantt-Diagramm untereinander im selben Zeitbereich angezeigt.

Technologisch voneinander abhängige Prozesse müssen nacheinander ausgeführt werden. Diese werden auf der Zeitachse somit nacheinander dargestellt.

Gantt-Diagramme eigenen sich gut, um den exakten zeitlichen Ablauf eines Prozesses, beziehungsweise einer Prozesskette darzustellen. Durch die nach Einzelprozessen aufgeschlüsselte Darstellungsweise können terminplantechnische oder auch kapazitätsbedingte

Abhängigkeiten sehr gut dargestellt werden. Einzig technologisch bedingte Abhängigkeiten können nicht eindeutig wiedergegeben werden.

5.7 Graphen

Ein Graph G beschreibt eine Menge von Objekten oder auch Knoten V, die durch Kanten E miteinander verbunden sind. Der Graph ist also ein Tupel G = (V,E) (Diestel, 1996). Mit Graphen können verschiedene Elemente, durch Knoten repräsentiert und miteinander über Kanten in Verbindung gebracht werden, um dadurch deren Beziehung zueinander darzustellen.

5.7.1 Gerichteter Graph

Bei einem gerichteten Graphen werden die Kanten durch Pfeile dargestellt um damit eine Richtung vorzugeben. Dies ist unter anderem von Bedeutung, um einseitige Abhängigkeiten (Abschnitt 5.8) exakt darstellen zu können.

5.7.2 Wege und Pfade

Wege beschreiben in der Graphentheorie eine Folge von aufeinanderfolgenden Knoten, die durch eine Kante verbunden sind. Pfade hingegen sind als Wege definiert, in denen kein Knoten mehrmals vorkommt (Henley & Williams, 1973).

5.7.3 Zusammenhang

In der Graphentheorie spricht man von Zusammenhang, wenn zwei Knoten über einen durchgängigen Weg miteinander verbunden sind. Als stark zusammenhängend wird ein gerichteter Graph beschrieben, bei dem zwischen dem Startknoten und jedem anderen Knoten einen Weg existiert.

5.7.4 Artikulation

Mit Artikulation oder auch Gelenkpunkt wird ein Knoten eines Graphen bezeichnet, der als einzige Verbindung von Teilgraphen vorhanden ist (Rao & Ramachandra Rao, 1972). Durch Entfernung dieses Knotens wird ein zuvor zusammenhängender Graph unzusammenhängend (siehe Abschnitt 5.7.3).

5.7.5 Rang

In einem gerichteten Graphen ergibt sich eine Reihenfolge der einzelnen Knoten durch deren Abhängigkeit zueinander. Der Rang eines Knotens ergibt sich aus seiner Position im Pfad zwischen dem Start- und Endknoten (Zimmermann, 2009).

5.8 Abhängigkeitsgraphen

Graphen eignen sich sehr gut, um die technologischen Abhängigkeiten in Bauabläufen darzustellen. Im Falle des Prozessmanagements entspricht jeder Prozess einem Knoten. Vereinfacht wird hier nun angenommen, dass ein Bauteil und dessen technologische Abhängigkeit als Prozess mit zugehöriger Abhängigkeit aufgefasst wird, wobei der Prozess den Knoten darstellt und die Vorbedingung durch die gerichteten Kanten repräsentiert wird.

Es ist stets eine eindeutige Abhängigkeit zwischen Bauteilen vorhanden. Eine Kante stellt dabei eine Reihenfolgebeziehung dar, der Prozess des Ausgangsknoten muss also abgeschlossen sein um die Arbeit an dem Folgeprozess beginnen zu können. Da hier keine gegenseitige Abhängigkeit vorhanden ist, werden die Kanten gerichtet (Kapitel 5.7.1) dargestellt.

5.8.1 Checkpoint Components

Als Checkpoint Components werden Prozesse definiert, deren repräsentative Knoten im Abhängigkeitsgraph einen Gelenkpunkt beziehungsweise eine Artikulation darstellen. Diese Knoten stellen bei der Bauteilerkennung eine wichtige Rolle. Sämtliche im Rang vor diesem Knoten liegenden Prozesse müssen bereits fertiggestellt sein, um mit diesem zu beginnen. Folglich ist umgekehrt davon auszugehen, dass sämtliche Bauteile, die im Rang vor diesem Prozess liegen, bereits fertiggestellt wurden, wenn das entsprechende Bauteil erkannt wurde.

5.8.2 Prozessabhängigkeiten

Mit Graphen können technologische Abhängigkeiten zweifelsfrei dargestellt werden, da stets eine eindeutige Abhängigkeit zu einem anderen Prozess gegeben ist. Zeitlich und finanziell bedingte Abhängigkeiten (siehe Abschnitt 5.6) lassen sich mit dieser Theorie nicht vereinbaren und müssen deswegen gesondert als Randbedingungen abgespeichert werden.

Die Abhängigkeit der einzelnen Prozesse untereinander ist immer eindeutig. Es besteht immer eine chronologische Abfolge der Abhängigkeiten. Für die in der Graphentheorie verwendeten Kanten können in diesem Fall also auch gerichtete Kanten verwendet werden, die definieren,

welcher Prozess von einem anderen abhängig ist. Dabei symbolisiert ein Pfeil die Richtung der Abhängigkeit.

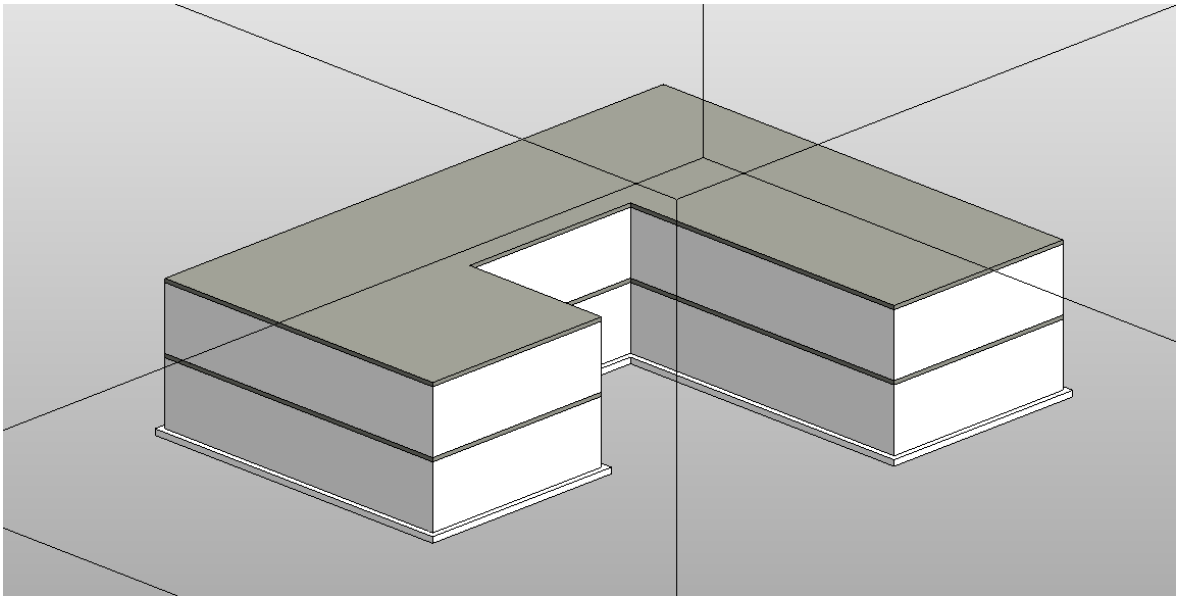

Abbildung 5.2: Zweistöckiges Gebäude mit Fundament (mit Autodesk Revit erstellt)

Anhand des Gebäudes in Abbildung 5.2 soll die technologische Abhängigkeit und der daraus resultierende gerichtete Graph verdeutlicht werden.

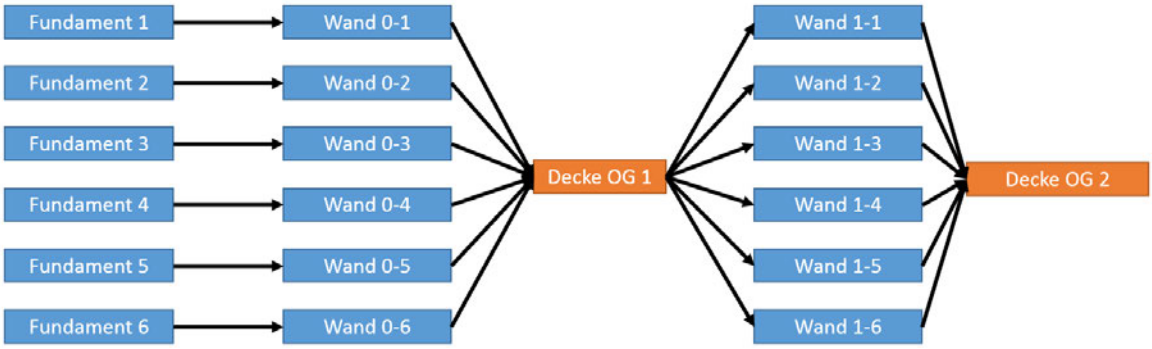

Abbildung 5.3: Gerichtete Graphen zur Abbildung von technologischen Abhängigkeiten

In Abbildung 5.3 ist der gerichtete Abhängigkeitsgraph für das genannte Beispiel zu sehen.

Dabei stellen die einzelnen Knoten jeweils den Prozess zur Fertigung eines bestimmten Bauteils dar. In diesem Beispiel wird von links mit Fundamenten begonnen. Davon technologisch abhängig ist jeweils die darauf erstellte Wand. Auf den Wänden soll eine Decke gebaut werden, welche von sämtlichen darunterliegenden Wänden logisch abhängig ist. Der den Prozess repräsentierende Knoten ist also eine Artikulation. Dies trifft ebenso auf die folgenden Deckenplatten zu.

Man kann gut erkennen, dass durch diesen Abhängigkeitsgraphen keine Baureihenfolge für die einzelnen Wände festgelegt wird und auch sonstige ressourcenbedingten Abhängigkeiten nicht berücksichtigt werden. Dies grenzt den gerichteten Abhängigkeitsgraphen und seine Darstellungsmöglichkeiten deutlich vom Gantt-Diagramm ab, welches den zeitlichen Ablauf des Bauprozesses darstellen kann, jedoch nicht die technologischen Abhängigkeiten.

Zum Aufbau des Abhängigkeitsgraphen können Task-Informationen herangezogen werden, die in der IFC-Datei hinterlegt sind (siehe Abschnitt 3.3.3: Objekte). Dort werden in *IfcTask* die technologischen Abhängigkeiten in *IsSuccessorFrom* und *IsPredecessorTo* gespeichert. Sämtliche sonstigen Abhängigkeiten werden einzig und allein über den Start- und Endzeitpunkt (*ScheduleStart* beziehungsweise *ScheduleFinish*) abgebildet und gespeichert.

Die Artikulation an den Prozessknoten der Deckenbauteile ist in dem Abhängigkeitsgraphen gut sichtbar. Solche Knoten sind von Vorteil, da sie als gute Referenzpunkte dienen (siehe Checkpoint Components, Abschnitt 5.8.1). Wenn zum Beispiel in der Punktwolke die Decke über dem OG1 erkannt worden ist, kann aufgrund des Lösungsansatzes davon ausgegangen werden, dass sämtliche Bauteile unterhalb der Decke schon vorhanden sein müssen.

Dies ist für die Verarbeitung der Punktwolke von großem Vorteil, da nie sichergestellt werden kann, dass sämtliche Bauteile korrekt erfasst werden können. Somit kann der Abhängigkeitsgraph die Bauteilerkennung aktiv unterstützen.

5.8.3 Wege und Pfade

Um die Abhängigkeiten der Bauteile darstellen zu können, muss also ein Pfad von dem jeweils gewählten Bauteil bis zum ersten Bauteil des Prozesses vorhanden sein. Ansonsten wäre das Bauteil unabhängig von der sonstigen Prozesskette oder die Prozesskette an sich weist Fehler auf. Einen Sonderfall stellen hier die ersten Bauteile im Prozessablauf dar, da diese noch von keinem anderen Bauteil abhängig sind.

5.9 Terminplanung

Die Terminplanung nimmt einen wichtigen Stellenwert dieser Arbeit ein. Besonders die Berücksichtigung von sämtlichen Abhängigkeiten (Abschnitt 5.4) spielt in die Terminplanung hinein.

Die Gantt-Diagramme sind für die Terminplanung ein wichtiges Arbeitswerkzeug um den zeitlichen Ablauf der Prozesse besser begreifen zu können. Nachdem hier die zeitlichen Abhängigkeiten visualisiert werden, kann schnell erkannt werden, welche Prozesse zu welchem Zeitpunkt ausgeführt werden sollen. In Kapitel 6 wird dies an einem Beispiel zusätzlich verdeutlicht.

Für die exakte Terminplanung im Rahmen des Building Information Modeling wird ein 4D-Modell verwendet. Dies entspricht einem 3D-Modell eines Bauwerkes, welches neben den Bauteilinformationen auch detaillierte Prozessdaten enthält.

(Tulke, 2010) hat untersucht, wie die einzelnen Prozesse am sinnvollsten und effektivsten mit einem dreidimensionalen Modell gekoppelt werden können. In Kapitel 4 wurde dabei bereits darauf eingegangen, dass dieser Vorgang nur teilweise automatisiert werden kann. Da die Verknüpfung von Prozessdaten mit dem Building Information Model jedoch die Grundvoraussetzung für die 4D-Terminplanung darstellt, ist es wünschenswert, diesen Prozess weitestgehend zu vereinfachen.

Grundsätzlich muss zur Verknüpfung jedes Bauteil einzeln ausgewählt werden und dann ein korrespondierender Knoten erstellt werden. Da jedoch viele Bauteile ähnliche Eigenschaften, wie zum Beispiel Material, Abmessungen oder Stockwerk haben, kann man, basierend auf diesen Daten während der Eingabe, Vorschläge für ähnliche Prozesse anzeigen. Dadurch lässt sich dieser Vorgang deutlich beschleunigen.

