

Ingenieur fakultät Bau Geo Umwelt

Lehrstuhl für Computergestützte Modellierung und Simulation

Prof. Dr.-Ing. André Borrmann

Webbasierte Prozessvisualisierung mit Hilfe eines 4D Modells

Barth, Alexander

Bachelorthesis

für den Bachelor of Science Studiengang Bauingenieurwesen

Autor: Barth, Alexander

Matrikelnummer:



Betreuer: Prof. Dr.-Ing. André Borrmann

Alexander Braun

Ausgabedatum: 06. Juli 2015

Abgabedatum: 14. Oktober 2015

Zusammenfassung

Die Baustellenüberwachung wird im Moment hauptsächlich manuell durchgeführt. Mit Hilfe eines Web Tools kann diese in einem Browser am Computer durchgeführt werden.

Die Bachelorthesis beschreibt die Grundlagen in Bezug auf die Baustellenüberwachung. Sie geht dabei näher auf die Terminplanung ein und stellt dar, welche verschiedenen Arten von Plänen es gibt.

Des Weiteren werden die Grundlagen beschrieben, die für das Verständnis von 3D Modellen und die technische Umsetzung des Web Tools notwendig sind.

Eine besondere Herausforderung besteht darin, die eingelesenen visuellen Daten einer Baustelle digital wiederzugeben. Hiermit beschäftigt sich die Arbeit am Ende. Dabei wird die Umsetzung sowie die Art und Weise, wie die Software programmiert wurde, beschrieben.

Abstract

The monitoring of the building site is mainly made by hand at the moment. With the help of an web tool this can be done on the web browser.

The bachelorthesis describes the basics, which are necessary for the monitoring of the building site. Furthermore the time schedules and their functions are shown in this thesis.

The Basics about 3D models, which are necessary for the web tool, are also written in the bachelorthesis.

The major task is the presentation of the imported informations. This will be done with the web tool.

These tasks are written at the end of this thesis. Particularly she describes the manner, how the web tool was programmed.

Inhaltsverzeichnis

1	Einführung	1
1.1	Idee und Ziel der Arbeit	2
1.2	Aufbau der Arbeit	2
2	Baufortschrittskontrolle	4
2.1	Projektgrundlage	4
2.2	Building Information Modeling	4
2.3	4D	5
2.4	IFC	5
2.5	Zusammenfassung	6
3	Bauablaufplan	7
3.1	Arten	7
3.1.1	Terminliste	7
3.1.2	Balkendiagramm	8
3.1.3	Weg-Zeit-Diagramm	9
3.1.4	Netzplan	10
3.2	Ressourcen	10
3.3	Beziehungen	10
3.3.1	Pufferzeiten	12
3.4	Probleme und Möglichkeiten	12
3.4.1	Abweichungen im Terminplan	12
3.4.2	Verzögerung des Bauablaufs	12
3.4.3	Verkürzung des Bauablaufs	13
3.4.4	Soll-Ist-Vergleich	13
3.4.5	Rechtliche Schritte bei Abweichungen vom Terminplan	13
3.5	Zusammenfassung	15
4	Theorie	16
4.1	HTML	16
4.2	CSS	17

4.3	PHP	18
4.4	MySQL	20
4.5	JavaScript	21
4.6	WebGL / three.js	22
4.7	STL	23
4.8	Mobile Responsive Design	24
4.9	Zusammenfassung	25
5	Umsetzung	26
5.1	Allgemein	26
5.1.1	Projektdatei	26
5.2	Software und Hardware	27
5.2.1	Coda	27
5.2.2	GitLab	27
5.2.3	LRZ Server	28
5.3	Soll-Ist-Vergleich: Bauablaufplan - Datenbank	28
5.4	Konzept und Überlegungen für die Programmierung der Webseite	29
5.4.1	Import	29
5.4.2	Darstellung	29
5.5	Datenbank	30
5.6	Skript	31
5.6.1	Import von Dateien	31
5.6.2	Darstellung des Modells	32
5.7	Dateiarchitektur	38
5.8	Gerätekompatibilität und Rechenleistung	38
5.9	Zusammenfassung	39
6	Zusammenfassung und Ausblick	40
6.1	Zusammenfassung	40
6.2	Fazit	41
6.3	Ausblick und Vision	41
6.3.1	Einpflegen der Daten	41
6.3.2	Optimierung der Performance	42
6.3.3	Lauffähigkeit auf Endgeräten	42
A	Datenträger	43

Abbildungsverzeichnis

3.1	Terminliste (Zanner, 2014)	8
3.2	Balkendiagramm	9
3.3	Weg-Zeit-Diagramm (Zanner, 2014)	9
3.4	Netzplan (Zanner, 2014)	10
3.5	Anordnungsbeziehungen (Zanner, 2014)	11
4.1	STL Binary Format (Ennex Corporation, 1999)	24
4.2	Triangulierung	24
5.1	Beziehungen der Tabellen in der Datenbank	31
5.2	Formular zur Anpassung der Darstellung	34

Tabellenverzeichnis

5.1	Beschreibung der Tabellen in der Datenbank	30
5.2	Zusammenfassung der Gerätetests	39

Abkürzungsverzeichnis

4D	Vierdimensional
BIM	Building Information Modeling
HTML	Hypertext Markup Language
PHP	PHP Hypertext Preprocessor
CSS	Cascading Stylesheets
IFC	Industry Foundation Classes
CAD	Computer-Aided Design
LRZ	Leibniz Rechenzentrum
STL	STereoLithography
WebGL	Web Graphics Library

Kapitel 1

Einführung

Die digitale Entwicklung schreitet immer weiter voran, doch beim Thema Baustelle ist diese noch nicht angekommen. Immer noch werden zahlreiche Aufgaben analog, also händisch durchgeführt. Was wäre, wenn einer der wichtigsten Vorgänge auf Baustellen, die Bauüberwachung, digital durchführbar wäre? Die Möglichkeiten, die damit zusammenhängen, wären enorm. Ein verbesserter und organisierter Ablauf wäre möglich, die Zusammenarbeit wäre einfacher und Verschiedenes mehr.

Die Idee dieser Arbeit besteht darin, genau diesen Aspekt umzusetzen: Die Bauablaufplanung zu digitalisieren; eine Möglichkeit zu schaffen, genau zu erkennen, wie weit eine Baustelle im Vergleich zu dem Zustand sein müsste, der auf der Baustelle vorzufinden ist.

In diesem Zusammenhang kommt die Forschungsarbeit von Alexander Braun ins Spiel. Dieser forscht im Moment an einem automatisierten Verfahren zur Baufortschrittskontrolle auf Basis der Integration von Punktwolkeninterpretation und 4D Bauwerksmodellierung. Sein Lösungsansatz ist eine automatische Bauüberwachung mit Hilfe von Fotografien, welche in regelmäßigen Abständen von der kompletten Baustelle aufgenommen werden. Anschließend werden diese mit Hilfe einer Punktwolkeninterpretation in ein 3D Modell übersetzt. Hierdurch ist es möglich, einen digitalen Abgleich verschiedener Zustände durchzuführen.

Auf Basis dieser Forschungsidee soll die Bachelorthesis entstehen. Ein 3D Modell einer Baustelle soll in eine Webseite eingebunden werden, um anschließend plattformunabhängig eine Bauüberwachung durchführen zu können.

Bei der Erstellung der Webseite sollen fertige Bibliotheken verwendet werden:

- three.js für die Darstellung des Modells

- Gantt Chart für die Darstellung eines Bauablaufplanes

Zudem soll die Webseite in PHP und JavaScript programmiert werden. Als Datenbanksystem soll ein MySQL Server verwendet werden.

Da die Webseite plattformunabhängig aufrufbar sein soll, wird ebenfalls ein großes Augenmerk auf ein Mobile Responsive Design gelegt.

1.1 Idee und Ziel der Arbeit

Baustellen werden immer größer und umfangreicher beziehungsweise auch unübersichtlicher. Aus diesem Grund ist eine optimale Bauüberwachung notwendig. Bisher wurde der Baufortschritt händisch aufgenommen und anschließend digital festgehalten.

Mit Hilfe dieses neuen Konzeptes soll die Möglichkeit geschaffen werden, die Baufortschritte direkt digital aufzunehmen. Dazu sollen die verschiedenen Bauzustände fotografiert werden und anschließend in ein 3D Modell transformiert werden.

Im Anschluss daran sollen die verschiedenen Zustände farblich markiert in einem Web Browser dargestellt werden. Hierdurch soll es möglich sein, die Unterschiede visuell wahrzunehmen. Ebenfalls ist es so möglich festzustellen, welche Bereiche des Rohbaus bereits fertig gestellt wurden und welche im Verzug liegen.

Die Darstellung soll durch einen Bauablaufplan unterstützt werden, welcher in einer Datenbank abgespeichert sein soll. Dieser wird dem Benutzer die Möglichkeit geben, einen direkten Vergleich zwischen Bauablaufplan und dargestelltem Modell vorzunehmen.

Neben einer Darstellung nach dem Baufortschritt soll eine Suchmaske implementiert werden. Mit dieser wird es dem Benutzer möglich sein, nach bestimmten Bauteilen zu suchen und sich diese in dem Modell darstellen zu lassen.

1.2 Aufbau der Arbeit

Die Arbeit ist in zwei große Teilbereiche aufgeteilt: Einen Theorieteil und einen Teil, in dem es um die explizite Umsetzung der Webseite geht.

In Kapitel 2 geht es zunächst um die Projektgrundlage. Anschließend wird genauer auf das Thema Building Information Modeling eingegangen; daraus resultierend eine kurze Definition zu 4D und IFC. Im nächsten Kapitel geht es um die verschiedenen Arten von Ablaufplänen. Zusätzlich wird in diesem Kapitel auf die Problematik, die Bauablaufpläne mit sich bringen,

eingegangen und auf die möglichen rechtlichen Handlungsmaßnahmen, die ein Auftragnehmer und ein Auftraggeber haben.

Im Kapitel 4 werden die verwendeten Programmiersprachen und Bibliotheken vorgestellt. Überleitend dazu beschreibt das Kapitel 5 die Umsetzung der Webseite. Hier wird auf die verwendete Software und das damit zu programmierende Skript eingegangen.

Den Abschluss bildet das Kapitel 6. Hierbei wird die Arbeit noch einmal zusammengefasst. Des Weiteren gibt es einen Ausblick auf mögliche Veränderungen beziehungsweise Erweiterungen.

Kapitel 2

Baufortschrittskontrolle

2.1 Projektgrundlage

Als Projektgrundlage für die Bachelorthesis dient die Masterarbeit von Alexander Braun. Hierin geht es um die automatische Baufortschrittskontrolle durch eine Integration eines Building Information Modeling (BIM) und einer photogrammetrisch erzeugten Punktwolke.

Eine zu überwachende Baustelle wird dabei mittels photogrammetrischen Aufnahmen digitalisiert, indem aus den Aufnahmen eine digitale Punktwolke erzeugt wird. Eine Punktwolke ist eine Ansammlung von Millionen von Punkten. Jeder Punkt hat seine eigenen Informationen. Wird die Punktwolke ausgewertet, entsteht ein 3D Modell, welches ein nahezu eins zu eins Abbild der Baustelle ist. Hierdurch ist es möglich, das Gebäude mit dem Soll-Zustand aus einem hinterlegten BIM zu vergleichen.

Für die Bachelorthesis wird ein bereits eingelestes Gebäude genutzt. Neben den Gebäudedaten stehen zusätzlich ein Ablaufplan und ein fertiges BIM zu Verfügung. Mit diesen Punkten ist es möglich, eine genaue digitale Bauüberwachung durchzuführen.

2.2 Building Information Modeling

Der Grundgedanke bezüglich BIM besteht bereits seit 1970. Es stellt eine Methode dar, Eigenschaften eines Bauwerks digital darzustellen. Dazu gehören physikalische und funktionale Eigenschaften. Werden all diese Eigenschaften zu einer Datenbank zusammengefasst, kann jedes Bauteil während des gesamten Lebenszyklus eines Gebäudes genau identifiziert werden und mit den hinterlegten Informationen beschrieben werden. Des Weiteren hilft diese Informationsdatenbank, Entscheidungen bezüglich des Bauwerks zu treffen (Egger *et al.*, 2013).

Mit dem Sammeln der Daten soll schon vor dem Beginn des Baus angefangen werden, so die Idee hinter BIM. Mit diesen gesammelten Gebäudedaten ist es einfach, ein Gebäude ohne viel Aufwand in Stand zu halten oder auch am *Lebenszyklusende* einen reibungsfreien Abriss des Gebäudes durchführen zu können.