Für die Eingabe von terminplantechnischen Abhängigkeiten ist jedoch in jedem Fall eine manuelle Einarbeitung erforderlich. Bei dieser Aufgabe unterstützen bereits mehrere Programme (siehe Abschnitt 4.2). Wichtig ist in diesem Zusammenhang die neuen Möglichkeiten der IFC4 zu nutzen, um diese Prozessdaten auch abspeichern und weiterverarbeiten zu können.

5.10 Zusammenfassung

Die Prozessplanung und deren Theorie bildet die methodische Grundlage dieser Thesis.

Aufgrund der Vielzahl an unterschiedlichen Abhängigkeiten ist es schwierig, diese in ihrer Gesamtheit sinnvoll abzubilden. Die technologischen Abhängigkeiten lassen sich mit Hilfe der Graphentheorie gut darstellen, jedoch beinhalten diese keine zeitlichen Informationen. Die kapazitiven Anordnungsbeziehungen sind durch Gantt-Diagramme gut abbildbar. Diese können als Abbildung des tatsächlichen Bauablaufs interpretiert werden. Für eine umfassende und sinnvolle Analyse von Alternativen im Bauablauf sind beide Repräsentationen erforderlich.

Das nächste Kapitel beschreibt den Soll-Ist-Abgleich basierend auf den hier erarbeiteten Grundlagen der Bauprozessmodellierung.

Kapitel 6

Soll-Ist-Abgleich

Der Soll-Ist-Abgleich steht in engem Zusammenhang mit den Abhängigkeitsgraphen (Abschnitt 5.8). Basierend darauf können Aussagen über das Vorhandensein einzelner Bauteile getroffen werden. Aufgrund der genannten Rahmenbedingungen kann es im Bauablauf jederzeit zu Abweichungen im Ablaufplan kommen. Dabei muss es sich nicht zwingend um gravierende Änderungen wie zum Beispiel zeitliche Verzögerungen durch terminliche Schwierigkeiten handeln.

Es kann sich auch um Alternativen handeln, welche den restlichen Bauablauf und dessen Zeitplan nicht beeinflussen. Diese werden in manchen Fällen aufgrund von mangelnder Ressourcenverfügbarkeit oder allgemein aufgrund von spontanen Ablaufoptimierungen auf der Baustelle angewandt.

6.1 Änderungen an Prozessabläufen

Das nachfolgende Beispiel soll anhand von Abbildung 6.1 verdeutlichen, was genau unter Alternativen zu verstehen ist und wie man Alternativen erfassen kann.

Betrachtet werden soll in diesem Fall der Prozess „Bauen einer Stütze“ unter der Randbedingung, dass aufgrund von Ressourcenbeschränkungen maximal drei Stützen parallel gefertigt werden können. Die Wände werden in diesem einfachen Beispiel vernachlässigt, da sie keinen Einfluss auf die Grundidee haben. Klar ist hier, dass somit die insgesamt sechs Stützen in mindestens zwei Arbeitsgängen gefertigt werden müssen. Aufgrund weiterer Engpässe, wie zum Beispiel begrenzt verfügbares Personal, könnten im schlechtesten Fall bis zu sechs Arbeitsgänge notwendig sein.

Als unverrückbare Randbedingungen bestehen strenge Reihenfolgebeziehungen zu anderen Bauteilen (siehe Abschnitt 5.8).

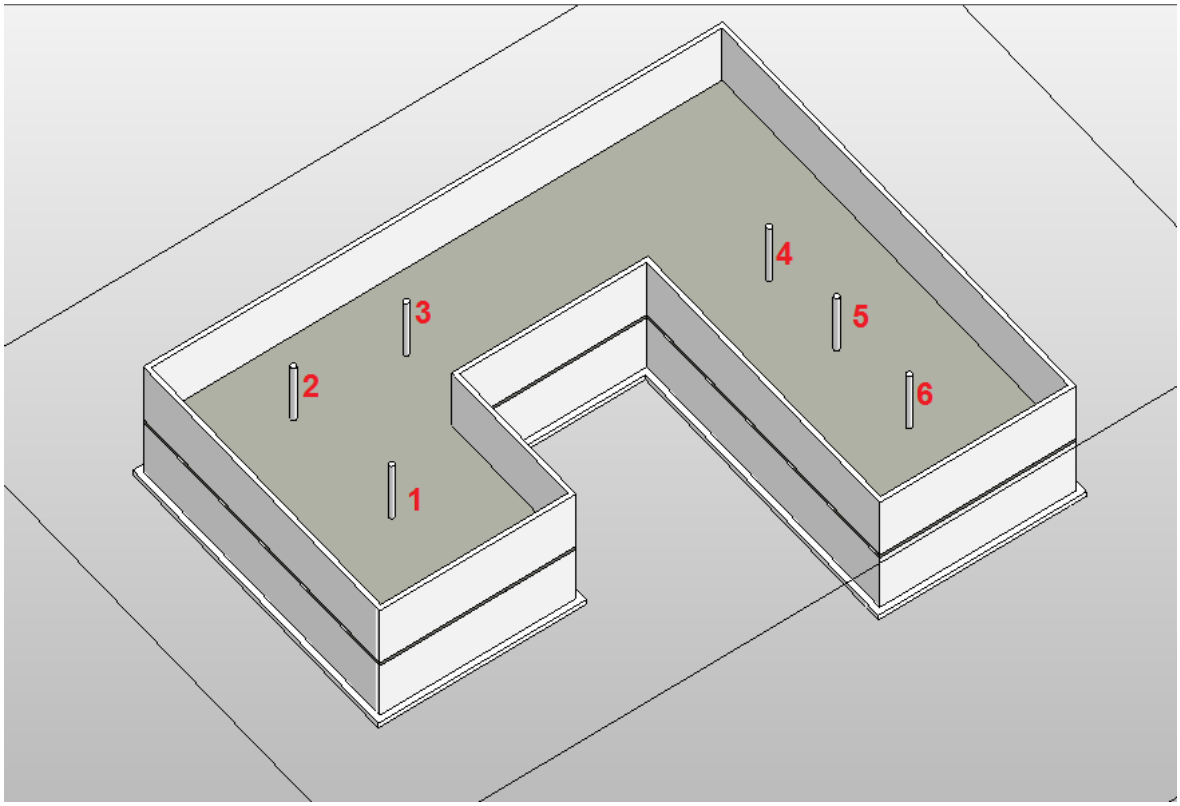

Abbildung 6.1: Bauablauf - Beispielprojekt in Revit modelliert

So kann der Prozess „Bauen einer Stütze“ nicht beginnen, bevor nicht die Bodenplatte gefertigt wurde. Des Weiteren kann die Decke des Erdgeschosses nicht gebaut werden, bevor nicht sämtliche Stützen darunter fertiggestellt worden sind.

Grundsätzlich ist dabei für den weiteren Bauverlauf irrelevant, welche der Stützen zu welchem Zeitpunkt gebaut werden. Das nun folgende Beispiel zeigt, dass sie im Abhängigkeitsgraphen den gleichen Rang (siehe Abschnitt 5.7.5) haben, also untereinander nicht voneinander abhängig sind, und dementsprechend in beliebiger Reihenfolge gebaut werden könnten. Besonders deutlich wird dies, wenn man sich den Abhängigkeitsgraph dieses Beispiels ansieht (Abbildung 6.2).

Abbildung 6.3 zeigt im ersten Diagramm eine mögliche Lösung für den Bauablauf der Stützen in Form eines Gantt-Diagramms. In diesem Fall soll dies den vorgegebenen Bauablauf der Prozessplaner darstellen. Aufgrund von fiktiven anfallenden Arbeiten im Bereich der Stützen 7-9 werden nun außerplanmäßig zuerst die Stützen 10-12 gebaut. Das Gantt-Diagramm ändert sich, wie in Abbildung 6.3 im unteren Teil zu sehen. Dabei ändert sich wie zu erwarten die Gesamtbauzeit des Bauwerks nicht. Es kommt also zu keinen Verzögerungen im Gesamtbauablauf.

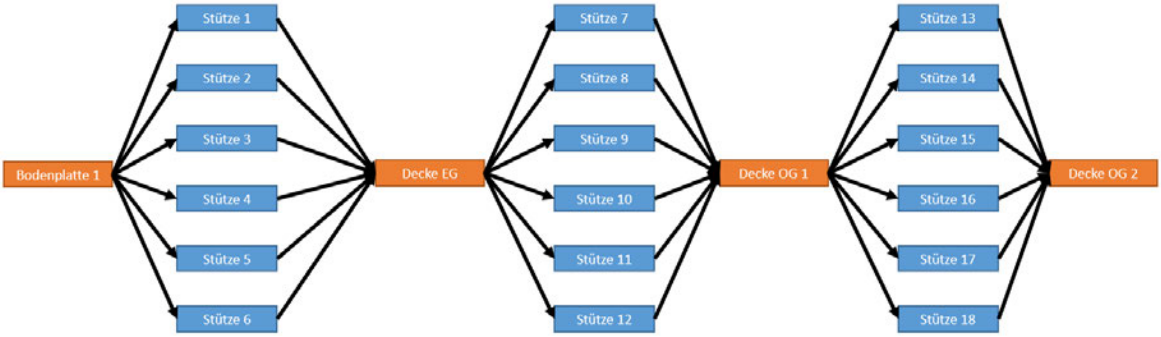

Abbildung 6.2: Bauablauf des Beispiels als Abhängigkeitsgraph

6.2 Varianten

Wichtig wird dies erst bei der Betrachtung der Gantt-Diagramme und unter Einbeziehung des Kosten- und Zeitfaktors. Hier zeigt sich, dass sich besonders bei ressourcenbedingtem Aufsplitten von Prozessen, die ohne dieses Limit parallel laufen könnten, Varianten finden (siehe Abbildung 6.3).

Die Erkennung solcher Varianten funktioniert jedoch nur bei exakter Erkennung beziehungsweise definitiver Feststellung des Nicht-Vorhandenseins sämtlicher in den Varianten enthaltenen Bauteile. Nachdem hier keine technologische Abhängigkeit besteht, kann auch keine softwaregestützte Bestimmung der Bauteile stattfinden. Wenn dies jedoch gegeben ist, kann die Variante bei Eintreten sofort erkannt werden.

Als Folge muss das Gantt-Diagramm beziehungsweise der betroffene Teilbereich neu berechnet werden. Dazu werden die Start- und Endzeiten oder die Dauer der betroffenen Prozesse ausgetauscht. Sollte bereits eine Projektverzögerung oder -verkürzung eingetreten sein, muss diese bei den terminlich festgelegten Prozessen berücksichtigt werden.

Bei der automatisierten Fortschrittskontrolle hingegen tritt hier das Problem auf, dass ein Algorithmus die kurzfristig entstandenen Randbedingungen nicht kennt. Wenn anhand von erfassten Punktwolken erkannt wird, dass die Stützen 10-12 existieren, darf also nicht automatisch davon ausgegangen werden, dass die Stützen 7-9 bereits erstellt wurden. Allgemein kann also für Prozessknoten, bei denen es sich nicht um Checkpoint Components handelt, keine exakte Aussage über deren erfolgreiche Fertigstellung getroffen werden, wenn diese im Rahmen einer Prozessüberwachung nicht eindeutig erkannt werden.

Bei der Analyse der Abhängigkeiten mit Hilfe der Abhängigkeitsgraphen wird deutlich, dass in dieser Darstellung nur die technologisch bedingten Abhängigkeiten dargestellt werden können. Der genaue zeitliche Ablauf ist im Abhängigkeitsgraph nicht hinterlegt.

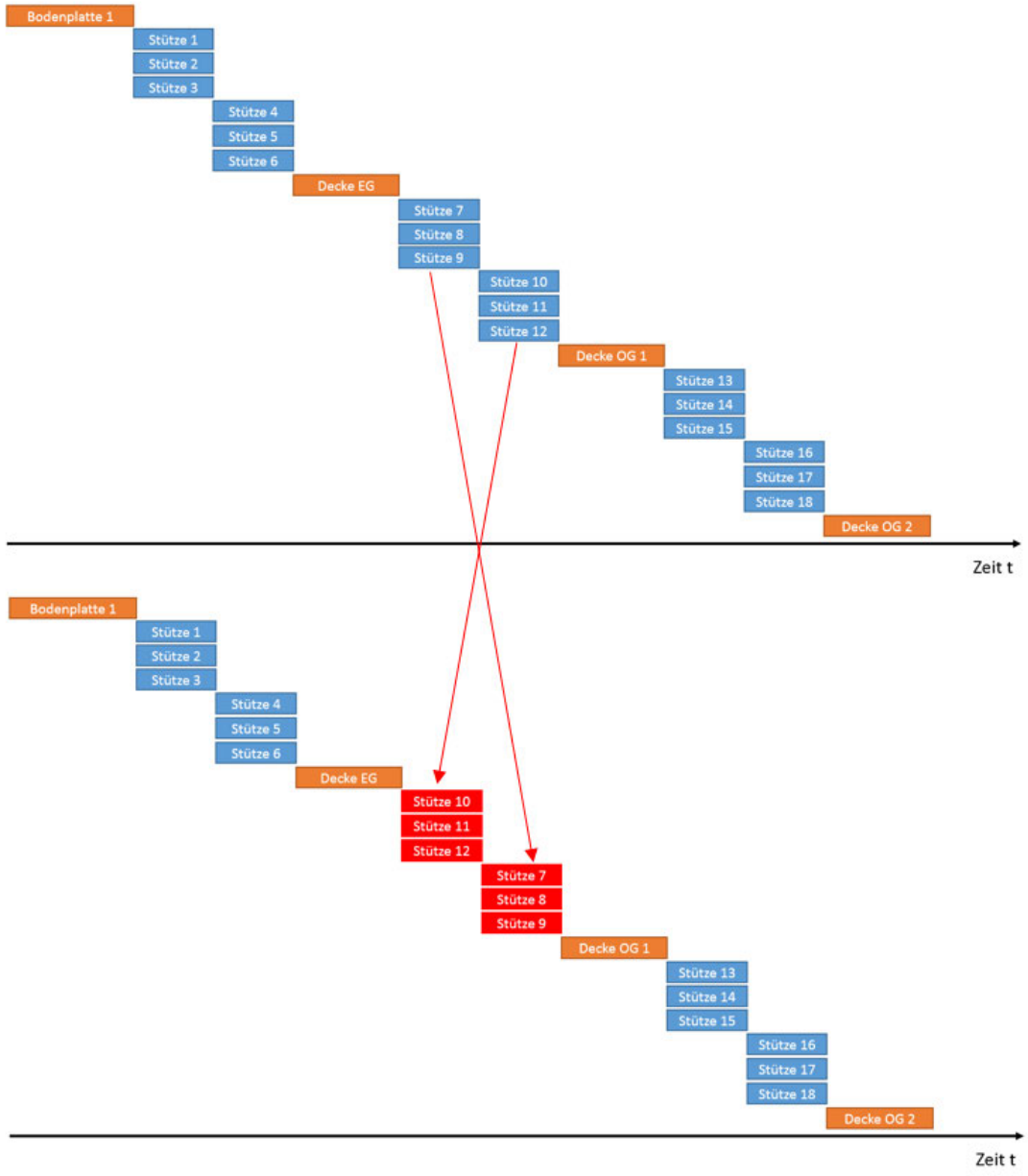

Abbildung 6.3: Bauablauf als Gantt-Diagramm - Vor und nach der Änderung

6.3 Suchalgorithmen

Um bei der Erkennung von Bauteilen die korrekten Rückschlüsse auf den Baufortschritt ziehen zu können, ist es notwendig, sämtliche Bauteile zu kennen, die zu einem bestimmten Zeitpunkt bereits vorhanden sein müssen. Nachdem der Graph der technologisch bedingten Abhängigkeiten gerichtet ist, genügt es, sämtliche Bauteile in die Liste der bereits gebauten Bauteile aufzunehmen, von denen das erkannte Bauteil abhängig ist.

Da ein Bauteil von mehreren anderen Bauteilen abhängig sein kann und der Abhängigkeitsgraph sich auch in einem früheren Stadium der Prozesskette aufteilen kann, muss ein passender Suchalgorithmus implementiert werden, der sämtliche Bauteile ermittelt, von denen das jeweils erkannte Bauteil abhängig ist.

Die beiden bekanntesten und für ungewichtete, gerichtete Graphen am weitesten verbreiteten Suchalgorithmen sind die Tiefen- und die Breitensuche (Depth-First Search und Breadth-First Search).

Ziel beider Algorithmen ist es, sämtliche Knoten auf dem Pfad zum Startknoten zu erfassen.

Bei der Tiefensuche wird dabei vom Startknoten aus nach und nach in die Tiefe gesucht um so am Ende sämtliche Knoten zu finden (siehe Abbildung 6.4).

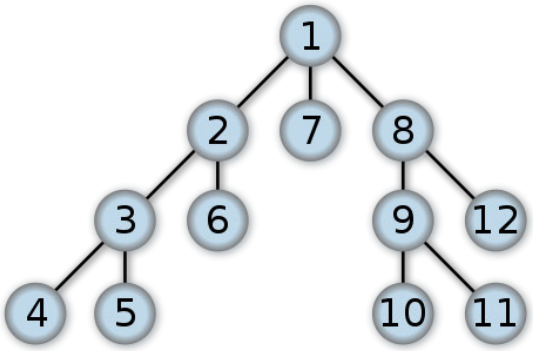

Abbildung 6.4: Depth-First: Reihenfolge, in der die Knoten gefunden werden (wikipedia.org).

Die Breitensuche basiert auf dem selben Prinzip, nur dass hier zuerst alle Knoten gesammelt werden, die mit dem Startknoten verbunden sind. Dies wird für jeden der gefundenen Knoten analog fortgesetzt (siehe Abbildung 6.5).

Die beiden Algorithmen sind für gerichtete Graphen gleich teuer, das bedeutet, sie benötigen die gleiche Anzahl an Operationen um an ihr Ziel zu gelangen. Bei dem Soll-Ist-Abgleich sind jedoch vor allem die im Rang unmittelbar vor dem detektierten Bauteil ablaufenden Prozesse von Bedeutung, weswegen eine Breitensuche hier vorzuziehen ist.

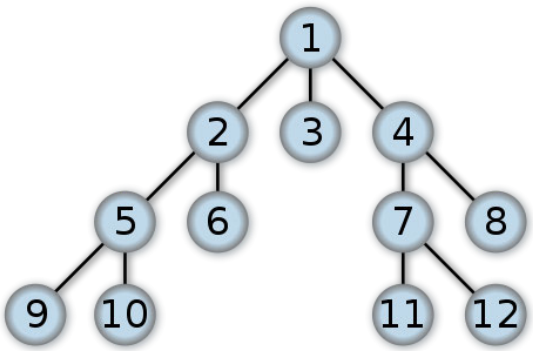

Abbildung 6.5: Breadth-First: Reihenfolge, in der die Knoten gefunden werden (wikipedia.org).

6.4 Soll-Ist-Abgleich

Eine weitere Möglichkeit der Prozessabweichung zwischen geplantem und tatsächlich erfolgtem Bauablauf soll mit Abbildung 6.6 gezeigt werden.

Hier ist es aufgrund einer Ressourcenverknappung zwischen der *Decke OG 2* und *Decke OG 3* nur möglich, maximal zwei Stützen parallel zu bauen. Dies hat nun nicht nur Auswirkungen auf den generellen Ablauf sondern auch auf die Gesamtdauer des Bauprozesses.

Anhand der Punktwolke wird nun beispielhaft Stütze 16 als bereits gebaut erkannt. Aufgrund dessen kann nun basierend auf dem Abhängigkeitsgraphen (Abbildung 6.2) davon ausgegangen werden, dass die Decke des zweiten Obergeschosses und sämtliche darunter liegenden Bauteile vorhanden sein müssen. Hervorzuheben ist hier, dass die Stützen 13 - 15 eigens mit Hilfe der Punktwolke erkannt werden müssen und dadurch ihre Existenz nachgewiesen werden muss. Gleichwohl sich diese im Gantt-Diagramm zeitlich vor Stütze 16 befinden, stehen diese nicht in logischer Abhängigkeit zueinander. Das Gantt-Diagramm kann hier also keine eindeutige Auskunft geben. Die Erkennung der betroffenen Stützen soll in diesem Fall jedoch erfolgt sein.

Nachdem es sich in diesem Fall bei Stütze 16 nicht um eine *Checkpoint Component* handelt, kann vorerst auch noch keine eindeutige Aussage über etwaige Verzögerungen für das Gesamtprojekt gemacht werden. Erst zum Zeitpunkt des verzögerten Prozessbeginns zum Bau der Decke des dritten Obergeschosses kann automatisiert festgestellt werden, dass sich der Bauprozess verzögert. Da der Prozess zu seinem eigentlichen Startpunkt noch nicht begonnen wurde, verschiebt sich nun der gesamte weitere Ablauf nach hinten.

Die korrekte Darstellung dieser zeitlichen Verschiebung muss hier logischerweise im zu Grunde liegenden, ursprünglichen Gantt-Diagramm erfolgen. Der Soll-Ist-Vergleich ist im zweiten Diagramm in Abbildung 6.6 zu sehen. Die bereits fertiggestellten Bauteile sind grau hinterlegt, noch zu bauende Bauteile in ihrer ursprünglichen Farbe.

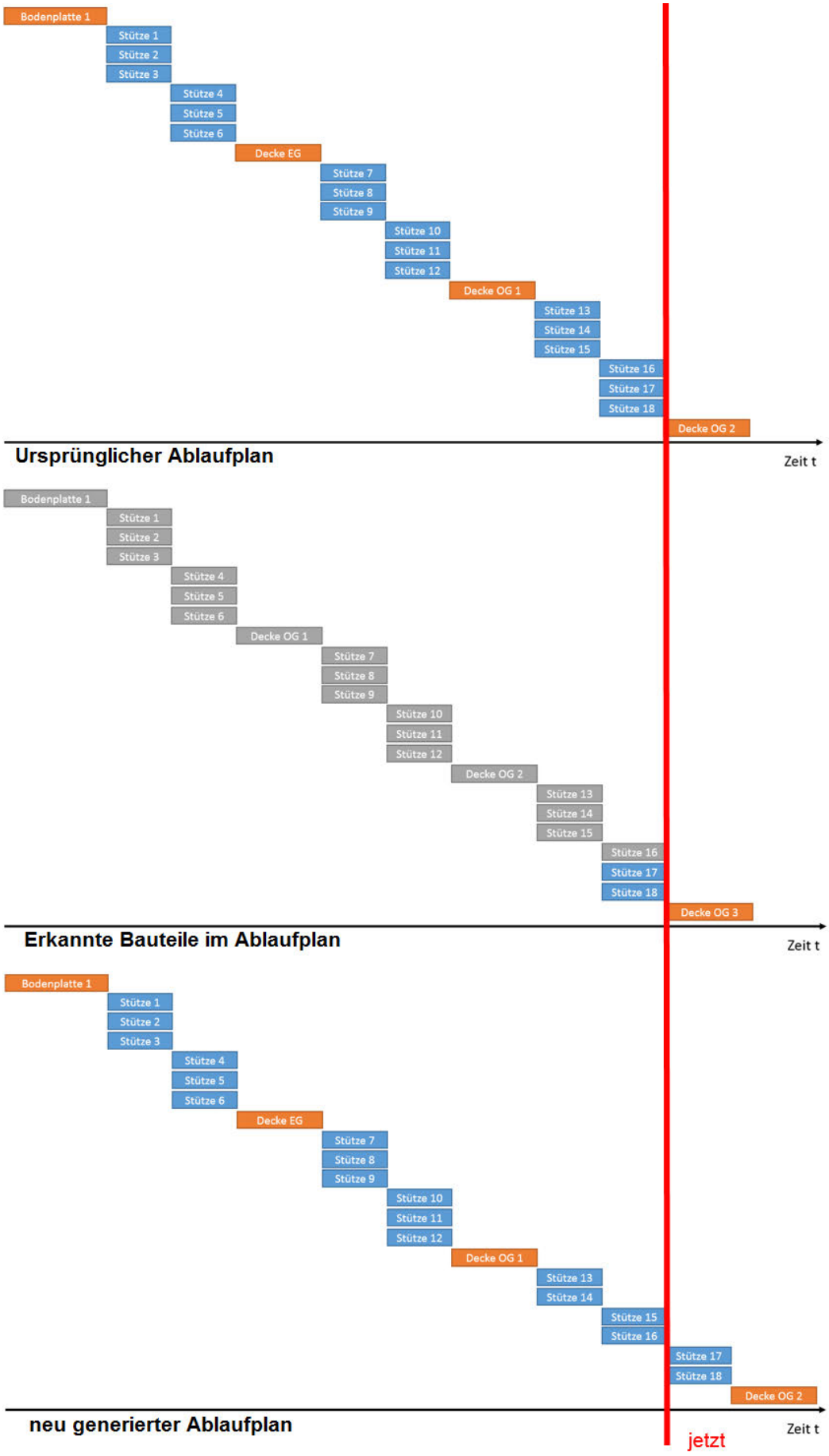

Abbildung 6.6: geänderter Bauablauf durch Resourcenverknappung als Gantt-Diagramm

Das Gantt-Diagramm muss basierend auf dieser Erkenntnis neu generiert werden. Das dritte Diagramm in Abbildung 6.6 zeigt abschließend diesen neu generierten Ablaufplan. Zu Erkennen ist hier, dass zwischen dem zweiten und dritten Obergeschoss nun, wie eingangs erwähnt, nur noch zwei Stützen gleichzeitig gebaut werden können. Dadurch verschiebt sich für den Bauprozess der Stützen 17 und 18 der Start- und Endzeitpunkt entsprechend. Dank des Gant-Diagramms ist auch deutlich die zusätzlich benötigte Zeit zur Fertigstellung des Gebäudes zu erkennen.

In Kapitel 8.7 wird diese Thematik aufgegriffen und dargelegt, wie die Implementierung in dem Softwaretool, das zum Nachweis der Tragfähigkeit der entwickelten Konzepte entstanden ist, umgesetzt wurde.

6.5 Punktwolken-basierte Detektion

Zu einem späteren Zeitpunkt soll die Ermittlung des Bauzustandes durch Punktwolken, welche mit photogrammetrischen Methoden erzeugt wurden, erfolgen. Wie bereits erläutert (Abschnitt 2.1), stellen Verschattungen durch andere Bauteile oder Behelfsgeräte ein Problem dar, da dadurch Bauteile teilweise nicht mehr erkannt werden können.

Abbildung 6.1 zeigt ein sich im Bau befindliches Gebäude. Eine aus dieser Ansichtsposition erstellte Punktwolke würde einen Großteil der Wände des Erdgeschosses sowie deren Fundamente nicht anzeigen, da sie durch die Decke über dem Erdgeschoss verschattet sind. Durch den Abhängigkeitsgraph (Abbildung 6.2) wird jedoch deutlich, dass die Elemente, wenngleich sie nicht erkannt werden, vorhanden sein müssen.

Hier helfen die *Checkpoint Components* trotzdem eine möglichst exakte Aussage über vorhandene Bauteile zu treffen (Abschnitt 5.8.1).

6.6 Zusammenfassung

Der Soll-Ist-Abgleich in Zusammenhang mit der Punktwolkeninterpretation hängt stark von der Qualität der Punktwolke und den darin erkannten Bauteilen ab. Die Abhängigkeitsgraphen stellen dabei eine zusätzliche Validierungsmöglichkeit dar, um das Vorhandensein von verschatteten, oder fehlerhaft nicht erkannten Objekten zu bestätigen. Die Checkpoint Components ünterstützen die Bauteilerkennung zusätzlich.