Aktuell unterstützen zahlreiche Programme diese BIM Informationsdatenbank. Noch nicht so verbreitet ist das Thema BIM in Unternehmen. Um die notwendige Vernetzung zwischen verschiedenen Firmen herzustellen, entwickelt der Verein buildingSMART e.V. bestimmte Konventionen. Diese sind unter der openBIM Methode in Deutschland, Österreich und der Schweiz zusammengefasst. (BuildingSMART, 2015)

2.3 4D

Werden Wände, Decken, Stützen und Ähnliches in einem CAD System mit einer weiteren Dimension versehen, nennt man diese Vorgehensweise Vierdimensional (4D). Im Bauwesen ist die vierte Dimension die Zeit (Braun, 2013). Mit Hilfe dieser Angabe ist es möglich, einen Zusammenhang zwischen Modell und Terminplan zu erstellen. Hierdurch ist es ebenfalls möglich, eine Aussage über einen Soll- und einen Ist-Zustand eines Bauwerks zu treffen.

Zusätzlich zu einer vierten Dimension ist es möglich, weitere Dimensionen einzufügen, die zum Beispiel die Kosten für das Modell beinhalten könnten. Dies ist besonders dann von Vorteil, wenn über die Planung der Logistik gesprochen werden soll.

2.4 IFC

IFC ist die Abkürzung für Industry Foundation Classes. Es stellt einen offenen Standard dar, welcher es ermöglicht, Gebäudemodelle digital zu beschreiben. Herausgeber dieses Standards ist die Organisation buildingSMART International.

Dieser Standard wurde notwendig, da immer öfter Daten zwischen verschiedenen Systemen ausgetauscht werden mussten. In den Anfängen hat man jegliche Änderungen händisch in das System eingepflegt. Da die Gebäude und der Zeitdruck im Laufe der Zeit größer wurden, entwarf die Organisation diesen Standard.

Heute unterstützen zahlreiche Systeme diesen Standard, wodurch der Austausch von Gebäudedaten gravierend erleichtert wurde. Der Standard kommt in verschiedenen Bereichen zur Anwendung, wie zum Beispiel in CAD Systemen, Statik- und Energieberechnungen

oder Mengen- und Kostenermittlungen.

Im Gegensatz zu klassischen CAD Programmen beinhaltet die IFC Datei Informationen über das Gebäude bezüglich Wänden, Fenstern oder Räumen mit ihren jeweiligen Eigenschaften. Klassische CAD Programme exportieren hingegen nur Linien, Punkte oder Blöcke ohne deren explizite Eigenschaften (Sagerschnig, 2006).

IFC ist seit dem Release IFC4 unter der ISO 16739 als internationaler Standard hinterlegt.

Aufbau

Bei IFC muss zwischen der IFC Datei und dem IFC Datenschema unterschieden werden. Die IFC Datei wird als Container für die Übertragung im Step Format verwendet. Das IFC Datenschema definiert die Spezifikationen im EXPRESS Format (BuildingSMART, 2015).

Das IFC Datenschema ist in der Sprache EXPRESS programmiert. EXPRESS ist in der ISO 10303 definiert. Es stellt die Informationen mit Hilfe von Datentypen, Entitäten, Regeln und Beziehungen und weiteren Objekten dar. Demnach ist es möglich, ein IFC Datenschema objektorientiert und damit neutral zu beschreiben. Bei der Weiterentwicklung des IFC Standard geht es besonders darum, weitere Objekte hinzuzufügen und zu aktualisieren. Seit 2014 steht der IFC Standard in der Version 4 zur Verfügung (Amann *et al.*, 2013).

2.5 Zusammenfassung

In diesem Kapitel werden die Grundbegriffe der Baufortschrittskontrolle erläutert. Dabei wird erklärt, was Building Information Modeling und was IFC ist. Um das Gesamtkonzept nachvollziehen zu können, wird ebenfalls auf den Begriff der vierten Dimension eingegangen.

Im nächsten Kapitel wird es um die verschiedenen Bauablaufpläne gehen. Des Weiteren werden die Beziehungen beziehungsweise Abhängigkeiten erklärt, die einzelne Termine und Ereignisse haben.

Kapitel 3

Bauablaufplan

Auf heutigen Baustellen ist ein Bauablaufplan nicht mehr wegzudenken. Er hilft dabei, Bauabläufe zeitlich zu organisieren und wirtschaftlich optimal durchzuführen. Ebenfalls wichtig ist er als Vertragsvereinbarung zwischen zwei Parteien und die daraus resultierenden Fristen. Die DIN 69900 unterscheidet zwischen [Terminliste](#), [Netzplan](#), [Balkendiagramm](#) und [Weg-Zeit-Diagramm](#).

Um den Bauablaufplan möglichst aktuell halten zu können, muss es immer wieder Soll-Ist-Vergleiche geben. Schnell dauern Bauabläufe länger oder auch kürzer, wodurch die Abhängigkeiten einzelner Arbeitsschritte nicht mehr zusammen passen und neu koordiniert werden müssen. Ein Bauablaufplan kann ebenfalls dabei helfen, eine Mengenermittlung durchzuführen: Beispielsweise wie lange ein Kran oder Bauschuttcontainer benötigt wird.

Ein Terminplan wird in der Regel mit Abgabe eines Angebots für einen Auftrag eingereicht. Wichtig für einen Terminplan ist die Plausibilität und die Übersichtlichkeit. Nur so ist es möglich, eine Baustelle wirtschaftlich durchführen zu können. Nicht zu vernachlässigen ist ebenfalls die Aussagekraft und die Vollständigkeit. Damit alle Vorgänge geplant und strukturiert ablaufen, muss zusätzlich auf die Verbindlichkeit eines Terminplanes geachtet werden.

3.1 Arten

3.1.1 Terminliste

Die Terminliste ist eine Auflistung verschiedener Vorgänge, einschließlich Anfang- und Endzeitpunkt. Sortiert werden die Einträge nach dem Datum ihres Startpunktes. Zur besseren Strukturierung können die Einträge nach ihrer Abhängigkeit gebündelt werden. Ein Beispiel für eine Terminliste wird in [Abbildung 3.1](#) gezeigt.

ID	Aufgabenname	Dauer	Anfang	Ende
1	Baustelleneinrichtung	10t	05.05.15	18.05.15
2	Baugrubenaushub	15t	13.05.15	04.06.15
3	Grundleitung	7t	26.05.15	04.06.15
4	Fundamente	10t	28.06.15	09.07.15
5	Rampe Tiefgarage	6t	11.06.15	18.06.15
6	Verfüllen + Bodenplatte	14t	04.06.15	23.06.15
7	Tiefgaragengeschoss	15t	25.06.15	15.07.15

Abbildung 3.1: Terminliste (Zanner, 2014)

3.1.2 Balkendiagramm

In der Baubranche findet das Balkendiagramm oder auch Gantt-Diagramm am häufigsten Verwendung. Es stellt die einzelnen Einträge als Balken dar, wodurch eine sehr gute Übersichtlichkeit und Strukturierung entsteht. Ähnlich wie bei den Terminlisten besteht die Möglichkeit, einzelne Termine zusammenzufassen. Die Gliederung eines Balkendiagramms erfolgt meist in mehreren Ebenen. Dies dient einer besseren Übersichtlichkeit. Unterschieden wird hierbei zwischen Grob- und Feinplanung. Die Grobplanung unterteilt den Bauablaufplan zunächst in Bauabschnitte und in einzelne Geschosse. Die Feinplanung stellt die nächste Unterebene dar. Hier wird meist in die unterschiedlichen Gewerke unterteilt.

Die einzelnen Abläufe werden untereinander gesetzt, die Dauer wird entsprechend der Zeitleiste als Balken neben den jeweiligen Abläufen dargestellt. Die Zeitleiste ist je nach Detaillierungsgrad aufgeteilt; es ist zum Beispiel eine wochenweise oder tageweise Skalierung möglich.

Zusätzlich zu den Balken können markante Zeitpunkte als so genannte Meilensteine in den Plan eingetragen werden. Meilensteine können zum Beispiel verschiedene Vertragstermine sein.

Bei einem Balkendiagramm werden des Weiteren Abhängigkeiten zwischen einzelnen Abläufen markiert. Die Darstellung erfolgt mittels Pfeilen, wobei es hier unterschiedliche Start- beziehungsweise Endbeziehungen gibt (mehr im [Abschnitt 3.3](#)). Ein Beispiel für ein Balkendiagramm ist in [Abbildung 3.2](#) dargestellt.

Für die Erstellung von Balkendiagrammen stehen verschiedene Programme zur Verfügung. Eines der bekanntesten Programme ist das MS Projekt. Hilfreich ist eine Software besonders dann, wenn Veränderungen im Ablauf entstehen. Hierbei berechnet das Programm die Abhängigkeiten neu und erstellt mit den neuen Daten ein neues Balkendiagramm. Weiter unten wird noch einmal genauer auf den Aufbau und die verschiedenen Beziehungen eingegangen.

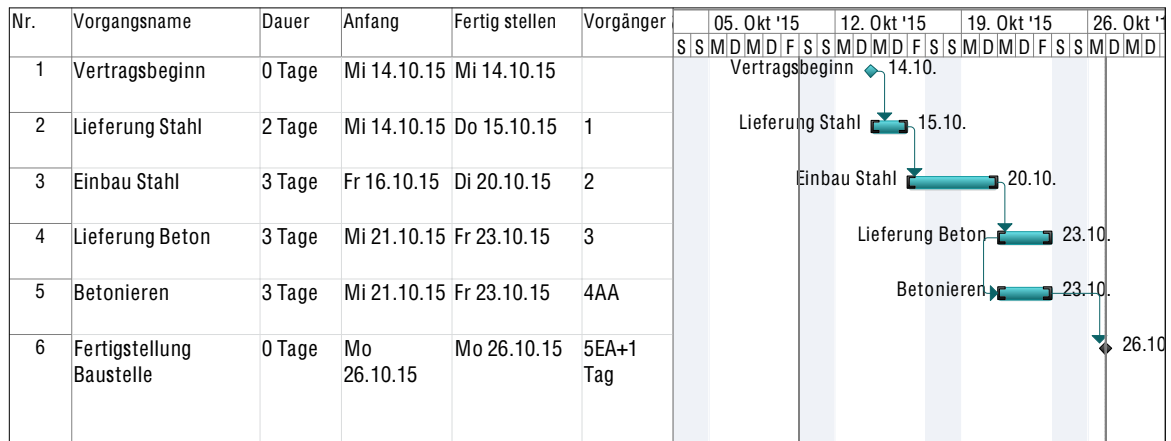


Abbildung 3.2: Balkendiagramm

3.1.3 Weg-Zeit-Diagramm

Für Straßen- und Tunnelbaustellen eignet sich besonders das Weg-Zeit-Diagramm, da hier eine topografische Ansicht mit einer Zeit Achse verknüpft wird. Anders als beim Balkendiagramm wird horizontal ein topografischer Schnitt dargestellt und vertikal die Ausführungszeiten. Die einzelnen Abläufe werden gezielt in dem Diagramm dargestellt, so dass sie einen bestimmten Start-Ort, Start-Termin, End-Ort und End-Termin haben (Zanner, 2014). An Hand dieser Darstellung sind gegebenenfalls auftretende Überschneidungen einzelner Abläufe besonders gut erkennbar. Ein Beispiel für ein Weg-Zeit-Diagramm zeigt die [Abbildung 3.3](#).

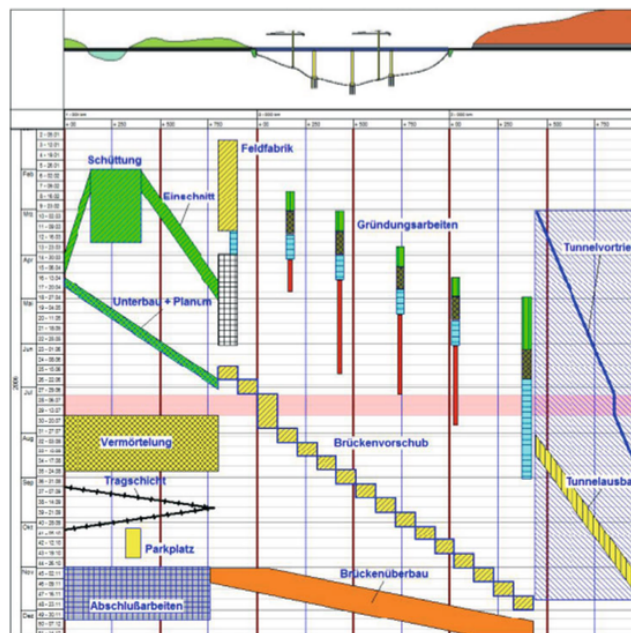


Abbildung 3.3: Weg-Zeit-Diagramm (Zanner, 2014)

3.1.4 Netzplan

Die Idee hinter einem Netzplan ist die optimale mathematische Berechnung der Abläufe. Der Netzplan resultiert aus dem Bereich der Graphentheorie. Markant für den Netzplan sind die Anordnungsbeziehungen zwischen den einzelnen Punkten und die Möglichkeit, Abhängigkeiten und jegliche Beziehungen darzustellen.