Kapitel 7

Technologien

Gerade im IT-Bereich ist bei der Wahl von Programmiersprachen und externen Tools darauf zu achten, dass diese sich auch wirklich für ein geplantes Projekt eignen.

Dazu müssen die Ziele des Vorhabens genau untersucht werden und die Anforderungen klar definiert sein. Analog müssen die Funktionen der verwendeten Bibliotheken auf ihre Tauglichkeit überprüft werden.

7.1 Programmiersprache

Für die Entwicklung des „4D-BIM-Viewers mit graphbezogener Darstellung von Bauabläufen und -alternativen“ standen zu Beginn drei Programmiersprachen zur Auswahl: C++, C# sowie Java.

Diese sogenannten höheren Programmiersprachen sind aufgrund ihrer großen Verbreitung im Wissenschaftsbereich und auch im OpenSource-Entwicklerkreis gut geeignet. Deshalb finden sich bei komplexen Problemstellungen viele Supportforen in denen Unterstützung gefunden werden kann. Des Weiteren stehen für diese Programmiersprachen Bibliotheken mit großem Funktionsumfang zur Verfügung.

Durch die der menschlichen Sprache ähnlichen Syntax sind diese Programmiersprachen gut lesbar.

Aufgrund der in diesem Kapitel zusammengefassten Technologien wird für das geplante Softwaretool die Programmiersprache C# verwendet. Dies wird in Abschnitt 7.5 begründet.

7.1.1 Bibliotheken

Mit dem Begriff Bibliothek bezeichnet man eine Sammlung von Funktionen und Klassen, welche thematisch zusammengehörend sind. Viele Bibliotheken stehen offen (Open Source) zur Verfügung, das bedeutet, dass deren Quelltext frei verfügbar ist.

In der Regel sind diese auch so lizenziert, dass sie von jedem Entwickler geändert beziehungsweise angepasst werden können.

7.2 IFC Bibliotheken

Dieser Abschnitt ist in Teilen in Zusammenarbeit mit Julian Amann und Simon Daum im Rahmen der Vorbereitung einer Vortragsreihe entstanden.

Das IFC-Datenformat ist äußerst komplex und umfangreich. Obwohl sämtliche Definitionen und Entwürfe frei verfügbar sind, ist es mit viel Entwicklungsarbeit verbunden, eine IFC-Datei einzulesen und zu verarbeiten.

Eine gewünschte Visualisierung der Bauteile erschwert diese Arbeit zusätzlich.

Da die Nachfrage nach Tools, die diese Aufgaben übernehmen, sehr groß ist, sind bereits mehrere Bibliotheken für unterschiedliche Programmiersprachen entstanden. Ein Großteil wird von der Open Source-Community getragen. Das bedeutet, dass der Quellcode frei verfügbar ist und somit auch Anpassungen für eine private Projektumgebung ohne Probleme umsetzbar sind.

Tabelle 7.1 gibt einen groben Überblick über einige bekannte Bibliotheken, erhebt aber keinen Anspruch auf Vollständigkeit. Dies ist bei der sich schnell entwickelnden BIM-Branche auch nur schwer möglich.

Bibliothek	Sprache	Webseite
IFC Engine	C++	www.rdf.bg/ifcenginedll/product_ifcdll.html
xBIM	C#	http://xbim.codeplex.com
IFC Tools Project	Java	www.ifctoolsproject.com
IFC OpenShell	C++	www.ifcopenshell.org
IfcPlusPlus	C++	www.ifcplusplus.com
STEPcode	C++	www.stepcode.org
JSDAI	Java	www.jsdai.net
Revit IFC Exporter	C#	www.sourceforge.net/p/ifcexporter/

Tabelle 7.1: Verfügbare IFC Bibliotheken

Um eine gewisse Aktualität der Datenerhebung zu gewährleisten, beschränkt sich die Auflistung auf Projekte, die in letzter Zeit Aktivität durch Codeänderungen oder neue Ankündigungen gezeigt haben.

Bei einer für dieses Forschungsprojekt so wichtigen Funktion ist es von großer Bedeutung, dass hier eine stabile und gut gepflegte Bibliothek gewählt wird, die auch das neue IFC4-Format unterstützen wird.

In den folgenden Tabellen werden die wichtigsten Eigenschaften der gewählten Bibliotheken zusammengefasst.

Neben der genutzten Programmiersprache werden dort folgende Eigenschaften untersucht:

Lizenzierung

Jede Bibliothek und auch allgemein jede Software wird von deren Entwickler mit bestimmten Vorgaben veröffentlicht. Diese Vorgaben werden in der Softwarebranche in Lizenzen zusammengefasst. In diesen wird geregelt, welche Tätigkeiten dem Nutzer erlaubt, beziehungsweise verboten sind. Im Normalfall wird hier dokumentiert, ob die Software kopiert und verändert werden darf. Zusätzlich wird geregelt, in welchem Umfang sie zu kommerziellen Zwecken genutzt werden kann.

Die Lizenzierung wird durch verschiedene standardisierte Lizenzverträge für die Entwickler vereinfacht. In der Open Source-Community übliche Lizenzen sind zum Beispiel *GNU General Public License* (GNU, 2011) oder *Creative Commons*. Diese erlauben eine unentgeltiche Nutzung des Codes, sowie dessen Veränderung und Verwendung in anderen Projekten.

Da in der Forschung nur in eingeschränktem Rahmen finanzielle Mittel zur Verfügung stehen, ist eine Nutzung von Open Source-Projekten zu empfehlen.

Visualisierung

Unter dem Punkt *Visualisierung* wird untersucht, ob die IFC-Bibliotheken bereits eine Möglichkeit besitzen, das importierte IFC-Modell grafisch darzustellen. Durch eine native Unterstützung der Visualisierung entsteht der Vorteil, dass die einzelnen Bauteile in der Visualisierung bereits logisch mit den restlichen Bauteilinformationen verknüpft sind. Eine solche Verbindung muss bei einer externen Visualisierungslösung erst hergestellt werden.

IFC4-Unterstützung

Wie bereits in Kapitel 3.3 erläutert, sind die in Version 4 eingeführten Neuerungen der IFC für die Bauprozessplanung sehr wichtig. Deshalb muss darauf geachtet werden, dass die verwendete Bibliothek IFC4-Dateien lesen und verarbeiten kann.

EXPRESS Schema Support

Das EXPRESS Schema beschreibt die möglichen Datensätze in der IFC-Datei und wie diese untereinander verknüpft sind.

Die meisten Bibliotheken lesen eine IFC-Datei ein und speichern die Daten in vorher genau für diesen Datentyp definierten Variablen ab. Dieses Verfahren wird *Early Binding* genannt. Durch die exakte Deklaration der Variablen kann eine Leistungssteigerung erreicht werden. Im Allgemeinen wird ein Programmcode dadurch auch besser lesbar und Fehler können leichter beseitigt werden, da Typfehler bereits beim Kompilieren und nicht erst beim Ausführen des Codes auffallen (Microsoft, 2005). Die Fehlersuche ist beim *Early Binding* somit deutlich einfacher, wodurch auch die Entwicklungszeit reduziert werden kann. Auf der anderen Seite ist *Late Binding* flexibler einsetzbar, nachdem zu Beginn noch nicht festgelegt ist, welche Funktionen und Variablen aufgerufen werden.

Unter *EXPRESS Schema Support* wird verstanden, dass die jeweilige Bibliothek die Möglichkeit hat, ein beliebiges Schema einzulesen und eine passende IFC-Datei basierend darauf zu verarbeiten. Dies hat zur Folge, dass Variablen hier nicht exakt definiert werden können, was als *Late Binding* bekannt ist. Dadurch muss mehr Speicher für Variablen freigehalten werden, die größer sind als sie eigentlich sein müssten. In einer weiteren Variante wird auf Basis des Schemas der entsprechende Programmcode generiert, was jedoch nicht als *Late Binding* zu verstehen ist.

Diese Funktionalität hat den Vorteil, dass die Entwickler der Bibliothek bei einer neuen IFC-Version nicht jedes Mal selbige aktualisieren müssen. Es reicht aus, wenn der Anwender das neue Schema einbindet. Diese Flexibilität geht jedoch zu Lasten der Leistung des Programms.

Letztes Update und Anzahl der Entwickler

Mit Hilfe dieser Kennzahlen wird versucht, die Aktivität des Projektes abzuschätzen. Die letzten Updates eines Projektes geben hierüber einen guten Überblick. Ferner ist es für Außenstehende meist schwierig, die Anzahl an Projektbeteiligten und Entwicklern zu erkennen. Dies ist aber gerade bei Open Source-Projekten ein wichtiger Faktor, um abschätzen zu können, wie groß die Unterstützer-Gemeinde des Tools ist. Je mehr Programmierer sich an einer Bibliothek beteiligen, um so unwahrscheinlicher ist eine unerwartete Einstellung des Projekts.

Bei den Einträgen „EXPRESS Schema Support“ und „IFC4-Unterstützung“ sollte zumindest ein Wert positiv sein. Die IFC4-Unterstützung ist für dieses Projekt sehr wichtig. Falls diese nicht vorhanden ist, kann jedoch bei vorhandener EXPRESS Schema-Unterstützung das IFC4-Schema selbst eingebunden werden. Dadurch ist eine Unterstützung sämtlicher IFC-Standards möglich.

7.2.1 IFC Engine

Die IFC Engine ist eine STEP-Toolbox mit Grafikunterstützung. Sie wird von der bulgarischen Firma RDF entwickelt und steht als Closed Source zur Verfügung.

Für ein wissenschaftliches Projekt ist sie deshalb leider nur begrenzt einsetzbar, da projektspezifische Modifikationen an der Engine selbst nicht vorgenommen werden können. Der Support durch die Entwickler ist sehr gut, es werden auch Hinweise und Vorschläge zur Weiterentwicklung angenommen.

Webseite	www.rdf.bg/ifcenginedll/product_ifcdll.html
Programmiersprache	C++
Lizenzierung	Kern Closed Source
Visualisierung	Ja
EXPRESS Schema Support	Nein
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	1

Tabelle 7.2: Eigenschaften IFC Engine

7.2.2 xBIM

xBIM wird von einer großen Gruppe an Entwicklern betreut und unter der CDDL-Lizenz auf Codeplex veröffentlicht. Der Code steht OpenSource zur Verfügung und kann beliebig modifiziert werden. Bei Veröffentlichung eines auf xBIM basierenden Projekts muss dieses gemäß Lizenzierung ebenfalls als OpenSource-Lösung angeboten werden.

Die Entwicklung geht zum jetzigen Zeitpunkt zügig voran, es werden in kurzem Rhythmus neue Entwicklerversionen veröffentlicht. Neben der Nutzung als klassische Software steht auch eine Web-Version von xBIM zur Verfügung, mit der eine Visualisierung von IFC-Dateien in einem Webbrowser möglich ist. Zur Grafikvisualisierung auf dem Desktop wird OpenCascade verwendet (OpenCascade, 2013).

Aufgrund der Vielzahl an Möglichkeiten und der großen Entwickler-Community ist diese Library sehr interessant.

Webseite	http://xbim.codeplex.com
Programmiersprache	C#
Lizenzierung	Common Development and Distribution License (CDDL)
Visualisierung	Ja
EXPRESS Schema Support	Nein
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	mind. 21

Tabelle 7.3: Eigenschaften xBIM

7.2.3 IFC Tools Project

Das IFC Tools Projekt ist aus den *Open IFC Tools* hervorgegangen. Diese wurden an der Bauhaus Universität Weimar entwickelt und werden nun als Fork eigenständig weiterentwickelt. Das Projekt wurde mit dem ersten Preis im Forschungswettbewerb „Auf IT gebaut“ ausgezeichnet und ist eine der umfangreichsten IFC-Bibliotheken auf Basis von Java.

Webseite	www.ifctoolsproject.com
Programmiersprache	Java, C#-Unterstützung
Lizenzierung	CreativeCommons
Visualisierung	Ja
EXPRESS Schema Support	Nein
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	mind. 5

Tabelle 7.4: Eigenschaften IFC Tools Project

7.2.4 IFC Open Shell

IFC Open Shell ist eine kleinere IFC-Bibliothek. Zur Visualisierung wird auch hier auf OpenCascade zurückgegriffen.

Das Entwicklerteam scheint relativ klein zu sein, Updates erfolgen ca. im Jahresrhythmus. Dadurch fällt dieses Projekt tendenziell nicht in die engere Auswahl, da eine Unterstützung von IFC4 in naher Zukunft nicht realistisch scheint.

Webseite	www.ifcopenshell.org
Programmiersprache	C++ / Java
Lizenzierung	OpenSceneGraph Public License (OSGPL)
Visualisierung	Ja
EXPRESS Schema Support	Nein
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	unbekannt

Tabelle 7.5: Eigenschaften IFC Open Shell

7.2.5 IfcPlusPlus

Das bei Google Code gehostete IfcPlusPlus war bis vor kurzem noch unter dem Namen IfcGears veröffentlicht. IfcPlusPlus basiert auf C++ und darf auch für kommerzielle Projekte verwendet werden, ohne dass diese als OpenSource zur Verfügung gestellt werden müssen. Aufgrund des EXPRESS Schema Supports können mit IfcPlusPlus bereits Erfahrungen mit IFC4 gesammelt werden.

Webseite	www.ifcplusplus.com
Programmiersprache	C++
Lizenzierung	OSGPL
Visualisierung	Ja
EXPRESS Schema Support	Ja
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	mind. 5

Tabelle 7.6: Eigenschaften IfcPlusPlus

7.2.6 STEPcode

STEPcode ist ein unter der BSD-Lizenz veröffentlichtes Projekt. Es beschränkt sich auf die Verarbeitung des EXPRESS Schemas und bietet keine eigene Visualisierungsmöglichkeit. Mit STEPcode können STEP Part21-Dateien sowohl gelesen als auch geschrieben werden.

Webseite	www.github.com/stepcode
Programmiersprache	C++
Lizenzierung	Berkeley Software Distribution (BSD)
Visualisierung	Nein
EXPRESS Schema Support	Ja
IFC4-Unterstützung	Ja
Letztes Update	2013
Anzahl Entwickler	unbekannt

Tabelle 7.7: Eigenschaften STEPcode

7.2.7 JSDAI

Auch das JSDAI-Projekt beschränkt sich auf die Verarbeitung der STEP-Dateien. Eine Visualisierung ist hiermit ebenfalls nicht möglich. Es unterstützt sowohl Early- als auch Late-Binding und ist damit sehr vielfältig einsetzbar. Das letzte Update für dieses Projekt liegt jedoch bereits über drei Jahre zurück.

Webseite	www.jsdai.net
Programmiersprache	C++
Lizenzierung	GNU Affero General Public License (AGPL) v3
Visualisierung	Nein
EXPRESS Schema Support	Ja
IFC4-Unterstützung	Ja
Letztes Update	2010
Anzahl Entwickler	mind. 5

Tabelle 7.8: Eigenschaften JSDAI

7.2.8 Revit IFC Exporter

Der Autodesk Revit IFC Exporter ist keine IFC Bibliothek im klassischen Sinne. Er wird hier der Vollständigkeit halber jedoch trotzdem aufgeführt, da Revit eine der führenden BIM-Modellierungslösungen ist. Nachdem viele Gebäudemodelle mit Revit erzeugt werden, ist es in der Konsequenz naheliegend, direkt hier anzusetzen und das zu exportierende Modell so zu modifizieren, dass es für die weitere Nutzung verwendbar und lesbar ist. Nachdem das von Revit verwendete Dateiformat proprietär und damit von anderen Tools nicht lesbar ist, muss auch hier auf das IFC-Format zurückgegriffen werden.

Der von Autodesk veröffentlichte Exporter für Revit wurde als OpenSource veröffentlicht (Autodesk, 2013b). Mit diesem Add-On kann der Export von IFC-Dateien aus Revit heraus gesteuert und sogar modifiziert werden. Dadurch ist es zum Beispiel möglich, für jedes exportierte Bauteil zusätzlich eine Datei zu exportieren, die dessen Geometrie gesondert beschreibt. Dies ist nötig, da in den IFC verschiedene Möglichkeiten bestehen, Geometrie zu beschreiben, welche nicht ohne Umwandlung darstellbar sind (siehe Abschnitt 3.3). Für die Visualisierung müssen die Bauteile jedoch als Dreieckselemente vorliegen. Diese Interpretation wird von einigen der genannten Bibliotheken intern erledigt.

Die Schnittstelle wurde dabei so modifiziert, dass neben der IFC-Datei auch jedes Bauteil in Form von x3d-Dateien exportiert wird. Dieses Format ist ein auf dem XML-Standard basierendes 3D-Grafikformat, welches durch die ISO/IEC 19776 standardisiert ist (web3d, 2013).

Durch eine Verknüpfung über die Bauteil-ID können die x3d-Dateien und das jeweilige Bauteil kombiniert werden (siehe Abbildung 7.1).

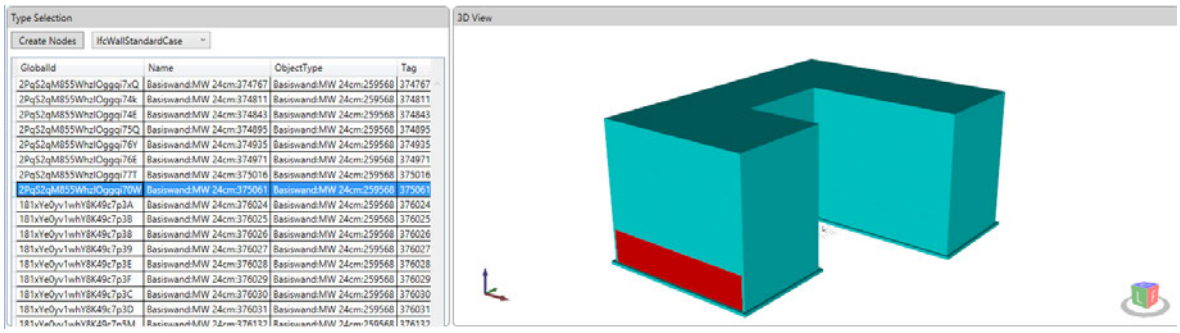

Abbildung 7.1: Verknüpfung eines Bauteils mit seiner Geometrie anhand dessen ID

7.2.9 Bewertung

Aufgrund der gesammelten Informationen wird offensichtlich, dass es viele Bibliotheken für den Einsatz in diesem Projekt in Frage kommen.

In Hinblick auf die Möglichkeit der Visualisierung und eine gesicherte Entwicklung in den nächsten Jahren werden folgende Präferenzen gesetzt:

Für die Entwicklung in einer Java-Umgebung ist das IFC Tools Project (7.2.3) mit Sicherheit die erste Wahl.

In einer C++-Umgebung ist die Entscheidung etwas schwieriger. Die IFC Engine hat hier einen sehr performanten IFC Viewer, jedoch steht das Projekt nicht als OpenSource-Lösung zur Verfügung. Hier ist wohl IfcPlusPlus die beste Wahl.

In einer C#-Umgebung ist auch aufgrund des Mangels an Alternativen das xBIM Toolkit eine gute Wahl. Vor allem die große Anzahl an Projektbeteiligten und die umfassenden Lösungen für verschiedene Aufgabengebiete[1] überzeugen hier.

7.3 Helix Toolkit

Der Hintergrund für den zusätzlichen Aufwand, der mit dem Revit-Exporter betrieben wird, ist, dass es für das IFC-Format kaum Visualisierungsbibliotheken gibt. Im Gegensatz dazu sind für das x3d-Format viele sehr gute Bibliotheken vorhanden. Als Beispiel sei hier das Helix Toolkit genannt (helix3d, 2013).

Helix 3D bietet neben der Visualisierung von x3d-Daten auch die Möglichkeit andere Dateiformate wie zum Beispiel .stl oder .3ds zu visualisieren. Zusätzlich kann die Ansicht des 3D-Modells beliebig angepasst werden. Rotationen, Zoom oder Verschieben des Modells sind kein Problem.

Die einzelnen Elemente können im Viewer selektiert werden und deren Eigenschaften können weiterverwendet werden. Somit lassen sich für dieses Projekt auch Bauteilselektionen bewerkstelligen, bei denen die Bauteil-ID übergeben wird.

7.4 Graph-Bibliotheken

Für die Verwendung von Graphen zur Modellierung und Darstellung wurden im Rahmen dieser Thesis mehrere Bibliotheken in Betracht gezogen.

Für die Java-Entwicklungsumgebung existieren bekannte Lösungen wie zum Beispiel yFiles (http://www.yworks.com/en/index.html). Leider sind hier der Großteil der angebotenen Lösungen nur als Closed Source-Version mit zum Teil sehr hohen Lizenzgebühren verfügbar. Als Alternative wurde hier in Erwägung gezogen, die Graph-Visualisierung selbst zu entwickeln.

Im C++ - und C#-Bereich gibt es hingegen mehrere OpenSource-Lösungen. Besonders positiv ist dabei NodeXL (http://nodexl.codeplex.com) aufgefallen. Hier ist die Visualisierung der Graphen zusätzlich mit gängigen Tools zur Bearbeitung versehen. So können Graphen gezoomt oder auch verschoben werden. Zusätzlich ist die Verknüpfung von Graphen untereinander wie bei den IFC über deren IDs gelöst, was bei der Implementierung weitere Vorteile mit sich bringt, da man hier auf selbige zurückgreifen kann.

[1] http://www.openbim.org/case-studies

7.5 Zusammenfassung

In diesem Kapitel wurde ein Überblick über die Vielzahl an IFC-Bibliotheken gegeben. Aufgrund der erlangten Kenntnisse wird für die Umsetzung des geplanten Softwaretools die Programmiersprache C# gewählt. Da durch den Export von Geometriedaten mit Hilfe des Revit IFC Exporters (Abschnitt 7.2.8) eine gute Lösung zur Visualisierung des 3D-Modells gefunden wurde, bietet sich C# in Zusammenhang mit dem Helix-Toolkit an.

In einem fortgeschrittenen Stadium des Softwaretools ist eine Nutzung der IFC-Bibliothek xBIM empfehlenswert, da der große Funktionsumfang und der Support überzeugen. Dies ist jedoch im Rahmen dieser Thesis nicht umgesetzt worden.

Im nun folgenden Kapitel 8 wird das Konzept und dessen Implementierung für ein Software-Tool vorgestellt, mit dessen Hilfe die theoretischen Lösungsansätze aufbereitet und visualisiert werden können.

Kapitel 8

Entwicklung eines Software-Prototyps

In diesem Kapitel soll nun abschließend das Konzept und die Implementierung eines Software-Tools besprochen werden, welches als Grundlage für die Arbeit an dem eingangs erwähnten DFG-Forschungsprojekt dienen soll.

Das Tool basiert auf einer Software von Simon Daum, der hierfür im Rahmen seines Forschungsprojektes „Raum-zeitliche Anfragesprache für die Prüfung und Analyse von 4D-Bauwerksmodellen“ bereits eine Grundlage für die Visualisierung von IFC-Modellen geschaffen hat. Durch die Zusammenarbeit können Synergien genutzt werden und bei Fortschritten in der Grundsoftware können diese für beide Forschungsprojekte genutzt werden.

8.1 Funktionalität

Der geplante Software-Prototyp soll neben der Visualisierung des 3D-Modells den Bauzeitplan mit Hilfe von Gantt-Diagrammen darstellen können. Zur Ermittlung des Bauzustandes soll die Möglichkeit bestehen, einen Soll-Ist-Abgleich auf Basis von Punktwolkendetektion durchführen zu können.

Im Rahmen dieser Thesis wurden hierzu erste Grundlagen geschaffen. Der Prototyp kann bereits 3D-Modelle darstellen. Dazu verfügt die Software über eine Importfunktion von IFC-Dateien.

Neben einer detaillierten Bauteilliste soll auch der zugehörige Abhängigkeitsgraph sowie ein Gantt-Chart darstellbar sein, um dem Benutzer alle Informationen und Prozessdaten gut aufbereitet anzeigen zu können. Nachdem der Abhängigkeitsgraph nicht automatisiert generiert

werden kann, besteht die Möglichkeit, die Bauteile einzeln auszuwählen und in Abhängigkeit zueinander zu setzen.

Zu einem späteren Zeitpunkt sollen auch die bereits besprochenen Punktwolken in das Tool eingelesen werden können. Zusätzlich muss ein Algorithmus implementiert werden, der die einzelnen Bauteile erkennt und mit dem vorhandenen 3D-Modell abgleichen kann (siehe Kapitel 2).

Nachdem in diesem frühen Stadium der Software noch keine Punktwolken eingelesen werden können, beschränkt sich die Fähigkeit des Tools zu Beginn darauf, dass man einzelne Bauteile selektieren, und deren Erkennung dadurch simulieren kann. Der in Abschnitt 6.3 beschriebene Algorithmus kann dann basierend auf dem Abhängigkeitsgraphen sämtliche Bauteile anzeigen, die von dem ausgewählten logisch abhängen.

In Abbildung 8.1 ist ein Screenshot des aktuellen Entwicklungsstandes des Tools zu sehen. Es verfügt über die grundlegenden Fähigkeiten, ein vorhandenes dreidimensionales Modell einzulesen und dieses auch darzustellen.

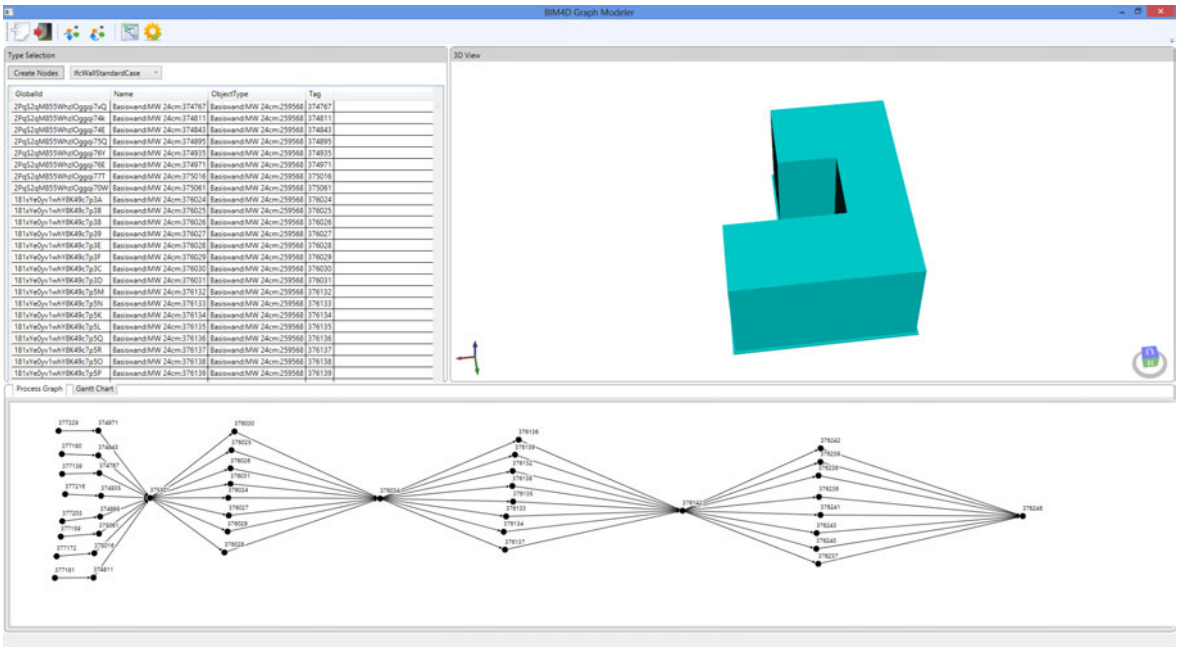

Abbildung 8.1: 3D-Modell mit graphgestützter Baufortschrittsmodellierung

Zusätzlich werden sämtliche Bauteile des Bauwerks aufgelistet und nach ihrer Bauteilgruppe sortiert. Des weiteren kann durch eine graphbasierte Oberfläche der Bauablauf dargestellt und modifiziert beziehungsweise ergänzt werden.