Einzelne Abhängigkeiten beziehungsweise Beziehungen werden mittels Pfeilen dargestellt. Anders als beim Balkenplan werden diese Beziehungen mathematisch berechnet, um somit das Optimum berechnen und gleichzeitig den frühesten und spätesten Anfangs und Endzeitpunkt definieren zu können. Zusätzlich wird durch die Berechnung eine Pufferzeit mit angegeben.

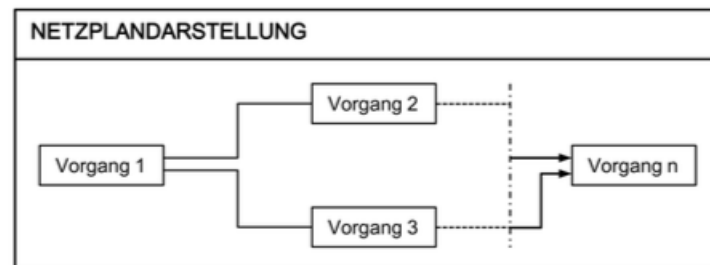


Abbildung 3.4: Netzplan (Zanner, 2014)

3.2 Ressourcen

Mittels dieser verschiedenen Terminpläne ist es möglich, die Ressourcen auf einer Baustelle optimal zu planen und zu steuern. Neben den Bauabläufen können mit diesen ebenfalls die Einsatzplanungen von Geräten und Personal gesteuert werden: Wie viel Personal ist zu welchem Zeitpunkt notwendig; wie lange muss ein Kran an einer bestimmten Stelle stehen; wie ist er ausgelastet. Hieran ist erkennbar, wie vielfältig die Möglichkeiten von Terminplänen sind. Da es in dieser Arbeit hauptsächlich um die Steuerung und Überwachung von Bauabläufen geht, wird der Schwerpunkt auf bauliche Terminpläne gelegt.

3.3 Beziehungen

Wie bereits weiter oben erwähnt, können zwischen den einzelnen Abläufen Beziehungen bestehen. Diese sind notwendig, um eine gezielte Abfolge der Ereignisse zu sichern. Je nachdem welchen Zustand der nächste Vorgang voraussetzt, gibt es unterschiedliche Arten von Beziehungen. Die wichtigsten Beziehungen beziehungsweise Folgen sind:

- Normalfolge
- Anfangsfolge
- Endfolge
- Sprungfolge

Ein Überblick über die wichtigsten Beziehungen zeigt ebenfalls die [Abbildung 3.5](#).

Die Normalfolge beschreibt die Abfolge, bei der der nächste Vorgang direkt oder auch mit einem zeitlichen Abstand an das Ende des vorherigen Vorgangs anschließt. Bei der Anfangsfolge wird der Anfang eines Vorgangs mit dem Anfang des nächsten Vorgangs verbunden. Die Endfolge ist mit der Anfangsfolge vergleichbar, nur dass hier die beiden Enden miteinander verknüpft werden. Die Sprungfolge beschreibt die Beziehung, bei der der Anfang mit dem Ende des nächsten Vorgangs verbunden wird. Ähnlich wie bei der Normalfolge bestehen bei der Anfangsfolge, Endfolge und der Sprungfolge auch die Möglichkeit, einen zeitlichen Abstand einzufügen.

Darstellung im Verknüpfung	Balkenplan	Volumen-Zeit-Diagramm	Netzplan
Normalfolge mit Ende-Anfangs- Beziehung			
Normalfolge Ende-Anfangs- Beziehung mit Wartezeit			
Anfangsfolge Anfangs-Anfangs- Beziehung			
Endfolge End-End-Beziehung			
Sprungfolge Anfangs-Ende- Beziehung			

Abbildung 3.5: Anordnungsbeziehungen (Zanner, 2014)

3.3.1 Pufferzeiten

In dem Kapitel Netzplan wurde bereits kurz auf die Pufferzeit eingegangen. Die DIN 69900 beschreibt die Pufferzeit wie folgt:

„Zeitspanne, um die, unter bestimmten Bedingungen, die Lage eines Ereignisses bzw. Vorgangs verändert oder die Dauer eines Vorgangs verlängert werden kann.“ (Deutsches Institut für Normung & Ausschuss Netzplantechnik und Projektmanagement, 1980)

Eine Pufferzeit ist besonders dann von Vorteil, wenn ein Vorgang nicht bis zum geplanten Endzeitpunkt fertig gestellt werden kann. Ohne Pufferzeit würden sich jegliche Vorgänge, die von dem nicht rechtzeitig fertig gestellten Vorgang abhängen, verschieben. Bereits bei der Planung eines Terminplanes sollten somit Puffer eingeplant werden. Zusätzliches Argument für die Einplanung von Puffern sind die hohen Kosten bei einem Bauverzug.

3.4 Probleme und Möglichkeiten

3.4.1 Abweichungen im Terminplan

Während eines Bauablaufes kann es sehr schnell zu Abweichungen im Terminplan kommen. Dies kann sowohl durch eine Verzögerung im Bauablauf passieren als auch durch Arbeiten, die schneller als vorgesehen erledigt werden.

Das Vorgehen bei einer Bauzeitverlängerung wird in der VOB/B geregelt. Hierbei muss unterschieden werden, von wem die Verzögerung beziehungsweise die Behinderung der Arbeit ausgeht. Auf der einen Seite kann sie durch mangelhaftes Handeln des Auftragnehmers entstehen, auf der anderen Seite kann eine solche jedoch auch durch den Auftraggeber entstehen. Des Weiteren muss zwischen den Ansprüchen des Auftraggebers und des Auftragnehmers unterschieden werden.

3.4.2 Verzögerung des Bauablaufs

Für einen Verzug im Bauablauf gibt es mehrere Gründe. Diese sind zum Beispiel:

- Verzug durch schlechte Witterungsverhältnisse
- Lieferschwierigkeiten
- Unvorhergesehene Ereignisse

Die Folgen können, je nachdem wie eng der Terminplan gestaltet wurde, gravierend sein. Wurde, wie weiter oben beschrieben, kein Puffer eingeplant, verzögern und verschieben sich alle folgenden Bauabläufe. Wurde hingegen ein Puffer eingeplant, verzögern sich gegebenenfalls nur die Bauabläufe bis zum nächsten Puffer. Für einen reibungslosen Ablauf ist es demnach unbedingt notwendig, den Terminplan immer aktuell zu halten. Bei einem Verzug ist es jedoch auch möglich, die Bauablaufgeschwindigkeit zu erhöhen. Durch ein solches Vorgehen kann versucht werden, die zuvor entstandene Verzögerung wieder einzuholen.

3.4.3 Verkürzung des Bauablaufs

Anders als bei einer Verzögerung ist eine Verkürzung wünschenswert, da jeder Verzug mit hohen Kosten verbunden ist. Verkürzungen können dadurch entstehen, dass Bauabläufe schneller fertiggestellt werden als geplant. Auch hier ist es sinnvoll, den Terminplan immer aktuell zu halten, da ein Gewerk, welches seine Arbeiten eigentlich erst später durchführen sollte, gegebenenfalls schon eher seine Arbeit aufnehmen kann.

3.4.4 Soll-Ist-Vergleich

Um den Bauablaufplan immer aktuell halten zu können, müssen regelmäßig Soll-Ist-Vergleiche stattfinden. Das heißt, es wird überprüft, wie weit die Arbeiten auf der Baustelle fortgeschritten sind und wie weit sie laut Terminplan sein sollten. Je nach Häufigkeit der Kontrollen ist eine Abweichung schnell erkennbar und kann behoben werden.

3.4.5 Rechtliche Schritte bei Abweichungen vom Terminplan

Ansprüche des Auftragnehmers

Wie bereits weiter oben erwähnt werden die rechtlichen Möglichkeiten in der VOB/B beschrieben. In § 6 VOB/B werden die Ansprüche des Auftragnehmers beschrieben, die aus einer Behinderung oder Unterbrechung resultieren. Der Auftragnehmer hat verschiedene rechtliche Möglichkeiten: Anspruch auf Bauzeitverlängerung, Anspruch auf Schadensersatz, Anspruch auf Entschädigung, Anspruch auf vorzeitige Abrechnung der erbrachten Leistungen und den Anspruch auf Kündigung. Um diese Ansprüche geltend machen zu können, muss ein Behinderungstatbestand, der nicht vorhersehbar war, vorliegen. Des Weiteren muss eine Behinderungsanzeige an den Auftraggeber erfolgen, in der erkennbar ist, dass der Auftragnehmer in seinem Handeln beeinträchtigt wurde.

Anspruch auf Bauzeitverlängerung Für einen Anspruch auf Bauzeitverlängerung muss, laut § 6 Abs. 2 VOB/B, eine Behinderung vorliegen, die durch einen Umstand aus dem

Risikobereich des Auftraggebers, durch einen Streik oder durch höhere Gewalt, welche für den Auftragnehmer nicht vorhersehbar ist, entstanden ist (§ 6 Abs. 2 Nr. 1 VOB/B). Dabei ist zu berücksichtigen, dass Witterungseinflüsse, die für den Auftragnehmer absehbar waren, nicht als höhere Gewalt gelten.

Anspruch auf Schadensersatz Um einen Anspruch auf Schadensersatz geltend machen zu können, müssen Gegebenheiten vorliegen, die die Ausführung der Leistung negativ beeinflussen. Zudem dürfen diese dem Auftragnehmer bei Abgabe des Angebots nicht bekannt sein (§ 6 Abs. 6 VOB/B).

Anspruch auf Entschädigung Einen Anspruch auf Entschädigung kann der Auftragnehmer dann stellen, wenn die hindernden Umstände durch noch nicht fertiggestellte Leistungen, die aus dem Bereich des Vorunternehmers stammen, resultieren. Der BGH hat dazu entschieden, dass der Auftragnehmer in solchen Fällen die zusätzlich entstandenen Kosten als Entschädigung einfordern kann, nicht aber als Schadensersatz (§ 6 Abs. 6 Satz 2 VOB/B).

Anspruch auf vorzeitige Abrechnung Müssen die Arbeiten auf einer Baustelle für eine längere Zeit unterbrochen werden, kann der Auftragnehmer für die bereits ausgeführten Leistungen und entstandenen Kosten eine vorzeitige Abrechnung fordern (§ 6 Abs. 5 VOB/B).

Anspruch auf Kündigung des Vertrages Der Auftragnehmer hat die Möglichkeit, den Vertrag mit dem Auftraggeber zu kündigen, wenn dieser Handlungen unterlässt, durch welche der Auftragnehmer seine Arbeit nicht fortführen kann. Diese sind zum Beispiel eine fehlende Baugenehmigung, fehlende Baufreiheit oder eine fehlende Ausführungsplanung. Bevor der Auftragnehmer einen Vertrag kündigen kann, muss er dem Auftraggeber eine angemessene Frist setzen, in welcher die vertraglichen Vereinbarungen erfüllt werden können. Werden die Arbeiten an der Baustelle sowohl auf Seiten des Auftragnehmers als auch auf Seiten des Auftraggebers für mehr als 3 Monate unterbrochen, haben beide Seiten die Möglichkeit, den Vertrag außerordentlich zu kündigen.

In § 2 VOB/B werden die Ansprüche des Auftragnehmers beschrieben, die aus Anordnungen des Auftraggebers resultieren. Im Laufe einer Baustelle ist es nicht unüblich, dass es zusätzlich zum Bauvertrag Nachträge gibt. Die Ansprüche des Auftragnehmers, die aus den durch den Auftraggeber gestellten Nachträgen entstehen, werden in diesem Paragraphen beschrieben. Diese sind zum Beispiel Vergütungsanpassungen oder Vergütung auf Grund einer Bauzeitbeschleunigung.