8.2 Programmiersprache und Bibliotheken

Basierend auf den Erkenntnissen aus Kapitel 7 wird im Folgenden erläutert, welche Bibliotheken zum Einsatz kommen.

Das Tool wurde mit der von Microsoft entwickelten Programmiersprache C# umgesetzt. Das Graphical User Interface (GUI) basiert auf der Windows Presentation Foundation (WPF) (Microsoft, 2012). Dieses Präsentationssystem bietet eine Vielzahl an Möglichkeiten, um Benutzeroberflächen und Funktionen schnell und effizient zu implementieren.

Für die dredimensionale Modelldarstellung wurde Helix3D[1] verwendet (Abschnitt 7.3).

Als Graphenbibliothek wurde auf NodeXL[2] zurückgegriffen (siehe Abschnitt 7.4).

Sämtliche verwendeten Bibliotheken wurden als Open Source-Software veröffentlicht und sind in WPF nahtlos integrier- und kombinierbar.

8.3 Aufbau des Tools

Das Tool besteht im Wesentlichen aus vier Arbeitsbereichen. Diese werden im folgenden vorgestellt.

8.3.1 Toolbar

Abbildung 8.2: Toolbar der entwickelten Software

Die Toolbar fasst hierbei die zur Verfügung stehenden Funktionen der Software zusammen.

Es besteht hierbei die Möglichkeit, ein 3D-Modell zu importieren. Außerdem können technologische Abhängigkeiten im XML-Format sowohl importiert als auch exportiert werden (siehe Abschnitt 8.6).

Die prozessrelevanten Funktionen werden in den letzten beiden Buttons zusammengefasst. Mit ersterem lässt sich aus den erstellten Abhängigkeiten ein Gantt-Diagramm erstellen. Mit dem zweiten Button kann für ein ausgewähltes Bauteil angezeigt werden, welche Bauteile von dem gewählten technologisch bedingt abhängen.

[1] Helix3D, http://helixtoolkit.codeplex.com/
[2] NodeXL, http://nodexl.codeplex.com

8.3.2 3D-Darstellung

Hier sind die üblichen Navigationsmöglichkeiten wie Rotation, Zoom oder Verschieben möglich, um das Gebäude von allen Seiten darstellen zu können. Die auf Helix3D basierende 3D-Darstellung auf der rechten Seite des Tools stellt die x3d-Daten des Modells, wie in Abschnitt 7.3 erläutert, dar.

8.3.3 Bauteilinformationen

In der linken Hälfte des Tools werden die Bauteilinformationen, sortiert nach Typ, aufgelistet. Die Liste wird aus dem IFC-Modell erstellt und enthält sämtliche im Modell enthaltenen Bauteile. Diese sind über die Bauteil-ID logisch mit dem 3D-Modell verbunden. Dies ist möglich, da jedes Bauteil in der 3D-Ansicht mit seiner Bauteil-ID abgespeichert ist. Diese ist auch in der Bauteilliste vorhanden, wodurch hier eine Verknüpfung hergestellt werden kann.

8.3.4 Graph-Visualisierung

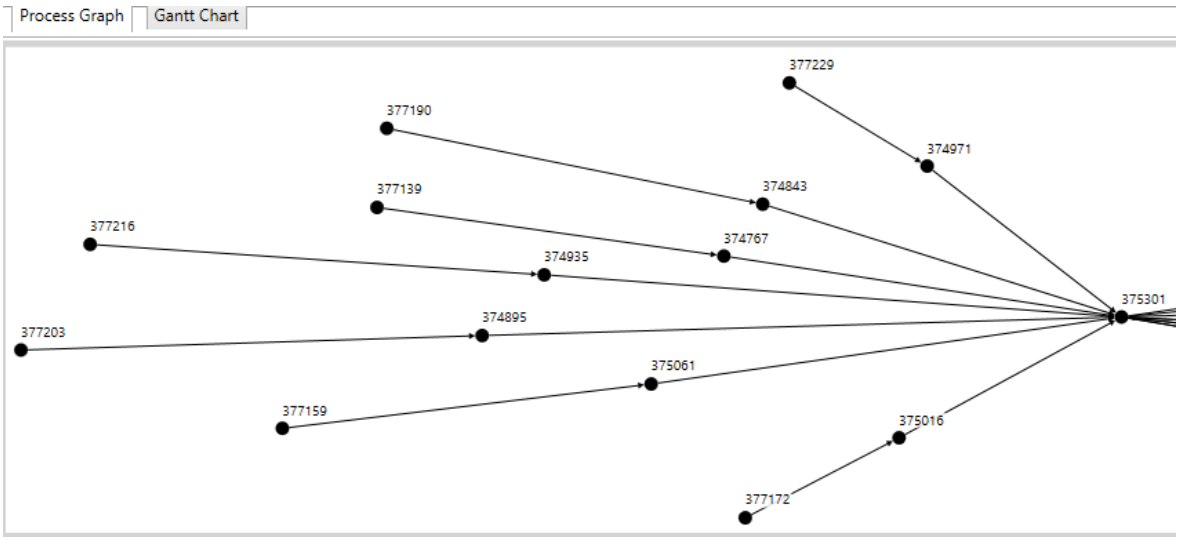

Abbildung 8.3: Bauteilabhängigkeiten in der Graphansicht

Die Graph-Visualisierung befindet sich im unteren Teil des GUI. Sie basiert auf der NodeXL-Bibliothek und ist sowohl mit der Bauteilliste als auch mit dem 3D-Modell verknüpft. Das bedeutet, dass Elemente in der Graph-Ansicht bei einer Selektion auch in der 3D-Ansicht markiert werden.

Aus der Bauteilliste können die einzelnen Elemente ausgewählt und in der Graph-Ansicht nach Belieben angeordnet werden. Zwischen den einzelnen Elementen lassen sich gerichtete Kanten erstellen (siehe Abschnitt 5.8), mit deren Hilfe sich die technologisch bedingten exter-

nen Abhängigkeiten (siehe Abschnitt 5.8.2) erstellen lassen. Diese können in eine gesonderte Datei exportiert und auch wieder importiert werden (Abschnitt 8.6).

Im aktuellen Entwicklungsstand der Software muss der gesamte Graph noch manuell erstellt werden. Es ist geplant, dass mit Einführung der IFC4 und nach Implementierung eines entsprechenden Importers auch entsprechende Informationen direkt aus der IFC-Datei übernommen werden können.

8.3.5 Gantt-Diagramm

Neben der Graph-Visualisierung kann auch ein Gantt-Diagramm angezeigt werden (siehe 5.6). Da aktuell noch keine Bauzeiten in der IFC unterstützt werden, wird hier pauschal eine Standard-Zeit angenommen und somit ist jeder Block in dem Diagramm gleich lang.

Für zukünftige Versionen mit realen Prozesszeiten ist geplant, diese entsprechend skaliert mit ihrer realen Prozessdauer anzuzeigen.

8.4 IFC Import

Wie in Kapitel 7.2 vorgestellt, steht eine Vielzahl an Open- und auch Closed-Source-Bibliotheken zur Verfügung, welche die Verarbeitung von IFC-Dateien unterstützen.

Für das entwickelte Softwaretool und auch das nachfolgende DFG-Forschungsprojekt ist es von essentieller Bedeutung, jegliche Arten von IFC-Dateien importieren zu können und das Modell sowohl darzustellen als auch einzelne Bauteile interaktiv auswählen zu können sowie die dreidimensionale Ansicht passend modifizieren zu können.

Im Rahmen dieser Thesis konnte keine Bibliothek in dieser Tiefe modifiziert und implementiert werden, so dass dieser Schritt möglich gewesen wäre. Die Komplexität des IFC-Formats stellt hier eine hohe Hürde dar.

Aus diesem Grund wurde der ebenfalls in Kapitel 7.2 beschriebene Umweg über die als Open Source zur Verfügung stehende Autodesk Revit 2013 - IFC - Schnittstelle gewählt. Zukünftig soll dieses Problem durch die Verwendung einer IFC-Bibliothek mit integrierter Visualisierung gelöst werden.

8.5 Prozesse

Die Prozess-Eigenschaften in dem Softwaretool basieren wie erwähnt bisher nur auf den Daten, die der Benutzer selbst manuell hinzufügt. Dabei liegt der Schwerpunkt auf den technolo-

gischen Abhängigkeiten, da diese für den Bauablauf und vor allem die Varianten-Verarbeitung benötigt werden.

Um einen Prozess für ein Bauteil in der Graphenansicht zu erstellen, müssen die Knoten für jedes Bauteil generiert werden. Dazu wählt man das gewünschte Bauteil aus und klickt auf „Create Nodes“ (Abbildung 8.4).

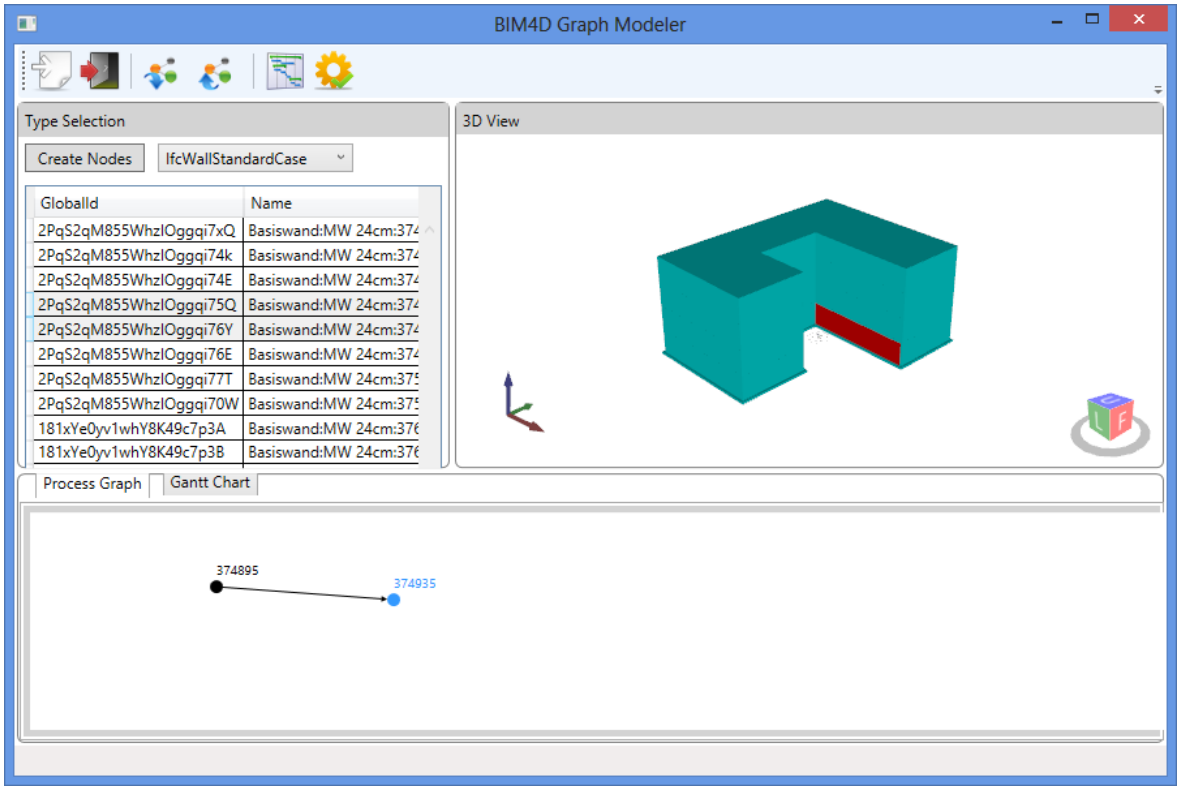

Abbildung 8.4: Erstellen von Prozessen und Abhängigkeiten

Die daraufhin erstellen Knoten sind genau wie die Bauteilliste mit dem 3D-Modell über die jeweilige ID verbunden und werden bei einer Selektion auch visuell hervorgehoben.

Um die logische Abhängigkeit zwischen den einzelnen Bauteilen darzustellen, wählt man die gewünschten Knoten per Rechtsklick aus, um dadurch eine Kante zwischen den beiden herzustellen. Der erste Klick erfolgt dabei auf das Basisbauteil, der zweite auf das Bauteil, dass darauf aufbaut, also davon abhängig ist.

8.6 Abspeichern der Abhängigkeiten

Der generierte Graph kann als XML-Datei sowohl exportiert als auch wieder importiert werden. Nach der geplanten Umstellung des Tools auf eine native IFC4-Unterstützung werden diese Abhängigkeiten direkt in der IFC-Datei gespeichert.

In dieser Thesis wurde für das Abspeichern der Abhängigkeiten auf das graphML-Dateiformat zurückgegriffen (graphml, 2007). Das zugrunde liegende Schema wird im Anhang A.1 aufgeführt.

Dieses wurde dabei so angepasst, dass die einzelnen Knoten mit der zugehörigen Bauteil-ID abgespeichert wurden. Dadurch lässt sich bei einem Import eines zuvor gespeicherten Abhängigkeitsgraphen eine erneute Verknüpfung mit den zugehörigen Bauteilen herstellen. So bleibt die Funktionalität des Tools in vollem Umfang erhalten.

8.7 Verarbeitung technologisch bedingter Abhängigkeiten

Diese Funktion stellt die eigentliche Kernaufgabe der Software dar und ist besonders wichtig für die künftige Weiterentwicklung, um das Ziel einer automatisierten Prozessüberwachung zu erreichen.

8.7.1 Verarbeitung von Punktwolken

Der Grundgedanke hinter dieser Funktion ist, dass zu einem späteren Zeitpunkt Punktwolken mit dem aktuellen Bauzustand eines Gebäudes in die Software eingespielt werden. Aufgabe der Software ist es dann, diese Punktwolken mit dem digitalen 3D-Modell abzugleichen.

Die Punktwolken können jedoch nur die Hülle des Gebäudes und sämtliche frei sichtbaren Elemente abbilden, da mit den Mitteln der Photogrammetrie keine verdeckten und innen liegenden Bauteile erkannt werden können.

Folglich müssen, basierend auf der erkannten Hüll-Punktwolke, Rückschlüsse über verdeckte Bauteile getroffen werden. Dabei helfen die technologischen Abhängigkeiten, welche auch in der Graph-Ansicht des Tools dargestellt werden.

Die Erkennung eines Bauteils aus der Punktwolke ist in diesem Stadium der Software noch nicht möglich. Es kann jedoch ein beliebiges Bauteil mit der Maus ausgewählt werden und eine Erkennung dieses Bauteiles simuliert werden.

8.7.2 Visualisierung

Nach Ausführen der Breitensuche (siehe Kapitel 6.3) werden sämtliche Bauteile, von denen das selektierte technologisch bedingt abhängig ist, in einer Liste gespeichert.

Um das Ergebnis auch visuell darzustellen, werden in dem Softwaretool alle Elemente farblich hervorgehoben (Abbildung 8.5).

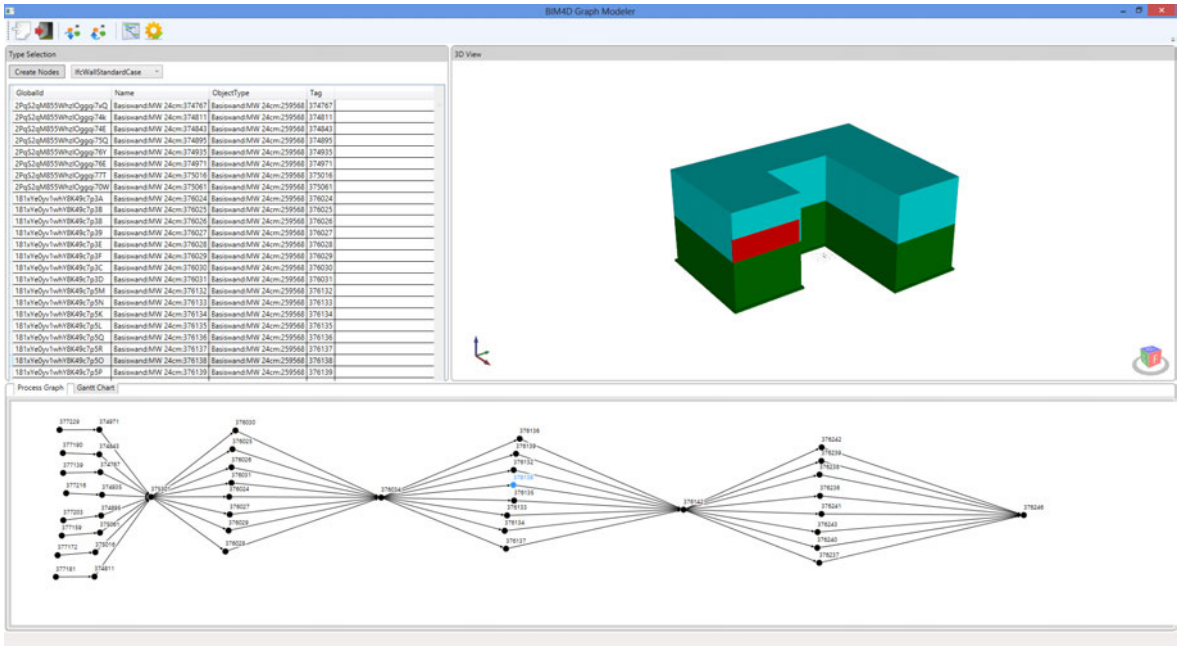

Abbildung 8.5: Visualisierung baubedingter Abhängigkeiten

8.8 Geplante Weiterentwicklung

Mit Veröffentlichung des neuen IFC4-Standards bieten sich hier aufgrund der Neuerungen im Prozessmanagement (Abschnitt 3.3.3) sehr gute Möglichkeiten einer tieferen Integration in das bestehende Tool. Sobald bestehende Bibliotheken angepasst werden, wird basierend auf bisherigen Erkenntnissen eine Implementierung angestrebt.

Kapitel 9

Projektbaustelle

Um während der Forschung an dem geplanten DFG-Projekt einen realitätsnahen Bauablauf abbilden zu können ist eine Referenzbaustelle unabdingbar. Von besonderer Bedeutung bei der Abbildung des Ablaufes ist neben dem jeweils aktuellen Bauzustand auch das Baugeschehen mit sämtlichen Baugeräten und Hilfskonstruktionen wie zum Beispiel Schalungen oder Betonmischmaschinen.

Für dieses Projekt konnte der Neubau eines Wohn- und Geschäftshauses in der Karlstraße in der Gemarkung Maxvorstadt in München gewonnen werden.

9.1 Standort

Das geplante Gebäude befindet sich auf einer dreieckigen Grundfläche im Münchner Stadtteil Maxvorstadt, welche von der Dachauer-, Augusten- sowie der Karlstraße begrenzt wird.

Die Lage der Baustelle ist, wie in Abbildung 9.1 gut zu sehen ist, für die wöchentliche Datenaufnahme durch Mitarbeiter des Photogrammetrie-Lehrstuhls der TU München äußerst gut geeignet, da sie fußläufig erreichbar ist.

9.2 Gebäudebeschreibung

Der sechsstöckige Neubau mit Tiefgarage hat eine Brutto-Grundfläche von $4.200 m^2$.

Der architektonische Entwurf stammt von Kühn Malvezzi Architekten aus Berlin. Die Bauausführung wird von der Firma Leitner Bauunternehmung aus Wolfratshausen geleitet.

Es handelt sich um einen dreieckigen Grundriss, dessen Ecken markant abgerundet wurden.

Abbildung 9.1: Standort der Referenzbaustelle (http://geoportal.bayern.de/bayernatlas/)

Abbildung 9.2: Modell der Baustelle (c) Kühn Malvezzi Architekten

Alle projektbeteiligten Firmen stehen einer Zusammenarbeit mit der TU München sehr offen gegenüber und sind bereit, sämtliche für das Forschungsprojekt relevanten Daten zur Verfügung zu stellen. Auch die Aufnahmen des Fachgebiets für Photogrammetrie dürfen im laufenden Baubetrieb unter Beachtung der geltenden Sicherheitsvorschriften gemacht werden.

9.3 Geplante Aufzeichnungen

Die Rohbauphase hat nach der Winterpause Anfang April begonnen. In der Regel wird die Rohbauphase in relativ hoher Taktfrequenz ausgeführt und es entsteht bei dieser Grundfläche in circa zwei bis drei Wochen ein neues Stockwerk.

Um eine korrekte Baufortschrittskontrolle aufnehmen zu können, ist geplant, wöchentlich Aufnahmen zu machen. Die geringe Grundfläche ist für das Forschungsprojekt in der Pilotphase von Vorteil, da das Grundstück so ohne größere Probleme relativ eng mit Zielmarken (siehe Abschnitt 3.1) ausgestattet werden kann, die jeweils geodätisch verortet werden.

Des Weiteren ist zu Beginn des Forschungsprojekts noch nicht genau geklärt, welche Foto-Dichte notwendig ist, um einen guten Detaillierungsgrad der Punktwolke zu generieren. So können ohne großen Mehraufwand mehr Fotos als eigentlich nötig gemacht werden, um sämtliche Eventualitäten abzudecken. Durch die geringe Größe der Baustelle entstehen so keine zu große Datenmengen.

Abbildung 9.3 zeigt die Erstellung der Aufnahmen mit einer handelsüblichen, digitalen Spiegelreflex-Kamera. Das Gebäude befindet sich zu diesem Zeitpunkt gerade in der Fertigstellung des Kellergeschosses. In dieser Phase können die Fotos noch ebenerdig rund um die Baustelle geschossen werden.

Wenn das Gebäude das erste Stockwerk erreicht hat, können aus dieser Perspektive jedoch keine verwertbaren Aufnahmen mehr gemacht werden. Aktuell laufen aus diesem Grund Gespräche mit Anwohnern, um deren Balkone als Aufnahmepunkte zu nutzen und so gute Ausgangspunkte für die weiteren Aufnahmen zu sichern. Zusätzlich ist der Kran auf der Baustelle begehbar. Dadurch können auch aus dieser Perspektive weitere Aufnahmen gemacht werden.

Abbildung 9.3: Aufzeichnungen auf der Baustelle

9.4 3D Modell

Das Architekturbüro Kühn Malvezzi hat zusätzlich ein digitales dreidimensionales Modell des geplanten Gebäudes zur Verfügung gestellt (Abbildung 9.4).

Abbildung 9.4: konvertiertes Modell der Baustelle in Autodesk Revit 2013

Dieses wurde mit der Software Rhinoceros 5 der Firma Robert McNeel & Associates erstellt (Associates, 2007). Rhino unterstützt nur eingeschränkt BIM, weswegen sämtliche Bauteile ausschließlich aus Flächen bestehen, die keinerlei Bauteilinformationen besitzen. Hier wird deutlich, dass sich die BIM-Technologie auch in diesem Projekt noch nicht durchgesetzt hat (siehe 3.2.2).

Eine grundlegende Aufgabe ist es deshalb, das digital vorhandene Modell so aufzubereiten, dass es auch als Building Information Model weiter genutzt werden kann. Um dies zu bewerkstelligen muss das 3D-Modell in eine BIM-Software überführt werden. Für dieses Projekt wurde dazu Autodesk Revit gewählt.

Sowohl Rhino als auch Revit unterstützen bereits den IFC-Standard, jedoch können in Rhino keine Bauteilinformationen hinterlegt werden. Die IFC-Schnittstellen wurden am *Lehrstuhl für Computergestützte Modellierung und Simulation* bereits mehrfach untersucht.

Dabei wurde festgestellt, dass selbst von buildingSMART zertifizierte Softwarehersteller teils fehlerhaft umgesetzte IFC-Importer und -Exporter zur Verfügung stellen (Ritter, 2011).

Diese Erfahrung wird auch hier bestätigt. Das von Rhino nach Revit importierte IFC-Modell enthält zahlreiche Fehler. Vor allem mit Öffnungen für Fenster o.Ä. scheint eines der beiden Programme erhebliche Probleme zu haben. Diese sind zwar vorhanden, jedoch mit Flächen ausgefüllt (Abbildung 9.5). Zusätzlich sind viele Flächen mehrfach übertragen worden und überlagern sich so im neuen Modell.

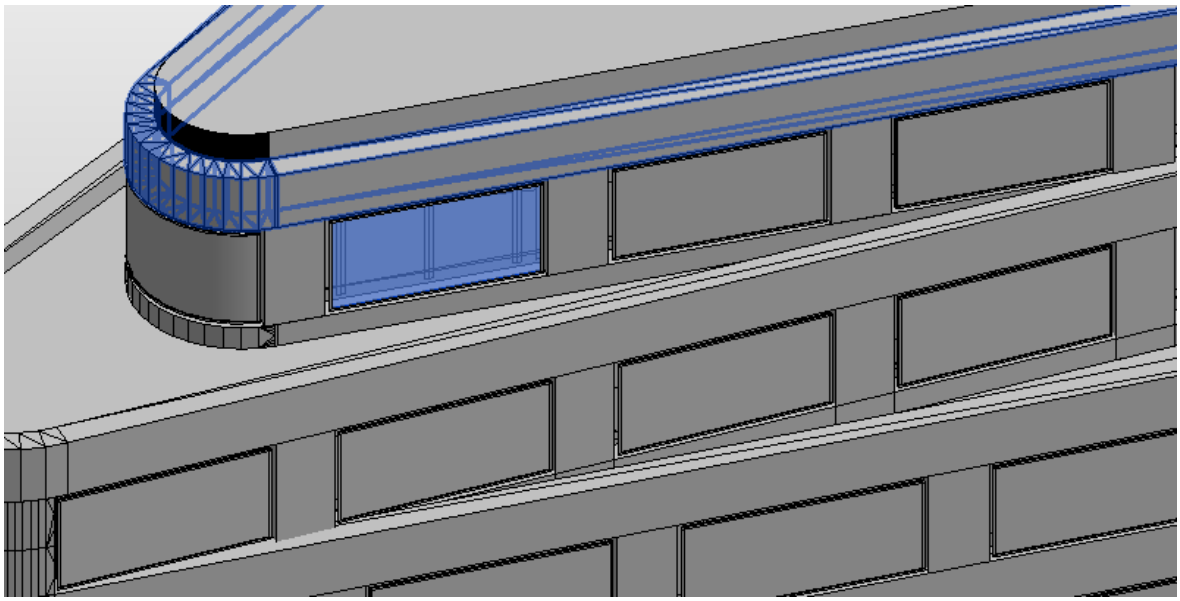

Abbildung 9.5: Datentransfer von Rhino nach Revit

Aufgrund dieser massiven Mängel muss das Modell komplett neu erstellt werden. Dabei müssen auch die Bauteilinformationen hinterlegt werden. Außerdem wird der zuständige Statiker der Baustelle kontaktiert, um weitere Baupläne und vor allem Abmessungen des Rohbaus zu erhalten.