Ansprüche des Auftraggebers

In § 6 VOB/B werden neben den Ansprüchen des Auftragnehmers auch die Ansprüche des Auftraggebers bezüglich Bauablaufstörungen geregelt. Entstehen Verzögerungen oder Mängel durch das Handeln des Auftragnehmers, hat der Auftraggeber ein außerordentliches Kündigungsrecht und zusätzlich die Möglichkeit, eine Vertragsstrafe geltend machen zu können. Um eine Vertragsstrafe geltend machen zu können, muss diese in dem Vertrag vereinbart sein (Zanner, 2014).

3.5 Zusammenfassung

In diesem Kapitel werden die verschiedenen Arten von Terminplänen beschrieben. Dabei wird genauer darauf eingegangen, an Hand welcher Merkmale die verschiedenen Pläne zu unterscheiden sind.

Da es bei Terminplänen oft Abhängigkeiten zwischen verschiedenen Terminen beziehungsweise Ereignissen gibt, wird in diesem Kapitel ebenfalls erklärt, welche Möglichkeiten es gibt, Termine und Ereignisse zu verbinden. Für den Fall, dass Abhängigkeiten zu eng gesteckt werden, werden ebenfalls die rechtlichen Möglichkeiten eines Auftragnehmers und eines Auftraggebers geschildert.

Im nächsten Kapitel wird es um die Theorie gehen, die für die Umsetzung der Webseite wichtig sein wird. Dabei werden die wichtigsten Programmiersprachen, die für die Erstellung von Webseiten genutzt werden, vorgestellt und erklärt. Für ein genaueres Verständnis wird ebenfalls ein kurzes Skript Beispiel gezeigt.

Kapitel 4

Theorie

Bei der Programmierung der webbasierten Darstellung werden verschiedene Programmiersprachen verwendet, die im Folgenden beschrieben werden. Die hier verwendeten Sprachen sind darüber hinaus die gängigsten Programmiersprachen für die Umsetzung von Webseiten. Nach jeder Beschreibung der Programmiersprache wird ein kurzes Skript beispielhaft wiedergegeben.

Neben den Programmiersprachen werden zusätzlich das Datenbanksystem MySQL und die Bibliothek three.js erklärt. Zusätzlich wird der Begriff STL, der für die Bibliothek three.js wichtig ist, erklärt.

Den Abschluss des Kapitels bildet das Thema Mobile Responsive Design, das für die Darstellung der Webseite sehr wichtig ist.

4.1 HTML

Hypertext Markup Language ([HTML](#)) wird für die Erstellung von Webseiten verwendet. Hiermit lassen sich einfache Webseiten erstellen. Die aktuelle Version ist HTML5. HTML wurde dafür entwickelt, Texte zu gliedern. Die visuelle Darstellung wird dabei nicht von HTML sondern von Cascading Stylesheets ([CSS](#)) (siehe [Abschnitt 4.2](#)) übernommen. Der Aufbau einer HTML Seite gliedert sich in zwei Bereiche: Head und Body. Im Head sind jegliche Befehle enthalten, die der Darstellung dienen. Im Body sind alle Befehle enthalten, die auf der Webseite angezeigt werden. In Webseiten werden diese beiden Bereiche von einem `<HTML>` Tag eingeschlossen, siehe HTML Beispiel.

Um Webseiten, die in HTML programmiert sind, darstellen zu können, müssen diese auf einem Webserver gespeichert werden. Anschließend sind sie von jedem Webbrowser aus auf-

rufbar. Da sich mit HTML nur statische Inhalte erzeugen lassen, muss man auf PHP Hypertext Preprocessor ([PHP](#)) (siehe [Abschnitt 4.3](#)) zurückgreifen, wenn dynamische Inhalte erstellt werden sollen.

HTML Beispiel

```
1 <html>
2     <head>
3         <meta charset = "utf8">
4         <title>Skript Bsp</title>
5     </head>
6     <body>
7         <p>Dies ist ein Skript Beispiel in HTML</p>
8     </body>
9 </html>
```

Die Ausgabe des Skript Beispiels lautet: Dies ist ein Skript Beispiel in HTML. Im Folgenden wird beschrieben, wie der eben ausgegebene Text formatiert werden kann.

4.2 CSS

Möchte man einen HTML Text gezielt formatieren, benötigt man, wie bereits unter HTML genannt, [CSS](#). Diese Form der Programmierung wurden explizit für HTML entwickelt und ist eine Formatierungssprache für HTML und XML Dokumente.

Um eine Webseite mit CSS formatieren zu können, gibt es zwei Varianten. Bei der ersten Variante werden die Formatierungsanweisungen direkt in den *HEAD* der Webseite geschrieben. Bei der zweiten Variante wird eine CSS Datei in die Webseite eingebunden. Vorteil hierbei ist, dass zwischen mehreren Fällen unterschieden werden kann. Es kann zum Beispiel unterschieden werden, welche Stylesheets bei der Druckansicht oder welche bei einer bestimmten Ansicht verwendet werden sollen. Bei umfangreicheren Webseiten wird die zweite Variante verwendet, da die CSS Verwaltung zentral in einer Datei passieren kann. Aktuell steht CSS in Version 3 zu Verfügung.

Anwendung

Soll zum Beispiel ein `< p >` Tag formatiert werden, gibt es zwei Methoden. In der einen Variante werden alle in einer Webseite vorkommenden `< p >` Tags formatiert:

CSS Beispiel 1

```
1 p {  
2     color: black;  
3     font-size: medium;  
4 }
```

Bei der zweiten Methode muss der `< p >` Tag mit einer Klasse erweitert werden: `< p class=Format1>`. In der CSS Datei wird dieser Tag wie folgt formatiert:

CSS Beispiel 2

```
1 .Format1 {  
2     color: black;  
3     font-size: medium;  
4 }
```

Bei der zweite Variante werden nur die HTML Tags formatiert, welche die Klasse `Format1` tragen. Alle anderen bleiben unangetastet.

4.3 PHP

[PHP](#) ist eine serverseitig interpretierte Skriptsprache, die hauptsächlich für die Erstellung von dynamischen Webseiten verwendet wird. Der Vorteil gegenüber HTML besteht darin, dass dem Programmierer viel mehr Möglichkeiten für die Umsetzung und Gestaltung gegeben sind. Besonders ist die Möglichkeit der Verwendung von Variablen hervorzuheben. Hierdurch können Formulare, Eingabefelder und verschiedenes mehr mittels Variablen vielseitig weiterverarbeitet werden.

Weiterhin zeichnet sich PHP nicht nur dadurch aus, dass mit Variablen gearbeitet werden kann, sondern auch die sehr gute Anbindung an Datenbanksysteme, wie zum Beispiel MySQL. Dies ist insbesondere dann hilfreich, wenn Daten, ob durch einen Dateiimport oder durch Formulare, abgespeichert werden sollen, um diese später individuell nutzen zu können. Eine weitere Möglichkeit ist es, Daten in eine Datei zu speichern und diese durch PHP auslesen zu lassen.

Anders als bei HTML, CSS und JavaScript ist der Quellcode von PHP nicht auslesbar. Dies ist besonders für die Sicherheit von Webseiten, die in PHP programmiert werden, von großer Bedeutung (Theis, 2014).

PHP schließt HTML Befehle nicht aus. Vielmehr ist das Zusammenspiel von beiden Programmiersprachen von großem Interesse. PHP übernimmt bei der Erstellung der Webseite

zum Beispiel die Umsetzung mittels Variablen und deren Funktionen und HTML die Gestaltung. Da die Programmiersprache PHP einfach zu erlernen ist, lassen sich so umfangreiche Webseiten erstellen.

Die Erweiterung für PHP lässt sich mit wenigen Kenntnissen im Bereich Serververwaltung sehr einfach installieren. Da es kostenlos und zudem OpenSource ist, kann es von der Webseite (<http://www.php.net>) heruntergeladen werden und in den Server integriert werden. Voraussetzung dafür ist ein vorhandener Webserver. Dieser kann zum Beispiel Apache oder *nginx* sein.

Um PHP immer auf dem neusten Stand zu halten, wird die Programmiersprache kontinuierlich weiter entwickelt. Hierbei geht es primär nicht nur um die Weiterentwicklung, sondern viel mehr um die Behebung von möglichen Sicherheitslücken und um PHP noch sicherer zu machen.

Aktuell kann PHP mit der Version 5.6.8 auf einem Webserver genutzt werden. Viele Firmen führen das Upgrade nicht direkt auf die neueste Version von PHP durch, da es vorkommen kann, dass die alte Syntax mit der neuesten Version von PHP nicht mehr lauffähig ist. Dies hätte zur Folge, dass Webseiten komplett umgeschrieben werden müssten, um sie auf den aktuellen Stand zu bringen und somit wieder lauffähig zu machen.

Auf dem Leibniz Rechenzentrum ([LRZ](#)) Server, auf dem die Projektdaten der Bachelorthesis liegen, wird aktuell PHP mit der Version 5.5.30 verwendet (Stand 13. Oktober 2015).

Im Folgenden sind die wichtigsten Funktionen von PHP noch einmal zusammengefasst:

- Verwendung von Variablen
- Quelltext wird nicht übermittelt
- Sehr gute Datenbankanbindung
- Zahlreiche Möglichkeiten für die Verarbeitung von externen Dateien

Aufbau

Der Aufbau einer PHP Datei ist wie folgt: Sie beginnt immer mit `<?php` und endet mit `? >`. Jeder Befehl muss mit einem Semikolon beendet werden. Um einen Text auszugeben, benutzt man den Befehl *echo*. Variable werden wie im Beispiel unten definiert. Zwei Variable können mit einem Punkt miteinander verbunden werden. Im Folgenden wird ein kurzes Skript Beispiel wiedergegeben. Die Ausgabe des Beispiels wäre wie folgt: Dies ist ein PHP Beispiel.

PHP Beispiel

```
1 <?php
2     $variable = 'PHP Beispiel';
3     echo 'Dies ist ein '.$variable;
4 ?>
```

4.4 MySQL

MySQL ist, wie oben bereits erwähnt, ein Datenbanksystem. Es ist einerseits OpenSource, andererseits eines der bekanntesten Systeme für Datenbanken im Zusammenspiel mit Webseiten. Weitere Datenbanksysteme sind zum Beispiel Microsoft Access oder Oracle.

MySQL steht aktuell mit der Version 5.6.25 zur Verfügung. Ähnlich wie bei der Programmiersprache PHP wird MySQL immer weiter entwickelt. Auch hier wird besonders auf die Sicherheit geachtet. Auf dem [LRZ](#) Server steht aktuell MySQL mit der Version 5.0.11 zur Verfügung.

Als Schnittstelle zwischen MySQL und PHP wird die Erweiterung MySQLi verwendet. Diese steht seit MySQL Version 4.1 und PHP Version 5 zur Verfügung. Besondere Eigenschaften von MySQLi sind:

- Möglichkeit objektorientierte Abfragen durchzuführen
- Möglichkeit prozessorientierte Abfragen durchzuführen
- Support for Prepared Statements
- Support for Multiple Statements

Eine Datenbank besteht meistens aus einer Vielzahl von Tabellen und dient der Speicherung von größeren Datenmengen. Für jede Tabelle legt der Programmierer eine Kopfzeile an, in der jede Spalte explizit mit Name und Datentyp des Inhaltes beschrieben wird. Die gängigsten Typen von Inhalten sind Text, Zahlen und Datum.

Um eine Datenbank mit PHP abfragen zu können, muss zunächst eine Verbindung zum Datenbank Server hergestellt werden. Anschließend kann ein SQL Befehl auf dem Datenbank Server ausgeführt werden. Die Standardbefehle sind:

- SELECT * FROM [Tabellennamen] WHERE [Bedingungen] (Ausgabe von Inhalten einer Datenbank)

- INSERT INTO [Tabellenname] (column1, column2 ...) VALUES (value1, value2 ...)
(Speichern von neuen Inhalten in die Datenbank)
- UPDATE [Tabellenname] SET column1=value1, column2=value2 WHERE [Bedingung] (Aktualisieren beziehungsweise überschreiben eines vorhandenen Eintrags in der Tabelle)
- DELETE FROM [Tabellenname] WHERE [Bedingung] (Löschen eines vorhandenen Eintrags in der Tabelle)

Jeder Befehl muss mit einem `mysql_query()` ausgeführt werden. Sind alle Abfragen durchgeführt, wird die Verbindung zum Datenbank Server mit einem `mysql_close()` geschlossen. Jede Abfrage muss mit einem Semikolon beendet werden.

Je umfangreicher das Webprojekt wird, um so umfangreicher und komplexer werden die SQL Befehle, da immer komplexere Bedingungen bei Abfragen eingehalten werden müssen. Neben den Abfragen einer einzelnen Tabelle besteht die Möglichkeit, mehrere Tabellen zu verbinden. Hierdurch ist es möglich, die einzelnen Tabellen übersichtlicher zu gestalten (Theis, 2014).

4.5 JavaScript

JavaScript ist - ähnlich wie PHP - eine Programmiersprache, die mit HTML agiert. Auch vom Funktionsumfang her ist JavaScript PHP ähnlich. Ein gravierender Unterschied zu PHP besteht darin, dass die Programme beziehungsweise Befehle im Web Browser des Nutzers ausgeführt werden (Theis, 2013). Bei der Programmiersprache PHP hingegen werden die Programme serverseitig verarbeitet. Ähnlich wie PHP kann JavaScript Inhalte dynamisch verändern. Zum Beispiel kann sich die Darstellung einer Webseite automatisch ändern, wenn der Browser erkennt, dass die Webseite mit einem Smartphone aufgerufen wird.

Um JavaScript im Browser nutzen zu können, bedarf es der Aktivierung durch den Nutzer. In der Regel ist JavaScript im Browser automatisch aktiviert, kann aber durch den Nutzer deaktiviert werden. Grund hierfür kann ein Sicherheitsaspekt sein, da Programme, die in JavaScript programmiert sind, im Hintergrund ablaufen können, ohne dass der Nutzer etwas bemerkt. Hierdurch kann zum Beispiel Schadsoftware auf den PC heruntergeladen werden, ohne dass der Nutzer der Webseite etwas davon mitbekommt.

Anders als bei PHP können JavaScript Befehle im Quelltext ausgelesen werden. Im Folgenden sind die Funktionen von JavaScript noch einmal zusammengefasst:

- Befehle werden im Browser des Nutzers ausgeführt

- Webseiten können durch JavaScript dynamisch verändert werden, ohne neu aufgerufen werden zu müssen

Aufbau

Ein JavaScript Programm beginnt immer mit `<script type = 'text/javascript'>` und endet mit `</script>`. Dieses kann sowohl in den *HEAD* oder in den *BODY* einer HTML Seite geschrieben werden.

JavaScript hat - ähnlich wie PHP - eine große Anzahl an Befehlen. Ein einfaches Skript Beispiel könnte so aussehen:

JavaScript Beispiel

```
1 <HTML>
2 <BODY>
3   <script type = 'text/javascript' >
4     document.write('<p>Dies ist ein Script Beispiel</p>');
5   </script >
6 </BODY>
7 </HTML>
```

Die Ausgabe des Beispiels ist wie folgt: Dies ist ein Script Beispiel. Ähnlich wie bei PHP muss auch bei JavaScript jeder Befehl mit einem Semikolon beendet werden.

4.6 WebGL / three.js

Web Graphics Library ([WebGL](#)) ist ein Bestandteil einiger Browser und wird von der Firma Khronos Group entwickelt. Damit ist es möglich, 3D Grafiken im Browser darzustellen, ohne zusätzliche Software herunterladen zu müssen. Es basiert auf OpenGL ES 2.0 (Khronos, 2015).

Es gibt verschiedene Möglichkeiten, um 3D Inhalte zu erstellen. Die einfachste Möglichkeit ist die Benutzung der Bibliothek von three.js. Ziel dieser ist es, 3D Modelle mit wenig Arbeitsaufwand durch einen Programmierer darzustellen. three.js stellt mehrere Schnittstellen zur Verfügung, um Modelle aus anderen Programmen darzustellen, zum Beispiel Modelle mit dem Dateityp STL (Three.js, 2015).

Im Kapitel 5.8 wird noch einmal genauer auf die Kompatibilität verschiedener Browser und Betriebssysteme eingegangen.

4.7 STL

StereoLithography (**STL**) ist ein Dateiformat, das von vielen 3D Programmen genutzt wird. Die Datei beinhaltet jegliche geometrischen Informationen, die zur Darstellung eines 3D Modells notwendig sind. Dabei werden die Flächen des Modells in eine Vielzahl von Dreiecken zerlegt. In der Datei sind die Koordinaten von Dreiecken in 3D angegeben. Neben den Koordinaten wird die Flächennormale zu jedem einzelnen Dreieck ebenfalls in der Datei mit gespeichert. Beim Code einer STL Datei unterscheidet man zusätzlich zwischen dem ASCII Format und dem Binärformat. Der Aufbau einer Datei im ASCII Format ist im folgenden Beispiel dargestellt.

STL File

```
1 solid ascii
2   facet normal 0.0000000000000000 129.4754041492420000 0.0000000000000000
3     outer loop
4       vertex 8.6847210571289100 -7.4750745437622100
5         0.0000000000000000
6       vertex 8.6847210571289100 -7.4750745437622100
7         3.2099999954223600
8       vertex 12.4319667480469000 -7.4750745437622100
9         3.2099999954223600
10    endloop
11  endfacet
12 endsolid ascii
```

Der Vorteil einer Datei im Binärformat liegt darin, dass die Größe der Datei sehr viel geringer ist. Anders als im ASCII Format wird jedes Dreieck durch zwölf Gleitkommazahlen und zwei Integerzahlen dargestellt. Die ersten drei Ziffern entsprechen der Normalen, die folgenden drei den jeweiligen Koordinaten der Eckpunkte des Dreiecks und zuletzt zwei Stellen, die einen vorzeichenlosen Integer darstellen (Ennex Corporation, 1999). Der Aufbau einer Datei im Binärformat ist in der [Abbildung 4.1](#) dargestellt.

4	float	<i>i</i> for normal
4	float	<i>j</i>
4	float	<i>k</i>
4	float	<i>x</i> for vertex 1
4	float	<i>y</i>
4	float	<i>z</i>
4	float	<i>x</i> for vertex 2
4	float	<i>y</i>
4	float	<i>z</i>
4	float	<i>x</i> for vertex 3
4	float	<i>y</i>
4	float	<i>z</i>
2	unsigned integer	Attribute byte count

Abbildung 4.1: STL Binary Format (Ennec Corporation, 1999)

Wie weiter oben beschrieben, wird sich die Triangulierung im STL Format zu Nutze gemacht. Hierbei werden große Flächen in eine Vielzahl von Dreiecken zerlegt. Dies ist notwendig, da runde Formen sonst nicht darstellbar wären. Ein Beispiel ist in [Abbildung 4.2](#) dargestellt. In dieser Abbildung wird die Aufteilung von runden Flächen und eines Rechtecks dargestellt. Dabei ist zu erkennen, dass je höher die Kantenanzahl ist, umso mehr eine Kreisform zu erkennen ist. Dieses Vorgehen lässt sich auch auf Flächen übertragen. Je feiner beziehungsweise kleiner die Dreiecke sind, um so runder wird die darzustellende Form. Demnach können runde Flächen oder auch Kreise nur annäherungsweise rund dargestellt werden.

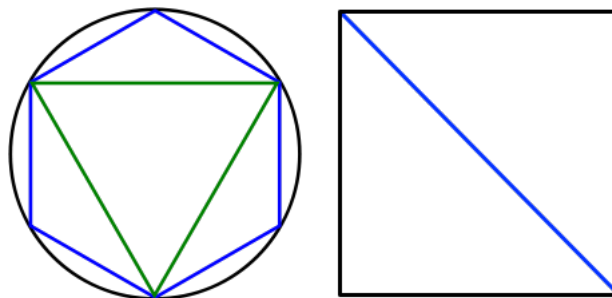


Abbildung 4.2: Triangulierung

4.8 Mobile Responsive Design

In der heutigen Zeit ist ein Smartphone oder Tablet nicht mehr wegzudenken. Um so größer ist das Interesse der Programmierer, Webseiten so zu erstellen, dass sie von jedem Endgerät aus ohne Probleme darzustellen sind.

Mittels JavaScript ist es möglich zu erkennen, mit welchem Browser und Endgerät der Nutzer die Webseite aufruft. Je nach Ergebnis wird die Darstellung der Webseite so automatisch angepasst, dass die Webseite ohne Einschränkungen angezeigt wird.

Eine der bekanntesten Webseiten für Responsive Design ist Bootstrap. Ursprünglich wurde Bootstrap für Twitter entwickelt. Hiermit ist es möglich, Webseiten ohne viel Aufwand so zu erstellen, dass sie für mobile Endgeräte geeignet sind. Um dies zu ermöglichen, setzen die Entwickler von Bootstrap auf bestimmte Bibliotheken und Stylesheets, die OpenSource und damit frei verfügbar sind. Beim Programmieren mit Stylesheets und Bibliotheken von Bootstrap, lässt sich die Darstellung explizit einstellen. So lässt sich zum Beispiel festlegen, ab welcher Darstellungsbreite eine dreispaltige Tabelle statt nebeneinander untereinander angezeigt werden soll. Aktuell steht Bootstrap in Version 3.3.5. zur Verfügung, wird aber immer weiter entwickelt (Bootstrap, 2015).

Da Bootstrap vorgefertigte Skriptbausteine zur Verfügung stellt, ist eine Webseite ohne viel Aufwand schnell erstellt. Es werden zum Beispiel Bausteine für ein Menü und eine fertig formatierte Tabelle zur Verfügung gestellt. Von Haus aus wird von Bootstrap eine Designvorlage bereitgestellt. Diese lässt sich aber auf der Webseite von Bootstrap nach eigenen Bedürfnissen anpassen. Anschließend kann man diese herunterladen und sie in das eigene System einbinden.

4.9 Zusammenfassung

Bei der Programmierung von Web Applikationen kommen verschiedene Programmiersprachen zum Einsatz. Dabei wird genauer auf HTML, CSS, PHP, MySQL und JavaScript eingegangen. Ebenfalls wird die fertige Bibliothek three.js erläutert, die dabei hilft, das 3D Modell im Browser darzustellen. Da, wie weiter oben bereits genannt, ein großes Augenmerk auf die Darstellung gelegt wird, wird Bootstrap beschrieben und dessen Entwicklung erklärt. Dabei wird ebenfalls darauf eingegangen, wie diese bei der Programmierung eingesetzt werden kann.

Dieses Kapitel soll einen kurzen Überblick geben und nicht als Anleitung für das Programmieren dienen. Explizite Anleitungen und Herangehensweisen sind der Fachliteratur zu entnehmen.

Im folgenden Kapitel wird es um die gezielte Umsetzung dieser Programmiersprachen für die Web Applikation gehen, die im Rahmen der Bachelorthesis programmiert wird.

Kapitel 5

Umsetzung

5.1 Allgemein

Ziel der Bachelorthesis ist es, die geographischen Informationen eines 4D Modells aus einer Datenbank zu exportieren und anschließend im Browser darzustellen. Der Anspruch an den Aufbau des Projektes ist es, die Webseite so zu programmieren, dass sie jegliche 4D Modelle verarbeiten kann. Hierfür wird besonders auf den Aufbau der Datenbank geachtet. Aus den genannten Punkten ist es ebenfalls für die Umsetzung wichtig, den Quelltext möglichst allgemein zu halten.

Neben dem Anspruch mehrere Projektdateien anzeigen zu können, ist es wichtig, die Webseite im Mobile Responsive Design zu programmieren. Die Webseite soll von jedem Endgerät aus einwandfrei darstellbar sein. Trotz des Mobile Responsive Designs gibt es auf Smartphones und Tablets minimale Funktionseinbußen.

5.1.1 Projektdatei

Als Projektgrundlage dient zunächst ein mehrstöckiges Haus im innerstädtischen Bereich von München, das bereits in einer vorherigen Arbeit digitalisiert wurde. Während der Bauphase wurden regelmäßig Bilder von der Baustelle gemacht, um eine vollständige Bilddokumentation zu erhalten. Mit Hilfe dieser Bilddokumentation kann mit wenigen Abweichungen der genaue Bauzustand zu einer beliebigen Zeit ermittelt werden. Dies ist für diese Bachelorthesis von besonderer Bedeutung, da hierdurch ein Abgleich des Bauablaufplanes mit dem vorgefundenen Bauzustand stattfinden kann. Wichtig für die Darstellung ist ebenfalls ein fertiges CAD Modell, das als STL Datei zur Verfügung gestellt wurde.

5.2 Software und Hardware

Bei der Programmierung kommen verschiedene Programme, Webservices und Hardware zur Verwendung.