9.5 Bauablauf

Für dieses Bauprojekt wird der vorhandene Bauablaufplan verwendet, um das erstellte *Building Information Model* mit Termininformationen zu versehen. Dieser Vorgang muss durch den Anwender manuell vorgenommen werden. In einem fortgeschrittenen Szenario wäre es im Rahmen des Building Information Modeling denkbar, die Prozessdaten zusammen mit dem 3D-Modell zu importieren. Dazu müssen diese jedoch zuvor, wie in Abschnitt 3.2 beschrieben, miteinander kombiniert werden.

Basierend darauf kann so auch nach Abschluss der Baustelle ein Baufortschritt dargestellt werden. Nachdem alle Prozessdaten manuell in das digitale Modell integriert werden müssen, können hier direkt Alternativen abgebildet und getestet werden.

Kapitel 10

Zusammenfassung und Ausblick

10.1 Zusammenfassung

Als Grundlage für diese Thesis werden aktuelle Entwicklungen in der Bauinformatik aus dem Bereich der Prozessmodellierung untersucht. Mit Building Information Modeling ergeben sich neue Möglichkeiten die Prozesssteuerung auf der Baustelle zu betreiben.

Diese Thesis beschäftigt sich mit Verfahren der automatisierten Baufortschrittskontrolle.

Die Weiterentwicklung der Industry Foundation Classes und der damit einhergehenden tiefgreifenden Unterstützung im Bereich der Bauprozesse ermöglicht eine gute Nutzungsmöglichkeit für detaillierte Bauplanungen. Durch die Einführung der IFC4 sind nun sämtliche wichtigen Prozess-Eigenschaften wie Abhängigkeiten oder Start- und Endzeitpunkte abbildbar.

Im methodischen Kern dieser Arbeit wird die Theorie der Bauablaufplanung beleuchtet. Der Darstellung und Ermittlung von Abhängigkeiten in Kombination mit der IFC wird besondere Aufmerksamkeit gewidmet. Zusätzlich wird untersucht, wie sich Alternativen im Bauablauf ermitteln und auch abbilden lassen.

Es werden die gesammelten Informationen zur Graphentheorie und dem Bauablauf abgebildet. Zusätzlich wird dargelegt, wie Abhängigkeiten erkannt werden können. Außerdem wird gezeigt, welche Möglichkeiten es gibt, Alternativen auf der Baustelle zu erkennen.

Besonderes Augenmerk wird auf die sogenannten Checkpoint Components gelegt, die in den Abhängigkeitsgraphen als Artikulation auftreten und für die Bauteilerkennung auf Basis von Punktwolken die Möglichkeit einer zusätzlichen Verifizierung bieten. Dadurch wird der Soll-Ist-Abgleich erleichtert und so eine zuverlässigere Aussage über den aktuellen Bauzustand gewährleistet.

Basierend darauf wird gezeigt, welchen Einfluss Verzögerungen im Bauablauf auf die Gesamtprozessdauer haben und wie diese in Gantt-Charts dargestellt werden können.

Darauf folgend werden verfügbare Bibliotheken für die Nutzung der IFC untersucht. Für die graphbezogene Darstellung der Bauabläufe wurden auch passende Graph-Bibliotheken untersucht.

Des Weiteren zeigt das Kapitel 8 (Implementierung) die Umsetzung der erarbeiteten Erkenntnisse.

Abschließend wird eine Projektbaustelle vorgestellt und dargelegt, wie diese für das Projekt genutzt wird. Besondere Beachtung findet hier die Nutzung eines zugehörigen *Building Information Models*, welches erst aus den vorhandenen Daten erstellt werden muss.

10.2 Fazit

Für die Automatisierung der Baufortschrittskontrolle gibt es durch die Photogrammetrie gute Ansatzpunkte. Die Prozessdarstellung kann digital erfolgen. Vorhandene Bibliotheken können als Unterstützung für die Nutzung der IFC und andere Aufgaben herangezogen werden.

Abhängigkeiten lassen sich unkompliziert ermitteln, insbesondere technologisch bedingte Abhängigkeiten können sowohl über die IFC gut abgespeichert als auch für die Graph-Darstellung genutzt werden. Die zeitlichen Abhängigkeiten müssen mit der IFC4-Einführung erneut geprüft und passend implementiert werden. Hierbei soll in Zukunft der Vorteil des *Building Information Models* genutzt werden und sämtliche Informationen in einer Datei abgespeichert werden.

10.3 Ausblick

Für das Forschungsprojekt gibt es bereits sehr konkrete Pläne für das weitere Vorgehen.

10.3.1 Photogrammetrie

Aktuell werden die Aufnahmen des Photogrammetrie-Lehrstuhls getätigt um die Baustelle und deren Baufortschritt ständig kontrollieren zu können.

Des Weiteren werden verschiedene Verfahren zur Punktwolkengenerierung getestet um mit möglichst wenigen Aufnahmen eine möglichst detaillierte Punktwolke generieren zu können.

10.3.2 Software

Neben der in Abschnitt 8.8 erwähnten Umstellung auf IFC4 sind bereits weitere Neuerungen für das Softwaretool geplant.

So soll eine IFC-Bibliothek implementiert werden (Abschnitt 7.2), mit deren Hilfe die Gebäudedaten gut importiert als auch modifiziert und exportiert werden können.

Eine große Herausforderung besteht in dem geplanten Import der Punktwolken aus der Photogrammetrie. Besonders der Abgleich mit dem 3D-Modell ist schwierig. Hier gilt es herauszufinden, wie digitale Bauteile mit der vorhandenen Punktwolke verglichen werden können. Nachdem auf einer Baustelle maximal mit einer Genauigkeit im Zentimeterbereich gearbeitet werden kann, müssen Parameter gefunden werden, um die Bauteile trotz geringfügiger Abweichungen erkennen zu können.

Weitere Herausforderungen stellen, wie bereits in Abschnitt 2.1 beschrieben, auch die Hilfswerkzeuge wie Betonmischer und Schalungen auf der Baustelle dar, die zwar in der Punktwolke existieren, für den Baufortschritt aber vollkommen irrelevant sind.

10.4 Weiterführende Literatur

Die Grundlagen der Prozessplanung in der Bauinformatik werden in dieser Thesis beschrieben. Für einen tieferen Einblick lohnt sich das Studium der Dissertation von Felix Enge, die auch im Literaturverzeichnis aufgeührt ist.

Die exakten Details zur automatisierten Erkennung von Prozessen und technologisch bedingten Abhängigkeiten inklusive wichtiger Kennwerte wird von Eike Tauscher in seiner Dissertation behandelt. Auch diese findet sich im Literaturverzeichnis.

Anhang A

Code

A.1 graphML

```
<?xml version="1.0" encoding="UTF-8"?>
<graphml xmlns="http://graphml.graphdrawing.org/xmlns">
  <key id="V-~VDLabel" for="node" attr.name="~VDLabel" attr.type="string" />
  <key id="V-~VDRadius" for="node" attr.name="~VDRadius" attr.type="string" />
  <key id="E-~EDLabel" for="edge" attr.name="~EDLabel" attr.type="string" />
  <graph edgedefault="directed">
    <node id="375016">
      <data key="V-~VDLabel">375016</data>
      <data key="V-~VDRadius">5</data>
    </node>
    <node id="375061">
      <data key="V-~VDLabel">375061</data>
      <data key="V-~VDRadius">5</data>
    </node>
    <node id="376024">
      <data key="V-~VDLabel">376024</data>
      <data key="V-~VDRadius">5</data>
    </node>
    <node id="376025">
      <data key="V-~VDLabel">376025</data>
      <data key="V-~VDRadius">5</data>
    </node>
    <edge source="375061" target="376025" />
    <edge source="376024" target="376025" />
```

```
    <edge source="375016" target="376025" />
    <edge source="376024" target="375016" />
  </graph>
</graphml>
```

Anhang B

Compact Disc

Auf der beigefügten CD befindet sich folgender Inhalt:

- Die vorliegende Thesis [PDF]
- Das in der Thesis verwendete Beispielprojekt [IFCXML + X3D]
- Der Bauablaufgraph für das in der Thesis verwendete Beispielprojekt [XML]
- Das 3D-Modell der Projektbaustelle [IFC, 3DM]
- Der Quellcode des erstellten Programms [SLN, XAML, CS, DLL]
- eine kompilierte Version des erstellten Programms [EXE]

Literaturverzeichnis

Andrae, M. (2012). Entwicklung eines Mangelaufnahme-Systems auf Mobilen Geräten für den Einsatz bei der Objektüberwachung zur weiteren zentralen Verarbeitung. Master thesis, Lehrstuhl für Computergestützte Modellierung und Simulation, Technische Universität München.

Associates, R. M. . (2007). Rhinoceros - Rhino5. http://www.rhino3d.com.

Autodesk (2013a). Autodesk BIM360 Field - Produktwebseite. http://www.bim360field.com.

Autodesk (2013b). Soureforge - IFC Exporter for Autodesk Revit. http://sourceforge.net/projects/ifcexporter/.

Brandt, J. (2012). Gespräch mit Jordan Brandt, Entwickler von Autodesk BIM 360 Glue, am 26.09.2012.

buildingSMART (2007a). buildingSMART Documentation for IFC2x3. http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/alphabeticalorder_entities.htm.

buildingSMART (2007b). buildingSMART Documentation for IfcGeometryResource. http://www.buildingsmart-tech.org/ifc/IFC2x3/TC1/html/ifcgeometryresource/ifcgeometryresource.htm.

buildingSMART (2013a). buildingSMART special supplement. http://www.buildingsmart.org/resources/publications/newsletters-magazines/ifc4-special-supplement.

buildingSMART (2013b). Organisationsstruktur von buildingSMART. http://www.buildingsmart.de/buildingsmart.

Diestel, R. (1996). *Graphentheorie*. Springer, Berlin.

Eastman, C., Teicholz, P., Sacks, R. & Liston, K. (2011). *BIM Handbook, A Guide to Building Infromation Modeling for Owners, Managers, Designers, Engineers, and Contractors* (Second Aufl.). John Wiley & Sons, Inc.

Enge, F. (2009). Muster in Prozessen der Bauablaufplanung - Ein Branch-and-Bound-Verfahren zur Mustererkennung in Planungs- und Ausführungsprozessen. Dissertation, Fakultät VI Planen, Bauen, Umwelt der Technischen Universität Berlin.

Fischer, M. & Aalami, F. (1996). Scheduling with Computer-Interpretable Construction MethodModels. *Journal of Construction Engineering and Management* 122(4), S. 337–347.

Fliegner, C. (2003). Umsetzung der IFC mit Java und XML am Beispiel der Tragwerksplanung. Diplomarbeit, Bauhaus-Universität Weimar.

Foley, J. D., Van Dam, A., Feiner, S. K., Hughes, J. F. & Phillips, R. L. (1994). *Introduction to computer graphics*, Volume 55. Addison-Wesley Reading.

GNU (2011). Lizenzen des GNU-Projekts. http://www.gnu.org/licenses/.

graphml (2007). GraphML Dateiformat. http://graphml.graphdrawing.org.

helix3d (2013). Helix 3D Toolkit. http://helixtoolkit.codeplex.com.

Henley, E. J. & Williams, R. (1973). *Mathematics in Science and Engineering*. Elsevier.

Huhnt, W. (2005). Generating sequences of construction tasks. In: *Proceedings of 22nd of W78 Conference on Information Technology in Construction, Dresden, Germany*, S. 17–22.

ISO 10303-11 (1998). Product data representation and exchange.

Kim, C., Son, H. & Kim, C. (2013). Automated construction progress measurement using a 4D building information model and 3D data. *Automation in Construction* 31(0), S. 75 – 82.

Microsoft (2005). MSDN - Early and Late Binding. http://msdn.microsoft.com/en-us/library/0tcf61s1%28v=vs.80%29.aspx.

Microsoft (2012). MSDN - Windows Presentation Foundation. http://msdn.microsoft.com/de-de/library/ms754130.aspx.

OpenCascade (2013). Open CASCADE Technology, 3D modeling numerical simulation. www.opencascade.org.

Rao, S. & Ramachandra Rao, A. (1972). The number of cut vertices and cut arcs in a strong directed graph. *Acta Mathematica Academiae Scientiarum Hungarica* 22(3-4), S. 411–421.

Ritter, F. (2011). Untersuchung der Möglichkeiten und Vorteile des modellgestützten Planens anhand von Autodesk Produkten. Master thesis, Lehrstuhl für Computergestützte Modellierung und Simulation, Technische Universität München.

Tauscher, E. (2011). Vom Bauwerksinformationsmodell zur Terminplanung - Ein Modell zur Generierung von Bauablaufplänen. Dissertation, Fakultät Bauingenieurwesen der Bauhaus-Universität Weimar.

Tulke, J. (2010). Kollaborative Terminplanung auf Basis von Bauwerksinformationsmodellen. Dissertation, Fakultät Bauingenieurwesen der Bauhaus-Universität Weimar.

Tuttas, S. (2013). Gespräch mit Sebastian Tuttas, wiss. Mitarbeiter am Fachgebiet für Photogrammetrie der TU München.

UK-HM-Government (2012). Industrial strategy: government and industry in partnership - Building Information Modeling. https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/34710/12-1327-building-information-modelling.pdf.

web3d (2013). Web3d Consortium Homepage. http://www.web3d.org/about/faq/.

Wilson, J. M. (2003). Gantt charts: A centenary appreciation. Paper, European Journal of Operational Research, Volume 149, Issue 2.

Würfele, F. & Bielefeld, B. (2007). *Bauobjektüberwachung: Kosten-Qualitäten-Termine-Organisation-Leistungsinhalt-Rechtsgrundlagen-Haftung-Vergütung.* Friedr. Vieweg und Sohn Verlag.

Zimmermann, J. (2009). *Grundkurs Bauprozessmanagement.* Lehrstuhl für Bauprozessmanagement.

Eidesstaatliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Master-Thesis selbstständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht.

Ich versichere außerdem, dass die vorliegende Arbeit noch nicht einem anderen Prüfungsverfahren zugrunde gelegen hat.

München, 15. Juni 2013

Alexander Braun

Alexander Braun
[redacted]