5.2.1 Coda

Zum Haupteinsatz kommt die Software Coda von der Firma Panic, die für den Einsatz auf Mac OS X entwickelt wurde. Es ist ein leistungsstarker Texteditor. Coda beinhaltet nützliche Funktionen wie zum Beispiel:

- Aufsplitten des Texteditors in zwei Bereiche, um Dateien besser vergleichen zu können
- Lokale Indexierung, um Dateien einfach und schnell zu verlinken
- Local und Remote Verbindungen: Die Möglichkeit direkt oder indirekt Webseiten zu bearbeiten

Die Wahl fiel auf Coda, da es nur wenige leistungsstarke Editoren gibt, die auf dem Betriebssystem Mac OS X laufen.

5.2.2 GitLab

Der Webservice GitLab ist ein Tool zur Versionierung. Hierbei werden jegliche Änderungen, die an einem zuvor definierten Datensystem durchgeführt werden, mit Kommentaren abgespeichert. Dies ist besonders dann hilfreich, wenn Änderungen nachvollziehbar sein sollen. GitLab ist ein umfangreiches Tool, das häufig für Projekte benutzt wird, an denen mehrere Entwickler gemeinsam arbeiten.

Wird eine Änderung an einer Datei lokal vorgenommen, muss diese zunächst kommentiert werden und anschließend auf einen Server hochgeladen werden, auf dem eine Version von GitLab läuft. An Hand von Kommentaren ist es einem Dritten möglich, die Veränderung nachzuvollziehen.

GitLab stellt den Entwicklern verschiedene Funktionen zur Verfügung, die es vereinfachen sollen, zusammen zu arbeiten. Zum einen ist dies die Funktion *auto merge*: Hierbei werden Teilbereiche, an denen jeweils ein Entwickler arbeitet, nach der Fertigstellung automatisch zusammengefügt. Kommt es beim Zusammenfügen trotzdem zu Konflikten, meldet GitLab dies und fordert den Entwickler auf, die beiden Teilbereiche händisch zusammenzufügen. Eine

weitere nützliche Funktion ist die visuelle Darstellung der Unterschiede zwischen verschiedenen Änderungen an einer Datei. Lädt der Entwickler eine neue Version einer Datei auf den Workspace, so erkennt das Tool, welchen Bereich dieser verändert und / oder ergänzt hat. So ist es möglich, Änderungen direkt nachzuvollziehen.

Es gibt mehrere Möglichkeiten, GitLab einzusetzen. Die Firma GitLab stellt selber einen Workspace zur Verfügung, den man nutzen kann. Eine weitere Möglichkeit bietet das [LRZ](#) an. Dieses stellt einen eigenen GitLab Server zur Verfügung, den Studenten und Mitarbeiter der TU München nutzen können. Möchte man jedoch einen GitLab Server lieber selber betreiben, können die Installationsdateien für einen Server von der GitLab Webseite heruntergeladen werden.

5.2.3 LRZ Server

Damit die Webseite getestet und zugänglich gemacht werden kann, wird ein Webserver mit PHP und MySQL benötigt. Hierfür wird ein Server vom [LRZ](#) zur Verfügung gestellt. Auf diesem läuft ein Apache Webserver mit der Unterstützung für PHP. Für die notwendige Datenbank läuft auf dem Server MySQL Version 5.0.11. Die Datenbank auf dem Server ist mit dem Tool phpMyAdmin in der Version 4.0.10.10. erreichbar. Mit Hilfe dieses Tools ist es möglich, Änderungen an der Datenbank vorzunehmen.

5.3 Soll-Ist-Vergleich: Bauablaufplan - Datenbank

Für den Abgleich des 4D Modells mit dem Bauablaufplan wird ein Soll-Ist-Vergleich durchgeführt. Hierbei wurde wöchentlich der Stand der Baustelle mit der Datenbank abgeglichen und aktualisiert. Problematisch hieran ist jedoch die große Zeitspanne von einer Woche. Zum Beispiel kann ein Bauteil direkt nach dem letzten Einlesen fertiggestellt worden sein, wird aber auf Grund der großen Zeitspanne erst bei der nächsten Überprüfung erkannt und eingelesen. Dadurch können Differenzen im Soll-Ist-Vergleich entstehen. Um diese Abweichungen zu minimieren, wäre es sinnvoll, die Zeitabstände zwischen den Überprüfungen zu verkürzen.

Neben dem zeitlichen Aspekt gibt es ein weiteres Problem, das zu Abweichungen führen kann: Die Baustelle wird mittels einer Vielzahl von Bildern analysiert. Mit Hilfe dieser wird eine Punktwolke von der Baustelle erstellt. Problematisch hierbei ist, dass der Fortschritt der Baustelle nicht auf allen Bildern einwandfrei erkennbar ist. Dies ist der Fall, wenn die Sicht auf die Baustelle durch gelagertes Material oder andere externe Einflüsse gestört wird. Ebenfalls ist die Sicht in das Innere der Baustelle nur eingeschränkt möglich. Aus diesem Grund müssen Annahmen getroffen werden, um den Fortschritt einwandfrei protokollieren zu können.

5.4 Konzept und Überlegungen für die Programmierung der Webseite

Bevor eine Webseite programmiert werden kann, bedarf es eines grundlegenden Konzeptes. Ein Konzept besteht darin, sich zunächst Gedanken darüber zu machen, welche Funktionen Webseiten haben und wie sie in etwa aussehen sollen. In diesem Fall ist es das Importieren von Projektdaten in eine Datenbank, die aus einem CAD Programm zuvor exportiert werden müssen. Eine weitere Funktion, um die es hauptsächlich in der Bachelorthesis geht, ist die Darstellung der Projektdaten in einem Browser, die ebenfalls vorher importiert wurden. Als Darstellung dient hierbei ein 3D Modell.

Zusätzlich zum Konzept ist es wichtig, sich zu entscheiden, in welcher Programmiersprache die Webseite umgesetzt werden soll. Für dieses Projekt soll hauptsächlich die Programmiersprache PHP und als Datenbank Server MySQL verwendet werden.

Im Folgendem wird noch einmal genauer auf die Vorüberlegungen und das Vorgehen der beiden Hauptfunktionen eingegangen.

5.4.1 Import

Ein automatischer Import von Dateien ist nur dann möglich, wenn ein Schema in der vorliegenden Datei vorhanden ist. Ist ein solches Schema gefunden, kann die Datei mittels einer Abfragenschleife eingelesen und anschließend in die Datenbank importiert werden. Für die Abfragenschleife wird eine FOR Schleife verwendet, bei der die zuvor bestimmten Grenzen verwendet werden. Die Grenzen sind jeweils die Zeile des Dateianfangs und die Zeile des Dateiendes. Für den Import der eingelesenen Daten wird eine MySQLi Schnittstelle zur MySQL Datenbank verwendet.

5.4.2 Darstellung

Die Vorüberlegungen bezüglich der Darstellung des 3D Modells im Browser sind umfangreicher.

Zunächst wird im Internet nach bereits vorhandenen Bibliotheken gesucht, die die Umsetzung erleichtern können. Sind solche gefunden, müssen diese meist nur noch wenig angepasst werden, um sie für die Webseite nutzen zu können.

In diesem Fall können zwei fertige Bibliotheken für die zu programmierende Webseite verwendet werden. Für die Darstellung des Modells wird die Bibliothek three.js verwendet und

für die Darstellung des Bauablaufplanes wird OpenSource Software dhtmlxGantt (mehr Informationen siehe unter [5.6.2](#)) verwendet.

Während der Programmierung kommen immer neue Ideen bezüglich neuer Funktionen, die zusätzlich in die Webseite integriert werden.

5.5 Datenbank

Aktuell ist in der Datenbank nur ein Modell implementiert. Es besteht aber die Möglichkeit ein weiteres Modell in die Datenbank zu integrieren. Die Datenbank für das eine Modell enthält 10 Tabellen:

Tabellenname	Inhalt
building_components	Element ID, Objekt Name, Material Typ, Geschoss
floors	Benennung der Geschosse
gantt_links	Informationen über die Verknüpfung einzelner Bauabschnitte miteinander
gantt_tasks	Zeitplan
login	Benutzername samt Passwort zum Login
observations	Zeitpunkte der Erstellung der Fotos
processes	Daten für den Bauablaufplan
triangles	Informationen über die Dreiecke
vertices	Informationen über die Koordinaten der Dreiecke
triangle_detection	Informationen, zu welchem Zeitpunkt welches Element erkannt wurde. Siehe auch Tabelle Observation

Tabelle 5.1: Beschreibung der Tabellen in der Datenbank

Bei der Erstellung der Datenbank wurde darauf geachtet, dass die Tabellen möglichst klein gehalten werden. Die Idee dabei ist: Lieber mehr Tabellen als zu große und zu umfangreiche Tabellen. Das bedeutet zwar eine größere Anzahl an Tabellen, jedoch ist die Übersicht in den einzelnen Tabellen besser. Die Beziehungen der einzelnen Tabellen untereinander ist auf folgendem Diagramm ([Abbildung 5.1](#)) erkennbar:

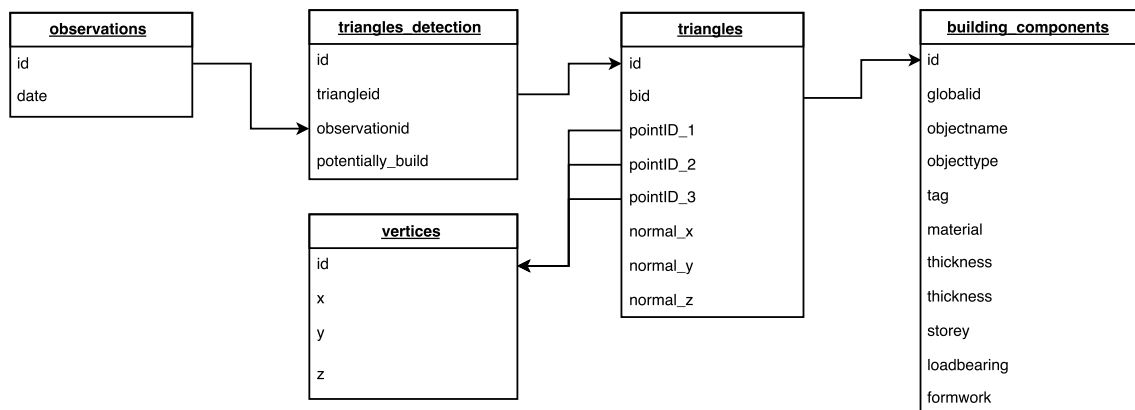


Abbildung 5.1: Beziehungen der Tabellen in der Datenbank

5.6 Skript

Das Projekt lässt sich in zwei Bereiche gliedern: Den Import von neuen Datensätzen in die Datenbank und die Darstellung des jeweiligen Modells. Die Programmierung findet in der Sprache PHP statt und die Daten für das Modell liegen in einer MySQL Datenbank.

5.6.1 Import von Dateien

Das Skript für das Importieren der Datei muss für jeden neuen Import angepasst werden. Dies hängt damit zusammen, dass die zur Verfügung gestellte Datei anders aufgebaut sein kann. Wird immer ein und dasselbe Dateiformat verwendet und ändert sich der Aufbau der Datei nicht, muss das Import Skript nicht umgeschrieben werden.

Zunächst werden zwei neue Tabellen in der Datenbank erstellt. Eine, die die Koordinaten in x-,y-,z-Richtung beinhaltet, und eine, die die IDs samt Beschreibung und die PointIDs der Dreiecke enthält. Die Koordinaten aus der ersten Tabelle werden über die PointIDs mit der zweiten Tabelle verknüpft (siehe Datenbank Diagramm). Dies ist aus dem Grund notwendig, da jedes Dreieck aus drei Punkten besteht, die wiederum jeweils eine Koordinate in x-,y- und z-Richtung haben. Für die Übersichtlichkeit und die Größe der Tabellen wurden diese Informationen getrennt.

Für jede Tabelle bedarf es einer Datei. Jede Datei wird separat importiert. Dafür wird die Datei mit folgendem Befehl in einen Array eingelesen:

Datei einlesen

```
1 $datei = file("triangles.txt");
```

Anschließend werden die Zeilen über eine Schleife ausgegeben. Da die Daten in einem Array gespeichert sind, lassen sich die Grenzen der Schleife sehr einfach herausfinden. Die untere ist der Wert 0 und die obere Grenze stellt die Größe des Arrays dar. Da bisher nur eine Einteilung in Zeilen stattgefunden hat, müssen die einzelnen Werte mit der PHP Funktion *explode* herausgefiltert werden. Hierbei wird jede Zeile in ihre Einzelteile zerlegt, so dass jeder Wert einzeln identifizierbar ist und über einen SQL Befehl in die Datenbank importiert werden kann:

Daten importieren

```
1 $sql = "INSERT INTO triangles (id, componentID, pointID_1, pointID_2,
      pointID_3, Normal_x, Normal_y, Normal_z) VALUES ($id, $componentID,
      $pointID_1, $pointID_2, $pointID_3, $Normal_x, $Normal_y, $Normal_z)";
2 $mysqli->query($sql);
```

Die zweite Tabelle wird nach einem ähnlichen Schema bestückt. Anders ist hier nur, dass eine andere Datei als Quelle verwendet wird und die Daten in die Tabelle *vertices* importiert werden (siehe Tabelle 5.1. und Abbildung 5.1.).

5.6.2 Darstellung des Modells

Sehr viel umfangreicher als der Import der Werte in die Datenbank ist die Darstellung des Modells. Hierfür wird neben PHP und MySQL auch JavaScript und die fertige Bibliothek *three.js Library* verwendet. Besonders wichtig bei der Darstellung ist es, dass die Webseite auch von Smartphones oder Tablets einwandfrei aufgerufen werden kann.

Die Darstellung der Webseite gliedert sich in mehrere Bereiche:

Exportieren der Daten in eine Datei

Für die Darstellung des Modells gibt es drei Möglichkeiten:

Variante 1 Die Darstellung vom kompletten Modell, wie es in der Datenbank gespeichert ist

Variante 2 Die Darstellung eines bestimmten Bauteils. Wobei das Bauteil vom restlichen Modell besonders hervorgehoben wird

Variante 3 Die Darstellung zu einem bestimmten Zeitpunkt.

Eine der Varianten kann durch ein Formular ausgewählt werden. Für jede Variante werden die Daten aus der Datenbank automatisch durch ein PHP Skript in gesonderte Dateien exportiert. Dies ist notwendig, da der Darstellungsprozess schneller abläuft, wenn die Informationen in einer Datei zwischengespeichert sind, als wenn sie direkt aus der Datenbank ausgelesen werden. Die Daten werden so exportiert, dass das Schema dem Format STL entspricht.

Der Export ist etwas komplexer, da darauf geachtet werden muss, dass es bei der späteren Darstellung des Modells zu keiner Überschneidung der einzelnen Elemente kommt. Besonders ist hierbei der Export für die zeitliche Darstellung zu nennen.

Für eine bessere Übersicht in dem Skript wird der eben beschriebene Export Prozess in eine gesonderte Datei ausgelagert (`webgl1_export.php`) und mittels PHP Befehl eingebunden.

Mobile Responsive Design

Damit die Webseite von einem mobilen Endgerät einwandfrei genutzt werden kann, wird im nächsten Abschnitt der Datei eine Abfrage eingebaut, die überprüft, ob die Webseite mit einem solchen Gerät aufgerufen wird:

Abfrage Device

```
1 if (preg_match("/( alcatel|amoi|android|avantgo|blackberry|benq|cell |
   cricket|docomo|elaine |htc |iemobile|iphone|ipad|ipaq|ipod|j2me|java|
   midp|mini|mmp|mobi|motorola|nec-|nokia|palm| panasonic|philips|phone|
   playbook|sagem|sharp|sie-|silk|smartphone|sony|symbian|t-mobile|telus
   |up\\.browser|up\\.link|vodafone|wap|webos|wireless|xda|xoom|zte)/i",
   $_SERVER['HTTP_USER_AGENT']))
2 {
3     $handy = true;
4 }
```

Ruft ein Nutzer die Webseite auf, kann mit der Variablen `$_[HTTP_USER_AGENT]` ausgelesen werden, von welchem Gerät der Nutzer die Webseite aufruft. Merkt das Skript, dass die Webseite von einem mobilen Endgerät aufgerufen wird, werden gewisse Elemente ausgeblendet. Damit dennoch der volle Funktionsumfang gegeben ist, kann der Nutzer diese Elemente über einen Button wieder einblenden. Das Einblenden geschieht über JavaScript, damit die Webseite nicht neu aufgerufen werden muss:

JavaScript für das Einblenden der Elemente

```
1 <script>
2     function Toggle( id ){
3         var e = document.getElementById(id);
4         if(e.style.display == 'block'){e.style.display = 'none';}
5         else {e.style.display = 'block';}
6     }
7 </script>
```

Mit diesem Skript können die Elemente auch auf dem Computer ausgeblendet oder eingeblendet werden.

Zusätzlich wurde darauf geachtet, dass die Webseiten und die darauf abgebildeten Elemente von der Breite her dynamisch sind. Wird die Breite der Webseite geändert, wird ein JavaScript Programm aktiv und passt die Elemente auf der Webseite an.

Ähnlich wie bei dem Export Prozess werden die oben genannten Funktionen ebenfalls in eine gesonderte Datei ausgelagert (`webgl1_style.php`) und anschließend wieder eingebunden.

Formular

Wie bereits beim Export der Informationen erwähnt, kann die Darstellung über ein Formular beeinflusst werden. Ein Ausschnitt der Webseite, auf dem das Formular erkennbar ist, zeigt die [Abbildung 5.2](#).

In der ersten Spalte der Webseite kann der Nutzer zwischen allen Bauteilen auswählen und sich diese im Gesamtmodell anzeigen lassen. Gibt der Nutzer ein Datum ein, wird das Gebäude so dargestellt, wie der Baufortschritt zu jenem Zeitpunkt ist.

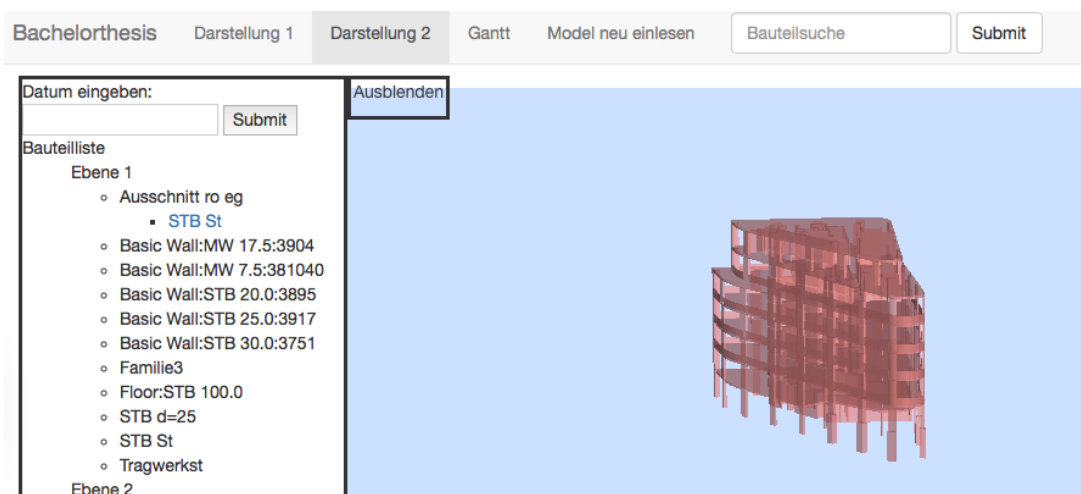


Abbildung 5.2: Formular zur Anpassung der Darstellung

Darstellung

Die Darstellung wird mit Hilfe der three.js Bibliothek realisiert. Für den Zugriff auf die Bibliothek muss diese zunächst eingebunden werden, anschließend erfolgen verschiedene Deklarationen von Variablen. Um ein Modell darstellen zu können, muss eine *Scene* erstellt werden, einschließlich Breite und Höhe der Ebene, auf der das Modell dargestellt werden soll.

Erstellung der *Scene*

```
1 this.scene = new THREE.Scene();
2 this.scene.fog = new THREE.FogExp2(0xcce0ff, 0.003);
3
4 var SCREEN_HEIGHT = window.innerHeight - 350;
5 var SCREEN_WIDTH = window.innerWidth - 550;
```

Mit dem Befehl `window.innerHeight` und `window.innerWidth` wird die Größe des Browsers ermittelt, um die *Scene* vollständig darzustellen. Der Abstand, der horizontal und vertikal abgezogen wird, wird für den Gantt Chart und das seitliche Menü gebraucht.

Zusätzlich zur Definition der Größe der *Scene* ist die Positionierung der Blickrichtung notwendig:

Positionierung Kamera

```
1 var VIEW_ANGLE = 10;
2 var ASPECT = SCREEN_WIDTH / SCREEN_HEIGHT;
3 var NEAR = 1;
4 var FAR = 20000;
5
6 this.camera = new THREE.PerspectiveCamera(VIEW_ANGLE, ASPECT, NEAR, FAR);
7 this.scene.add(this.camera);
8 this.camera.position.set(-40, 10, -5);
9 this.camera.lookAt(new THREE.Vector3(0,0,0));
```

Um das Modell sichtbar zu machen, muss zudem eine Lichtquelle definiert werden. Dies geschieht mit dem Befehl: `THREE.AmbientLight`. Das eigentliche Modell wird mittels eines STL Loader implementiert. Dieser liest die angegebene STL Datei aus und stellt sie je nach Einstellungen dar:

STL Loader

```
1 var oStlLoader2 = new THREE.STLLoader();
2
3 oStlLoader2.load('models/export_datei2.stl', function(geometry)
4 {
5     var meshMaterial2 = new THREE.MeshPhongMaterial( { color:0xffd760,
6         transparent:true, opacity:1.0} );
7     var mesh2 = new THREE.Mesh(geometry, meshMaterial2);
8
9     mesh2.rotation.set( - Math.PI / 2, 0, 0);
10    mesh2.position.set(-2, 0, -2);
11    mesh2.scale.set(0.10001, 0.10001, 0.10001);
12
13    lesson7.scene.add(mesh2);
14 }
```

Da für die Darstellung des Modells unterschiedliche Ebenen notwendig sind, muss für jede STL Datei ein Abschnitt hinzugefügt werden, der mit Bedingungen verbunden ist, um so nur die benötigten STL Dateien darzustellen.

three.js bietet die Möglichkeit, mittels einer GUI Box die Modelle beziehungsweise auch die *Scene* zu modifizieren, während das Modell angezeigt wird. So ist es möglich, die Farbe oder die Transparenz anzupassen, um gezielt Bereiche besser sichtbar zu machen. Bei der Programmierung wurde für jede Ebene eine solche Möglichkeit geschaffen. Hierbei ist es möglich, die Farbe oder die Transparenz zu verändern. Des Weiteren kann der Nutzer entscheiden, ob die jeweilige Ebene angezeigt werden soll. Zu guter Letzt kann er entscheiden, ob und wie schnell sich das Modell drehen soll.

Um das Modell darzustellen, müssen die Informationen des kompletten Modells umgesetzt werden.

Erstellen des Modells

```
1 // Animate the scene
2 function animate()
3 {
4     requestAnimationFrame(animate);
5     render();
6     update();
7 }
8 // Update controls and stats
9 function update()
10 {
11     lesson7.controls.update(lesson7.clock.getDelta());
12 }
```

```
13 |
14 | // Render the scene
15 | function render()
16 | {
17 |     if (lesson7.renderer){
18 |         lesson7.renderer.render(lesson7.scene , lesson7.camera);
19 |     }
20 | }
21 |
22 | // Initialize lesson on page load
23 | function initializeLesson()
24 | {
25 |     lesson7.init();
26 |     animate();
27 | }
```

Gantt Chart

Für die Darstellung des Bauablaufplans (Gantt Chart) wurde ein OpenSource Programm verwendet, das auf JavaScript basiert: *dhtmlxGantt*. Es wurde vom *DHTMLX engineering team* entwickelt und steht aktuell in der Version 3.2 zur Verfügung.

Ähnlich wie die Bibliothek von *three.js* muss das JavaScript Programm ebenfalls eingebunden werden. Um das Thema Mobile Responsive Design wieder aufzugreifen, wurde die Webseite so programmiert, dass, wenn sich ein Nutzer mit dem Smartphone einloggt, der Gantt Chart zunächst ausgeblendet ist. Über einen Button kann dieser jedoch wieder eingeblendet werden. Die Einstellungen zu dem Gantt Chart erfolgen in JavaScript:

Einstellungen Gantt Chart

```
1 | gantt.config.xml_date = "%Y-%m-%d %H:%i:%s";
2 | gantt.config.task_height = 16;
3 | gantt.config.row_height = 40;
4 | gantt.config.grid_width = 500;
5 | gantt.config.columns = [
6 |     {name:" text",          width:"220", tree:true },
7 |     {name:" start_date",   align: "center" },
8 |     {name:" end_date",     align: "center" }];
9 | gantt.config.scale_unit = "month";
10 | gantt.config.date_scale = "%F, %Y";
11 | gantt.config.scale_height = 50;
12 | gantt.config.subscales = [
13 |     {unit:" day", step:1, date:"%j, %D" }];
```

Die Daten werden über eine vorgegebene Schnittstelle aus der Datenbank ausgelesen und dargestellt:

Ausgeben der Daten

```
1 gantt.load("gantt/include/connector.php");  
2 var dp = new dataProcessor("gantt/include/connector.php");
```

Wie auch andere Teile des Skripts wird das Gantt Chart ebenfalls ausgelagert und anschließend wieder eingebunden (`webgl_gantt.php`).

5.7 Dateiarchitektur

Bei der Dateiarchitektur wurde, wie schon weiter oben erwähnt, darauf geachtet, dass die Webseite mit jeglichen Modellen benutzt werden kann. Aus diesem Grund wurde darauf geachtet, dass eine klare und übersichtliche Ordnerstruktur aufgebaut wurde: Daten wie CSS Dateien oder Konfigurationsdateien sind im *Includes* Ordner. Die ausgelagerten Skriptbausteine liegen ebenfalls im *Includes* Ordner. Die Dateien bezüglich des Gantt Charts stehen in einem separaten Ordner. Die Dateien, die beim Export zwischenzeitlich abgespeichert werden, sind im Models Ordner abgelegt. Zusätzlich gibt es einen Backup Ordner, in dem die Datenbank gesichert wird, bevor ein neues Modell eingebunden wird.

5.8 Gerätekompatibilität und Rechenleistung

Für die Darstellung des Modells ist die Schnittstelle WebGL notwendig. Diese ist in den neueren Browsern komplett enthalten. Die Webseite <http://www.khronos.org/> listet alle kompatiblen Browser auf. Der Internet Explorer unterstützt WebGL seit Version 11. Für die Browser jüngerer Generationen muss ein Zusatzpaket installiert werden. Google Chrome unterstützt WebGL seit Version 9.x. Safari, der firmeneigene Browser von Apple, unterstützt WebGL seit Version 5.1. Einzige Einschränkung hier: WebGL ist nur in der Safari Version auf dem Mac integriert. Safari für Windows unterstützt WebGL nicht. Mozilla Firefox unterstützt WebGL in der aktuellen Version.

Auf dem Smartphone unterstützen die Browser Safari (iOS), Google Chrome und Firefox (Android) in der neuesten Version WebGL.

Die Möglichkeit der Darstellung hängt nicht nur von der Wahl des Browsers ab, sondern auch von der Leistung des gewählten Endgerätes. Getestet wird die Webseite auf verschiedenen Geräten. Die Testergebnisse werden in einer Tabelle zusammengefasst (siehe [Tabelle 5.2](#)).

Für einen Schnelltest kann die Webseite <http://get.webgl.org> aufgerufen werden. Wird auf dieser Seite ein drehendes Modell angezeigt, unterstützt der Browser WebGL.

Betriebssystem	Browser	Arbeitsspeicher [GB]	CPU [GHz]	Bemerkung
Windows XP	Firefox	2	2,66	Darstellung nicht möglich
Android Tablet	Chrome	k.A.	k.A.	Darstellung nicht möglich
Windows 7	Firefox	8	3,36	Darstellung möglich
Mac OS X 10.10.4	Google Chrome	8 GB	2,66	Darstellung möglich
iOS 9	Safari	1 GB	A8 Chip mit 64-Bit Architektur	Darstellung möglich

Tabelle 5.2: Zusammenfassung der Gerätetests

Die Tatsache, dass die Webseite auf einem Windows XP Rechner mit Firefox Version 41.0.1 nicht lauffähig ist, liegt daran, dass die verwendete Hardware zu alt ist und die Version von Firefox WebGL noch nicht unterstützt. Ebenfalls ist die Leistung und Hardware des Android Tablet nicht kompatibel mit der Webseite.

Bei der Programmierung der Webseite wurde darauf geachtet, dass die Webseite von allen Endgeräten - solange sie aktuelle Hardware und Software verwenden - benutzbar ist.

5.9 Zusammenfassung

Dieses Kapitel beschreibt, wie die Web Applikation programmiert wird. Es ist erkennbar, wie komplex und umfangreich das Einlesen von Daten ist, um diese endgültig darstellen zu können. Ebenfalls ist ersichtlich, dass die Web Applikation einige wenige Systemanforderungen benötigt, um einwandfrei lauffähig zu sein.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In der Bachelorthesis wird eine neue Art und Weise der Bauüberwachung dargestellt. Ein Bauwerk wird mittels Photogrammetrie eingelesen und anschließend in einer Datenbank abgespeichert. Notwendig hierfür sind Fotos, die in regelmäßigen Abständen gemacht werden müssen. Mit diesen Fotos wird eine Punktwolke erstellt, mit der es wiederum möglich ist, ein 3D Modell am Computer zu erstellen. Anschließend wird das Modell mittels eines Web Tools, das für die Bachelorthesis programmiert wird, in die Datenbank eingelesen. Durch die Fotos, die zu unterschiedlichen Zeitpunkten gemacht werden, ist es möglich, den Baufortschritt darzustellen. Da der Bauablaufplan in der Datenbank ebenfalls gespeichert und in die Webseite integriert ist, können Abweichungen im Ablauf visuell dargestellt werden.

Genaueres Augenmerk wird neben der Umsetzung des Web Tools auf die Theorie, die hinter der Bauüberwachung und der Umsetzung steht, gelegt. Dabei wird genauer auf den Bauablaufplan eingegangen. Wesentliche Punkte hierbei sind die verschiedenen Arten von Terminplänen und der Soll-Ist-Vergleich in der Bauüberwachung.

Für die Umsetzung werden die benutzten Programmiersprachen genannt und kurz beschrieben. Neben der Beschreibung ist bei jeder Programmiersprache ein kurzes Beispiel des Skripts angefügt. Dadurch ist es einfacher, die Umsetzung in [Kapitel 5](#) nachzuvollziehen.

Die Webseite wird so umgesetzt, dass die Benutzung der Webseite keinerlei Vorkenntnisse bedarf. Die Seite ist intuitiv bedienbar und somit von jedem Nutzer einsetzbar. Lediglich für den Import neuer Datensätze bedarf es gewisser Kenntnisse in PHP und MySQL, da das Import Skript gegebenenfalls angepasst werden muss, um die Daten erfolgreich in die Datenbank übernehmen zu können.

6.2 Fazit

Die Idee, eine Bauüberwachung digital durchzuführen, ist keine Vision mehr. Solange aber keine Möglichkeit gefunden wird, Bilder vom Inneren mit einer Art Röntgenblick durchzuführen zu können, ist diese hier umgesetzte Möglichkeit nur auf den Rohbau anwendbar. Trotzdem kann und soll diese Webseite weiterhin eingesetzt werden, da es bei Großbaustellen bereits im Rohbau zu Baumängeln und Verzögerungen kommen kann. Dass diese Webseite, wie möglicherweise viele andere Webseiten auch, noch ausbaufähig ist, zeigt der folgende Abschnitt.

6.3 Ausblick und Vision

Die Seite stellt den Anfang einer neuen Möglichkeit dar, eine Baustelle zu überwachen. Da sie auf PHP und MySQL basiert, kann sie vielseitig ausgebaut werden. Dies könnte zum Beispiel ein einfacheres Einpflegen der Daten in der Datenbank oder auch die Optimierung der Performance sein. Ebenfalls kann weiter an der Lauffähigkeit auf Endgeräten gearbeitet werden

6.3.1 Einpflegen der Daten

Aktuell kann es passieren, dass das PHP Skript angepasst werden muss, damit neue Datensätze eingespielt werden können. Das Skript könnte in Zukunft so verändert werden, dass es jegliche Dateiformate kennt und so je nach Art und Weise die Daten richtig importiert.

Ebenfalls muss aktuell der Bauablaufplan händisch übertragen werden. Dies liegt daran, dass keine digitale Version von diesem zur Verfügung steht. Zukünftig könnte es möglich sein, dass es eine Schnittstelle zwischen der Software, mit der der Ablaufplan geschrieben wird, und der Webseite gibt, um so immer den aktuellsten Ablaufplan in der Datenbank zu haben. Gegebenenfalls wäre es auch möglich, den Ablaufplan komplett auf der Webseite zu erstellen.

Ein weiterer Punkt ist die Aufnahme der Daten auf der Baustelle. Diese könnten zukünftig beispielsweise mittels einer Drohne automatisch aufgenommen werden und anschließend direkt in die Datenbank importiert werden.

6.3.2 Optimierung der Performance

Aktuell wird das gesamte Modell in den Arbeitsspeicher des Computers geladen, um so das Modell einwandfrei darstellen zu können. Zukünftig wäre es vielleicht denkbar, die Rechenleistung der Server zu verwenden. So könnten auch leistungsschwächere Computer oder mobile Geräte die Webseite noch einfacher nutzen, da nicht mehr so viel Rechenleistung auf den Endgeräten notwendig wäre.

6.3.3 Lauffähigkeit auf Endgeräten

Die Tabelle 5.2 zeigt eine Übersicht des Hardware Tests, der im Zusammenhang mit der Umsetzung des Web Tools durchgeführt wurde.

Ein Umprogrammieren der Seite könnte dazu führen, dass das Web Tool wirklich auf allen Endgeräten lauffähig ist.

Anhang A

Datenträger

Auf dem beigefügten Datenträger befinden sich folgende Unterlagen:

- Die vorliegende Thesis [PDF]
- Dateien der Webseite (komprimiert) [zip]
- Abbild der Datenbank [sql]

Literaturverzeichnis

- Amann, J., Borrmann, A., Hegemann, F., Jubierre, J. R., Flurl, M., Koch, C. & König, M. (2013). A Refined Product Model for Shield Tunnels Based on a Generalized Approach for Alignment Representation. In: *Proc. of the ICCBEI 2013*, Tokyo and Japan.
- Bootstrap (2015). Bootstrap · The world's most popular mobile-first and responsive front-end framework. <http://getbootstrap.com/>.
- Braun, A. (2013). Entwicklung eines 4D-BIM-Viewers mit graphbezogener Darstellung von Bauabläufen und - alternativen.
- BuildingSMART (2015). Besser Planen mit BIM. <http://www.buildingsmart.de/>.
- Deutsches Institut für Normung & Ausschuss Netzplantechnik und Projektmanagement (1980). *Netzplantechnik: DIN 69900 ; Teil 1 - 2 = Project network techniques*. Berlin [u.a.]: Beuth.
- Egger, M., Hausknecht, K., Liebich, T. & Przybylo, J. (2013). BIM-Leitfaden für Deutschland.
- Ennex Corporation (1999). The StL Format. http://www.fabbers.com/tech/STL_Format.
- Khronos (2015). WebGL - OpenGL ES 2.0 for the Web. <https://www.khronos.org/webgl/>.
- Sagerschnig, C. (2006). Virtuelle Behaglichkeit. *Diplomarbeit: Das Produktdatenmodell der Industry Foundation Classes (IFC) in der thermischen Gebäudesimulation am Beispiel RIUSKA*.
- Theis, T. (2013). *Einstieg in JavaScript* (1 Aufl.). Bonn: Galileo Computing.
- Theis, T. (2014). *Einstieg in PHP 5.6 und MySQL 5.6: [ideal für Programmieranfänger geeignet ; schnell und einfach dynamische Webseiten entwickeln ; mit vielen Beispielprojekten und Übungsaufgaben ; Formulardaten auswerten und speichern, Sessions, Sicherheit, Grafiken, Aja* (10., aktua Aufl.). Galileo Computing. Bonn: Galileo Press.
- Three.js (2015). three.js - Javascript 3D library. <http://threejs.org/>.
- Zanner, C. (2014). *Rechte aus gestörtem Bauablauf nach Ansprüchen*. Bau- und Architektenrecht nach Ansprüchen. Wiesbaden: Springer Fachmedien Wiesbaden.

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorthesis 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, 14. Oktober 2015

Barth, Alexander

Barth, Alexander

████████████████████

████████████████████

██